aws

Panduan Developer

Amazon DocumentDB

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

Amazon DocumentDB: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin atau mungkin tidak berafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Amazon DocumentDB 1
Gambaran Umum 1
Klaster 3
Instans 4
Wilayah dan AZ 7
Wilayah 7
Zona Ketersediaan 7
Harga 9
Uji coba gratis 10
Pemantauan 10
Antarmuka 10
AWS Management Console 10
AWS CLI 10
Shell mongo 11
Driver MongoDB 11
Apa Selanjutnya? 11
Cara Kerjanya 11
Titik Akhir Amazon DocumentDB 13
Dukungan TLS 17
Penyimpanan Amazon DocumentDB 17
Replikasi Amazon DocumentDB 18
Keandalan Amazon DocumentDB 18
Opsi Preferensi Baca 20
Penghapusan TTL 24
Sumber Daya Dapat Ditagih 25
Apa itu Basis Data Dokumen? 28
Kasus Penggunaan 28
Memahami Dokumen 29
Bekerja dengan Dokumen 35
Panduan Memulai 48
Prasyarat 49
Langkah 1: Buat AWS Cloud9 lingkungan 50
Langkah 2: Membuat grup keamanan 51
Langkah 3: Buat klaster Amazon DocumentDB 54

Langkah 4: Instal shell mongo 56
Langkah 5: Buat koneksi ke klaster Amazon DocumentDB Anda 57
Langkah 6: Masukkan dan ajukan kueri data 59
Langkah 7: Jelajahi 61
Mulai cepat menggunakan AWS CloudFormation 62
Prasyarat 63
Izin IAM yang Diperlukan 63
Pasangan Kunci Amazon EC2 65
Meluncurkan Tumpukan AWS CloudFormation Amazon DocumentDB 65
Mengakses Klaster Amazon DocumentDB 70
Perlindungan Penghentian dan Perlindungan Penghapusan 71
Kompatibilitas MongoDB 72
MongoDB 5.0 Kompatibilitas 72
Apa yang baru di Amazon DocumentDB 5.0 72
Memulai Amazon DocumentDB 5.0 73
Meningkatkan atau Memigrasi ke Amazon DocumentDB 4.0 74
Perbedaan Fungsional 74
Kompatibilitas MongoDB 4.0 75
Fitur Amazon DocumentDB 4.0 76
Memulai dengan Amazon DocumentDB 4.0 77
Meningkatkan atau Memigrasi ke Amazon DocumentDB 4.0 78
Perbedaan Fungsional 78
Transaksi 80
Persyaratan 80
Praktik Terbaik 81
Keterbatasan: 81
Pemantauan dan Diagnosis 82
Tingkat Isolasi Transaksi 83
Kasus Penggunaan 83
Transaksi Multi-Pernyataan 83
Transaksi Multi-Koleksi 85
Contoh API Transaksi untuk API Panggilan Balik 87
Contoh API Transaksi untuk API Inti 87
Perintah yang Didukung 121
Kemampuan yang Tidak Didukung 121
Sesi 122

Konsistensi kausal ... 122
Penulisan yang dapat dicoba lagi ... 123
Kesalahan Transaksi ... 124
Praktik Terbaik ... 125
Pedoman Operasional Dasar ... 125
Pengukuran Instans ... 126
Bekerja dengan Indeks ... 128
Membangun Indeks ... 128
Selektivitas Indeks ... 128
Dampak Indeks pada Penulisan Data ... 129
Mengidentifikasi Indeks yang Hilang ... 129
Mengidentifikasi Indeks yang Tidak Terpakai ... 130
Praktik Terbaik Keamanan ... 130
Pengoptimalan Biaya ... 131
Menggunakan Metrik untuk Mengidentifikasi Masalah Kinerja ... 131
Melihat Metrik Kinerja ... 131
Mengatur CloudWatch Alarm ... 132
Mengevaluasi Metrik Kinerja ... 132
Penyetelan Kueri ... 134
Beban Kerja TTL dan Deret Waktu ... 134
Migrasi ... 135
Bekerja dengan Grup Parameter Klaster ... 135
Kueri Alur Agregasi ... 135
`batchInsert` dan `batchUpdate` ... 135
Perbedaan fungsional pada MongoDB ... 137
Manfaat fungsional dari Amazon DocumentDB ... 137
Transaksi Implisit ... 137
Perbedaan Fungsional yang Diperbarui ... 138
Pengindeksan Array ... 139
Indeks Multi-kunci ... 140
Karakter Null dalam String ... 141
Kontrol Akses Berbasis Peran ... 141
`$regex` Pengindeksan ... 141
Proyeksi untuk Dokumen Bersarang ... 142
Perbedaan fungsional pada MongoDB ... 142
Operator `$vectorSearch` ... 143

- `OpCountersCommand` ... 143
- Basis Data Admin dan Koleksi ... 143
- `cursormaxTimeMS` ... 143
- jelaskan() ... 144
- Pembatasan Nama Bidang ... 144
- Pembuatan indeks ... 145
- Pencarian dengan kunci kosong di jalur ... 145
- API, Operasi, dan Tipe Data MongoDB ... 145
- Utilitas `mongodump` dan `mongorestore` ... 145
- Pengurutan Hasil ... 146
- Penulisan yang dapat dicoba lagi ... 146
- Indeks Jarang ... 147
- Menggunakan $elemMatch Dalam Ekspresi $all ... 147
- Pengindeksan `$ne`, `$nin`, `$nor`, `$not`, `$exists`, dan `$elemMatch` ... 148
- `$lookup` ... 148

API, Operasi, dan Tipe Data MongoDB Yang Didukung ... 153

- Perintah Basis Data ... 153
  - Perintah administratif ... 154
  - Agregasi ... 155
  - Autentikasi ... 156
  - Perintah diagnostik ... 156
  - Operasi Kueri dan Tulis ... 157
  - Perintah Pengelolaan Peran ... 158
  - Perintah Sesi ... 159
  - Pengelolaan Pengguna ... 159
  - Perintah Sharding ... 160
- Kueri dan Proyeksi Operator ... 162
  - Operator Array ... 162
  - Operator Bitwise ... 162
  - Operator Komentar ... 163
  - Operator Perbandingan ... 163
  - Operator Elemen ... 164
  - Operator kueri evaluasi ... 164
  - Operator Logis ... 164
  - Operator proyeksi ... 165
- Operator Pembaruan ... 165

Operator Array ... 165
Operator Bitwise ... 166
Operator Bidang ... 166
Pembaruan Pemodifikasi ... 167
Geospasial ... 167
Penentu Geometri ... 167
Pemilih Kueri ... 168
Metode kursor ... 168
Operator Agregasi Pipeline ... 170
Ekspresi Akumulator ... 171
Operator aritmatika ... 172
Operator Array ... 173
Operator Boolean ... 174
Operator Perbandingan ... 174
Operator Ekspresi Bersyarat ... 174
Operator Tipe Data ... 175
Operator Ukuran Data ... 175
Operator Tanggal ... 175
Operator Literal ... 176
Operator Penggabungan ... 177
Operator Alami ... 177
Operator Set ... 177
Operator Tahap ... 178
Operator String ... 179
Variabel sistem ... 180
Operator Pencarian Teks ... 181
Operator Konversi Tipe ... 181
Operator variabel ... 182
Operator Lain-lain ... 182
Jenis Data ... 182
Indeks dan Properti Indeks ... 184
Indeks ... 184
Properti Indeks ... 184
AI generatif ... 186
SageMaker Kanvas ... 186
Cara membuat model MLtanpa kode dengan Canvas SageMaker ... 186

Mengkonfigurasi SageMaker domain dan profil pengguna 187
Mengkonfigurasi izin akses IAM untuk Amazon DocumentDB dan Canvas SageMaker 187
Membuat pengguna database dan peran untuk SageMaker Canvas 188
Wilayah yang tersedia 188
Pencarian Vektor 189
Memasukkan vektor 189
Membuat indeks vektor 190
Mendapatkan definisi indeks 194
Menanyakan vektor 195
Fitur dan keterbatasan 199
Praktik terbaik 201
Migrasi ke Amazon DocumentDB 202
Migrasi Antar Versi 202
Langkah 1: Aktifkan Streaming Perubahan 203
Langkah 2: Ubah Durasi Retensi Streaming Perubahan 203
Langkah 3: Migrasikan Indeks Anda 204
Langkah 4: Buat Instance AWS DMS Replikasi 205
Langkah 5: Buat AWS DMS Endpoint Sumber 208
Langkah 6: Buat Endpoint AWS DMS Target 210
Langkah 7: Buat dan jalankan tugas migrasi 212
Langkah 8: Mengubah titik akhir aplikasi ke cluster Amazon DocumentDB target 214
Alat Migrasi 214
AWS Database Migration Service 214
Utilitas Baris Perintah 215
Penemuan 215
Perencanaan: Persyaratan Klaster Amazon DocumentDB 219
Pendekatan Migrasi 222
Offline 222
Online 224
Hibrida 226
Sumber Migrasi 228
Konektivitas Migrasi 228
Pengujian 231
Pertimbangan Pengujian Rencana Migrasi 232
Pengujian Performa 235
Pengujian Failover 236

Sumber Daya Tambahan 236
Buku pedoman migrasi 236
Proses migrasi 236
Sumber daya tambahan 241
Memutakhirkan Versi Mesin Amazon DocumentDB 242
Prasyarat dan batasan 242
Praktik terbaik untuk peningkatan versi utama di tempat 245
Uji peningkatan versi utama di tempat menggunakan cluster kloning 245
Sebelum upgrade versi utama di tempat 246
Selama upgrade versi utama di tempat 247
Setelah peningkatan versi utama di tempat 249
Melakukan upgrade versi utama di tempat 250
Perbedaan antara Amazon DocumentDB 3.6/4.0 hingga 5.0 cluster yang ditingkatkan dan cluster Amazon DocumentDB 5.0 baru 253
Memecahkan masalah upgrade versi utama di tempat 253
Keamanan 255
Perlindungan Data 256
Enkripsi di sisi klien 257
Mengenkripsi Data at Rest 265
Mengenkripsi Data dalam Transit 270
Manajemen kunci 281
Identity and Access Management 281
Audiens 282
Mengautentikasi dengan identitas 282
Mengelola akses menggunakan kebijakan 286
Bagaimana Amazon DocumentDB bekerja dengan IAM 289
Contoh kebijakan berbasis identitas 297
Pemecahan Masalah 300
Mengelola Izin Akses ke Sumber Daya Amazon DocumentDB Anda 302
Menggunakan Kebijakan Berbasis Identitas (Kebijakan IAM) 307
AWS kebijakan terkelola untuk Amazon DocumentDB 312
Referensi Izin Amazon DocumentDB API 330
Mengelola Pengguna Amazon DocumentDB 339
Primer dan`serviceadmin` Pengguna 339
Membuat Pengguna Tambahan 340
Kata Sandi Berputar Otomatis 342

Kontrol Akses Berbasis Peran ... 343
Konsep RBAC ... 344
Memulai dengan peran bawaan RBAC ... 346
Memulai dengan peran yang ditentukan pengguna RBAC ... 349
Menghubungkan ke Amazon DocumentDB sebagai Pengguna ... 354
Perintah Umum ... 356
Perbedaan Fungsional ... 361
Batas ... 361
Akses Database Menggunakan Kontrol Akses Berbasis Peran ... 361
Pencatatan dan Pemantauan ... 370
Memperbarui Sertifikat ... 370
Memperbarui Aplikasi dan Klaster Amazon DocumentDB Anda ... 371
Pemecahan Masalah ... 375
Pertanyaan yang Sering Diajukan ... 375
Memperbarui Sertifikat - GovCloud (AS-Barat) ... 382
Memperbarui Aplikasi dan Klaster Amazon DocumentDB Anda ... 371
Pemecahan Masalah ... 375
Pertanyaan yang Sering Diajukan ... 375
Validasi Kepatuhan ... 393
Ketahanan ... 394
Keamanan Infrastruktur ... 395
Praktik Terbaik Keamanan ... 396
Mengaudit Peristiwa ... 397
Peristiwa yang didukung ... 398
Mengaktifkan Audit ... 403
Mengaktifkan Audit ... 410
Mengakses Acara Audit Anda ... 412
Membuat Cadangan dan Memulihkan ... 414
Mencadangkan dan Memulihkan: Konsep ... 415
Memahami Penggunaan Penyimpanan Backup ... 417
Membuang, Memulihkan, Mengimpor, dan Mengekspor Data ... 418
`mongodump` ... 419
`mongorestore` ... 420
`mongoexport` ... 420
`mongoimport` ... 421
Tutorial ... 421

Pertimbangan Snapshot Klaster 424
Penyimpanan Backup 425
Jendela Backup 425
Periode Retensi Cadangan 427
Salin Enkripsi Snapshot Cluster 427
Membandingkan Snapshot Otomatis dan Manual 428
Membuat Manual Klaster Snapshot 430
Menyalin Snapshot Klaster 433
Menyalin Snapshot Bersama 434
Menyalin Snapshot Di Seluruh Wilayah AWS 434
Batasan 434
Menangani Enkripsi 435
Pertimbangan Grup Parameter 435
Menyalin Snapshot Klaster 436
Membagikan Snapshot Klaster 443
Membagikan Snapshot Terenkripsi 443
Membagikan Snapshot 447
Memulihkan dari Snapshot Klaster 448
Memulihkan ke Titik Waktu 456
Menghapus Snapshot Klaster 462
Mengelola Amazon DocumentDB 464
Gambaran Umum Tugas Operasi 464
Menambahkan Replika ke Klaster Amazon DocumentDB 465
Menggambarkan klaster dan instans 466
Membuat Snapshot Klaster 468
Memulihkan dari snapshot 469
Menghapus sebuah Instans dari sebuah Klaster 470
Menghapus sebuah Klaster 470
Klaster Global 471
Apa itu klaster global? 471
Bagaimana klaster global berguna? 471
Apa keterbatasan terkini klaster global? 472
Panduan Quick Start 473
Mengelola Klaster Global 489
Menghubungkan Klaster Global 497
Memantau Klaster Global 497

Pemulihan Bencana ... 498
Mengelola klaster ... 501
Memahami cluster ... 501
Pengaturan cluster ... 504
Konfigurasi penyimpanan cluster ... 507
Menentukan status klaster ... 510
Siklus hidup cluster ... 511
Menskalakan klaster ... 554
Mengkloning volume untuk cluster ... 557
Memahami toleransi kesalahan cluster ... 570
Mengelola Instans ... 572
Mengelola Kelas Instans ... 572
Menentukan Status Instans ... 582
Siklus hidup instans ... 582
Mengelola Grup Subnet ... 606
Membuat Grup Subnet ... 607
Menjelaskan Grup Subnet ... 612
Mengubah grup subnet ... 615
Menghapus Grup Subnet ... 618
Ketersediaan Tinggi dan Replikasi ... 620
Penskalaan Baca ... 620
Ketersediaan Yang Tinggi ... 620
Menambahkan Replika ... 622
Failover ... 622
Lag Replikasi ... 627
Mengelola indeks ... 628
Pembuatan indeks Amazon DocumentDB ... 628
Mengelola kompresi dokumen ... 634
Pedoman ... 634
Mengaktifkan kompresi dokumen ... 634
Memantau kompresi dokumen ... 635
Mengelola koleksi yang ada ... 636
Mengelola peristiwa peristiwa peristiwa ... 636
Melihat kategori event ... 636
Menampilkan Peristiwa Amazon DocumentDB ... 639
Memilih Wilayah dan Availability Zone ... 642

Ketersediaan Wilayah ... 643
Mengelola kelompok parameter cluster ... 644
Menjelaskan kelompok parameter cluster ... 645
Membuat kelompok parameter cluster ... 652
Memodifikasi kelompok parameter cluster ... 655
Memodifikasi cluster untuk menggunakan kelompok parameter cluster yang disesuaikan ... 659
Menyalin kelompok parameter cluster ... 661
Menyetel ulang kelompok parameter cluster ... 663
Menghapus grup parameter cluster ... 666
Referensi parameter cluster ... 669
Memahami Titik Akhir ... 684
Menemukan Titik Akhir Klaster ... 685
Menemukan Titik Akhir Instans ... 687
Menyambung ke Titik Akhir ... 691
Memahami Amazon DocumentDB ARN ... 692
Membuat konsep ARN ... 692
Menemukan ARN ... 695
Menandai Sumber Daya ... 698
Ikhtisar Tag Sumber Daya ... 698
Kendala Tanda ... 699
Menambahkan atau Memperbarui Tanda ... 700
Mencantumkan Tanda ... 701
Menghapus Tanda ... 703
Memelihara Amazon DocumentDB ... 704
Menentukan Tindakan Pemeliharaan Tertunda ... 705
Menentukan tindakan pemeliharaan yang tertunda ... 707
Menerapkan pembaruan mesin ... 709
Pembaruan yang diprakarsai pengguna ... 712
Mengelola jendela pemeliharaan Anda ... 714
Pembaruan sistem operasi ... 716
Memahami Peran Tertaut Layanan ... 719
Izin Peran Tertaut Layanan ... 720
Membuat Peran Tertaut Layanan ... 722
Memodifikasi Peran Tertaut Layanan ... 722
Menghapus Peran Tertaut Layanan ... 722
Wilayah yang Didukung untuk Peran Tertaut Layanan Amazon DocumentDB ... 724

Menggunakan cluster elastis Amazon DocumentDB 725
Kasus penggunaan cluster elastis 725
Profil pengguna 726
Manajemen konten dan catatan sejarah 726
Keuntungan dari cluster elastis 726
AWS Integrasi layanan 726
Wilayah dan ketersediaan versi 727
Ketersediaan wilayah 727
Ketersediaan versi 728
Batasan 728
Manajemen cluster elastis 728
Operasi kueri dan tulis 728
Pengumpulan dan manajemen indeks 729
Administrasi dan diagnostik 729
Fitur keikutsertaan 729
Cara kerjanya 729
Sharding cluster elastis Amazon DocumentDB 730
Migrasi cluster elastis 734
Penskalaan cluster elastis 734
Keandalan cluster elastis 734
Penyimpanan dan ketersediaan cluster elastis 734
Perbedaan fungsional antara Amazon DocumentDB 4.0 dan cluster elastis 735
Memulai 736
Mengatur 737
Langkah 1: Buat cluster elastis 738
Langkah 2: Buat AWS Cloud9 lingkungan 745
Langkah 3: Instal cangkang mongo 748
Langkah 4: Connect ke cluster elastis baru Anda 749
Langkah 5: Pecahkan koleksi Anda; masukkan dan kueri data 750
Praktik terbaik 752
Memilih Shard Keys 752
Manajemen Koneksi 753
Koleksi Unsharded 753
Menskalakan klaster 753
Memantau klaster 754
Mengelola cluster elastis 754

Memodifikasi konfigurasi cluster elastis ... 755
Memantau cluster elastis ... 758
Menghapus cluster elastis ... 762
Mengelola snapshot cluster elastis ... 764
Menghentikan dan memulai cluster elastis ... 779
Enkripsi data saat diam ... 783
Bagaimana klaster elastis Amazon DocumentDB menggunakan hibah di AWS KMS ... 785
Buat kunci terkelola pelanggan ... 786
Memantau kunci enkripsi Anda untuk klaster elastis Amazon DocumentDB ... 787
Pelajari selengkapnya ... 793
Peran terkait layanan ... 793
Izin peran terkait layanan untuk cluster elastis ... 793
Memantau Amazon DocumentDB ... 797
Memantau status klaster ... 798
Nilai Status Klaster ... 799
Memantau status klaster ... 800
Memantau status instans ... 802
Nilai status status status status status status status status ... 803
Memantau status instans menggunakanAWS Management Console atauAWS CLI ... 805
Nilai status status status status status status status status status kesehatan ... 807
Memantau status kesehatan instans menggunakanAWS Management Console ... 808
Menampilkan Rekomendasi Amazon DocumentDB ... 809
Berlangganan peristiwa ... 812
Berlangganan Peristiwa ... 813
Kelola Langganan ... 816
Kategori dan Pesan ... 820
Memantau Amazon DocumentDB dengan CloudWatch ... 823
Metrik Amazon DocumentDB ... 823
Melihat CloudWatch Data ... 837
Dimensi Amazon DocumentDB ... 843
Pemantauan Opcounters ... 844
Pemantauan Koneksi Basis Data ... 844
Mencatat Panggilan API Amazon DocumentDB dengan CloudTrail ... 844
Informasi Amazon DocumentDB di CloudTrail ... 845
Membuat Profil Operasi ... 846
Operasi yang Didukung ... 847

Keterbatasan: ... 847
Mengaktifkan Profiler ... 847
Menonaktifkan Profiler ... 852
Menonaktifkan Ekspor Log Profiler ... 853
Mengakses Log Profiler Anda ... 855
Kueri umum ... 856
Memantau dengan Wawasan Kinerja ... 856
Konsep Performance Insights ... 857
Mengaktifkan dan menonaktifkan Wawasan Kinerja ... 861
Konfigurasikan kebijakan akses untuk Wawasan Kinerja ... 864
Menganalisis metrik dengan dasbor Performance Insights ... 869
Mengambil metrik dengan API Wawasan Kinerja ... 887
CloudWatch Metrik Amazon untuk Performance Insights ... 902
Performance Insights untuk metrik penghitung ... 904
OpenSearch integrasi ... 907
OpenSearch Layanan Amazon sebagai tujuan ... 907
Langkah 1: Buat domain OpenSearch Layanan Amazon atau koleksi OpenSearch tanpa server ... 908
Langkah 2: Aktifkan aliran perubahan di cluster Amazon DocumentDB ... 908
Langkah 3: Siapkan peran pipeline dengan izin untuk menulis ke bucket Amazon S3 dan domain atau koleksi tujuan ... 908
Langkah 4: Tambahkan izin yang diperlukan pada peran pipeline untuk membuat X-ENI ... 909
Langkah 5: Buat pipa ... 910
Batasan ... 910
Melaksanakan Pengembangan dengan Amazon DocumentDB ... 912
Menghubungkan Secara Terprogram ... 912
Menentukan Nilai `tls` ... 913
Menghubungkan dengan TLS yang Diaktifkan ... 915
Menghubungkan dengan TLS yang Dinonaktifkan ... 929
Menggunakan Aliran Perubahan ... 937
Operasi yang Didukung ... 938
Penagihan ... 938
Keterbatasan: ... 939
Mengaktifkan Aliran Perubahan ... 939
Contoh ... 941
Pencarian Dokumen Lengkap ... 944

Melanjutkan Aliran Perubahan 945
Melanjutkan Aliran Perubahan dengan `startAtOperationTime` 946
Transaksi di aliran perubahan 948
Memodifikasi Durasi Retensi Log Aliran Perubahan 948
MenggunakanAWS Lambdadengan Change Streams 952
Keterbatasan: 953
Menggunakan validasi skema JSON 953
Membuat dan menggunakan validasi skema JSON 953
Kata kunci yang didukung 961
bypassDocumentValidation 963
Batasan 963
Menghubungkan sebagai Set Replika 963
Menggunakan Koneksi Klaster 966
Beberapa Kolam Koneksi 967
Ringkasan 968
Menghubungkan dari Luar Amazon VPC 968
Connect menggunakan Studio 3T 970
Prasyarat 970
Connect dengan Studio 3T 970
Connect menggunakan DataGrip 981
Prasyarat 981
Connect menggunakan DataGrip 982
DataGrip fitur 988
Connect menggunakan Amazon EC2 989
Prasyarat 989
Hubungkan Amazon EC2 secara otomatis 991
Hubungkan Amazon EC2 secara manual 1015
Connect menggunakan driver JDBC 1032
Memulai 1033
Connect dari Tableau Desktop 1034
Connect dari DbVisualizer 1038
Pembuatan skema otomatis JDBC 1040
Dukungan dan Batasan SQL 1049
Pemecahan Masalah 1049
Connect menggunakan driver ODBC 1050
Mulai 1050

Menyiapkan driver ODBC di Windows 1052
Connect dari Microsoft Excel 1057
Connect dari Microsoft Power BI Desktop 1059
Generasi skema otomatis 1065
Support dan batasan SQL 1066
Pemecahan Masalah 1066
Kuota dan Batas 1067
Tipe Instans Yang Didukung 1067
Wilayah yang Didukung 1069
Kuota Wilayah 1070
Batas Agregasi 1073
Batas Klaster 1073
Batas Instans 1075
Kendala Penamaan 1077
Kendala TTL 1079
Batas cluster elastis 1079
Batas pecahan cluster elastis 1080
CPU cluster elastis, memori, koneksi, dan batas kursor per pecahan 1081
Melakukan kueri 1082
Menanyakan dokumen 1082
Mengambil semua dokumen 1083
Mencocokkan nilai bidang 1083
Dokumen yang disematkan 1083
Nilai bidang dalam dokumen yang disematkan 1084
Mencocokkan array 1084
Mencocokkan nilai dalam array 1084
Menggunakan operator 1085
Rencana kueri 1085
Rencana Kueri 1085
Cache rencana kueri 1087
Jelaskan hasil 1087
Tahap pemindaian dan filter 1088
Persimpangan indeks 1089
Serikat indeks 1090
Beberapa persilangan indeks/serikat 1091
Indeks majemuk 1091

Sortir tahap 1092
Panggung grup 1092
Data geospasial 1092
Gambaran Umum 1
Mengindeks dan menyimpan data geospasial 1093
Meminta data geospasial 1095
Batasan 1098
Indeks sebagian 1099
Buat indeks sebagian 1099
Operator yang didukung 1099
Kueri menggunakan indeks paral 1100
Fungsionalitas indeks sebagian 1101
Keterbatasan indeks sebagian 1105
Pencarian teks 1106
Fungsionalitas yang didukung 1106
Menggunakan indeks teks Amazon DocumentDB 1107
Perbedaan dengan MongoDB 1112
Praktik dan pedoman terbaik 1113
Batasan 1113
Pemecahan Masalah 1114
Masalah koneksi 1114
Tidak dapat terhubung ke titik akhir Amazon DocumentDB 1114
Menguji koneksi ke instans Amazon DocumentDB 1120
Menghubungkan ke titik akhir yang tidak valid 1120
Konfigurasi driver yang memengaruhi jumlah koneksi 1121
Pembuatan indeks 1121
Pembuatan indeks gagal 1121
Indeks latar belakang membangun masalah latensi dan gagal 1122
Kinerja dan pemanfaatan sumber daya 1123
Lihat statistik sisipan, perbarui, dan hapus 1123
Menganalisis kinerja cache 1125
Temukan dan Akhiri Kueri yang Berjalan Lama atau Diblokir 1126
Lihat Rencana Kueri dan Mengoptimalkan Kueri 1128
Bagaimana saya bisa melihat rencana kueri dalam cluster elastis? 1130
Daftar Semua Operasi yang Berjalan pada Instans 1132
Ketahui Kapan Kueri Sedang Berjalan 1135

Tentukan Mengapa Sistem Tiba-tiba Berjalan Lambat 1138
Tentukan Penyebab Pemanfaatan CPU yang Tinggi 1139
Temukan kursor terbuka pada sebuah instance 1140
Lihat versi mesin Amazon DocumentDB saat ini 1140
Menganalisis penggunaan indeks dan mengidentifikasi indeks yang tidak digunakan 1141
Identifikasi indeks yang terlewat 1143
Ringkasan pertanyaan yang berguna 1145
Referensi API Manajemen Sumber Daya 1147
Tindakan 1147
Amazon DocumentDB (with MongoDB compatibility) 1150
Cluster Elastis Amazon DocumentDB 1328
Tipe Data 1389
Amazon DocumentDB (with MongoDB compatibility) 1391
Cluster Elastis Amazon DocumentDB 1467
Kesalahan Umum 1482
Parameter Umum 1484
Catatan Rilis 1487
29 Mei 2024 1489
Fitur baru 1489
April 3, 2024 1489
Fitur baru 1490
Perbaikan bug dan perubahan lainnya 1490
Februari 22, 2024 1490
Fitur baru 1490
Januari 30, 2024 1491
Fitur baru 1491
10 Januari 2024 1491
Fitur baru 1491
Perbaikan bug dan perubahan lainnya 1493
20 Desember 2023 1493
Perubahan lainnya 1493
13 Desember 2023 1493
Fitur baru 1493
November 29, 2023 1493
Fitur baru 1493
21 November 2023 1493

Fitur baru 1494
17 November 2023 1494
Fitur baru 1494
Perbaikan bug dan perubahan lainnya 1494
6 November 2023 1494
Fitur baru 1494
Perbaikan bug dan perubahan lainnya 1495
20 Oktober 2023 1495
Perubahan lainnya 1495
25 September 2023 1495
Fitur baru 1495
20 September 2023 1496
Fitur baru 1496
15 September 2023 1496
Fitur baru 1496
11 September 2023 1496
Fitur baru 1496
3 Agustus 2023 1496
Fitur baru 1496
13 Juli 2023 1497
Fitur baru 1497
Perbaikan bug dan perubahan lainnya 1497
Juni 7, 2023 1498
Perbaikan bug dan perubahan lainnya 1498
10 Mei 2023 1498
Perbaikan bug dan perubahan lainnya 1498
4 April 2023 1498
Perbaikan bug dan perubahan lainnya 1498
22 Maret 2023 1499
Fitur baru 1499
1 Maret 2023 1499
Fitur baru 1499
27 Februari 2023 1500
Perbaikan bug dan perubahan lainnya 1500
2 Februari 2023 1500
Perbaikan bug dan perubahan lainnya 1500

30 November 2022 1500
Fitur baru 1500
Agustus 9, 2022 1501
Fitur baru 1501
Perbaikan bug dan perubahan lainnya 1501
25 Juli 2022 1501
Fitur baru 1501
Juni 27, 2022 1502
Fitur baru 1502
29 April 2022 1502
Fitur baru 1502
7 April 2022 1502
Fitur baru 1502
16 Maret 2022 1502
Fitur baru 1502
8 Februari 2022 1503
Fitur baru 1503
24 Januari 2022 1503
Fitur baru 1503
Januari 21, 2022 1503
Fitur baru 1503
25 Oktober 2021 1504
Fitur baru 1504
Perbaikan bug dan perubahan lainnya 1504
24 Juni 2021 1505
Fitur baru 1505
4 Mei 2021 1505
Fitur baru 1505
Perbaikan bug dan perubahan lainnya 1505
15 Januari 2021 1506
Fitur baru 1506
9 November 2020 1506
Fitur baru 1506
Perbaikan bug dan perubahan lainnya 1507
30 Oktober 2020 1508
Fitur baru 1508

Perbaikan bug dan perubahan lainnya 1509
22 September 2020 1509
Fitur baru 1509
Perbaikan bug dan perubahan lainnya 1510
10 Juli 2020 1510
Fitur baru 1510
Perbaikan bug dan perubahan lainnya 1510
30 Juni 2020 1510
Fitur baru 1510
Perbaikan bug dan perubahan lainnya 1510
Riwayat Dokumen 1512
..... mdxxiv

Apa Itu Amazon DocumentDB (dengan Kompatibilitas MongoDB)

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan basis data terkelola penuh yang cepat dan andal. Amazon DocumentDB memudahkan untuk mengatur, mengoperasikan, dan menskalakan basis data kompatibel MongoDB di cloud. Dengan Amazon DocumentDB, Anda dapat menjalankan kode aplikasi yang sama dan menggunakan driver dan alat-alat yang sama yang Anda gunakan dengan MongoDB.

Sebelum menggunakan Amazon DocumentDB, Anda harus meninjau konsep dan fitur yang dijelaskan dalam Cara Kerjanya. Setelah itu, selesaikan langkah-langkah di Panduan Memulai.

Topik

- Gambaran umum Amazon DocumentDB
- Klaster
- Instans
- Wilayah dan Zona Ketersediaan
- Harga Amazon DocumentDB
- Pemantauan
- Antarmuka
- Apa Selanjutnya?
- Amazon DocumentDB: Cara Kerjanya
- Apa itu Basis Data Dokumen?

Gambaran umum Amazon DocumentDB

Berikut ini adalah beberapa fitur tingkat tinggi dari Amazon DocumentDB:

- Amazon DocumentDB mendukung dua jenis cluster: cluster berbasis instance dan cluster elastis. Cluster elastis mendukung beban kerja dengan jutaan baca/tulis per detik dan kapasitas penyimpanan petabyte. Untuk informasi lebih lanjut tentang gugus elastis, lihatMenggunakan cluster elastis Amazon DocumentDB. Konten di bawah ini mengacu pada cluster berbasis instans Amazon DocumentDB.

- Amazon DocumentDB secara otomatis meningkatkan ukuran volume penyimpanan Anda seiring kebutuhan penyimpanan basis data Anda semakin besar. Volume penyimpanan Anda bertambah 10 GB, hingga maksimum 128 TiB. Anda tidak perlu menyediakan kelebihan penyimpanan apa pun untuk klaster Anda untuk menangani pertumbuhan masa depan.
- Dengan Amazon DocumentDB, Anda dapat meningkatkan throughput baca untuk mendukung permintaan aplikasi volume tinggi dengan membuat hingga 15 instans replika. Replika Amazon DocumentDB berbagi penyimpanan yang mendasari sama, menurunkan biaya dan menghindari kebutuhan untuk melakukan penulisan pada simpul replika. Kemampuan ini membebaskan lebih banyak daya pemrosesan untuk melayani permintaan baca dan mengurangi waktu jeda replika—sering kali hingga satu digit milidetik. Anda dapat menambahkan replika dalam hitungan menit terlepas dari ukuran volume penyimpanan. Amazon DocumentDB juga menyediakan reader endpoint, sehingga aplikasi dapat terhubung tanpa harus melacak replika karena mereka ditambahkan dan dihapus.
- Amazon DocumentDB memungkinkan Anda menskalakan ke atas atau ke bawah sumber daya komputasi dan memori untuk setiap instans Anda. Operasi penskalaan komputasi biasanya selesai dalam beberapa menit.
- Amazon DocumentDB berjalan di Amazon Virtual Private Cloud (Amazon VPC), sehingga Anda dapat mengisolasi basis data Anda di jaringan virtual Anda sendiri. Anda juga dapat mengonfigurasi pengaturan firewall untuk mengontrol akses jaringan ke klaster Anda.
- Amazon DocumentDB terus memantau kesehatan klaster Anda. Pada kegagalan instans, Amazon DocumentDB secara otomatis memulai ulang instans dan proses yang berkaitan. Amazon DocumentDB tidak memerlukan pemutaran ulang pemulihan kecelakaan pada log redo basis data, yang sangat mengurangi waktu pemulaian ulang. Amazon DocumentDB juga mengisolasi cache basis data dari proses basis data, mengaktifkan cache untuk mempertahankan pemulaian ulang instans.
- Pada kegagalan instans, Amazon DocumentDB mengotomatisasi failover ke salah satu hingga 15 replika Amazon DocumentDB yang Anda buat di Availability Zone lainnya. Jika tidak ada replika telah disediakan dan terjadi kegagalan, Amazon DocumentDB mencoba untuk membuat instans Amazon DocumentDB baru secara otomatis.
- Kemampuan pencadangan di Amazon DocumentDB point-in-time memungkinkan pemulihan untuk klaster Anda. Fitur ini memungkinkan Anda untuk memulihkan klaster ke detik berapa pun selama periode retensi, hingga 5 menit terakhir. Anda dapat mengonfigurasi periode retensi cadangan otomatis hingga 35 hari. Pencadangan otomatis disimpan di Amazon Simple Storage Service (Amazon S3), yang dirancang untuk daya tahan 99,999999999%. Backup Amazon DocumentDB

bersifat otomatis, inkremental, dan terus menerus, dan mereka tidak berdampak pada kinerja klaster Anda.

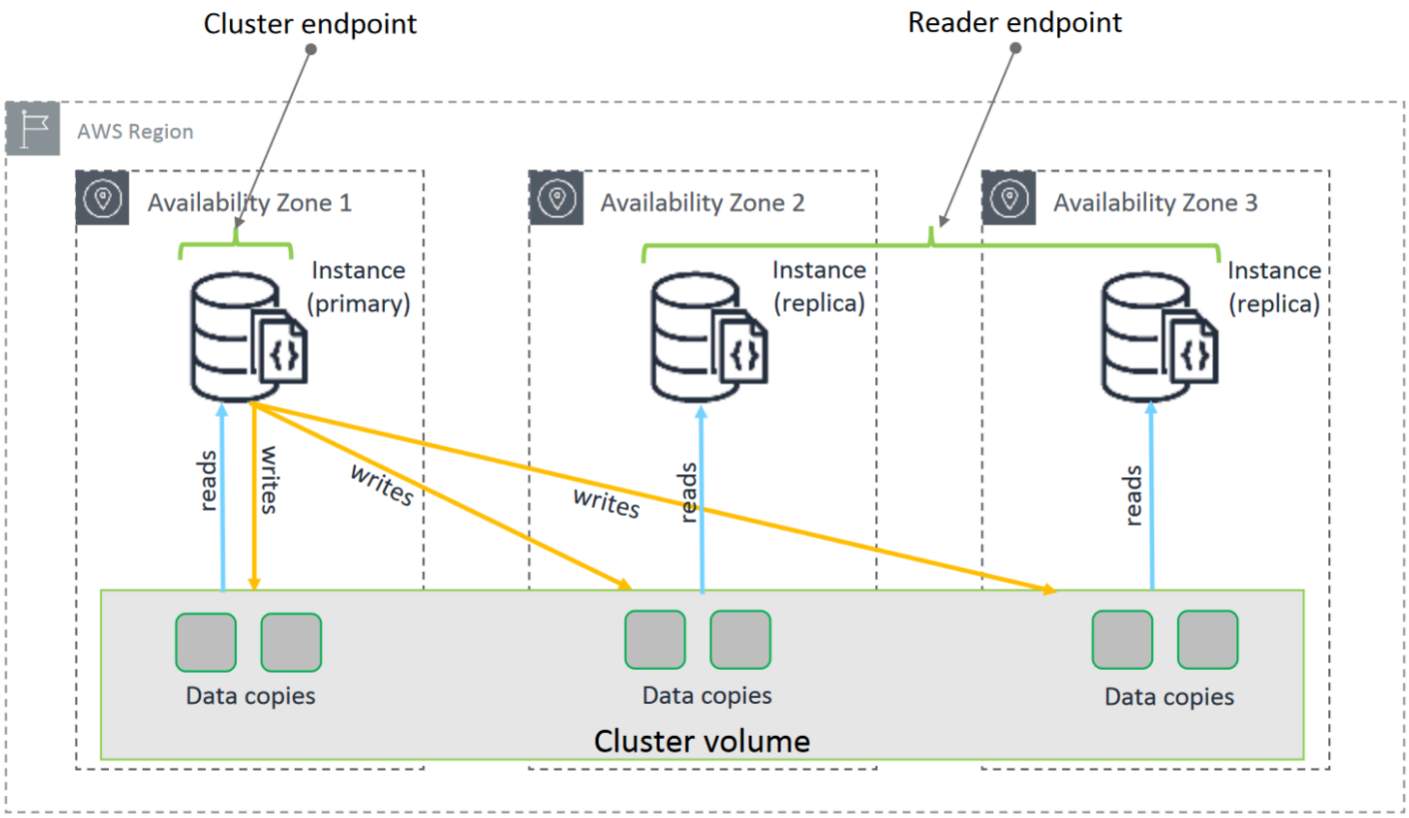
- Dengan Amazon DocumentDB, Anda dapat mengenkripsi database menggunakan kunci yang Anda buat dan kontrol melalui (). AWS Key Management Service AWS KMS Pada klaster basis data yang berjalan dengan enkripsi Amazon DocumentDB, data yang disimpan di rest dalam penyimpanan yang mendasari dienkripsi. Backup otomatis, snapshot, dan replika di klaster yang sama juga dienkripsi.

Jika Anda baru mengenal AWS layanan, gunakan sumber daya berikut untuk mempelajari lebih lanjut:

- AWS menawarkan layanan untuk komputasi, database, penyimpanan, analitik, dan fungsionalitas lainnya. Untuk ikhtisar semua AWS layanan, lihat Komputasi Awan dengan Amazon Web Services.
- AWS menyediakan sejumlah layanan database. Untuk panduan tentang layanan terbaik untuk lingkungan Anda, lihat Basis data pada AWS.

Klaster

Klaster terdiri dari 0 sampai 16 instans dan volume penyimpanan klaster yang mengelola data untuk instans tersebut. Semua penulisan dilakukan melalui instans primer. Semua instans (primer dan replika) mendukung pembacaan. Data klaster disimpan dalam volume klaster dengan salinan dalam tiga Availability Zone yang berbeda.



Cluster berbasis instans Amazon DocumentDB 5.0 mendukung dua konfigurasi penyimpanan untuk kluster database: standar Amazon DocumentDB dan Amazon DocumentDB I/O yang dioptimalkan. Untuk mengetahui informasi selengkapnya, lihat Konfigurasi penyimpanan klaster Amazon DocumentDB.

Instans

Instans Amazon DocumentDB adalah lingkungan basis data terisolasi di cloud. Instans Anda dapat berisi beberapa basis data yang dibuat pengguna. Anda dapat membuat dan memodifikasi instance menggunakan AWS Management Console atau AWS CLI.

Komputasi dan kapasitas memori dari instans ditentukan oleh kelas instans-nya. Anda dapat memilih instans yang paling sesuai dengan kebutuhan Anda. Jika kebutuhan Anda berubah seiring waktu, Anda dapat memilih kelas instans yang berbeda. Untuk spesifikasi kelas instans, lihat Spesifikasi Kelas Instans.

Instans Amazon DocumentDB berjalan hanya di lingkungan Amazon VPC. Amazon VPC memberikan Anda kontrol atas lingkungan jaringan virtual Anda: Anda dapat memilih rentang alamat IP Anda sendiri, membuat subnet, dan mengonfigurasi perutean dan daftar kontrol akses (ACL).

Sebelum Anda dapat membuat instans Amazon DocumentDB, Anda harus membuat klaster yang akan berisi instans.

Tidak semua kelas instans didukung di setiap wilayah. Tabel berikut ini menunjukkan kelas instans mana yang didukung di setiap wilayah.

Kelas instans yang didukung oleh Wilayah

Wilayah	R6G	R5	R4	T4G	T3
AS Timur (Ohio)	Didukung	Didukung	Didukung	Didukung	Didukung
US East (N. Virginia)	Didukung	Didukung	Didukung	Didukung	Didukung
US West (Oregon)	Didukung	Didukung	Didukung	Didukung	Didukung
Amerika Selatan (Sao Paulo)	Didukung	Didukung		Didukung	Didukung
Asia Pasifik (Hong Kong)	Didukung	Didukung		Didukung	Didukung
Asia Pasifik (Hyderabad)		Didukung			Didukung
Asia Pasifik (Mumbai)	Didukung	Didukung		Didukung	Didukung
Asia Pasifik (Seoul)	Didukung	Didukung		Didukung	Didukung
Asia Pacific (Sydney)	Didukung	Didukung		Didukung	Didukung
Asia Pacific (Singapore)	Didukung	Didukung		Didukung	Didukung

Wilayah	R6G	R5	R4	T4G	T3
Asia Pacific (Tokyo)	Didukung	Didukung		Didukung	Didukung
Kanada (Pusat)	Didukung	Didukung		Didukung	Didukung
Eropa (Frankfurt)	Didukung	Didukung		Didukung	Didukung
Eropa (Irlandia)	Didukung	Didukung	Didukung	Didukung	Didukung
Eropa (London)	Didukung	Didukung		Didukung	Didukung
Eropa (Milan)	Didukung	Didukung		Didukung	Didukung
Eropa (Paris)	Didukung	Didukung		Didukung	Didukung
Timur Tengah (UEA)	Didukung	Didukung		Didukung	Didukung
Wilayah Tiongkok (Beijing)	Didukung	Didukung		Didukung	Didukung
Tiongkok (Ningxia)	Didukung	Didukung		Didukung	Didukung
AWS GovCloud (AS-Barat)	Didukung	Didukung		Didukung	Didukung
AWS GovCloud (AS-Timur)	Didukung	Didukung		Didukung	Didukung

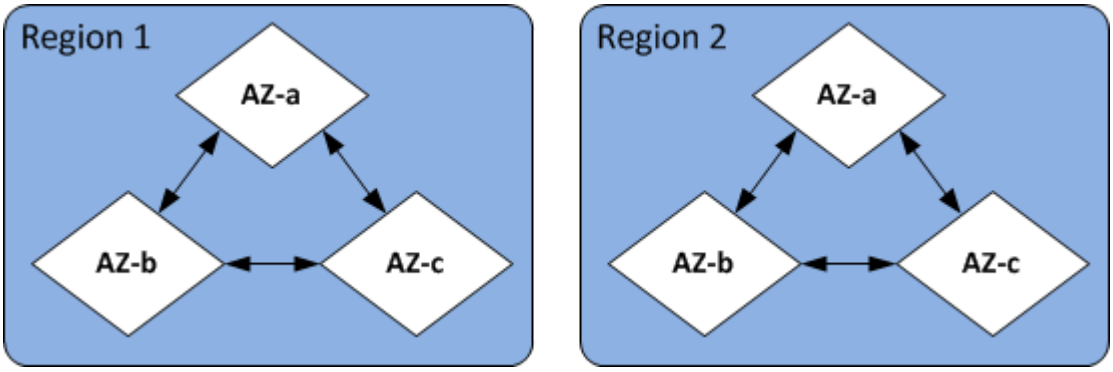
Wilayah dan Zona Ketersediaan

Wilayah dan Availability Zone menentukan lokasi fisik klaster dan instans Anda.

Wilayah

AWS Sumber daya komputasi awan ditempatkan di fasilitas pusat data yang sangat tersedia di berbagai wilayah di dunia (misalnya, Amerika Utara, Eropa, atau Asia). Setiap lokasi pusat data disebut Wilayah.

Setiap AWS Wilayah dirancang untuk sepenuhnya terisolasi dari AWS Wilayah lain. Di dalam setiap wilayah terdapat beberapa Zona Ketersediaan. Dengan meluncurkan simpul Anda di Zona Ketersediaan yang berbeda, Anda dapat mencapai toleransi kesalahan sebesar mungkin. Diagram berikut menunjukkan tampilan tingkat tinggi tentang cara kerja AWS Wilayah dan Zona Ketersediaan.



Zona Ketersediaan

Setiap AWS Wilayah berisi beberapa lokasi berbeda yang disebut Availability Zone. Setiap Availability Zone direkayasa untuk diisolasi dari kegagalan di Availability Zone lainnya, dan untuk menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama. Dengan meluncurkan instans untuk klaster tertentu di beberapa Availability Zone, Anda dapat melindungi aplikasi Anda dari peristiwa kegagalan Availability Zone yang tidak diinginkan.

Arsitektur Amazon DocumentDB memisahkan penyimpanan dan komputasi. Untuk lapisan penyimpanan, Amazon DocumentDB mereplikasi enam salinan data Anda di tiga Availability Zone. AWS Sebagai contoh, jika Anda meluncurkan klaster Amazon DocumentDB di Wilayah yang hanya mendukung dua Availability Zone, penyimpanan data Anda akan direplikasi enam cara di tiga Availability Zone tetapi instans komputasi Anda hanya akan tersedia di dua Availability Zone.

Tabel berikut mencantumkan jumlah Availability Zone yang dapat Anda gunakan dalam instance komputasi yang diberikan Wilayah AWS untuk menyediakan untuk klaster Anda.

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
AS Timur (Ohio)	us-east-2	3
AS Timur (Virginia Utara)	us-east-1	6
AS Barat (Oregon)	us-west-2	4
Amerika Selatan (Sao Paulo)	sa-east-1	3
Asia Pasifik (Hong Kong)	ap-east-1	3
Asia Pasifik (Hyderabad)	ap-south-2	3
Asia Pasifik (Mumbai)	ap-south-1	3
Asia Pasifik (Seoul)	ap-northeast-2	4
Asia Pasifik (Singapura)	ap-southeast-1	3
Asia Pasifik (Sydney)	ap-southeast-2	3
Asia Pasifik (Tokyo)	ap-northeast-1	3
Kanada (Pusat)	ca-central-1	3
Wilayah Tiongkok (Beijing)	cn-north-1	3
Tiongkok (Ningxia)	cn-northwest-1	3
Eropa (Frankfurt)	eu-central-1	3
Eropa (Irlandia)	eu-west-1	3
Eropa (London)	eu-west-2	3

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
Eropa (Milan)	`eu-south-1`	3
Eropa (Paris)	`eu-west-3`	3
Timur Tengah (UEA)	`me-central-1`	3
AWS GovCloud (AS-Barat)	`us-gov-west-1`	3
AWS GovCloud (AS-Timur)	`us-gov-east-1`	3

Harga Amazon DocumentDB

Cluster Amazon DocumentDB ditagih berdasarkan komponen berikut:

- Jam instans (per jam)—Berdasarkan kelas instans dari instans (sebagai contoh, `db.r5.xlarge`). Harga dicantumkan per jam, tetapi tagihan dihitung turun menjadi detik dan menunjukkan waktu dalam bentuk desimal. Penggunaan Amazon DocumentDB ditagihkan dalam kenaikan satu detik, dengan minimum 10 menit. Untuk informasi selengkapnya, lihat Mengelola Kelas Instans.
- Permintaan I/O (per 1 juta permintaan per bulan) — Total jumlah permintaan I/O penyimpanan yang telah Anda buat dalam siklus penagihan.
- Penyimpanan backup (per GiB per bulan) — Penyimpanan backup adalah penyimpanan yang berkaitan dengan backup basis data otomatis dan setiap snapshot basis data aktif yang telah Anda ambil. Meningkatkan periode retensi cadangan atau mengambil snapshot basis data tambahan akan meningkatkan penyimpanan cadangan yang digunakan oleh basis data Anda. Penyimpanan Backup diukur dalam GB-bulan dan per detik tidak berlaku. Untuk informasi selengkapnya, lihat Membuat Cadangan dan Memulihkan di Amazon DocumentDB.
- Transfer data (per GB) — Transfer data masuk dan keluar dari instans Anda dari atau ke internet atau AWS Wilayah lain.

Untuk informasi lebih lanjut, lihat harga Amazon DocumentDB.

Uji coba gratis

Anda dapat mencoba Amazon DocumentDB secara gratis menggunakan uji coba gratis 1 bulan. Untuk informasi selengkapnya, lihat Uji coba gratis di harga Amazon DocumentDB atau lihat FAQ uji coba gratis Amazon DocumentDB.

Pemantauan

Terdapat beberapa cara untuk melacak performa dan kesehatan instans. Anda dapat menggunakan CloudWatch layanan Amazon gratis untuk memantau kinerja dan kesehatan suatu instans. Anda dapat menemukan grafik performa di konsol Amazon DocumentDB. Anda dapat berlangganan peristiwa Amazon DocumentDB untuk mendapatkan notifikasi tentang perubahan yang terjadi pada instans, snapshot, grup parameter, atau grup keamanan.

Untuk informasi selengkapnya, lihat berikut ini:

- Memantau Amazon DocumentDB dengan CloudWatch
- Mencatat Panggilan API Amazon DocumentDB dengan AWS CloudTrail

Antarmuka

Ada beberapa cara bagi Anda untuk berinteraksi dengan Amazon DocumentDB, termasuk AWS Management Console dan. AWS CLI

AWS Management Console

AWS Management Console Ini adalah antarmuka pengguna berbasis web yang sederhana. Anda dapat mengelola klaster dan instans Anda dari konsol tanpa perlu pemrograman. Untuk mengakses konsol Amazon DocumentDB, masuk ke AWS Management Console dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.

AWS CLI

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk mengelola klaster dan instans Amazon DocumentDB Anda. Dengan konfigurasi minimal, Anda dapat mulai menggunakan semua fungsionalitas yang disediakan oleh konsol Amazon DocumentDB dari program terminal favorit Anda.

- Untuk menginstal AWS CLI, lihat Menginstal Antarmuka Baris AWS Perintah.

- Untuk mulai menggunakan Amazon DocumentDB AWS CLI untuk Amazon, AWS lihat Referensi Antarmuka Baris Perintah untuk Amazon DocumentDB.

Shell mongo

Untuk menghubungkan ke klaster Anda untuk membuat, membaca, memperbarui, menghapus dokumen dalam basis data Anda, Anda dapat menggunakan shell `mongo` dengan Amazon DocumentDB. Untuk mengunduh dan menginstal shell `mongo` 4.0, lihat Langkah 4: Instal shell mongo.

Driver MongoDB

Untuk mengembangkan dan menulis aplikasi terhadap klaster Amazon DocumentDB, Anda juga dapat menggunakan driver MongoDB dengan Amazon DocumentDB.

Apa Selanjutnya?

Bagian sebelumnya memperkenalkan Anda pada komponen infrastruktur dasar yang ditawarkan Amazon DocumentDB. Apa yang harus Anda lakukan selanjutnya? Tergantung pada keadaan Anda, lihat salah satu topik berikut untuk memulai:

- Memulai Amazon DocumentDB dengan membuat cluster dan instance menggunakan. AWS CloudFormation Amazon DocumentDB Mulai Cepat Menggunakan AWS CloudFormation
- Memulai dengan Amazon DocumentDB dengan membuat klaster dan instans menggunakan petunjuk di Panduan Memulai kami.
- Mulailah dengan Amazon DocumentDB dengan membuat cluster elastis menggunakan instruksi di. Memulai dengan cluster elastis Amazon DocumentDB
- Migrasikan implementasi MongoDB Anda ke Amazon DocumentDB menggunakan panduan di Migrasi ke Amazon DocumentDB

Amazon DocumentDB: Cara Kerjanya

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan basis data yang kompatibel dengan MongoDB yang terkelola penuh. Dengan Amazon DocumentDB, Anda dapat menjalankan kode aplikasi yang sama dan menggunakan driver dan alat yang sama yang Anda gunakan dengan MongoDB. Amazon DocumentDB kompatibel dengan MongoDB 3.6, 4.0, dan 5.0.

Topik

- Titik Akhir Amazon DocumentDB
- Dukungan TLS
- Penyimpanan Amazon DocumentDB
- Replikasi Amazon DocumentDB
- Keandalan Amazon DocumentDB
- Opsi Preferensi Baca
- Penghapusan TTL
- Sumber Daya Dapat Ditagih

Saat Anda menggunakan Amazon DocumentDB, Anda mulai dengan membuat klaster. Sebuah klaster terdiri dari nol atau lebih instans basis data dan volume klaster yang mengelola data untuk instans tersebut. Volume klaster Amazon DocumentDB adalah volume penyimpanan basis data virtual yang mencakup beberapa Availability Zone. Setiap Availability Zone memiliki salinan data klaster.

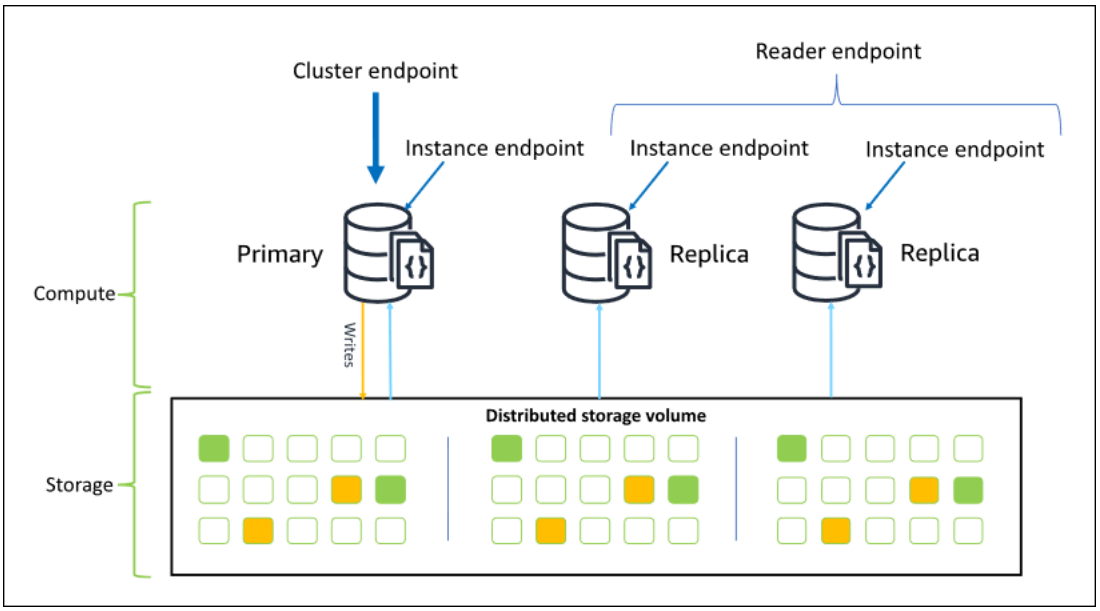
Klaster Amazon DocumentDB terdiri dari dua komponen:

- Volume klaster—Menggunakan layanan penyimpanan cloud-native untuk mereplikasi data dengan enam cara di tiga Availability Zone, menyediakan penyimpanan yang sangat berdaya tahan dan tersedia. Cluster Amazon DocumentDB memiliki tepat satu volume cluster, yang dapat menyimpan hingga 128 TiB data.
- Instans—Menyediakan kekuatan pemrosesan untuk basis data, menulis data ke, dan membaca data dari, volume penyimpanan klaster. Klaster Amazon DocumentDB dapat memiliki 0–16 instans.

Instans melayani salah satu dari dua peran:

- Instans primer—Mendukung operasi baca dan tulis, dan melakukan semua modifikasi data ke volume klaster. Setiap klaster Amazon DocumentDB memiliki satu instans primer.
- Instans replika—Mendukung hanya operasi baca. Setiap klaster Amazon DocumentDB dapat memiliki hingga 15 instans replika selain instans primer. Memiliki beberapa replika memungkinkan Anda untuk mendistribusikan beban kerja baca. Selain itu, dengan menempatkan replika di Availability Zone terpisah, Anda juga meningkatkan ketersediaan klaster.

Diagram berikut mengilustrasikan hubungan antara volume klaster, instans primer, dan replika di klaster Amazon DocumentDB:



Instans klaster tidak harus dari kelas instans yang sama, dan mereka dapat disediakan dan diakhiri sesuai yang diinginkan. Arsitektur ini memungkinkan Anda menskalakan kapasitas komputasi klaster Anda secara independen dari penyimpanannya.

Saat aplikasi Anda menulis data ke instans primer, yang primer mengeksekusi penulisan yang tahan lama ke volume klaster. Itu kemudian mereplikasi status penulisan itu (bukan data) ke setiap replika aktif. Replika Amazon DocumentDB tidak berpartisipasi dalam pemrosesan tulis, dan dengan demikian replika Amazon DocumentDB menguntungkan untuk penskalaan baca. Pembacaan dari replika Amazon DocumentDB pada akhirnya konsisten dengan jeda replika minimal—biasanya kurang dari 100 milidetik setelah instans primer menulis data. Bacaan dari replika dijamin akan dibaca sesuai urutan penulisannya ke yang primer. Jeda replika bervariasi tergantung pada kecepatan perubahan data, dan periode aktivitas tulis yang tinggi dapat meningkatkan jeda replika. Untuk informasi lebih, lihat metrik `ReplicationLag` di Metrik Amazon DocumentDB.

Titik Akhir Amazon DocumentDB

Amazon DocumentDB menyediakan beberapa opsi koneksi untuk melayani berbagai kasus penggunaan. Untuk terhubung ke instans di klaster Amazon DocumentDB, Anda menentukan titik akhir instans. Sebuah titik akhir adalah alamat host dan nomor port, dipisahkan oleh titik dua.

Kami merekomendasikan Anda terhubung ke klaster menggunakan titik akhir klaster dan dalam mode set replika (lihat Menghubungkan ke Amazon DocumentDB sebagai Set Replika) kecuali

jika Anda memiliki kasus penggunaan khusus untuk menghubungkan ke reader endpoint atau titik akhir instans. Untuk merutekan permintaan ke replika Anda, pilih pengaturan preferensi baca driver yang memaksimalkan penskalaan baca sekaligus memenuhi persyaratan konsistensi baca aplikasi Anda. Preferensi baca `secondaryPreferred` mengaktifkan pembacaan replika dan membebaskan instans primer untuk melakukan lebih banyak pekerjaan.

Titik akhir berikut tersedia dari klaster Amazon DocumentDB.

Titik Akhir klaster

Titik akhir klaster terhubung ke instans primer klaster Anda saat ini. Titik akhir klaster dapat digunakan untuk operasi baca dan tulis. Sebuah klaster Amazon DocumentDB memiliki tepat satu titik akhir klaster.

Titik akhir klaster menyediakan dukungan failover untuk koneksi baca dan tulis ke klaster. Jika instans primer klaster Anda saat ini gagal, dan klaster Anda memiliki setidaknya satu replika baca aktif, titik akhir klaster secara otomatis mengalihkan permintaan koneksi ke instans primer baru. Saat menghubungkan ke klaster Amazon DocumentDB Anda, kami merekomendasikan Anda menghubungkan ke klaster Anda menggunakan titik akhir klaster dan dalam mode set replika (lihat Menghubungkan ke Amazon DocumentDB sebagai Set Replika).

Berikut ini adalah contoh titik akhir klaster Amazon DocumentDB:

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

Berikut ini adalah contoh koneksi string menggunakan titik akhir klaster ini:

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

Untuk informasi tentang menemukan titik akhir klaster, lihat Menemukan Titik Akhir Klaster.

Reader Endpoint

Beban reader endpoint menyeimbangkan koneksi baca-saja di semua replika yang tersedia di klaster Anda. Titik akhir pembaca cluster akan tampil sebagai titik akhir cluster jika Anda terhubung melalui `replicaSet` mode, artinya dalam string koneksi, parameter set replika adalah. `&replicaSet=rs0` Dalam hal ini, Anda akan dapat melakukan operasi penulisan pada primer. Namun, jika Anda

terhubung ke klaster yang menentukan`directConnection=true`, maka mencoba melakukan operasi tulis melalui koneksi ke titik akhir pembaca menghasilkan kesalahan. Sebuah klaster Amazon DocumentDB memiliki tepat satu reader endpoint.

Jika klaster hanya berisi satu instans (primer), reader endpoint terhubung ke instans primer. Saat Anda menambahkan instans replika ke klaster Amazon DocumentDB, reader endpoint membuka koneksi baca saja ke replika baru setelah itu aktif.

Berikut ini adalah contoh reader endpoint untuk klaster Amazon DocumentDB:

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

Berikut ini adalah contoh string koneksi menggunakan reader endpoint:

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

Beban reader endpoint menyeimbangkan koneksi baca-saja, bukan permintaan baca. Jika beberapa koneksi reader endpoint lebih banyak digunakan daripada yang lain, permintaan baca Anda mungkin tidak seimbang di antara instans di dalam klaster. Direkomendasikan untuk mendistribusikan permintaan dengan menghubungkan ke titik akhir klaster sebagai set replika dan menggunakan opsi preferensi baca secondaryPreferred.

Untuk informasi tentang menemukan titik akhir klaster, lihat Menemukan Titik Akhir Klaster.

Titik Akhir Instans

Titik akhir instans terhubung ke instans tertentu dalam klaster Anda. Titik akhir instans untuk instans primer saat ini dapat digunakan untuk operasi baca dan tulis. Namun, mencoba melakukan operasi tulis ke titik akhir instans untuk replika baca akan menghasilkan kesalahan. Klaster Amazon DocumentDB memiliki satu titik akhir instans per instans aktif.

Titik akhir instans memberikan kontrol langsung atas koneksi ke instans tertentu untuk skenario di mana titik akhir klaster atau reader endpoint mungkin tidak sesuai. Contoh kasus penggunaan adalah penyediaan beban kerja analitik baca-saja secara berkala. Anda dapat menyediakan instance larger-than-normal replika, terhubung langsung ke instans baru yang lebih besar dengan titik akhir instansnya, menjalankan kueri analitik, dan kemudian menghentikan instance. Menggunakan titik akhir instans menjaga lalu lintas analitik agar tidak memengaruhi instans klaster lainnya.

Berikut ini adalah contoh titik akhir instans untuk instans tunggal di klaster Amazon DocumentDB:

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

Berikut ini adalah contoh string koneksi menggunakan titik akhir instans ini:

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

> **Note**
>
> Peran instans sebagai primer atau replika dapat berubah karena peristiwa failover. Aplikasi Anda tidak boleh berasumsi bahwa titik akhir instans tertentu adalah instans primer. Kami tidak menyarankan menghubungkan ke titik akhir instans untuk aplikasi produksi. Sebagai gantinya, kami merekomendasikan Anda menyambungkan ke klaster menggunakan titik akhir klaster dan dalam mode set replika (lihat Menghubungkan ke Amazon DocumentDB sebagai Set Replika). Untuk kendali lebih lanjut dari prioritas failover instans, lihat Memahami toleransi kesalahan klaster Amazon DocumentDB.

Untuk informasi tentang menemukan titik akhir klaster, lihat Menemukan Titik Akhir Instans.

Mode Set Replika

Anda dapat terhubung ke titik akhir klaster Amazon DocumentDB Anda dalam mode set replika dengan menentukan nama set replika `rs0`. Menghubungkan dalam mode set replika menyediakan kemampuan untuk menentukan opsi Perhatian Baca, Perhatian Tulis, dan Preferensi Baca. Untuk informasi selengkapnya, lihat Konsistensi Baca.

Berikut ini adalah contoh koneksi string yang menghubungkan dalam mode set replika:

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

Saat Anda terhubung dalam mode set replika, klaster Amazon DocumentDB Anda muncul ke driver dan klien Anda sebagai set replika. Instans yang ditambahkan dan dihapus dari klaster Amazon DocumentDB Anda direfleksikan secara otomatis dalam konfigurasi set replika.

Setiap klaster Amazon DocumentDB terdiri dari set replika tunggal dengan nama default `rs0`. Nama set replika tidak dapat diubah.

Menghubungkan ke titik akhir klaster dalam mode set replika adalah metode yang disarankan untuk penggunaan umum.

> **Note**
>
> Semua instans dalam klaster Amazon DocumentDB mendengarkan pada port TCP yang sama untuk koneksi.

Dukungan TLS

Untuk detail selengkapnya tentang menghubungkan ke Amazon DocumentDB menggunakan Transport Layer Security (TLS), lihat Mengenkripsi Data dalam Transit.

Penyimpanan Amazon DocumentDB

Data Amazon DocumentDB disimpan dalam volume klaster, yang merupakan volume virtual tunggal yang menggunakan solid state drive (SSD). Volume klaster terdiri dari enam salinan data Anda, yang direplikasi secara otomatis di beberapa Availability Zone dalam satu Wilayah AWS. Replikasi ini membantu memastikan bahwa data Anda sangat berdaya tahan, dengan kemungkinan kehilangan data yang lebih kecil. Ini juga membantu memastikan bahwa klaster Anda lebih tersedia selama failover karena salinan data Anda sudah ada di Availability Zone lainnya. Salinan ini dapat terus melayani permintaan data ke instans di klaster Amazon DocumentDB Anda.

Bagaimana Penyimpanan Data Ditagih

Amazon DocumentDB secara otomatis meningkatkan ukuran volume klaster saat jumlah data meningkat. Volume cluster Amazon DocumentDB dapat tumbuh hingga ukuran maksimum 128 TiB; Namun, Anda hanya dikenakan biaya untuk ruang yang Anda gunakan dalam volume cluster Amazon DocumentDB. Dimulai dengan Amazon DocumentDB 4.0, ketika data dihapus, seperti dengan menjatuhkan koleksi atau indeks, keseluruhan ruang yang dialokasikan berkurang dengan jumlah yang sebanding. Dengan demikian, Anda dapat mengurangi biaya penyimpanan dengan menghapus koleksi, indeks, dan database yang tidak lagi Anda perlukan. Dengan Amazon DocumentDB 3.6, ketika data dihapus, seperti dengan menjatuhkan koleksi atau indeks, keseluruhan ruang yang dialokasikan tetap sama. Ruang kosong digunakan kembali secara otomatis ketika volume data meningkat di masa depan.

> **Note**
>
> Dengan Amazon DocumentDB 3.6, biaya penyimpanan didasarkan pada penyimpanan "tanda air tinggi" (jumlah maksimum yang dialokasikan untuk cluster Amazon DocumentDB kapan saja). Anda dapat mengelola biaya dengan menghindari praktik ETL yang membuat volume besar informasi sementara, atau yang memuat volume besar data baru sebelum menghapus data lama yang tidak dibutuhkan. Jika menghapus data dari klaster Amazon DocumentDB menghasilkan sejumlah besar ruang yang dialokasikan tetapi tidak digunakan, mengatur ulang tanda air yang tinggi memerlukan pembuangan data logis dan memulihkan ke klaster baru, menggunakan alat seperti `mongodump` atau `mongorestore`. Membuat dan memulihkan snapshot tidak mengurangi penyimpanan yang dialokasikan karena tata letak fisik penyimpanan yang mendasarinya tetap sama di snapshot yang dipulihkan.

> **Note**
>
> Menggunakan utilitas seperti `mongodump` dan `mongorestore` dikenakan biaya I/O berdasarkan ukuran data yang sedang dibaca dan ditulis ke volume penyimpanan.

Untuk informasi tentang penyimpanan data Amazon DocumentDB dan harga I/O, lihat Harga Amazon DocumentDB (dengan kompatibilitas MongoDB) dan FAQ Harga.

Replikasi Amazon DocumentDB

Dalam klaster Amazon DocumentDB, setiap instans replika menampilkan titik akhir independen. Titik akhir replika ini menyediakan akses baca-saja ke data dalam volume klaster. Mereka memungkinkan Anda untuk menskalakan beban kerja baca untuk data Anda melalui beberapa instans yang direplikasi. Mereka juga membantu meningkatkan performa pembacaan data dan meningkatkan ketersediaan data di klaster Amazon DocumentDB Anda. Replika Amazon DocumentDB juga merupakan target failover dan dengan cepat dipromosikan jika instans primer untuk klaster Amazon DocumentDB Anda gagal.

Keandalan Amazon DocumentDB

Amazon DocumentDB dirancang agar andal, tahan lama, dan toleran terhadap kesalahan. (Untuk meningkatkan ketersediaan, Anda harus mengonfigurasi klaster Amazon DocumentDB Anda

sehingga memiliki beberapa instans replika di Availability Zone yang berbeda.) Amazon DocumentDB menyertakan beberapa fitur otomatis yang menjadikannya solusi basis data yang andal.

Perbaikan-Otomatis Penyimpanan

Amazon DocumentDB menyimpan banyak salinan data Anda di tiga Availability Zone, sangat mengurangi kemungkinan kehilangan data karena kegagalan penyimpanan. Amazon DocumentDB secara otomatis mendeteksi kegagalan di dalam volume klaster. Saat segmen dari sebuah volume klaster gagal, Amazon DocumentDB segera memperbaiki segmen tersebut. Ini menggunakan data dari volume lain yang membentuk volume klaster untuk membantu memastikan bahwa data di segmen yang diperbaiki adalah yang terkini. Akibatnya, Amazon DocumentDB menghindari kehilangan data dan mengurangi kebutuhan untuk melakukan pemulihan untuk memulihkan dari point-in-time kegagalan instans.

Penghangatan Cache Yang Bisa Bertahan

Amazon DocumentDB mengelola cache halamannya dalam proses terpisah dari basis data sehingga cache halaman dapat bertahan secara independen dari basis data. Jika terjadi kegagalan basis data yang tidak terduga, cache halaman tetap berada di memori. Ini memastikan bahwa kumpulan buffer dihangatkan dengan status terkini saat basis data dimulai ulang.

Pemulihan Crash

Amazon DocumentDB dirancang untuk pulih dari crash hampir seketika, dan untuk terus menyajikan data aplikasi Anda. Amazon DocumentDB melakukan pemulihan crash secara asinkron pada utas paralel sehingga basis data Anda terbuka dan tersedia segera setelah kerusakan.

Pengelolaan Sumber Daya

Amazon DocumentDB melindungi sumber daya yang diperlukan untuk menjalankan proses penting dalam layanan, seperti pemeriksaan kondisi. Untuk melakukan ini, dan ketika instans mengalami tekanan memori tinggi, Amazon DocumentDB akan membatasi permintaan. Akibatnya, beberapa operasi mungkin diantrekan untuk menunggu tekanan memori mereda. Jika tekanan memori berlanjut, operasi yang diantrekan mungkin habis. Anda dapat memantau apakah operasi pelambatan layanan karena memori rendah atau tidak dengan CloudWatch metrik berikut:`LowMemThrottleQueueDepth`,,,`LowMemThrottleMaxQueueDepth`. `LowMemNumOperationsThrottled` `LowMemNumOperationsTimedOut` Untuk informasi selengkapnya, lihat Memantau Amazon CloudWatch DocumentDB dengan. Jika Anda melihat

tekanan memori berkelanjutan pada instans Anda sebagai akibat dari LowMem CloudWatch metrik, kami menyarankan Anda meningkatkan skala instance Anda untuk menyediakan memori tambahan untuk beban kerja Anda.

Opsi Preferensi Baca

Amazon DocumentDB menggunakan layanan penyimpanan bersama cloud-native yang mereplikasi data enam kali di tiga Availability Zone untuk memberikan tingkat ketahanan yang tinggi. Amazon DocumentDB tidak bergantung pada replikasi data ke beberapa instans untuk mencapai ketahanan. Data klaster Anda tahan lama baik berisi satu instans atau 15 instans.

Daya Tahan Tulis

Amazon DocumentDB menggunakan sistem penyimpanan self-healing yang unik, terdistribusi, toleran terhadap kesalahan. Sistem ini mereplikasi enam salinan (V=6) data Anda di tiga AWS Availability Zone untuk memberikan ketersediaan dan daya tahan yang tinggi. Ketika menulis data, Amazon DocumentDB memastikan bahwa semua penulisan dicatat secara tahan lama di sebagian besar node sebelum menyatakan penulisan ke klien. Jika Anda menjalankan set replika MongoDB tiga simpul, menggunakan perhatian tulis `{w:3, j:true}` akan menghasilkan konfigurasi terbaik saat dibandingkan dengan Amazon DocumentDB.

Penulisan ke klaster Amazon DocumentDB harus diproses oleh instans tulis klaster. Mencoba menulis ke pembaca menghasilkan kesalahan. Penulisan yang dinyatakan dari instans primer Amazon DocumentDB tahan lama, dan tidak dapat dibatalkan. Amazon DocumentDB sangat berdaya tahan secara default dan tidak mendukung opsi tulis yang tidak tahan lama. Anda tidak dapat mengubah tingkat daya tahan (yaitu, perhatian tulis). Amazon DocumentDB mengabaikan w=anything dan secara efektif w: 3 dan j: true. Anda tidak dapat menguranginya.

Karena penyimpanan dan komputasi dipisahkan di dalam arsitektur Amazon DocumentDB, klaster dengan satu instans sangat berdaya tahan. Daya tahan ditangani di lapisan penyimpanan. Hasilnya, klaster Amazon DocumentDB dengan satu instans dan satu dengan tiga instans mencapai tingkat ketahanan yang sama. Anda dapat mengonfigurasi klaster Anda ke kasus penggunaan spesifik Anda sambil tetap memberikan daya tahan tinggi untuk data Anda.

Penulisan ke klaster Amazon DocumentDB bersifat atomik dalam satu dokumen.

Amazon DocumentDB tidak mendukung opsi `wtimeout` dan tidak akan mengembalikan kesalahan jika nilai ditentukan. Penulisan ke instans Amazon DocumentDB primer dijamin tidak akan diblokir tanpa batas waktu.

Isolasi Baca

Pembacaan dari instans Amazon DocumentDB hanya mengembalikan data yang tahan lama sebelum kueri dimulai. Pembacaan tidak pernah mengembalikan data yang dimodifikasi setelah kueri memulai eksekusi, juga bacaan kotor tidak mungkin dilakukan dalam keadaan apa pun.

Konsistensi Baca

Data yang dibaca dari klaster Amazon DocumentDB tahan lama dan tidak akan dibatalkan. Anda dapat mengubah konsistensi baca untuk pembacaan Amazon DocumentDB dengan menentukan preferensi baca untuk permintaan atau koneksi. Amazon DocumentDB tidak mendukung opsi baca yang tidak tahan lama.

Pembacaan dari instans utama cluster Amazon DocumentDB sangat konsisten dalam kondisi operasi normal dan memiliki konsistensi. read-after-write Jika peristiwa failover terjadi antara penulisan dan pembacaan berikutnya, sistem dapat secara singkat mengembalikan pembacaan yang tidak sangat konsisten. Semua pembacaan dari replika baca pada akhirnya konsisten dan mengembalikan data dalam urutan yang sama, dan seringkali dengan jeda replika kurang dari 100 mili detik.

Preferensi Baca Amazon DocumentDB

Amazon DocumentDB mendukung pengaturan opsi preferensi baca hanya saat membaca data dari titik akhir klaster dalam mode set replika. Mengatur opsi preferensi baca memengaruhi cara klien atau driver MongoDB Anda merutekan permintaan baca ke instans di klaster Amazon DocumentDB Anda. Anda dapat mengatur opsi preferensi baca untuk kueri tertentu, atau sebagai opsi umum di driver MongoDB Anda. (Konsultasikan dokumentasi klien atau driver Anda untuk petunjuk tentang bagaimana cara mengatur opsi preferensi baca.)

Jika klien atau driver Anda tidak terhubung ke titik akhir klaster Amazon DocumentDB dalam mode set replika, hasil penentuan preferensi baca tidak ditentukan.

Amazon DocumentDB tidak mendukung pengaturan set tag sebagai preferensi baca.

Opsi Preferensi Baca Yang Didukung

- **`primary`**—Menentukan preferensi baca `primary` membantu memastikan bahwa semua pembacaan dirutekan ke instans primer klaster. Jika instans primer tidak tersedia, operasi baca gagal. Preferensi `primary` baca menghasilkan read-after-write konsistensi dan sesuai untuk kasus penggunaan yang memprioritaskan read-after-write konsistensi daripada ketersediaan tinggi dan penskalaan baca.

Contoh berikut menentukan preferensi baca `primary`:

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

- **`primaryPreferred`**—Menentukan rute preferensi baca `primaryPreferred` membaca ke instans primer dalam operasi normal. Jika ada failover primer, klien merutekan permintaan ke replika. Preferensi `primaryPreferred` baca menghasilkan read-after-write konsistensi selama operasi normal, dan akhirnya pembacaan yang konsisten selama peristiwa failover. Preferensi `primaryPreferred` baca sesuai untuk kasus penggunaan yang memprioritaskan read-after-write konsistensi daripada penskalaan baca, tetapi masih membutuhkan ketersediaan tinggi.

  Contoh berikut menentukan preferensi baca `primaryPreferred`:

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

- **`secondary`**—Menentukan preferensi baca `secondary` memastikan bahwa pembacaan hanya dirutekan ke replika, tidak pernah ke instans primer. Jika tidak ada instans replika dalam sebuah klaster, permintaan baca akan gagal. Preferensi `secondary` baca pada akhirnya menghasilkan pembacaan yang konsisten dan sesuai untuk kasus penggunaan yang memprioritaskan throughput penulisan instance utama daripada ketersediaan dan konsistensi yang tinggi. read-after-write

  Contoh berikut menentukan preferensi baca `secondary`:

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

- **`secondaryPreferred`**—Menentukan preferensi baca `secondaryPreferred` memastikan bahwa pembacaan dirutekan ke replika baca saat satu atau beberapa replika aktif. Jika tidak ada instans replika aktif dalam sebuah klaster, permintaan baca akan dirutekan ke instans primer. Preferensi baca `secondaryPreferred` menghasilkan bacaan akhir konsisten saat pembacaan dilayani oleh replika baca. Ini menghasilkan read-after-write konsistensi ketika pembacaan dilayani oleh instance utama (kecuali peristiwa failover). Preferensi `secondaryPreferred` baca sesuai untuk kasus penggunaan yang memprioritaskan penskalaan baca dan ketersediaan tinggi daripada konsistensi. read-after-write

Contoh berikut menentukan preferensi baca `secondaryPreferred`:

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

- **`nearest`**—Menentukan rute preferensi baca `nearest` membaca hanya berdasarkan latensi terukur antara klien dan semua instans di klaster Amazon DocumentDB. Preferensi baca `nearest` menghasilkan bacaan akhir konsisten saat pembacaan dilayani oleh replika baca. Ini menghasilkan read-after-write konsistensi ketika pembacaan dilayani oleh instance utama (kecuali peristiwa failover). Preferensi `nearest` baca sesuai untuk kasus penggunaan yang memprioritaskan pencapaian latensi baca serendah mungkin dan ketersediaan tinggi daripada read-after-write konsistensi dan penskalaan baca.

  Contoh berikut menentukan preferensi baca `nearest`:

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

Ketersediaan Yang Tinggi

Amazon DocumentDB mendukung konfigurasi klaster yang sangat tersedia dengan menggunakan replika sebagai target failover untuk instans primer. Jika instans primer gagal, replika Amazon DocumentDB dipromosikan sebagai primer baru, dengan gangguan singkat selama permintaan baca dan tulis yang dibuat ke instans primer gagal dengan pengecualian.

Jika klaster Amazon DocumentDB Anda tidak menyertakan replika apa pun, instans primer akan dibuat ulang selama kegagalan. Namun, mempromosikan replika Amazon DocumentDB jauh lebih cepat daripada membuat ulang instans primer. Jadi kami merekomendasikan Anda membuat satu atau beberapa replika Amazon DocumentDB sebagai target failover.

Replika yang dimaksudkan untuk digunakan sebagai target failover harus dari kelas instans yang sama dengan instans primer. Mereka harus disediakan di Availability Zone yang berbeda dari yang primer. Anda dapat mengontrol replika mana yang lebih disukai sebagai target failover. Untuk praktik terbaik dalam mengonfigurasi Amazon DocumentDB untuk ketersediaan tinggi, lihat Memahami toleransi kesalahan klaster Amazon DocumentDB.

Penskalaan Baca

Replika Amazon DocumentDB ideal untuk penskalaan baca. Mereka sepenuhnya didedikasikan untuk membaca operasi pada volume klaster Anda, yaitu, replika tidak memproses penulisan. Replikasi data terjadi dalam volume klaster dan bukan di antara instans. Jadi setiap sumber daya replika didedikasikan untuk memproses kueri Anda, bukan mereplikasi dan menulis data.

Jika aplikasi Anda membutuhkan lebih banyak kapasitas baca, Anda dapat menambahkan replika ke klaster Anda dengan cepat (biasanya dalam waktu kurang dari sepuluh menit). Jika persyaratan kapasitas baca Anda berkurang, Anda dapat menghapus replika yang tidak dibutuhkan. Dengan replika Amazon DocumentDB, Anda hanya membayar untuk kapasitas baca yang Anda butuhkan.

Amazon DocumentDB mendukung penskalaan baca sisi klien melalui penggunaan opsi Preferensi Baca. Untuk informasi selengkapnya, lihat Preferensi Baca Amazon DocumentDB.

Penghapusan TTL

Penghapusan dari area indeks TTL yang dicapai melalui proses latar belakang adalah upaya terbaik dan tidak dijamin dalam jangka waktu tertentu. Faktor-faktor seperti ukuran instans, pemanfaatan sumber daya instans, ukuran dokumen, dan throughput keseluruhan dapat memengaruhi waktu penghapusan TTL.

Saat monitor TTL menghapus dokumen Anda, setiap penghapusan dikenakan biaya IO, yang akan menambah tagihan Anda. Jika throughput dan tingkat penghapusan TTL meningkat, Anda akan mengalami peningkatan tagihan Anda karena peningkatan penggunaan IO.

Saat Anda membuat indeks TTL pada koleksi yang ada, Anda harus menghapus semua dokumen yang kedaluwarsa sebelum membuat indeks. Implementasi TTL saat ini dioptimalkan untuk menghapus sebagian kecil dokumen dalam koleksi, yang biasa terjadi jika TTL diaktifkan pada koleksi dari awal, dan dapat menghasilkan IOPS yang lebih tinggi daripada yang diperlukan jika sejumlah besar dokumen perlu dihapus dalam sekali jalan.

Jika Anda tidak ingin membuat indeks TTL untuk menghapus dokumen, Anda dapat mengelompokkan dokumen ke dalam koleksi berdasarkan waktu, dan cukup membuang koleksi tersebut saat dokumen tidak lagi diperlukan. Misalnya: Anda dapat membuat satu koleksi per minggu dan membuangnya tanpa mengeluarkan biaya IO. Ini dapat secara signifikan menghemat biaya daripada menggunakan indeks TTL.

Sumber Daya Dapat Ditagih

Mengidentifikasi Sumber Daya Amazon DocumentDB yang Dapat Ditagih

Sebagai layanan basis data terkelola penuh, Amazon DocumentDB membebankan biaya untuk instans, penyimpanan, I/O, pencadangan, dan transfer data. Untuk informasi selengkapnya, lihat Harga Amazon DocumentDB (dengan kompatibilitas MongoDB).

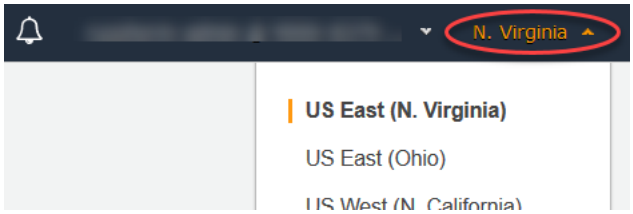
Untuk menemukan sumber daya yang dapat ditagih di akun Anda dan berpotensi menghapus sumber daya, Anda dapat menggunakan atau. AWS Management Console AWS CLI

Menggunakan AWS Management Console

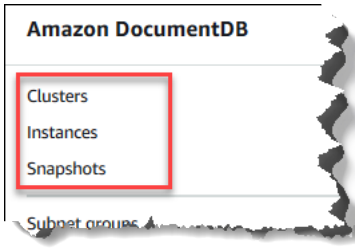
Dengan menggunakan AWS Management Console, Anda dapat menemukan klaster Amazon DocumentDB, instance, dan snapshot yang telah Anda sediakan untuk diberikan. Wilayah AWS

Untuk menemukan klaster, instans, dan snapshot

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Untuk menemukan sumber daya yang dapat ditagih di Wilayah selain Wilayah default Anda, di sudut kanan atas layar, pilih yang ingin Anda cari. Wilayah AWS



3. Di panel navigasi, pilih tipe sumber daya yang dapat ditagih yang Anda minati: Klaster, Instans, atau Snapshot.



4. Semua klaster, instans, atau snapshot yang Anda sediakan untuk Wilayah terdaftar di panel kanan. Anda akan dikenakan biaya untuk klaster, instans, dan snapshot.

Menggunakan AWS CLI

Dengan menggunakan AWS CLI, Anda dapat menemukan klaster Amazon DocumentDB, instance, dan snapshot yang telah Anda sediakan untuk diberikan. Wilayah AWS

Untuk menemukan klaster dan instans

Kode berikut mencantumkan semua klaster dan instans Anda untuk Wilayah yang ditentukan. Jika Anda ingin mencari klaster dan instans di Wilayah default, Anda dapat menghilangkan parameter `--region`.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Output dari operasi ini terlihat seperti berikut ini.

```
"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",
```

Untuk menemukan snapshot

Kode berikut mencantumkan semua snapshot Anda untuk Wilayah yang ditentukan. Jika Anda ingin mencari snapshot di Wilayah default Anda, Anda dapat menghilangkan parameter `--region`.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Output dari operasi ini terlihat seperti berikut ini.

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

Anda hanya perlu menghapus snapshot `manual`. Snapshot `Automated` dihapus saat Anda menghapus klaster.

Menghapus Sumber Daya Dapat Ditagih yang Tidak Diinginkan

Untuk menghapus klaster, Anda harus menghapus semua instans di klaster terlebih dahulu.

- Untuk menghapus instans, lihat Menghapus instans Amazon DocumentDB .

> **Important**
>
> Meskipun Anda menghapus instans dalam sebuah klaster, Anda masih ditagih untuk penggunaan penyimpanan dan pencadangan yang terkait dengan klaster tersebut. Untuk menghentikan semua tagihan, Anda juga harus menghapus klaster dan snapshot manual.

- Untuk menghapus klaster, lihat Menghapus cluster Amazon DocumentDB.

- Untuk menghapus snapshot manual, lihat Menghapus Snapshot Klaster.

Apa itu Basis Data Dokumen?

Beberapa developer tidak memikirkan model data mereka dalam hal baris dan kolom normal. Biasanya, di tingkat aplikasi, data direpresentasikan sebagai dokumen JSON karena lebih intuitif bagi developer untuk memikirkan model data mereka sebagai dokumen.

Popularitas basis data dokumen telah berkembang karena basis data dokumen ini membuat Anda dapat mempertahanan data dalam basis data dengan menggunakan format model dokumen yang sama yang Anda gunakan dalam kode aplikasi Anda. Basis data dokumen menyediakan API yang kuat dan intuitif untuk pengembangan yang fleksibel dan tangkas.

Topik

- Kasus Penggunaan Basis Data Dokumen
- Memahami Dokumen
- Bekerja dengan Dokumen

Kasus Penggunaan Basis Data Dokumen

Drive kasus penggunaan Anda apakah Anda memerlukan basis data dokumen atau beberapa jenis basis data lain untuk mengelola data Anda. Basis data dokumen berguna untuk beban kerja yang memerlukan skema fleksibel untuk pengembangan yang cepat dan berulang. Berikut ini adalah beberapa contoh kasus penggunaan di mana basis data dokumen dapat memberikan keuntungan yang signifikan:

Topik

- Profil Pengguna
- Big Data Waktu Nyata
- Manajemen Konten

Profil Pengguna

Karena basis data dokumen memiliki skema yang fleksibel, mereka dapat menyimpan dokumen yang memiliki atribut dan nilai data yang berbeda. Basis data dokumen adalah solusi praktis untuk

profil online di mana pengguna yang berbeda menyediakan berbagai jenis informasi. Dengan menggunakan basis data dokumen, Anda dapat menyimpan profil setiap pengguna efisien dengan menyimpan hanya atribut yang spesifik untuk setiap pengguna.

Misalkan pengguna memilih untuk menambahkan atau menghapus informasi dari profil mereka. Dalam kasus ini, dokumen mereka dapat dengan mudah diganti dengan versi terbaru yang berisi atribut dan data baru apa pun yang baru ditambahkan atau menghilangkan atribut dan data baru apa pun yang dihilangkan. Basis data dokumen dengan mudah mengelola tingkat individualitas dan fluiditas ini.

Big Data Waktu Nyata

Secara historis, kemampuan untuk mengekstrak informasi dari data operasional terhambat oleh fakta bahwa basis data operasional dan basis data analitik dipertahankan dalam lingkungan yang berbeda—operasional dan bisnis/pelaporan secara berurutan. Mampu mengekstrak informasi operasional secara langsung sangat penting dalam lingkungan bisnis yang sangat kompetitif. Dengan menggunakan basis data dokumen, bisnis dapat menyimpan dan mengelola data operasional dari sumber mana pun dan secara bersamaan mengumpankan data ke mesin BI pilihan untuk analisis. Tidak ada persyaratan untuk memiliki dua lingkungan.

Manajemen Konten

Untuk mengelola konten secara efektif, Anda harus dapat mengumpulkan dan mengagregasikan konten dari berbagai sumber, dan kemudian mengirimkannya ke pelanggan. Akibat skema fleksibel mereka, basis data dokumen sempurna untuk mengumpulkan dan menyimpan jenis data apa pun. Anda dapat menggunakannya untuk membuat dan menggabungkan jenis konten baru, termasuk konten buatan pengguna, seperti gambar, komentar, dan video.

Memahami Dokumen

Basis data dokumen digunakan untuk menyimpan data semistruktur sebagai dokumen—alih-alih menormalkan data di beberapa tabel, masing-masing dengan struktur yang unik dan tetap, seperti dalam basis data relasional. Dokumen yang disimpan dalam basis data dokumen menggunakan pasangan nilai kunci nest untuk menyediakan struktur atau skema dokumen. Namun demikian, jenis dokumen yang berbeda dapat disimpan dalam basis data dokumen yang sama, sehingga memenuhi persyaratan untuk memproses data serupa yang dalam format yang berbeda. Sebagai contoh, karena setiap dokumen dijelaskan sendiri, dokumen yang dienkodekan JSON untuk toko online yang dijelaskan dalam topik Contoh Dokumen dalam Basis Data Dokumen dapat disimpan dalam basis data dokumen yang sama.

Topik

- SQL vs Terminologi Nonrelasional
- Dokumen Sederhana
- Dokumen Tersemat
- Contoh Dokumen dalam Basis Data Dokumen
- Memahami Normalisasi dalam Basis Data Dokumen

SQL vs Terminologi Nonrelasional

Tabel berikut ini membandingkan terminologi yang digunakan oleh basis data dokumen (MongoDB) dengan terminologi yang digunakan oleh basis data SQL.

SQL	MongoDB
Tabel	Koleksi
Baris	Dokumen
Kolom	Bidang
Kunci Primer	ObjectId
Indeks	Indeks
Lihat	Lihat
Tabel atau objek nest	Dokumen tersemat
Susunan	Susunan

Dokumen Sederhana

Semua dokumen dalam basis data dokumen adalah dijelaskan sendiri. Dokumentasi ini menggunakan dokumen berformat seperti JSON, meskipun Anda dapat menggunakan cara pengenkodean lain.

Dokumen sederhana memiliki satu atau lebih bidang yang semuanya pada tingkat yang sama di dalam dokumen. Dalam contoh berikut ini, bidang `SSN`, `LName`, `FName`, `DOB`, `Street`, `City`, `State-Province`, `PostalCode`, dan`Country` semuanya adalah saudara kandung di dalam dokumen.

```
{
   "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"
}
```

Ketika informasi diatur dalam dokumen sederhana, setiap bidang terkelola secara individual. Untuk mengambil alamat seseorang, Anda harus mengambil `Street`, `City`,`State-Province`, `PostalCode`, dan `Country` sebagai item data individu.

Dokumen Tersemat

Dokumen kompleks mengatur datanya dengan membuat dokumen tersemat di dalam dokumen. Dokumen tersemat membantu mengelola data dalam pengelompokan dan sebagai item data individual, mana yang lebih efisien dalam kasus tertentu. Dengan menggunakan contoh sebelumnya, Anda dapat menyematkan dokumen `Address` dalam dokumen utama. Melakukan hal ini akan menghasilkan struktur dokumen berikut ini:

```
{
   "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"
   }
```

```
}
```

Anda sekarang dapat mengakses data dalam dokumen sebagai bidang individu (`"SSN":`), sebagai dokumen tersemat (`"Address":`), atau sebagai anggota dokumen tersemat (`"Address": {"Street":}`).

Contoh Dokumen dalam Basis Data Dokumen

Seperti yang dinyatakan sebelumnya, karena setiap dokumen dalam basis data dokumen adalah dijelaskan sendiri, struktur dokumen di dalam basis data dokumen dapat berbeda satu sama lain. Dua dokumen berikut ini, satu untuk buku dan satu lagi untuk majalah, berbeda secara struktural. Namun keduanya dapat dalam basis data dokumen yang sama.

Berikut ini adalah dokumen buku sampel:

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

Berikut ini adalah dokumen majalah sampel dengan dua artikel:

```
{
    "_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"
}
```

Bandingkan struktur kedua dokumen tersebut. Dengan basis data relasional, Anda perlu memisah tabel “majalah” dan “buku”, atau satu tabel dengan bidang yang tidak terpakai, seperti “Publikasi”, “Terbitan”, “Artikel”, dan “MI”, sebagai nilai `null`. Karena basis data dokumen adalah semistruktur, dengan setiap dokumen mendefinisikan strukturnya sendiri, kedua dokumen tersebut dapat hidup berdampingan dalam basis data dokumen yang sama tanpa bidang `null`. Basis data dokumen baik dalam menangani data yang jarang.

Mengembangkan terhadap basis data dokumen memungkinkan pengembangan yang cepat dan berulang. Hal ini karena Anda dapat mengubah struktur data dokumen secara dinamis, tanpa harus mengubah skema untuk seluruh koleksi. Basis data dokumen sangat cocok untuk pengembangan tangkas dan secara dinamis mengubah lingkungan.

Memahami Normalisasi dalam Basis Data Dokumen

Basis data dokumen tidak dinormalisasi; data yang ditemukan dalam satu dokumen dapat diulang dalam dokumen lainnya. Lebih lanjut, beberapa perbedaan data dapat ada di antara dokumen. Sebagai contoh, pertimbangkan skenario di mana Anda melakukan pembelian di toko online dan semua detail pembelian Anda disimpan dalam dokumen tunggal. Dokumen mungkin terlihat seperti dokumen JSON berikut ini:

```
{
    "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"
}
```

Semua informasi ini disimpan sebagai dokumen dalam koleksi transaksi. Kemudian, Anda menyadari bahwa Anda lupa membeli satu item. Sehingga Anda kembali masuk ke toko yang sama dan melakukan pembelian lainnya, yang juga disimpan sebagai dokumen lainnya dalam koleksi transaksi.

```
{
    "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"
}
```

Perhatikan redundansi antara kedua dokumen tersebut—nama dan ID pembelanja Anda (dan, jika Anda menggunakan kartu kredit yang sama, informasi kartu kredit Anda). Tapi hal ini tidak apa-apa karena penyimpanan murah, dan setiap dokumen benar-benar mencatat transaksi tunggal yang dapat diambil dengan cepat dengan kueri nilai kunci sederhana tanpa harus bergabung.

Terdapat juga perbedaan yang jelas antara kedua dokumen—informasi kartu kredit Anda. Ini hanya perbedaan yang jelas karena kemungkinan Anda akan menggunakan kartu kredit yang berbeda untuk setiap pembelian. Setiap dokumen akurat untuk transaksi yang didokumentasikan.

Bekerja dengan Dokumen

Sebagai basis data dokumen, Amazon DocumentDB membuatnya mudah untuk menyimpan, membuat kueri, dan membuat indeks data JSON. Di Amazon DocumentDB, koleksi analog dengan tabel dalam basis data relasional, kecuali tidak ada skema tunggal ditegakkan pada semua dokumen. Koleksi memungkinkan Anda mengelompokkan dokumen yang serupa bersama-sama sembari menjaga mereka semua dalam basis data yang sama, tanpa mengharuskan bahwa mereka identik dalam struktur.

Dengan menggunakan contoh dokumen dari bagian sebelumnya, kemungkinan bahwa Anda akan memiliki koleksi untuk `reading_material` dan `office_supplies`. Merupakan tanggung jawab perangkat lunak Anda untuk menegakkan koleksi yang memiliki dokumen tersebut.

Contoh berikut ini menggunakan MongoDB API untuk menunjukkan bagaimana untuk menambahkan, kueri, pembaruan, dan menghapus dokumen.

Topik

- Menambahkan Dokumen
- Membuat Kueri Dokumen
- Memperbarui Dokumen

- Menghapus Dokumen

Menambahkan Dokumen

Di Amazon DocumentDB, basis data dibuat saat pertama kali Anda menambahkan dokumen ke koleksi. Dalam contoh ini, Anda membuat koleksi yang diberi nama `example` di basis data `test`, yang merupakan basis data default ketika Anda terhubung ke klaster. Karena koleksi secara implisit dibuat ketika dokumen pertama dimasukkan, tidak ada pemeriksaan kesalahan nama koleksi. Oleh karena itu, salah ketik dalam nama koleksi, seperti `eexample` alih-alih `example`, akan membuat dan menambahkan dokumen ke koleksi `eexample` alih-alih ke koleksi yang dimaksudkan. Pemeriksaan kesalahan harus ditangani oleh aplikasi Anda.

Contoh berikut ini menggunakan API MongoDB untuk menambahkan dokumen.

Topik

- Menambahkan Dokumen Tunggal
- Menambahkan Beberapa Dokumen

Menambahkan Dokumen Tunggal

Untuk menambahkan dokumen tunggal ke koleksi, gunakan operasi `insertOne( {} )` dengan dokumen yang ingin Anda tambahkan ke koleksi.

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

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
```

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

Menambahkan Beberapa Dokumen

Untuk menambahkan beberapa dokumen ke koleksi, gunakan operasi `insertMany( [{},..., {}] )` dengan daftar dokumen yang ingin Anda tambahkan ke koleksi. Meskipun dokumen dalam daftar khusus ini memiliki skema yang berbeda, mereka semua dapat ditambahkan ke koleksi yang sama.

```
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
            }
        }
    ]
)
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

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

Membuat Kueri Dokumen

Terkadang, Anda mungkin perlu mencari inventaris toko online Anda sehingga pelanggan dapat melihat dan membeli apa yang Anda jual. Melakukan kueri sebuah koleksi relatif mudah, apakah Anda ingin semua dokumen dalam koleksi atau hanya dokumen yang memenuhi kriteria tertentu.

Untuk melakukan kueri dokumen, gunakan operasi `find()`. Perintah `find()` memiliki parameter dokumen tunggal yang mendefinisikan kriteria yang akan digunakan dalam memilih dokumen yang akan dikembalikan. Output dari `find()` adalah dokumen yang diformat sebagai satu baris teks tanpa jeda baris. Untuk memformat dokumen output agar lebih mudah dibaca, gunakan `find().pretty()`. Semua contoh dalam topik ini menggunakan `.pretty()` untuk memformat keluaran.

Gunakan empat dokumen yang Anda masukkan ke dalam koleksi `example` dalam dua latihan sebelumnya — `insertOne()` dan `insertMany()`.

Topik

- Mengambil Semua Dokumen dalam Koleksi
- Mengambil Dokumen yang Cocok Dengan Nilai Bidang
- Mengambil Dokumen yang Cocok Dengan Dokumen Tersemat
- Mengambil Dokumen yang Cocok Dengan Nilai Bidang dalam Dokumen Tersemat
- Mengambil Dokumen yang Cocok Dengan Susunan
- Mengambil Dokumen yang Cocok Dengan Nilai dalam Susunan
- Mengambil Dokumen Menggunakan Operator

Mengambil Semua Dokumen dalam Koleksi

Untuk mengambil semua dokumen dalam koleksi Anda, gunakan operasi `find()` dengan dokumen kueri yang kosong.

Kueri berikut mengembalikan semua dokumen di dalam koleksi `example`.

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

Mengambil Dokumen yang Cocok Dengan Nilai Bidang

Untuk mengambil semua dokumen yang cocok dengan bidang dan nilai, gunakan operasi `find()` dengan dokumen kueri yang mengidentifikasi bidang dan nilai untuk mencocokkan.

Menggunakan dokumen sebelumnya, kueri ini mengembalikan semua dokumen di mana bidang "Item" sama dengan "Pena".

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

Mengambil Dokumen yang Cocok Dengan Dokumen Tersemat

Untuk menemukan semua dokumen yang cocok dengan dokumen tersemat, gunakan operasi `find()` dengan dokumen kueri yang menentukan nama dokumen tersemat dan semua bidang dan nilai untuk dokumen tersemat.

Saat mencocokkan dokumen yang ditanamkan, dokumen yang ditanamkan tersebut harus memiliki nama yang sama seperti dalam kueri. Selain itu, bidang dan nilai dalam dokumen yang ditanamkan harus cocok dengan kueri.

Kueri berikut hanya mengembalikan dokumen "Cat Poster". Ini karena "Pena" memiliki nilai yang berbeda untuk "`OnHand`" dan "`MinOnHand`", dan "Cat Semprot" memiliki satu bidang (`OrderQnty`) lebih banyak daripada dokumen kueri.

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

Mengambil Dokumen yang Cocok Dengan Nilai Bidang dalam Dokumen Tersemat

Untuk menemukan semua dokumen yang cocok dengan dokumen tersemat, gunakan operasi `find()` dengan dokumen kueri yang menentukan nama dokumen tersemat dan semua bidang dan nilai untuk dokumen tersemat.

Mengingat dokumen sebelumnya, kueri berikut menggunakan "notasi titik" untuk menentukan dokumen yang ditanamkan dan bidang yang diminati. Setiap dokumen yang cocok dengan

ini dikembalikan, terlepas dari apa bidang lain mungkin hadir dalam dokumen tertanam. Kueri mengembalikan "Cat Poster" dan "Cat Semprot" karena keduanya cocok dengan bidang dan nilai yang ditentukan.

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

Mengambil Dokumen yang Cocok Dengan Susunan

Untuk menemukan semua dokumen yang cocok dengan susunan, gunakan operasi `find()` dengan nama susunan yang menjadi perhatian Anda dan semua nilai dalam susunan tersebut. Kueri mengembalikan semua dokumen yang memiliki array dengan nama itu di mana nilai arraynya identik dengan dan dalam urutan yang sama seperti dalam kueri.

Kueri berikut hanya mengembalikan "Pena" karena "Cat Poster" memiliki warna tambahan (Putih), dan "Cat Semprot" memiliki warna dalam urutan yang berbeda.

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

Mengambil Dokumen yang Cocok Dengan Nilai dalam Susunan

Untuk menemukan semua dokumen yang memiliki nilai susunan tertentu, gunakan operasi `find()` dengan nama susunan dan nilai yang menjadi perhatian Anda.

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

Operasi sebelumnya mengembalikan ketiga dokumen karena masing-masing memiliki array bernama `Colors` dan nilai "`Red`" di suatu tempat dalam array. Jika Anda menentukan nilai "`White`", kueri hanya akan mengembalikan "Cat Poster."

Mengambil Dokumen Menggunakan Operator

Kueri berikut ini mengembalikan semua dokumen di mana nilai "`Inventory.OnHand`" adalah kurang dari 50.

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

Untuk daftar operator kueri yang didukung, lihat Kueri dan Proyeksi Operator.

Memperbarui Dokumen

Biasanya, dokumen Anda tidak statis dan diperbarui sebagai bagian dari alur kerja aplikasi Anda. Contoh berikut ini menunjukkan beberapa cara yang dapat lakukan Anda untuk memperbarui dokumen.

Untuk memperbarui dokumen yang ada, gunakan operasi `update()`. Operasi `update()` memiliki dua parameter dokumen. Dokumen pertama mengidentifikasi dokumen atau dokumen-dokumen mana yang akan diperbarui. Dokumen kedua menentukan pembaruan yang akan dilakukan.

Ketika Anda memperbarui bidang yang ada — apakah bidang itu adalah bidang sederhana, susunan, atau dokumen tersemat -— Anda harus menentukan nama bidang dan nilainya. Pada akhir operasi, seolah-olah bidang dalam dokumen lama telah digantikan oleh bidang dan nilai baru.

Topik

- Memperbarui Nilai dari Bidang yang Ada
- Menambahkan Bidang Baru
- Mengganti Dokumen Tersemat
- Memasukkan Bidang Baru ke Dokumen Tersemat
- Menghapus Bidang dari Dokumen
- Menghapus Bidang dari Beberapa Dokumen

Memperbarui Nilai dari Bidang yang Ada

Gunakan empat dokumen berikut ini yang Anda tambahkan sebelumnya untuk operasi pembaruan berikut ini.

```
{
    "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
    }
}
```

Untuk memperbarui bidang sederhana

Untuk memperbarui bidang sederhana, gunakan `update()` dengan `$set` untuk menentukan nama bidang dan nilai baru. Contoh berikut ini mengubah `Item` dari “Pen” ke “Gel Pen”.

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

Hasil dari operasi ini terlihat seperti berikut ini.

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

```
}
```

Untuk memperbarui susunan

Contoh berikut ini menggantikan susunan warna yang ada dengan susunan baru yang mencakup `Orange` dan menghapus `White` dari daftar warna. Daftar warna baru adalah dalam urutan yang ditentukan dalam operasi `update()`.

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

Hasil dari operasi ini terlihat seperti berikut ini.

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

Menambahkan Bidang Baru

Untuk mengubah dokumen dengan menambahkan satu atau lebih bidang baru, gunakan operasi `update()` dengan dokumen kueri yang mengidentifikasi dokumen yang akan dimasukkan dan bidang dan nilai baru yang akan dimasukkan menggunakan operator `$set`.

Contoh berikut ini menambahkan bidang `UnitPrice` dengan nilai `3.99` ke dokumen Spray Paints. Perhatikan bahwa nilai `3.99` adalah numerik dan bukan string.

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

Hasil dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
```

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

Mengganti Dokumen Tersemat

Untuk mengubah dokumen dengan mengganti dokumen tersemat, gunakan operasi `update()` dengan dokumen yang mengidentifikasi dokumen tersemat dan bidang dan nilai barunya menggunakan operator `$set`.

Mengingat dokumen berikut.

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

Untuk mengganti dokumen tersemat

Contoh berikut ini menggantikan dokumen Tanggal saat ini dengan yang dokumen baru yang hanya memiliki bidang `Month` dan `Day`; `Year` telah dihilangkan.

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

Hasil dari operasi ini terlihat seperti berikut ini.

```
{
    "DocName": "Document 1",
    "Date": {
```

```
        "Month": 4,
        "Day": 18
    }
}
```

Memasukkan Bidang Baru ke Dokumen Tersemat

Untuk menambahkan bidang ke dokumen tersemat

Untuk mengubah dokumen dengan menambahkan satu atau lebih bidang baru ke dokumen tersemat, gunakan operasi `update()` dengan dokumen yang mengidentifikasi dokumen tersemat dan “notasi dot” untuk menentukan dokumen tersemat dan bidang dan nilai baru yang akan dimasukkan menggunakan operator `$set`.

Mengingat dokumen berikut ini, kode berikut ini menggunakan “notasi dot” untuk memasukkan bidang `Year` dan `DoW` ke dokumen `Date`yang disematkan, dan `Words` ke dalam dokumen induk.

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

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

Hasil dari operasi ini terlihat seperti berikut ini.

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

```
    },
    "Words": 2482
}
```

Menghapus Bidang dari Dokumen

Untuk mengubah dokumen dengan menghapus bidang dari dokumen, gunakan operasi `update()` dengan dokumen kueri yang mengidentifikasi dokumen yang akan dihapus bidangnya, dan operator `$unset` untuk menentukan bidang yang akan dihapus.

Contoh berikut ini akan menghapus bidang `Words`dari dokumen sebelumnya.

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

Hasil dari operasi ini terlihat seperti berikut ini.

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

Menghapus Bidang dari Beberapa Dokumen

Untuk mengubah dokumen dengan menghapus bidang dari beberapa dokumen, gunakan operasi `update()` dengan operator `$unset` dan `multi` diatur ke `true`.

Contoh berikut ini akan menghapus bidang `Inventory` dari semua dokumen dalam koleksi contoh. Jika dokumen tidak memiliki bidang `Inventory`, tidak ada tindakan yang dilakukan pada dokumen tersebut. Jika `multi: true` dihilangkan, tindakan dilakukan hanya pada dokumen pertama yang memenuhi kriteria.

```
db.example.update(
```

```
    {},
    { $unset: { Inventory:1 } },
    { multi: true }
)
```

Menghapus Dokumen

Untuk menghapus dokumen dari basis data Anda, gunakan operasi `remove()`, yang menentukan dokumen yang akan dihapus. Kode berikut ini menghapus “Gel Pen” dari koleksi `example` Anda.

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

Untuk menghapus semua dokumen dari basis data Anda, gunakan operasi `remove()` dengan kueri kosong, sebagaimana ditunjukkan berikut ini.

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

Memulai dengan Amazon DocumentDB

Ada banyak cara untuk terhubung dan memulai dengan Amazon DocumentDB. Kami membuat panduan ini karena kami menemukan cara ini sebagai cara tercepat, paling sederhana dan termudah bagi pengguna untuk mulai menggunakan basis data dokumen kami yang kuat. Panduan ini menggunakan AWS Cloud9, terminal berbasis web untuk menghubungkan dan mengajukan kueri untuk klaster Amazon DocumentDB Anda dengan menggunakan mongo shell secara langsung dari AWS Management Console. Pelanggan baru yang memenuhi syarat untuk Tingkat AWS Gratis dapat menggunakan Amazon DocumentDB AWS Cloud9 dan gratis. Jika AWS Cloud9 lingkungan Anda atau klaster Amazon DocumentDB menggunakan sumber daya di luar tingkat gratis, Anda akan dikenakan tarif AWS normal untuk sumber daya tersebut. Panduan ini akan membantu Anda memulai Amazon DocumentDB dalam waktu kurang dari 15 menit.

> **Note**
>
> Petunjuk dalam panduan ini khusus untuk membuat dan menghubungkan ke cluster berbasis instans Amazon DocumentDB. Jika Anda ingin membuat dan terhubung ke cluster elastis Amazon DocumentDB, lihat. Memulai dengan cluster elastis Amazon DocumentDB

Topik

- Prasyarat
- Langkah 1: Buat AWS Cloud9 lingkungan
- Langkah 2: Membuat grup keamanan
- Langkah 3: Buat klaster Amazon DocumentDB
- Langkah 4: Instal shell mongo
- Langkah 5: Buat koneksi ke klaster Amazon DocumentDB Anda
- Langkah 6: Masukkan dan ajukan kueri data
- Langkah 7: Jelajahi

Jika Anda lebih memilih untuk terhubung ke Amazon DocumentDB dari mesin lokal dengan membuat koneksi SSH ke instans Amazon EC2, silakan lihat petunjuk Hubungkan dengan EC2

Prasyarat

Sebelum membuat klaster Amazon DocumentDB pertama, Anda harus melakukan hal berikut:

Buatlah akun Amazon Web Services (AWS)

Sebelum Anda dapat mulai menggunakan Amazon DocumentDB, Anda harus memiliki akun Amazon Web Services (AWS). AWS Akun ini gratis. Anda hanya membayar untuk layanan dan sumber daya yang Anda gunakan.

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Siapkan izin yang diperlukan AWS Identity and Access Management (IAM).

Akses untuk mengelola sumber daya Amazon DocumentDB seperti cluster, instance, dan grup parameter cluster memerlukan kredensional yang dapat digunakan untuk AWS mengautentikasi permintaan Anda. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon DocumentDB.

1. Di bilah pencarian AWS Management Console, ketik IAM dan pilih IAM di menu tarik-turun yang muncul.
2. Setelah Anda berada di konsol IAM, pilih Pengguna dari panel navigasi.
3. Pilih nama pengguna Anda.
4. Klik tombol Tambahkan izin.
5. Pilih Lampirkan kebijakan yang ada secara langsung.

6. Ketik `AmazonDocDBFullAccess` di bilah pencarian dan pilih itu setelah muncul di hasil pencarian.
7. Klik tombol biru di bagian bawah yang bertuliskan Berikutnya: Tinjau.
8. Klik tombol biru di bagian bawah yang bertuliskan Tambahkan izin.

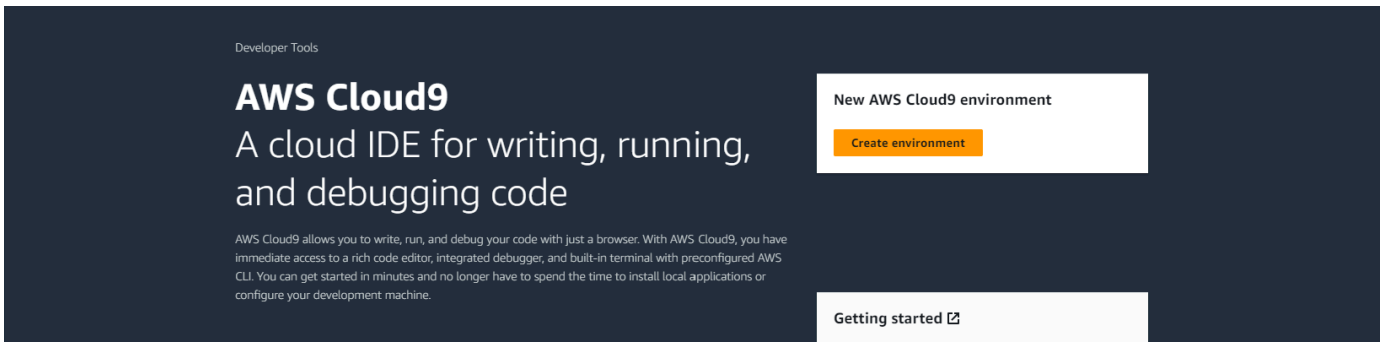
Buat Amazon Virtual Private Cloud (Amazon VPC)

Langkah ini diperlukan hanya jika Anda belum memiliki Amazon VPC default. Jika tidak, selesaikan langkah 1 dari Memulai Amazon VPC di Panduan Pengguna Amazon VPC. Ini akan memakan waktu kurang dari lima menit.

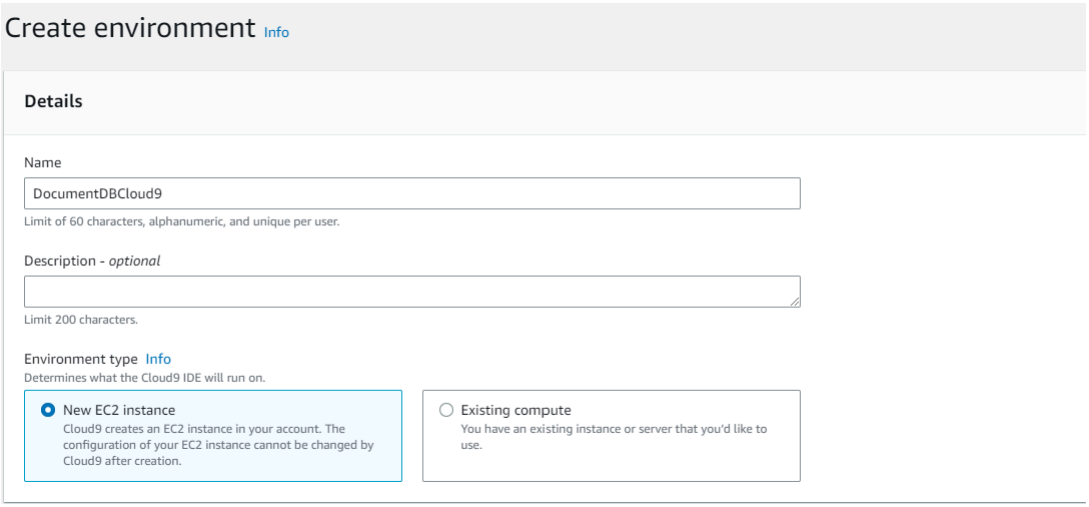
Langkah 1: Buat AWS Cloud9 lingkungan

AWS Cloud9 menyediakan terminal berbasis web yang dapat Anda gunakan untuk menghubungkan dan menanyakan klaster Amazon DocumentDB Anda menggunakan shell mongo.

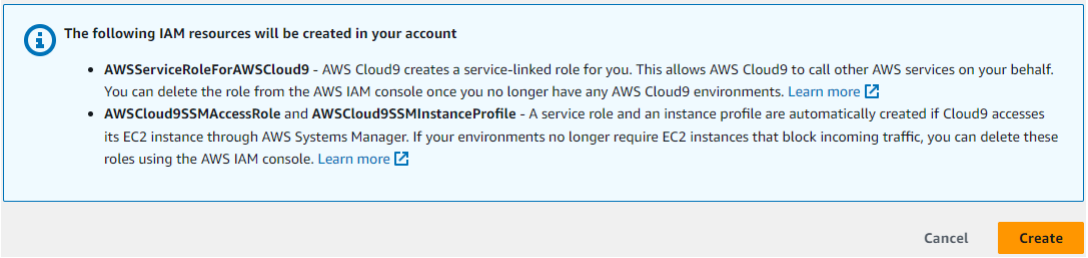
1. Dari AWS Management Console navigasi ke AWS Cloud9 konsol dan pilih Buat lingkungan.



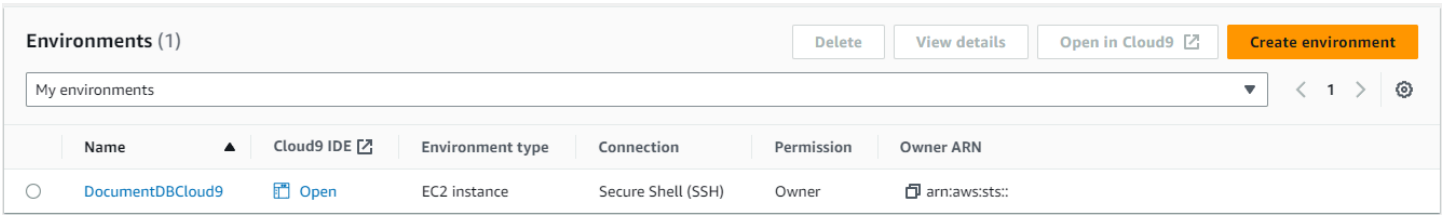
2. Di bagian Detail dari dialog Buat lingkungan, masukkan `DocumentDBCloud9` di bidang Nama.



3. Untuk bagian Instans EC2 Baru, Pengaturan Jaringan, dan Tag, biarkan pengaturan default apa adanya dan klik Buat di bagian bawah layar.



AWS Cloud9 Lingkungan baru Anda muncul di tabel Lingkungan:

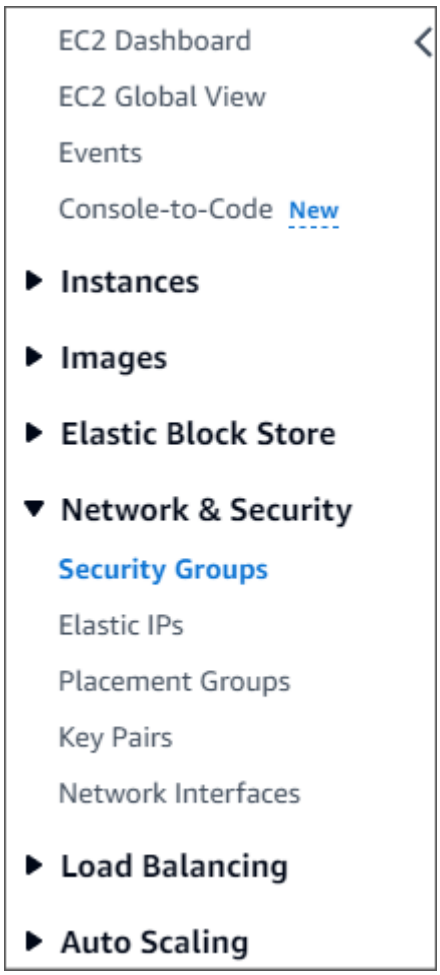


> **Note**
>
> Penyediaan AWS Cloud9 lingkungan bisa memakan waktu hingga tiga menit.

Langkah 2: Membuat grup keamanan

Grup keamanan ini akan memungkinkan Anda untuk terhubung ke klaster Amazon DocumentDB Anda dari lingkungan AWS Cloud9 Anda.

1. Di Konsol Manajemen Amazon EC2, di bawah Jaringan dan Keamanan, pilih Grup keamanan.



2. Pilih Buat grup keamanan.

3. Di bagian Detail dasar:

   a. Untuk Nama grup keamanan, masukkan `demoDocDB`.

   b. Untuk Deskripsi, masukkan deskripsi.

   c. Untuk VPC, terima penggunaan VPC default Anda.

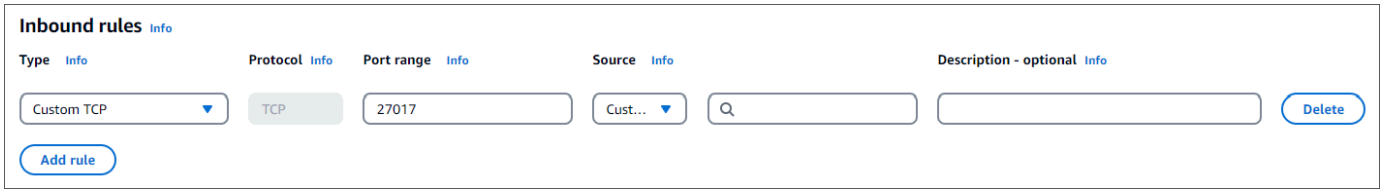
4. Di bagian Aturan masuk, pilih Tambahkan aturan.

   a. Untuk Jenis, pilih Aturan TCP Kustom.

   b. Untuk Rentang port, masukkan `27017`.

   c. Untuk Sumber, pilih grup keamanan untuk AWS Cloud9 lingkungan yang baru saja Anda buat. Untuk melihat daftar grup keamanan yang tersedia, masukkan `cloud9` di bidang pencarian di sisi kanan bidang Sumber. Pilih grup keamanan dengan nama `aws-cloud9-<environment name>`.

   d. Untuk Tujuan, pilih Kustom. Di bidang di sebelahnya, cari grup keamanan yang baru saja Anda buat bernama `demoEC2`. Anda mungkin perlu untuk me-refresh peramban Anda bagi konsol Amazon EC2 untuk mengisi otomatis nama sumber `demoEC2`.



> **Note**
>
> Port 27017 adalah port default untuk Amazon DocumentDB.

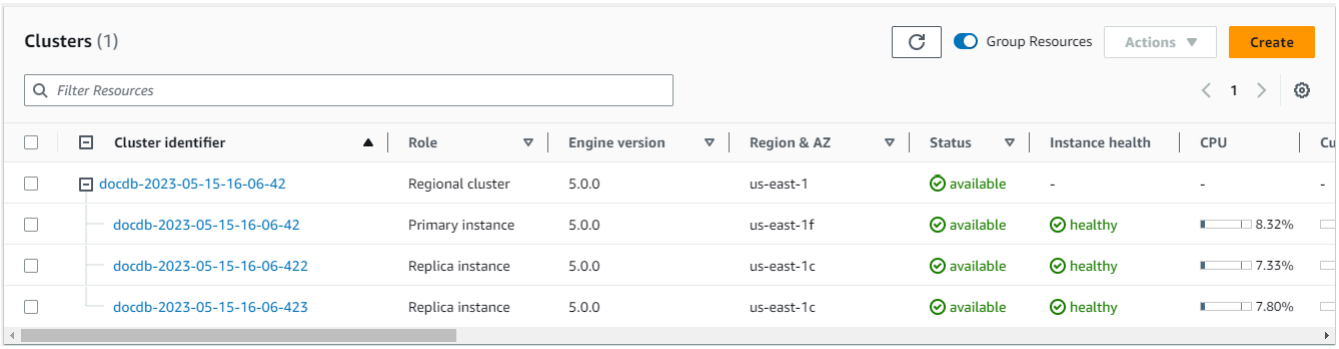
5. Terima semua default lainnya dan pilih Buat grup keamanan.

Langkah 3: Buat klaster Amazon DocumentDB

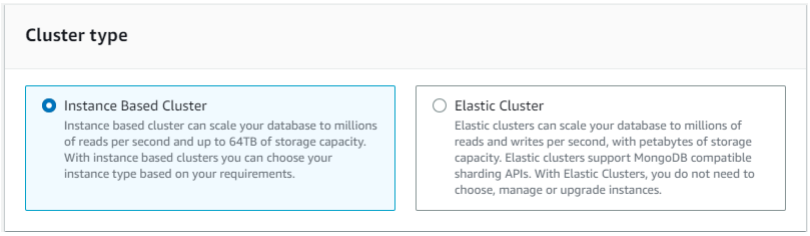
Pada langkah ini, Anda akan membuat klaster Amazon DocumentDB dengan menggunakan grup keamanan yang Anda buat di langkah sebelumnya.

> **Note**
>
> Petunjuk dalam langkah ini khusus untuk membuat cluster berbasis instans Amazon DocumentDB. Jika Anda ingin membuat cluster elastis Amazon DocumentDB, lihat. Memulai dengan cluster elastis Amazon DocumentDB

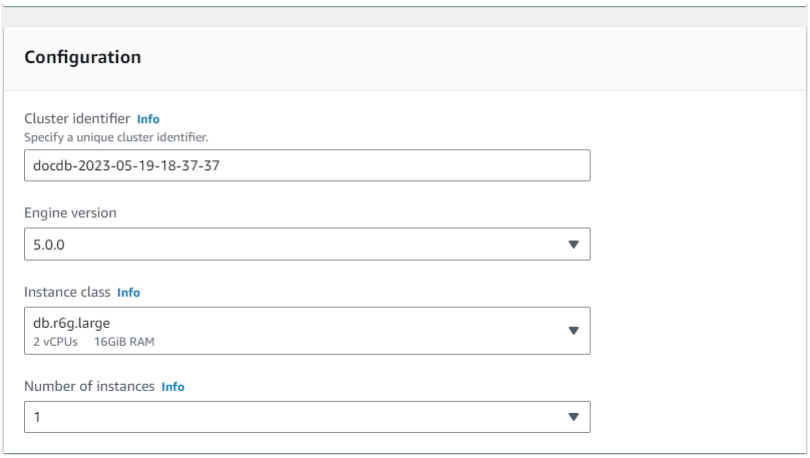
1. Di konsol manajemen Amazon DocumentDB, di bawah Klaster, pilih Buat.



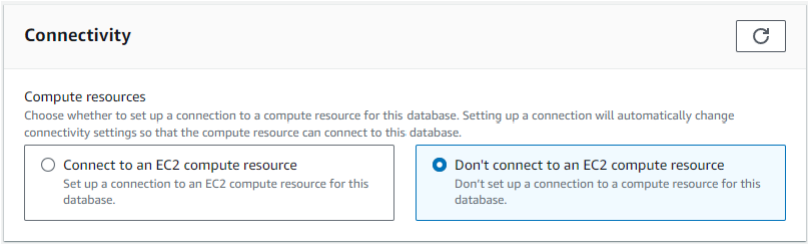
2. Pada halaman cluster Create Amazon DocumentDB, di bagian tipe Cluster, pilih Instance Based Clusters (ini adalah opsi default).



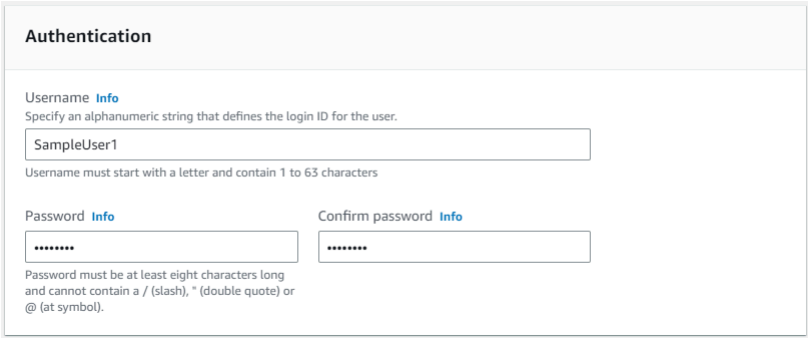
3. Di bagian Konfigurasi, pilih 1 instance. Memilih satu instans membantu meminimalkan biaya. Jika ini adalah sistem produksi, kami sarankan Anda menyediakan tiga instans untuk ketersediaan tinggi. Anda dapat meninggalkan pengaturan lain di bagian Konfigurasi pada defaultnya.



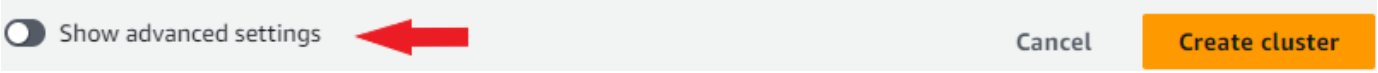
4. Untuk Konektivitas, biarkan pengaturan default Jangan sambungkan ke sumber daya komputasi EC2.



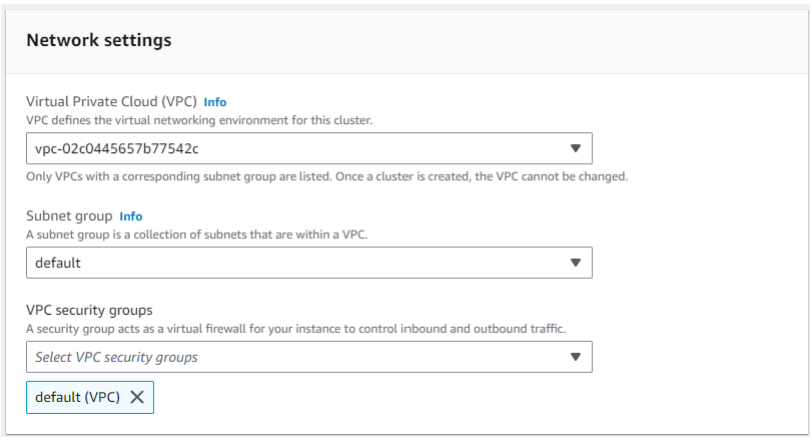
5. Di bagian Otentikasi, masukkan kredensi masuk.



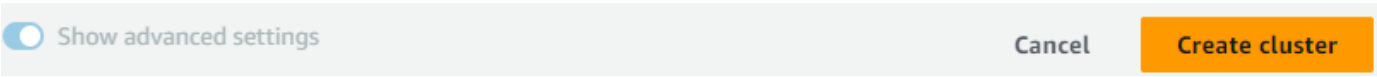
6. Hidupkan Tampilkan pengaturan lanjutan.



7. Di bagian Pengaturan jaringan, untuk grup keamanan VPC, pilih DemoDocDB (VPC) jika Anda membuat cluster pengujian atau demo. Jika Anda membuat klaster untuk sistem produksi, pilih default (VPC) atau jika Anda ingin membuat grup keamanan VPC tertentu, lihat Grup keamanan di Panduan Pengguna Amazon Virtual Private Cloud.



8. Pilih Buat klaster.



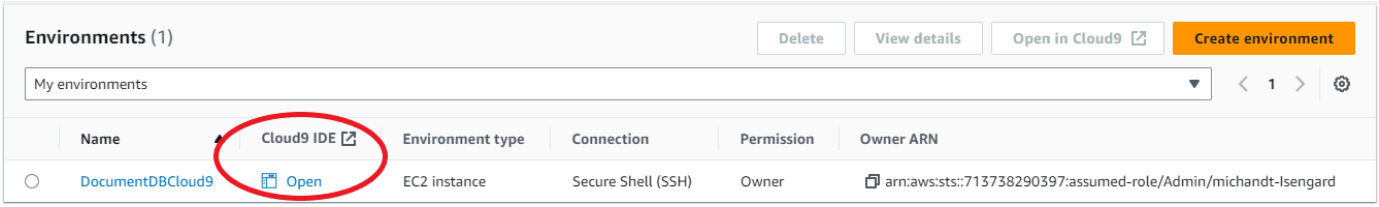
Amazon DocumentDB sekarang menyediakan klaster Anda, dan penyelesaian proses ini dapat memakan waktu hingga beberapa menit. Anda dapat membuat koneksi ke klaster ketika klaster dan instans sama-sama menunjukkan status **`available`**.

> **Note**
>
> Untuk informasi tentang nilai status klaster, lihat Nilai Status Klaster di bagian Memantau Amazon DocumentDB.
> Untuk informasi tentang nilai status instans, lihat Nilai status status status status status status status status di bagian Memantau Amazon DocumentDB.

Langkah 4: Instal shell mongo

Anda sekarang akan menginstal shell mongo di AWS Cloud9 lingkungan Anda yang Anda buat di Langkah 1. Mongo shell adalah utilitas baris perintah yang Anda gunakan untuk menghubungkan dan mengajukan kueri pada klaster Amazon DocumentDB Anda.

1. Jika AWS Cloud9 lingkungan Anda masih terbuka dari Langkah 1, kembali ke lingkungan itu dan lompat ke instruksi 3. Jika Anda menavigasi jauh dari AWS Cloud9 lingkungan Anda, di konsol AWS Cloud9 manajemen, di bawah Lingkungan, temukan lingkungan berlabel DocumentDbCloud9. Pilih Buka di kolom Cloud9 IDE.



2. Pada prompt perintah, buat file repositori dengan perintah berikut:

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

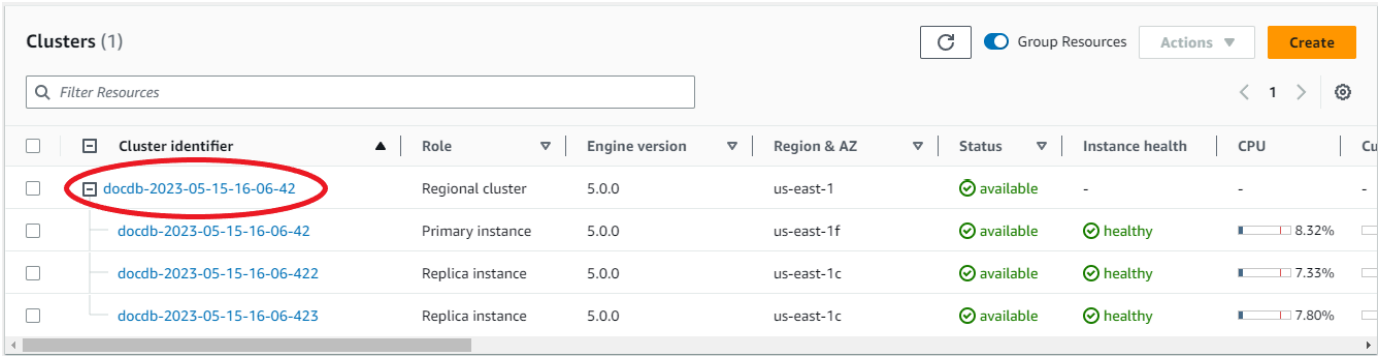
3. Setelah selesai, instal mongo shell dengan perintah berikut:

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

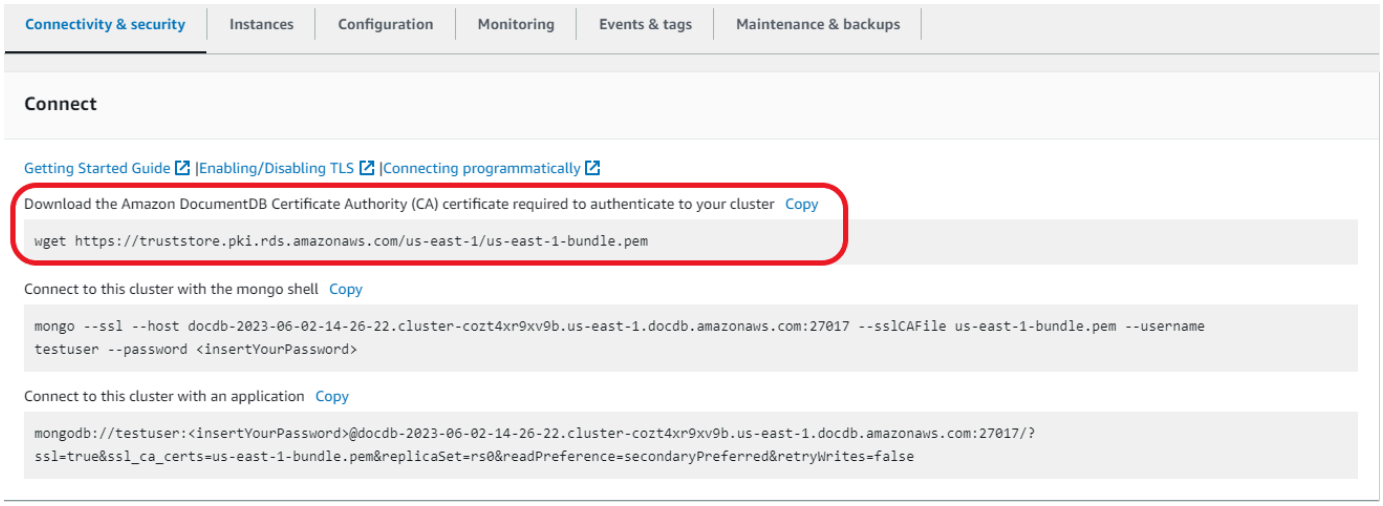
Langkah 5: Buat koneksi ke klaster Amazon DocumentDB Anda

Anda sekarang akan terhubung ke klaster Amazon DocumentDB dengan menggunakan mongo shell yang Anda instal di Langkah 4.

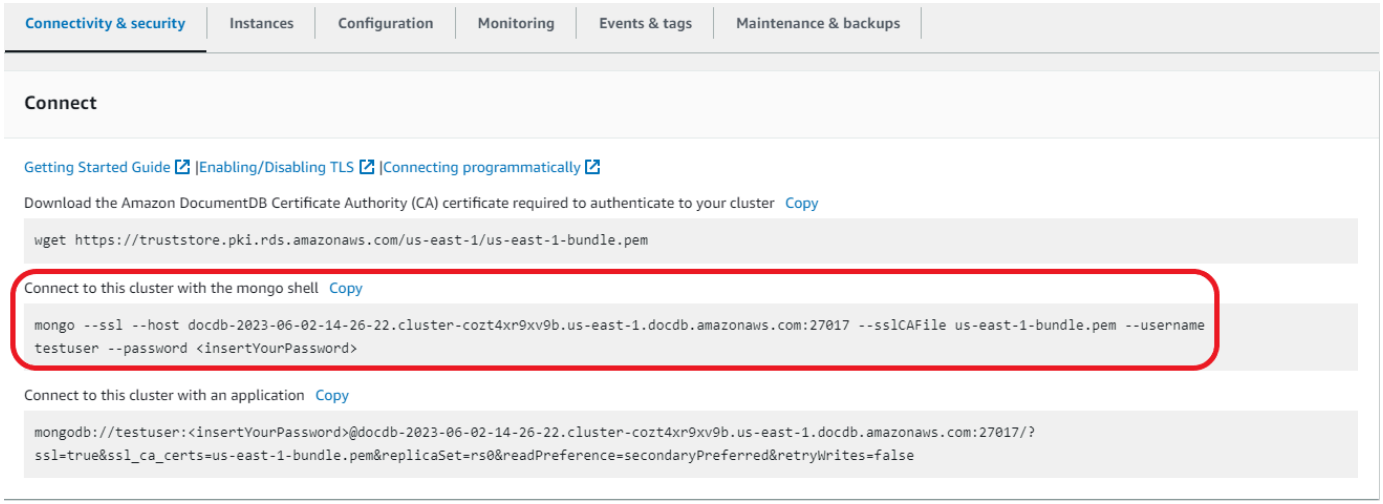
1. Di konsol manajemen Amazon DocumentDB, di bawah Klaster, temukan klasster Anda. Pilih klaster yang Anda buat dengan mengeklik pengidentifikasi klaster.



2. E ncryption-in-transit diaktifkan secara default di Amazon DocumentDB. Anda dapat menonaktifkan TLS secara opsional. Untuk mengunduh sertifikat saat ini yang diperlukan untuk mengautentikasi ke klaster Anda, di tab Konektivitas & keamanan, di bagian Connect, di bawah Unduh sertifikat Amazon DocumentDB Certificate Authority (CA) yang diperlukan untuk mengautentikasi ke cluster Anda, salin string koneksi yang disediakan. Kembali ke AWS Cloud9 lingkungan Anda dan tempel string koneksi.



3. Kembali ke cluster Anda di konsol Amazon DocumentDB, buka tab Konektivitas & keamanan, di bagian Connect, di bawah Connect to this cluster dengan shell mongo, salin string koneksi yang disediakan. Abaikan penyalinan `<insertYourPassword>` sehingga Anda dimintai kata sandi oleh mongo shell saat Anda terhubung.



   Kembali ke AWS Cloud9 lingkungan Anda dan tempel string koneksi.

Saat Anda memasukkan kata sandi dan prompt Anda menjadi prompt `rs0:PRIMARY>`, Anda berhasil terhubung ke klaster Amazon DocumentDB Anda.

> **Note**
>
> Untuk informasi pemecahan masalah, lihat Memecahkan Masalah Amazon DocumentDB.

Langkah 6: Masukkan dan ajukan kueri data

Sekarang setelah Anda terhubung ke klaster, Anda dapat menjalankan beberapa kueri untuk mengenali penggunaan basis data dokumen.

1. Untuk memasukkan satu dokumen, masukkan hal berikut ini:

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

2. Anda mendapatkan output berikut ini:

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

3. Anda dapat membaca dokumen yang Anda tulis dengan perintah `findOne()` (karena hanya mengembalikan satu dokumen). Masukkan yang berikut ini:

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

4. Anda mendapatkan output berikut ini:

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

5. Untuk mengajukan beberapa kueri lagi, pertimbangkan kasus penggunaan profil game. Pertama, masukkan beberapa entri ke dalam koleksi berjudul `profiles`. Masukkan yang berikut ini:

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

6. Anda mendapatkan output berikut ini:

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

7. Gunakan perintah `find()` untuk mengembalikan semua dokumen dalam koleksi profil. Masukkan yang berikut ini:

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

8. Anda akan mendapatkan output yang akan cocok dengan data yang Anda ketik di Langkah 5.
9. Gunakan kueri untuk satu dokumen dengan menggunakan filter. Masukkan yang berikut ini:

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

10. Anda harus mendapatkan kembali output ini:

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

11. Sekarang mari kita coba untuk menemukan profil dan memodifikasinya dengan menggunakan perintah `findAndModify`. Kami akan memberi pengguna Matt sepuluh poin tambahan dengan kode berikut:

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

12. Anda mendapatkan output berikut (perhatikan bahwa skornya belum meningkat):

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

13. Anda dapat memverifikasi bahwa skornya telah berubah dengan kueri berikut:

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

14. Anda mendapatkan output berikut ini:

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

```
                        : 212 }
```

Langkah 7: Jelajahi

Selamat! Anda telah berhasil menyelesaikan Panduan Memulai untuk Amazon DocumentDB.

Apa selanjutnya? Pelajari cara memanfaatkan basis data ini sepenuhnya dengan beberapa fitur populernya:

- Mengelola Amazon DocumentDB
- Penskalaan
- Mencadangkan dan memulihkan

> **Note**
>
> Klaster yang Anda buat sejak memulai latihan ini akan terus dikenakan biaya kecuali Anda menghapusnya. Untuk mengetahui petunjuknya, lihat Menghapus Klaster Amazon DocumentDB.

Amazon DocumentDB Mulai Cepat Menggunakan AWS CloudFormation

Bagian ini berisi langkah-langkah dan informasi lain untuk membantu Anda memulai dengan cepat menggunakan Amazon DocumentDB (dengan kompatibilitas MongoDB) menggunakan. AWS CloudFormation Untuk informasi umum tentang Amazon DocumentDB, lihat. Apa Itu Amazon DocumentDB (dengan Kompatibilitas MongoDB)

Petunjuk ini menggunakan AWS CloudFormation template untuk membuat cluster dan instance di Amazon VPC default Anda. Untuk petunjuk tentang membuat sumber daya ini sendiri, lihat Memulai dengan Amazon DocumentDB.

> **Important**
>
> AWS CloudFormation Tumpukan yang dibuat oleh template ini menciptakan beberapa sumber daya, termasuk sumber daya di Amazon DocumentDB (misalnya, cluster dan instance) dan Amazon Elastic Compute Cloud (misalnya, grup subnet).
> Beberapa sumber daya ini bukanlah sumber daya tingkat gratis. Untuk informasi selengkapnya, lihat Harga Amazon DocumentDB dan Harga Amazon EC2. Anda dapat menghapus tumpukan ketika Anda selesai dengannya untuk menghentikan biaya apa pun.

AWS CloudFormation Tumpukan ini ditujukan untuk tujuan tutorial saja. Jika Anda menggunakan templat ini untuk lingkungan produksi, kami sarankan Anda menggunakan kebijakan IAM dan keamanan yang lebih ketat. Untuk informasi tentang mengamankan sumber daya, lihat Keamanan Amazon VPC dan Jaringan dan Keamanan Amazon EC2.

Topik

- Prasyarat
- Meluncurkan Tumpukan AWS CloudFormation Amazon DocumentDB
- Mengakses Klaster Amazon DocumentDB
- Perlindungan Penghentian dan Perlindungan Penghapusan

Prasyarat

Sebelum Anda membuat cluster Amazon DocumentDB, Anda harus memiliki yang berikut:

- Amazon VPC default
- Izin IAM yang diperlukan

Izin IAM yang Diperlukan

Izin berikut ini memungkinkan Anda untuk membuat sumber daya untuk tumpukan AWS CloudFormation :

AWS Kebijakan Terkelola

- `AWSCloudFormationReadOnlyAccess`
- `AmazonDocDBFullAccess`

Izin IAM Tambahan

Kebijakan berikut menguraikan izin tambahan yang diperlukan untuk membuat dan menghapus tumpukan ini AWS CloudFormation .

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:PutRolePolicy",
                "iam:CreateInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:GetAccountSummary",
                "iam:ListAccountAliases",
                "iam:GetRole",
                "iam:DeleteRole",
```

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

> **Note**
>
> Izin yang digaris tebal dalam kebijakan sebelumnya hanya diperlukan untuk menghapus tumpukan: `iam:DeleteRole`, `iam:RemoveRoleFromInstanceProfile`, `iam:DeleteRolePolicy`, `iam:DeleteInstanceProfile`, dan

`ec2:DeleteVpcEndpoints`. Perhatikan juga bahwa `ec2:*Vpc` memberikan izin `ec2:DeleteVpc`.

Pasangan Kunci Amazon EC2

Anda harus memiliki key pair (dan file PEM) yang tersedia di Region tempat Anda akan membuat AWS CloudFormation tumpukan. Jika Anda perlu membuat key pair, lihat Membuat Pasangan Kunci Menggunakan Amazon EC2 di Panduan Pengguna Amazon EC2.

Meluncurkan Tumpukan AWS CloudFormation Amazon DocumentDB

Bagian ini menguraikan cara untuk memulai dan mengonfigurasi tumpukan AWS CloudFormation Amazon DocumentDB.

1. Masuk ke AWS Management Console athttps://console.aws.amazon.com/.
2. Tabel berikut mencantumkan templat tumpukan Amazon DocumentDB untuk setiap Wilayah AWS. Pilih Launch Stack untuk tempat Wilayah AWS Anda ingin meluncurkan tumpukan Anda.

Wilayah	Lihat Templat	Lihat di Perancang	Luncurkan
AS Timur (Ohio)	Lihat Template	Lihat di Desainer	Launch Stack
US East (N. Virginia)	Lihat Template	Lihat di Desainer	Launch Stack
US West (Oregon)	Lihat Template	Lihat di Desainer	Launch Stack
Asia Pasifik (Mumbai)	Lihat Template	Lihat di Desainer	Launch Stack
Asia Pasifik (Seoul)	Lihat Template	Lihat di Desainer	Launch Stack
Asia Pacific (Singapore)	Lihat Template	Lihat di Desainer	Launch Stack

Wilayah	Lihat Templat	Lihat di Perancang	Luncurkan
Asia Pacific (Sydney)	Lihat Template	Lihat di Desainer	
Asia Pacific (Tokyo)	Lihat Template	Lihat di Desainer	
Kanada (Pusat)	Lihat Template	Lihat di Desainer	
Eropa (Frankfurt)	Lihat Template	Lihat di Desainer	
Eropa (Irlandia)	Lihat Template	Lihat di Desainer	
Eropa (London)	Lihat Template	Lihat di Desainer	
Eropa (Paris)	Lihat Template	Lihat di Desainer	

3. Buat tumpukan — Menguraikan templat Amazon DocumentDB yang Anda pilih. Setiap tumpukan didasarkan pada template - file JSON atau YAML - yang berisi konfigurasi tentang AWS sumber daya yang ingin Anda sertakan dalam tumpukan. Karena Anda memilih untuk meluncurkan tumpukan dari template yang disediakan di atas, template Anda telah dikonfigurasi untuk membuat tumpukan Amazon DocumentDB untuk Wilayah AWS yang Anda pilih.

   Saat Anda meluncurkan AWS CloudFormation tumpukan, perlindungan penghapusan untuk klaster Amazon DocumentDB Anda dinonaktifkan secara default. Jika Anda ingin mengaktifkan perlindungan penghapusan untuk klaster Anda, selesaikan langkah-langkah berikut ini. Jika tidak, pilih Selanjutnya untuk melanjutkan ke langkah berikutnya.

   Untuk mengaktifkan perlindungan penghapusan untuk klaster Amazon DocumentDB Anda:

   1. Pilih Lihat di Perancang dari sudut kanan bawah halaman Buat tumpukan.
   2. Ubah template menggunakan editor JSON dan YAMM terintegrasi di halaman AWS CloudFormation Desainer konsol yang dihasilkan. Gulir ke bagian `Resources` dan ubah untuk menyertakan `DeletionProtection`, sebagai berikut. Untuk informasi selengkapnya tentang menggunakan AWS CloudFormation Desainer, lihat Apa Itu AWS CloudFormation Desainer? .

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. Pilih Buat Tumpukan (

)
dari pojok kiri atas halaman untuk menyimpan perubahan Anda dan buat tumpukan dengan perubahan tersebut diaktifkan.

4. Setelah menyimpan perubahan Anda, Anda akan diarahkan ke halaman Buat tumpukan.

5. Pilih Next untuk melanjutkan.

4. Tentukan detail Tumpukan — Masukkan nama tumpukan dan parameter untuk templat Anda. Parameter ditetapkan dalam templat Anda dan mengizinkan Anda untuk memasukkan nilai kustom ketika Anda membuat atau memperbarui tumpukan.

   - Di bawah Nama Tumpukan, masukkan nama untuk tumpukan Anda atau terima nama yang disediakan. Nama tumpukan dapat mencakup huruf (A—Z dan a—z), angka (0—9), dan tanda hubung (—).
   - Di bawah Parameter, masukkan detail berikut ini:

     - DB ClusterName - Masukkan nama untuk cluster Amazon DocumentDB Anda atau terima nama yang disediakan.

       Aturan penamaan klaster:

       - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
       - Karakter pertama harus berupa huruf.
       - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
       - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

     - DB InstanceClass - Dari daftar drop-down, pilih kelas instans untuk cluster Amazon DocumentDB Anda.
     - DB InstanceName — Masukkan nama untuk instans Amazon DocumentDB Anda atau terima nama yang diberikan.

       Kendala penamaan instans:

       - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
       - Karakter pertama harus berupa huruf.
       - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
       - Harus unik untuk semua instans di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

     - MasterPassword— Kata sandi akun admin database.
     - MasterUser— Nama pengguna akun admin database. MasterUser Harus dimulai dengan huruf dan hanya dapat berisi karakter alfanumerik.

   Pilih Selanjutnya untuk menyimpan perubahan Anda dan melanjutkan.

5. Konfigurasi opsi tumpukan — Konfigurasi tag, izin, dan opsi tambahan tumpukan Anda.

- Tag — Tentukan pasangan tag (nilai kunci) untuk diterapkan ke sumber daya Anda di tumpukan Anda. Anda dapat menambahkan hingga 50 tag unik untuk setiap tumpukan.
- Izin — Opsional. Pilih peran IAM untuk secara eksplisit menentukan cara AWS CloudFormation membuat, memodifikasi, atau menghapus sumber daya dalam tumpukan. Jika Anda tidak memilih peran, AWS CloudFormation gunakan izin berdasarkan kredensil pengguna Anda. Sebelum Anda menentukan peran layanan, pastikan bahwa Anda memiliki izin untuk memberikannya (`iam:PassRole`). Izin `iam:PassRole` menentukan peran yang dapat Anda gunakan.

  > **Note**
  >
  > Saat Anda menentukan peran layanan, AWS CloudFormation selalu gunakan peran itu untuk semua operasi yang dilakukan pada tumpukan tersebut. Pengguna lainnya yang memiliki izin untuk melakukan operasi pada tumpukan ini akan dapat menggunakan peran ini, meskipun mereka tidak memiliki izin untuk meneruskannya. Jika peran tersebut mencakup izin yang tidak dimiliki pengguna, Anda dapat secara tidak sengaja meningkatkan izin pengguna. Pastikan bahwa peran memberikan hak istimewa paling rendah..

- Opsi lanjutan — Anda dapat mengatur opsi lanjutan berikut ini:
  - Kebijakan tumpukan — Opsional. Menetapkan sumber daya yang Anda ingin lindungi dari pembaruan yang tidak disengaja selama pembaruan tumpukan. Secara default, semua sumber daya dapat diperbarui selama pembaruan tumpukan.

    Anda dapat memasukkan kebijakan tumpukan secara langsung sebagai JSON, atau unggah file JSON yang berisi kebijakan tumpukan. Untuk informasi selengkapnya, lihat Mencegah Pembaruan untuk Sumber Daya Tumpukan.
  - Konfigurasi rollback — Opsional. Tentukan alarm CloudWatch Log AWS CloudFormation untuk dipantau saat membuat dan memperbarui tumpukan. Jika operasi melanggar ambang alarm, putar AWS CloudFormation kembali.
  - Opsi notifikasi — Opsional. Tentukan topik untuk Simple Notification System (SNS).
  - Opsi pembuatan tumpukan — Opsional. Anda dapat menentukan opsi berikut ini:
    - Rollback pada kegagalan — Apakah tumpukan harus dikembalikan jika pembuatan tumpukan gagal.
    - Waktu habis —Jumlah menit sebelum waktu pembuatan tumpukan habis.

- Perlindungan penghentian — Mencegah tumpukan tidak sengaja dihapus.

> **Note**
>
> AWS CloudFormation perlindungan terminasi berbeda dari konsep perlindungan penghapusan Amazon DocumentDB. Untuk informasi selengkapnya, lihat Perlindungan Penghentian dan Perlindungan Penghapusan.

Pilih Next untuk melanjutkan.

6. Tinjauan <nama–tumpukan> — Tinjau templat, detail, dan opsi konfigurasi tumpukan Anda. Anda juga dapat membuka tautan buat cepat pada bagian bawah halaman untuk membuat tumpukan dengan konfigurasi dasar yang sama seperti yang satu ini.

   - Pilih Buat untuk membuat tumpukan.
   - Sebagai alternatif, Anda dapat memilih Buat set perubahan. Set perubahan adalah pratinjua tentang bagaimana tumpukan ini akan dikonfigurasi sebelum membuat tumpukan. Hal ini mengizinkan Anda untuk memeriksa berbagai konfigurasi sebelum menjalankan set perubahan.

Mengakses Klaster Amazon DocumentDB

Setelah AWS CloudFormation tumpukan selesai, Anda dapat menggunakan instans Amazon EC2 untuk terhubung ke cluster Amazon DocumentDB Anda. Untuk informasi tentang menghubungkan ke instans Amazon EC2 menggunakan SSH, lihat Connect to Your Linux Instance di Panduan Pengguna Amazon EC2.

Setelah Anda terhubung, lihat bagian berikut ini, yang berisi informasi tentang menggunakan Amazon DocumentDB.

- Langkah 4: Instal shell mongo
- Menghapus cluster Amazon DocumentDB

Perlindungan Penghentian dan Perlindungan Penghapusan

Ini adalah praktik terbaik Amazon DocumentDB untuk mengaktifkan perlindungan penghapusan dan perlindungan penghentian. CloudFormation perlindungan terminasi adalah fitur yang sangat berbeda dari fitur perlindungan penghapusan Amazon DocumentDB.

- Perlindungan terminasi — Anda dapat mencegah tumpukan dihapus secara tidak sengaja dengan mengaktifkan perlindungan terminasi untuk CloudFormation tumpukan Anda. Jika pengguna mencoba untuk menghapus tumpukan dengan penghentian perlindungan yang diaktifkan padanya, penghapusan gagal dan tumpukan tetap tidak berubah. Perlindungan penghentian dinonaktifkan secara default saat Anda membuat tumpukan menggunakan CloudFormation. Anda dapat mengaktifkan perlindungan terminasi di tumpukan saat membuatnya. Untuk informasi selengkapnya, lihat Mengatur Opsi AWS CloudFormation Tumpukan.
- Perlindungan penghapusan — Amazon DocumentDB juga menyediakan kemampuan untuk mengaktifkan perlindungan penghapusan untuk klaster. Jika pengguna mencoba untuk menghapus klaster Amazon DocumentDB dengan perlindungan penghapusan yang diaktifkan padanya, penghapusan gagal dan klaster tetap tidak berubah. Perlindungan penghapusan, saat diaktifkan, melindungi terhadap penghapusan yang tidak disengaja dari Amazon DocumentDB,, dan. AWS Management Console AWS CLI CloudFormation Untuk informasi lebih lanjut tentang mengaktifkan dan menonaktifkan perlindungan penghapusan untuk klaster Amazon DocumentDB, lihat Perlindungan penghapusan.

Kompatibilitas MongoDB

Amazon DocumentDB mendukung kompatibilitas MongoDB termasuk MongoDB 4.0 dan MongoDB 5.0. Kompatibilitas MongoDB berarti bahwa sebagian besar aplikasi, driver, dan alat yang sudah Anda gunakan hari ini dengan database MongoDB Anda dapat digunakan dengan Amazon DocumentDB dengan sedikit atau tanpa perubahan. Bagian ini menjelaskan semua yang perlu Anda ketahui tentang kompatibilitas Amazon DocumentDB dengan MongoDB termasuk kemampuan dan fitur baru, memulai, jalur migrasi, dan perbedaan fungsional.

Topik

- MongoDB 5.0 Kompatibilitas
- Kompatibilitas MongoDB 4.0

MongoDB 5.0 Kompatibilitas

Topik

- Apa yang baru di Amazon DocumentDB 5.0
- Memulai Amazon DocumentDB 5.0
- Meningkatkan atau Memigrasi ke Amazon DocumentDB 4.0
- Perbedaan Fungsional

Apa yang baru di Amazon DocumentDB 5.0

Amazon DocumentDB 5.0 memperkenalkan fitur dan kemampuan baru yang mencakup batas penyimpanan dan enkripsi tingkat bidang sisi klien. Ringkasan di bawah ini memperkenalkan beberapa fitur utama yang diperkenalkan di Amazon DocumentDB 5.0. Untuk melihat daftar lengkap kemampuan baru, lihat Catatan Rilis.

- Peningkatan batas penyimpanan menjadi 128 TiB untuk semua cluster Amazon DocumentDB berbasis instans dan cluster elastis berbasis shard.
- Memperkenalkan Amazon DocumentDB 5.0 Engine Versi 3.0.775)
  - Dukungan untuk driver MongoDB 5.0 API

- Support untuk Client-side Field Level Encryption (FLE). Anda sekarang dapat mengenkripsi bidang di sisi klien sebelum menulis data ke cluster Amazon DocumentDB. Untuk informasi selengkapnya, lihat Enkripsi tingkat bidang sisi klien
- Operator agregasi baru:`$dateAdd`, `$dateSubtract`
- Mendukung indeks dengan `$elemMatch` operator. Akibatnya, kueri yang memiliki `$elemMatch` akan menghasilkan pemindaian indeks.

Amazon DocumentDB tidak mendukung setiap fitur MongoDB 5.0. Ketika kami membangun Amazon DocumentDB 5.0, kami bekerja mundur dari fitur dan kemampuan yang diminta pelanggan kami untuk membangun paling banyak. Kami akan terus menambahkan kemampuan MongoDB 5.0 tambahan berdasarkan apa yang diminta pelanggan untuk kami bangun. Untuk daftar terbaru API yang didukung, silakan lihat API, Operasi, dan Tipe Data MongoDB Yang Didukung.

Memulai Amazon DocumentDB 5.0

Untuk memulai Amazon DocumentDB 5.0, silakan lihat Panduan Memulai. Anda dapat membuat klaster Amazon DocumentDB 5.0 baru menggunakan atau SDK AWS Management Console AWS CLI, AWS , atau. AWS CloudFormation Saat menghubungkan ke Amazon DocumentDB, Anda harus menggunakan driver MongoDB atau utilitas yang kompatibel dengan MongoDB 5.0 atau lebih tinggi.

> **Note**
>
> Saat menggunakan AWS SDK,, atau AWS CLI AWS CloudFormation, versi mesin akan default ke 5.0.0. Anda harus secara eksplisit menentukan parameter `engineVersion = 4.0.0` untuk membuat cluster Amazon DocumentDB 4.0 baru `engineVersion = 3.6.0` atau untuk membuat cluster Amazon DocumentDB 3.6 baru. Untuk klaster Amazon DocumentDB tertentu, Anda dapat menentukan versi cluster menggunakan to `describe-db-clusters` call atau menggunakan AWS CLI konsol manajemen Amazon DocumentDB untuk melihat nomor versi engine untuk klaster tertentu.

Amazon DocumentDB 5.0 mendukung prosesor Amazon EC2 Graviton2 `r6g` seperti `t4.medium` dan jenis instans untuk cluster Anda dan tersedia di semua wilayah yang didukung. Untuk informasi selengkapnya tentang harga, lihat Harga Amazon DocumentDB (dengan kompatibilitas MongoDB).

Meningkatkan atau Memigrasi ke Amazon DocumentDB 4.0

Anda dapat bermigrasi dari MongoDB 3.6 atau MongoDB 4.0 ke Amazon DocumentDB 5.0 menggunakan atau utilitas seperti,,, dan. AWS DMS`mongodump` `mongorestore` `mongoimport` `mongoexport` Untuk petunjuk tentang cara bermigrasi, lihat Memutakhirkan cluster Amazon DocumentDB Anda menggunakan AWS Database Migration Service.

Perbedaan Fungsional

Perbedaan Fungsional Antara Amazon DocumentDB 4.0 dan 5.0

Dengan dirilisnya Amazon DocumentDB 5.0, ada perbedaan fungsional antara Amazon DocumentDB 3.6 dan Amazon DocumentDB 4.0:

- Peran bawaan cadangan sekarang mendukung`serverStatus`. Tindakan - Pengembang dan aplikasi dengan peran cadangan dapat mengumpulkan statistik tentang keadaan cluster Amazon DocumentDB.
- `SecondaryDelaySecs`Bidang menggantikan `slaveDelay` `replSetGetConfig` output.
- helloPerintah menggantikan `isMaster` - hello mengembalikan dokumen yang menjelaskan peran cluster Amazon DocumentDB.
- Amazon DocumentDB 5.0 sekarang mendukung pemindaian indeks dengan operator di tingkat `$elemMatch` bersarang pertama. Pemindaian indeks didukung ketika filter kueri hanya memiliki satu tingkat `$elemMatch` filter tetapi tidak didukung jika `$elemMatch` kueri bersarang disertakan.

  Misalnya, di Amazon DocumentDB 5.0, jika Anda menyertakan `$elemMatch` operator di level bersarang, itu tidak akan mengembalikan nilai seperti yang terjadi di Amazon DocumentDB 4.0:

```
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 ] } ] ] }
```

- Proyeksi “$” di Amazon DocumentDB 4.0 mengembalikan semua dokumen dengan semua bidang. Dengan Amazon DocumentDB 5.0, find perintah dengan proyeksi “$” mengembalikan dokumen yang cocok dengan parameter kueri yang hanya berisi bidang yang cocok dengan proyeksi “$”.
- Di Amazon DocumentDB 5.0, find perintah `$regex` dengan `$options` dan parameter kueri mengembalikan kesalahan: “Tidak dapat mengatur opsi di `$regex` keduanya dan”. `$options`
- Dengan Amazon DocumentDB 5.0`$indexOfCP`, sekarang mengembalikan “-1" ketika:
  - substring tidak ditemukan dalam ekspresi string, atau
  - awal adalah angka yang lebih besar dari akhir, atau
  - start adalah angka yang lebih besar dari panjang byte string.
- Di Amazon DocumentDB 4.0`$indexOfCP`, mengembalikan “0" ketika posisi awal adalah angka yang lebih besar dari akhir atau panjang byte string.
- Dengan Amazon DocumentDB 5.0, operasi proyeksi `_id fields` di, `{"_id.nestedField" : 1}` misalnya, mengembalikan dokumen yang hanya menyertakan bidang yang diproyeksikan. Sedangkan di Amazon DocumentDB 4.0, perintah proyeksi bidang bersarang tidak memfilter dokumen apa pun.

Kompatibilitas MongoDB 4.0

Topik

- Fitur Amazon DocumentDB 4.0
- Memulai dengan Amazon DocumentDB 4.0
- Meningkatkan atau Memigrasi ke Amazon DocumentDB 4.0
- Perbedaan Fungsional

Fitur Amazon DocumentDB 4.0

Amazon DocumentDB 4.0 memperkenalkan banyak fitur dan kemampuan baru yang mencakup transaksi ACID dan peningkatan untuk mengubah aliran. Ringkasan di bawah ini menunjukkan beberapa fitur utama yang diperkenalkan di Amazon DocumentDB 4.0. Untuk melihat daftar lengkap kemampuan, lihatCatatan Rilis.

- Transaksi ACID: Amazon DocumentDB sekarang mendukung kemampuan untuk melakukan transaksi di beberapa dokumen, pernyataan, koleksi, dan basis data. Transaksi menyederhanakan pengembangan aplikasi dengan memungkinkan Anda untuk melakukan operasi atomik, konsisten, terisolasi, dan tahan lama (ACID) di satu atau lebih dokumen dalam klaster Amazon DocumentDB. Untuk informasi selengkapnya, lihat Transaksi.
- Aliran perubahan: Anda sekarang memiliki kemampuan untuk membuka aliran perubahan di tingkat klaster (`client.watch()` atau `mongo.watch()`) dan basis data (`db.watch()`), Anda dapat menentukan `startAtOperationTime` untuk membuka kursor aliran perubahan, dan terakhir Anda sekarang dapat memperpanjang periode retensi aliran perubahan menjadi 7 hari (sebelumnya 24 jam). Untuk informasi selengkapnya, lihat Menggunakan Aliran Perubahan dengan Amazon DocumentDB.
- AWS Database Migration Service(AWS DMS): Anda sekarang dapat menggunakan AWS DMS untuk memigrasikan beban kerja MongoDB 4.0 Anda ke Amazon DocumentDB. AWS DMS sekarang mendukung sumber MongoDB 4.0, target Amazon DocumentDB 4.0, dan sumber Amazon DocumentDB 3.6 untuk melakukan peningkatan antara Amazon DocumentDB 3.6 dan 4.0. Untuk informasi selengkapnya, lihat Dokumentasi AWS DMS .
- Performa dan pengindeksan: Anda sekarang dapat memanfaatkan indeks dengan `$lookup`, menemukan kueri dengan proyeksi yang berisi satu bidang atau satu bidang dan bidang `_id` dapat dilayani langsung dari indeks dan tanpa perlu membaca dari koleksi (kueri tercakup), kemampuan untuk `hint()` dengan `findAndModify`, optimasi performa untuk `$addToSet`, dan peningkatan untuk mengurangi ukuran indeks secara keseluruhan. Untuk informasi selengkapnya, lihat Catatan Rilis.
- Operator: Amazon DocumentDB 4.0 sekarang mendukung sejumlah operator agregasi baru: `$ifNull`, `$replaceRoot`, `$setIsSubset`, `$setIntersection`, `$setUnion`, `$setEquals`. Anda dapat melihat semua API, Operasi, dan Tipe Data MongoDB yang kami dukung di API, Operasi, dan Tipe Data MongoDB Yang Didukung.
- Kontrol akses berbasis peran (RBAC): Dengan perintah `ListCollection` dan `ListDatabase` Anda sekarang dapat secara opsional menggunakan parameter `authorizedCollections` dan `authorizedDatabases` untuk memungkinkan pengguna membuat daftar koleksi dan basis

data yang memiliki izin untuk mereka akses tanpa memerlukan peran `listCollections` dan `listDatabase`, masing-masing. Anda juga memiliki kemampuan untuk menghilangkan kursor Anda sendiri tanpa memerlukan peran `KillCursor`.

Amazon DocumentDB tidak mendukung setiap fitur MongoDB 4.0. Ketika kami membangun Amazon DocumentDB 4.0, kami bekerja ke belakang dari fitur dan kemampuan yang paling banyak diminta pelanggan kami untuk kami bangun. Kami akan terus menambahkan kemampuan MongoDB 4.0 tambahan berdasarkan apa yang diminta pelanggan untuk kami bangun. Misalnya, Amazon DocumentDB 4.0 saat ini tidak mendukung operator konversi tipe atau operator string yang diperkenalkan di MongoDB 4.0. Untuk daftar terbaru API yang didukung, silakan lihat API, Operasi, dan Tipe Data MongoDB Yang Didukung.

Memulai dengan Amazon DocumentDB 4.0

Untuk memulai dengan Amazon DocumentDB 4.0, silakan lihat Panduan Memulai. Anda dapat membuat klaster Amazon DocumentDB 4.0 baru menggunakan atau SDK AWS Management Console AWS CLI, AWS , atau. AWS CloudFormation Saat menghubungkan ke Amazon DocumentDB, Anda harus menggunakan driver atau utilitas MongoDB yang kompatibel dengan MongoDB 4.0 atau yang lebih tinggi.

> **Note**
>
> Saat menggunakan AWS SDK,, atau AWS CLI AWS CloudFormation, versi mesin akan default ke 5.0.0. Anda harus secara eksplisit menentukan parameter `engineVersion = 4.0.0` untuk membuat cluster Amazon DocumentDB 4.0 baru `engineVersion = 3.6.0` atau untuk membuat cluster Amazon DocumentDB 3.6 baru. Untuk klaster Amazon DocumentDB tertentu, Anda dapat menentukan versi cluster menggunakan to `describe-db-clusters` call atau menggunakan AWS CLI konsol manajemen Amazon DocumentDB untuk melihat nomor versi engine untuk klaster tertentu.

Amazon DocumentDB 4.0 `r5` mendukung`r6g`,,`t3.medium`, `t4g.medium` dan jenis instans untuk cluster Anda dan tersedia di semua wilayah yang didukung. Tidak ada biaya tambahan untuk menggunakan Amazon DocumentDB 4.0. Untuk informasi selengkapnya tentang harga, lihat Harga Amazon DocumentDB (dengan kompatibilitas MongoDB).

Meningkatkan atau Memigrasi ke Amazon DocumentDB 4.0

Anda dapat bermigrasi dari MongoDB 3.6 atau MongoDB 4.0 ke Amazon DocumentDB 4.0 memanfaatkan AWS DMS atau utilitas seperti `mongodump`, `mongorestore`, `mongoimport`, dan `mongoexport`. Demikian pula, Anda dapat menggunakan alat yang sama untuk meningkatkan dari Amazon DocumentDB 3.6 ke Amazon DocumentDB 4.0. Untuk petunjuk tentang cara bermigrasi, lihat Memutakhirkan cluster Amazon DocumentDB Anda menggunakan AWS Database Migration Service.

Perbedaan Fungsional

Perbedaan Fungsional Antara Amazon DocumentDB 3.6 dan 4.0

Dengan dirilisnya Amazon DocumentDB 4.0, ada perbedaan fungsional antara Amazon DocumentDB 3.6 dan Amazon DocumentDB 4.0:

- Proyeksi untuk dokumen bersarang: Amazon DocumentDB 3.6 menganggap bidang pertama dalam dokumen bersarang ketika menerapkan proyeksi. Namun, Amazon DocumentDB 4.0 akan mengurai sub dokumen dan juga menerapkan proyeksi untuk setiap sub dokumen. Misalnya: jika proyeksinya adalah `"a.b.c": 1`, maka perilaku di kedua versi itu identik. Namun, jika proyeksinya adalah `{a:{b:{c:1}}}` maka Amazon DocumentDB 3.6 hanya akan menerapkan proyeksi ke 'a' dan bukan 'b' atau 'c'.
- Perilaku untuk **`minKey`**, **`maxKey`**: Di Amazon DocumentDB 4.0, perilaku untuk `{x:{$gt:MaxKey}}` tidak mengembalikan apa-apa, dan untuk `{x:{$lt:MaxKey}}` mengembalikan segalanya.
- Perbedaan perbandingan dokumen: Membandingkan nilai-nilai numerik dari berbagai jenis (ganda, int, panjang) di sub dokumen (misalnya, b dalam `{"_id" :1, "a" :{"b":1}}`) sekarang menyediakan output yang konsisten di seluruh tipe data numerik dan untuk setiap tingkat dokumen.

Perbedaan Fungsional Antara Amazon DocumentDB 4.0 dan MongoDB 4.0

Di bawah ini adalah perbedaan fungsional antara Amazon DocumentDB 4.0 dan MongoDB 4.0.

- Pencarian dengan kunci kosong di jalur: Ketika koleksi berisi dokumen dengan kunci kosong di dalam array (misalnya `{"x" : [ { "" : 10 }, { "b" : 20 } ]}`), dan ketika kunci yang digunakan dalam kueri berakhir dalam string kosong (misalnya `x.`), maka Amazon DocumentDB akan mengembalikan dokumen itu karena melintasi semua dokumen dalam array sedangkan MongoDB tidak akan mengembalikan dokumen itu.

- **`$setOnInsert`** bersama dengan **`$`** di jalur: Operator bidang `$setOnInsert` tidak akan bekerja dalam kombinasi dengan $ di jalur di Amazon DocumentDB, yang juga konsisten dengan MongoDB 4.0.

Transaksi

Amazon DocumentDB (dengan kompatibilitas MongoDB) sekarang mendukung kompatibilitas MongoDB 4.0 termasuk transaksi. Anda dapat melakukan transaksi di beberapa dokumen, pernyataan, koleksi, dan basis data. Transaksi menyederhanakan pengembangan aplikasi dengan memungkinkan Anda untuk melakukan operasi atomik, konsisten, terisolasi, dan tahan lama (ACID) di satu atau lebih dokumen dalam klaster Amazon DocumentDB. Kasus penggunaan umum untuk transaksi termasuk pemrosesan keuangan, pemenuhan dan pengelolaan pesanan, dan membangun permainan multi-pemain.

Tidak ada biaya tambahan untuk transaksi. Anda hanya membayar untuk IO baca dan tulis yang Anda gunakan sebagai bagian dari transaksi.

Topik

- Persyaratan
- Praktik Terbaik
- Keterbatasan:
- Pemantauan dan Diagnosis
- Tingkat Isolasi Transaksi
- Kasus Penggunaan
- Perintah yang Didukung
- Kemampuan yang Tidak Didukung
- Sesi
- Kesalahan Transaksi

Persyaratan

Untuk menggunakan fitur transaksi, Anda perlu memenuhi persyaratan berikut ini:

- Anda harus menggunakan mesin Amazon DocumentDB 4.0.
- Anda harus menggunakan driver yang kompatibel dengan MongoDB 4.0 atau yang lebih baru.

Praktik Terbaik

Berikut ini adalah beberapa praktik terbaik sehingga Anda bisa mendapatkan yang paling baik menggunakan transaksi dengan Amazon DocumentDB.

- Selalu lanjutkan atau batalkan transaksi setelah selesai. Meninggalkan transaksi dalam keadaan tidak lengkap akan mengikat sumber daya basis data dan dapat menyebabkan konflik tulis.
- Disarankan untuk menyimpan transaksi ke jumlah terkecil perintah yang dibutuhkan. Jika Anda memiliki transaksi dengan beberapa pernyataan yang dapat dibagi menjadi beberapa transaksi yang lebih kecil, disarankan untuk melakukannya, untuk mengurangi kemungkinan waktu habis. Selalu bertujuan untuk membuat transaksi pendek, pembacaan yang tidak berjalan lama.

Keterbatasan:

- Amazon DocumentDB tidak mendukung kursor di dalam transaksi.
- Amazon DocumentDB tidak dapat membuat koleksi baru dalam transaksi dan tidak dapat membuat kueri/pembaruan terhadap koleksi yang tidak ada.
- Kunci tulis tingkat dokumen tunduk pada batas waktu 1 menit, yang tidak dikonfigurasi oleh pengguna.
- Penulisan yang dapat dicoba ulang, komit yang dapat dicoba ulang, dan perintah pembatalan yang dapat dicoba ulang tidak didukung di Amazon DocumentDB. Pengecualian: Jika Anda menggunakan shell mongo, jangan sertakan`retryWrites=false` perintah dalam string kode apa pun. Secara default, penulisan retryable dinonaktifkan. Termasuk`retryWrites=false` dapat menyebabkan kegagalan dalam perintah baca normal.
- Setiap instans Amazon DocumentDB memiliki batas batas atas pada jumlah transaksi bersamaan yang dibuka pada instans pada satu waktu. Untuk batas, silakan lihat Batas Instans.
- Untuk transaksi tertentu, ukuran log transaksi harus kurang dari 32MB.
- Amazon DocumentDB tidak mendukung `count()` di dalam transaksi, tetapi tidak semua driver mendukung kemampuan ini. Alternatifnya adalah dengan menggunakan `countDocuments()` API, yang menerjemahkan kueri hitungan ke kueri agregasi pada sisi klien.
- Transaksi memiliki batas eksekusi satu menit dan sesi memiliki batas waktu 30 menit. Jika waktu transaksi habis, maka akan dibatalkan, dan perintah berikutnya dikeluarkan di dalam sesi untuk transaksi yang ada yang akan menghasilkan kesalahan berikut ini:

```
WriteCommandError({
```

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

Pemantauan dan Diagnosis

Dengan dukungan untuk transaksi di Amazon DocumentDB 4.0, CloudWatch metrik tambahan ditambahkan untuk membantu Anda memantau transaksi Anda.

CloudWatch Metrik Baru

- `DatabaseTransactions`: Jumlah transaksi terbuka yang diambil pada periode satu menit.
- `DatabaseTransactionsAborted`: Jumlah transaksi yang dibatalkan yang diambil pada periode satu menit.
- `DatabaseTransactionsMax`: Jumlah maksimum transaksi terbuka dalam periode satu menit.
- `TransactionsAborted`: Jumlah transaksi yang dibatalkan pada instans dalam periode satu menit.
- `TransactionsCommitted`: Jumlah transaksi yang dilakukan pada instans dalam periode satu menit.
- `TransactionsOpen`: Jumlah transaksi terbuka pada instans yang diambil pada periode satu menit.
- `TransactionsOpenMax`: Jumlah maksimum transaksi terbuka pada instans dalam periode satu menit.
- `TransactionsStarted`: Jumlah transaksi yang dimulai pada instans dalam periode satu menit.

> **Note**
>
> Untuk CloudWatch metrik lainnya untuk Amazon DocumentDB, bukaMemantau Amazon DocumentDB dengan CloudWatch.

Selain itu, bidang baru ditambahkan ke `currentOp` `lsid`, `transactionThreadId`, dan keadaan baru untuk transaksi "`idle transaction`" dan `serverStatus`: `currentActive`, `currentInactive`, `currentOpen`, `totalAborted`, `totalCommitted`, dan `totalStarted`.

Tingkat Isolasi Transaksi

Ketika memulai transaksi, Anda memiliki kemampuan untuk menentukan `readConcern` dan `writeConcern` sebagaimana ditunjukkan dalam contoh di bawah ini:

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

Untuk `readConcern`, Amazon DocumentDB mendukung isolasi snapshot secara default. Jika `readConcern` lokal, tersedia, atau mayoritas ditentukan, Amazon DocumentDB akan meng-upgrade tingkat `readConcern` untuk snapshot. Amazon DocumentDB tidak mendukung `readConcern` yang dapat dilinierkan dan menentukan kekhawatiran baca semacam itu akan mengakibatkan kesalahan.

Untuk `writeConcern`, Amazon DocumentDB mendukung mayoritas secara default dan kuorum tulis dicapai ketika empat salinan data bertahan di tiga AZ. Jika `writeConcern` lebih rendah ditentukan, Amazon DocumentDB akan meng-upgrade `writeConcern` untuk mayoritas. Selanjutnya, semua penulisan Amazon DocumentDB dijurnalkan dan penjurnalan tidak dapat dinonaktifkan.

Kasus Penggunaan

Dalam bagian ini, kita akan berjalan melalui dua kasus penggunaan untuk transaksi: multi-pernyataan dan multi-koleksi.

Transaksi Multi-Pernyataan

Transaksi Amazon DocumentDB adalah multi-pernyataan, yang berarti Anda dapat menulis transaksi yang mencakup beberapa pernyataan dengan penyimpanan atau rollback eksplisit. Anda dapat mengelompokkan tindakan `insert`, `update`, `delete`, dan `findAndModify` sebagai operasi atomik tunggal.

Kasus penggunaan umum untuk transaksi multi-pernyataan adalah transaksi debet-kredit. Sebagai contoh: Anda berutang kepada uang teman untuk membeli pakaian. Dengan demikian, Anda perlu mendebet (menarik) $500 dari akun Anda dan kredit $500 (deposit) ke akun teman Anda. Untuk melakukan operasi itu, Anda melakukan operasi debet dan kredit di dalam transaksi tunggal untuk memastikan atomisitas. Melakukan hal ini akan mencegah skenario di mana $500 didebet dari akun Anda, tetapi tidak dikreditkan ke akun teman Anda. Seperti inilah kasus penggunaan ini akan terlihat:

```
// *** 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();
```

Transaksi Multi-Koleksi

Transaksi kami juga bersifat multi-koleksi, yang berarti teransaksi tersebut dapat digunakan untuk melakukan beberapa operasi di dalam satu transaksi dan di beberapa koleksi. Hal ini menyediakan tampilan data yang konsisten dan menjaga integritas data Anda. Ketika Anda melakukan perintah sebagai tunggal<>, transaksi adalah all-or-nothing eksekusi — dalam hal itu, transaksi semuanya berhasil maupun semuanya gagal.

Berikut adalah contoh transaksi multi-koleksi, menggunakan skenario dan data yang sama dari contoh untuk transaksi multi-pernyataan.

```
// *** 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
```

Contoh API Transaksi untuk API Panggilan Balik

API panggilan balik hanya tersedia untuk driver 4.2+.

Javascript

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Javascript.

```
// *** 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Node.js.

```
// 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#

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan C#.

```
// 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB API dengan Ruby.

```
// 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

Kode berikut menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Go.

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

ctx := context.TODO()

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

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

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

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

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

 SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
_, err = session.WithTransaction(ctx, func(sessionCtx mongo.SessionContext)
 (interface{}, error) {
    var result Account
    // deduct $500 from Alice's account
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
```

```
    _, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Alice"},
 bson.M{"$set": bson.M{"balance": newAliceBalance}})
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance = result.Balance
    assert.Equal(t, aliceBalance, newAliceBalance)

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

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

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

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

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

 SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
```

```
    if err := sessionContext.StartTransaction(transactionOptions); err != nil {
        panic(err)
    }

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

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

    err = sessionContext.CommitTransaction(sessionContext)
    return err
}

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

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

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

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

Java

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Java.

```
// 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan C.

```
// 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Python.

```
// Sample Python code with callback api

import pymongo

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

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

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

```
amount_to_transfer = 500

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

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

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

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

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

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

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

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

amount_to_transfer = 500

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

```
new_alice_balance = alice_balance - amount_to_transfer

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

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

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

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

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

Contoh API Transaksi untuk API Inti

Javascript

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Javascript.

```
// *** 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#

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan C#.

```
// 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB API dengan Ruby.

```
# Ruby Core API

def transfer_money_w_retry(session, accountColl)
    amountToTransfer = 500

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

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

    session.commit_transaction

end
```

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

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

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

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

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

end
```

Java

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Java.

```
// 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan C.

```
// 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

Kode berikut ini menunjukkan cara untuk memanfaatkan API transaksi Amazon DocumentDB dengan Scala.

```
// 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()

}
```

Perintah yang Didukung

Perintah	Didukung
abortTransaction	Ya
commitTransaction	Ya
endSessions	Ya
killSession	Ya
killAllSession	Ya
killAllSessionsByPattern	Tidak
refreshSessions	Tidak
startSession	Ya

Kemampuan yang Tidak Didukung

Metode	Tahapan atau Perintah
db.collection.aggregate()	$collStats
	$currentOp
	$indexStats
	$listSessions
	$out

Metode	Tahapan atau Perintah

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

`db.collection.insert()`	`insert` tidak didukung jika tidak dijalankan terhadap koleksi yang ada. Metode ini didukung jika menargetkan koleksi yang sudah ada sebelumnya.

Sesi

Sesi MongoDB adalah kerangka kerja yang digunakan untuk mendukung operasi penulisan yang dapat dicoba lagi, konsistensi kausal, transaksi, dan pengelolaan di seluruh basis data. Ketika sesi dibuat, pengidentifikasi sesi logis (lsid) dihasilkan oleh klien dan digunakan untuk menandai semua operasi di dalam sesi tersebut ketika mengirim perintah ke server.

Amazon DocumentDB mendukung penggunaan sesi untuk mengaktifkan transaksi, tetapi tidak mendukung konsistensi kausal atau penulisan yang dapat dicoba lagi.

Ketika memanfaatkan transaksi di dalam Amazon DocumentDB, transaksi akan dimulai dari dalam sesi menggunakan `session.startTransaction()` API dan sesi mendukung transaksi tunggal pada satu waktu. Serupa dengan itu, transaksi diselesaikan menggunakan penyimpanan (`session.commitTransaction()`) maupun pembatalan (`session.abortTransaction()`) API.

Konsistensi kausal

Konsistensi kausal menjamin bahwa di dalam sesi klien tunggal, klien akan mengamati read-after-write konsistensi, baca/tulis monatomik, baca/tulis monatomik, dan penulisan akan mengikuti pembacaan dan penulisan monatomik, bukan hanya di primer. Amazon DocumentDB tidak mendukung konsistensi kausal dan pernyataan berikut ini akan mengakibatkan kesalahan.

```
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()
```

Anda dapat menonaktifkan konsistensi kausal di dalam sesi. Harap dicatat, melakukannya akan memungkinkan Anda untuk memanfaatkan kerangka kerja sesi, tetapi tidak akan memberikan jaminan konsistensi kausal untuk pembacaan. Ketika menggunakan Amazon DocumentDB, pembacaan dari primer akan read-after-write konsisten dan pembacaan dari instans replika pada akhirnya akan konsisten. Transaksi adalah kasus penggunaan primer untuk memanfaatkan sesi.

```
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 }
```

Penulisan yang dapat dicoba lagi

Penulisan yang dapat dicoba lagi adalah kemampuan di mana klien akan berupaya untuk mencoba lagi menulis operasi, satu kali, ketika kesalahan jaringan terjadi atau jika klien tidak dapat menemukan primer. Di Amazon DocumentDB, penulisan yang dapat dicoba lagi tidak didukung dan harus dinonaktifkan. Anda dapat menonaktifkannya dengan perintah (`retryWrites=false`) dalam string koneksi.

Pengecualian: Jika Anda menggunakan shell mongo, jangan sertakan`retryWrites=false` perintah dalam string kode apa pun. Secara default, penulisan retryable dinonaktifkan. Termasuk`retryWrites=false` dapat menyebabkan kegagalan dalam perintah baca normal.

Kesalahan Transaksi

Ketika menggunakan transaksi, terdapat skenario yang dapat mengakibatkan kesalahan yang menyatakan bahwa nomor transaksi tidak cocok dengan apa pun dalam transaksi yang sedang berlangsung.

Kesalahan dapat dihasilkan dalam setidaknya dua skenario yang berbeda:

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

Cara terbaik untuk menangani kesalahan ini adalah untuk membuat idempotensi pembaruan transaksional -- sebagai contoh, dengan menggunakan mutator `$set` alih-alih operasi kenaikan/penurunan. Lihat di bawah ini:

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

Praktik Terbaik untuk Amazon DocumentDB

Pelajari praktik terbaik untuk bekerja dengan Amazon DocumentDB (dengan kompatibilitas MongoDB). Bagian ini terus diperbarui saat praktik terbaik baru diidentifikasi.

Topik

- Pedoman Operasional Dasar
- Pengukuran Instans
- Bekerja dengan Indeks
- Praktik Terbaik Keamanan
- Pengoptimalan Biaya
- Menggunakan Metrik untuk Mengidentifikasi Masalah Kinerja
- Beban Kerja TTL dan Deret Waktu
- Migrasi
- Bekerja dengan Grup Parameter Klaster
- Kueri Alur Agregasi
- batchInsert dan batchUpdate

Pedoman Operasional Dasar

Berikut ini adalah panduan operasional dasar yang harus diikuti setiap orang saat bekerja dengan Amazon DocumentDB. Perjanjian Tingkat Layanan Amazon DocumentDB mewajibkan Anda untuk mengikuti panduan berikut ini.

- Menerapkan cluster yang terdiri dari dua atau lebih instans AWS Amazon DocumentDB di dua Availability Zone. Untuk beban kerja produksi, kami merekomendasikan men-deploy klaster yang terdiri dari tiga atau lebih instans Amazon DocumentDB di tiga Availability Zone.
- Gunakan layanan dalam kuota layanan yang dinyatakan. Untuk informasi selengkapnya, lihat Kuota dan Batas Amazon DocumentDB.
- Memantau memori, CPU, koneksi, dan penggunaan penyimpanan Anda. Untuk membantu Anda mempertahankan performa dan ketersediaan sistem, siapkan Amazon CloudWatch untuk memberi tahu Anda saat pola penggunaan berubah atau saat Anda mendekati kapasitas penerapan.

- Menaikkan skala instans Anda saat Anda mendekati batas kapasitas. Instans Anda harus disediakan dengan sumber daya komputasi yang cukup (yaitu, RAM, CPU) untuk mengakomodasi peningkatan permintaan yang tidak terduga dari aplikasi Anda.
- Atur periode retensi cadangan agar selaras dengan tujuan titik pemulihan Anda.
- Tes failover untuk klaster Anda untuk memahami berapa lama proses untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Failover Amazon DocumentDB.
- Hubungkan ke klaster Amazon DocumentDB Anda dengan titik akhir klaster (lihat Titik Akhir Amazon DocumentDB) dan dalam mode set replika (lihat Menghubungkan ke Amazon DocumentDB sebagai Set Replika) untuk meminimalkan dampak failover pada aplikasi Anda.
- Pilih pengaturan preferensi baca driver yang memaksimalkan penskalaan baca sekaligus memenuhi persyaratan konsistensi baca aplikasi Anda. Preferensi baca `secondaryPreferred` mengaktifkan pembacaan replika dan membebaskan instans primer untuk melakukan lebih banyak pekerjaan. Untuk informasi selengkapnya, lihat Opsi Preferensi Baca.
- Rancang aplikasi Anda agar menjadi tangguh jika terjadi peristiwa kesalahan jaringan dan basis data. Gunakan mekanisme kesalahan driver Anda untuk membedakan antara kesalahan sementara dan kesalahan terus-menerus. Coba lagi kesalahan sementara menggunakan mekanisme backoff eksponensial bila sesuai. Pastikan bahwa aplikasi Anda mempertimbangkan konsistensi data ketika menerapkan logika coba lagi.
- Aktifkan perlindungan penghapusan klaster untuk semua klaster produksi, atau klaster mana pun yang memiliki data berharga. Sebelum menghapus klaster Amazon DocumentDB, ambil snapshot akhir. Jika Anda menggunakan sumber daya dengan AWS CloudFormation, aktifkan perlindungan penghentian. Untuk informasi selengkapnya, lihat Perlindungan Penghentian dan Perlindungan Penghapusan.
- Ketika membuat klaster Amazon DocumentDB, --engine-version adalah parameter opsional yang ditetapkan secara default ke versi mesin utama terbaru. Versi mesin utama saat ini adalah 4.0.0. Ketika versi mesin utama baru dirilis, versi mesin default untuk --engine-version akan diperbarui untuk mencerminkan versi mesin utama yang terakhir. Akibatnya, untuk beban kerja produksi, dan terutama yang bergantung pada skrip, otomatisasi, atau AWS CloudFormation templat, kami menyarankan Anda secara eksplisit menentukan --engine-version ke versi utama yang dimaksud.

Pengukuran Instans

Salah satu aspek terpenting dalam memilih ukuran instans di Amazon DocumentDB adalah jumlah RAM untuk cache Anda. Amazon DocumentDB mencadangkan sepertiga RAM untuk layanannya

sendiri, artinya hanya dua pertiga dari RAM instans yang tersedia untuk cache. Dengan demikian, itu adalah praktik terbaik Amazon DocumentDB untuk memilih tipe instans dengan RAM yang cukup agar sesuai dengan set kerja Anda (yaitu, data dan indeks) dalam memori. Memiliki instans berukuran tepat akan membantu mengoptimalkan performa secara keseluruhan dan berpotensi meminimalkan biaya I/O. Anda dapat menggunakan kalkulator ukuran Amazon DocumentDB pihak ketiga untuk memperkirakan ukuran instans untuk beban kerja tertentu.

Untuk menentukan apakah set kerja aplikasi Anda cocok dengan memori, pantau `BufferCacheHitRatio` penggunaan Amazon CloudWatch untuk setiap instance dalam klaster yang sedang dimuat.

`BufferCacheHitRatio` CloudWatch Metrik mengukur persentase data dan indeks yang disajikan dari cache memori instance (versus volume penyimpanan). Secara umum, nilai `BufferCacheHitRatio` harus setinggi mungkin, karena membaca data dari memori set kerja lebih cepat dan lebih hemat biaya daripada membaca dari volume penyimpanan. Selagi itu diinginkan untuk menjaga `BufferCacheHitRatio` sedekat mungkin dengan 100%, nilai terbaik yang dapat dicapai akan tergantung pada pola akses dan persyaratan performa aplikasi Anda. Untuk mempertahankan `BufferCacheHitRatio` setinggi mungkin, dianjurkan agar instans di klaster Anda disediakan dengan RAM yang cukup untuk dapat menyesuaikan indeks dan set data kerja dalam memori.

Jika indeks Anda tidak sesuai dengan memori, Anda akan melihat `BufferCacheHitRatio` yang lebih rendah. Membaca terus-menerus dari disk menimbulkan biaya I/O tambahan dan tidak seperforma seperti membaca dari memori. Jika rasio `BufferCacheHitRatio` Anda lebih rendah dari yang diharapkan, tingkatkan ukuran instans untuk klaster Anda guna menyediakan lebih banyak RAM untuk menyesuaikan data set kerja dalam memori. Jika menskalakan ke atas kelas instans menghasilkan peningkatan dramatis dalam `BufferCacheHitRatio`, maka set kerja aplikasi anda tidak sesuai di memori. Lanjutkan untuk menaikkan skala sampai `BufferCacheHitRatio` tidak lagi meningkat secara dramatis setelah operasi penskalaan. Untuk informasi tentang pemantauan metrik instans, lihat Metrik Amazon DocumentDB.

Tergantung pada beban kerja dan persyaratan latensi Anda, itu mungkin dapat diterima untuk aplikasi Anda untuk memiliki nilai `BufferCacheHitRatio` yang lebih tinggi selama penggunaan kondisi stabil, tetapi memiliki penurunan `BufferCacheHitRatio` secara berkala arena kueri analitik yang perlu memindai seluruh koleksi dijalankan pada instans. Penurunan periodik dalam `BufferCacheHitRatio` dapat bermanifestasi sebagai latensi yang lebih tinggi untuk kueri berikutnya yang perlu untuk mengisi ulang data set kerja dari volume penyimpanan kembali ke cache buffer. Kami menyarankan Anda menguji beban kerja Anda di lingkungan pra-produksi dengan

beban kerja produksi yang representatif terlebih dahulu untuk memahami karakteristik kinerja dan **`BufferCacheHitRatio`** sebelum menerapkan beban kerja ke produksi.

`BufferCacheHitRatio` adalah metrik khusus instans, jadi instans yang berbeda dalam klaster yang sama mungkin memiliki nilai-nilai `BufferCacheHitRatio` yang berbeda tergantung pada cara pembacaan didistribusikan di antara instans primer dan replika. Jika beban kerja operasional Anda tidak dapat menangani peningkatan periodik latensi dari mengisi ulang cache set kerja setelah menjalankan kueri analitik, Anda harus mencoba untuk mengisolasi cache buffer beban kerja reguler dari kueri analitik. Anda dapat mencapai isolasi `BufferCacheHitRatio` lengkap dengan mengarahkan kueri operasional ke instans primer dan kueri analitik hanya ke instans replika. Anda juga dapat mencapai isolasi parsial dengan mengarahkan kueri analitik ke instans replika tertentu dengan pemahaman bahwa beberapa persentase kueri reguler juga akan berjalan pada replika tersebut dan berpotensi terpengaruh.

Nilai `BufferCacheHitRatio` yang sesuai tergantung pada kasus penggunaan dan persyaratan aplikasi Anda. Tidak ada satu nilai terbaik atau minimum untuk metrik ini; hanya Anda yang dapat memutuskan apakah tradeoff dari `BufferCacheHitRatio` yang sementara lebih rendah dapat diterima dari perspektif biaya dan performa.

Bekerja dengan Indeks

Membangun Indeks

Ketika mengimpor data ke Amazon DocumentDB, Anda harus membuat indeks Anda sebelum mengimpor set data besar. Anda dapat menggunakan Alat Indeks Amazon DocumentDB untuk mengekstrak indeks dari instans MongoDB atau direktori `mongodump` yang sedang berjalan, dan membuat indeks tersebut di klaster Amazon DocumentDB. Untuk panduan lebih lanjut tentang migrasi, lihat Migrasi ke Amazon DocumentDB.

Selektivitas Indeks

Kami merekomendasikan Anda membatasi pembuatan indeks ke bidang di mana jumlah nilai duplikat kurang dari 1% dari jumlah total dokumen dalam koleksi. Sebagai contoh, jika pengumpulan Anda berisi 100.000 dokumen, buat indeks hanya pada bidang tempat nilai yang sama muncul 1000 kali atau kurang.

Memilih indeks dengan jumlah nilai unik yang tinggi (yaitu, kardinalitas tinggi) memastikan bahwa operasi filter mengembalikan sejumlah kecil dokumen, sehingga menghasilkan performa yang baik

selama pemindaian indeks. Contoh indeks kardinalitas tinggi adalah indeks unik, yang menjamin bahwa predikat kesetaraan mengembalikan paling banyak satu dokumen. Contoh kardinalitas rendah termasuk indeks di atas bidang Boolean dan indeks di atas hari dalam seminggu. Karena performanya yang buruk, indeks kardinalitas rendah tidak mungkin dipilih oleh pengoptimal kueri basis data. Pada saat yang sama, indeks kardinalitas rendah terus mengonsumsi sumber daya seperti ruang disk dan I/O. Sebagai aturan praktis, Anda harus menargetkan indeks pada bidang di mana frekuensi nilai khas adalah 1% dari total ukuran koleksi atau kurang.

Selain itu, direkomendasikan untuk hanya membuat indeks pada bidang yang umumnya digunakan sebagai filter dan secara teratur mencari indeks yang tidak terpakai. Untuk informasi selengkapnya, lihat Bagaimana cara menganalisis penggunaan indeks dan mengidentifikasi indeks yang tidak digunakan?.

Dampak Indeks pada Penulisan Data

Meskipun indeks dapat meningkatkan performa kueri dengan menghindari kebutuhan untuk memindai setiap dokumen dalam koleksi, peningkatan ini disertai dengan tradeoff. Untuk setiap indeks pada koleksi, setiap kali dokumen dimasukkan, diperbarui, atau dihapus, basis data harus memperbarui koleksi dan menulis bidang untuk masing-masing indeks untuk koleksi. Sebagai contoh, jika koleksi memiliki sembilan indeks, basis data harus melakukan sepuluh penulisan sebelum mengakui operasi ke klien. Dengan demikian, setiap indeks tambahan menimbulkan latensi tulis tambahan, I/O, dan peningkatan penyimpanan yang digunakan secara keseluruhan.

Instans klaster harus berukuran tepat untuk menjaga semua memori set kerja. Ini menghindari kebutuhan untuk terus membaca halaman indeks dari volume penyimpanan, yang berdampak negatif pada performa dan menghasilkan biaya I/O yang lebih tinggi. Untuk informasi selengkapnya, lihat Pengukuran Instans.

Untuk performa terbaik, meminimalkan jumlah indeks dalam koleksi Anda, tambahkan hanya indeks yang diperlukan untuk meningkatkan performa untuk kueri umum. Meskipun beban kerja bervariasi, pedoman yang baik adalah untuk menjaga jumlah indeks per koleksi menjadi lima atau lebih sedikit.

Mengidentifikasi Indeks yang Hilang

Mengidentifikasi indeks yang hilang adalah praktik terbaik yang kami rekomendasikan untuk dilakukan secara teratur. Untuk informasi selengkapnya, lihat Bagaimana cara mengidentifikasi indeks yang hilang?.

Mengidentifikasi Indeks yang Tidak Terpakai

Mengidentifikasi dan menghapus indeks yang tidak terpakai adalah praktik terbaik yang kami rekomendasikan untuk dilakukan secara teratur. Untuk informasi selengkapnya, lihat Bagaimana cara menganalisis penggunaan indeks dan mengidentifikasi indeks yang tidak digunakan?.

Praktik Terbaik Keamanan

Untuk praktik terbaik keamanan, Anda harus menggunakan akun AWS Identity and Access Management (IAM) untuk mengontrol akses ke operasi Amazon DocumentDB API, terutama operasi yang membuat, memodifikasi, atau menghapus sumber daya Amazon DocumentDB. Sumber daya tersebut termasuk klaster, grup keamanan, dan grup parameter. Anda juga harus menggunakan IAM untuk mengontrol tindakan yang melakukan tindakan administrasi umum seperti mencadangkan pemulihan klaster. Saat membuat IAM role, terapkan prinsip hak istimewa terendah.

- Terapkan hak istimewa terendah dengan kontrol akses berbasis peran.
- Tetapkan akun IAM individu untuk setiap orang yang mengelola sumber daya Amazon DocumentDB. Jangan gunakan pengguna Akun AWS root untuk mengelola sumber daya Amazon DocumentDB. Buat pengguna IAM untuk semua orang, termasuk Anda sendiri.
- Berikan setiap pengguna IAM set izin minimum yang diperlukan untuk melakukan tugas mereka.
- Gunakan grup IAM untuk mengelola izin secara efektif bagi beberapa pengguna. Untuk informasi lebih lanjut tentang IAM, lihat Panduan Pengguna IAM. Untuk informasi tentang praktik terbaik IAM, lihat Praktik Terbaik IAM.
- Secara rutin putar kredensial IAM Anda.
- Konfigurasikan AWS Secrets Manager untuk secara otomatis memutar rahasia untuk Amazon DocumentDB. Untuk informasi selengkapnya, lihat Memutar AWS Rahasia Secrets Manager Anda dan Rotating Secrets untuk Amazon DocumentDB di Panduan Pengguna AWS Secrets Manager.
- Berikan setiap pengguna Amazon DocumentDB set izin minimum yang diperlukan untuk melakukan tugas mereka. Untuk informasi selengkapnya, lihat Akses Database Menggunakan Kontrol Akses Berbasis Peran.
- Gunakan Transport Layer Security (TLS) untuk mengenkripsi data Anda dalam perjalanan dan AWS KMS mengenkripsi data Anda saat istirahat.

Pengoptimalan Biaya

Praktik terbaik berikut dapat membantu Anda mengelola dan meminimalkan biaya Anda ketika menggunakan Amazon DocumentDB. Untuk informasi harga, lihat Harga Amazon DocumentDB (dengan kompatibilitas MongoDB) dan FAQ Amazon DocumentDB (dengan kompatibilitas MongoDB).

- Buat pemberitahuan penagihan pada ambang batas 50 persen dan 75 persen dari tagihan yang Anda harapkan untuk bulan tersebut. Untuk informasi selengkapnya tentang membuat pemberitahuan penagihan, lihat Membuat Alarm Penagihan.
- Arsitektur Amazon DocumentDB memisahkan penyimpanan dan komputasi, sehingga klaster instans tunggal pun sangat berdaya tahan. Volume penyimpanan klaster mereplikasi data enam cara di tiga Availability Zone, menyediakan daya tahan yang sangat tinggi terlepas dari jumlah instans dalam klaster. Klaster produksi khas memiliki tiga atau lebih instans untuk menyediakan ketersediaan tinggi. Namun, Anda dapat mengoptimalkan biaya dengan menggunakan klaster pengembangan instans tunggal ketika ketersediaan tinggi tidak diperlukan.
- Untuk skenario pengembangan dan pengujian, hentikan klaster ketika tidak lagi diperlukan dan mulai klaster ketika pengembangan dilanjutkan. Untuk informasi selengkapnya, lihat Menghentikan dan memulai cluster Amazon DocumentDB.
- Baik TTL dan aliran perubahan dikenakan I/O saat data ditulis, dibaca, dan dihapus. Jika Anda telah mengaktifkan fitur ini tetapi tidak memanfaatkannya dalam aplikasi Anda, menonaktifkan fitur tersebut dapat membantu mengurangi biaya.

Menggunakan Metrik untuk Mengidentifikasi Masalah Kinerja

Untuk mengidentifikasi masalah performa yang disebabkan sumber daya yang tidak mencukupi dan kemacetan umum lainnya, Anda dapat memantau metrik yang tersedia untuk klaster Amazon DocumentDB.

Melihat Metrik Kinerja

Pantau metrik kinerja secara rutin untuk melihat nilai rata-rata, maksimum, dan minimum untuk berbagai rentang waktu. Hal ini membantu Anda mengidentifikasi saat performa menurun. Anda juga dapat mengatur CloudWatch alarm Amazon untuk ambang metrik tertentu sehingga Anda diberi tahu jika mereka tercapai.

Untuk memecahkan masalah kinerja, penting untuk memahami kinerja dasar sistem. Setelah Anda menyiapkan klaster baru dan menjalankannya dengan beban kerja tipikal, tangkap nilai rata-rata, maksimum, dan minimum dari semua metrik kinerja pada interval yang berbeda (misalnya, 1 jam, 24 jam, 1 minggu, 2 minggu). Ini memberi Anda gambaran tentang apa yang normal. Sehingga membantu mendapatkan perbandingan untuk jam sibuk dan tidak sibuk. Kemudian, Anda dapat menggunakan informasi ini untuk mengidentifikasi saat performa turun di bawah tingkat standar.

Anda dapat melihat metrik kinerja menggunakan AWS Management Console atau AWS CLI. Untuk informasi selengkapnya, lihat Melihat CloudWatch Data.

Mengatur CloudWatch Alarm

Untuk menyetel CloudWatch alarm, lihat Menggunakan CloudWatch Alarm Amazon di Panduan CloudWatch Pengguna Amazon.

Mengevaluasi Metrik Kinerja

Instans memiliki beberapa kategori metrik yang berbeda. Cara Anda menentukan nilai yang dapat diterima tergantung pada metrik.

CPU

- Pemanfaatan CPU — Persentase kapasitas pemrosesan komputer yang digunakan.

Memori

- Memori yang Bisa Dibebaskan — Berapa banyak RAM yang tersedia pada instans.
- Penggunaan Swap — Berapa banyak ruang tukar yang digunakan oleh instans, dalam megabyte.

Operasi input/output

- IOPS Baca, IOPS Tulis — Jumlah rata-rata operasi baca atau tulis disk per detik.
- Latensi Baca, Latensi Tulis — Waktu rata-rata untuk operasi baca atau tulis dalam milidetik.
- Throughput Baca, Throughput Tulis — Rata-rata jumlah megabyte yang dibaca dari atau ditulis ke disk per detik.
- Kedalaman Antrean Disk — Jumlah operasi I/O yang menunggu untuk ditulis ke atau dibaca dari disk.

Lalu lintas jaringan

- Throughput Penerimaan Jaringan, Throughput Pengiriman Jaringan — Tingkat lalu lintas jaringan ke dan dari instans dalam megabyte per detik.

Koneksi basis data

- Koneksi DB — Jumlah sesi klien yang terhubung ke instans.

Secara umum, nilai yang dapat diterima untuk metrik performa bergantung pada seperti apa garis dasar Anda dan apa yang dilakukan aplikasi Anda. Periksa varian yang konsisten atau sedang tren dari garis dasar Anda.

Berikut ini adalah rekomendasi dan saran tentang jenis metrik tertentu:

- Konsumsi CPU tinggi — Nilai tinggi untuk konsumsi CPU mungkin sesuai, jika mereka menjaga tujuan untuk aplikasi Anda (seperti throughput atau konkurensi) dan diharapkan. Jika konsumsi CPU Anda secara konsisten lebih dari 80 persen, pertimbangkan untuk menskalakan ke atas instans Anda.
- Konsumsi RAM tinggi — Jika `FreeableMemory` metrik Anda sering turun di bawah 10% dari total memori instans, pertimbangkan untuk meningkatkan instans Anda. Untuk informasi selengkapnya tentang apa yang terjadi ketika instans DocumentDB Anda mengalami tekanan memori tinggi, lihat Tata Kelola Sumber Daya Amazon DocumentDB.
- Penggunaan swap — Metrik ini harus tetap pada atau mendekati nol. Jika penggunaan swap Anda signifikan, pertimbangkan untuk menskalakan ke atas instans Anda.
- Lalu lintas jaringan — Untuk lalu lintas jaringan, bicarakan dengan administrator sistem Anda untuk memahami throughput yang diharapkan untuk jaringan domain dan koneksi internet Anda. Selidiki lalu lintas jaringan jika throughput selalu di bawah yang diharapkan.
- Koneksi basis data — Pertimbangkan untuk membatasi koneksi basis data jika Anda melihat jumlah koneksi pengguna yang tinggi bersamaan dengan penurunan performa instans dan waktu respons. Jumlah koneksi pengguna terbaik untuk instans Anda bervariasi berdasarkan kelas instans Anda dan kerumitan operasi yang sedang dilakukan. Untuk masalah dengan metrik kinerja apa pun, salah satu hal pertama yang dapat Anda lakukan untuk meningkatkan perfroma adalah menyesuaikan kueri yang paling banyak digunakan dan paling mahal untuk melihat apakah hal tersebut menurunkan tekanan pada sumber daya sistem.

Jika kueri Anda disetel dan masalah tetap ada, pertimbangkan untuk meningkatkan kelas instans Amazon DocumentDB Anda ke yang memiliki lebih banyak sumber daya (CPU, RAM, ruang disk, bandwidth jaringan, kapasitas I/O) yang terkait dengan masalah yang Anda alami.

Penyetelan Kueri

Salah satu cara terbaik untuk meningkatkan kinerja klaster adalah dengan menyetel kueri yang paling sering digunakan dan paling intensif sumber daya untuk membuatnya lebih murah untuk dijalankan.

Anda dapat menggunakan profiler (lihat Membuat Profil Operasi Amazon DocumentDB) untuk mencatat waktu eksekusi dan detail operasi yang dilakukan di klaster Anda. Profiler berguna untuk memantau operasi paling lambat di klaster Anda untuk membantu Anda meningkatkan performa kueri individual dan performa klaster secara keseluruhan.

Anda juga dapat menggunakan perintah `explain` untuk mempelajari cara menganalisis rencana kueri untuk kueri tertentu. Gunakan informasi ini untuk memodifikasi kueri atau koleksi yang mendasari untuk meningkatkan kinerja kueri Anda (misalnya, menambahkan indeks).

Beban Kerja TTL dan Deret Waktu

Penghapusan dokumen yang dihasilkan dari kedaluwarsanya indeks TTL merupakan proses upaya terbaik. Dokumen tidak dijamin akan dihapus dalam periode tertentu. Faktor-faktor seperti ukuran instans, pemanfaatan sumber daya instans, ukuran dokumen, throughput keseluruhan, jumlah indeks, dan apakah indeks dan set kerja sesuai dalam memori, semuanya dapat memengaruhi waktu ketika dokumen yang kedaluwarsa dihapus oleh proses TTL.

Saat monitor TTL menghapus dokumen Anda, setiap penghapusan dikenakan biaya IO, yang menambah tagihan Anda. Jika throughput dan tingkat penghapusan TTL meningkat, Anda sebaiknya mengharapkan tagihan yang lebih tinggi karena peningkatan penggunaan I/O. Namun, jika Anda tidak membuat indeks TTL untuk menghapus dokumen, melainkan mengelompokkan dokumen ke dalam koleksi berdasarkan waktu, dan cukup membuang koleksi tersebut saat tidak lagi diperlukan, Anda tidak akan dikenakan biaya IO apa pun. Ini dapat secara signifikan menghemat biaya daripada menggunakan indeks TTL.

Untuk beban kerja deret waktu, Anda dapat mempertimbangkan untuk membuat koleksi bergulir alih-alih indeks TTL karena koleksi bergulir dapat menjadi cara yang lebih berperforma untuk menghapus data dan lebih sedikit I/O yang intensif. Jika Anda memiliki koleksi besar (terutama koleksi lebih dari 1TB) atau biaya I/O penghapusan TTL yang menjadi perhatian, kami rekomendasikan Anda

mempartisi dokumen ke dalam koleksi berdasarkan waktu, dan membuang koleksi ketika dokumen tidak lagi diperlukan. Anda dapat membuat satu koleksi per hari atau satu per minggu, tergantung pada tingkat menyerap data Anda. Meskipun persyaratan akan bervariasi tergantung pada aplikasi Anda, aturan praktis yang baik adalah memiliki lebih banyak koleksi yang lebih kecil daripada beberapa koleksi besar. Menjatuhkan koleksi ini tidak dikenakan biaya I/O, dan dapat lebih cepat dan lebih hemat biaya daripada menggunakan indeks TTL.

Migrasi

Sebagai praktik terbaik, kami merekomendasikan bahwa ketika memigrasikan data ke Amazon DocumentDB, Anda terlebih dahulu membuat indeks Anda di Amazon DocumentDB sebelum memigrasikan data. Membuat indeks terlebih dahulu dapat mengurangi waktu keseluruhan dan meningkatkan kecepatan migrasi. Untuk melakukannya, Anda dapat menggunakan Alat Indeks Amazon DocumentDB. Untuk informasi lebih lanjut tentang migrasi, lihat Panduan migrasi Amazon DocumentDB.

Kami juga merekomendasikan bahwa sebelum Anda memigrasikan basis data produksi Anda, praktik terbaik adalah untuk sepenuhnya menguji aplikasi Anda di Amazon DocumentDB, dengan mempertimbangkan fungsionalitas, kinerja, operasi, dan biaya.

Bekerja dengan Grup Parameter Klaster

Kami menyarankan agar Anda mencoba perubahan grup parameter klaster pada klaster uji sebelum menerapkan perubahan ke klaster produksi Anda. Untuk informasi tentang pencadangan kluster Anda, lihat Membuat Cadangan dan Memulihkan di Amazon DocumentDB.

Kueri Alur Agregasi

Ketika membuat kueri alur agregasi dengan beberapa tahap dan mengevaluasi hanya sebagian dari data dalam kueri, gunakan tahap `$match` sebagai tahap pertama atau di awal alur. Menggunakan `$match` terlebih dahulu akan mengurangi jumlah dokumen yang perlu diproses pada tahap selanjutnya dalam kueri alur agregasi, sehingga meningkatkan kinerja kueri Anda.

`batchInsert` dan `batchUpdate`

Saat melakukan tingkat konkuren `batchInsert` dan/atau `batchUpdate` operasi yang tinggi, dan jumlah `FreeableMemory` (CloudWatch Metrik) menjadi nol pada instans utama Anda, Anda dapat

mengurangi konkurensi penyisipan batch atau memperbarui beban kerja atau, jika konkurensi beban kerja tidak dapat dikurangi, tingkatkan ukuran instans untuk menambah jumlah. `FreeableMemory`

Perbedaan Fungsional: Amazon DocumentDB dan MongoDB

Berikut ini adalah perbedaan fungsional antara Amazon DocumentDB (dengan kompatibilitas MongoDB) dan MongoDB.

Topik

- Manfaat fungsional dari Amazon DocumentDB
- Perbedaan Fungsional yang Diperbarui
- Perbedaan fungsional pada MongoDB

Manfaat fungsional dari Amazon DocumentDB

Transaksi Implisit

Di Amazon DocumentDB, semua pernyataan CRUD (`findAndModify`, `update`, `insert`, `delete`) menjamin atomisitas dan konsistensi, bahkan untuk operasi yang memodifikasi beberapa dokumen. Dengan peluncuran Amazon DocumentDB 4.0, transaksi eksplisit yang menyediakan properti ACID untuk operasi multi-pernyataan dan multi-koleksi kini didukung. Untuk lebih lanjut tentang menggunakan transaksi di Amazon DocumentDB, silakan lihat tampilkan Transaksi.

Berikut ini adalah contoh operasi di Amazon DocumentDB yang memodifikasi beberapa dokumen yang memenuhi perilaku atomik dan konsisten.

```
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 }]
})
```

Operasi individu yang menyusun operasi massal seperti `updateMany` dan `deleteMany` bersifat atomik tetapi keseluruhan operasi massal tidak bersifat atomik. Misalnya, keseluruhan operasi `insertMany` bersifat atomik jika operasi penyisipan individual berhasil dijalankan tanpa kesalahan. Jika kesalahan ditemukan pada operasi `insertMany`, setiap pernyataan penyisipan individu dalam operasi `insertMany` akan dijalankan sebagai operasi atom. Jika Anda memerlukan properti ACID untuk operasi `insertMany`, `updateMany`, and `deleteMany`, Anda sebaiknya menggunakan transaksi.

Perbedaan Fungsional yang Diperbarui

Amazon DocumentDB terus meningkatkan kompatibilitas pada MongoDB dengan merujuk kembali pada kemampuan yang perlu kami bangun sesuai permintaan pelanggan. Bagian ini berisi perbedaan fungsional yang telah kami hilangkan dari Amazon DocumentDB untuk mempermudah migrasi dan pembangunan aplikasi bagi pelanggan kami.

Topik

- Pengindeksan Array
- Indeks Multi-kunci
- Karakter Null dalam String
- Kontrol Akses Berbasis Peran
- $regex Pengindeksan
- Proyeksi untuk Dokumen Bersarang

Pengindeksan Array

Mulai 23 April 2020, Amazon DocumentDB kini mendukung kemampuan untuk mengindeks array yang lebih besar daripada 2.048 byte. Batas untuk setiap item individual dalam array masih tetap 2.048 byte, dan ini konsisten dengan MongoDB.

Jika Anda membuat indeks baru, tidak ada tindakan yang diperlukan untuk memanfaatkan fungsionalitas yang ditingkatkan. Jika Anda sudah memiliki indeks, Anda dapat memanfaatkan peningkatan fungsionalitas dengan menghapus indeks tersebut dan membuatnya lagi. Versi indeks saat ini dengan kemampuan yang ditingkatkan adalah `"v" : 3`.

> **Note**
>
> Untuk klaster produksi, penghapusan indeks mungkin berdampak pada performa aplikasi Anda. Kami menyarankan Anda untuk terlebih dahulu menguji dan melanjutkan dengan hati-hati ketika membuat perubahan pada sistem produksi. Selain itu, waktu yang dibutuhkan untuk menciptakan indeks akan menjadi fungsi ukuran data secara keseluruhan pada pengumpulan.

Anda dapat mengajikan kueri untuk versi indeks Anda dengan menggunakan perintah berikut.

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

Output dari operasi ini terlihat seperti berikut ini. Dalam output ini, versi indeks adalah `"v" : 3`, yang merupakan versi indeks terbaru.

```
[
```

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

Indeks Multi-kunci

Mulai 23 April 2020, Amazon DocumentDB sekarang mendukung kemampuan untuk membuat indeks gabungan dengan beberapa kunci dalam array yang sama.

Jika Anda membuat indeks baru, tidak ada tindakan yang diperlukan untuk memanfaatkan fungsionalitas yang ditingkatkan. Jika Anda sudah memiliki indeks, Anda dapat memanfaatkan peningkatan fungsionalitas dengan menghapus indeks tersebut dan membuatnya lagi. Versi indeks saat ini dengan kemampuan yang ditingkatkan adalah `"v" : 3`.

> **Note**
>
> Untuk klaster produksi, penghapusan indeks mungkin berdampak pada performa aplikasi Anda. Kami menyarankan Anda untuk terlebih dahulu menguji dan melanjutkan dengan hati-hati ketika membuat perubahan pada sistem produksi. Selain itu, waktu yang dibutuhkan untuk menciptakan indeks akan menjadi fungsi ukuran data secara keseluruhan pada pengumpulan.

Anda dapat mengajikan kueri untuk versi indeks Anda dengan menggunakan perintah berikut.

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

Output dari operasi ini terlihat seperti berikut ini. Dalam output ini, versi indeks adalah `"v" : 3`, yang merupakan versi indeks terbaru.

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

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

Karakter Null dalam String

Mulai 22 Juni 2020, Amazon DocumentDB sekarang mendukung karakter null (`'\0'`) dalam string.

Kontrol Akses Berbasis Peran

Mulai 26 Maret 2020, Amazon DocumentDB mendukung kontrol akses berbasis peran (RBAC) untuk peran bawaan. Untuk mempelajari selengkapnya, lihat Kontrol Akses Berbasis Peran.

`$regex` Pengindeksan

Mulai 22 Juni 2020, Amazon DocumentDB kini mendukung kemampuan operator `$regex` untuk menggunakan indeks.

Untuk menggunakan indeks dengan operator `$regex`, Anda harus menggunakan perintah `hint()`. Saat menggunakan `hint()`, Anda harus menentukan nama bidang tempat Anda menerapkan `$regex`. Misalnya, jika Anda memiliki indeks pada bidang `product` dengan nama indeks sebagai `p_1`, `db.foo.find({product: /^x.*/}).hint({product:1})` akan menggunakan indeks `p_1`, tetapi `db.foo.find({product: /^x.*/}).hint("p_1")` tidak akan menggunakan indeks. Anda dapat memverifikasi apakah suatu indeks dipilih dengan menggunakan perintah `explain()` atau menggunakan profiler untuk mencatat kueri lambat. Misalnya, `db.foo.find({product: /^x.*/}).hint("p_1").explain()`.

> **Note**
>
> Metode `hint()` ini hanya dapat digunakan dengan satu indeks pada satu waktu.

Penggunaan indeks untuk kueri `$regex` dioptimalkan untuk kueri ekspresi reguler yang menggunakan prefiks dan tidak menentukan opsi ekspresi reguler `I`, `m`, atau `o`.

Saat menggunakan indeks dengan `$regex`, Anda disarankan untuk membuat indeks pada bidang yang sangat selektif yang memuat jumlah nilai duplikat kurang dari 1% dari jumlah total dokumen dalam koleksi. Sebagai contoh, jika pengumpulan Anda berisi 100.000 dokumen, buat indeks hanya pada bidang tempat nilai yang sama muncul 1000 kali atau kurang.

Proyeksi untuk Dokumen Bersarang

Pada operator `$project`, terdapat perbedaan fungsional antara Amazon DocumentDB dan MongoDB di versi 3.6 yang telah diselesaikan di Amazon DocumentDB 4.0 tetapi akan tetap tidak didukung di Amazon DocumentDB 3.6.

Amazon DocumentDB 3.6 hanya mempertimbangkan bidang pertama dalam dokumen bersarang saat menerapkan proyeksi, sedangkan MongoDB 3.6 akan mengurai subdokumen dan menerapkan proyeksi ke setiap sub dokumen juga.

Misalnya: jika proyeksinya adalah `“a.b.c”: 1`, maka perilaku tersebut berfungsi seperti yang diharapkan di Amazon DocumentDB dan MongoDB. Namun, jika proyeksinya adalah `{a:{b:{c:1}}}` maka Amazon DocumentDB 3.6 hanya akan menerapkan proyeksi ke `a` dan bukan `b` atau `c`. Di Amazon DocumentDB 4.0, proyeksi `{a:{b:{c:1}}}` akan diterapkan ke `a`, `b`, dan `c`.

Perbedaan fungsional pada MongoDB

Topik

- Operator $vectorSearch
- OpCountersCommand
- Basis Data Admin dan Koleksi
- cursormaxTimeMS
- jelaskan()
- Pembatasan Nama Bidang
- Pembuatan indeks
- Pencarian dengan kunci kosong di jalur
- API, Operasi, dan Tipe Data MongoDB
- Utilitas mongodump dan mongorestore
- Pengurutan Hasil

- Penulisan yang dapat dicoba lagi
- Indeks Jarang
- Menggunakan $elemMatch Dalam Ekspresi $all
- Pengindeksan $ne, $nin, $nor, $not, $exists, dan $elemMatch
- $lookup

Operator **$vectorSearch**

Amazon DocumentDB tidak `$vectorSearch` mendukung sebagai operator independen. Sebaliknya kami mendukung, `vectorSearch` di dalam `$search` operator. Untuk informasi selengkapnya, lihat Pencarian vektor untuk Amazon DocumentDB.

OpCountersCommand

`OpCountersCommand`Perilaku Amazon DocumentDB menyimpang dari MongoDB sebagai berikut: `opcounters.command`

- MongoDB `opcounters.command` menghitung semua perintah kecuali menyisipkan, memperbarui, dan menghapus sementara Amazon DocumentDB juga `OpCountersCommand` mengecualikan perintah. `find`
- Amazon DocumentDB menghitung perintah internal (seperti) ke arah. `getCloudWatchMetricsV2` `OpCountersCommand`

Basis Data Admin dan Koleksi

Amazon DocumentDB tidak mendukung basis data admin atau lokal atau koleksi MongoDB `system.*` atau `startup_log` secara berurutan.

cursormaxTimeMS

Di Amazon DocumentDB, `cursor.maxTimeMS` menyetel ulang penghitung untuk setiap permintaan `getMore`. Jadi, jika `maxTimeMS` 3000MS ditentukan, kueri membutuhkan 2800MS, dan setiap permintaan `getMore` berikutnya membutuhkan 300MS, maka kursor tidak akan kehabisan waktu. Kursor hanya akan kehabisan waktu jika satu operasi, baik kueri atau permintaan individu `getMore`, membutuhkan lebih dari `maxTimeMS` yang ditentukan. Selanjutnya, penyapu yang memeriksa waktu eksekusi kursor berjalan pada granularitas lima (5) menit.

jelaskan()

Amazon DocumentDB mengemulasi API MongoDB 4.0 pada mesin basis data yang dibuat khusus yang memanfaatkan sistem penyimpanan yang didistribusikan, toleran terhadap kesalahan, dan penyembuhkan diri sendiri. Akibatnya, rencana kueri dan keluaran dari `explain()` mungkin berbeda antara Amazon DocumentDB dan MongoDB. Pelanggan yang ingin kontrol atas rencana kueri mereka dapat menggunakan operator `$hint` untuk memaksa pemilihan indeks yang diutamakan.

Pembatasan Nama Bidang

Amazon DocumentDB tidak mendukung titik-titik “.” dalam nama bidang dokumen, misalnya, `db.foo.insert({‘x.1’:1})`.

Amazon DocumentDB juga tidak mendukung prefiks $ dalam nama bidang.

Misalnya, coba perintah berikut di Amazon DocumentDB atau MongoDB:

```
rs0:PRIMARY< db.foo.insert({"a":{"$a":1}})
```

MongoDB akan menampilkan hal berikut:

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

Amazon DocumentDB akan menampilkan kesalahan:

```
WriteResult({
"nInserted" : 0,
"writeError" : {
"code" : 2,
"errmsg" : "Document can't have $ prefix field names: $a"
}
})
```

> **Note**
>
> Ada pengecualian untuk perbedaan fungsional ini. Nama bidang berikut yang dimulai dengan prefiks $ telah masuk daftar putih dan dapat berhasil digunakan di Amazon DocumentDB: $id, $ref, dan $db.

Pembuatan indeks

Amazon DocumentDB memungkinkan hanya satu build indeks terjadi pada koleksi pada waktu tertentu. Baik di latar depan atau latar belakang. Jika operasi seperti `createIndex()` atau `dropIndex()` terjadi pada koleksi yang sama saat pembuatan indeks sedang berlangsung, operasi yang baru dicoba akan gagal.

Secara default, build indeks di Amazon DocumentDB dan MongoDB versi 4.0 terjadi di latar belakang. MongoDB versi 4.2, dan yang lebih baru mengabaikan opsi pembuatan indeks latar belakang jika ditentukan ke createIndexes atau pembantu shell-nya dan. `createIndex()` `createIndexes()`

Indeks Time to Live (TTL) mulai kedaluwarsa dokumen setelah pembuatan indeks selesai.

Pencarian dengan kunci kosong di jalur

Saat Anda melakukan pencarian dengan kunci yang menyertakan string kosong sebagai bagian dari jalur (mis. `x.`, `x..b`), dan objek memiliki jalur kunci string kosong (mis. `{"x" : [ { "" : 10 }, { "b" : 20 } ]}`) di dalam array, Amazon DocumentDB akan mengembalikan hasil yang berbeda dibandingkan jika Anda menjalankan pencarian yang sama di MongoDB.

Di MongoDB, pencarian jalur kunci kosong di dalam array berfungsi seperti yang diharapkan ketika kunci string kosong tidak berada di ujung pencarian jalur. Namun, ketika berada di ujung pencarian jalur, kunci string kosong tidak melihat ke dalam array.

Namun di Amazon DocumentDB, hanya elemen pertama dalam array yang dibaca, karena `getArrayIndexFromKeyString` mengonversi string kosong menjadi `0`, sehingga pencarian kunci string diperlakukan sebagai pencarian indeks array.

API, Operasi, dan Tipe Data MongoDB

Amazon DocumentDB kompatibel dengan API MongoDB 3.6 dan 4.0. Untuk up-to-date daftar fungsionalitas yang didukung, lihatAPI, Operasi, dan Tipe Data MongoDB Yang Didukung.

Utilitas **`mongodump`** dan **`mongorestore`**

Amazon DocumentDB tidak mendukung basis data admin dan dengan demikian tidak membuang atau memulihkan bssis data admin saat menggunakan `mongodump` atau utilitas `mongorestore`. Saat Anda membuat basis data baru di Amazon DocumentDB dengan menggunakan `mongorestore`, Anda perlu membuat ulang peran pengguna selain operasi pemulihan.

> **Note**
>
> Kami merekomendasikan MongoDB Database Tools hingga dan termasuk versi 100.6.1 untuk Amazon DocumentDB. Anda dapat mengakses unduhan Alat Database MongoDB di sini.

Pengurutan Hasil

Amazon DocumentDB tidak menjamin pengurutan kategori hasil secara impilisit dari kumpulan hasil. Untuk memastikan pengurutan kumpulan hasil, tentukan pengurutan kategori secara eksplisit dengan menggunakan `sort()`.

Contoh berikut mengategorikan item dalam koleksi inventaris dalam urutan menurun berdasarkan bidang stok.

```
db.inventory.find().sort({ stock: -1 })
```

Saat menggunakan tahap agregasi `$sort`, urutan kategori tidak dipertahankan kecuali jika tahap `$sort` tersebut adalah tahap terakhir dalam alur agregasi. Saat menggunakan tahap agregasi `$sort` bersama dengan tahap agregasi `$group`, tahap agregasi `$sort` hanya diterapkan pada akumulator `$first` dan `$last`. Di Amazon DocumentDB 4.0, dukungan ditambahkan untuk `$push` sesuai pengurutan kategori dari tahap `$sort` sebelumnya.

Penulisan yang dapat dicoba lagi

Dimulai dengan driver yang kompatibel dengan MongoDB 4.2, penulisan yang dapat dicoba kembali diaktifkan secara default. Namun, Amazon DocumentDB saat ini tidak mendukung penulisan yang dapat dicoba lagi. Perbedaan fungsional akan memanifestasikan dirinya dalam pesan kesalahan yang mirip dengan hal berikut ini.

```
{"ok":0,"errmsg":"Unrecognized field: 'txnNumber'","code":9,"name":"MongoError"}
```

Tulisan yang dapat dicoba ulang dapat dinonaktifkan melalui string koneksi (misalnya, `MongoClient("mongodb://my.mongodb.cluster/db?retryWrites=false"))` atau argumen kata kunci MongoClient konstruktor (misalnya,. `MongoClient("mongodb://my.mongodb.cluster/db", retryWrites=False))`

Berikut ini adalah contoh Python yang menonaktifkan penulisan yang dapat dicoba lagi dalam string koneksi.

```
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)
```

Indeks Jarang

Untuk menggunakan indeks jarang yang telah Anda buat dalam kueri, Anda harus menggunakan klausa `$exists` pada bidang yang mencakup indeks tersebut. Jika Anda menghilangkan `$exists`, Amazon DocumentDB tidak menggunakan indeks jarang.

Berikut ini adalah contoh.

```
db.inventory.count({ "stock": { $exists: true }})
```

Untuk indeks multi-kunci yang jarang, Amazon DocumentDB tidak mendukung batasan kunci unik jika pencarian dokumen menghasilkan sekumpulan nilai dan hanya sebagian dari bidang yang diindeks yang hilang. Misalnya, `createIndex({"a.b" : 1 }, { unique : true, sparse :true })` tidak didukung, mengingat input `"a" : [ { "b" : 2 }, { "c" : 1 } ]`, seperti `"a.c"` yang disimpan dalam indeks.

Menggunakan $elemMatch Dalam Ekspresi $all

Amazon DocumentDB saat ini tidak mendukung penggunaan operator `$elemMatch` dalam ekspresi `$all`. Sebagai solusinya, Anda dapat menggunakan operator `$and` dengan `$elemMatch` sebagai berikut.

Operasi asli:

```
db.col.find({
  qty: {
    $all: [
      { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } },
      { "$elemMatch": { num: 40, size: "XL" } }
    ]
  }
```

```
})
```

Operasi yang diperbarui:

```
db.col.find({
  $and: [
    { qty: { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } } },
    { qty: { "$elemMatch": { qty: 40, size: "XL" } } }
  ]
})
```

Pengindeksan **$ne**, **$nin**, **$nor**, **$not**, **$exists**, dan **$elemMatch**

Amazon DocumentDB saat ini tidak mendukung kemampuan untuk menggunakan indeks dengan operator `$ne`, `$nin`, `$nor`, `$not`, `$exists`, dan `$distinct`. Akibatnya, menggunakan operator ini akan menghasilkan pemindaian koleksi. Melakukan filter atau pencocokan sebelum menggunakan salah satu operator ini akan mengurangi jumlah data yang perlu dipindai, dan dengan demikian dapat meningkatkan kinerja.

Amazon DocumentDB menambahkan dukungan untuk pemindaian indeks dengan operator `$elemMatch` di Amazon DocumentDB 5.0 dan cluster elastis. Pemindaian indeks didukung ketika filter kueri hanya memiliki satu tingkat `$elemMatch` filter tetapi tidak didukung jika `$elemMatch` kueri bersarang disertakan.

`$elemMatch`bentuk kueri yang mendukung pemindaian indeks di Amazon DocumentDB 5.0:

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

`$elemMatch`bentuk kueri yang tidak mendukung pemindaian indeks di Amazon DocumentDB 5.0:

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e": "abc"} }} } })
```

$lookup

Amazon DocumentDB mendukung kemampuan untuk melakukan kecocokan kesetaraan (misalnya, gabungan luar kiri) dan juga mendukung subkueri yang tidak berkorelasi, tetapi tidak mendukung subkueri yang berkorelasi.

Memanfaatkan indeks dengan **$lookup**

Anda sekarang dapat menggunakan indeks dengan operator tahapan `$lookup`. Berdasarkan kasus penggunaan Anda, ada beberapa algoritme pengindeksan yang dapat Anda gunakan untuk mengoptimalkan kinerja. Bagian ini akan menjelaskan berbagai algoritme pengindeksan untuk `$lookup` dan membantu Anda memilih yang terbaik untuk beban kerja Anda.

Secara default, Amazon DocumentDB akan menggunakan algoritme hash saat `allowDiskUse:false` digunakan dan mengurutkan gabungan saat `allowDiskUse:true` digunakan. Untuk beberapa kasus penggunaan, akan lebih bermanfaat jika pengoptimal kueri didesak untuk menggunakan algoritme yang berbeda. Di bawah ini adalah berbagai algoritma pengindeksan yang dapat digunakan oleh operator agregasi `$lookup`:

- Nested loop: Rencana loop bersarang biasanya bermanfaat untuk beban kerja jika koleksi asing <1 GB dan bidang dalam koleksi asing memiliki indeks. Jika algoritma loop bersarang sedang digunakan, rencana penjelasan akan menunjukkan tahap sebagai`NESTED_LOOP_LOOKUP`.
- Penggabungan pengurutan: Rencana penggabungan pengurutan biasanya bermanfaat untuk beban kerja jika koleksi asing tidak memiliki indeks pada bidang yang digunakan dalam pencarian dan kumpulan data yang berfungsi tidak sesuai dengan memori. Jika algoritma penggabungan pengurutan sedang digunakan, rencana penjelasan akan menampilkan tahapan sebagai`SORT_LOOKUP`.
- Hash: Paket hash biasanya bermanfaat untuk beban kerja jika koleksi asing < 1GB dan kumpulan data yang berfungsi sesuai dengan memori. Jika algoritma hash sedang digunakan, rencana penjelasan akan menunjukkan tahap sebagai`HASH_LOOKUP`.

Anda dapat mengidentifikasi algoritme pengindeksan yang digunakan untuk operator `$lookup` dengan menggunakan penjelasan pada kueri. Berikut adalah contoh.

```
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
}
```

Sebagai alternatif terhadap penggunaan metode `explain()` ini, Anda dapat menggunakan profiler untuk meninjau algoritme yang digunakan saat Anda menggunakan operator `$lookup`. Untuk informasi lebih lanjut tentang profiler, silakan lihat Membuat Profil Operasi Amazon DocumentDB.

Menggunakan **planHint**

Jika Anda ingin memaksa pengoptimal kueri untuk menggunakan algoritme pengindeksan yang berbeda dengan `$lookup`, Anda dapat menggunakan `planHint`. Untuk melakukannya, gunakan komentar dalam opsi tahap agregasi untuk memaksa rencana yang berbeda. Di bawah ini adalah contoh sintaks untuk komentar:

```
comment : {
    comment :  “<string>”,
    lookupStage : { planHint : “SORT” | “HASH” | "NESTED_LOOP" }
}
```

Di bawah ini adalah contoh penggunaan `planHint` untuk memaksa pengoptimal kueri untuk menggunakan algoritme pengindeksan `HASH`:

```
db.foo.aggregate(
  [
    {
        $lookup:
        {
            from: "foo",
            localField: "_id",
            foreignField: "_id",
            as: "joined"
        },
    }
  ],
  {
    comment : "{ \\"lookupStage\\" : { \\"planHint\\": \\"HASH\\" }}"
```

Untuk menguji algoritme mana yang terbaik untuk beban kerja Anda, Anda dapat menggunakan parameter `executionStats` dari metode `explain` untuk mengukur waktu eksekusi tahapan `$lookup` saat memodifikasi algoritme pengindeksan (yaitu, `HASH`/`SORT`/`NESTED_LOOP`).

Contoh berikut menunjukkan cara menggunakan `executionStats` untuk mengukur waktu eksekusi tahap `$lookup` dengan menggunakan algoritme `SORT`.

```
db.foo.explain(“executionStats”).aggregate(
  [
    {
        $lookup:
```

```
        {
            from: "foo",
            localField: "_id",
            foreignField: "_id",
            as: "joined"
        },
    }
],
{
  comment : "{ \\"lookupStage\\" : { \\"planHint\": \\"SORT\\" }}"
```

API, Operasi, dan Tipe Data MongoDB Yang Didukung

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan basis data dokumen yang cepat, dapat diskalakan, sangat tersedia, dan terkelola penuh yang mendukung beban kerja MongoDB. Amazon DocumentDB kompatibel dengan API MongoDB 3.6, 4.0, dan 5.0. Bagian ini mencantumkan fungsionalitas yang didukung. Untuk dukungan menggunakan API dan driver MongoDB, silakan berkonsultasi dengan Forum Komunitas MongoDB. Untuk dukungan menggunakan layanan Amazon DocumentDB, silakan hubungi tim dukungan yang AWS sesuai. Untuk perbedaan fungsional antara Amazon DocumentDB dan MongoDB, silakan lihat Perbedaan Fungsional: Amazon DocumentDB dan MongoDB.

Perintah dan operator MongoDB yang hanya berlaku untuk internal atau tidak berlaku untuk layanan terkelola penuh tidak didukung dan tidak termasuk dalam daftar fungsi yang didukung.

Kami telah menambahkan lebih dari 50+ kemampuan tambahan sejak diluncurkan, dan akan terus bekerja mundur dari pelanggan kami untuk memberikan kemampuan yang mereka butuhkan. Untuk informasi tentang peluncuran terbaru, lihat Pengumuman Amazon DocumentDB.

Jika ada fitur yang tidak didukung yang Anda ingin kami bangun, beri tahu kami dengan mengirimkan email dengan accountID Anda, fitur yang diminta, dan kasus penggunaan ke tim layanan Amazon DocumentDB.

Topik

- Perintah Basis Data
- Kueri dan Proyeksi Operator
- Operator Pembaruan
- Geospasial
- Metode kursor
- Operator Agregasi Pipeline
- Jenis Data
- Indeks dan Properti Indeks

Perintah Basis Data

Topik

- Perintah administratif
- Agregasi
- Autentikasi
- Perintah diagnostik
- Operasi Kueri dan Tulis
- Perintah Pengelolaan Peran
- Perintah Sesi
- Pengelolaan Pengguna
- Perintah Sharding

Perintah administratif

Perintah	3.6	4.0	5.0	Cluster elastis
Koleksi Terbatas	Tidak	Tidak	Tidak	Tidak
klon Tertutup CollectionAs	Tidak	Tidak	Tidak	Tidak
collMod	Sebagian	Sebagian	Sebagian	Sebagian
CollMod: expireAft erSeconds	Ya	Ya	Ya	Ya
mengkonversi ToCapped	Tidak	Tidak	Tidak	Tidak
copydb	Tidak	Tidak	Tidak	Tidak
Buat	Ya	Ya	Ya	Ya
createView	Tidak	Tidak	Tidak	Tidak
createIndex	Ya	Ya	Ya	Ya
currentOp	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
drop	Ya	Ya	Ya	Ya
dropDatabase	Ya	Ya	Ya	Ya
dropIndexes	Ya	Ya	Ya	Ya
filemd5	Tidak	Tidak	Tidak	Tidak
killCursors	Ya	Ya	Ya	Ya
killOp	Ya	Ya	Ya	Ya
ListCollection*	Ya	Ya	Ya	Ya
listDatabases	Ya	Ya	Ya	Ya
listIndexes	Ya	Ya	Ya	Ya
reIndex	Tidak	Tidak	Tidak	Tidak
renameCollection	Ya	Ya	Ya	Tidak

* `type` Kunci dalam opsi filter tidak didukung.

Agregasi

Perintah	3.6	4.0	5.0	Cluster elastis
agregasi	Ya	Ya	Ya	Ya
hitungan	Ya	Ya	Ya	Ya
berbeda	Ya	Ya	Ya	Ya
mapReduce	Tidak	Tidak	Tidak	Tidak

Autentikasi

Perintah	3.6	4.0	5.0	Cluster elastis
diautentikasi	Ya	Ya	Ya	Ya
logout	Ya	Ya	Ya	Ya

Perintah diagnostik

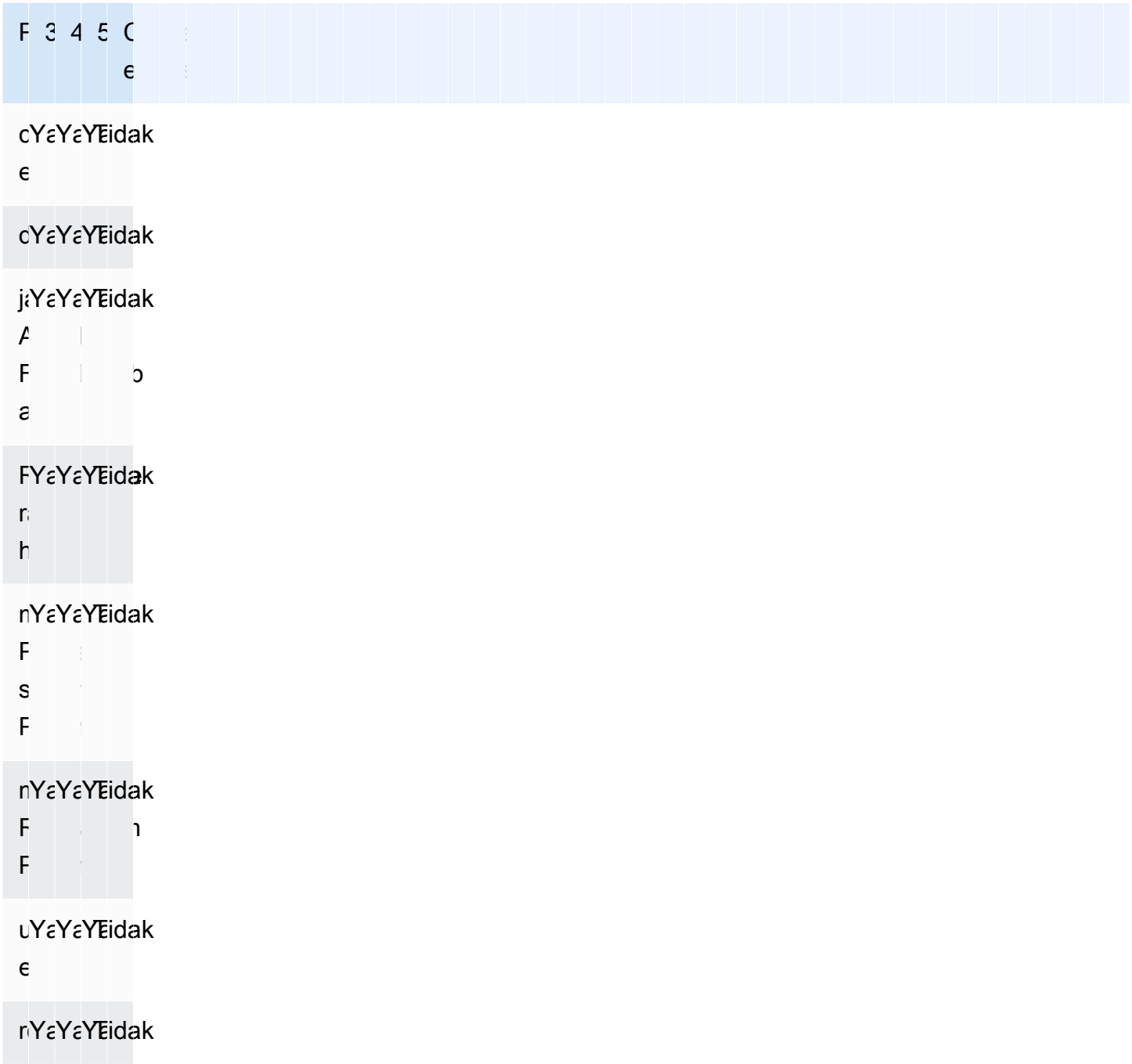
Perintah	3.6	4.0	5.0	Cluster elastis
buildInfo	Ya	Ya	Ya	Ya
collStats	Ya	Ya	Ya	Ya
conn PoolStats	Tidak	Tidak	Tidak	Tidak
connectionStatus	Ya	Ya	Ya	Ya
dataSize	Ya	Ya	Ya	Ya
dbHash	Tidak	Tidak	Tidak	Tidak
dbStats	Ya	Ya	Ya	Ya
jelaskan	Ya	Ya	Ya	Ya
jelaskan: executionStats	Ya	Ya	Ya	Ya
fitur	Tidak	Tidak	Tidak	Tidak
hostInfo	Ya	Ya	Ya	Ya
listCommands	Ya	Ya	Ya	Ya
profiler	Ya	Ya	Ya	Tidak

Perintah	3.6	4.0	5.0	Cluster elastis
serverStatus	Ya	Ya	Ya	Ya
atas	Ya	Ya	Ya	Ya

Operasi Kueri dan Tulis

Perintah	3.6	4.0	5.0	Cluster elastis
hapus	Ya	Ya	Ya	Ya
cari	Ya	Ya	Ya	Ya
menemukan AndModify	Ya	Ya	Ya	Ya
mendapatkan LastError	Tidak	Tidak	Tidak	Tidak
getMore	Ya	Ya	Ya	Ya
mendapatkan PrevError	Tidak	Tidak	Tidak	Tidak
sisipkan	Ya	Ya	Ya	Ya
parallel Collectio nScan	Tidak	Tidak	Tidak	Tidak
resetError	Tidak	Tidak	Tidak	Tidak
perbarui	Ya	Ya	Ya	Ya
Mengelola aliran	Ya	Ya	Ya	Tidak
GridFS	Tidak	Tidak	Tidak	Tidak
ReplaceOne	Ya	Ya	Ya	Ya

Perintah Pengelolaan Peran



Perintah Sesi

Perintah	3.6	4.0	5.0	Cluster elastis
abortTransaction	Tidak	Ya	Ya	Tidak
commitTransaction	Tidak	Ya	Ya	Tidak
endSessions	Tidak	Tidak	Tidak	Tidak
killAllSessions	Tidak	Ya	Ya	Tidak
membunuhAllSessionsByPattern	Tidak	Tidak	Tidak	Tidak
killSessions	Tidak	Ya	Ya	Tidak
refreshsessions	Tidak	Tidak	Tidak	Tidak
startSession	Tidak	Ya	Ya	Tidak

Pengelolaan Pengguna

Perintah	3.6	4.0	5.0	Cluster elastis
createUser	Ya	Ya	Ya	Ya
jatuhkanAllUsersFromDatabase	Ya	Ya	Ya	Ya
dropUser	Ya	Ya	Ya	Ya
memberikan RolesToPengguna	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
mencabut Pengguna RolesFrom	Ya	Ya	Ya	Ya
updateUser	Ya	Ya	Ya	Ya
userInfo	Ya	Ya	Ya	Ya

Perintah Sharding

Perintah	Cluster elastis
menggugurkan ReshardCollection	Tidak
AddsHard	Tidak
tambahkan ShardTo Zona	Tidak
penyeimbang CollectionStatus	Tidak
BalancerStart	Tidak
BalancerStatus	Tidak
BalancerSTOP	Tidak
periksa ShardingIndex	Tidak
jelas JumboFlag	Tidak
cleanupOrphaned	Tidak
pembersihan ReshardCollection	Tidak
berkomitmen ReshardCollection	Tidak
Aktifkan Sharding	Ya

Perintah	Cluster elastis
siram RouterConfig	Tidak
mendapatkan ShardMap	Tidak
mendapatkan ShardVersion	Tidak
ISDBgrid	Tidak
ListShards	Tidak
MedianKey	Tidak
MoveChunk	Tidak
MovePrimary	Tidak
MergeChunks	Tidak
memperbaiki Kunci CollectionShard	Tidak
MenghapusHard	Tidak
hapus ShardFrom Zona	Tidak
ResHardCollection	Tidak
set AllowMigrations	Tidak
set ShardVersion	Tidak
ShardCollection	Ya
ShardingState	Tidak
terbelah	Tidak
SplitVector	Tidak
UnsetSharding	Tidak

Perintah	Cluster elastis
Perbarui ZoneKey Rentang	Tidak

Kueri dan Proyeksi Operator

Topik

- Operator Array
- Operator Bitwise
- Operator Komentar
- Operator Perbandingan
- Operator Elemen
- Operator kueri evaluasi
- Operator Logis
- Operator proyeksi

Operator Array

Perintah	3.6	4.0	5.0	Cluster elastis
$all	Ya	Ya	Ya	Ya
$elemMatch	Ya	Ya	Ya	Ya
$size	Ya	Ya	Ya	Ya

Operator Bitwise

Perintah	3.6	4.0	5.0	Cluster elastis
$ bit AllSet	Ya	Ya	Ya	Ya
$ bit AnySet	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
$ bit AllClear	Ya	Ya	Ya	Ya
$ bit AnyClear	Ya	Ya	Ya	Ya

Operator Komentar

Perintah	3.6	4.0	5.0	Cluster elastis
$comment	Ya	Ya	Ya	Ya

Operator Perbandingan

Perintah	3.6	4.0	5.0	Cluster elastis
$eq	Ya	Ya	Ya	Ya
$gt	Ya	Ya	Ya	Ya
$gte	Ya	Ya	Ya	Ya
$lt	Ya	Ya	Ya	Ya
$lte	Ya	Ya	Ya	Ya
$ne	Ya	Ya	Ya	Ya
$in	Ya	Ya	Ya	Ya
$nin	Ya	Ya	Ya	Ya

Operator Elemen

Perintah	3.6	4.0	5.0	Cluster elastis
$exists	Ya	Ya	Ya	Ya
$type	Ya	Ya	Ya	Ya

Operator kueri evaluasi

Perintah	3.6	4.0	5.0	Cluster elastis
$expr	Tidak	Ya	Ya	Tidak
$JSONSchema	Tidak	Ya	Ya	Tidak
$mod	Ya	Ya	Ya	Ya
$regex	Ya	Ya	Ya	Ya
$text	Tidak	Tidak	Ya	Tidak
$where	Tidak	Tidak	Tidak	Tidak

Operator Logis

Perintah	3.6	4.0	5.0	Cluster elastis
$or	Ya	Ya	Ya	Ya
$and	Ya	Ya	Ya	Ya
$not	Ya	Ya	Ya	Ya
$nor	Ya	Ya	Ya	Ya

Operator proyeksi

Perintah	3.6	4.0	5.0	Cluster elastis
$	Ya	Ya	Ya	Ya
$elemMatch	Ya	Ya	Ya	Ya
$meta	Tidak	Tidak	Ya	Tidak
$slice	Ya	Ya	Ya	Ya

Operator Pembaruan

Topik

- Operator Array
- Operator Bitwise
- Operator Bidang
- Pembaruan Pemodifikasi

Operator Array

Perintah	3.6	4.0	5.0	Cluster elastis
$	Ya	Ya	Ya	Ya
$[]	Ya	Ya	Ya	Ya
$[<identifier>]	Ya	Ya	Ya	Ya
$ tambahkan ToSet	Ya	Ya	Ya	Ya
$pop	Ya	Ya	Ya	Ya
$pullAll	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
$pull	Ya	Ya	Ya	Ya
$push	Ya	Ya	Ya	Ya

Operator Bitwise

Perintah	3.6	4.0	5.0	Cluster elastis
$bit	Ya	Ya	Ya	Ya

Operator Bidang

Operator	3.6	4.0	5.0	Cluster elastis
$inc	Ya	Ya	Ya	Ya
$mul	Ya	Ya	Ya	Ya
$rename	Ya	Ya	Ya	Ya
$ set OnInsert	Ya	Ya	Ya	Ya
$set	Ya	Ya	Ya	Ya
$unset	Ya	Ya	Ya	Ya
$min	Ya	Ya	Ya	Ya
$max	Ya	Ya	Ya	Ya
$currentDate	Ya	Ya	Ya	Ya

Pembaruan Pemodifikasi

Operator	3.6	4.0	5.0	Cluster elastis
$each	Ya	Ya	Ya	Ya
$slice	Ya	Ya	Ya	Ya
$sort	Ya	Ya	Ya	Ya
$position	Ya	Ya	Ya	Ya

Geospasial

Penentu Geometri

Pemilih Kueri	3.6	4.0	5.0	Cluster elastis
$box	Tidak	Tidak	Tidak	Tidak
$center	Tidak	Tidak	Tidak	Tidak
$centerSphere	Tidak	Tidak	Tidak	Tidak
$nearSphere	Ya	Ya	Ya	Tidak
$geometry	Ya	Ya	Ya	Tidak
$maxDistance	Ya	Ya	Ya	Tidak
$minDistance	Ya	Ya	Ya	Tidak
$polygon	Tidak	Tidak	Tidak	Tidak
$uniqueDocs	Tidak	Tidak	Tidak	Tidak

Pemilih Kueri

Perintah	3.6	4.0	5.0	Cluster elastis
$geointersects	Ya	Ya	Ya	Tidak
$geowithin	Ya	Ya	Ya	Tidak
$near	Tidak	Tidak	Tidak	Tidak
$nearSphere	Ya	Ya	Ya	Tidak
$polygon	Tidak	Tidak	Tidak	Tidak
$uniqueDocs	Tidak	Tidak	Tidak	Tidak

Metode kursor

Perintah	3.6	4.0	5.0	Cluster elastis
cursor.batchsize ()	Ya	Ya	Ya	Ya
cursor.close ()	Ya	Ya	Ya	Ya
cursor.isClosed ()	Ya	Ya	Ya	Ya
cursor.collation()	Tidak	Tidak	Tidak	Tidak
cursor.comment ()	Ya	Ya	Ya	Ya
cursor.count ()	Ya	Ya	Ya	Ya
cursor.explain ()	Ya	Ya	Ya	Tidak
cursor.foreach ()	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
cursor.hasNext ()	Ya	Ya	Ya	Ya
cursor.hint ()	Ya	Ya	Ya	Ya*
cursor.is Exhausted ()	Ya	Ya	Ya	Tidak
cursor.itcount ()	Ya	Ya	Ya	Tidak
cursor.limit ()	Ya	Ya	Ya	Tidak
cursor.map ()	Ya	Ya	Ya	Tidak
cursor.maxScan ()	Ya	Ya	Ya	Tidak
cursor.maxTimes ()	Ya	Ya	Ya	Tidak
cursor.max ()	Tidak	Tidak	Tidak	Tidak
cursor.min ()	Tidak	Tidak	Tidak	Tidak
cursor.next ()	Ya	Ya	Ya	Ya
kursor.no CursorTimeout ()	Tidak	Tidak	Tidak	Tidak
kursor.objs Batch () LeftIn	Ya	Ya	Ya	Tidak
cursor.pretty ()	Ya	Ya	Ya	Tidak
cursor.re adconcern ()	Ya	Ya	Ya	Tidak
cursor.readPref ()	Ya	Ya	Ya	Tidak

Perintah	3.6	4.0	5.0	Cluster elastis
cursor.returnKey ()	Tidak	Tidak	Tidak	Tidak
kursor.show RecordId ()	Tidak	Tidak	Tidak	Tidak
cursor.size ()	Ya	Ya	Ya	Tidak
cursor.skip ()	Ya	Ya	Ya	Tidak
cursor.sort ()	Ya	Ya	Ya	Tidak
cursor.tailable ()	Tidak	Tidak	Tidak	Tidak
cursor.toArray ()	Ya	Ya	Ya	Tidak

* Indeks `hint` didukung dengan ekspresi indeks. Misalnya, `db.foo.find().hint({x:1})`.

Operator Agregasi Pipeline

Topik

- Ekspresi Akumulator
- Operator aritmatika
- Operator Array
- Operator Boolean
- Operator Perbandingan
- Operator Ekspresi Bersyarat
- Operator Tipe Data
- Operator Ukuran Data
- Operator Tanggal
- Operator Literal
- Operator Penggabungan
- Operator Alami

- Operator Set
- Operator Tahap
- Operator String
- Variabel sistem
- Operator Pencarian Teks
- Operator Konversi Tipe
- Operator variabel
- Operator Lain-lain

Ekspresi Akumulator

Ekspresi	3.6	4.0	5.0	Cluster elastis
$sum	Ya	Ya	Ya	Ya
$avg	Ya	Ya	Ya	Ya
$first	Ya	Ya	Ya	Ya
$last	Ya	Ya	Ya	Ya
$max	Ya	Ya	Ya	Ya
$min	Ya	Ya	Ya	Ya
$push	Ya	Ya	Ya	Ya
$ tambahkan ToSet	Ya	Ya	Ya	Ya
$ std DevPop	Tidak	Tidak	Tidak	Tidak
$ std DevSamp	Tidak	Tidak	Tidak	Tidak
$ akumulator	-	-	Tidak	Tidak
$count	-	-	Tidak	Tidak

Operator aritmatika

Perintah	3.6	4.0	5.0	Cluster elastis
$abs	Ya	Ya	Ya	Ya
$add	Ya	Ya	Ya	Ya
$ceil	Tidak	Ya	Ya	Ya
$divide	Ya	Ya	Ya	Ya
$exp	Tidak	Ya	Ya	Ya
$floor	Tidak	Ya	Ya	Ya
$ln	Tidak	Ya	Ya	Ya
$log	Tidak	Ya	Ya	Ya
$log10	Tidak	Ya	Ya	Ya
$mod	Ya	Ya	Ya	Ya
$multiply	Ya	Ya	Ya	Ya
$pow	Tidak	Tidak	Tidak	Tidak
$sqrt	Tidak	Ya	Ya	Ya
$subtract	Ya	Ya	Ya	Ya
$trunc	Tidak	Tidak	Tidak	Tidak
$ bulat	-	-	Tidak	Tidak

Operator Array

Perintah	3.6	4.0	5.0	Cluster elastis
$ array ElemAt	Ya	Ya	Ya	Ya
$ array ToObject	Ya	Ya	Ya	Ya
$concatArrays	Ya	Ya	Ya	Ya
$filter	Ya	Ya	Ya	Ya
$ indeks OfArray	Ya	Ya	Ya	Ya
$isArray	Ya	Ya	Ya	Ya
$ objek ToArray	Ya	Ya	Ya	Ya
$range	Ya	Ya	Ya	Ya
$reverSearray	Ya	Ya	Ya	Ya
$reduce	Ya	Ya	Ya	Ya
$size	Ya	Ya	Ya	Ya
$slice	Ya	Ya	Ya	Ya
$zip	Ya	Ya	Ya	Ya
$in	Ya	Ya	Ya	Ya
$first	-	-	Tidak	Tidak
$last	-	-	Tidak	Tidak

Operator Boolean

Perintah	3.6	4.0	5.0	Cluster elastis
$and	Ya	Ya	Ya	Ya
$or	Ya	Ya	Ya	Ya
$not	Ya	Ya	Ya	Ya

Operator Perbandingan

Perintah	3.6	4.0	5.0	Cluster elastis
$cmp	Ya	Ya	Ya	Ya
$eq	Ya	Ya	Ya	Ya
$gt	Ya	Ya	Ya	Ya
$gte	Ya	Ya	Ya	Ya
$lt	Ya	Ya	Ya	Ya
$lte	Ya	Ya	Ya	Ya
$ne	Ya	Ya	Ya	Ya

Operator Ekspresi Bersyarat

Perintah	3.6	4.0	5.0	Cluster elastis
$cond	Ya	Ya	Ya	Ya
$ifNull	Ya	Ya	Ya	Ya
$switch	Tidak	Ya	Ya	Tidak

Operator Tipe Data

Perintah	3.6	4.0	5.0	Cluster elastis
$type	Ya	Ya	Ya	Ya

Operator Ukuran Data

Perintah	3.6	4.0	5.0	Cluster elastis
$ BinarySize	-	-	Tidak	Tidak
$ BSONSize	-	-	Tidak	Tidak

Operator Tanggal

Perintah	3.6	4.0	5.0	Cluster elastis
$ DateAdd	Tidak	Tidak	Ya	Ya
$ DateSubtract	Tidak	Tidak	Ya	Ya
$ hari OfYear	Ya	Ya	Ya	Ya
$ hari OfMonth	Ya	Ya	Ya	Ya
$ hari OfWeek	Ya	Ya	Ya	Ya
$year	Ya	Ya	Ya	Ya
$month	Ya	Ya	Ya	Ya
$week	Ya	Ya	Ya	Ya
$hour	Ya	Ya	Ya	Ya
$minute	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
$second	Ya	Ya	Ya	Ya
$millisecond	Ya	Ya	Ya	Ya
$ tanggal ToString	Ya	Ya	Ya	Ya
$iso Minggu DayOf	Ya	Ya	Ya	Ya
$isoWeek	Ya	Ya	Ya	Ya
$ tanggal FromParts	Tidak	Tidak	Tidak	Tidak
$ tanggal ToParts	Tidak	Tidak	Tidak	Tidak
$ tanggal FromString	Ya	Ya	Ya	Ya
$ iso WeekYear	Ya	Ya	Ya	Ya
$ DataTrunc	-	-	Tidak	Tidak
$ dataDiff	-	-	Tidak	Tidak

Operator Literal

Perintah	3.6	4.0	5.0	Cluster elastis
$literal	Ya	Ya	Ya	Ya

Operator Penggabungan

Perintah	3.6	4.0	5.0	Cluster elastis
$mergeObjects	Ya	Ya	Ya	Ya

Operator Alami

Perintah	3.6	4.0	5.0	Cluster elastis
$natural	Ya	Ya	Ya	Ya

Operator Set

Perintah	3.6	4.0	5.0	Cluster elastis
$setEquals	Ya	Ya	Ya	Ya
$setIntersection	Ya	Ya	Ya	Ya
$setUnion	Ya	Ya	Ya	Ya
$setDifference	Tidak	Ya	Ya	Ya
$ set IsSubset	Ya	Ya	Ya	Ya
$ apa saja ElementTrue	Tidak	Ya	Ya	Ya
$ semua ElementsTrue	Tidak	Ya	Ya	Ya

Operator Tahap

Perintah	3.6	4.0	5.0	Cluster elastis
$collStats	Tidak	Tidak	Tidak	Tidak
$project	Ya	Ya	Ya	Ya
$match	Ya	Ya	Ya	Ya
$redact	Ya	Ya	Ya	Ya
$limit	Ya	Ya	Ya	Ya
$skip	Ya	Ya	Ya	Ya
$unwind	Ya	Ya	Ya	Ya
$group	Ya	Ya	Ya	Ya
%sample	Ya	Ya	Ya	Ya
$sort	Ya	Ya	Ya	Ya
$geoNear	Ya	Ya	Ya	Tidak
$lookup	Ya	Ya	Ya	Ya
$out	Ya	Ya	Ya	Tidak
$indexStats	Ya	Ya	Ya	Ya
$facet	Tidak	Tidak	Tidak	Tidak
$bucket	Tidak	Tidak	Tidak	Tidak
$bucketAuto	Tidak	Tidak	Tidak	Tidak
$ sortir ByCount	Tidak	Tidak	Tidak	Tidak
$addFields	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
$replaceRoot	Ya	Ya	Ya	Ya
$count	Ya	Ya	Ya	Ya
$currentOp	Ya	Ya	Ya	Ya
$ daftar LocalSessions	Tidak	Tidak	Tidak	Tidak
$listSessions	Tidak	Tidak	Tidak	Tidak
$graphLookup	Tidak	Tidak	Tidak	Tidak
$ bergabung	-	-	Tidak	Tidak
$ rencana CacheStats	-	-	Tidak	Tidak
$ set WindowFie lds	-	-	Tidak	Tidak
$unionDengan	-	-	Tidak	Tidak
$unset	-	-	Tidak	Tidak

Operator String

Perintah	3.6	4.0	5.0	Cluster elastis
$concat	Ya	Ya	Ya	Ya
$ indeks OfBytes	Ya	Ya	Ya	Ya
$indexOfCP	Ya	Ya	Ya	Ya
$ltrim	Tidak	Tidak	Tidak	Tidak
$rtrim	Tidak	Tidak	Tidak	Tidak

Perintah	3.6	4.0	5.0	Cluster elastis
$split	Ya	Ya	Ya	Ya
$strcasecmp	Ya	Ya	Ya	Ya
$ str LenBytes	Ya	Ya	Ya	Ya
$strLenCP	Ya	Ya	Ya	Ya
$substr	Ya	Ya	Ya	Ya
$substrBytes	Ya	Ya	Ya	Ya
$substrCP	Ya	Ya	Ya	Ya
$toLower	Ya	Ya	Ya	Ya
$toUpper	Ya	Ya	Ya	Ya
$trim	Tidak	Tidak	Tidak	Tidak
$regxFind	-	-	Tidak	Tidak
$ regex FindAll	-	-	Tidak	Tidak
$RegexMatch	-	-	Tidak	Tidak
$ GantiSatu	-	-	Tidak	Tidak
$ GantiSemua	-	-	Tidak	Tidak

Variabel sistem

Perintah	3.6	4.0	5.0	Cluster elastis
$$CURRENT	Tidak	Tidak	Tidak	Tidak
$$DESCEND	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
$$KEEP	Ya	Ya	Ya	Ya
$$PRUNE	Ya	Ya	Ya	Ya
$$REMOVE	Tidak	Tidak	Tidak	Tidak
$$ROOT	Ya	Ya	Ya	Ya

Operator Pencarian Teks

Perintah	3.6	4.0	5.0	Cluster elastis
$ pencarian	Tidak	Tidak	Ya	Tidak
$meta	Tidak	Tidak	Ya	Tidak

Operator Konversi Tipe

Perintah	3.6	4.0	5.0	Cluster elastis
$convert	Tidak	Ya	Ya	Ya
$toBool	Tidak	Ya	Ya	Ya
$toDate	Tidak	Ya	Ya	Ya
$toDecimal	Tidak	Ya	Ya	Ya
$toDouble	Tidak	Ya	Ya	Ya
$toInt	Tidak	Ya	Ya	Ya
$toLong	Tidak	Ya	Ya	Ya
$ untuk ObjectId	Tidak	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
$toString	Tidak	Ya	Ya	Ya
$ isNumber	-	-	Tidak	Tidak

Operator variabel

Perintah	3.6	4.0	5.0	Cluster elastis
$map	Ya	Ya	Ya	Ya
$let	Ya	Ya	Ya	Ya

Operator Lain-lain

Perintah	3.6	4.0	5.0	Cluster elastis
$ rand	-	-	Tidak	Tidak
$ Samplerate	-	-	Tidak	Tidak
$ GetField	-	-	Tidak	Tidak

Jenis Data

Perintah	3.6	4.0	5.0	Cluster elastis
Ganda	Ya	Ya	Ya	Ya
Tali	Ya	Ya	Ya	Ya
Objek	Ya	Ya	Ya	Ya
Array	Ya	Ya	Ya	Ya

Perintah	3.6	4.0	5.0	Cluster elastis
Data Biner	Ya	Ya	Ya	Ya
ObjectId	Ya	Ya	Ya	Ya
Boolean	Ya	Ya	Ya	Ya
Tanggal	Ya	Ya	Ya	Ya
Null	Ya	Ya	Ya	Ya
32-bit Integer (int)	Ya	Ya	Ya	Ya
Stempel Waktu	Ya	Ya	Ya	Ya
64-bit Integer (long)	Ya	Ya	Ya	Ya
MinKey	Ya	Ya	Ya	Ya
MaxKey	Ya	Ya	Ya	Ya
Decimal128	Ya	Ya	Ya	Ya
Ekspresi reguler	Ya	Ya	Ya	Ya
JavaScript	Tidak	Tidak	Tidak	Tidak
JavaScript(dengan ruang lingkup)	Tidak	Tidak	Tidak	Tidak
Tidak terdefinisi	Tidak	Tidak	Tidak	Tidak
Simbol	Tidak	Tidak	Tidak	Tidak
DBPointer	Tidak	Tidak	Tidak	Tidak

Indeks dan Properti Indeks

Topik

- Indeks
- Properti Indeks

Indeks

Perintah	3.6	4.0	5.0	Cluster elastis
Indeks Bidang Tunggal	Ya	Ya	Ya	Ya
Indeks Gabungan	Ya	Ya	Ya	Ya
Indeks Multikey	Ya	Ya	Ya	Ya
Indeks Teks	Tidak	Tidak	Ya	Tidak
2dsphere	Ya	Ya	Ya	Tidak
Indeks 2d	Tidak	Tidak	Tidak	Tidak
Indeks hashed	Tidak	Tidak	Tidak	Tidak

Properti Indeks

Perintah	3.6	4.0	5.0	Cluster elastis
TTL	Ya	Ya	Ya	Ya
Unik	Ya	Ya	Ya	Ya
Sebagian	Tidak	Tidak	Ya	Tidak

Perintah	3.6	4.0	5.0	Cluster elastis
Tidak peka huruf besar/kecil	Tidak	Tidak	Tidak	Tidak
Jarang	Ya	Ya	Ya	Ya
Latar Belakang	Ya	Ya	Ya	Tidak

Amazon DocumentDB kecerdasan buatan generatif

Amazon DocumentDB menawarkan kemampuan untuk memungkinkan model pembelajaran mesin (ML) dan kecerdasan buatan generatif (AI) untuk bekerja dengan data yang disimpan di Amazon DocumentDB secara real time. Pelanggan tidak lagi harus menghabiskan waktu mengelola infrastruktur terpisah, menulis kode untuk terhubung dengan layanan lain, dan menduplikasi data dari database utama mereka.

Untuk informasi lebih lanjut tentang kecerdasan buatan dan bagaimana AWS dapat mendukung kebutuhan AI Anda, lihat artikel "Apa itu" ini.

Topik

- Pembelajaran mesin tanpa kode dengan Amazon Canvas SageMaker
- Pencarian vektor untuk Amazon DocumentDB

Pembelajaran mesin tanpa kode dengan Amazon Canvas SageMaker

Amazon SageMaker Canvas memungkinkan Anda untuk membangun model AI/ML Anda sendiri tanpa harus menulis satu baris kode pun. Anda dapat membuat model ML untuk kasus penggunaan umum seperti regresi dan peramalan dan dapat mengakses dan mengevaluasi model pondasi (FM) dari Amazon Bedrock. Anda juga dapat mengakses FM publik dari Amazon SageMaker JumpStart untuk pembuatan konten, ekstraksi teks, dan ringkasan teks untuk mendukung solusi AI generatif.

Cara membuat model MLtanpa kode dengan Canvas SageMaker

Amazon DocumentDB sekarang terintegrasi dengan SageMaker Amazon Canvas untuk mengaktifkan pembelajaran mesin tanpa kode (ML) dengan data yang disimpan di Amazon DocumentDB. Anda sekarang dapat membangun model ML untuk kebutuhan regresi dan peramalan dan menggunakan model dasar untuk ringkasan dan pembuatan konten menggunakan data yang disimpan di Amazon DocumentDB tanpa menulis satu baris kode pun.

SageMaker Canvas menyediakan antarmuka visual yang memungkinkan pelanggan Amazon DocumentDB menghasilkan prediksi tanpa memerlukan keahlian AI/ML atau menulis satu baris kode. Pelanggan sekarang dapat meluncurkan ruang kerja SageMaker Canvas dari AWS Management Console, mengimpor, dan bergabung dengan data Amazon DocumentDB untuk persiapan data

dan pelatihan model. Data di Amazon DocumentDB sekarang dapat digunakan SageMaker di Canvas untuk membangun dan menambah model untuk memprediksi churn pelanggan, mendeteksi penipuan, memprediksi kegagalan pemeliharaan, memperkirakan metrik bisnis, dan menghasilkan konten. Pelanggan sekarang dapat mempublikasikan dan berbagi wawasan berbasis ML di seluruh tim menggunakan integrasi asli SageMaker Canvas dengan Amazon. QuickSight Saluran konsumsi data di SageMaker Canvas berjalan di instans sekunder Amazon DocumentDB secara default, memastikan bahwa kinerja aplikasi dan beban kerja penyerapan Canvas tidak terhalang. SageMaker

Pelanggan Amazon DocumentDB dapat memulai SageMaker dengan Canvas dengan menavigasi ke halaman Amazon DocumentDB No-Code MLConsole baru dan menghubungkan ke ruang kerja Canvas baru atau yang tersedia. SageMaker

Mengkonfigurasi SageMaker domain dan profil pengguna

Anda dapat terhubung ke cluster Amazon DocumentDB SageMaker dari domain yang berjalan dalam mode VPC Only. Dengan meluncurkan SageMaker domain di VPC, Anda dapat mengontrol aliran data dari lingkungan SageMaker Studio dan Canvas Anda. Ini memungkinkan Anda untuk membatasi akses internet, memantau dan memeriksa lalu lintas menggunakan AWS jaringan standar dan kemampuan keamanan, dan terhubung ke AWS sumber daya lain melalui titik akhir VPC. Silakan merujuk ke Amazon SageMaker Canvas Memulai dan Mengonfigurasi SageMaker Kanvas Amazon di VPC tanpa akses internet yang terletak di Panduan SageMaker Pengembang Amazon untuk membuat SageMaker domain Anda agar terhubung ke klaster Amazon DocumentDB Anda.

Mengkonfigurasi izin akses IAM untuk Amazon DocumentDB dan Canvas SageMaker

Pengguna Amazon DocumentDB yang `AmazonDocDBConsoleFullAccess` telah melekat pada peran dan identitas terkait mereka dapat mengakses file. AWS Management Console Tambahkan tindakan berikut ke peran atau identitas yang disebutkan di atas untuk menyediakan akses ke pembelajaran mesin tanpa kode dengan Amazon SageMaker Canvas.

```
"sagemaker:CreatePresignedDomainUrl",
"sagemaker:DescribeDomain",
"sagemaker:ListDomains",
"sagemaker:ListUserProfiles"
```

Membuat pengguna database dan peran untuk SageMaker Canvas

Anda dapat membatasi akses ke tindakan yang dapat dilakukan pengguna pada database menggunakan kontrol akses berbasis peran (RBAC) di Amazon DocumentDB. RBAC bekerja dengan memberikan satu atau lebih peran kepada pengguna. Peran ini menentukan operasi yang dapat dilakukan pengguna pada sumber daya database.

Sebagai pengguna Canvas, Anda terhubung ke database Amazon DocumentDB dengan kredensi nama pengguna dan kata sandi. Anda dapat membuat pengguna/peran database untuk pengguna Canvas yang memiliki akses baca ke database tertentu menggunakan fungsionalitas Amazon DocumentDBB RBAC.

Misalnya, gunakan `createUser` operasi:

```
db.createUser({
user: "canvas_user",
pwd: "<insert-password>",
roles: [{role: "read", db: "sample-database-1"}]
})
```

Ini menciptakan `canvas_user` yang telah membaca izin ke `sample-database-1` database. Analis Canvas Anda dapat menggunakan kredensi ini untuk mengakses data di klaster Amazon DocumentDB Anda. Lihat Akses Database Menggunakan Kontrol Akses Berbasis Peran untuk mempelajari lebih lanjut.

Wilayah yang tersedia

Integrasi tanpa kode tersedia di wilayah di mana Amazon DocumentDB SageMaker dan Amazon Canvas didukung. Daerah-daerah tersebut meliputi:

- us-east-1 (Virginia Utara)
- us-east-2 (Ohio)
- us-west-2 (Oregon)
- ap-northeast-1 (Tokyo)
- ap-northeast-2 (Seoul)
- ap-south-1 (Mumbai)
- ap-southeast-1 (Singapura)
- ap-southeast-2 (Sydney)

- eu-central-1 (Frankfurt am Main)
- eu-west-1 (Irlandia)

Silakan lihat SageMaker Kanvas Amazon di Panduan SageMaker Pengembang Amazon untuk ketersediaan wilayah terbaru.

Pencarian vektor untuk Amazon DocumentDB

Pencarian vektor adalah metode yang digunakan dalam pembelajaran mesin untuk menemukan titik data yang serupa dengan titik data tertentu dengan membandingkan representasi vektor mereka menggunakan metrik jarak atau kesamaan. Semakin dekat kedua vektor berada di ruang vektor, semakin mirip item yang mendasarinya. Teknik ini membantu menangkap makna semantik dari data. Pendekatan ini berguna dalam berbagai aplikasi, seperti sistem rekomendasi, pemrosesan bahasa alami, dan pengenalan gambar.

Pencarian vektor untuk Amazon DocumentDB menggabungkan fleksibilitas dan kemampuan query yang kaya dari database dokumen berbasis JSON dengan kekuatan pencarian vektor. Jika Anda ingin menggunakan data Amazon DocumentDB yang ada atau struktur data dokumen yang fleksibel untuk membangun pembelajaran mesin dan kasus penggunaan AI generatif, seperti pengalaman pencarian semantik, rekomendasi produk, personalisasi, chatbots, deteksi penipuan, dan deteksi anomali, maka pencarian vektor untuk Amazon DocumentDB adalah pilihan ideal untuk Anda. Pencarian vektor tersedia di cluster berbasis instans Amazon DocumentDB 5.0.

Topik

- Memasukkan vektor
- Membuat indeks vektor
- Mendapatkan definisi indeks
- Menanyakan vektor
- Fitur dan keterbatasan
- Praktik terbaik

Memasukkan vektor

Untuk menyisipkan vektor ke dalam database Amazon DocumentDB, Anda dapat menggunakan metode penyisipan yang ada:

Contoh

Dalam contoh berikut, kumpulan lima dokumen dalam database pengujian dibuat. Setiap dokumen mencakup dua bidang: nama produk dan penyematan vektor yang sesuai.

```
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]}
]);
```

Membuat indeks vektor

Amazon DocumentDB mendukung pengindeksan Hierarchical Navigable Small World (HNSW) dan metode pengindeksan File Terbalik dengan Kompresi Datar (iVFFlat). Indeks iVFFlat memisahkan vektor ke dalam daftar dan selanjutnya mencari subset terpilih dari daftar yang paling dekat dengan vektor kueri. Di sisi lain, indeks HNSW mengatur data vektor ke dalam grafik berlapis-lapis. Meskipun HNSW memiliki waktu pembuatan yang lebih lambat dibandingkan dengan iVFFlat, HNSW memberikan kinerja kueri dan penarikan yang lebih baik. Tidak seperti iVFFlat, HNSW tidak memiliki langkah pelatihan yang terlibat, memungkinkan indeks dihasilkan tanpa beban data awal. Untuk sebagian besar kasus penggunaan, sebaiknya gunakan tipe indeks HNSW untuk pencarian vektor.

Jika Anda tidak membuat indeks vektor, Amazon DocumentDB melakukan pencarian tetangga terdekat yang tepat, memastikan penarikan yang sempurna. Namun, dalam skenario produksi, kecepatan sangat penting. Kami merekomendasikan penggunaan indeks vektor, yang dapat memperdagangkan beberapa penarikan untuk meningkatkan kecepatan. Penting untuk dicatat bahwa menambahkan indeks vektor dapat menyebabkan hasil kueri yang berbeda.

Template

Anda dapat menggunakan `runCommand` templat `createIndex` atau templat berikut untuk membuat indeks vektor pada bidang vektor:

Using createIndex

> Pada driver tertentu, seperti mongosh dan Java, menggunakan `vectorOptions` parameter di `createIndex` dapat mengakibatkan kesalahan. Dalam kasus seperti itu, kami sarankan menggunakan`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

Pada driver tertentu, seperti mongosh dan Java, menggunakan `vectorOptions` parameter di `createIndex` dapat mengakibatkan kesalahan. Dalam kasus seperti itu, kami sarankan menggunakan`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	Persyaratan	Tipe data	Deskripsi	Nilai (s)
`name`	opsional	string	Menentukan nama indeks.	Alfanumerik
`type`	opsional		Menentukan jenis indeks.	Didukung: hnsw atau ivfflat Default: HNSW (patch mesin 3.0.4574 dan seterusnya)
`dimensions`	wajib	integer	Menentukan jumlah dimensi dalam data vektor.	Maksimal 2.000 dimensi.
`similarity`	wajib	string	Menentukan metrik jarak yang digunakan untuk perhitungan kesamaan.	• **`euclidean`** • **`cosine`** • **`dotProduct`**
`lists`	diperlukan untuk iVFFlat	integer	Menentukan jumlah cluster yang indeks iVFFlat menggunakan untuk mengelompokkan data vektor. Pengaturan yang disarankan adalah # dokumen/1000 untuk hingga 1	Minimal: 1 Maksimum: Lihat daftar per tabel jenis instans di Fitur dan keterbatasan bawah ini.

Parameter	Persyaratan	Tipe data	Deskripsi	Nilai (s)
			juta dokumen dan `sqrt(# of documents)` untuk lebih dari 1 juta dokumen.	
`m`	opsional	integer	Menentukan jumlah max koneksi untuk indeks HNSW	Default: 16 Rentang [2, 100]
`efConstruction`	opsional	integer	Menentukan ukuran daftar kandidat dinamis untuk membangun grafik untuk indeks HNSW. `efConstruction` harus lebih besar dari atau sama dengan (2 * m)	Default: 64 Rentang [4, 1000]

Penting bagi Anda untuk menetapkan nilai sub-parameter seperti `lists` untuk iVFFlat dan `m` dan `efConstruction` untuk HNSW dengan tepat karena akan memengaruhi akurasi/penarikan, waktu pembuatan, dan kinerja pencarian Anda. Nilai daftar yang lebih tinggi meningkatkan kecepatan kueri karena mengurangi jumlah vektor di setiap daftar, menghasilkan wilayah yang lebih kecil. Namun, ukuran wilayah yang lebih kecil dapat menyebabkan lebih banyak kesalahan penarikan, menghasilkan akurasi yang lebih rendah. Untuk HNSW, meningkatkan nilai `m` dan `efConstruction` meningkatkan akurasi, tetapi juga meningkatkan waktu dan ukuran pembuatan indeks. Lihat contoh berikut:

Contoh

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
    }
  }
)
```

Mendapatkan definisi indeks

Anda dapat melihat detail indeks Anda, termasuk indeks vektor, menggunakan perintah: `getIndexes`

Contoh

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

Contoh keluaran

```
[
```

```
 {
  "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"
 }
]
```

Menanyakan vektor

Templat kueri vektor

Gunakan template berikut untuk menanyakan vektor:

```
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	Persyaratan	Jenis	Deskripsi	Nilai (s)
vectorSearch	wajib	operator	Digunakan di dalam perintah $search untuk query vektor.	
vector	wajib	array	Menunjukkan vektor kueri yang akan digunakan untuk menemukan vektor serupa.	
path	wajib	string	Mendefinisikan nama bidang vektor.	
k	wajib	integer	Menentukan jumlah hasil yang pencarian kembali.	
similarity	wajib	string	Menentukan metrik jarak yang digunakan untuk perhitungan kesamaan.	• **euclidean** • **cosine** • **dotProduct**
probes	opsional	integer	Jumlah cluster yang Anda inginkan pencarian vektor untuk diperiksa. Nilai yang lebih tinggi memberika	Default: 1

Parameter	Persyaratan	Jenis	Deskripsi	Nilai (s)
			n penarikan yang lebih baik dengan mengorbankan kecepatan. Ini dapat diatur ke jumlah daftar untuk pencarian tetangga terdekat yang tepat (di mana perencana tidak akan menggunakan indeks). Pengaturan yang disarankan untuk memulai fine-tuning adalah. `sqrt(# of lists)`	

Parameter	Persyaratan	Jenis	Deskripsi	Nilai (s)
`efSearch`	opsional	integer	Menentukan ukuran daftar kandidat dinamis yang digunakan indeks HNSW selama pencarian. Nilai yang lebih tinggi `efSearch` memberikan penarikan yang lebih baik dengan mengorbankan kecepatan.	Default: 40 Rentang [1, 1000]

Penting untuk menyempurnakan nilai `efSearch` (HNSW) atau `probes` (IVFlat) untuk mencapai kinerja dan akurasi yang Anda inginkan. Lihat contoh operasi berikut:

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
      }
    }
  }
]);
```

Contoh keluaran

Keluaran dari operasi ini terlihat seperti berikut ini:

```
{ "_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 ] }
```

Fitur dan keterbatasan

Kompatibilitas versi

- Pencarian vektor untuk Amazon DocumentDB hanya tersedia di cluster berbasis instans Amazon DocumentDB 5.0.

vektor

- Amazon DocumentDB dapat mengindeks vektor hingga 2.000 dimensi. Namun, hingga 16.000 dimensi dapat disimpan tanpa indeks.

Indeks

- Untuk pembuatan indeks iVFFlat, pengaturan yang disarankan untuk parameter daftar adalah jumlah dokumen/1000 hingga 1 juta dokumen dan untuk lebih dari 1 juta dokumen. `sqrt(# of documents)` Karena batas memori yang berfungsi, Amazon DocumentDB mendukung nilai maksimum tertentu dari parameter daftar tergantung pada jumlah dimensi. Untuk referensi Anda, tabel berikut memberikan nilai maksimum parameter daftar untuk vektor 500, 1000, dan 2.000 dimensi:

Jenis instans	Daftar dengan 500 dimensi	Daftar dengan 1000 dimensi	Daftar dengan 2000 dimensi
t3.med	372	257	150
r5.l	915	741	511
r5.xl	1,393	1,196	901
r5.2xl	5,460	5,230	4,788
r5.4xl	7,842	7,599	7,138
r5.8xl	11.220	10.974	10.498
r5.12xl	13,774	13,526	13,044
r5.16xl	15,943	15,694	15,208
r5.24xl	19,585	19,335	18,845

- Tidak ada opsi indeks lain seperticompound, `sparse` atau `partial` didukung dengan indeks vektor.
- Build indeks paralel tidak didukung untuk indeks HNSW. Ini hanya didukung untuk indeks iVFFlat.

kueri vektor

- Untuk kueri pencarian vektor, penting untuk menyempurnakan parameter seperti `probes` atau `efSearch` untuk hasil yang optimal. Semakin tinggi nilai `probes` atau `efSearch` parameter, semakin tinggi recall dan semakin rendah kecepatannya. Pengaturan yang disarankan untuk mulai menyempurnakan parameter probe adalah. `sqrt(# of lists)`

Praktik terbaik

Pelajari praktik terbaik untuk bekerja dengan pencarian vektor di Amazon DocumentDB. Bagian ini terus diperbarui saat praktik terbaik baru diidentifikasi.

- Pembuatan indeks Inverted File with Flat Compression (iVFFlat) melibatkan pengelompokan dan pengorganisasian titik data berdasarkan kesamaan. Oleh karena itu, agar indeks menjadi lebih efektif, kami menyarankan Anda setidaknya memuat beberapa data sebelum membuat indeks.
- Untuk kueri pencarian vektor, penting untuk menyempurnakan parameter seperti `probes` atau `efSearch` untuk hasil yang optimal. Semakin tinggi nilai `efSearch` parameter `probes` atau, semakin tinggi recall dan semakin rendah kecepatannya. Pengaturan yang disarankan untuk memulai fine tuning `probes` parameter adalah`sqrt(lists)`.

Sumber Daya

- Pencarian vektor apa posting blog baru
- Contoh kode pencarian semantik
- Sampel kode pencarian vektor Amazon DocumentDB

Migrasi ke Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan basis data terkelola sepenuhnya yang kompatibel dengan API MongoDB. Anda dapat memigrasikan data Anda ke Amazon DocumentDB dari basis data MongoDB yang berjalan di lokasi atau di Amazon Elastic Compute Cloud (Amazon EC2) menggunakan proses yang diperinci pada bagian ini.

Topik

- Memutakhirkan cluster Amazon DocumentDB Anda menggunakan AWS Database Migration Service
- Alat Migrasi
- Penemuan
- Perencanaan: Persyaratan Klaster Amazon DocumentDB
- Pendekatan Migrasi
- Sumber Migrasi
- Konektivitas Migrasi
- Pengujian
- Pengujian Performa
- Pengujian Failover
- Sumber Daya Tambahan
- Buku pedoman migrasi: MongoDB ke Amazon DocumentDB

Memutakhirkan cluster Amazon DocumentDB Anda menggunakan AWS Database Migration Service

> **Important**
>
> Amazon DocumentDB tidak mengikuti siklus hidup dukungan yang sama seperti MongoDB dan jadwal MongoDB tidak berlaku untuk Amazon DocumentDB. end-of-life Tidak ada rencana saat ini end-of-life untuk Amazon DocumentDB 3.6, dan driver, aplikasi, dan alat MongoDB 3.6 Anda yang ada, akan terus bekerja dengan Amazon DocumentDB.

Anda dapat memutakhirkan cluster Amazon DocumentDB Anda ke versi yang lebih tinggi dengan penggunaan downtime minimal. AWS DMS AWS DMS adalah layanan yang dikelola sepenuhnya yang memudahkan untuk bermigrasi dari versi Amazon DocumentDB yang lebih lama, database relasional, dan basis data non-relasional ke cluster Amazon DocumentDB target Anda.

Topik

- Langkah 1: Aktifkan Streaming Perubahan
- Langkah 2: Ubah Durasi Retensi Streaming Perubahan
- Langkah 3: Migrasikan Indeks Anda
- Langkah 4: Buat Instance AWS DMS Replikasi
- Langkah 5: Buat AWS DMS Endpoint Sumber
- Langkah 6: Buat Endpoint AWS DMS Target
- Langkah 7: Buat dan jalankan tugas migrasi
- Langkah 8: Mengubah titik akhir aplikasi ke cluster Amazon DocumentDB target

Langkah 1: Aktifkan Streaming Perubahan

Untuk melakukan migrasi downtime minimal, AWS DMS memerlukan akses ke aliran perubahan kluster. Streaming perubahan Amazon DocumentDB menyediakan urutan waktu kejadian pembaruan yang terjadi dalam koleksi dan basis data klaster Anda. Membaca dari aliran perubahan memungkinkan AWS DMS untuk melakukan pengambilan data perubahan (CDC) dan menerapkan pembaruan tambahan ke cluster Amazon DocumentDB target.

Untuk mengaktifkan streaming perubahan bagi semua koleksi pada basis data tertentu, lakukan autentikasi ke klaster Amazon DocumentDB Anda dengan menggunakan mongo shell dan jalankan perintah berikut:

```
db.adminCommand({modifyChangeStreams: 1,
    database: "db_name",
    collection: "",
    enable: true});
```

Langkah 2: Ubah Durasi Retensi Streaming Perubahan

Selanjutnya, ubah periode retensi streaming perubahan berdasarkan berapa lama Anda ingin mempertahankan peristiwa perubahan di streaming perubahan. Misalnya, jika Anda mengharapkan

migrasi AWS DMS klaster Amazon DocumentDB memakan waktu 12 jam, Anda harus menyetel retensi aliran perubahan ke nilai yang lebih besar dari 12 jam. Periode retensi default untuk klaster Amazon DocumentDB Anda adalah tiga jam. Anda dapat mengubah durasi retensi log aliran perubahan untuk klaster Amazon DocumentDB Anda menjadi antara satu jam dan tujuh hari menggunakan AWS Management Console atau. AWS CLI Untuk detail selengkapnya, lihat Memodifikasi Durasi Retensi Log Streaming Perubahan.

Langkah 3: Migrasikan Indeks Anda

Buat indeks yang sama pada cluster Amazon DocumentDB target yang Anda miliki di cluster Amazon DocumentDB sumber Anda. Meskipun AWS DMS menangani migrasi data, itu tidak memigrasikan indeks. Untuk memigrasikan indeks, gunakan Alat Indeks Amazon DocumentDB untuk mengekspor indeks dari cluster Amazon DocumentDB sumber. Anda bisa mendapatkan alat ini dengan membuat tiruan dari repo alat Amazon GitHub DocumentDB dan mengikuti instruksi di. `README.md` Anda dapat menjalankan alat dari instans Amazon EC2 atau AWS Cloud9 lingkungan yang berjalan di VPC Amazon yang sama dengan cluster Amazon DocumentDB Anda.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Kode berikut membuang indeks dari cluster Amazon DocumentDB sumber Anda:

```
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/
```

Setelah indeks Anda berhasil diekspor, pulihkan indeks tersebut di cluster Amazon DocumentDB target Anda. Untuk memulihkan indeks yang Anda ekspor di langkah sebelumnya, gunakan Alat Indeks Amazon DocumentDB. Perintah berikut mengembalikan indeks di cluster Amazon DocumentDB target Anda dari direktori yang ditentukan.

```
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
```

Untuk mengonfirmasi bahwa Anda memulihkan indeks dengan benar, sambungkan ke cluster Amazon DocumentDB target Anda dengan shell mongo dan daftarkan indeks untuk koleksi tertentu. Jalankan kode berikut:

```
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()
```

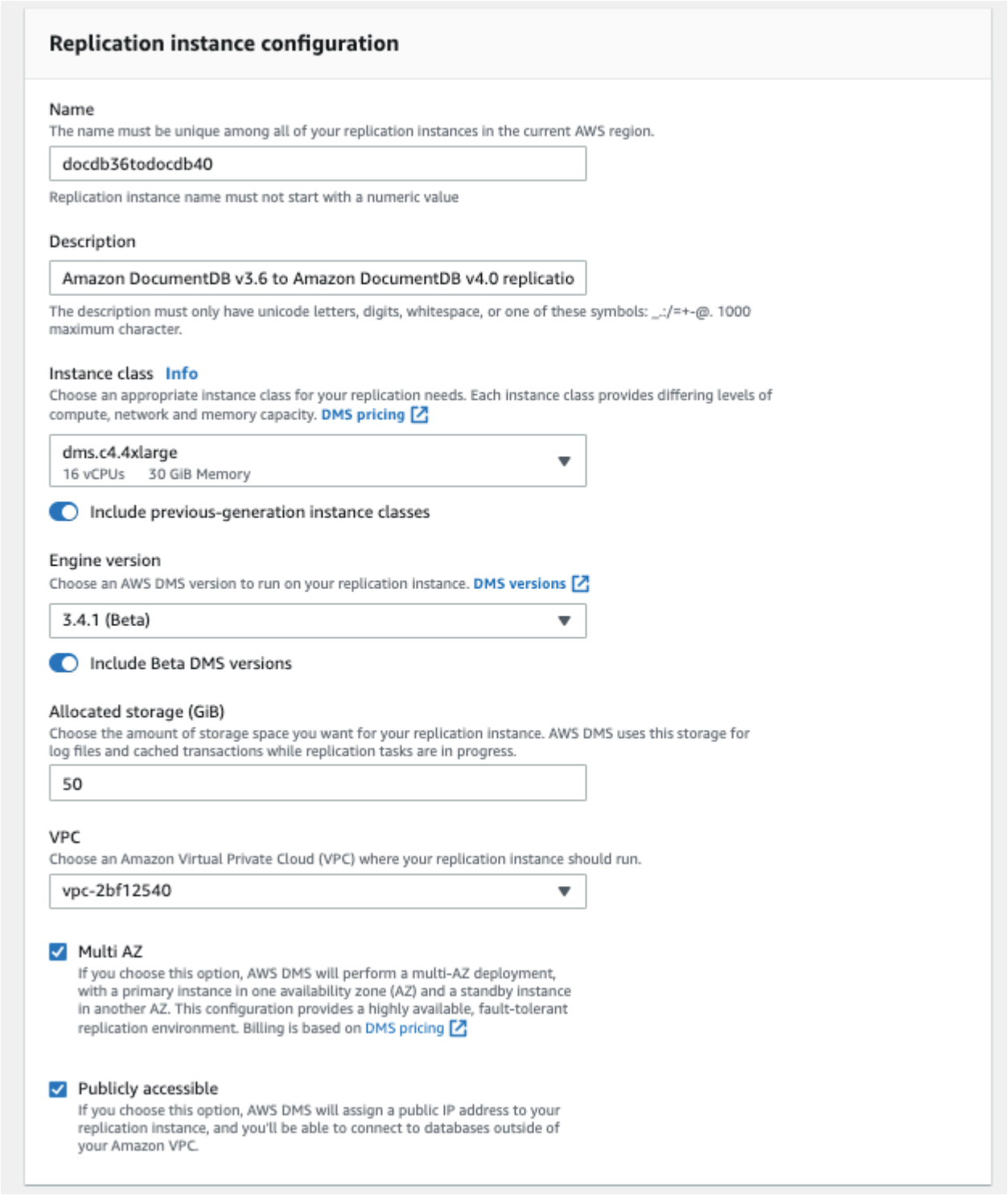
Langkah 4: Buat Instance AWS DMS Replikasi

Instans AWS DMS replikasi menghubungkan dan membaca data dari kluster Amazon DocumentDB sumber Anda dan menuliskannya sebagai klaster Amazon DocumentDB target Anda. Instance AWS DMS replikasi dapat melakukan operasi beban massal dan CDC. Sebagian besar pemrosesan ini terjadi di memori. Namun, operasi besar mungkin memerlukan buffering pada disk. Transaksi cache dan file log juga ditulis pada disk. Setelah data dimigrasikan, instans replikasi juga mengalirkan peristiwa perubahan apa pun untuk memastikan sinkronisasi antara sumber dan target.

Untuk membuat contoh AWS DMS replikasi:

1. Buka AWS DMS konsol.
2. Di panel navigasi, pilih Instans replikasi.
3. Pilih Membuat instans replikasi dan masukkan informasi berikut:
   - Untuk Nama, masukkan nama pilihan Anda. Misalnya, `docdb36todocdb40`.
   - Untuk Deskripsi, masukkan deskripsi pilihan Anda. Untuk item daftar, instans replikasi Amazon DocumentDB 3.6 ke Amazon DocumentDB 4.0.
   - Untuk Kelas instans, pilih ukuran berdasarkan kebutuhan Anda.

- Untuk Versi mesin, pilih `3.4.1.`
- Untuk Amazon VPC, pilih Amazon VPC yang menampung sumber Anda dan menargetkan cluster Amazon DocumentDB.
- Untuk Penyimpanan yang dialokasikan (GiB), gunakan default 50 GiB. Jika Anda memiliki beban kerja throughput tulis yang tinggi, tingkatkan nilai ini agar sesuai dengan beban kerja Anda.
- Untuk Multi-AZ, pilih Ya jika Anda membutuhkan ketersediaan tinggi dan dukungan failover.
- Untuk Dapat diakses publik, aktifkan opsi ini.



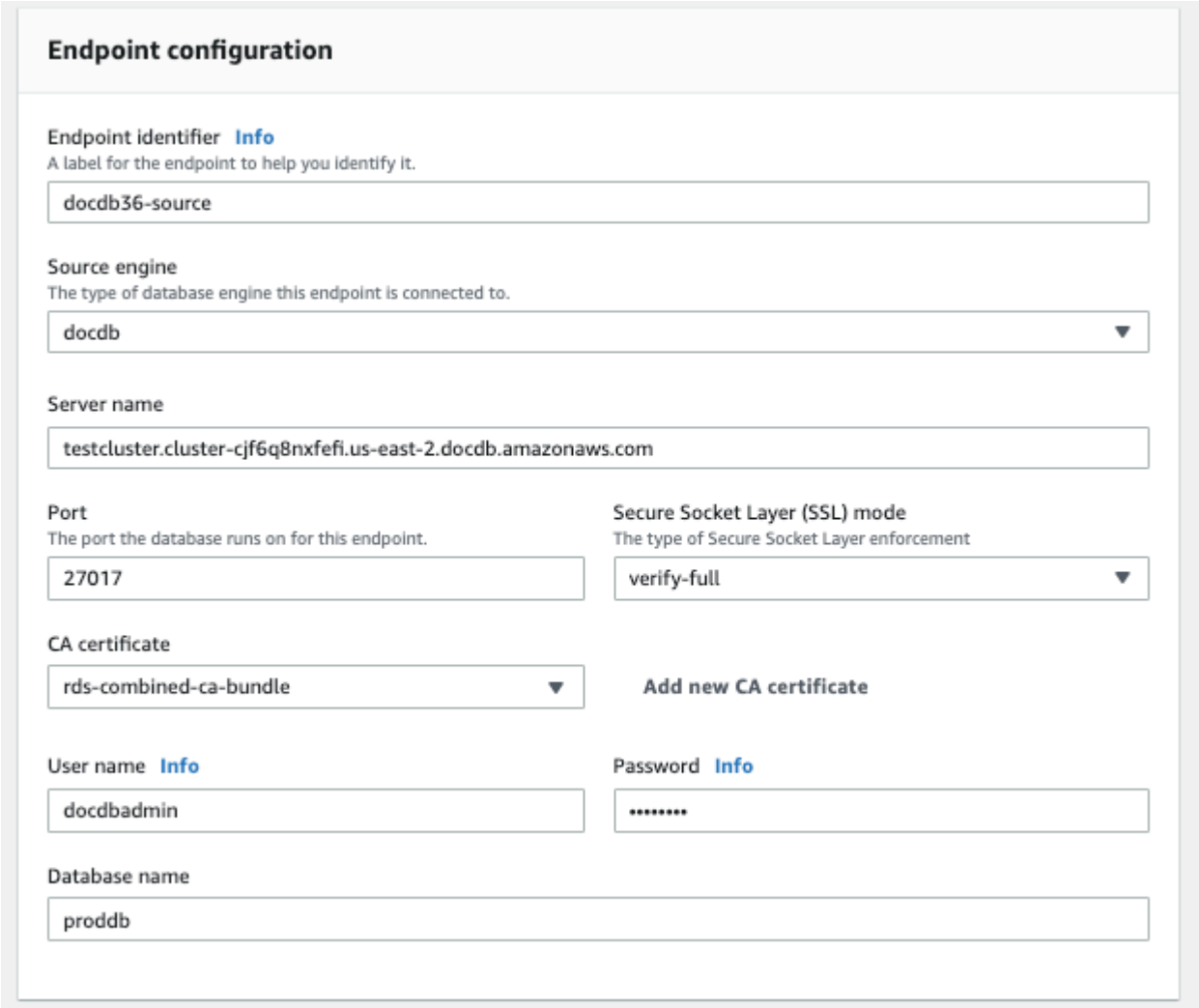
4. Pilih Buat instans replikasi.

Langkah 5: Buat AWS DMS Endpoint Sumber

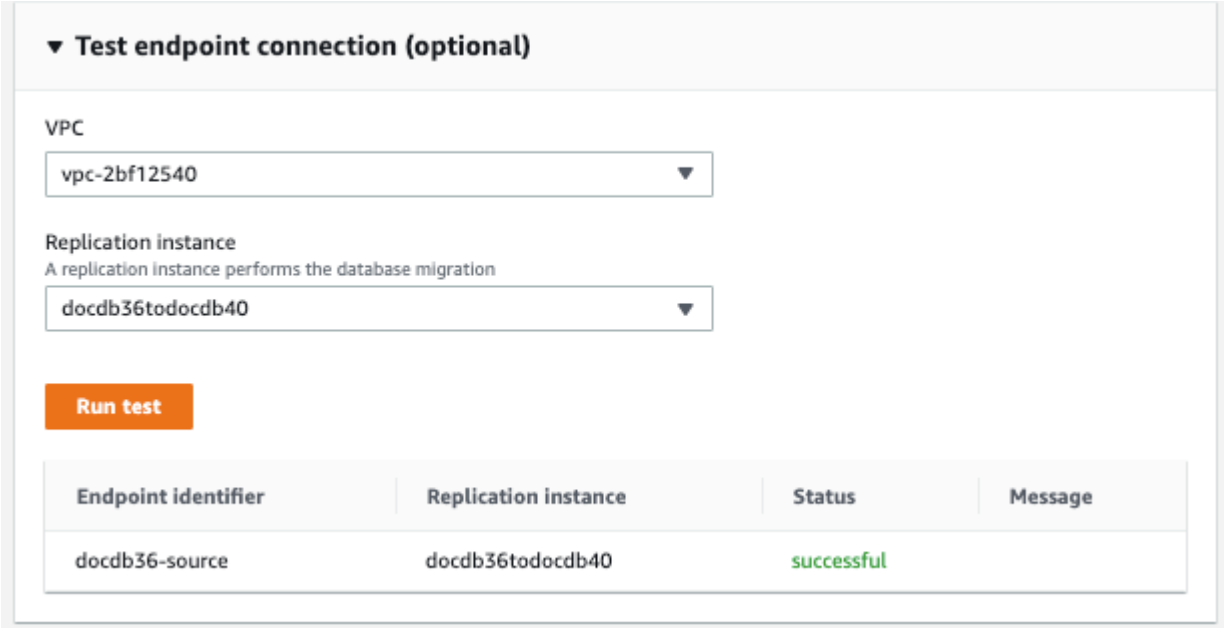
Titik akhir sumber digunakan untuk cluster Amazon DocumentDB sumber.

Untuk membuat titik akhir sumber

1. Buka AWS DMS konsol.
2. Di panel navigasi, pilih Titik akhir.
3. Pilih `Create endpoint`, dan masukkan informasi berikut:
   - Untuk Jenis titik akhir, pilih Sumber.
   - >Untuk Pengenal Titik Akhir, masukkan nama yang mudah diingat, misalnya `docdb-source`.
   - Untuk Mesin sumber, pilih `docdb`.
   - Untuk nama Server, masukkan nama DNS dari cluster Amazon DocumentDB sumber Anda.
   - Untuk Port, masukkan nomor port cluster Amazon DocumentDB sumber Anda.
   - Untuk Mode SSL, pilih `verify-full`.
   - Untuk Sertifikat CA, pilih Menambahkan sertifikat CA baru. Unduh sertifikat CA baru untuk membuat paket koneksi TLS. Untuk Pengenal sertifikat, masukkan `rds-combined-ca-bundle` . Untuk Impor file sertifikat, pilih Pilih file dan navigasikan ke `.pem` yang Anda unduh sebelumnya. Pilih dan buka file. Pilih Impor sertifikat, lalu pilih `rds-combined-ca-bundle` dari menu pilihan menurun Pilih sertifikat
   - Untuk nama Pengguna, masukkan nama pengguna utama cluster Amazon DocumentDB sumber Anda.
   - Untuk Kata Sandi, masukkan kata sandi utama cluster Amazon DocumentDB sumber Anda.
   - Untuk Nama basis data, masukkan nama basis data yang Anda cari untuk melakukan peningkatan.



4. Uji koneksi Anda untuk memverifikasi keberhasilan pengaturannya.



5. Pilih Buat Titik Akhir.

> **Note**
>
> AWS DMS hanya dapat memigrasikan satu database pada satu waktu.

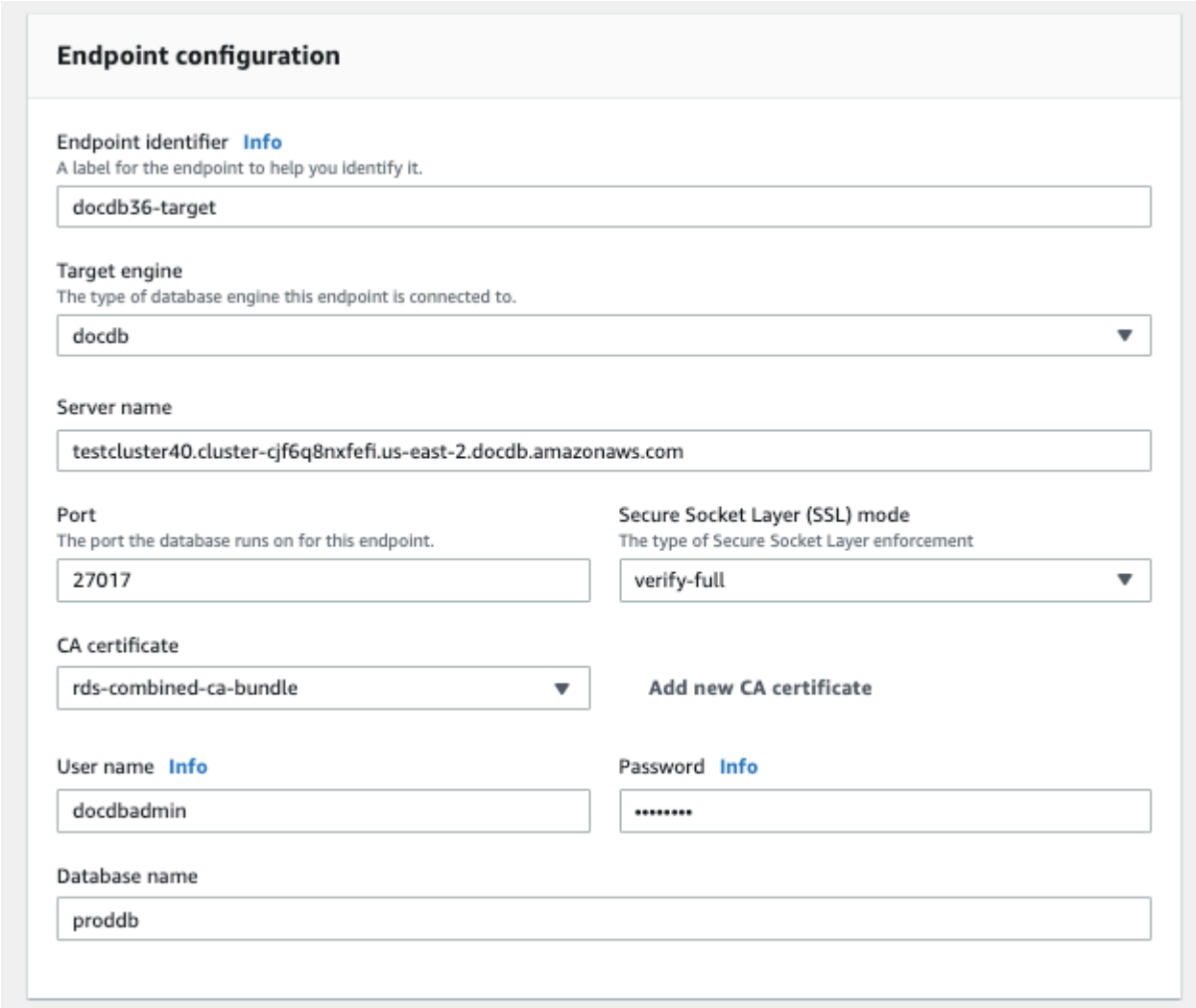
Langkah 6: Buat Endpoint AWS DMS Target

Titik akhir target adalah untuk cluster Amazon DocumentDB target Anda.

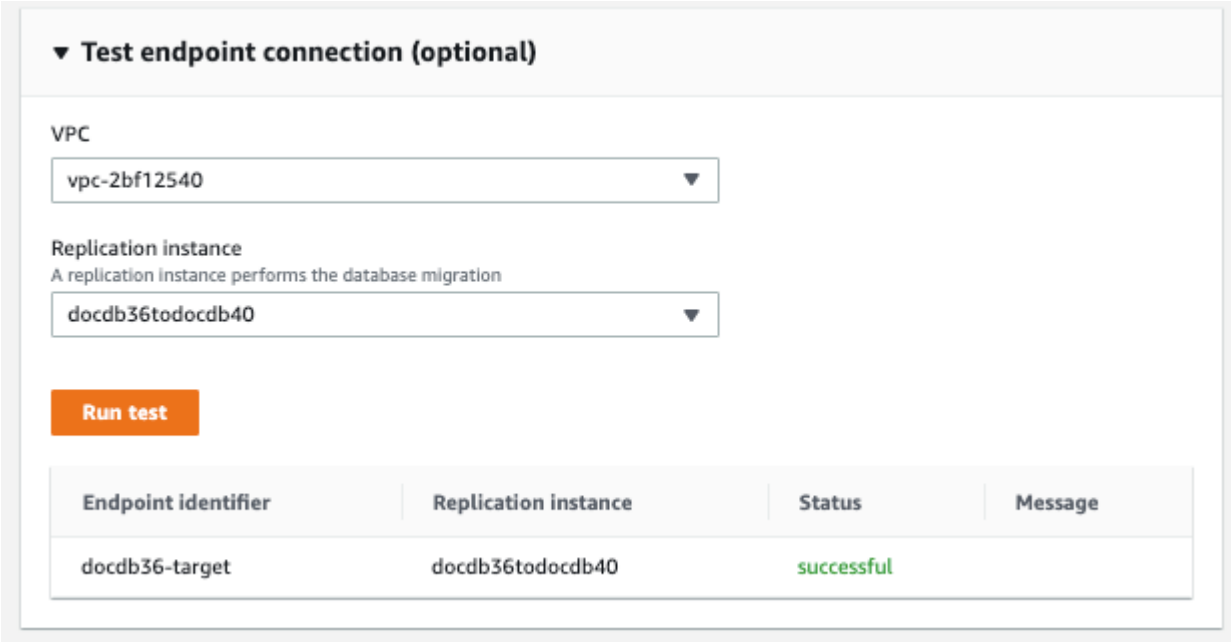
Untuk membuat titik akhir target:

1. Buka konsol AWS DMS.
2. Di panel navigasi, pilih Titik akhir.
3. Pilih Buat titik akhir dan masukkan informasi berikut:
   - Untuk Jenis titik akhir, pilih Target.
   - Untuk Pengenal Titik Akhir, masukkan nama yang mudah diingat, misalnya `docdb-target`.
   - Untuk Mesin sumber, pilih `docdb`.
   - Untuk nama Server, masukkan nama DNS kluster Amazon DocumentDB target Anda.
   - Untuk Port, masukkan nomor port cluster Amazon DocumentDB target Anda.

- Untuk Mode SSL, pilih `verify-full`.
- Untuk sertifikat CA, pilih sertifikat yang ada `rds-combined-ca-bundle` dari pilihan menurun Pilih sertifikat.
- Untuk nama Pengguna, masukkan nama pengguna utama klaster Amazon DocumentDB target Anda.
- Untuk Kata Sandi, masukkan kata sandi utama klaster Amazon DocumentDB target Anda.
- Untuk nama Basis data, masukkan nama basis data yang sama dengan yang Anda gunakan untuk mengatur titik akhir sumber Anda.



4. Uji koneksi Anda untuk memverifikasi keberhasilan pengaturannya.



5. Pilih Buat Titik Akhir.

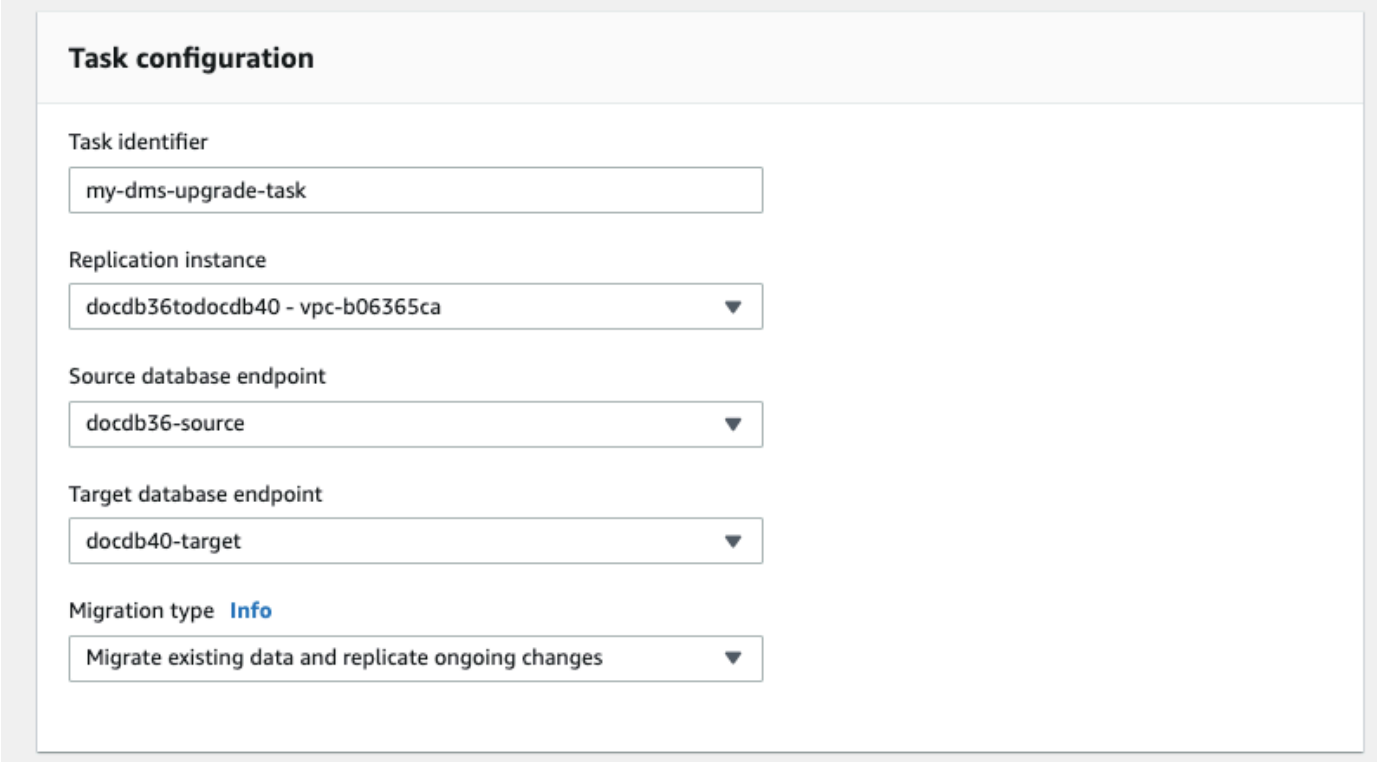
Langkah 7: Buat dan jalankan tugas migrasi

AWS DMS Tugas mengikat instance replikasi dengan instance sumber dan target Anda. Saat membuat tugas migrasi, Anda menentukan titik akhir sumber, titik akhir target, contoh replikasi, dan setelan migrasi yang diinginkan. AWS DMS Tugas dapat dibuat dengan tiga jenis migrasi yang berbeda - memigrasikan data yang ada, memigrasikan data yang ada, dan mereplikasi perubahan yang sedang berlangsung atau hanya mereplikasi perubahan data. Karena tujuan dari panduan ini adalah untuk memutakhirkan cluster Amazon DocumentDB dengan waktu henti minimal, langkah-langkahnya menggunakan opsi untuk memigrasikan data yang ada dan mereplikasi perubahan yang sedang berlangsung. Dengan opsi ini, AWS DMS menangkap perubahan saat memigrasi data yang ada. AWS DMS terus menangkap dan menerapkan perubahan bahkan setelah data massal dimuat. Terakhir, basis data sumber dan target akan disinkronkan, sehingga migrasi waktu henti yang minimal bisa didapatkan.

Berikut adalah langkah-langkah untuk membuat tugas migrasi untuk migrasi downtime minimal:

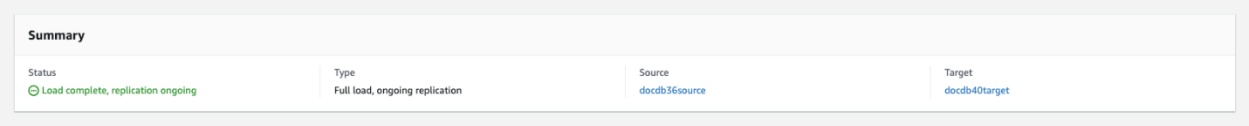
1. Buka AWS DMS konsol.
2. Di panel navigasi, pilih Tugas.
3. Pilih Buat tugas dan masukkan informasi berikut:

- Untuk pengenal Tugas, masukkan nama yang mudah diingat, misalnya`my-dms-upgrade-task`.
- Untuk contoh Replikasi, pilih instance replikasi yang Anda buat di Step3: Create an Replication Instance AWS Database Migration Service
- Untuk titik akhir basis data Sumber, pilih titik akhir sumber yang Anda buat di Langkah 4: Buat AWS Database Migration Service Titik Akhir Sumber
- Untuk titik akhir database Target, pilih titik akhir target yang Anda buat di Langkah 5: Buat Titik Akhir Target AWS Database Migration Service
- Untuk Jenis Migrasi, pilih Migrasikan data yang ada dan replikasi perubahan yang sedang berlangsung.



4. Di bagian Pengaturan Tugas, aktifkan CloudWatch log.
5. Untuk bagian Pemetaan tabel, pilih Jangan lakukan apa-apa. Ini akan memastikan bahwa indeks yang dibuat pada langkah 3 tidak akan dijatuhkan.
6. Untuk konfigurasi startup tugas Migrasi, pilih Secara otomatis saat membuat. Pilihan ini akan memulai tugas migrasi secara otomatis setelah Anda membuatnya.
7. Pilih Buat tugas.

AWS DMS sekarang mulai memigrasikan data dari cluster Amazon DocumentDB sumber Anda ke cluster Amazon DocumentDB target Anda. Status tugas harus berubah dari Mulai Menjalankan. Anda dapat memantau kemajuan dengan memilih Tugas di AWS DMS konsol. Setelah beberapa menit/jam (bergantung pada ukuran migrasi Anda), statusnya akan berubah dari menjadi Pemuatan selesai, replikasi sedang berlangsung. Ini berarti bahwa AWS DMS telah menyelesaikan migrasi muatan penuh dari cluster Amazon DocumentDB sumber Anda ke cluster Amazon DocumentDB target dan sekarang mereplikasi peristiwa perubahan.



Akhirnya sumber dan target Anda akan sinkron. Anda dapat memverifikasi apakah semuanya telah sinkron dengan menjalankan operasi `count()` pada koleksi Anda guna memverifikasi bahwa semua peristiwa perubahan telah dimigrasikan.

Langkah 8: Mengubah titik akhir aplikasi ke cluster Amazon DocumentDB target

Setelah pemuatan penuh selesai dan proses CDC mereplikasi terus menerus, Anda siap untuk mengubah titik akhir koneksi database aplikasi Anda dari cluster Amazon DocumentDB sumber Anda ke cluster Amazon DocumentDB target Anda.

Alat Migrasi

Untuk bermigrasi ke Amazon DocumentDB, dua alat utama yang digunakan sebagian besar pelanggan adalah AWS Database Migration Service (AWS DMS) dan utilitas baris perintah seperti `mongodump` dan `mongorestore`. Sebagai praktik terbaik, dan untuk salah satu dari opsi ini, Anda sebaiknya membuat indeks terlebih dahulu di Amazon DocumentDB sebelum memulai migrasi Anda karena ini dapat mengurangi keseluruhan waktu dan meningkatkan kecepatan migrasi. Untuk melakukannya, Anda dapat menggunakan Alat Indeks Amazon DocumentDB.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) adalah layanan cloud yang memudahkan migrasi database relasional dan database non-relasional ke Amazon DocumentDB. Anda dapat menggunakannya AWS DMS untuk memigrasikan data ke Amazon DocumentDB dari database yang dihosting di tempat atau di EC2. Dengan AWS DMS, Anda dapat melakukan migrasi satu kali, atau

Anda dapat mereplikasi perubahan yang sedang berlangsung untuk menjaga sumber dan target tetap sinkron.

Untuk informasi lebih lanjut tentang penggunaan AWS DMS untuk bermigrasi ke Amazon DocumentDB, silakan lihat:

- Menggunakan MongoDB sebagai Sumber untuk AWS DMS
- Menggunakan Amazon DocumentDB sebagai Target untuk AWS Database Migration Service
- Panduan: Bermigrasi dari MongoDB ke Amazon DocumentDB

Utilitas Baris Perintah

Utilitas umum untuk memigrasikan data ke dan dari Amazon DocumentDB termasuk `mongodump`, `mongorestore`, `mongoexport`, dan `mongoimport`. Biasanya, `mongodump` dan `mongorestore` merupakan utilitas paling efisien karena mereka membuang dan memulihkan data dari basis data Anda dalam format biner. Ini umumnya merupakan opsi berkinerja terbaik dan menghasilkan ukuran data yang lebih kecil dibandingkan dengan ekspor logis. `mongoexport` dan `mongoimport` berguna jika Anda ingin mengekspor dan mengimpor data dalam format logis seperti JSON atau CSV karena data dapat dibaca manusia tetapi umumnya lebih lambat daripada `mongodump`/`mongorestore` dan menghasilkan ukuran data yang lebih besar.

Pendekatan MigrasiBagian di bawah ini akan membahas kapan yang terbaik untuk menggunakan AWS DMS dan utilitas baris perintah berdasarkan kasus penggunaan dan persyaratan Anda.

Penemuan

Untuk setiap deployment MongoDB, Anda harus mengidentifikasi dan mencatat dua set data: Detail Arsitektur dan Karakteristik Operasional. Informasi ini akan membantu Anda memilih pendekatan migrasi dan ukuran klaster yang sesuai.

Detail arsitektur

- Nama

  Pilih nama unik untuk melacak deployment ini.

- Versi

Catat versi MongoDB yang sedang dijalankan oleh deployment Anda. Untuk menemukan versinya, buat sambungan ke anggota kumpulan replika dengan mongo shell dan jalankan operasi `db.version()`.

- Jenis

  Catat apakah deployment Anda adalah instans mongo mandiri, kumpulan replika, atau klaster serpihan.

- Anggota

  Catat nama host, alamat, dan port setiap klaster, kumpulan replika, atau anggota mandiri.

  Untuk klaster deployment, Anda dapat menemukan anggota serpihan setelah menghubungkan ke host mongo dengan mongo shell dan menjalankan operasi `sh.status()`.

  Untuk kumpulan replika, Anda dapat memperoleh anggota dengan membuat koneksi ke anggota kumpulan replika dengan mongo shell dan menjalankan operasi `rs.status()`.

- Ukuran oplog

  Untuk kumpulan replika atau serpihan klaster, catat ukuran oplog untuk setiap anggota kumpulan replika. Untuk menemukan ukuran oplog anggota, sambungkan ke anggota kumpulan replika dengan mongo shell dan jalankan operasi `ps.printReplicationInfo()`.

- Replika menetapkan prioritas anggota

  Untuk kumpulan replika atau serpihan klaster, catat prioritas untuk setiap anggota kumpulan replika. Untuk menemukan prioritas anggota kumpulan replika, buat koneksi ke anggota kumpulan replika dengan mongo shell dan jalankan operasi `rs.conf()`. Prioritas ditampilkan sebagai nilai kunci `priority`.

- Penggunaan TLS/SSL

  Catat apakah Transport Layer Security (TLS)/Secure Sockets Layer (SSL) digunakan pada setiap node untuk enkripsi saat transit.

Karakteristik Operasional

- Statistik basis data

  Untuk setiap koleksi, catat informasi berikut:

  - Nama
  - Ukuran data
  - Jumlah koleksi

  Untuk mencari statistik basis data, buat koneksi ke basis data Anda dengan mongo shell dan jalankan perintah `db.runCommand({dbstats: 1})`.

- Statistik koleksi

  Untuk setiap koleksi, catat informasi berikut:

  - Namespace
  - Ukuran data
  - Jumlah indeks
  - Apakah koleksi ditutup

- Statistik indeks

  Untuk setiap koleksi, catat informasi indeks berikut:

  - Namespace
  - ID
  - Size

- Kunci
- TTL
- Tersebar
- Latar Belakang

Untuk menemukan informasi indeks, buat koneksi ke basis data Anda dengan mongo shell dan jalankan perintah `db.collection.getIndexes()`.

- Opcounter

  Informasi ini membantu Anda memahami pola beban kerja MongoDB Anda saat ini (read-heavy, write-heavy, atau seimbang). Hal ini juga memberikan panduan tentang instans Amazon DocumentDB awal yang Anda pilih.

  Berikut ini adalah bagian kunci dari informasi yang perlu dikumpulkan selama periode pemantauan (dalam hitungan/detik):

  - Mengajukan Kueri
  - Menyisipkan
  - Pembaruan
  - Menghapus

  Anda dapat memeroleh informasi ini dengan membuat grafik output perintah `db.serverStatus()` dari waktu ke waktu. Anda juga dapat menggunakan alat mongostat untuk mendapatkan nilai-nilai seketika untuk statistik ini. Namun, dengan opsi ini Anda berisiko merencanakan migrasi pada periode penggunaan selain beban puncak.

- Statistik jaringan

Informasi ini membantu Anda memahami pola beban kerja MongoDB Anda saat ini (read-heavy, write-heavy, atau seimbang). Hal ini juga memberikan panduan tentang instans Amazon DocumentDB awal yang Anda pilih.

Berikut ini adalah bagian kunci dari informasi yang perlu dikumpulkan selama periode pemantauan (dalam hitungan/detik):

- Koneksi
- Byte jaringan masuk
- Byte jaringan keluar

Anda bisa mendapatkan informasi ini dengan membuat grafik output perintah `db.serverStatus()` dari waktu ke waktu. Anda juga dapat menggunakan alat mongostat untuk mendapatkan nilai-nilai seketika untuk statistik ini. Namun, dengan opsi ini Anda berisiko merencanakan migrasi pada periode penggunaan selain beban puncak.

Perencanaan: Persyaratan Klaster Amazon DocumentDB

Jika migrasi berhasil, Anda akan diminta untuk secara saksama mempertimbangkan konfigurasi klaster Amazon DocumentDB dan bagaimana aplikasi akan mengakses klaster Anda. Pertimbangkan masing-masing dimensi berikut saat menentukan persyaratan klaster Anda:

- Ketersediaan

  Amazon DocumentDB menyediakan ketersediaan tinggi melalui deployment instans replika, yang dapat dipromosikan ke instans utama dalam proses yang dikenal sebagai failover. Dengan men-deploy instans replika untuk Availability Zone yang berbeda, Anda dapat mencapai tingkat ketersediaan yang lebih tinggi.

  Tabel berikut memberikan pedoman untuk konfigurasi deployment Amazon DocumentDB guna memenuhi tujuan ketersediaan tertentu.

Target Ketersediaan	Total Instans	Replika	Zona Ketersediaan
99%	1	0	1
99,9%	2	1	2
99,99%	3	2	3

Secara keseluruhan, keandalan sistem harus mempertimbangkan semua komponen, bukan hanya basis data. Terkait praktik terbaik dan rekomendasi untuk memenuhi kebutuhan keandalan sistem secara keseluruhan, lihat AWS Laporan Resmi Pilar Keandalan Well-Architected.

- Kinerja

  Instans Amazon DocumentDB memungkinkan Anda membaca dari dan menulis ke volume penyimpanan klaster Anda. Instans klaster tersedia dalam beberapa jenis, dengan jumlah memori dan vCPU yang bervariasi, yang memengaruhi kinerja baca dan tulis pada klaster Anda. Dengan menggunakan informasi yang Anda kumpulkan dalam fase penemuan, pilih jenis instans yang dapat mendukung persyaratan performa beban kerja Anda. Untuk daftar tipe instans yang didukung, lihat Mengelola Kelas Instans.

  Saat memilih jenis instans untuk klaster Amazon DocumentDB Anda, pertimbangkan aspek berikut dari persyaratan performa beban kerja Anda:

  - vCPU—Arsitektur dengan kebutuhan koneksi lebih banyak mungkin diuntungkan jika instans memiliki lebih banyak vCPUS.

  - Memori—Bila memungkinkan, penyimpanan set data kerja Anda dalam memori akan memberikan performa maksimal. Menurut panduan awal, yang perlu dilakukan adalah mencadangkan sepertiga memori instans untuk mesin Amazon DocumentDB, dan menyisakan dua pertiga untuk set data kerja.

- Koneksi—Jumlah koneksi optimal minimum adalah delapan koneksi per vCPU instans Amazon DocumentDB. Meskipun batas koneksi instans Amazon DocumentDB jauh lebih tinggi, keuntungan performa koneksi tambahan menurun di atas delapan koneksi per vCPU.

- Jaringan—Beban kerja dengan sejumlah besar klien atau koneksi harus mempertimbangkan performa jaringan agregat yang diperlukan untuk data yang dimasukkan dan diambil. Operasi massal dapat membuat penggunaan sumber daya jaringan lebih efisien.

- Performa Sisipan—Penyisipan dokumen tunggal biasanya merupakan cara paling lambat untuk menyisipkan data ke Amazon DocumentDB. Operasi sisipan massal dapat secara dramatis lebih cepat daripada sisipan tunggal.

- Performa Baca—Membaca dari memori kerja selalu lebih cepat daripada membaca yang dikembalikan dari volume penyimpanan. Oleh karenanya, sangat ideal untuk mengoptimalkan ukuran memori instans Anda untuk mempertahankan set kerja Anda di memori.

Selain melayani pembacaan dari instans utama Anda, klaster Amazon DocumentDB secara otomatis dikonfigurasi sebagai set replika. Anda kemudian dapat merutekan kueri hanya-baca untuk membaca replika dengan menyetel preferensi baca di driver MongoDB Anda. Anda dapat menskalakan lalu lintas baca dengan menambahkan replika, sehingga mengurangi beban keseluruhan pada instans utama.

Deployment replika Amazon DocumentDB dari jenis instans yang berbeda dalam klaster yang sama dapat dilakukan. Contoh kasus penggunaan mungkin adalah pembuatan replika dengan jenis instans yang lebih besar untuk melayani lalu lintas analitik sementara. Jika Anda men-deploy kumpulan jenis instans campuran, pastikan prioritas failover telah dikonfigurasi pada setiap instans. Ini membantu memastikan bahwa peristiwa failover selalu mendukung replika dengan ukuran yang cukup untuk menangani beban tulis Anda.

- Pemulihan

  Amazon DocumentDB terus mencadangkan data Anda seperti yang tertulis. Ini memberikan kemampuan point-in-time pemulihan (PITR) dalam periode yang dapat dikonfigurasi 1-35 hari, yang dikenal sebagai periode retensi cadangan. Periode retensi cadangan default adalah satu hari. Secara otomatis, Amazon DocumentDB juga membuat snapshot harian dari volume penyimpanan Anda, yang juga disimpan selama periode retensi cadangan terkonfigurasi.

  Jika Anda ingin menyimpan snapshot di luar periode retensi cadangan, Anda juga dapat memulai snapshot manual kapan saja menggunakan AWS Management Console dan AWS Command Line Interface ().AWS CLI Untuk informasi selengkapnya, lihat Membuat Cadangan dan Memulihkan di Amazon DocumentDB.

  Pertimbangkan hal berikut saat Anda merencanakan migrasi:

  - Pilih periode retensi cadangan 1–35 hari yang sesuai dengan tujuan titik pemulihan (RPO).
  - Tentukan apakah Anda memerlukan snapshot manual, dan jika demikian, pada interval apa.

Pendekatan Migrasi

Terdapat tiga pendekatan utama untuk migrasi data Anda ke Amazon DocumentDB.

> **Note**
>
> Meskipun Anda dapat membuat indeks kapan saja di Amazon DocumentDB, secara keseluruhan lebih cepat untuk membuatnya sebelum mengimpor set data besar. Sebagai praktik terbaik, kami menyarankan agar Anda terlebih dahulu membuat indeks di Amazon DocumentDB sebelum melakukan migrasi untuk setiap pendekatan di bawah ini. Untuk melakukannya, Anda dapat menggunakan Alat Indeks Amazon DocumentDB.

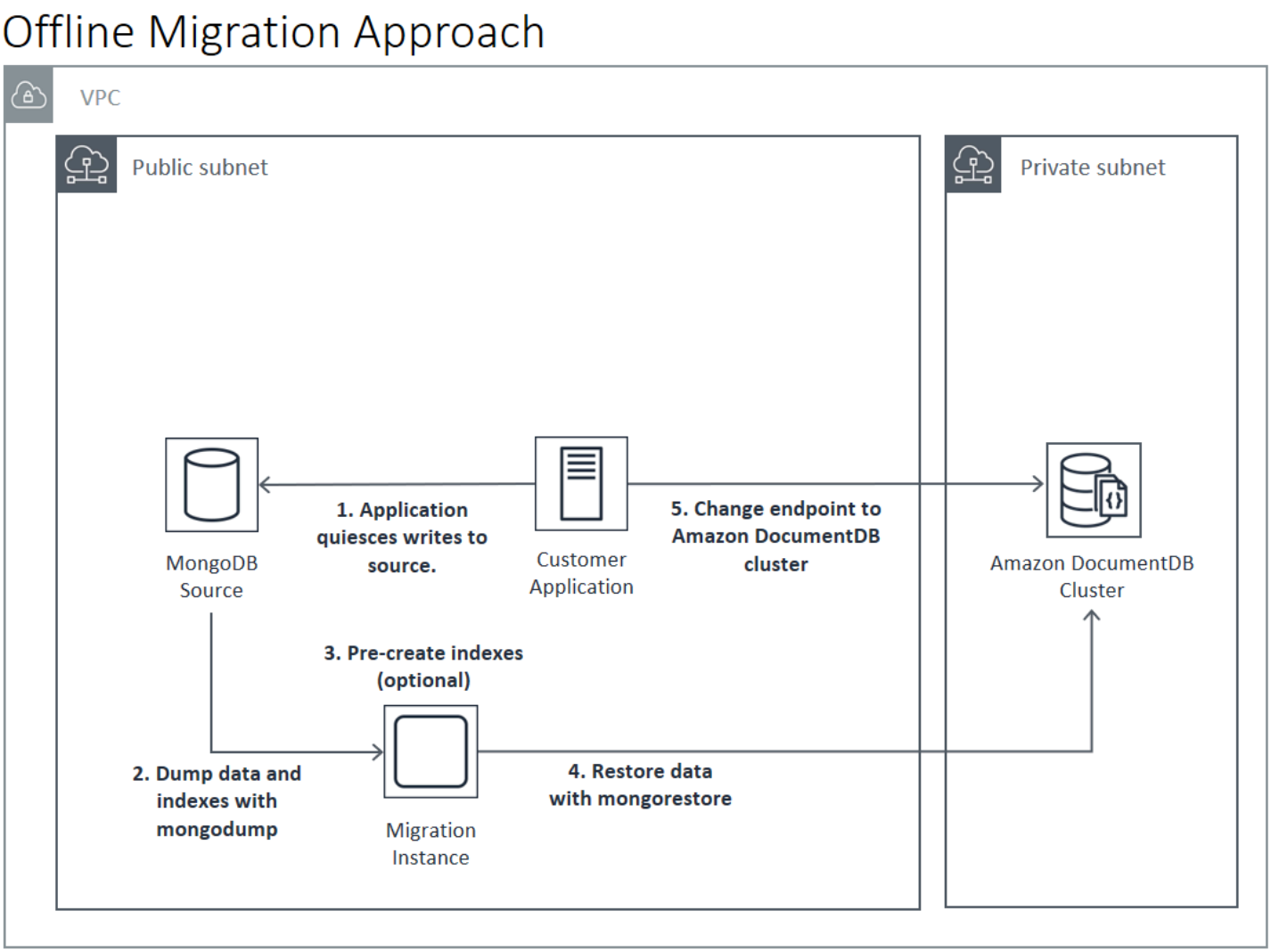
Offline

Pendekatan offline menggunakan alat `mongodump` dan `mongorestore` untuk memigrasikan data Anda dari deployment MongoDB sumber ke klaster Amazon DocumentDB Anda. Metode offline

adalah pendekatan migrasi yang paling sederhana, namun turut menimbulkan waktu henti terbanyak untuk klaster Anda.

Proses dasar untuk migrasi offline adalah sebagai berikut:

1. Quiesce menulis ke sumber MongoDB Anda.
2. Buang data pengumpulan dan indeks dari deployment MongoDB sumber.
3. Jika Anda bermigrasi ke Elastic Cluster, buat koleksi sharded Anda menggunakan perintah. `sh.shardCollection()` Jika Anda bermigrasi ke klaster berbasis instans, lewati ke langkah berikutnya.
4. Kembalikan indeks ke klaster Amazon DocumentDB.
5. Kembalikan data pengumpulan ke klaster Amazon DocumentDB.
6. Mengubah titik akhir aplikasi Anda untuk menulis ke klaster Amazon DocumentDB.



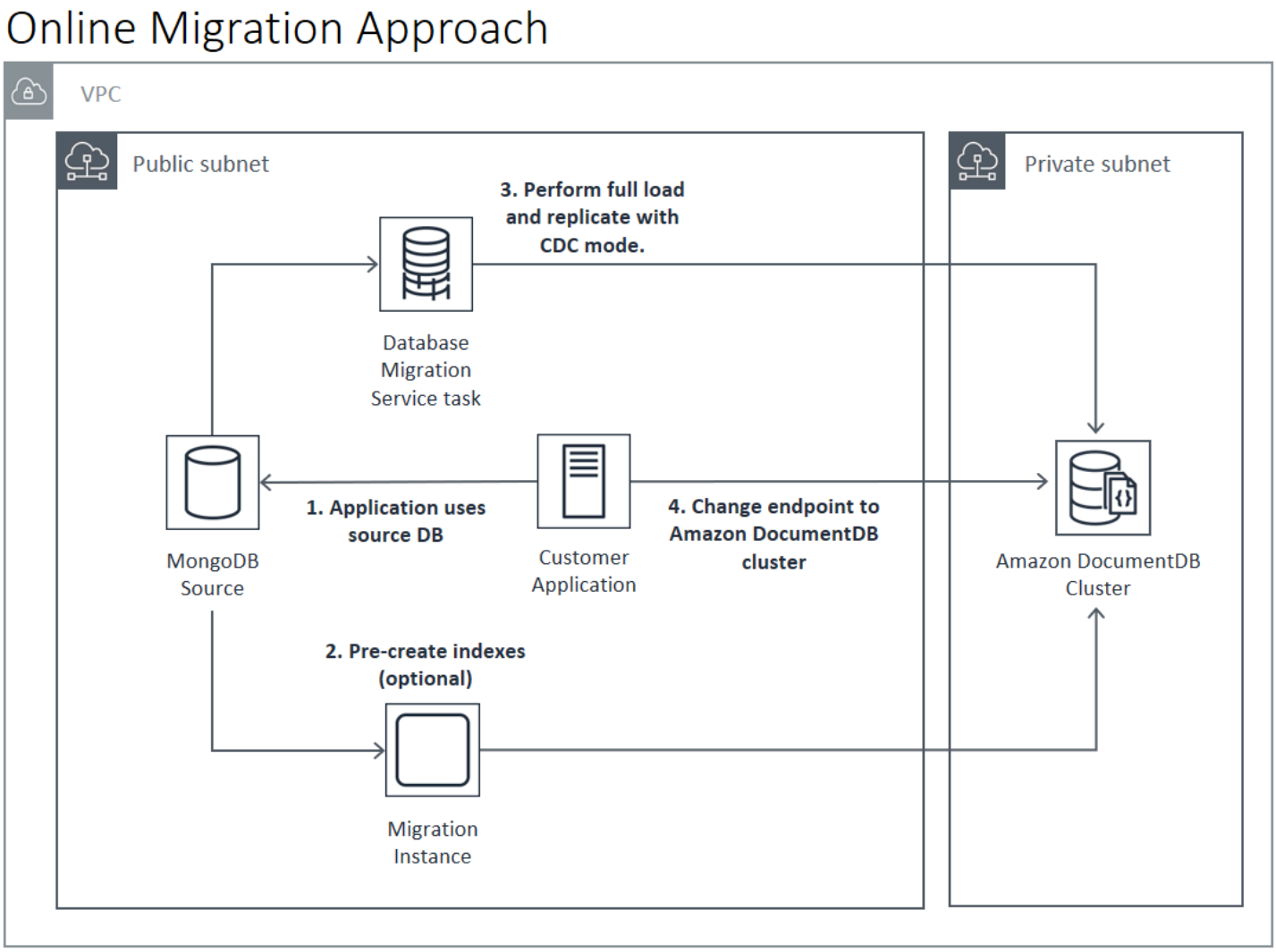
Online

Pendekatan online menggunakan AWS Database Migration Service (AWS DMS). Ini menjalankan beban penuh data dari deployment MongoDB sumber ke klaster Amazon DocumentDB Anda. Kemudian ia beralih ke mode pengambilan data perubahan (CDC) untuk mereplikasi perubahan. Pendekatan online meminimalkan waktu henti untuk klaster Anda, tetapi ini adalah metode paling lambat dari ketiga metode tersebut.

Proses dasar migrasi online adalah sebagai berikut:

1. Aplikasi Anda umumnya menggunakan DB sumber.
2. Jika Anda bermigrasi ke Elastic Cluster, buat koleksi sharded Anda menggunakan perintah. `sh.shardCollection()` Jika Anda bermigrasi ke klaster berbasis instans, lewati ke langkah berikutnya.

3. Pra-buat indeks di cluster Amazon DocumentDB.
4. Buat AWS DMS tugas untuk melakukan pemuatan penuh, lalu aktifkan CDC dari penerapan MongoDB sumber ke cluster Amazon DocumentDB.
5. Setelah AWS DMS tugas menyelesaikan beban penuh dan mereplikasi perubahan ke Amazon DocumentDB, alihkan titik akhir aplikasi ke cluster Amazon DocumentDB.



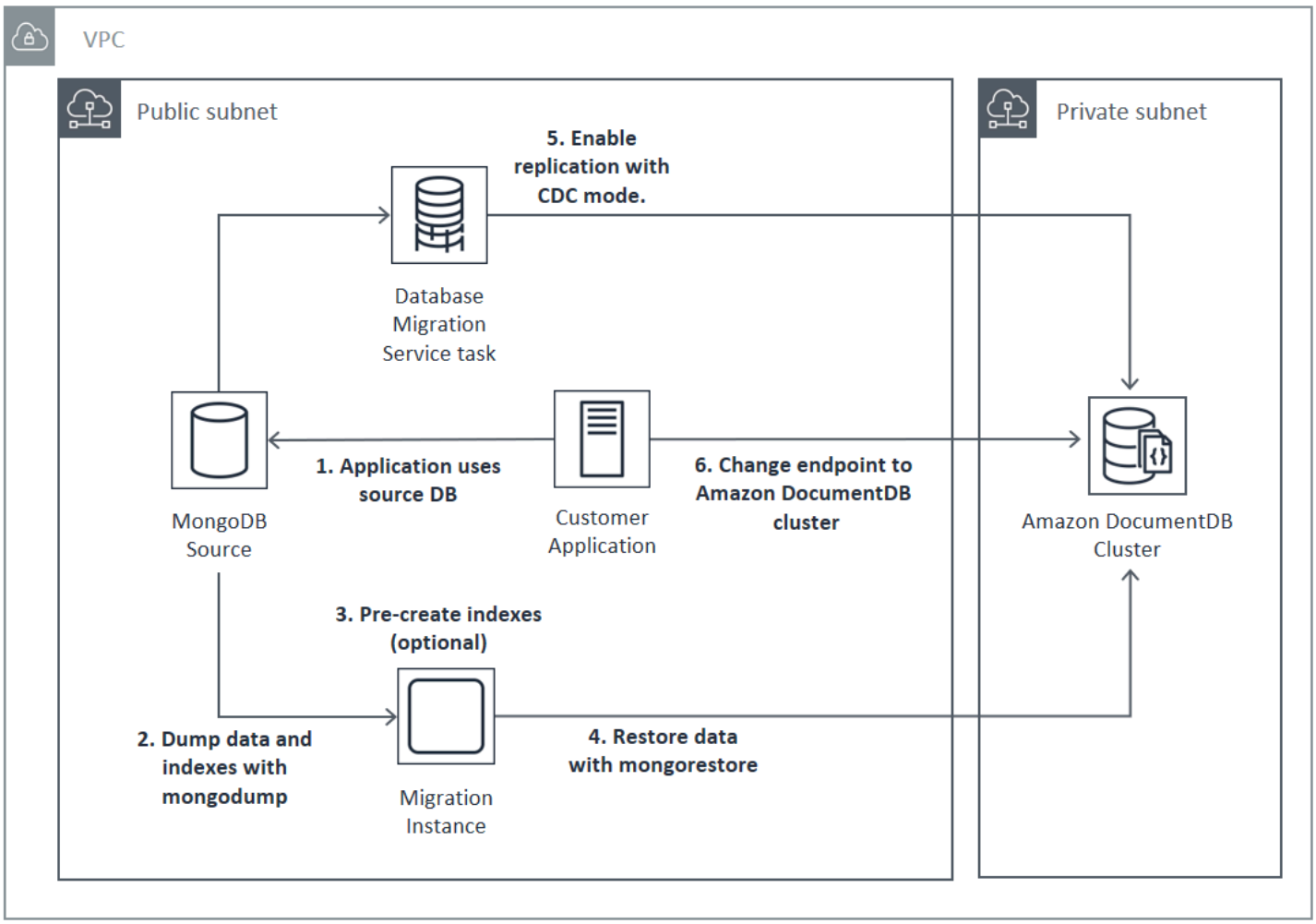
Untuk informasi selengkapnya tentang penggunaan AWS DMS untuk bermigrasi, lihat Menggunakan Amazon DocumentDB sebagai AWS Database Migration Service Target untuk dan Tutorial AWS Database Migration Service terkait di Panduan Pengguna.

Hibrida

Pendekatan hidbrid menggunakan alat `mongodump` dan `mongorestore` untuk memigrasikan data Anda dari deployment MongoDB sumber ke klaster Amazon DocumentDB. Kemudian digunakan AWS DMS dalam mode CDC untuk mereplikasi perubahan. Pendekatan hibrida menyeimbangkan kecepatan migrasi dan waktu henti, tetapi ini adalah yang paling kompleks dari tiga pendekatan tersebut.

Proses dasar untuk migrasi hibrida adalah sebagai berikut:

1. Aplikasi Anda umumnya menggunakan deployment MongoDB sumber.
2. Buang data pengumpulan dan indeks dari deployment MongoDB sumber.
3. Kembalikan indeks ke klaster Amazon DocumentDB.
4. Jika Anda bermigrasi ke Elastic Cluster, buat koleksi sharded Anda menggunakan perintah. `sh.shardCollection()` Jika Anda bermigrasi ke klaster berbasis instans, lewati ke langkah berikutnya.
5. Kembalikan data pengumpulan ke klaster Amazon DocumentDB.
6. Buat AWS DMS tugas untuk mengaktifkan CDC dari penyebaran MongoDB sumber ke cluster Amazon DocumentDB.
7. Saat AWS DMS tugas mereplikasi perubahan dalam jendela yang dapat diterima, ubah titik akhir aplikasi Anda untuk menulis ke klaster Amazon DocumentDB.



> **Important**
>
> Sebuah AWS DMS tugas saat ini hanya dapat memigrasikan satu database. Jika sumber MongoDB Anda memiliki basis data dalam jumlah besar, Anda mungkin perlu mengotomatiskan pembuatan tugas migrasi, atau mencoba menggunakan metode offline.

Terlepas dari pendekatan migrasi yang Anda pilih, yang paling efisien adalah membuat indeks terlebih dahulu di klaster Amazon DocumentDB sebelum memigrasikan data. Ini karena indeks Amazon DocumentDB adalah data yang dimasukkan secara paralel, sedangkan pembuatan indeks pada data yang ada adalah operasi utas tunggal.

Karena AWS DMS tidak memigrasikan indeks (hanya data Anda), tidak ada langkah tambahan yang diperlukan untuk menghindari pembuatan indeks untuk kedua kalinya.

Sumber Migrasi

Jika sumber MongoDB Anda adalah proses mongo mandiri dan Anda ingin menggunakan pendekatan migrasi online atau hibrid, pertama-tama konversikan mongo mandiri Anda ke kumpulan replika sehingga oplog dibuat dan digunakan sebagai sumber CDC.

Jika Anda bermigrasi dari kumpulan replika MongoDB atau serpihan klaster, cobalah membuat sekunder berantai atau tersembunyi untuk setiap kumpulan atau serpihan replika yang akan digunakan sebagai sumber migrasi Anda. Pembuangan data dapat memaksa kumpulan data yang bekerja keluar dari memori dan memengaruhi performa pada instans produksi. Anda dapat mengurangi risiko ini dengan bermigrasi dari node yang tidak melayani data produksi.

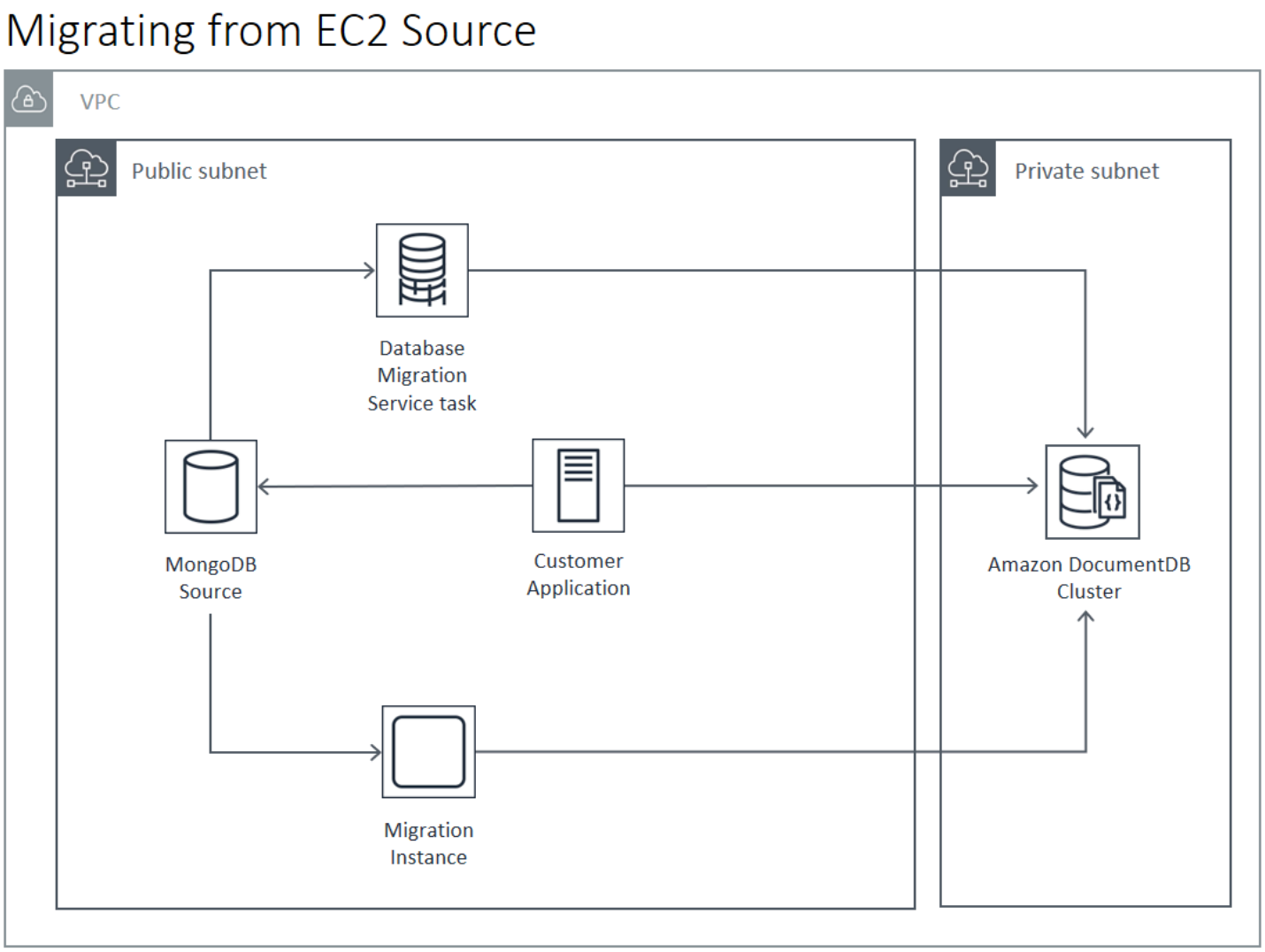
Versi Sumber Migrasi

Jika versi basis data MongoDB sumber Anda berbeda dengan versi kompatibilitas klaster Amazon DocumentDB tujuan, Anda mungkin perlu mengambil langkah persiapan lain untuk memastikan keberhasilan migrasi. Dua persyaratan yang paling umum ditemui adalah kebutuhan untuk meningkatkan penginstalan MongoDB sumber ke versi yang didukung untuk migrasi (MongoDB versi 3.0 atau lebih tinggi), dan memutakhirkan driver aplikasi Anda untuk mendukung versi Amazon DocumentDB target.

Pastikan bahwa jika migrasi Anda memiliki salah satu persyaratan ini, Anda menyertakan langkah-langkah tersebut dalam rencana migrasi Anda untuk meningkatkan dan menguji setiap perubahan driver.

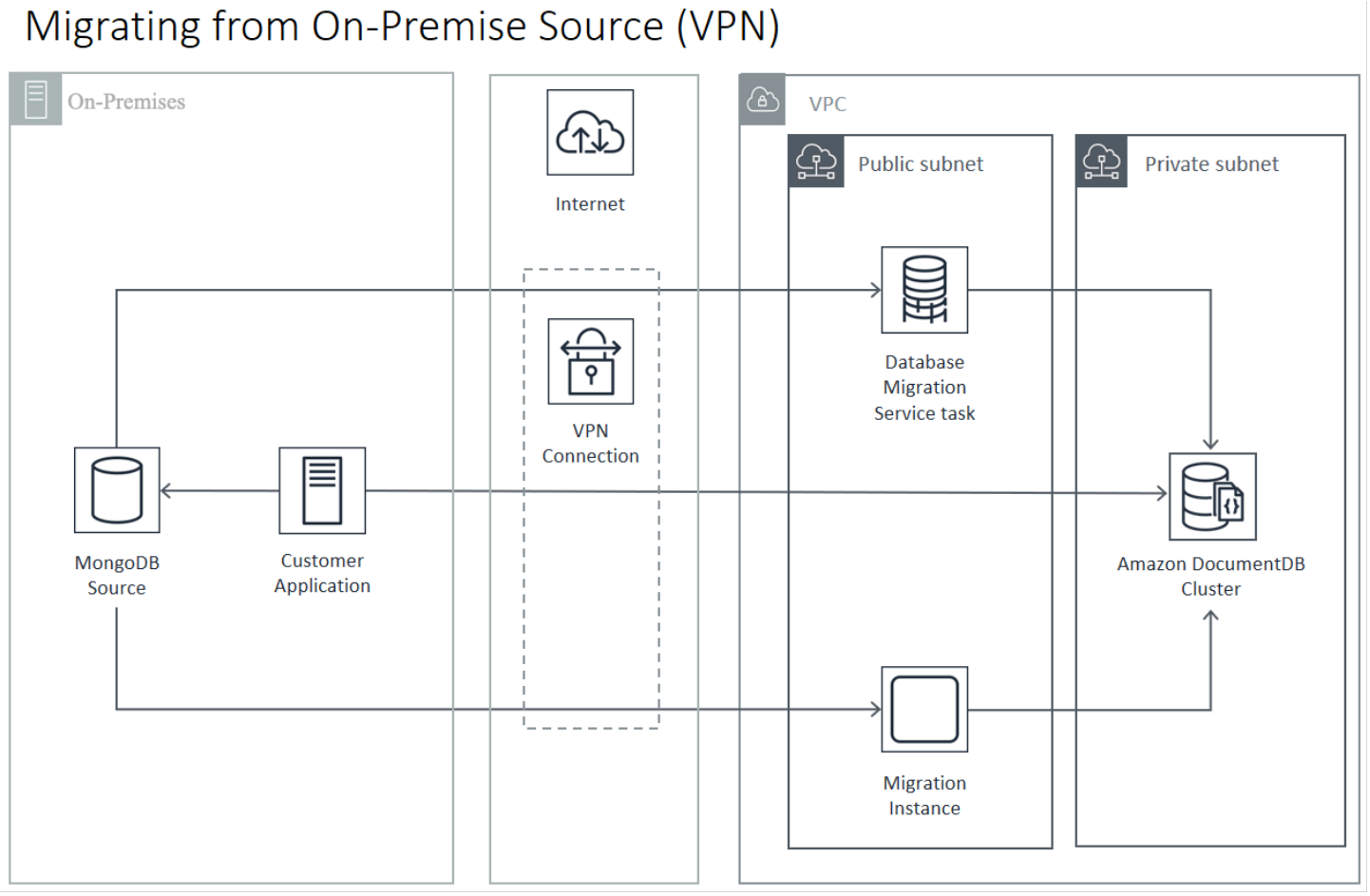
Konektivitas Migrasi

Anda dapat bermigrasi ke Amazon DocumentDB dari deployment MongoDB sumber yang beroperasi di pusat data Anda atau dari deployment MongoDB yang beroperasi pada instans Amazon EC2. Migrasi dari MongoDB yang beroperasi di EC2 sangatlah mudah, dan hanya mengharuskan Anda mengonfigurasi grup dan subnet keamanan dengan benar.

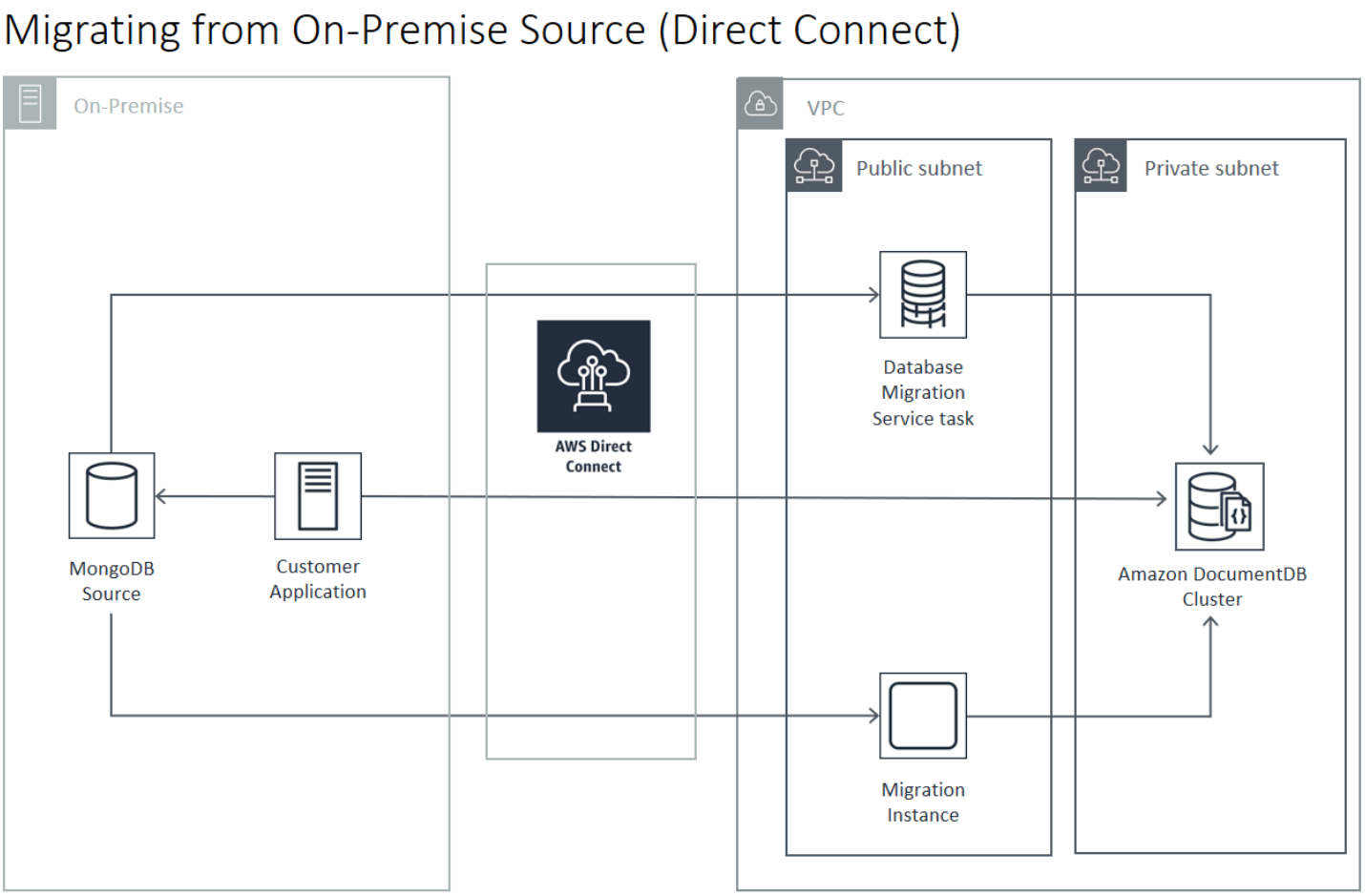


Bermigrasi dari basis data lokal memerlukan konektivitas antara deployment MongoDB dan virtual private cloud (VPC) Anda. Anda dapat melakukannya melalui koneksi jaringan pribadi virtual (VPN), atau dengan menggunakan AWS Direct Connect layanan ini. Meskipun Anda dapat bermigrasi melalui internet ke VPC Anda, metode koneksi ini paling tidak diinginkan dari sudut pandang keamanan.

Diagram berikut mengilustrasikan migrasi ke Amazon DocumentDB dari sumber lokal melalui koneksi VPN.



Berikut ini merupakan migrasi ke Amazon DocumentDB dari sumber lokal menggunakan AWS Direct Connect.



Pendekatan migrasi online dan hibrid memerlukan penggunaan instans AWS DMS , yang harus berjalan di Amazon EC2 pada Amazon VPC. Semua pendekatan memerlukan server migrasi untuk menjalankan `mongodump` dan `mongorestore`. Biasanya lebih mudah untuk menjalankan server migrasi pada instans Amazon EC2 di VPC tempat klaster Amazon DocumentDB Anda diluncurkan karena secara dramatis menyederhanakan konektivitas ke klaster Amazon DocumentDB Anda.

Pengujian

Berikut ini adalah tujuan pengujian pra-migrasi:

- Verifikasi bahwa pendekatan yang Anda pilih mencapai hasil migrasi yang Anda inginkan.
- Pastikan tipe instans Anda dan pilihan preferensi baca memenuhi persyaratan performa aplikasi Anda.
- Verifikasi perilaku aplikasi Anda selama failover.

Pertimbangan Pengujian Rencana Migrasi

Pertimbangkan hal berikut saat menguji rencana migrasi Amazon DocumentDB Anda.

Topik

- Mengembalikan Indeks
- Pembuangan Data
- Memulihkan Data
- Pengukuran Oplog
- AWS Database Migration Service Konfigurasi
- Migrasi dari Serpihan Klaster

Mengembalikan Indeks

Secara default, `mongorestore` membuat indeks untuk koleksi buangan, tetapi ia membuatnya setelah data dipulihkan. Secara keseluruhan, pembuatan indeks di Amazon DocumentDB berlangsung lebih cepat sebelum data dipulihkan ke klaster. Hal ini karena operasi pengindeksan diparalelkan selama pemuatan data.

Jika memilih untuk membuat indeks sebelumnya, Anda dapat mengabaikan langkah pembuatan indeks saat memulihkan data dengan `mongorestore` dengan menyediakan opsi `--noIndexRestore`.

Pembuangan Data

Alat `mongodump` ini adalah metode yang disukai untuk membuang data dari deployment MongoDB sumber Anda. Bergantung pada sumber daya yang tersedia pada instans migrasi, Anda mungkin dapat mempercepat `mongodump` dengan meningkatkan jumlah koneksi paralel yang dibuang dari 4 default dengan menggunakan opsi `--numParallelCollections`.

Memulihkan Data

Alat `mongorestore` ini adalah metode yang disukai untuk memulihkan data yang dibuang ke instans Amazon DocumentDB Anda. Anda dapat meningkatkan performa pemulihan dengan meningkatkan jumlah pekerja untuk setiap koleksi selama pemulihan dengan opsi `--numInsertionWorkersPerCollection`. Satu pekerja per vCPU pada instans utama klaster Amazon DocumentDB Anda adalah tempat yang baik untuk memulai.

Amazon DocumentDB saat ini tidak mendukung opsi `mongorestore` alat `--oplogReplay`.

Secara default, `mongorestore` mengabaikan kesalahan penyisipan dan melanjutkan proses pemulihan. Hal ini dapat terjadi jika Anda memulihkan data yang tidak didukung untuk instans Amazon DocumentDB Anda. Misalnya, hal itu bisa terjadi jika Anda memiliki dokumen yang berisi kunci atau nilai dengan string null. Jika Anda lebih memilih menggagalkan operasi `mongorestore` sepenuhnya saat terjadi kesalahan pemulihan, gunakan opsi `--stopOnError`.

Pengukuran Oplog

Log operasi MongoDB (`oplog`) adalah kumpulan terbatas yang berisi semua modifikasi data ke basis data Anda. Anda dapat melihat ukuran oplog dan rentang waktu di dalamnya dengan menjalankan operasi `db.printReplicationInfo()` pada set replika atau anggota serpihan.

Jika Anda menggunakan pendekatan online atau hibrida, pastikan oplog pada setiap set replika atau pecahan cukup besar untuk menampung semua perubahan yang dibuat selama seluruh durasi proses migrasi data (baik melalui `mongodump` atau beban penuh AWS DMS tugas), ditambah buffer yang masuk akal. Untuk informasi selengkapnya, lihat Memeriksa Ukuran Oplog di dokumentasi MongoDB. Tentukan ukuran oplog minimum yang diperlukan dengan mencatat waktu yang telah berlalu selama pengujian pertama pada proses `mongodump` atau `mongorestore` atau tugas beban penuh AWS DMS .

AWS Database Migration Service Konfigurasi

AWS Database Migration Service Panduan Pengguna mencakup komponen dan langkah-langkah yang diperlukan untuk memigrasikan data sumber MongoDB Anda ke klaster Amazon DocumentDB Anda. Berikut ini adalah proses dasar yang digunakan AWS DMS untuk melakukan migrasi online atau hibrida:

Untuk melakukan migrasi menggunakan AWS DMS

1. Membuat titik akhir sumber MongoDB. Untuk informasi selengkapnya, lihat Menggunakan MongoDB sebagai Sumber untuk AWS DMS.
2. Buat titik akhir target Amazon DocumentDB. Untuk informasi selengkapnya, lihat Bekerja dengan AWS DMS Titik Akhir.

   Jika Anda mengonfigurasi titik akhir target sebagai cluster elastis, perhatikan bahwa sertifikat SSL Amazon DocumentDB yang ada tidak akan berfungsi dengan cluster elastis dan Anda harus melampirkan sertifikat SSL baru ke titik akhir Anda menggunakan langkah-langkah berikut:

sebuah. Kunjungi https://www.amazontrust.com/repository/SFSRootCAG2.pem dan simpan konten sebagai file “SFSRootCag2.pem”. Ini adalah file sertifikat yang perlu Anda impor pada langkah selanjutnya.

b. Saat membuat titik akhir cluster elastis, di bawah Konfigurasi Titik Akhir, pilih Tambahkan sertifikat CA baru.

- Untuk pengenal Sertifikat, masukkan`SFSRootCAG2.pem`.
- Untuk Impor file sertifikat, pilih Pilih file dan navigasikan ke `SFSRootCAG2.pem` yang Anda unduh sebelumnya. Pilih dan buka file. Pilih Impor sertifikat, lalu pilih **`SFSRootCAG2.pem`** dari menu drop-down Pilih sertifikat.

3. Buat setidaknya satu contoh AWS DMS replikasi. Untuk informasi selengkapnya, lihat Bekerja dengan Instance AWS DMS Replikasi.
4. Buat setidaknya satu tugas AWS DMS replikasi. Untuk informasi selengkapnya, lihat Bekerja dengan AWS DMS Tugas.

   Untuk migrasi online, tugas migrasi Anda menggunakan jenis migrasi Migrasikan data yang ada dan meniru perubahan yang sedang berlangsung.

   Untuk migrasi hibrid, tugas migrasi Anda menggunakan jenis migrasi Replikasi perubahan data saja. Anda dapat memilih waktu mulai CDC agar selaras dengan waktu buang Anda dari operasi `mongodump`. Oplog MongoDB adalah idempoten. Untuk menghindari perubahan yang hilang, ada baiknya untuk meninggalkan beberapa menit tumpang tindih antara waktu selesai `mongodump` Anda dan waktu mulai CDC Anda.

Migrasi dari Serpihan Klaster

Proses untuk memigrasikan data dari cluster sharded MongoDB ke instance Amazon DocumentDB Anda pada dasarnya adalah beberapa migrasi set replika secara paralel. Pertimbangan utama saat menguji migrasi serpihan klaster adalah bahwa beberapa serpihan mungkin lebih banyak digunakan daripada yang lain. Situasi ini mengarah ke berbagai waktu berlalu untuk migrasi data. Pastikan bahwa Anda mengevaluasi setiap persyaratan serpihan `oplog` saat melaksanakan perencanaan dan pengujian.

Berikut ini adalah beberapa masalah konfigurasi yang perlu dipertimbangkan saat memigrasikan serpihan klaster:

- Sebelum menjalankan `mongodump` atau memulai tugas migrasi AWS DMS , Anda harus menonaktifkan penyeimbang serpihan klaster dan menunggu hingga migrasi yang sedang diproses selesai. Untuk informasi selengkapnya, lihat Nonaktifkan Penyeimbang dalam dokumentasi MongoDB.
- Jika Anda menggunakan AWS DMS untuk mereplikasi data, jalankan `cleanupOrphaned` perintah pada setiap pecahan sebelum menjalankan tugas migrasi. Jika Anda tidak menjalankan perintah ini, tugas mungkin gagal karena ID dokumen duplikat. Perhatikan bahwa perintah ini dapat memengaruhi performa. Untuk informasi selengkapnya, lihat cleanupOrphaned di dokumentasi MongoDB.
- Jika Anda menggunakan alat `mongodump` untuk membuang data, Anda harus menjalankan satu proses `mongodump` per serpihan. Pendekatan yang paling efisien dari segi waktu mungkin memerlukan beberapa server migrasi untuk memaksimalkan performa buang Anda.
- Jika Anda menggunakan AWS Database Migration Service untuk mereplikasi data, Anda harus membuat titik akhir sumber untuk setiap pecahan. Jalankan juga setidaknya satu tugas migrasi untuk setiap serpihan yang Anda migrasikan. Pendekatan yang paling efisien dari segi waktu mungkin memerlukan beberapa instans replikasi untuk memaksimalkan performa migrasi Anda.

Pengujian Performa

Setelah berhasil memigrasikan data ke klaster Amazon DocumentDB pengujian Anda, jalankan beban kerja pengujian Anda terhadap klaster tersebut. Verifikasi melalui CloudWatch metrik Amazon bahwa kinerja Anda memenuhi atau melebihi throughput penerapan sumber MongoDB Anda saat ini.

Verifikasi metrik Amazon DocumentDB utama berikut:

- Throughput Jaringan
- Throughput Tulis
- Throughput Baca
- Keterlambatan replika

Untuk informasi selengkapnya, lihat Memantau Amazon DocumentDB.

Pengujian Failover

Pastikan bahwa perilaku aplikasi Anda selama peristiwa failover Amazon DocumentDB memenuhi persyaratan ketersediaan Anda. Untuk memulai failover manual klaster Amazon DocumentDB di konsol, pada halaman Klaster, pilih tindakan Failover pada menu Tindakan.

Anda juga dapat memulai failover dengan menjalankan `failover-db-cluster` operasi dari AWS CLI. Untuk informasi selengkapnya, lihat `failover-db-cluster`di bagian Amazon DocumentDB pada referensi. AWS CLI

Sumber Daya Tambahan

Lihat topik berikut di AWS Database Migration Service Panduan Pengguna:

- Menggunakan Amazon DocumentDB sebagai Target untuk AWS Database Migration Service
- Panduan: Bermigrasi dari MongoDB ke Amazon DocumentDB

Buku pedoman migrasi: MongoDB ke Amazon DocumentDB

Buku pedoman migrasi ini memberi Anda sumber daya dan langkah-langkah untuk membantu Anda bermigrasi dari database MongoDB ke Amazon DocumentDB.

Proses migrasi

Di bawah ini adalah langkah-langkah tingkat tinggi yang biasanya terlibat dalam memigrasikan data Anda dari database MongoDB ke Amazon DocumentDB.

Topik

- Langkah 1: Kompatibilitas dan perbedaan fungsional
- Langkah 2: Bukti konsep
- Langkah 3: Migrasikan data
- Langkah 4: Validasi data
- Langkah 5: Cutover aplikasi

Langkah 1: Kompatibilitas dan perbedaan fungsional

Amazon DocumentDB berinteraksi dengan Apache 2.0 open-source MongoDB 3.6, 4.0, dan 5.0 API. Akibatnya, Anda dapat menggunakan driver, aplikasi, dan alat MongoDB yang sama dengan Amazon DocumentDB dengan sedikit atau tanpa perubahan.

Langkah pertama adalah memeriksa kompatibilitas antara operator dan indeks yang digunakan aplikasi Anda pada database MongoDB Anda dan ketersediaannya di Amazon DocumentDB, serta untuk memahami perbedaan fungsional di antara mereka.

Kompatibilitas operator

Gunakan alat kompatibilitas Amazon DocumentDB* untuk dengan mudah mengetahui apakah aplikasi Anda menggunakan operator yang tidak didukung dalam kuerinya. Alat ini dapat memindai file log server database MongoDB Anda atau kode sumber aplikasi Anda untuk memberikan laporan operator yang tidak didukung. Jika Anda menemukan penggunaan operator yang tidak didukung, Anda perlu memodifikasi aplikasi Anda untuk bekerja di sekitar operator yang tidak didukung.

Untuk memeriksa kompatibilitas antara operator MongoDB yang digunakan dalam penyiapan Anda dan operator Amazon DocumentDB yang didukung, jalankan yang berikut ini:

```
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>
```

Untuk informasi selengkapnya, lihat API, Operasi, dan Tipe Data MongoDB Yang Didukung.

* Tidak didukung secara resmi olehAWS.

Kompatibilitas indeks

Anda dapat menggunakan alat indeks Amazon DocumentDB* untuk mengetahui apakah Anda menggunakan jenis indeks apa pun yang tidak didukung di Amazon DocumentDB. Alat ini membutuhkan koneksi ke database sumber Anda untuk membaca definisi indeks.

Untuk ini, Anda harus terlebih dahulu membuang definisi indeks ke dalam direktori menggunakan `--dump-indexes` opsi. Kemudian jalankan alat dengan `--show-issues` opsi, menyediakan direktori untuk menemukan indeks yang tidak kompatibel.

Indeks ekspor:

```
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>
```

Periksa indeks yang tidak kompatibel:

```
python3 migrationtools/documentdb_index_tool.py --show-issues --dir <dumped-index-
definitions-directory>
```

Jika Anda menemukan penggunaan jenis indeks yang tidak didukung, Anda harus memodifikasi aplikasi atau model data Anda untuk bekerja atau melanjutkan tanpa indeks yang tidak kompatibel.

Untuk informasi selengkapnya tentang jenis dan properti indeks yang didukung di Amazon DocumentDB, Indeks dan Properti Indeks lihat dan Cara mengindeks di Amazon DocumentDB.

* Tidak didukung secara resmi olehAWS.

Perbedaan fungsional

Tinjau Perbedaan fungsional pada MongoDB untuk membiasakan diri dengan perbedaannya.

Langkah 2: Bukti konsep

Lakukan bukti konsep dengan menjalankan aplikasi Anda atau rangkaian pengujian reguler Anda di Amazon DocumentDB untuk menguji fungsionalitas dan kinerja. Anda mungkin perlu mengisi klaster Amazon DocumentDB Anda dengan data untuk melakukan pengujian. Misalnya, Anda dapat menggunakan `mongorestore` alat `mongodump` dan untuk menyalin data dari MongoDB sumber Anda.

Pengujian fungsional

Buat klaster Amazon DocumentDB (Membuat cluster Amazon DocumentDBlihat) dan jalankan aplikasi atau rangkaian pengujian fungsional Anda untuk memvalidasi jika semua alur kerja aplikasi terus bekerja dengan mulus di Amazon DocumentDB.

Pengujian kinerja

Jalankan pengujian kinerja pada aplikasi atau rangkaian pengujian kinerja yang berjalan di Amazon DocumentDB dengan beban kerja yang mirip dengan beban kerja produksi untuk melihat apakah

penyiapan memenuhi persyaratan latensi Anda. Sesuaikan beban kerja Anda untuk kinerja atau skala klaster Amazon DocumentDB Anda jika berlaku. Lihat informasi yang lebih lengkap di Kinerja dan pemanfaatan sumber daya dan Menskalakan cluster Amazon DocumentDB.

Penting untuk mengukur cluster Amazon DocumentDB Anda dengan jenis instans yang tepat untuk kinerja yang optimal. Untuk informasi selengkapnya, lihat praktik terbaik untukPengukuran Instans.

Anda dapat menggunakan kalkulator ukuran Amazon DocumentDB* untuk membantu Anda memperkirakan ukuran cluster Amazon DocumentDB Anda.

* Tidak didukung secara resmi olehAWS.

Pengujian failover

Anda mungkin ingin mengamati bagaimana aplikasi Anda merespons reboot node primer Amazon DocumentDB, failover node primer, atau penghapusan node primer di cluster multi-node, serta saat node replika di-boot ulang atau dihapus. Ini akan membantu Anda mengonfirmasi bahwa aplikasi Anda tahan terhadap peristiwa ini. Untuk informasi selengkapnya, lihat Failover Pengujian.

Untuk memahami pengecualian yang harus ditoleransi aplikasi dan cara menanganinya secara efisien, lihat Membangun aplikasi tangguh dengan Amazon DocumentDB.

> **Note**
>
> Tidak ada pengganti untuk menguji beban kerja Anda di Amazon DocumentDB

Langkah 3: Migrasikan data

Setelah bukti konsep berhasil, migrasi data Anda ke Amazon DocumentDB. Sebagian besar pelanggan kami menggunakan pendekatan migrasi online atau offline untuk memigrasikan data mereka.

Migrasi online

Dengan menggunakan metode migrasi online, Anda dapat memigrasikan data dari database sumber Anda, mulai dari beberapa gigabyte hingga beberapa terabyte, ke Amazon DocumentDB dengan waktu henti mendekati nol. Untuk informasi lebih lanjut, lihat AWS Database Migration Service(AWS DMS).

Jika Anda bermigrasi dari database MongoDB, Anda dapat AWS DMS menggunakan untuk melakukan pemuatan penuh dan mereplikasi perubahan yang sedang berlangsung.

Untuk step-by-step prosesnya, lihat Migrasi ke Amazon DocumentDB dengan metode online.

Informasi tambahan dapat ditemukan di bagian Menggunakan Amazon DocumentDB sebagai target AWS Database Migration Service untuk di AWS Database Migration Service Panduan Pengguna.

Poin yang perlu diperhatikan denganAWS DMS:

- Segmentasi: Saat memigrasikan database multi-terabyte menggunakanAWS DMS, mungkin lambat dengan pengaturan default, karena beban penuh DMS adalah single-threaded per koleksi secara default, menghasilkan waktu migrasi yang lebih lama. Untuk mempercepat beban penuh untuk migrasi database besar, Anda dapat menggunakan fitur segmentasi di. AWS DMS

  Untuk detail selengkapnya tentang cara menggunakan segmentasi denganAWS DMS, lihat Menggunakan Segmentasi Otomatis Dengan. AWS DMS
- Jenis instans DMS: Untuk mempercepat migrasi data, Anda harus memilih instans DMS yang tepat.

Migrasi offline

Migrasi offline adalah pendekatan paling mudah untuk memindahkan database ke Amazon DocumentDB. Pendekatan ini terutama digunakan untuk POC dan untuk beban kerja yang dapat mengambil downtime penulisan selama migrasi.

Untuk step-by-step prosesnya, lihat Memigrasi dari MongoDB ke Amazon DocumentDB menggunakan metode offline.

Langkah 4: Validasi data

Setelah data berhasil dimigrasi, validasi data untuk kebenaran untuk mendapatkan kepercayaan diri. Di konsol tugas AWS DMS migrasi, Anda dapat menemukan metrik data yang dimigrasi. Untuk informasi selengkapnya, lihat memverifikasi data yang dimigrasi.

Anda juga dapat menggunakan Alat Amazon DataDiffer DocumentDB* untuk memvalidasi konsistensi data antara koleksi sumber dan target.

* Tidak didukung secara resmi olehAWS.

Langkah 5: Cutover aplikasi

Ini melibatkan mengubah string koneksi database aplikasi Anda untuk menggunakan klaster Amazon DocumentDB Anda.

Untuk informasi selengkapnya tentang menghubungkan ke Amazon DocumentDB, lihat. Menghubungkan ke Amazon DocumentDB sebagai Set Replika

Migrasi Online

Setelah pemuatan data lengkap selesai, AWS DMS terus mereplikasi perubahan yang sedang berlangsung dari sumber Anda ke Amazon DocumentDB. Setelah perubahan tertangkap dan pemeriksaan validasi data Anda selesai, Anda dapat melakukan cutover ke Amazon DocumentDB.

Migrasi Offline

Setelah selesai dengan pemuatan data penuh dan pemeriksaan validasi data, Anda dapat melakukan cutover ke Amazon DocumentDB.

Sumber daya tambahan

Berikut adalah beberapa sumber daya tambahan yang dapat membantu migrasi Anda:

- Video: Praktik terbaik untuk bermigrasi ke Amazon DocumentDB
- Video: Memulai Observabilitas dan Pemantauan Amazon DocumentDB
- Utilitas tambahan: Alat Amazon DocumentDB*
- Panduan pengembang migrasi: Migrasi ke Amazon DocumentDB

* Tidak didukung secara resmi olehAWS.

Peningkatan versi utama Amazon DocumentDB di tempat

Amazon DocumentDB membuat versi baru dari mesin database umumnya hanya tersedia setelah pengujian ekstensif. Anda dapat memilih bagaimana dan kapan untuk meningkatkan cluster Amazon DocumentDB Anda ke versi baru.

Saat ini, Amazon DocumentDB mendukung tiga versi utama: Amazon DocumentDB 3.6, 4.0, dan 5.0. Anda dapat melakukan upgrade versi utama (MVU) di tempat dari database Anda sambil menjaga titik akhir, penyimpanan, dan tag cluster yang sama dan dapat terus menggunakan aplikasi Anda tanpa modifikasi apa pun. Fitur ini tersedia secara gratis di semua wilayah di mana Amazon DocumentDB 5.0 tersedia.

> **Important**
>
> Cluster Amazon DocumentDB Anda tidak akan tersedia selama upgrade versi utama di tempat dan cluster Anda akan mengalami beberapa reboot. Upgrade downtime dapat bervariasi dari cluster ke cluster tergantung pada jumlah koleksi, indeks, database, dan instance. Kami merekomendasikan untuk melakukan peningkatan selama jendela pemeliharaan Anda atau selama jam pemanfaatan rendah. Setelah klaster Anda ditingkatkan, Anda tidak dapat menurunkan versi klaster ke versi sebelumnya, tetapi Anda dapat memilih untuk mengembalikan snapshot pra-upgrade ke cluster baru.

Topik

- Prasyarat dan batasan
- Praktik terbaik untuk peningkatan versi utama di tempat
- Melakukan upgrade versi utama di tempat
- Perbedaan antara Amazon DocumentDB 3.6/4.0 hingga 5.0 cluster yang ditingkatkan dan cluster Amazon DocumentDB 5.0 baru
- Memecahkan masalah upgrade versi utama di tempat

Prasyarat dan batasan

Berikut ini adalah prasyarat dan batasan untuk peningkatan versi utama di tempat yang mungkin perlu Anda pahami dan ditindaklanjuti sebelum melakukan pemutakhiran:

- Jenis Instans - Amazon DocumentDB 4.0/5.0 tidak mendukung instans r4.*. Untuk melanjutkan dengan upgrade versi utama di tempat, ubah instance r4.* ke instance r5.*. Untuk informasi selengkapnya, lihat Memodifikasi instance Amazon DocumentDB. Silakan merujuk ke Kelas Instans yang Didukung berdasarkan Wilayah untuk instance yang didukung berdasarkan versi mesin Amazon DocumentDB.
- Instans OS patch — Upgrade versi utama di tempat membutuhkan patch sistem operasi (OS) terbaru untuk melanjutkan. Harap terapkan tindakan pemeliharaan OS yang tertunda pada instans sebelum melanjutkan dengan peningkatan di tempat. Untuk informasi selengkapnya, lihat Bekerja dengan pembaruan sistem operasi.

> **Note**
>
> Dalam beberapa situasi, jika Anda memiliki patch mesin tingkat cluster yang tertunda, patch OS instance tidak terlihat. Anda mungkin perlu menerapkan patch mesin tingkat cluster sebelum melanjutkan dengan menerapkan patch OS instance dan, selanjutnya, upgrade versi utama di tempat. Lihat Melakukan pembaruan tambalan ke versi mesin cluster.

- Upgrade versi utama di tempat tersedia di semua wilayah di mana Amazon DocumentDB 5.0 tersedia.
- Upgrade versi utama di tempat tidak didukung dengan Amazon DocumentDB 4.0 sebagai versi target.
- Mulai di Amazon DocumentDB 4.0, “.” di nama pengguna tidak didukung. Jika Anda memutakhirkan dari Amazon DocumentDB 3.6 ke 5.0 dan memiliki nama pengguna yang berisi” . “, silakan buat ulang nama pengguna Anda tanpa” . “, sebelum melanjutkan dengan MVU di tempat.
- Upgrade versi utama di tempat saat ini tidak didukung pada cluster global Amazon DocumentDB dan cluster elastis.

> **Note**
>
> Untuk memutakhirkan kluster global Anda, hapus klaster sekunder Anda dari klaster global, ubah klaster primer menjadi kluster regional, lakukan peningkatan versi utama di tempat pada klaster regional (primer), lalu buat ulang klaster global dengan menambahkan cluster sekunder menggunakan nama yang sama untuk mempertahankan titik akhir yang sama seperti sebelumnya. Perhatikan bahwa Anda akan dikenakan biaya IO saat klaster utama yang ditingkatkan mereplikasi data ke cluster sekunder yang baru ditambahkan.

Untuk langkah-langkah rinci tentang cara menghapus cluster sekunder dari cluster global sebelum menghapus, lihat. Menghapus klaster dari klaster global Amazon DocumentDB

- Jika Anda memiliki sejumlah besar indeks (>10.000) dan beroperasi pada instance yang lebih kecil (misalnya, t3.medium), Anda harus meningkatkan instance utama Anda ke instance yang lebih besar (misalnya, setidaknya r5.xlarge) untuk menyimpan cukup memori dalam instance untuk melakukan peningkatan versi utama di tempat. Anda dapat memilih untuk menurunkan ukuran instans setelah pemutakhiran versi utama di tempat selesai. Lihat tabel di bawah untuk mengetahui jumlah maksimum indeks yang didukung pada setiap jenis instans untuk peningkatan versi utama di tempat:

  Untuk instance yang dioptimalkan memori (db.r5.*):

Instans	Indeks maksimum yang didukung untuk MVU di tempat
db.r5.large	100K
db.r5.xlarge	200K
db.r5.2xlarge	300K
db.r5.4xlarge	400K
db.r5.8xlarge	500K
db.r5.12xlarge	700K
db.r5.16xlarge	800K
db.r5.24xlarge	1M

  Untuk instance kinerja burstable (db.t3, db.t4g)

Instans	Indeks maksimum yang didukung untuk MVU di tempat
db.t4g.medium	3K

Instans	Indeks maksimum yang didukung untuk MVU di tempat
db.t3.medium	10K

Untuk instance graviton yang dioptimalkan memori (db.r6g.*):

Instans	Indeks maksimum yang didukung untuk MVU di tempat
db.r6g.large	100K
db.r6g.xlarge	200K
db.r6g.2xlarge	300K
db.r6g.4xlarge	400K
db.r6g.8xlarge	500K
db.r6g.12xlarge	700K
db.r6g.16xlarge	800K

> **Note**
>
> Jika Anda memiliki lebih dari 1M indeks, silakan hubungi AWS dukungan dan jangan melanjutkan dengan upgrade versi utama di tempat.

Praktik terbaik untuk peningkatan versi utama di tempat

Uji peningkatan versi utama di tempat menggunakan cluster kloning

1. Untuk menguji peningkatan versi utama di tempat, sebaiknya gunakan fitur kloning cepat untuk membuat klon dari kluster target Anda. Anda tidak akan dikenakan biaya penyimpanan apa pun untuk menguji peningkatan versi utama di tempat pada volume kloning, kecuali jika

Anda memodifikasi data apa pun di cluster. Untuk informasi lebih lanjut tentang klon volume, lihatMengkloning volume untuk cluster Amazon DocumentDB.

2. Untuk mendapatkan perkiraan yang lebih realistis tentang waktu yang dibutuhkan untuk menyelesaikan peningkatan versi utama di tempat, cocokkan jumlah instance cluster kloning dengan cluster yang ditargetkan.
3. Kami merekomendasikan untuk menguji sepenuhnya klaster Amazon DocumentDB 5.0 yang baru ditingkatkan untuk setiap perbedaan fungsional guna memastikan semuanya berfungsi seperti yang diharapkan.

Sebelum upgrade versi utama di tempat

1. Siapkan grup parameter cluster yang kompatibel dengan versi.

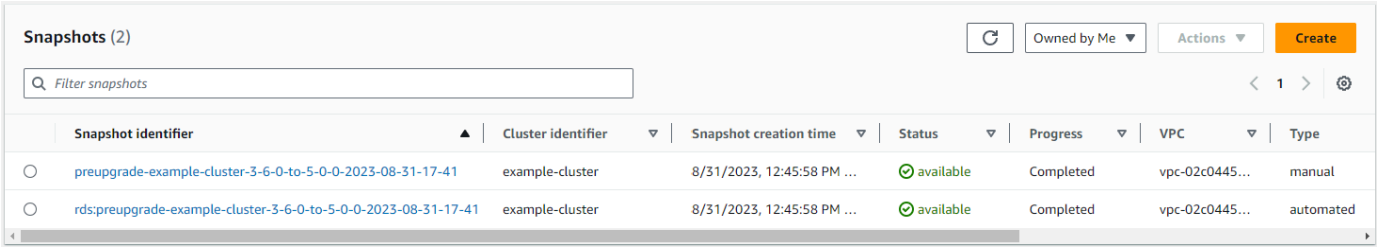
   Gunakan grup parameter cluster default Amazon DocumentDB untuk versi engine baru atau buat grup parameter cluster kustom Anda sendiri untuk versi engine yang baru.

   Jika Anda mengaitkan grup parameter klaster Amazon DocumentDB sebagai bagian dari permintaan pemutakhiran, pemutakhiran versi utama di tempat akan secara otomatis me-reboot klaster untuk menerapkan grup parameter baru.
2. Pastikan Anda telah memenuhi prasyarat untuk peningkatan versi utama di tempat seperti yang disebutkan di bagian Prasyarat dan batasan.
3. Buat snapshot manual.

   Proses upgrade membuat snapshot dari cluster database Anda selama upgrade. Sangat disarankan untuk membuat snapshot manual Anda sendiri sebelum proses upgrade. Lihat Membuat Manual Klaster Snapshot.

   > **Note**
   >
   > Snapshot otomatis yang dibuat oleh proses pemutakhiran tidak akan dihapus secara otomatis setelah pemutakhiran versi utama di tempat selesai. Cuplikan ini tidak akan dikenakan biaya apa pun selama berada dalam periode retensi. Anda dapat memilih untuk menghapus snapshot ini setelah Anda memverifikasi peningkatan klaster yang berhasil.

   Snapshot ini dinamai sebagai`preupgrade-<name>-<version>-<timestamp>`.



4. Periksa apakah Anda sudah menjadwalkan upgrade versi utama klaster Anda di tempat.

   Jika Anda telah memodifikasi cluster dan memilih untuk menerapkannya di jendela pemeliharaan berikutnya, jadwal peningkatan versi utama di tempat tidak akan terlihat di konsol, tetapi Anda dapat melihatnya di CLI. Anda dapat menjalankan perintah berikut untuk memeriksa apakah upgrade versi utama di tempat sudah dijadwalkan:

   ```
   aws docdb describe-db-cluster \
   --region $REGION \
   --db-cluster-identifier $CLUSTER_NAME

   "PendingModifiedValues": {
       "EngineVersion": "5.0.0"
   },
   ```

5. Lakukan beberapa proses kering menggunakan klon volume di lingkungan yang lebih rendah untuk menguji peningkatan versi utama pasca klaster di tempat pada rencana eksekusi dan perbedaan fungsional apa pun. Kami merekomendasikan kloning dengan jumlah dan ukuran instance yang sama untuk mendapatkan perkiraan yang lebih baik dari waktu proses peningkatan versi utama di tempat. Untuk informasi selengkapnya, lihat Mengkloning volume untuk cluster Amazon DocumentDB.
6. Jika langkah sebelumnya berhasil, lanjutkan dengan upgrade versi utama di tempat pada cluster produksi.

Selama upgrade versi utama di tempat

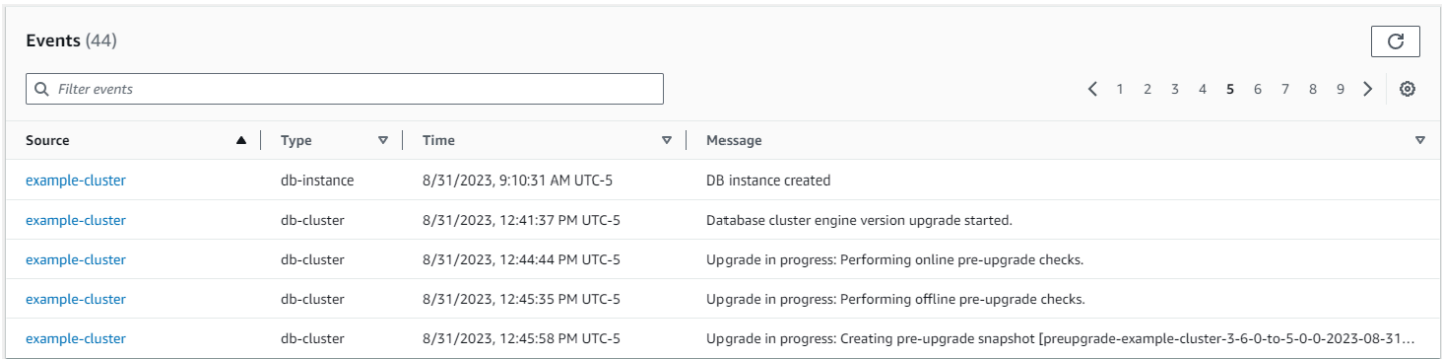
Anda dapat memantau kemajuan peningkatan versi utama di tempat dengan berlangganan acara pemeliharaan klaster. Saat pemutakhiran selesai, Anda akan menerima acara “Database cluster major version has been upgraded”. Ini dan peristiwa lain yang terjadi selama pemutakhiran muncul di bagian 'Acara dan Tag' pada halaman detail cluster di konsol Amazon DocumentDB. Status cluster kemudian berubah dari 'upgrade' menjadi 'tersedia'.

Dari CLI, Anda dapat menjalankan `aws docdb create-event-subscription` untuk membuat acara dan `aws docdb describe-events` memantau kemajuan. Anda juga dapat mengatur pemberitahuan acara untuk peristiwa di atas ke Amazon SNS sebagai target yang akan diberitahukan melalui email, pesan push, dan metode lainnya. Untuk informasi selengkapnya, lihat Berlangganan Langganan Peristiwa Amazon DocumentDB.

Pemutakhiran versi utama di tempat menghasilkan peristiwa berikut selama pemutakhiran:

- <cluster-name><timestamp>Upgrade sedang berlangsung: Membuat snapshot pra-upgrade [preupgrade- -]
- Upgrade sedang berlangsung: Volume kloning.
- Upgrade sedang berlangsung: Mengupgrade penulis.
- Upgrade sedang berlangsung: Meningkatkan pembaca.
- Database cluster versi utama telah ditingkatkan.

Acara juga terlihat di konsol di bawah halaman Acara:



Di dalam AWS CLI, Anda dapat menggunakan perintah berikut untuk melacak kemajuan:

```
aws docdb describe-events --source-identifier $CLUSTER_NAME --source-type db-cluster
{
    "Events": [
        {
            "SourceIdentifier": "mycluster",
            "SourceType": "db-cluster",
            "Message": "Database cluster engine version upgrade started.",
            "EventCategories": [
                "maintenance"
            ],
            "Date": "2023-07-11T23:20:32.444000+00:00",
            "SourceArn": "arn:aws:rds:us-east-1:xxxx:cluster:mycluster"
```

```
        }
    ]
}
```

Setelah peningkatan versi utama di tempat

Untuk Amazon DocumentDB 3.6, tambahkan tag ke cluster untuk membedakan bahwa cluster telah ditingkatkan ke Amazon DocumentDB 5.0 dari Amazon DocumentDB 3.6 sebagai lawan dari cluster Amazon DocumentDB 5.0 yang baru dibuat. Lihat bagian tentang perbedaan antara klaster Amazon DocumentDB 5.0 yang ditingkatkan dan cluster Amazon DocumentDB 5.0 yang baru.

Ambil snapshot manual setelah pemutakhiran versi utama di tempat selesai jika Anda perlu mengembalikan ke status pasca-peningkatan. Proses snapshot otomatis akan dilanjutkan segera setelah peningkatan versi utama di tempat selesai. Snapshot manual tidak akan dikenakan biaya apa pun selama berada dalam periode retensi.

Untuk menggunakan fitur baru yang terkait dengan Amazon DocumentDB 5.0, misalnya, enkripsi level bidang sisi klien, sebaiknya tingkatkan versi driver Anda ke versi API MongoDB 5.0. Untuk informasi selengkapnya, lihat Apa yang baru di Amazon DocumentDB 5.0 daftar fitur Amazon DocumentDB 5.0.

> **Important**
>
> Segera setelah melakukan upgrade versi utama (MVU) di tempat, klaster Amazon DocumentDB 5.0 Anda akan mengisi kembali metadata indeks, berdasarkan mana mesin database mengoptimalkan rencana eksekusi kueri. Kinerja kueri yang diharapkan pada klaster Amazon DocumentDB Anda akan dilanjutkan setelah proses perhitungan ulang metadata indeks selesai. Biasanya, proses ini selesai dalam beberapa menit tetapi dapat bertahan hingga dua jam tergantung pada jumlah indeks pada cluster Anda.
> Selain itu, reboot langsung, failover, atau skala naik/turun instance penulis Anda setelah MVU di tempat, dapat mengganggu proses perhitungan metadata indeks di cluster Anda. Setelah MVU di tempat selesai, sebaiknya lakukan perubahan tersebut setelah Anda mengamati performa kueri yang diharapkan di klaster Amazon DocumentDB 5.0.
> Silakan hubungi AWS dukungan jika Anda melihat penurunan kinerja sementara ini bertahan selama lebih dari dua jam setelah MVU di tempat.

Uji sepenuhnya klaster Amazon DocumentDB 5.0 yang ditingkatkan untuk memastikan semuanya berfungsi seperti yang diharapkan.

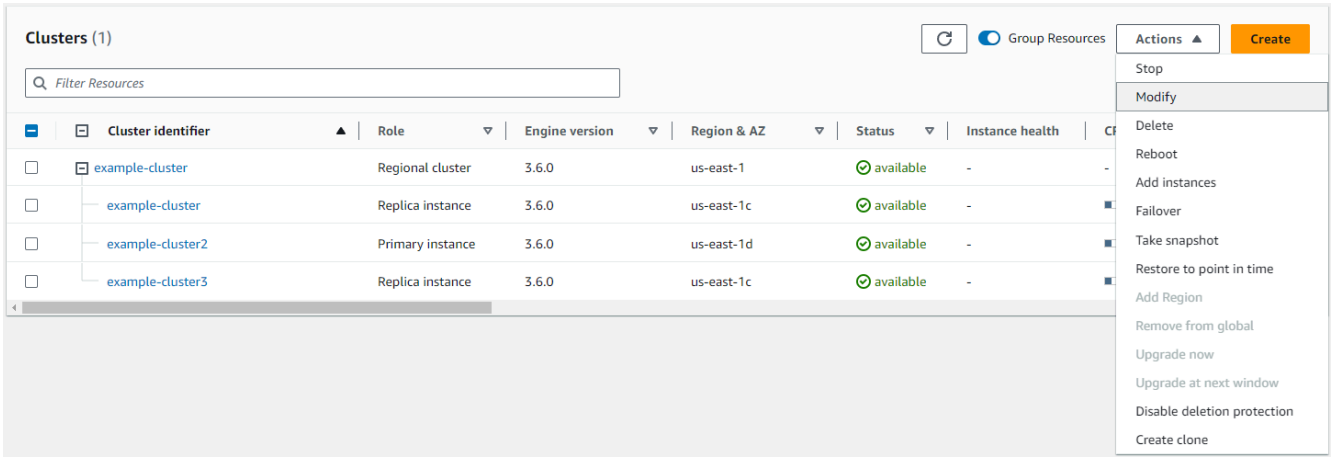
> **Note**
>
> Setelah melakukan MVU di tempat pada klaster Amazon DocumentDB dengan aliran perubahan diaktifkan, peristiwa aliran perubahan sebelumnya dipertahankan dan dapat dilanjutkan menggunakan atau. `resumeToken` `startAtOperationTime` Seperti halnya di klaster Amazon DocumentDB yang baru dibuat, ubah log peristiwa aliran yang lebih lama `change_stream_log_retention_duration` dari dihapus jika ukuran log lebih besar dari 51.200MB.

Melakukan upgrade versi utama di tempat

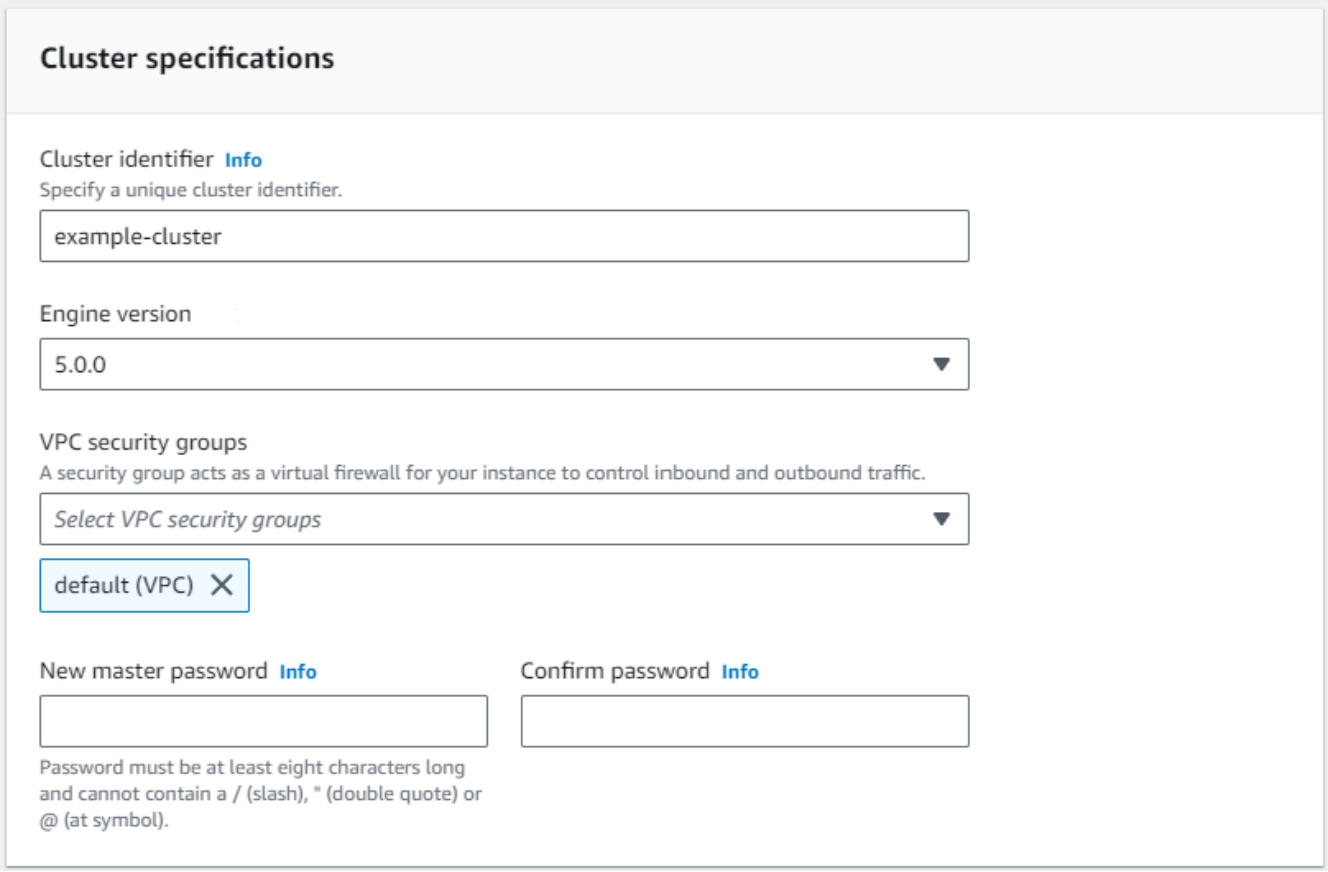
Using the AWS Management Console

Untuk melakukan upgrade versi utama di tempat menggunakan: AWS Management Console

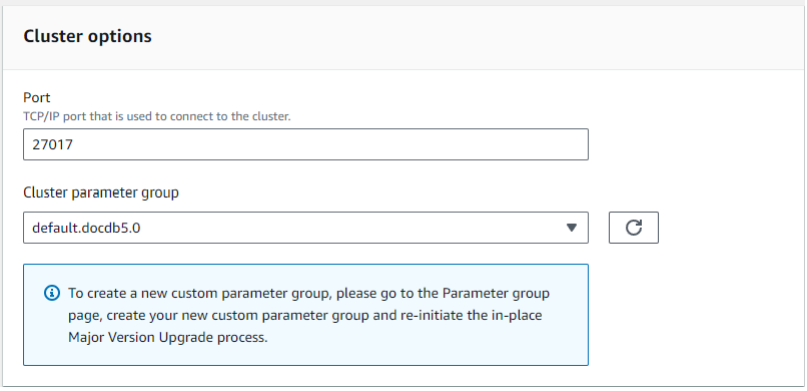
1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di tabel Clusters, pilih cluster sumber, klik Tindakan, lalu Ubah.



3. Pada dialog Modify cluster di bagian Spesifikasi cluster, pilih versi database yang ditargetkan (5.0) dari menu drop-down versi Engine.

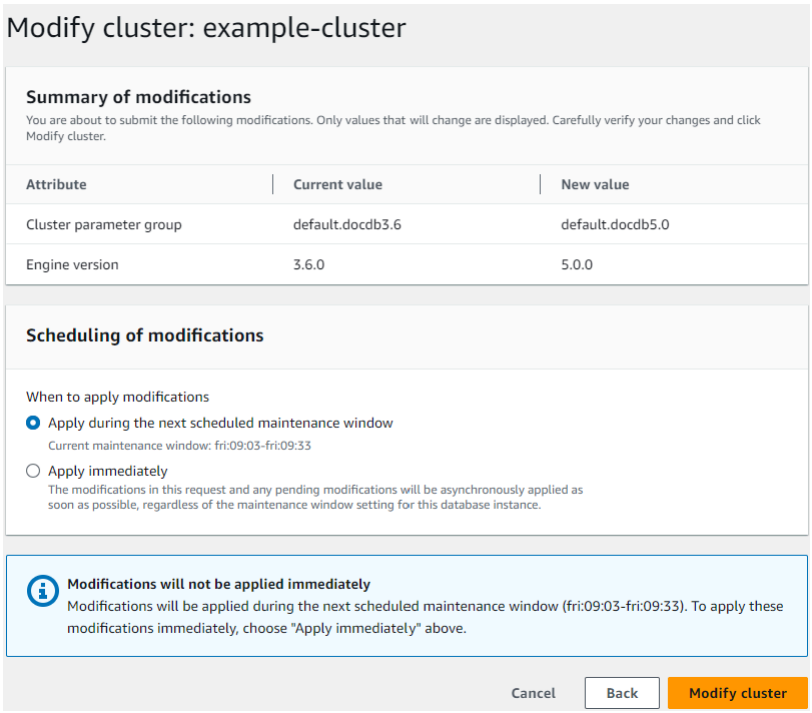


4. Di bagian opsi Cluster, pilih grup parameter cluster yang sesuai (default.docdb5.0) atau grup parameter yang dibuat khusus.

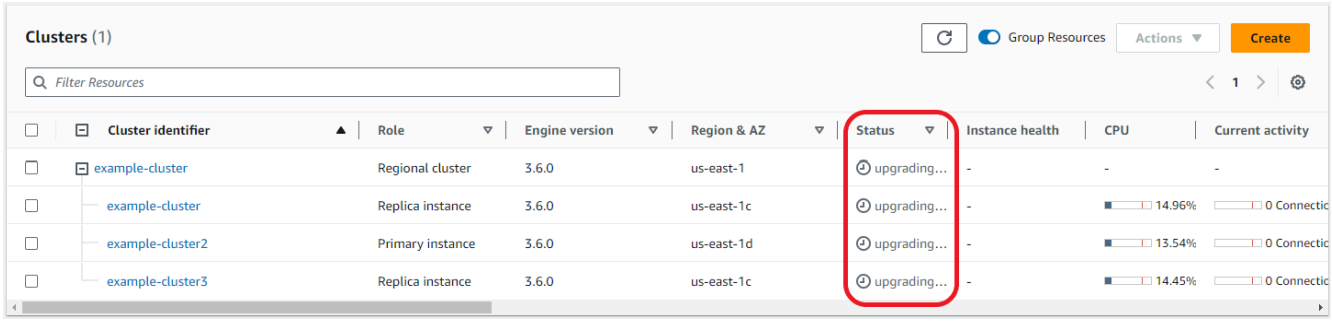


5. Setelah selesai, gulir ke bawah dan pilih Lanjutkan.
6. Di bagian Penjadwalan modifikasi, pilih rencana penjadwalan pilihan Anda: segera terapkan atau terapkan di jendela pemeliharaan berikutnya.

   Lalu pilih Ubah klaster.



7. Dalam tabel cluster, perhatikan status klaster Anda saat sedang ditingkatkan:



Using the AWS CLI

Gunakan `modify-db-cluster` API dengan versi engine yang diinginkan dan set `allow-major-version-upgrade` flag:

```
aws docdb modify-db-cluster \
 --db-cluster-identifier $CLUSTER_NAME \
 --allow-major-version-upgrade \
 --engine-version 5.0 \
 --apply-immediately \
 --cluster-parameter-group $PARAMETER_GROUP \
 --region $REGION
```

Perbedaan antara Amazon DocumentDB 3.6/4.0 hingga 5.0 cluster yang ditingkatkan dan cluster Amazon DocumentDB 5.0 baru

- Perbandingan subdokumen untuk beberapa tipe data numerik:
  - Jika cluster dimigrasikan dari Amazon DocumentDB 3.6, ia akan mewarisi perilaku perbandingan subdokumen Amazon DocumentDB 3.6. Perbedaan fungsional terbatas pada tipe numerik (seperti Long, Double, Decimal128) dalam subdokumen. Misalnya, `{a: {b: {NumberLong(1)}}` tidak sama `{a: {b: 1}}` di Amazon DocumentDB 3.6, sementara mereka dibandingkan sama di Amazon DocumentDB 4.0 dan setelahnya.
  - Perilaku perbandingan subdokumen ini hanya ada di Amazon DocumentDB 3.6, dan di Amazon DocumentDB 5.0 cluster yang ditingkatkan dari versi 3.6 menggunakan upgrade versi utama di tempat. Ini tidak berlaku untuk cluster Amazon DocumentDB 5.0 yang baru dibuat.
- Upgrade versi utama di tempat mempertahankan indeks asli pada cluster yang ditingkatkan. Sebagai praktik terbaik umum, kami merekomendasikan untuk menjatuhkan dan membuat ulang indeks Anda setelah MVU di tempat berhasil diselesaikan. Dengan Amazon DocumentDB 5.0, kami telah meningkatkan efisiensi keseluruhan proses pengumpulan sampah, terutama untuk indeks kardinalitas rendah. Jika Anda secara historis mengalami masalah dengan pengumpulan sampah di klaster Amazon DocumentDB 3.6 atau 4.0 Anda, klaster tersebut akan mendapat manfaat dari menjatuhkan dan membuat ulang indeks pasca-MVU. Membuat ulang indeks bukanlah persyaratan. Namun, membuat ulang indeks mungkin melibatkan I/O dan waktu tambahan. Untuk informasi selengkapnya, lihat Mengelola indeks Amazon DocumentDB.

> **Note**
>
> Untuk daftar perbedaan fungsional antara Amazon DocumentDB 3.6/4.0 dan Amazon DocumentDB 5.0, lihat. Kompatibilitas MongoDB

Memecahkan masalah upgrade versi utama di tempat

- Jika terjadi kegagalan, pemutakhiran versi utama di tempat akan mencoba rollback pemutakhiran untuk mengasumsikan status operasional klaster terakhir sebelum pemutakhiran dimulai. Rollback yang berhasil akan menghasilkan peristiwa: “Kluster database berada dalam keadaan yang tidak dapat ditingkatkan: cluster DocumentDB berada dalam keadaan di mana pemutakhiran versi utama tidak dapat diselesaikan dengan sukses.” Pada titik ini, Anda harus menghubungi tim AWS

dukungan untuk memecahkan masalah dan mencoba kembali peningkatan versi. Anda dapat terus menggunakan beban kerja Anda seperti sebelumnya. Dalam skenario langka lainnya di mana peningkatan memakan waktu lebih lama dari yang diharapkan, hubungi tim AWS dukungan untuk mendapatkan bantuan.

- Setelah MVU di tempat Anda berhasil diselesaikan, klaster yang ditingkatkan mungkin mengalami penurunan kinerja sementara dan pemanfaatan CPU yang tinggi untuk durasi waktu yang singkat, sementara proses penyegaran metadata indeks sedang berjalan. Jika Anda terus mengalami penurunan kinerja selama lebih dari 2 jam, silakan hubungi AWS dukungan.

Keamanan di Amazon DocumentDB

Keamanan cloud di AWS menjadi prioritas tertinggi. Sebagai seorang pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan dari organisasi yang paling sensitif terhadap keamanan.

Keamanan menjadi tanggung jawab bersama antara AWS dan Anda. Dokumentasi ini membantu Anda memahami cara untuk menerapkan model tanggung jawab bersama saat menggunakan Amazon DocumentDB. Model tanggung jawab bersama menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari program kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon DocumentDB (dengan kompatibilitas MongoDB), lihat Layanan AWS dalam Cakupan melalui Program Kepatuhan.
- Keamanan dalam cloud — Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, serta undang-undang dan peraturan yang berlaku.

> **Note**
>
> Bab ini berlaku untuk cluster berbasis instance dan Elastic Cluster. Untuk informasi lebih lanjut, lihat topik di bawah.

Anda juga akan mempelajari cara untuk menggunakan layanan AWS lain yang dapat membantu Anda memantau dan mengamankan sumber daya Amazon DocumentDB Anda. Topik berikut ini menunjukkan kepada Anda cara untuk mengonfigurasi Amazon DocumentDB untuk memenuhi tujuan keamanan dan kepatuhan Anda.

Topik

- Perlindungan Data di Amazon DocumentDB
- Identity and Access Management untuk Amazon DocumentDB
- Mengelola Pengguna Amazon DocumentDB

- Akses Database Menggunakan Kontrol Akses Berbasis Peran
- Pencatatan dan Pemantauan di Amazon DocumentDB
- Memperbarui Sertifikat TLS Amazon DocumentDB Anda
- Memperbarui Sertifikat Amazon DocumentDB TLS Anda - (AS-Barat) GovCloud
- Validasi Kepatuhan di Amazon DocumentDB
- Ketahanan di Amazon DocumentDB
- Keamanan Infrastruktur di Amazon DocumentDB
- Praktik Terbaik Keamanan untuk Amazon DocumentDB
- Mengaudit Peristiwa Amazon DocumentDB

Perlindungan Data di Amazon DocumentDB

Model tanggung jawab bersama AWS berlaku pada perlindungan data di . Sebagaimana dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya lindungi kredensial Akun AWS dan siapkan untuk masing-masing pengguna AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya AWS. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pengelogan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama semua kontrol keamanan bawaan dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.

- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon DocumentDB atau Layanan AWS lainnya menggunakan konsol, APIAWS CLI, atau SDK. AWS Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

- Enkripsi di sisi klien
- Mengenkripsi Amazon DocumentDB Data at Rest
- Mengenkripsi Data dalam Transit
- Manajemen kunci

Enkripsi di sisi klien

Enkripsi tingkat bidang sisi klien (FLE) Amazon DocumentDB memungkinkan Anda mengenkripsi data sensitif dalam aplikasi klien Anda sebelum ditransfer ke klaster Amazon DocumentDB. Data sensitif tetap dienkripsi ketika disimpan dan diproses dalam cluster dan didekripsi pada aplikasi klien ketika diambil.

Topik

- Mulai
- Kueri di sisi klien
- Keterbatasan:

Mulai

Konfigurasi awal FLE sisi klien di Amazon DocumentDB adalah proses empat langkah yang mencakup pembuatan kunci enkripsi, mengaitkan peran ke aplikasi, mengonfigurasi aplikasi, dan menentukan operasi CRUD dengan opsi enkripsi.

Topik

- Langkah 1: Buat kunci enkripsi
- Langkah 2: Kaitkan peran dengan aplikasi
- Langkah 3: Konfigurasikan aplikasi
- Langkah 4: Tentukan operasi CRUD
- Contoh: file konfigurasi enkripsi tingkat bidang sisi klien

Langkah 1: Buat kunci enkripsi

Dengan menggunakanAWS Key Management Service, buat kunci simetris yang digunakan untuk mengenkripsi dan mendekripsi bidang data sensitif dan berikan izin penggunaan IAM yang diperlukan. AWS KMSmenyimpan Customer Key (CK) yang digunakan untuk mengenkripsi Data Keys (DK). Kami merekomendasikan untuk menyimpan Kunci Pelanggan di KMS untuk memperkuat postur keamanan Anda. Kunci Data adalah kunci sekunder yang disimpan dalam koleksi Amazon DocumentDB dan diperlukan untuk mengenkripsi bidang sensitif sebelum menyimpan dokumen di Amazon DocumentDB. Kunci Pelanggan mengenkripsi Kunci Data yang pada gilirannya mengenkripsi dan mendekripsi data Anda. Jika Anda menggunakan klaster global, Anda dapat membuat kunci multi-wilayah yang dapat digunakan oleh peran layanan yang berbeda di berbagai wilayah.

Untuk informasi selengkapnya tentangAWS Key Management Service, termasuk cara membuat kunci, lihat Panduan Pengembang Layanan ManajemenAWS Kunci.

Langkah 2: Kaitkan peran dengan aplikasi

Buat kebijakan IAM denganAWS KMS izin yang sesuai. Kebijakan ini mengizinkan identitas IAM yang dilampirkan untuk mengenkripsi dan mendekripsi kunci KMS yang ditentukan di bidang sumber daya. Aplikasi Anda mengasumsikan peran IAM ini untuk mengautentikasi denganAWS KMS.

Kebijakan harus terlihat serupa dengan ini:

```
{ "Effect": "Allow",
```

```
"Action": ["kms:Decrypt", "kms:Encrypt"],
"Resource": "Customer Key ARN"
}
```

Langkah 3: Konfigurasikan aplikasi

Sekarang Anda mendefinisikan Kunci PelangganAWS KMS dan membuat peran IAM dan memberikan izin IAM yang tepat untuk mengakses Kunci Pelanggan. Impor paket yang diperlukan.

```
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. Tentukan 'aws' sebagai jenis penyedia KMS dan masukkan kredensi akun Anda yang diambil pada langkah sebelumnya.

   ```
   provider = "aws"
   kms_providers = {
       provider: {
           "accessKeyId": current_credentials.access_key,
           "secretAccessKey": current_credentials.secret_key
       }
   }
   ```

2. Tentukan kunci pelanggan yang digunakan untuk mengenkripsi kunci data:

   ```
   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. Konfigurasikan MongoClient objek:

```
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. Hasilkan Kunci Data Anda:

```
data_key_id = client_encryption.create_data_key(provider,
    customer_key,
    key_alt_name = [key_alt_name])
```

5. Ambil Kunci Data Anda yang ada:

```
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})
```

Langkah 4: Tentukan operasi CRUD

Tentukan operasi CRUD dengan opsi enkripsi.

1. Tentukan koleksi untuk menulis/membaca/menghapus satu dokumen:

```
coll = client.gameinfo.users
```

2. Enkripsi Eksplisit - mengenkripsi bidang dan masukkan:

> **Note**
>
> Tepat satu dari “key_id” atau “key_alt_name” harus disediakan.

```
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"]
})
```

Contoh: file konfigurasi enkripsi tingkat bidang sisi klien

Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

```
# 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()
```

Kueri di sisi klien

Amazon DocumentDB mendukung kueri kesetaraan titik dengan FLE sisi klien. Ketidaksetaraan dan kueri perbandingan dapat mengembalikan hasil yang tidak akurat. Operasi baca dan tulis mungkin memiliki perilaku yang tidak terduga atau salah dibandingkan dengan mengeluarkan operasi yang sama terhadap nilai yang didekripsi.

Misalnya, untuk query filter untuk dokumen di mana gamerscore lebih besar dari 500:

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
})
```

Klien menggunakan metode enkripsi eksplisit untuk mengenkripsi nilai kueri:

```
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 }
} )
```

Dalam operasi find, Amazon DocumentDB membandingkan nilai terenkripsi 500 dengan nilai bidang terenkripsi yang disimpan di setiap dokumen menggunakan pemeriksaan ketidaksetaraan yang lebih besar daripada. Pemeriksaan ketidaksetaraan dalam operasi find dapat mengembalikan hasil yang berbeda ketika dilakukan menggunakan data dan nilai yang didekripsi, meskipun operasi berhasil menghasilkan hasil.

Keterbatasan:

Batasan berikut berlaku untuk pengenkripsi tingkat bidang sisi klien Amazon DocumentDB:

- Amazon DocumentDB hanya mendukung kueri kesetaraan titik. Ketidaksetaraan dan kueri perbandingan dapat mengembalikan hasil yang tidak akurat. Operasi baca dan tulis mungkin memiliki perilaku yang tidak terduga atau salah dibandingkan dengan mengeluarkan operasi yang

sama terhadap nilai yang didekripsi. Untuk query filter untuk dokumen di mana gamerscore lebih besar dari 500.

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
    })
```

Klien menggunakan metode enkripsi eksplisit untuk mengenkripsi nilai kueri.

```
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 }
})
```

Dalam operasi find, Amazon DocumentDB membandingkan nilai terenkripsi 500 dengan nilai bidang terenkripsi yang disimpan di setiap dokumen menggunakan pemeriksaan ketidaksetaraan yang lebih besar daripada. Pemeriksaan ketidaksetaraan dalam operasi find dapat mengembalikan hasil yang berbeda ketika dilakukan menggunakan data dan nilai yang didekripsi, meskipun operasi berhasil menghasilkan hasil.

- Amazon DocumentDB tidak mendukung FLE sisi klien eksplisit dari Shell Mongo. Namun, fitur ini berfungsi dengan salah satu driver kami yang didukung.

Mengenkripsi Amazon DocumentDB Data at Rest

> **Note**
>
> AWS KMSmengganti istilah customer master key (CMK) dengan AWS KMS keydan kunci KMS. Konsepnya tidak berubah. Untuk mencegah melanggar perubahan,AWS KMS adalah menjaga beberapa variasi istilah ini.

Anda mengenkripsi data at rest di klaster Amazon DocumentDB dengan menentukan opsi enkripsi penyimpanan saat Anda membuat klaster. Enkripsi penyimpanan diaktifkan di seluruh klaster dan

diterapkan ke semua instans, termasuk instans primer dan replika apa pun. Hal ini juga diterapkan pada volume penyimpanan, data, indeks, log, backup otomatis, dan snapshot klaster Anda.

Amazon DocumentDB menggunakan Advanced Encryption Standard 256-bit (AES-256) untuk mengenkripsi data Anda menggunakan kunci enkripsi yang disimpan dalam AWS Key Management Service (AWS KMS). Saat menggunakan klaster Amazon DocumentDB dengan enkripsi saat istirahat diaktifkan, Anda tidak perlu mengubah logika aplikasi atau koneksi klien Anda. Amazon DocumentDB menangani enkripsi dan dekripsi data Anda secara transparan, dengan dampak minimal terhadap performa.

Amazon DocumentDB terintegrasi dengan AWS KMS dan menggunakan metode yang dikenal sebagai enkripsi amplop untuk melindungi data Anda. Ketika sebuah klaster Amazon DocumentDB dienkripsi dengan sebuahAWS KMS, Amazon DocumentDB memintaAWS KMS untuk menggunakan kunci KMS Anda guna menghasilkan kunci data ciphertext untuk mengenkripsi volume penyimpanan. Kunci data ciphertext dienkripsi menggunakan kunci KMS yang Anda tentukan, dan disimpan bersama dengan data terenkripsi dan metadata penyimpanan. Ketika Amazon DocumentDB perlu mengakses data terenkripsi Anda, Amazon DocumentDB memintaAWS KMS untuk mendekripsi kunci data ciphertext menggunakan kunci data KMS dan men-cache kunci data plaintext di memori untuk secara efisien mengenkripsi dan mendekripsi data dalam volume penyimpanan.

Fasilitas enkripsi penyimpanan di Amazon DocumentDB tersedia untuk semua ukuran instans yang didukung dan di semua Wilayah AWS di mana Amazon DocumentDB tersedia.

Mengaktifkan Enkripsi saat Istirahat untuk Klaster Amazon DocumentDB

Anda dapat mengaktifkan atau menonaktifkan enkripsi saat istirahat di klaster Amazon DocumentDB bila klaster tersebut disediakan menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI). Klaster yang Anda buat menggunakan konsol memiliki enkripsi saat istirahat yang diaktifkan secara default. Klaster yang Anda buat menggunakan AWS CLI memiliki enkripsi saat istirahat yang dinonaktifkan secara default. Oleh karena itu, Anda harus secara eksplisit mengaktifkan enkripsi saat istirahat menggunakan parameter `--storage-encrypted`. Dalam kedua kasus tersebut, setelah klaster dibuat, Anda tidak dapat mengubah opsi enkripsi saat istirahat.

Amazon DocumentDB menggunakan AWS KMS untuk mengambil dan mengelola kunci enkripsi, dan menetapkan kebijakan yang mengendalikan bagaimana kunci ini dapat digunakan. Jika Anda tidak menentukan pengidentifikasiAWS KMS kunci, Amazon DocumentDB menggunakan kunci KMS layananAWS terkelola default. Amazon DocumentDB membuat kunci KMS terpisah untuk setiapWilayah AWS di AndaAkun AWS. Untuk informasi selengkapnya, lihatAWS Key Management ServiceKonsep.

Untuk memulai membuat kunci KMS Anda sendiri, lihat Memulai dalam PanduanAWS Key Management Service Developer.

> **Important**
>
> Anda harus menggunakan kunci KMS enkripsi simetris untuk mengenkripsi klaster Anda karena Amazon DocumentDB hanya mendukung kunci KMS enkripsi simetris. Jangan gunakan kunci KMS asimetris untuk mencoba mengenkripsi data di klaster Amazon DocumentDB Anda. Untuk informasi selengkapnya, lihat Kunci asimetrisAWS KMS di dalam PanduanAWS Key Management Service Pengembang.

Jika Amazon DocumentDB tidak bisa lagi mendapatkan akses ke kunci enkripsi untuk sebuah klaster — misalnya, saat akses ke kunci dicabut — klaster terenkripsi masuk ke status terminal. Dalam hal ini, Anda hanya dapat memulihkan klaster dari backup. Untuk Amazon DocumentDB, backup selalu diaktifkan selama 1 hari.

Selain itu, jika Anda menonaktifkan kunci untuk klaster Amazon DocumentDB terenkripsi, Anda akhirnya akan kehilangan akses baca dan tulis ke klaster tersebut. Ketika Amazon DocumentDB menemukan klaster yang dienkripsi dengan kunci yang tidak dapat diaksesnya, klaster akan dimasukkan ke status terminal. Dalam kondisi ini, klaster DB tidak lagi tersedia, dan status basis data saat ini tidak dapat dipulihkan. Untuk memulihkan klaster, Anda harus mengaktifkan kembali akses ke kunci enkripsi untuk Amazon DocumentDB, lalu memulihkan klaster dari backup.

> **Important**
>
> Anda tidak dapat mengubah kunci KMS untuk klaster terenkripsi setelah Anda membuatnya. Pastikan untuk menentukan persyaratan kunci enkripsi Anda sebelum membuat klaster terenkripsi Anda.

Using the AWS Management Console

Anda menentukan opsi enkripsi saat istirahat saat membuat klaster. Enkripsi saat istirahat diaktifkan secara default saat Anda membuat klaster menggunakan AWS Management Console. Itu tidak dapat diubah setelah klaster dibuat.

Untuk menentukan opsi enkripsi saat istirahat saat membuat klaster Anda

1. Buat klaster Amazon DocumentDB seperti yang dijelaskan di bagian Memulai. Namun, pada langkah 6, jangan pilih Buat klaster.
2. Di bawah bagian Autentikasi, pilih Tampilkan pengaturan lanjutan.
3. Gulir ke bawah ke bawah ke bawah ke bawah ke bawah ke bawah kencryption-at-rest
4. Pilih opsi yang Anda inginkan untuk enkripsi saat istirahat. Opsi mana pun yang Anda pilih, Anda tidak dapat mengubahnya setelah klaster dibuat.
   - Untuk mengenkripsi data at rest di klaster ini, pilih Aktifkan enkripsi.
   - Jika Anda tidak ingin mengenkripsi data at rest di klaster ini, pilih Nonaktifkan enkripsi.
5. Pilih kunci utama yang Anda inginkan. Amazon DocumentDB menggunakan AWS Key Management Service (AWS KMS) untuk mengambil dan mengelola kunci enkripsi, dan menetapkan kebijakan yang mengendalikan bagaimana kunci ini dapat digunakan. Jika Anda tidak menentukan pengidentifikasiAWS KMS kunci, Amazon DocumentDB menggunakan kunci KMS layananAWS terkelola default. Untuk informasi selengkapnya, lihatAWS Key Management ServiceKonsep.

   > **Note**
   >
   > Setelah Anda membuat klaster terenkripsi, Anda tidak dapat mengubah kunci KMS untuk klaster tersebut. Pastikan untuk menentukan persyaratan kunci enkripsi Anda sebelum membuat klaster terenkripsi Anda.

6. Lengkapi bagian lain yang diperlukan, dan buat klaster Anda.

Using the AWS CLI

Untuk mengenkripsi klaster Amazon DocumentDB menggunakan AWS CLI, Anda harus menentukan opsi `--storage-encrypted` saat membuat klaster. Klaster Amazon DocumentDB yang dibuat menggunakan AWS CLI tidak mengaktifkan enkripsi penyimpanan secara default.

Contoh berikut membuat klaster Amazon DocumentDB dengan enkripsi penyimpanan yang diaktifkan.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --port 27017 \
      --engine docdb \
      --master-username yourMasterUsername \
      --master-user-password yourMasterPassword \
      --storage-encrypted
```

Untuk Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --port 27017 ^
      --engine docdb ^
      --master-username yourMasterUsername ^
      --master-user-password yourMasterPassword ^
      --storage-encrypted
```

Saat Anda membuat klaster Amazon DocumentDB terenkripsi, Anda dapat menentukan pengidentifikasi kunci AWS KMS, seperti dalam contoh berikut.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --port 27017 \
      --engine docdb \
      --master-username yourMasterUsername \
      --master-user-password yourMasterPassword \
      --storage-encrypted \
      --kms-key-id key-arn-or-alias
```

Untuk Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --port 27017 ^
      --engine docdb ^
      --master-username yourMasterUsername ^
```

```
    --master-user-password yourMasterPassword ^
    --storage-encrypted ^
    --kms-key-id key-arn-or-alias
```

> **Note**
>
> Setelah Anda membuat klaster terenkripsi, Anda tidak dapat mengubah kunci KMS untuk klaster tersebut. Pastikan untuk menentukan persyaratan kunci enkripsi Anda sebelum membuat klaster terenkripsi Anda.

Keterbatasan untuk Klaster Terenkripsi Amazon DocumentDB

Keterbatasan berikut ada untuk klaster terenkripsi Amazon DocumentDB.

- Anda dapat mengaktifkan atau menonaktifkan enkripsi saat istirahat untuk klaster Amazon DocumentDB hanya pada saat klaster dibuat, bukan setelah klaster telah dibuat. Namun, Anda dapat membuat salinan terenkripsi dari klaster yang tidak terenkripsi dengan membuat snapshot dari klaster yang tidak terenkripsi, lalu memulihkan snapshot yang tidak terenkripsi sebagai klaster baru sambil menentukan opsi enkripsi saat istirahat.

  Untuk informasi lain, lihat topik berikut:

  - Membuat Manual Klaster Snapshot
  - Memulihkan dari Snapshot Klaster
  - Menyalin Snapshot Klaster Amazon DocumentDB
- Klaster Amazon DocumentDB dengan enkripsi penyimpanan yang diaktifkan tidak dapat dimodifikasi untuk menonaktifkan enkripsi.
- Semua instans, backup otomatis, snapshot, dan indeks dalam klaster Amazon DocumentDB dienkripsi dengan kunci KMS yang sama.

Mengenkripsi Data dalam Transit

Anda dapat menggunakan Keamanan Lapisan Pengangkutan (TLS) untuk mengenkripsi koneksi antara aplikasi Anda dan klaster Amazon DocumentDB. Secara default, enkripsi dalam transit diaktifkan untuk klaster Amazon DocumentDB yang baru dibuat. Itu dapat secara opsional dinonaktifkan ketika klaster dibuat, atau di lain waktu. Ketika enkripsi dalam transit diaktifkan, koneksi aman menggunakan TLS diperlukan untuk terhubung ke klaster. Untuk informasi selengkapnya

menghubungkan ke Amazon DocumentDB menggunakan TLS, lihat Menghubungkan Secara Terprogram ke Amazon DocumentDB.

Mengelola Pengaturan TLS Klaster Amazon DocumentDB

Enkripsi dalam transit untuk klaster Amazon DocumentDB dikelola melalui parameter TLS dalam grup parameter klaster. Anda dapat mengelola pengaturan TLS cluster Amazon DocumentDB menggunakan atau () AWS Management Console . AWS Command Line Interface AWS CLI Lihat bagian berikut untuk mempelajari cara memverifikasi dan memodifikasi pengaturan TLS Anda saat ini.

Using the AWS Management Console

Ikuti langkah-langkah berikut untuk melakukan tugas manajemen untuk enkripsi TLS menggunakan konsol—seperti mengidentifikasi grup parameter, memverifikasi nilai TLS, dan membuat modifikasi yang diperlukan.

> **Note**
>
> Kecuali jika Anda menentukan secara berbeda saat membuat klaster, klaster Anda dibuat dengan grup parameter klaster default. Parameter dalam grup parameter klaster `default` tidak dapat diubah (misalnya, `tls` diaktifkan/dinonaktifkan). Jadi jika klaster Anda menggunakan grup parameter klaster `default`, Anda perlu memodifikasi klaster untuk menggunakan grup parameter klaster non-default. Pertama, Anda mungkin perlu membuat grup parameter klaster kustom. Untuk informasi selengkapnya, lihat Membuat grup parameter cluster Amazon DocumentDB.

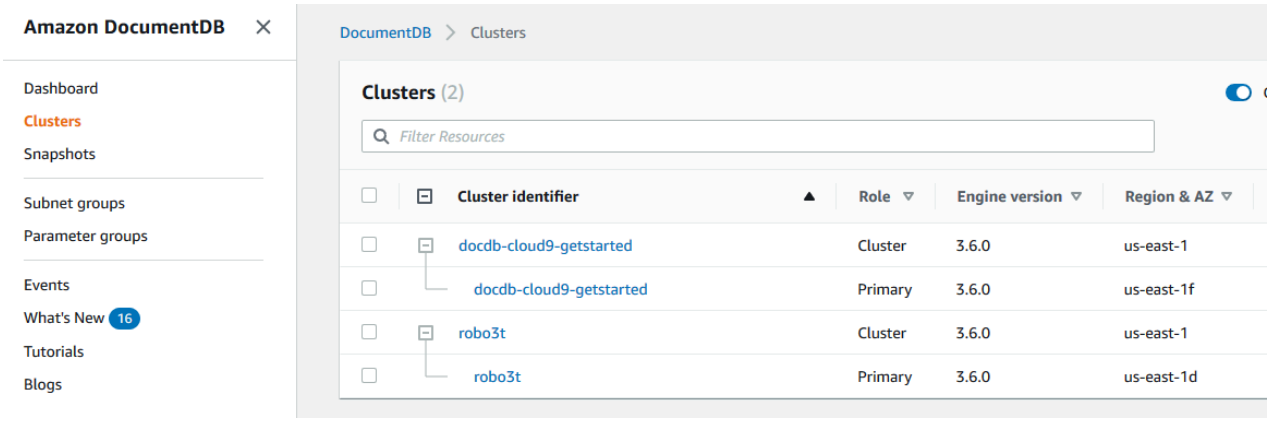
1. Tentukan grup parameter cluster yang digunakan cluster Anda.

   a. Buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.

   b. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

c. Perhatikan bahwa di kotak navigasi Clusters, kolom Cluster Identifier menunjukkan cluster dan instance. Instans terdaftar di bawah klaster. Lihat tangkapan layar di bawah untuk referensi.



d. Pilih klaster yang Anda minati.

e. Pilih tab Konfigurasi dan gulir ke bawah ke bagian bawah detail Cluster dan temukan grup parameter Cluster. Perhatikan nama grup parameter klaster.

Jika nama grup parameter klaster adalah `default` (misalnya, `default.docdb3.6`), Anda harus membuat grup parameter klaster kustom dan menjadikannya grup parameter klaster sebelum melanjutkan. Untuk informasi selengkapnya, lihat hal berikut:

1. Membuat grup parameter cluster Amazon DocumentDB — Jika Anda tidak memiliki grup parameter klaster kustom yang dapat Anda gunakan, buatlah.
2. Memodifikasi cluster Amazon DocumentDB — Modifikasi klaster Anda untuk menggunakan grup parameter klaster kustom.

2. Tentukan nilai parameter **`tls`** cluster saat ini.

a. Buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.

b. Di panel navigasi, pilih Grup parameter.

c. Dalam daftar grup parameter klaster, pilih nama grup parameter klaster yang Anda minati.

d. Temukan bagian Parameter klaster. Dalam daftar parameter klaster, temukan baris parameter klaster `tls`. Pada titik ini, empat kolom berikut ini penting:

- Nama parameter klaster — Nama dari parameter klaster. Untuk mengelola TLS, Anda tertarik dengan parameter klaster `tls`.
- Nilai — Nilai saat ini dari setiap parameter klaster.

- Nilai yang diizinkan — Daftar nilai yang dapat diterapkan pada parameter klaster.
- Tipe penerapan — Baik statis atau dinamis. Perubahan pada parameter klaster statis hanya dapat diterapkan saat instans di-boot ulang. Perubahan pada parameter klaster dinamis hanya dapat diterapkan baik segera atau saat instans di-boot ulang.

3. Ubah nilai parameter **`tls`** cluster.

   Jika nilai `tls` tidak sesuai dengan kebutuhan, modifikasi nilainya untuk grup parameter klaster ini. Untuk mengubah nilai parameter klaster `tls`, lanjutkan dari bagian sebelumnya dengan mengikuti langkah-langkah berikut.

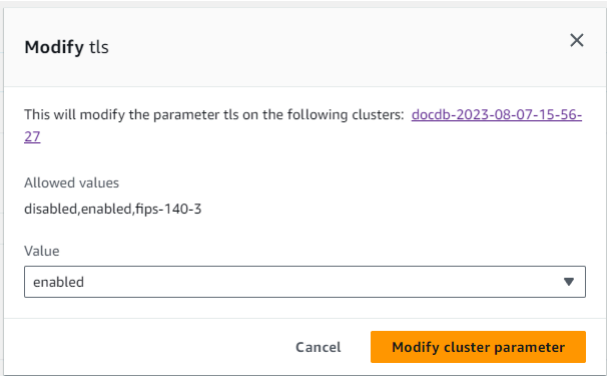
   a. Pilih tombol di sebelah kiri nama parameter klaster (`tls`).

   b. Pilih Edit.

   c. Untuk mengubah nilai `tls`, di kotak dialog Modifikasi **`tls`**, pilih nilai yang Anda inginkan untuk parameter klaster dalam daftar tarik-turun.

      Nilai yang valid adalah:

      - dinonaktifkan - Menonaktifkan TLS
      - diaktifkan - Mengaktifkan TLS (versi 1.0, 1.1, 1.2, dan 1.3)
      - fips-140-3 - Mengaktifkan TLS dengan FIPS. Cluster hanya menerima koneksi aman sesuai persyaratan publikasi Federal Information Processing Standards (FIPS) 140-3. Ini hanya didukung mulai dengan cluster Amazon DocumentDB 5.0 (versi engine 3.0.3727) di wilayah ini: ca-central-1, us-west-2, us-east-1, us-east-1, us-east-2, us-east-2, -1. us-gov-east us-gov-west



   d. Pilih Modifikasi parameter klaster. Perubahan diterapkan untuk setiap instans klaster saat di-boot ulang.

4. Nyalakan ulang instans Amazon DocumentDB.

   Boot ulang setiap instans klaster sehingga perubahan diterapkan ke semua instans di klaster.

   a. Buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
   b. Di panel navigasi, pilih Instans.
   c. Untuk menentukan instans yang akan di-boot ulang, temukan instans dalam daftar instans, dan pilih tombol di sebelah kiri namanya.
   d. Pilih Tindakan, dan kemudian Boot ulang. Konfirmasikan bahwa Anda ingin melakukan boot ulang dengan memilih Boot ulang.

Using the AWS CLI

Ikuti langkah-langkah berikut untuk melakukan tugas manajemen untuk enkripsi TLS menggunakan AWS CLI—seperti mengidentifikasi grup parameter, memverifikasi nilai TLS, dan membuat modifikasi yang diperlukan.

> **Note**
>
> Kecuali jika Anda menentukan secara berbeda saat membuat klaster, klaster dibuat dengan grup parameter klaster default. Parameter dalam grup parameter klaster `default` tidak dapat diubah (misalnya, `tls` diaktifkan/dinonaktifkan). Jadi jika klaster Anda menggunakan grup parameter klaster `default`, Anda perlu memodifikasi klaster untuk menggunakan grup parameter klaster non-default. Pertama Anda mungkin perlu membuat grup parameter klaster kustom. Untuk informasi selengkapnya, lihat Membuat grup parameter cluster Amazon DocumentDB.

1. Tentukan grup parameter cluster yang digunakan cluster Anda.

   Gunakan perintah `describe-db-clusters` dengan parameter berikut:

   - **`--db-cluster-identifier`** — Diperlukan. Nama klaster yang diinginkan.
   - **`--query`** — Opsional. Sebuah kueri yang membatasi output hanya pada bidang yang diminati, dalam hal ini, nama klaster dan nama grup parameter klasternya.

```
aws docdb describe-db-clusters \
```

```
        --db-cluster-identifier docdb-2019-05-07-13-57-08 \
        --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
        [
            "docdb-2019-05-07-13-57-08",
            "custom3-6-param-grp"
        ]
]
```

Jika nama grup parameter klaster adalah `default` (misalnya, `default.docdb3.6`), Anda harus memiliki grup parameter klaster kustom dan menjadikannya grup parameter klaster sebelum melanjutkan. Untuk informasi selengkapnya, lihat topik berikut:

1. Membuat grup parameter cluster Amazon DocumentDB — Jika Anda tidak memiliki grup parameter klaster kustom yang dapat Anda gunakan, buatlah.
2. Memodifikasi cluster Amazon DocumentDB — Modifikasi klaster Anda untuk menggunakan grup parameter klaster kustom.

2. Tentukan nilai parameter **`tls`** cluster saat ini.

Untuk mendapatkan informasi selengkapnya tentang grup parameter klaster ini, gunakan operasi `describe-db-cluster-parameters` dengan parameter berikut:

- **`--db-cluster-parameter-group-name`** — Diperlukan. Gunakan nama grup parameter klaster dari output perintah sebelumnya.
- **`--query`** — Opsional. Sebuah kueri yang membatasi output hanya pada bidang yang diminati, dalam hal ini, `ParameterName`, `ParameterValue`, `AllowedValues`, dan `ApplyType`.

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --query 'Parameters[*].
[ParameterName,ParameterValue,AllowedValues,ApplyType]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    [
        "audit_logs",
        "disabled",
        "enabled,disabled",
        "dynamic"
    ],
    [
        "tls",
        "disabled",
        "disabled,enabled,fips-140-3",
        "static"
    ],
    [
        "ttl_monitor",
        "enabled",
        "disabled,enabled",
        "dynamic"
    ]
]
```

3. Ubah nilai parameter **`tls`** cluster.

   Jika nilai `tls` tidak sesuai dengan kebutuhan, modifikasi nilainya untuk grup parameter klaster ini. Untuk mengubah nilai parameter klaster `tls`, gunakan operasi `modify-db-cluster-parameter-group` dengan parameter berikut.

   - **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang akan dimodifikasi. Ini tidak dapat menjadi grup parameter klaster `default.*`.
   - **`--parameters`** — Diperlukan. Daftar parameter grup parameter klaster yang akan dimodifikasi.
     - **`ParameterName`** — Diperlukan. Nama parameter klaster yang akan dimodifikasi.
     - **`ParameterValue`** — Diperlukan. Nilai baru untuk parameter klaster ini. Harus salah satu dari `AllowedValues` parameter klaster.
       - **`enabled`**- Cluster hanya menerima koneksi aman menggunakan TLS versi 1.0, 1.1, 1.2, atau 1.3.

  - **`disabled`** — Klaster tidak menerima koneksi aman menggunakan TLS.
  - **`fips-140-3`**Cluster hanya menerima koneksi aman sesuai persyaratan publikasi Federal Information Processing Standards (FIPS) 140-3. Ini hanya didukung mulai dengan cluster Amazon DocumentDB 5.0 (versi engine 3.0.3727) di wilayah ini: ca-central-1, us-west-2, us-east-1, us-east-1, us-east-2, us-east-2, -1. us-gov-east us-gov-west
- **`ApplyMethod`** — Kapan modifikasi ini diterapkan. Untuk parameter klaster statis seperti `tle`, nilai ini harus berupa `pending-reboot`.
  - **`pending-reboot`** — Perubahan diterapkan ke sebuah instans hanya setelah di-boot ulang. Anda harus mem-boot ulang setiap instans klaster satu per satu agar perubahan ini terjadi di semua instans klaster.

Kode berikut menonaktifkan `tls`, menerapkan perubahan ke setiap instans DB ketika di-boot ulang.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

Kode berikut memungkinkan `tls` (versi 1.0, 1.1. 1.2 dan 1.3), menerapkan perubahan ke setiap instans DB saat di-boot ulang.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters "ParameterName=tls,ParameterValue=enabled,ApplyMethod=pending-
reboot"
```

Kode berikut memungkinkan TLS dengan`fips-140-3`, menerapkan perubahan ke setiap instance DB saat di-boot ulang.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom5-0-param-grp \
    --parameters
 "ParameterName=tls,ParameterValue=fips-140-3,ApplyMethod=pending-reboot"
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

4. Nyalakan ulang instans Amazon DocumentDB Anda.

   Boot ulang setiap instans klaster sehingga perubahan diterapkan ke semua instans di klaster. Untuk mem-boot ulang instans Amazon DocumentDB, gunakan operasi `reboot-db-instance` dengan parameter berikut:

   - **`--db-instance-identifier`** — Diperlukan. Pengidentifikasi untuk instans yang akan di-boot ulang.

   Kode berikut mem-boot ulang instans `sample-db-instance`.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-db-instance
```

   Untuk Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-db-instance
```

   Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "DBInstance": {
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
        "PendingModifiedValues": {},
        "DBInstanceStatus": "rebooting",
        "DBSubnetGroup": {
            "Subnets": [
                {
```

```
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetIdentifier": "subnet-b3806e8f"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetIdentifier": "subnet-29ab1025"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupDescription": "default",
```

```
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupName": "default"
        },
        "PromotionTier": 2,
        "DBInstanceClass": "db.r5.4xlarge",
        "InstanceCreateTime": "2018-11-05T23:10:49.905Z",
        "PreferredBackupWindow": "00:00-00:30",
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-a50b-44d4-
b6a0-a177d5ff730b",
        "StorageEncrypted": true,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "EngineVersion": "3.6.0",
        "DbiResourceId": "db-SAMPLERESOURCEID",
        "DBInstanceIdentifier": "sample-cluster-instance-00",
        "Engine": "docdb",
        "AvailabilityZone": "us-east-1a",
        "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-cluster-
instance-00",
        "BackupRetentionPeriod": 1,
        "Endpoint": {
            "Address": "sample-cluster-instance-00.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        "DBClusterIdentifier": "sample-cluster"
    }
}
```

Diperlukan waktu beberapa menit agar instans di-boot ulang. Anda dapat menggunakan instans hanya ketika statusnya tersedia. Anda dapat memantau status instans menggunakan konsol atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status instans Amazon DocumentDB.

Manajemen kunci

Amazon DocumentDB menggunakan AWS Key Management Service (AWS KMS) untuk mengambil dan mengelola kunci enkripsi. AWS KMS menggabungkan perangkat keras dan perangkat lunak yang aman dan selalu tersedia untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Menggunakan AWS KMS, Anda dapat membuat kunci enkripsi dan menetapkan kebijakan yang mengendalikan bagaimana kunci ini dapat digunakan. AWS KMS mendukung AWS CloudTrail, sehingga Anda dapat mengaudit penggunaan kunci untuk memverifikasi bahwa kunci digunakan dengan tepat.

Kunci AWS KMS Anda dapat digunakan dalam kombinasi dengan Amazon DocumentDB dan layanan AWS yang didukung seperti Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), Amazon Elastic Block Store (Amazon EBS), dan Amazon Redshift. Untuk daftar layanan yang mendukung AWS KMS, lihat Cara Layanan AWS menggunakan AWS KMS dalam Panduan Developer AWS Key Management Service. Untuk informasi tentang AWS KMS, lihat Apa yang dimaksud dengan AWS Key Management Service?

Identity and Access Management untuk Amazon DocumentDB

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon DocumentDB. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Amazon DocumentDB bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Amazon DocumentDB
- Memecahkan masalah identitas dan akses Amazon DocumentDB
- Mengelola Izin Akses ke Sumber Daya Amazon DocumentDB Anda
- Menggunakan Kebijakan Berbasis Identitas (Kebijakan IAM) untuk Amazon DocumentDB
- AWS kebijakan terkelola untuk Amazon DocumentDB
- Referensi Izin Amazon DocumentDB API: Tindakan, Tindakan, Sumber Daya, dan Syarat

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon DocumentDB.

Pengguna layanan - Jika Anda menggunakan layanan Amazon DocumentDB untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Amazon DocumentDB untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon DocumentDB, lihat. Memecahkan masalah identitas dan akses Amazon DocumentDB

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Amazon DocumentDB di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amazon DocumentDB. Tugas Anda adalah menentukan fitur dan sumber daya Amazon DocumentDB mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Amazon Bagaimana Amazon DocumentDB bekerja dengan IAM DocumentDB, lihat.

Administrator IAM - Jika Anda administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Amazon DocumentDB. Untuk melihat contoh kebijakan berbasis identitas Amazon DocumentDB yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) dalam AWS dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensi yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin ke grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kapan harus membuat pengguna IAM (bukan peran) dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna IAM atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
  - Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan

tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Memilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL) dalam Panduan Developer Amazon Simple Storage Service.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana Amazon DocumentDB bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon DocumentDB, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Amazon DocumentDB.

Fitur IAM yang dapat Anda gunakan dengan Amazon DocumentDB

Fitur IAM	Cluster berbasis instans	Cluster elastis
Kebijakan berbasis identitas	Ya	Ya
Kebijakan berbasis sumber daya	Tidak	Tidak
Tindakan kebijakan	Ya	Ya
Sumber daya kebijakan	Ya	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya	Ya
ACL	Tidak	Tidak
ABAC (tanda dalam kebijakan)	Parsial	Ya
Kredensial sementara	Ya	Ya
Izin prinsipal	Ya	Ya
Peran layanan	Ya	Ya
Peran terkait layanan	Tidak	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon DocumentDB AWS dan layanan lainnya dengan sebagian besar fitur IAM AWS , lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Amazon DocumentDB

Mendukung kebijakan berbasis identitas	Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

Untuk melihat contoh kebijakan berbasis identitas Amazon DocumentDB, lihat. Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

Kebijakan berbasis sumber daya dalam Amazon DocumentDB

Mendukung kebijakan berbasis sumber daya	Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon DocumentDB

Mendukung tindakan kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

> **Note**
>
> Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS).
> Untuk melihat daftar tindakan RDS, lihat Tindakan yang ditentukan oleh Amazon Relational Database Service di Referensi Otorisasi Layanan.
> Untuk melihat tindakan kebijakan untuk klaster eleastic Amazon DocumentDB, lihat Tindakan yang ditentukan oleh klaster elastis Amazon DocumentDB di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon DocumentDB menggunakan awalan berikut sebelum tindakan:

```
aws
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [
      "aws:action1",
      "aws:action2"
         ]
```

Untuk melihat contoh kebijakan berbasis identitas Amazon DocumentDB, lihat. Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

Sumber daya kebijakan untuk Amazon DocumentDB

Mendukung sumber daya kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

> **Note**
>
> Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS).
> Untuk melihat daftar jenis sumber daya RDS dan ARNnya, lihat Sumber daya yang ditentukan oleh Amazon Relational Database Service di Referensi Otorisasi Layanan.
> Untuk mempelajari tindakan yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang ditentukan oleh Amazon Relational Database Service.
> Untuk melihat jenis sumber daya untuk klaster eleastic Amazon DocumentDB, lihat Jenis sumber daya yang ditentukan oleh klaster elastis Amazon DocumentDB dalam Referensi Otorisasi Layanan.

Untuk melihat contoh kebijakan berbasis identitas Amazon DocumentDB, lihat. Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

Kunci kondisi kebijakan untuk Amazon DocumentDB

Mendukung kunci kondisi kebijakan khusus layanan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tag dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

> **Note**
>
> Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS).
> Untuk melihat daftar kunci kondisi RDS, lihat Kunci kondisi untuk Amazon Relational Database Service di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh Amazon Relational Database Service.
> Untuk melihat kunci kondisi untuk klaster eleastic Amazon DocumentDB, lihat Kunci kondisi untuk klaster elastis Amazon DocumentDB di Referensi Otorisasi Layanan.

Untuk melihat contoh kebijakan berbasis identitas Amazon DocumentDB, lihat. Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

ACL di Amazon DocumentDB

Mendukung ACL	Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Amazon DocumentDB

> **Note**
>
> ABAC hanya didukung sebagian untuk cluster berbasis instance tetapi didukung untuk cluster elastis.

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tag milik prinsipal cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Apa itu ABAC? dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensi sementara dengan Amazon DocumentDB

Mendukung penggunaan kredensial sementara	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensyal sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensil sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Peralihan peran (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensyal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS

AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk Amazon DocumentDB

Mendukung sesi akses maju (FAS)	Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

Peran layanan untuk Amazon DocumentDB

Mendukung peran layanan	Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas Amazon DocumentDB. Edit peran layanan hanya jika Amazon DocumentDB memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon DocumentDB

> **Note**
>
> Peran terkait layanan tidak didukung untuk cluster berbasis instans tetapi didukung untuk cluster elastis.

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Amazon DocumentDB

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon DocumentDB. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon DocumentDB, termasuk format ARN untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Relational Database Service dalam Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol Amazon DocumentDB
- Mengizinkan pengguna melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon DocumentDB di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan.

Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan dalam IAM dalam Panduan Pengguna IAM.

Menggunakan konsol Amazon DocumentDB

Untuk mengakses konsol Amazon DocumentDB (dengan kompatibilitas MongoDB), Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon DocumentDB di Anda. Akun AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Amazon DocumentDB, lampirkan juga Amazon DocumentDB atau kebijakan terkelola ke *ConsoleAccess* entitas *ReadOnly* AWS . Untuk informasi selengkapnya, lihat Menambah izin untuk pengguna dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. 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": "*"
        }
    ]
}
```

Memecahkan masalah identitas dan akses Amazon DocumentDB

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon DocumentDB dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di Amazon DocumentDB
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya mengakses sumber daya Akun AWS Amazon DocumentDB saya

Saya tidak berwenang untuk melakukan tindakan di Amazon DocumentDB

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `aws:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aws:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `aws:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon DocumentDB.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon DocumentDB. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya mengakses sumber daya Akun AWS Amazon DocumentDB saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang

dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah Amazon DocumentDB mendukung fitur ini, lihat. Bagaimana Amazon DocumentDB bekerja dengan IAM
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.

Mengelola Izin Akses ke Sumber Daya Amazon DocumentDB Anda

Setiap AWS sumber daya dimiliki oleh Akun AWS, dan izin untuk membuat atau mengakses sumber daya diatur oleh kebijakan izin. Administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran), dan beberapa layanan (seperti AWS Lambda) juga mendukung melampirkan kebijakan izin ke sumber daya.

> **Note**
>
> Administrator akun (atau pengguna administrator) adalah pengguna dengan izin administrator. Untuk informasi selengkapnya, lihat Praktik Terbaik IAM dalam Panduan Pengguna IAM.

Topik

- Sumber Daya dan Operasi Amazon DocumentDB

- Memahami Kepemilikan Sumber Daya
- Mengelola Akses ke Sumber Daya
- Menentukan Elemen Kebijakan: Tindakan, Efek, Sumber daya, dan Penanggung Jawab
- Menentukan Syarat dalam Kebijakan

Sumber Daya dan Operasi Amazon DocumentDB

Di Amazon DocumentDB, sumber daya primer adalah klaster. Amazon DocumentDB mendukung sumber daya lain yang dapat digunakan dengan sumber daya primer seperti instans, grup parameter, dan langganan peristiwa. Sumber daya tersebut dirujuk sebagai sub-sumber daya.

Sumber daya dan sub-sumber daya ini memiliki Amazon Resource Names (ARN) unik yang terkait dengannya, seperti yang ditunjukkan dalam tabel berikut.

Jenis Sumber Daya	Format ARN
Klaster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster:` *`db-cluster-name`*
Grup parameter klaster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-pg:` *`cluster-parameter-group-name`*
Snapshot klaster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-snapshot:` *`cluster-snapshot-name`*
Instans	`arn:aws:rds:` *`region`*`:`*`account-id`* `:db:`*`db-instance-name`*
Grup keamanan	`arn:aws:rds:` *`region`*`:`*`account-id`* `:secgrp:`*`security-group-name`*
Grup subnet	`arn:aws:rds:` *`region`*`:`*`account-id`* `:subgrp:`*`subnet-group-name`*

Amazon DocumentDB menyediakan serangkaian operasi untuk bekerja dengan sumber daya Amazon DocumentDB. Untuk daftar operasi yang tersedia, lihat Tindakan.

Memahami Kepemilikan Sumber Daya

Pemilik sumber daya adalah Akun AWS yang menciptakan sumber daya. Artinya, pemilik sumber daya adalah entitas utama (akun root, pengguna IAM, atau peran IAM) yang mengautentikasi permintaan yang membuat sumber daya. Akun AWS Contoh berikut menggambarkan cara kerjanya:

- Jika Anda menggunakan kredensi akun root Anda Akun AWS untuk membuat sumber daya Amazon DocumentDB, seperti instance, Akun AWS Anda adalah pemilik sumber daya Amazon DocumentDB.
- Jika Anda membuat pengguna IAM di Anda Akun AWS dan memberikan izin untuk membuat sumber daya Amazon DocumentDB kepada pengguna tersebut, pengguna dapat membuat sumber daya Amazon DocumentDB. Namun, milik Anda Akun AWS, yang menjadi milik pengguna, memiliki sumber daya Amazon DocumentDB.
- Jika Anda membuat peran IAM Akun AWS dengan izin untuk membuat sumber daya Amazon DocumentDB, siapa pun yang dapat mengambil peran tersebut dapat membuat sumber daya Amazon DocumentDB. Anda Akun AWS, yang menjadi milik peran tersebut, memiliki sumber daya Amazon DocumentDB.

Mengelola Akses ke Sumber Daya

Kebijakan izin menjelaskan siapa yang memiliki akses ke suatu objek. Bagian berikut menjelaskan opsi yang tersedia untuk membuat kebijakan izin.

> **Note**
>
> Bagian ini membahas penggunaan IAM dalam konteks Amazon DocumentDB. Bagian ini tidak memberikan informasi yang mendetail tentang layanan IAM. Untuk dokumentasi IAM lengkap, lihat Apa yang Dimaksud dengan IAM? dalam Panduan Pengguna IAM. Untuk informasi tentang sintaks dan deskripsi kebijakan IAM, lihat Referensi AWSIAM Kebijakan di Panduan Pengguna IAM.

Kebijakan yang terlampir pada identitas IAM disebut sebagai kebijakan berbasis identitas (kebijakan IAM). Kebijakan yang terlampir pada sumber daya disebut sebagai kebijakan berbasis sumber daya. Amazon DocumentDB hanya mendukung kebijakan berbasis identitas (kebijakan IAM).

Topik

- Kebijakan Berbasis Identitas (Kebijakan IAM)
- Kebijakan Berbasis-Sumber Daya

Kebijakan Berbasis Identitas (Kebijakan IAM)

Anda dapat melampirkan kebijakan ke identitas IAM Anda. Misalnya, Anda dapat melakukan hal berikut:

- Lampirkan kebijakan izin ke pengguna atau grup dalam akun Anda – Administrator akun dapat menggunakan kebijakan izin yang berkaitan dengan pengguna tertentu untuk memberikan izin bagi pengguna tersebut untuk membuat sumber daya Amazon DocumentDB, seperti instans.
- Lampirkan kebijakan izin untuk peran (memberikan izin lintas akun) – Anda dapat melampirkan kebijakan izin berbasis identitas ke IAM role untuk memberikan izin lintas akun. Misalnya, administrator dapat membuat peran untuk memberikan izin lintas akun kepada orang lain Akun AWS atau AWS layanan sebagai berikut:
  1. Administrator akun A membuat peran IAM dan melampirkan kebijakan izin ke peran ini yang memberikan izin pada sumber daya di akun A.
  2. Administrator akun A melampirkan kebijakan kepercayaan ke peran yang mengidentifikasi Akun B sebagai prinsipal yang dapat mengambil peran tersebut.
  3. Administrator Akun B kemudian dapat mendelegasikan izin untuk mengambil peran kepada pengguna mana pun di Akun B. Melakukan hal ini memungkinkan pengguna di Akun B untuk membuat atau mengakses sumber daya di Akun A. Prinsip dalam kebijakan kepercayaan juga dapat menjadi prinsipal AWS layanan jika Anda ingin memberikan izin ke AWS layanan untuk mengambil peran tersebut.

  Untuk informasi selengkapnya tentang penggunaan IAM untuk mendelegasikan izin, lihat Manajemen Akses dalam Panduan Pengguna IAM.

Berikut ini adalah kebijakan contoh yang mengizinkan pengguna dengan ID `123456789012` untuk membuat instans untuk Akun AWS Anda. Instans yang baru harus menggunakan grup opsi dan grup parameter yang dimulai dengan `default`, dan harus menggunakan grup subnet `default`.

```
{
    "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"
            ]
        }
    ]
}
```

Untuk informasi selengkapnya tentang menggunakan kebijakan berbasis identitas dengan Amazon DocumentDB, lihat Menggunakan Kebijakan Berbasis Identitas (Kebijakan IAM) untuk Amazon DocumentDB. Untuk informasi lebih lanjut tentang pengguna, grup, peran, dan izin, lihat Identitas (Pengguna, Grup, dan Peran) dalam Panduan Pengguna IAM.

Kebijakan Berbasis-Sumber Daya

Layanan lainnya, seperti Amazon Simple Storage Service (Amazon S3), juga mendukung kebijakan izin berbasis sumber daya. Misalnya, Anda dapat melampirkan kebijakan ke bucket Amazon S3 untuk mengelola izin akses ke bucket tersebut. Amazon DocumentDB tidak mendukung kebijakan berbasis sumber daya.

Menentukan Elemen Kebijakan: Tindakan, Efek, Sumber daya, dan Pananggug Jawab

Untuk setiap sumber daya Amazon DocumentDB (lihat Sumber Daya dan Operasi Amazon DocumentDB), layanan menentukan serangkaian operasi API. Untuk informasi selengkapnya, lihat Tindakan. Untuk memberikan izin bagi operasi API tersebut, Amazon DocumentDB menentukan serangkaian tindakan yang dapat Anda tentukan dalam kebijakan. Melakukan operasi API dapat memerlukan izin untuk lebih dari satu tindakan.

Berikut ini adalah elemen-elemen kebijakan dasar:

- Sumber Daya – Dalam kebijakan, Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya yang diberlakukan oleh kebijakan tersebut.
- Tindakan – Anda menggunakan kata kunci tindakan untuk mengidentifikasi operasi sumber daya yang ingin Anda izinkan atau tolak. Misalnya, izin `rds:DescribeDBInstances` memungkinkan pengguna untuk melakukan operasi `DescribeDBInstances`.

- Efek – Anda menetapkan efek ketika pengguna meminta tindakan tertentu—hal ini dapat mengizinkan maupun menolak. Jika Anda tidak secara eksplisit memberikan akses ke (mengizinkan) sumber daya, akses akan ditolak secara implisit. Anda juga dapat secara eksplisit menolak akses ke sumber daya, yang mungkin Anda lakukan untuk memastikan bahwa pengguna tidak dapat mengaksesnya, meskipun kebijakan yang berbeda memberikan akses.
- Principal – Dalam kebijakan berbasis identitas (Kebijakan IAM), pengguna yang kebijakannya terlampir adalah principal yang implisit. Untuk kebijakan berbasis sumber daya, Anda menentukan pengguna, akun, layanan, atau entitas lain yang diinginkan untuk menerima izin (berlaku hanya untuk kebijakan berbasis sumber daya). Amazon DocumentDB tidak mendukung kebijakan berbasis sumber daya.

Untuk mempelajari selengkapnya tentang sintaksis dan penjelasan kebijakan IAM, lihat AWS Referensi Kebijakan IAM dalam Panduan Pengguna IAM.

Untuk tabel yang menampilkan semua Tindakan Amazon DocumentDB API dan sumber daya yang diterapkan, lihat Referensi Izin Amazon DocumentDB API: Tindakan, Tindakan, Sumber Daya, dan Syarat.

Menentukan Syarat dalam Kebijakan

Ketika Anda memberikan izin, Anda dapat menggunakan bahasa kebijakan IAM untuk menentukan kondisi ketika kebijakan harus berlaku. Misalnya, Anda mungkin ingin kebijakan diterapkan hanya setelah tanggal tertentu. Untuk informasi selengkapnya tentang menentukan kondisi dalam bahasa kebijakan, lihat Kondisi dalam Panduan Pengguna IAM.

Untuk menyatakan kondisi, Anda menggunakan kunci kondisi standar. Amazon DocumentDB tidak memiliki kunci konteks layanan khusus yang dapat digunakan dalam kebijakan IAM. Untuk daftar kunci konteks syarat global yang tersedia untuk semua layanan, lihat Kunci yang Tersedia untuk Syarat di Panduan Pengguna IAM.

Menggunakan Kebijakan Berbasis Identitas (Kebijakan IAM) untuk Amazon DocumentDB

> **Important**
>
> Untuk fitur manajemen tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon RDS. Konsol Amazon DocumentDB AWS CLI,, dan panggilan API dicatat sebagai panggilan yang dilakukan ke Amazon RDS API.

Kami menyarankan Anda untuk terlebih dahulu meninjau topik pendahuluan yang menjelaskan konsep dasar dan opsi yang tersedia bagi Anda untuk mengelola akses ke sumber daya Amazon DocumentDB Anda. Untuk informasi selengkapnya, lihat Mengelola Izin Akses ke Sumber Daya Amazon DocumentDB Anda.

Topik ini menyediakan contoh kebijakan berbasis identitas di mana administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran).

Berikut ini adalah contoh kebijakan IAM.

```
{
    "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"
            ]
        }
    ]
}
```

Kebijakan ini mencakup pernyataan tunggal yang menentukan izin berikut untuk pengguna IAM:

- Kebijakan ini memungkinkan pengguna IAM untuk membuat instance menggunakan tindakan createdBInstance (ini juga berlaku untuk operasi dan). create-db-instance AWS CLI AWS Management Console
- Elemen `Resource` menentukan bahwa pengguna dapat melakukan tindakan pada atau dengan sumber daya. Anda menentukan sumber daya menggunakan Amazon Resource Name (ARN). ARN ini mencakup nama layanan yang dimiliki sumber daya (`rds`), Wilayah AWS (*menunjukkan Wilayah apa pun dalam contoh ini), nomor akun pengguna (123456789012adalah ID pengguna dalam contoh ini), dan jenis sumber daya.

Elemen `Resource` dalam contoh menentukan batasan kebijakan berikut pada sumber daya untuk pengguna:

- Pengidentifikasi instans untuk instans baru harus dimulai dengan `test` (sebagai contoh, `testCustomerData1`, `test-region2-data`).
- Grup parameter klaster untuk instans baru harus dimulai dengan `default`.
- Grup subnet untuk instans baru harus grup subnet `default`.

Kebijakan tidak menentukan elemen `Principal` karena dalam kebijakan berbasis identitas, Anda tidak menentukan prinsipal yang mendapatkan izin. Saat Anda melampirkan kebijakan kepada pengguna, pengguna tersebut menjadi prinsipal secara implisit. Saat Anda melampirkan kebijakan izin pada IAM role, penanggung jawab yang diidentifikasi dalam kebijakan kepercayaan peran mendapatkan izin.

Untuk tabel yang menampilkan semua operasi Amazon DocumentDB API dan sumber daya yang mereka terapkan, lihat Referensi Izin Amazon DocumentDB API: Tindakan, Tindakan, Sumber Daya, dan Syarat.

Izin yang Diperlukan untuk Menggunakan Konsol Amazon DocumentDB

Agar pengguna dapat bekerja dengan konsol Amazon DocumentDB, pengguna tersebut harus memiliki rangkaian izin minimum. Izin ini memungkinkan pengguna untuk mendeskripsikan sumber daya Amazon DocumentDB Akun AWS untuk mereka dan untuk memberikan informasi terkait lainnya, termasuk keamanan Amazon EC2 dan informasi jaringan.

Jika Anda membuat kebijakan IAM yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya bagi pengguna dengan kebijakan IAM tersebut. Untuk memastikan bahwa pengguna tersebut masih dapat menggunakan konsol Amazon DocumentDB, lampirkan juga kebijakan terkelola `AmazonDocDBConsoleFullAccess` kepada pengguna, sebagaimana dijelaskan dalam AWS kebijakan terkelola untuk Amazon DocumentDB.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke API Amazon DocumentDB AWS CLI atau Amazon DocumentDB.

Contoh Kebijakan yang Dikelola Pelanggan

Dalam bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk berbagai tindakan Amazon DocumentDB. Kebijakan ini berfungsi saat Anda menggunakan tindakan Amazon DocumentDB API AWS , SDK, atau file. AWS CLI Saat Anda menggunakan konsol,

Anda perlu memberikan izin tambahan yang khusus untuk konsol. yang dibahas dalam Izin yang Diperlukan untuk Menggunakan Konsol Amazon DocumentDB.

Untuk fitur manajemen tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS) dan Amazon Neptune.

> **Note**
>
> Semua contoh menggunakan Wilayah US East (N. Virginia) (`us-east-1`) dan berisi ID akun fiktif.

Contoh

- Contoh 1: Mengizinkan Pengguna untuk Melakukan Setiap Tindakan yang Dijelaskan pada Setiap Sumber Daya Amazon DocumentDB
- Contoh 2: Mencegah Pengguna Menghapus Instans
- Contoh 3: Mencegah Pengguna Membuat Klaster kecuali Enkripsi Penyimpanan Diaktifkan

Contoh 1: Mengizinkan Pengguna untuk Melakukan Setiap Tindakan yang Dijelaskan pada Setiap Sumber Daya Amazon DocumentDB

Kebijakan izin berikut ini memberikan izin kepada pengguna untuk menjalankan semua tindakan yang dimulai dengan `Describe`. Tindakan tersebut menunjukkan informasi tentang sumber daya Amazon DocumentDB, seperti instans. Karakter wildcard (*) di elemen `Resource` menunjukkan bahwa tindakan diizinkan untuk semua sumber daya Amazon DocumentDB yang dimiliki akun tersebut.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowRDSDescribe",
         "Effect":"Allow",
         "Action":"rds:Describe*",
         "Resource":"*"
      }
   ]
}
```

Contoh 2: Mencegah Pengguna Menghapus Instans

Kebijakan izin berikut ini memberikan izin untuk mencegah pengguna menghapus instans tertentu. Sebagai contoh, Anda mungkin ingin menolak kemampuan untuk menghapus instans produksi Anda kepada setiap pengguna yang bukan administrator.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyDelete1",
         "Effect":"Deny",
         "Action":"rds:DeleteDBInstance",
         "Resource":"arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
      }
   ]
}
```

Contoh 3: Mencegah Pengguna Membuat Klaster kecuali Enkripsi Penyimpanan Diaktifkan

Kebijakan izin berikut ini menolak izin bagi pengguna untuk membuat klaster Amazon DocumentDB kecuali enkripsi penyimpanan diaktifkan.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PreventUnencryptedDocumentDB",
         "Effect": "Deny",
         "Action": "RDS:CreateDBCluster",
         "Condition": {
         "Bool": {
         "rds:StorageEncrypted": "false"
      },
         "StringEquals": {
         "rds:DatabaseEngine": "docdb"
         }
      },
      "Resource": "*"
      }
   ]
}
```

AWS kebijakan terkelola untuk Amazon DocumentDB

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan yang dikelola pelanggan IAM yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di AWS akun Anda. Untuk informasi selengkapnya tentang kebijakan AWS terkelola, lihat kebijakan AWS terkelola di Panduan Pengguna AWS Identity and Access Management.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan AWS terkelola untuk mendukung fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan AWS terkelola saat fitur baru diluncurkan atau saat operasi baru tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ViewOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke banyak AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk daftar dan deskripsi kebijakan fungsi pekerjaan, lihat kebijakan AWS terkelola untuk fungsi pekerjaan di Panduan Pengguna AWS Identity and Access Management.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk Amazon DocumentDB:

- AmazonDocDB FullAccess— Memberikan akses penuh ke semua sumber daya Amazon DocumentDB untuk akun root. AWS
- AmazonDocDB ReadOnlyAccess— Memberikan akses hanya-baca ke semua sumber daya Amazon DocumentDB untuk akun root. AWS
- AmazonDocDB ConsoleFullAccess— Memberikan akses penuh untuk mengelola sumber daya cluster elastis Amazon DocumentDB dan Amazon DocumentDB menggunakan file. AWS Management Console
- AmazonDocDB ElasticReadOnlyAccess— Memberikan akses hanya-baca ke semua sumber daya cluster elastis Amazon DocumentDB untuk akun root. AWS

- AmazonDocDB ElasticFullAccess— Memberikan akses penuh ke semua sumber daya cluster elastis Amazon DocumentDB untuk akun root. AWS

AmazonDocDB FullAccess

Kebijakan ini memberikan izin administratif yang memungkinkan akses penuh utama ke semua tindakan Amazon DocumentDB. Izin dalam kebijakan ini dikelompokkan sebagai berikut:

- Izin Amazon DocumentDB memungkinkan semua tindakan Amazon DocumentDB.
- Beberapa izin Amazon EC2 dalam kebijakan ini diperlukan untuk memvalidasi sumber daya yang diteruskan dalam permintaan API. Ini untuk memastikan Amazon DocumentDB berhasil menggunakan sumber daya dengan cluster. Izin Amazon EC2 lainnya dalam kebijakan ini memungkinkan Amazon DocumentDB membuat AWS sumber daya yang diperlukan untuk memungkinkan Anda terhubung ke cluster Anda.
- Izin Amazon DocumentDB digunakan selama panggilan API untuk memvalidasi sumber daya yang diteruskan dalam permintaan. Mereka diperlukan untuk Amazon DocumentDB untuk dapat menggunakan kunci yang diteruskan dengan cluster Amazon DocumentDB.
- CloudWatch Log diperlukan untuk Amazon DocumentDB untuk dapat memastikan bahwa tujuan pengiriman log dapat dijangkau, dan valid untuk penggunaan log broker.

```
{
    "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"
```

```
                }
            }
        }
    ]
}
```

AmazonDocDB ReadOnlyAccess

Kebijakan ini memberikan izin hanya-baca yang memungkinkan pengguna melihat informasi di Amazon DocumentDB. Prinsipal dengan kebijakan terlampir ini tidak dapat membuat pembaruan atau menghapus sumber daya yang keluar, juga tidak dapat membuat sumber daya Amazon DocumentDB baru. Misalnya, prinsipal dengan izin ini dapat melihat daftar cluster dan konfigurasi yang terkait dengan akun mereka, tetapi tidak dapat mengubah konfigurasi atau pengaturan cluster apa pun. Izin dalam kebijakan ini dikelompokkan sebagai berikut:

- Izin Amazon DocumentDB memungkinkan Anda mencantumkan sumber daya Amazon DocumentDB, menjelaskannya, dan mendapatkan informasi tentangnya.
- Izin Amazon EC2 digunakan untuk menggambarkan VPC Amazon, subnet, grup keamanan, dan ENI yang terkait dengan kluster.
- Izin Amazon DocumentDB digunakan untuk menggambarkan kunci yang terkait dengan cluster.

```
{
    "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:*"
            ]
        }
    ]
}
```

AmazonDocDB ConsoleFullAccess

Memberikan akses penuh untuk mengelola sumber daya AWS Management Console Amazon DocumentDB menggunakan berikut ini:

- Izin Amazon DocumentDB untuk mengizinkan semua tindakan cluster Amazon DocumentDB dan Amazon DocumentDB.
- Beberapa izin Amazon EC2 dalam kebijakan ini diperlukan untuk memvalidasi sumber daya yang diteruskan dalam permintaan API. Ini untuk memastikan Amazon DocumentDB berhasil menggunakan sumber daya untuk menyediakan dan memelihara cluster. Izin Amazon EC2 lainnya dalam kebijakan ini memungkinkan Amazon DocumentDB membuat AWS sumber daya yang diperlukan untuk memungkinkan Anda terhubung ke cluster seperti VPcendPoint.
- AWS KMS izin digunakan selama panggilan API AWS KMS untuk memvalidasi sumber daya yang diteruskan dalam permintaan. Mereka diperlukan untuk Amazon DocumentDB untuk dapat menggunakan kunci yang diteruskan untuk mengenkripsi dan mendekripsi data saat istirahat dengan cluster elastis Amazon DocumentDB.
- CloudWatch Log diperlukan untuk Amazon DocumentDB agar dapat memastikan bahwa tujuan pengiriman log dapat dijangkau, dan valid untuk mengaudit dan membuat profil penggunaan log.
- Izin Secrets Manager diperlukan untuk memvalidasi rahasia yang diberikan dan menggunakannya untuk mengatur pengguna admin untuk cluster elastis Amazon DocumentDB.
- Izin Amazon RDS diperlukan untuk tindakan pengelolaan klaster Amazon DocumentDB. Untuk fitur manajemen tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon RDS.
- Izin SNS memungkinkan prinsipal untuk langganan dan topik Amazon Simple Notification Service (Amazon SNS), dan untuk mempublikasikan pesan Amazon SNS.

- Izin IAM diperlukan untuk membuat peran terkait layanan yang diperlukan untuk metrik dan penerbitan log.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbSids",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster",
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:CreateGlobalCluster",
                "rds:DeleteDBCluster",
```

```
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
                "rds:DeleteDBSubnetGroup",
                "rds:DeleteEventSubscription",
                "rds:DeleteGlobalCluster",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeGlobalClusters",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:ModifyGlobalCluster",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
```

```
                "rds:RemoveFromGlobalCluster",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DependencySids",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssignPrivateIpAddresses",
                "ec2:AssociateAddress",
                "ec2:AssociateRouteTable",
                "ec2:AssociateSubnetCidrBlock",
                "ec2:AssociateVpcCidrBlock",
                "ec2:AttachInternetGateway",
                "ec2:AttachNetworkInterface",
                "ec2:CreateCustomerGateway",
                "ec2:CreateDefaultSubnet",
                "ec2:CreateDefaultVpc",
                "ec2:CreateInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
```

```
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeInstances",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:ModifyVpcEndpoint",
                "kms:DescribeKey",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DocdbSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "rds.amazonaws.com"
                }
            }
        },
```

```
        {
            "Sid": "DocdbElasticSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDB ElasticReadOnlyAccess

Kebijakan ini memberikan izin hanya-baca yang memungkinkan pengguna melihat informasi klaster elastis di Amazon DocumentDB. Prinsipal dengan kebijakan terlampir ini tidak dapat membuat pembaruan atau menghapus sumber daya yang keluar, juga tidak dapat membuat sumber daya Amazon DocumentDB baru. Misalnya, prinsipal dengan izin ini dapat melihat daftar cluster dan konfigurasi yang terkait dengan akun mereka, tetapi tidak dapat mengubah konfigurasi atau pengaturan cluster apa pun. Izin dalam kebijakan ini dikelompokkan sebagai berikut:

- Izin cluster elastis Amazon DocumentDB memungkinkan Anda mencantumkan sumber daya cluster elastis Amazon DocumentDB, mendeskripsikannya, dan mendapatkan informasi tentangnya.
- CloudWatch izin digunakan untuk memverifikasi metrik layanan.

```
{
   "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": "*"
      }
   ]
}
```

AmazonDocDB ElasticFullAccess

Kebijakan ini memberikan izin administratif yang memungkinkan akses penuh utama ke semua tindakan Amazon DocumentDB untuk klaster elastis Amazon DocumentDB.

Kebijakan ini menggunakan AWS tag (https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html) dalam kondisi untuk cakupan akses ke sumber daya. Jika Anda menggunakan rahasia, itu harus ditandai dengan kunci tag `DocDBElasticFullAccess` dan nilai tag. Jika Anda menggunakan kunci yang dikelola pelanggan, itu harus ditandai dengan kunci tag `DocDBElasticFullAccess` dan nilai tag.

Izin dalam kebijakan ini dikelompokkan sebagai berikut:

- Izin cluster elastis Amazon DocumentDB memungkinkan semua tindakan Amazon DocumentDB.
- Beberapa izin Amazon EC2 dalam kebijakan ini diperlukan untuk memvalidasi sumber daya yang diteruskan dalam permintaan API. Ini untuk memastikan Amazon DocumentDB berhasil menggunakan sumber daya untuk menyediakan dan memelihara cluster. Izin Amazon EC2 lainnya dalam kebijakan ini memungkinkan Amazon DocumentDB membuat AWS sumber daya yang diperlukan untuk memungkinkan Anda terhubung ke cluster Anda seperti titik akhir VPC.
- AWS KMS izin diperlukan agar Amazon DocumentDB dapat menggunakan kunci yang diteruskan untuk mengenkripsi dan mendekripsi data saat istirahat dalam cluster elastis Amazon DocumentDB.

> **Note**
>
> Kunci yang dikelola pelanggan harus memiliki tag dengan kunci `DocDBElasticFullAccess` dan nilai tag.

- SecretsManager izin diperlukan untuk memvalidasi rahasia yang diberikan dan menggunakannya untuk mengatur pengguna admin untuk cluster elastis Amazon DocumentDB.

> **Note**
>
> Rahasia yang digunakan harus memiliki tag dengan kunci `DocDBElasticFullAccess` dan nilai tag.

- Izin IAM diperlukan untuk membuat peran terkait layanan yang diperlukan untuk metrik dan penerbitan log.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbElasticSid",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "EC2Sid",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeVpcEndpoints",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeAvailabilityZones",
                "secretsmanager:ListSecrets"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "KMSSid",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ],
                    "aws:ResourceTag/DocDBElasticFullAccess": "*"
                }
```

```
            }
        },
        {
            "Sid": "KMSGrantSid",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/DocDBElasticFullAccess": "*",
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ]
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        },
        {
            "Sid": "SecretManagerSid",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecretVersionIds",
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:GetResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "secretsmanager:ResourceTag/DocDBElasticFullAccess": "*"
                },
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudwatchSid",
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "SLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDB- ElasticServiceRolePolicy

Anda tidak dapat melampirkan `AmazonDocDBElasticServiceRolePolicy` ke AWS Identity and Access Management entitas Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Amazon DocumentDB melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Peran terkait layanan dalam kelompok elastis.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
```

```
                    ]
                }
            }
        }
    ]
}
```

Amazon DocumentDB memperbarui kebijakan terkelola AWS

Perubahan	Deskripsi	Tanggal
AmazonDocDB ElasticFullAccess, AmazonDocDB ConsoleFullAccess - Ubah	Kebijakan diperbarui untuk menambahkan cluster start/stop dan menyalin tindakan snapshot cluster.	2/21/2024
AmazonDocDB ElasticReadOnlyAccess, AmazonDocDB ElasticFullAccess - Ubah	Kebijakan diperbarui untuk menambahkan `cloudwatch:GetMetricData` tindakan.	6/21/2023
AmazonDocDB ElasticReadOnlyAccess- Kebijakan baru	Kebijakan terkelola baru untuk cluster elastis Amazon DocumentDB	6/8/2023
AmazonDocDB ElasticFullAccess- Kebijakan baru	Kebijakan terkelola baru untuk cluster elastis Amazon DocumentDB	6/5/2023
AmazonDocDB- ElasticServiceRolePolicy – Kebijakan baru	Amazon DocumentDB menciptakan peran terkait layanan DB-Elastic AWS ServiceRoleForDoc baru untuk cluster elastis Amazon DocumentDB	11/30/2022
AmazonDocDB ConsoleFullAccess- Ubah	Kebijakan diperbarui untuk menambahkan izin klaster	11/30/2022

Perubahan	Deskripsi	Tanggal
	global dan elastis Amazon DocumentDB	
AmazonDocDB ConsoleFullAccess,AmazonDocDB FullAccess, AmazonDocDB ReadOnlyAccess - Kebijakan Baru	Peluncuran layanan	1/19/2017

Referensi Izin Amazon DocumentDB API: Tindakan, Tindakan, Sumber Daya, dan Syarat

Gunakan bagian berikut ini sebagai referensi ketika Anda mengatur Menggunakan Kebijakan Berbasis Identitas (Kebijakan IAM) untuk Amazon DocumentDB dan menulis kebijakan izin yang dapat Anda lampirkan ke identitas IAM (kebijakan berbasis identitas).

Berikut ini daftar setiap operasi Amazon DocumentDB API. Termasuk dalam daftar adalah tindakan terkait yang dapat Anda berikan izin untuk melakukan tindakan, AWS sumber daya yang dapat Anda berikan izin, dan kunci kondisi yang dapat Anda sertakan untuk kontrol akses berbutir halus. Anda menentukan tindakan di bidang `Action` kebijakan, nilai sumber daya di bidang `Resource` kebijakan, dan syarat di bidang `Condition` kebijakan. Untuk informasi lebih lanjut tentang syarat, lihat Menentukan Syarat dalam Kebijakan.

Anda dapat menggunakan kunci kondisi AWS-wide dalam kebijakan Amazon DocumentDB Anda untuk menyatakan kondisi. Untuk daftar lengkap tombol AWS-wide, lihat Kunci yang Tersedia di Panduan Pengguna IAM.

Anda dapat menguji kebijakan IAM dengan simulator kebijakan IAM. Secara otomatis menyediakan daftar sumber daya dan parameter yang diperlukan untuk setiap AWS tindakan, termasuk tindakan Amazon DocumentDB. Simulator kebijakan IAM menentukan izin yang diperlukan untuk setiap tindakan yang Anda tentukan. Untuk informasi tentang simulator kebijakan IAM, lihat Menguji Kebijakan IAM dengan Simulator Kebijakan IAM di Panduan Pengguna IAM.

> **Note**
>
> Untuk menentukan tindakan, gunakan awalan `rds:` yang diikuti dengan nama operasi API (sebagai contoh, `rds:CreateDBInstance`).

Berikut ini daftar operasi Amazon RDS API dan tindakan, sumber daya, dan kunci syarat yang berkaitan.

Topik

- Tindakan Amazon DocumentDB yang Mendukung Izin Tingkat Sumber Daya
- Tindakan Amazon DocumentDB yang Tidak Mendukung Izin Tingkat Sumber Daya

Tindakan Amazon DocumentDB yang Mendukung Izin Tingkat Sumber Daya

Izin tingkat sumber daya menyediakan kemampuan untuk menentukan sumber daya di mana pengguna diizinkan untuk melakukan tindakan. Amazon DocumentDB memiliki dukungan parsial untuk izin tingkat sumber daya. Artinya untuk tindakan Amazon DocumentDB tertentu, Anda dapat mengontrol kapan pengguna diizinkan untuk menggunakan tindakan tersebut berdasarkan syarat yang harus dipenuhi, atau sumber daya tertentu yang diizinkan untuk digunakan oleh pengguna. Sebagai contoh, Anda dapat memberikan izin kepada pengguna untuk memodifikasi hanya instans tertentu.

Berikut ini daftar operasi Amazon DocumentDB API dan tindakan, sumber daya, dan kunci syarat miliknya yang berkaitan.

> **Note**
>
> Untuk fitur manajemen tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon RDS. Untuk tindakan dan izin Amazon DocumentDB lainnya, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon RDS di Referensi Otorisasi Layanan.

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
AddTagsToResource rds:AddTagsToResource	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ApplyPendingMaintenanceAction rds:ApplyPendingMaintenanceAction	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
CopyDB ClusterSnapshot rds:CopyDBClusterSnapshot	Snapshot klaster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBCluster rds:CreateDBCluster	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Grup parameter klaster	rds:cluster-pg-tag

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
	arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	
	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
dibuatB ClusterParameterGroup rds:CreateDBClusterParameterGroup	Grup parameter klaster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
dibuatB ClusterSnapshot rds:CreateDBClusterSnapshot	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Snapshot klaster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBInstance rds:CreateDBInstance	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
dibuatB SubnetGroup rds:Creat eDBSubnetGroup	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DeleteDBInstance rds:Delet eDBInstance	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
DihapusB SubnetGro up rds:Delet eDBSubnetGroup	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DijelaskanB ClusterPa rameterGroups rds:Descr ibeDBClus terParame terGroups	Grup parameter klaster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
DijelaskanB ClusterParameters rds:DescribeDBClusterParameters	Grup parameter klaster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
DescribeDBClusters rds:DescribeDBClusters	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
DijelaskanB ClusterSnapshotAttributes rds:DescribeDBClusterSnapshotAttributes	Snapshot klaster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
DijelaskanB SubnetGroups rds:DescribeDBSubnetGroups	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DescribePendingMaintenanceActions rds:DescribePendingMaintenanceActions	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
FailOverdbCluster rds:FailoverDBCluster	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
ListTagsForResource rds:ListTagsForResource	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ModifyDBCluster rds:ModifyDBCluster	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Grup parameter klaster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
ModifyDB ClusterParameterGroup rds:ModifyDBClusterParameterGroup	Grup parameter klaster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
ModifyDB ClusterSnapshotAttribute rds:ModifyDBClusterSnapshotAttribute	Snapshot klaster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
ModifyDBInstance rds:ModifyDBInstance	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
RebootdBInstance rds:RebootDBInstance	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
RemoveTagsFromResource rds:RemoveTagsFromResource	Instans arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ResetDB ClusterParameterGroup rds:ResetDBClusterParameterGroup	Grup parameter klaster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Operasi dan Tindakan Amazon DocumentDB API	Sumber daya	Kunci Kondisi
DipindahkanBClusterFromSnapshot rds:RestoreDBClusterFromSnapshot	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Snapshot klaster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
DipindahkanBClusterToPointInTime rds:RestoreDBClusterToPointInTime	Klaster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Grup subnet arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag

Tindakan Amazon DocumentDB yang Tidak Mendukung Izin Tingkat Sumber Daya

Anda dapat menggunakan semua tindakan Amazon DocumentDB dalam kebijakan IAM untuk memberikan maupun menolak izin pengguna untuk menggunakan tindakan tersebut. Namun demikian, tidak semua tindakan Amazon DocumentDB mendukung izin tingkat sumber daya, yang memungkinkan Anda untuk menentukan sumber daya di mana tindakan dapat dilakukan. Tindakan Amazon DocumentDB API berikut ini saat ini tidak mendukung izin tingkat sumber daya. Oleh karena itu, untuk menggunakan tindakan tersebut dalam kebijakan IAM, Anda harus memberikan izin pengguna untuk menggunakan semua sumber daya untuk tindakan dengan menggunakan wildcard * untuk elemen `Resource` dalam pernyataan Anda.

- `rds:DescribeDBClusterSnapshots`
- `rds:DescribeDBInstances`

Mengelola Pengguna Amazon DocumentDB

Di Amazon DocumentDB, pengguna mengautentikasi ke klaster bersama dengan kata sandi. Setiap klaster memiliki kredenatif masuk utama yang dibuat selama pembuatan klaster.

> **Note**
>
> Semua pengguna baru yang dibuat sebelum 26 Maret 2020 telah diberikan peran `dbAdminAnyDatabase`, `readWriteAnyDatabase`, dan `clusterAdmin`. Sebaiknya Anda mengevaluasi kembali semua pengguna dan mengubah peran seperlunya untuk menerapkan hak istimewa paling sedikit untuk semua pengguna di klaster Anda.
> Untuk informasi selengkapnya, lihat Akses Database Menggunakan Kontrol Akses Berbasis Peran.

Primer dan**`serviceadmin`** Pengguna

Klaster Amazon DocumentDB yang baru dibuat memiliki dua pengguna: pengguna: pengguna utama dan`serviceadmin` pengguna.

Pengguna utama adalah pengguna tunggal dengan hak istimewa yang dapat melakukan tugas administratif dan membuat pengguna tambahan dengan peran. Saat Anda terhubung ke klaster Amazon DocumentDB untuk pertama kalinya, Anda harus mengautentikasi menggunakan kredenial masuk utama. Pengguna utama menerima izin administratif ini untuk klaster Amazon DocumentDB saat klaster tersebut dibuat, dan diberi peran`root`.

Pengguna `serviceadmin` dibuat secara implisit saat klaster dibuat. Setiap klaster Amazon DocumentDB memiliki pengguna `serviceadmin` yang memberikan AWS kemampuan untuk mengelola klaster Anda. Anda tidak dapat masuk sebagai, menjatuhkan, mengganti nama, mengubah kata sandi, atau mengubah izin untuk `serviceadmin`. Setiap upaya untuk melakukannya menghasilkan kesalahan.

> **Note**
>
> Pengguna utama dan`serviceadmin` pengguna untuk klaster Amazon DocumentDB tidak dapat dihapus dan peran`root` untuk pengguna utama tidak dapat dicabut.
> Jika Anda lupa kata sandi pengguna utama, Anda dapat mengatur ulang kata sandi menggunakanAWS Management Console atauAWS CLI.

Membuat Pengguna Tambahan

Setelah Anda terhubung sebagai pengguna utama (atau pengguna apa pun yang memiliki peran`createUser`), Anda dapat membuat pengguna baru, seperti yang ditunjukkan di bawah ini.

```
db.createUser(
    {
        user: "sample-user-1",
        pwd: "password123",
        roles:
            [{"db":"admin", "role":"dbAdminAnyDatabase" }]
    }
)
```

Untuk melihat detail pengguna, Anda dapat menggunakan perintah `show users` sebagai berikut. Anda juga dapat menghapus pengguna dengan perintah `dropUser`. Untuk informasi selengkapnya, lihat Perintah Umum.

```
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"
     }
    ]
}
```

Dalam contoh di atas, pengguna baru `sample-user-1` dikaitkan dengan basis data `admin`. Hal ini selalu terjadi bagi pengguna baru. Amazon DocumentDB tidak memiliki konsep `authenticationDatabase` dan dengan demikian semua autentikasi dilakukan dalam konteks basis data `admin`.

Saat membuat pengguna, jika Anda menghilangkan bidang db saat menentukan peran, Amazon DocumentDB akan secara implisit mengaitkan peran tersebut ke basis data tempat koneksi diterbitkan. Misalnya, jika koneksi Anda dikeluarkan terhadap basis data `sample-database` dan Anda menjalankan perintah berikut, pengguna `sample-user-2` akan dibuat di basis data `admin` dan akan memiliki izin `readWrite` ke basis data `sample-database`.

```
db.createUser(
    {
        user: "sample-user-2",
        pwd: "password123",
        roles:
            ["readWrite"]
    }
```

```
)
```

Membuat pengguna dengan peran yang dicakup di semua basis data (misalnya, `readInAnyDatabase`) mengharuskan Anda berada dalam konteks basis data `admin` saat membuat pengguna atau Anda secara eksplisit menyatakan basis data untuk peran saat membuat pengguna.

Untuk mengganti konteks dari basis data Anda, Anda dapat menggunakan perintah berikut.

```
use admin
```

Untuk mempelajari selengkapnya tentang Kontrol Akses Berbasis Peran dan menerapkan hak istimewa paling rendah di antara pengguna di klaster Anda, lihat Akses Database Menggunakan Kontrol Akses Berbasis Peran.

Kata Sandi Berputar Otomatis untuk Amazon DocumentDB

Dengan AWS Secrets Manager, Anda dapat mengganti kredensial hardcode dalam kode Anda (termasuk kata sandi) dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Ini membantu memastikan bahwa rahasia tidak dapat dikompromikan oleh seseorang yang memeriksa kode Anda, karena rahasianya tidak ada di sana. Juga, Anda dapat mengonfigurasi Secrets Manager untuk secara otomatis memutar rahasia untuk Anda sesuai dengan jadwal yang Anda tentukan. Ini memungkinkan Anda untuk mengganti rahasia jangka panjang dengan rahasia jangka pendek, yang membantu mengurangi risiko kompromi secara signifikan.

Dengan menggunakan Secrets Manager, Anda dapat secara otomatis memutar kata sandi Amazon DocumentDB Anda (yaitu, rahasia) menggunakan fungsi AWS Lambda yang disediakan oleh Secrets Manager.

Untuk informasi lebih lanjut tentang AWS Secrets Manager dan integrasi asli dengan Amazon DocumentDB, lihat berikut ini:

- Blog: Cara memutar kredenatif Amazon DocumentDB dan Amazon Redshift diAWS Secrets Manager
- Apa ituAWS Secrets Manager?
- Memutar Rahasia untuk Amazon DocumentDB

Akses Database Menggunakan Kontrol Akses Berbasis Peran

Anda dapat membatasi akses ke tindakan yang dapat dilakukan pengguna pada basis data menggunakan kontrol akses berbasis peran (RBAC) di Amazon DocumentDB (dengan kompatibilitas MongoDB). RBAC bekerja dengan memberikan satu atau lebih peran kepada pengguna. Peran ini menentukan operasi yang dapat dilakukan pengguna pada sumber daya database. Amazon DocumentDB saat ini mendukung kedua peran bawaan yang dicakup pada tingkat database, `read` seperti,,, `readWrite` `readAnyDatabase``clusterAdmin`, dan peran yang ditentukan pengguna yang dapat dicakup untuk tindakan tertentu dan sumber daya terperinci seperti koleksi berdasarkan kebutuhan Anda.

Kasus penggunaan umum untuk RBAC termasuk menerapkan hak istimewa paling sedikit dengan membuat pengguna dengan akses hanya-baca ke database atau koleksi dalam klaster, dan desain aplikasi multi-penyewa yang memungkinkan satu pengguna mengakses database atau koleksi tertentu dalam sebuah cluster.

> **Note**
>
> Semua pengguna baru yang dibuat sebelum 26 Maret 2020 telah diberikan peran `dbAdminAnyDatabase`, `readWriteAnyDatabase`, dan `clusterAdmin`. Disarankan bahwa Anda mengevaluasi kembali semua pengguna yang ada dan mengubah peran seperlunya untuk menerapkan hak istimewa terendah untuk klaster Anda.

Topik

- Konsep RBAC
- Memulai dengan peran bawaan RBAC
- Memulai dengan peran yang ditentukan pengguna RBAC
- Menghubungkan ke Amazon DocumentDB sebagai Pengguna
- Perintah Umum
- Perbedaan Fungsional
- Batas
- Akses Database Menggunakan Kontrol Akses Berbasis Peran

Konsep RBAC

Berikut ini adalah istilah dan konsep penting yang berkaitan dengan kontrol akses berbasis peran. Untuk informasi lebih lanjut tentang pengguna Amazon DocumentDB, lihat Mengelola Pengguna Amazon DocumentDB.

- Pengguna — Entitas individu yang dapat mengautentikasi ke basis data dan melakukan operasi.
- Kata Sandi — Rahasia yang digunakan untuk mengautentikasi pengguna.
- Peran — Mengotorisasi pengguna untuk melakukan tindakan pada satu atau lebih basis data.
- Basis Data Admin — Basis data di mana pengguna disimpan dan diberi otorisasi terhadapnya.
- Basis Data (**db**) — Namespace di dalam klaster yang berisi koleksi untuk menyimpan dokumen.

Perintah berikut ini membuat pengguna yang bernama `sample-user`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

Dalam contoh ini:

- `user: "sample-user"` — Mengindikasikan nama pengguna.
- `pwd: "abc123"` — Mengindikasikan kata sandi pengguna.
- `role: "read", "db: "sample-database"` — Mengindikasikan bahwa pengguna `sample-user` akan memiliki izin baca di `sample-database`.

Contoh berikut ini menunjukkan keluaram setelah Anda mendapatkan `sample-user` pengguna dengan `db.getUser(sample-user)`. Dalam contoh ini, `sample-user` pengguna berada di basis data `admin` tetapi memiliki peran baca untuk `sample-database` basis data.

Saat membuat pengguna, jika Anda menghilangkan bidang db saat menentukan peran, Amazon DocumentDB akan secara implisit mengaitkan peran tersebut ke basis data tempat koneksi diterbitkan. Sebagai contoh, jika koneksi Anda dikeluarkan terhadap `sample-database` basis data dan Anda menjalankan perintah berikut ini, `sample-user` pengguna akan dibuat dalam basis data `admin` dan akan memiliki izin `readWrite` ke `sample-database` basis data.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: ["readWrite"]})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "user":"sample-user",
   "roles":[
      {
         "db":"sample-database",
         "role":"readWrite"
      }
   ]
}
```

Membuat pengguna dengan peran yang tercakup di semua basis data (sebagai contoh, `readAnyDatabase`) mengharuskan Anda untuk berada dalam konteks basis data `admin` saat membuat pengguna, maupun Anda secara eksplisit menyatakan basis data untuk peran saat membuat pengguna. Untuk mengeluarkan perintah terhadap basis data `admin`, Anda dapat menggunakan `use admin` perintah. Untuk informasi selengkapnya, lihat Perintah Umum.

Memulai dengan peran bawaan RBAC

Untuk membantu Anda memulai kontrol akses berbasis peran, bagian ini memandu Anda melalui skenario contoh untuk menerapkan hak istimewa terendah dengan membuat peran untuk tiga pengguna dengan fungsi pekerjaan yang berbeda.

- `user1` adalah manajer baru yang perlu dapat melihat dan mengakses semua basis data dalam klaster.
- `user2` adalah karyawan baru yang memerlukan akses hanya ke satu basis data, `sample-database-1`, di klaster yang sama.
- `user3` adalah karyawan yang sudah ada yang perlu melihat dan mengakses basis data yang berbeda, `sample-database-2` yang tidak mereka miliki aksesnya sebelumnya, dalam klaster yang sama.

Pada suatu saat, keduanya `user1` dan `user2` meninggalkan perusahaan dan akses mereka harus dicabut.

Untuk membuat pengguna dan memberikan peran, pengguna yang Anda autentikasi ke klaster harus memiliki peran yang berkaitan yang dapat melakukan tindakan untuk `createUser` dan `grantRole`. Sebagai contoh, peran `admin` dan `userAdminAnyDatabase` keduanya dapat memberikan kemampuan semacam itu, sebagai contoh. Untuk tindakan sesuai peran, lihat Akses Database Menggunakan Kontrol Akses Berbasis Peran.

> **Note**
>
> Di Amazon DocumentDB, semua pengguna dan operasi peran (sebagai contoh, `create`, `get`, `drop`, `grant`, `revoke`, dll.) secara implisit dilakukan dalam basis data `admin` baik Anda mengeluarkan perintah terhadap basis data `admin` ataupun tidak.

Pertama, untuk memahami siapa pengguna saat ini dan perannya dalam klaster, Anda dapat menjalankan perintah `show users`, sebagaimana dalam contoh berikut. Anda akan melihat dua pengguna, `serviceadmin` dan pengguna master untuk klaster. Kedua pengguna tersebut selalu ada dan tidak dapat dihapus. Untuk informasi selengkapnya, lihat Mengelola Pengguna Amazon DocumentDB.

```
show users
```

Untuk `user1`, buat peran dengan akses baca dan tulis untuk semua basis data di seluruh klaster dengan perintah berikut ini.

```
db.createUser({user: "user1", pwd: "abc123", roles: [{role: "readWriteAnyDatabase", db:
 "admin"}]})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Untuk `user2`, buat peran dengan akses hanya-baca ke `sample-database-1` basis data dengan perintah berikut ini.

```
db.createUser({user: "user2", pwd: "abc123", roles: [{role: "read", db: "sample-
database-1"}]})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "user":"user2",
   "roles":[
      {
         "role":"read",
         "db":"sample-database-1"
      }
   ]
}
```

Untuk menyimulasikan skenario bahwa `user3` adalah pengguna yang sudah ada, pertama buat `user3` pengguna, dan kemudian tetapkan peran baru ke `user3`.

```
db.createUser({user: "user3", pwd: "abc123", roles: [{role: "readWrite", db: "sample-
database-1"}]})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "user":"user3",
   "roles":[
      {
         "role":"readWrite",
         "db":"sample-database-1"
      }
   ]
}
```

Sekarang setelah `user3` pengguna telah dibuat, tetapkan `user3` peran `read` ke `sample-database-2`.

```
db.grantRolesToUser("user3", [{role: "read", db: "sample-database-2"}])
```

Terakhir, `user1` dan `user2` meninggalkan perusahaan dan akses mereka ke klaster harus dicabut. Anda bisa melakukannya dengan menghapus pengguna, sebagai berikut.

```
db.dropUser("user1")
db.dropUser("user2")
```

Untuk memastikan bahwa semua pengguna memiliki peran yang tepat, Anda dapat mendaftar semua pengguna dengan perintah berikut ini.

```
show users
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
   "_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"
      }
   ]
}
```

Memulai dengan peran yang ditentukan pengguna RBAC

Untuk membantu Anda memulai dengan peran yang ditentukan pengguna, bagian ini memandu Anda melalui skenario contoh penerapan hak istimewa paling sedikit dengan membuat peran untuk tiga pengguna dengan fungsi pekerjaan yang berbeda.

Dalam contoh ini, berikut ini berlaku:

- `user1` adalah manajer baru yang perlu dapat melihat dan mengakses semua basis data dalam klaster.

- `user2`adalah karyawan baru yang hanya membutuhkan tindakan 'temukan' untuk hanya satu database,`sample-database-1`, di cluster yang sama.
- `user3`adalah karyawan yang sudah ada yang perlu melihat dan mengakses koleksi tertentu, col2 dalam database yang berbeda, `sample-database-2` yang sebelumnya tidak mereka akses, di cluster yang sama.
- Untuk `user1`, buat peran dengan akses baca dan tulis untuk semua basis data di seluruh klaster dengan perintah berikut ini.

```
db.createUser(
 {
     user: "user1", pwd: "abc123",
     roles: [{role: "readWriteAnyDatabase", db: "admin"}]
 }
 )
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Untuk`user2`, buat peran dengan hak istimewa 'temukan' untuk semua koleksi dalam database `sample-database-1` dengan perintah berikut. Perhatikan bahwa peran ini akan memastikan bahwa setiap pengguna terkait hanya dapat menjalankan kueri find.

```
db.createRole(
{
    role: "findRole",
    privileges: [
    {
        resource: {db: "sample-database-1", collection: ""}, actions: ["find"]
    }],
    roles: []
```

```
}
)
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
   "role":"findRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "find"
         ]
      }
   ],
   "roles":[

   ]
}
```

Selanjutnya, buat user (`user2`) dan lampirkan peran yang baru dibuat `findRole` ke pengguna.

```
db.createUser(
{
    user: "user2",
    pwd: "abc123",
    roles: []
})

 db.grantRolesToUser("user2",["findRole"])
```

Untuk mensimulasikan skenario yang `user3` merupakan pengguna yang sudah ada, pertama buat penggunauser3, dan kemudian buat peran baru yang disebut CollectionRole yang akan kita lakukan pada langkah berikutnya. `user3`

Sekarang Anda dapat menetapkan peran baru. `user3` Peran baru ini akan memungkinkan `user3` untuk dapat menyisipkan, memperbarui, menghapus, dan menemukan akses ke satu koleksi tertentu col2 di`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: []
}
)
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Sekarang pengguna `user3` telah dibuat, Anda dapat memberikan `user3` perancollectionFind.

```
db.grantRolesToUser("user3",["collectionRole"])
```

Terakhir, `user1` dan `user2` meninggalkan perusahaan dan akses mereka ke klaster harus dicabut. Anda bisa melakukannya dengan menghapus pengguna, sebagai berikut.

```
db.dropUser("user1")
db.dropUser("user2")
```

Untuk memastikan bahwa semua pengguna memiliki peran yang tepat, Anda dapat mendaftar semua pengguna dengan perintah berikut ini.

```
show users
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "_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"
      }
   ]
}
```

Menghubungkan ke Amazon DocumentDB sebagai Pengguna

Saat menghubungkan ke klaster Amazon DocumentDB, Anda terhubung dalam konteks basis data tertentu. Secara default, jika Anda tidak menentukan basis data dalam string koneksi Anda, Anda secara otomatis terhubung ke klaster dalam konteks basis data `test`. Semua perintah tingkat koleksi seperti `insert` dan `find` dikeluarkan terhadap koleksi di basis data `test`.

Untuk melihat basis data dalam konteks Anda dari atau — dengan kata lain — mengeluarkan perintah padanya, gunakan perintah db dalam mongo shell, sebagai berikut.

Kueri:

```
db
```

Output:

```
test
```

Meskipun koneksi default mungkin dalam konteks basis data `test`, hal ini tidak selalu berarti bahwa pengguna yang berkaitan dengan koneksi diotorisasi untuk melakukan tindakan pada basis data `test`. Dalam skenario contoh sebelumnya, jika Anda mengautentikasi sebagai `user3` pengguna, yang memiliki peran `readWrite` untuk basis data `sample-database-1`, konteks default koneksi Anda adalah basis data `test`. Namun demikian, jika Anda mencoba untuk memasukkan dokumen ke koleksi pada basis data `test`, Anda akan menerima pesan kesalahan Kegagalan otorisasi. Hal ini karena pengguna tersebut tidak diotorisasi untuk melakukan perintah tersebut pada database tersebut, sebagaimana ditunjukkan di bawah ini.

Kueri:

```
db
```

Output:

```
test
```

Kueri:

```
db.col.insert({x:1})
```

Output:

```
WriteCommandError({ "ok" : 0, "code" : 13, "errmsg" : "Authorization failure" })
```

Jika Anda mengubah konteks koneksi Anda ke basis data `sample-database-1`, Anda dapat menulis ke koleksi di mana pengguna memiliki otorisasi untuk melakukannya.

Kueri:

```
use sample-database-1
```

Output:

```
switched to db sample-database-1
```

Kueri:

```
db.col.insert({x:1})
```

Output:

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

Ketika Anda mengautentikasi ke klaster dengan pengguna tertentu, Anda juga dapat menentukan basis data dalam string koneksi. Dengan melakukannya akan menghilangkan kebutuhan untuk melakukan perintah `use` setelah pengguna telah diautentikasi ke basis data `admin`.

String koneksi berikut mengautentikasi pengguna terhadap basis data `admin`, tetapi konteks koneksi akan melawan basis data `sample-database-1`.

```
mongo "mongodb://user3:abc123@sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
sample-database-2"
```

Perintah Umum

Bagian ini menyediakan contoh perintah umum menggunakan kontrol akses berbasis peran di Amazon DocumentDB. Anda harus berada dalam konteks basis data `admin` untuk membuat dan memodifikasi pengguna dan peran. Anda dapat menggunakan perintah `use admin` untuk beralih ke basis data `admin`.

> **Note**
>
> Modifikasi untuk pengguna dan peran secara implisit akan terjadi di basis data `admin`. Membuat pengguna dengan peran yang tercakup di semua basis data (sebagai contoh, `readAnyDatabase`) mensyaratkan bahwa Anda dalam konteks basis data `admin` (yaitu, `use admin`) saat membuat pengguna, maupun Anda secara eksplisit menyatakan basis data untuk peran saat membuat pengguna (sebagaimana ditunjukkan dalam Contoh 2 di bagian ini).

Contoh 1: Buat pengguna dengan `read` peran untuk database`foo`.

```
db.createUser({user: "readInFooBar", pwd: "abc123", roles: [{role: "read", db:
 "foo"}]})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "user":"readInFooBar",
   "roles":[
      {
         "role":"read",
         "db":"foo"
      }
   ]
}
```

Contoh 2: Buat pengguna dengan akses baca di semua database.

```
db.createUser({user: "readAllDBs", pwd: "abc123", roles: [{role: "readAnyDatabase", db:
 "admin"}]})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "user":"readAllDBs",
   "roles":[
      {
         "role":"readAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Contoh 3: Berikan `read` peran kepada pengguna yang sudah ada di database baru.

```
db.grantRolesToUser("readInFooBar", [{role: "read", db: "bar"}])
```

Contoh 4: Perbarui peran pengguna.

```
db.updateUser("readInFooBar", {roles: [{role: "read", db: "foo"}, {role: "read", db:
 "baz"}]})
```

Contoh 5: Mencabut akses ke database untuk pengguna.

```
db.revokeRolesFromUser("readInFooBar", [{role: "read", db: "baz"}])
```

Contoh 6: Jelaskan peran bawaan.

```
db.getRole("read", {showPrivileges:true})
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
{
   "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"
         ]
      }
   }
```

Contoh 7: Jatuhkan pengguna dari cluster.

```
db.dropUser("readInFooBar")
```

Output dari operasi ini akan terlihat seperti berikut.

```
true
```

Contoh 8: Buat peran dengan akses baca dan tulis ke koleksi tertentu

```
db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Contoh 9: Buat pengguna dan tetapkan peran yang ditentukan pengguna

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.grantRolesToUser("user3",["collectionRole"])
```

Contoh 10: Berikan hak istimewa tambahan untuk peran yang ditentukan pengguna

```
db.grantPrivilegesToRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col1" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Contoh 11: Hapus hak istimewa dari peran yang ditentukan pengguna

```
db.revokePrivilegesFromRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col2" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Contoh 12: Perbarui peran yang ditentukan pengguna yang ada

```
db.updateRole(
  "collectionRole",
  {
    privileges: [
    {
        resource: {db: "sample-database-3", collection: "sample-collection-3"},
 actions: ["find", "update", "insert", "remove"]
```

```
    }],
    roles: []
  }
)
```

Perbedaan Fungsional

Di Amazon DocumentDB, definisi pengguna dan peran disimpan dalam basis data `admin` dan pengguna diautentikasi terhadap basis data `admin`. Fungsionalitas ini berbeda dari Edisi Komunitas MongoDB, tetapi konsisten dengan MongoDB Atlas.

Amazon DocumentDB juga mendukung perubahan pengaliran, yang menyediakan urutan sesuai waktu dari peristiwa perubahan yang terjadi di dalam koleksi klaster Anda. Tindakan `listChangeStreams` diterapkan pada tingkat klaster (yaitu, di semua basis data), dan tindakan `modifyChangeStreams` dapat diterapkan pada tingkat basis data dan tingkat klaster.

Batas

Tabel berikut berisi batasan untuk Kontrol Akses Berbasis Peran di Amazon DocumentDB.

Deskripsi	Kuota
Jumlah pengguna per klaster	1000
Jumlah peran yang berkaitan dengan pengguna	1000
Jumlah peran yang ditentukan pengguna	100
Jumlah sumber daya yang terkait dengan hak istimewa	100

Akses Database Menggunakan Kontrol Akses Berbasis Peran

Dengan kontrol akses berbasis peran, Anda dapat membuat pengguna dan memberikan satu atau lebih peran untuk menentukan operasi apa yang dapat dilakukan pengguna dalam basis data atau klaster.

Berikut ini adalah daftar peran built-in yang saat ini didukung di Amazon DocumentDB.

> **Note**
>
> Di Amazon DocumentDB 4.0 dan 5.0, `ListCollection` perintah `ListDatabase` dan secara opsional dapat menggunakan `authorizedCollections` parameter `authorizedDatabases` dan untuk mencantumkan koleksi dan database yang diizinkan pengguna untuk mengakses dengan memerlukan `listCollections` peran dan masing-masing. `listDatabase` Juga, pengguna sekarang memiliki kemampuan untuk menghilangkan kursor mereka sendiri tanpa memerlukan peran `KillCursor`.

Database user

Nama peran	Deskripsi	Tindakan

`read`	Memberikan kepada pengguna akses baca ke basis data yang ditetapkan.	`changeStreams`
		`collStats`
		`dbStats`
		`find`
		`killCursors`
		`listIndexes`
		`listCollections`

`readWrite`	Memberikan kepada pengguna akses baca dan tulis ke basis data yang ditetapkan.	Semua tindakan dari izin `read`.
		`createCollection`
		`dropCollection`
		`createIndex`
		`dropIndex`

Nama peran	Deskripsi	Tindakan

		`insert`
		`killCursors`
		`listIndexes`
		`listCollections`
		`remove`
		`update`

Cluster user

Nama peran	Deskripsi	Tindakan

readAnyDatabase	Memberikan kepada pengguna akses baca ke semua basis data dalam klaster.	Semua tindakan dari izin `read`.
		`listChangeStreams`
		`listDatabases`

`readWriteAnyDatabase`	Memberikan kepada pengguna akses baca dan tulis ke semua basis data dalam klaster.	Semua tindakan dari izin `readWrite` .
		`listChangeStreams`
		`listDatabases`

`userAdminAnyDatabase`	Memberikan kepada pengguna kemampuan untuk menetapkan dan memodifikasi peran atau hak istimewa yang dimiliki setiap pengguna untuk basis data yang ditetapkan.	`changeCustomData` `changePassword`

Nama peran	Deskripsi	Tindakan

		`createUser`
		`dropRole`
		`dropUser`
		`grantRole`
		`listDatabases`
		`revokeRole`
		`viewRole`
		`viewUser`

`dbAdminAnyDatabase`	Memberikan kepada pengguna kemampuan untuk melakukan peran administrasi basis data pada setiap basis data yang ditetapkan.	Semua tindakan dari izin `dbAdmin`.
		`dropCollection`
		`listDatabases`
		`listChangeStreams`
		`modifyChangeStreams`

Superuser

Nama peran	Deskripsi	Tindakan
`root`	Memberikan kepada pengguna akses ke sumber daya dan operasi dari semua gabungan peran berikut ini: `readWriteAnyDatabase` , `dbAdminAnyDatabase`	Semua tindakan dari `readWriteAnyDatabase` , `dbAdminAnyDatabase`

Nama peran	Deskripsi	Tindakan
	, `userAdminAnyDatabase` , `clusterAdmin` , `restore`, dan `backup`.	, `userAdminAnyDatabase` , `clusterAdmin` , `restore`, dan `backup`.

Database administrator

Nama peran	Deskripsi	Tindakan

`dbAdmin`	Memberikan kepada pengguna kemampuan untuk melakukan tugas administratif pada basis data yang ditetapkan.	`bypassDocumentValidation`
		`collMod`
		`collStats`
		`createCollection`
		`createIndex`
		`dropCollection`
		`dropDatabase`
		`dropIndex`
		`dbStats`
		`find`
		`killCursors`
		`listIndexes`

Nama peran	Deskripsi	Tindakan

		`listCollections`
		
		`modifyChangeStreams`

`dbOwner`	Memberikan kepada pengguna kemampuan untuk melakukan tugas administratif pada basis data yang ditetapkan dengan menggabungkan peran `dbAdmin` dan `readWrite` .	Semua tindakan dari `dbAdmin` dan `readWrite` .

Cluster administrator

Nama peran	Deskripsi	Tindakan

`clusterAdmin`	Memberikan kepada pengguna akses manajemen klaster terbesar dengan menggabungkan peran `clusterManager` , `clusterMonitor` , dan `hostManager` .	Semua tindakan dari `clusterManager` , `clusterMonitor` , dan `hostManager` .
		
		`listChangeStreams`
		
		`dropDatabase`
		
		`modifyChangeStreams`

`clusterManager`	Memberikan kepada pengguna kemampuan untuk mengambil tindakan manajemen dan pemantauan pada klaster yang ditetapkan.	`listChangeStreams`
		
		`listSessions`
		
		`modifyChangeStreams`

Nama peran	Deskripsi	Tindakan
		`replSetGetConfig`

`clusterMonitor`	Memberikan kepada pengguna kemampuan untuk memiliki akses hanya-baca ke alat pemantauan.	`collStats`
		`dbStats`
		`find`
		`getParameter`
		`hostInfo`
		`indexStats`
		`killCursors`
		`listChangeStreams`
		`listCollections`
		`listDatabases`
		`listIndexes`
		`listSessions`
		`replSetGetConfig`
		`serverStatus`
		`top`

Nama peran	Deskripsi	Tindakan

`hostManager`	Memberikan kepada pengguna kemampuan untuk memantau dan mengelola server.	`killCursors`
		`killAnyCursor`
		`killAnySession`
		`killop`

Backup administrator

Nama peran	Deskripsi	Tindakan

`backup`	Memberikan kepada pengguna akses yang diperlukan untuk mencadangkan data.	`getParameter`
		`insert`
		`find`
		`listChangeStreams`
		`listCollections`
		`listDatabases`
		`listIndexes`
		`update`

`restore`	Memberikan kepada pengguna akses yang diperlukan untuk memulihkan data.	`bypassDocumentValidation`
		`changeCustomData`
		`changePassword`

Nama peran	Deskripsi	Tindakan
		collMod createCollection createIndex createUser dropCollection dropRole dropUser getParameter grantRole find insert listCollections modifyChangeStreams revokeRole remove viewRole viewUser update

Pencatatan dan Pemantauan di Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) menyediakan berbagai CloudWatch metrik Amazon yang dapat Anda pantau untuk menentukan kondisi dan performa klaster dan instans Amazon DocumentDB Anda. Anda dapat melihat metrik Amazon DocumentDB menggunakan berbagai alat, termasuk konsol Amazon DocumentDBAWS CLI,, CloudWatch konsol Amazon, dan CloudWatch API. Untuk informasi selengkapnya tentang pemantauan, lihat Memantau Amazon DocumentDB.

Selain CloudWatch metrik Amazon, Anda dapat menggunakan profiler untuk mencatat waktu eksekusi dan detail operasi yang dilakukan di klaster Anda. Profiler berguna untuk memantau operasi paling lambat di klaster Anda untuk membantu Anda meningkatkan performa kueri individual dan performa klaster secara keseluruhan. Saat diaktifkan, operasi dicatat ke Amazon CloudWatch Logs dan Anda dapat menggunakan CloudWatch Insight untuk menganalisis, memantau, dan mengarsipkan data profil Amazon DocumentDB Anda. Untuk informasi selengkapnya, lihat Membuat Profil Operasi Amazon DocumentDB.

Amazon DocumentDB juga terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWS layanan di Amazon DocumentDB (dengan kompatibilitas MongoDB). CloudTrail menangkap semua panggilanAWS CLI API untuk Amazon DocumentDB sebagai peristiwa, termasuk panggilan dari Amazon DocumentDBAWS Management Console dan dari panggilan kode ke Amazon DocumentDB SDK. Untuk informasi selengkapnya, lihat Mencatat Panggilan API Amazon DocumentDB dengan AWS CloudTrail.

Dengan Amazon DocumentDB, Anda dapat mengaudit peristiwa yang dilakukan di klaster Anda. Contoh log acara termasuk upaya autentikasi yang berhasil dan gagal, membuang koleksi dalam basis data, atau membuat indeks. Secara default, pengauditan dinonaktifkan di Amazon DocumentDB dan mengharuskan Anda untuk ikut serta dalam fitur ini. Untuk informasi selengkapnya, lihat Mengaudit Peristiwa Amazon DocumentDB.

Memperbarui Sertifikat TLS Amazon DocumentDB Anda

Topik

- Memperbarui Aplikasi dan Klaster Amazon DocumentDB Anda
- Pemecahan Masalah
- Pertanyaan yang Sering Diajukan

Sertifikat otoritas sertifikat (CA) untuk klaster Amazon DocumentDB akan diperbarui mulai Agustus 2024. Jika Anda menggunakan klaster Amazon DocumentDB dengan Keamanan Lapisan Pengangkutan (TLS) diaktifkan (pengaturan default) dan Anda belum merotasi aplikasi klien dan sertifikat server, langkah-langkah berikut diperlukan untuk mengurangi masalah konektivitas antara aplikasi Anda dan klaster Amazon DocumentDB Anda.

- Langkah 1: Unduh Sertifikat CA Baru dan Perbarui Aplikasi Anda
- Langkah 2: Perbarui Sertifikat Server

Sertifikat CA dan server diperbarui sebagai bagian dari praktik terbaik pemeliharaan dan keamanan standar untuk Amazon DocumentDB. Aplikasi klien harus menambahkan sertifikat CA baru ke penyimpanan kepercayaan mereka, dan instans Amazon DocumentDB yang ada harus diperbarui untuk menggunakan sertifikat CA baru sebelum tanggal kedaluwarsa ini.

Memperbarui Aplikasi dan Klaster Amazon DocumentDB Anda

Ikuti langkah-langkah di bagian ini untuk memperbarui paket sertifikat CA aplikasi Anda (Langkah 1) dan sertifikat server klaster Anda (Langkah 2). Sebelum Anda menerapkan perubahan ke lingkungan produksi, kami sangat merekomendasikan untuk menguji langkah-langkah ini di lingkungan pengembangan atau penahapan.

> **Note**
>
> Anda harus menyelesaikan Langkah 1 dan 2 Wilayah AWS di masing-masing tempat Anda memiliki cluster Amazon DocumentDB.

Langkah 1: Unduh Sertifikat CA Baru dan Perbarui Aplikasi Anda

Unduh sertifikat CA baru dan perbarui aplikasi Anda untuk menggunakan sertifikat CA baru guna membuat koneksi TLS ke Amazon DocumentDB. Unduh paket sertifikat CA baru dari https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem. Operasi ini mengunduh file bernama `global-bundle.pem`.

> **Note**
>
> Jika Anda mengakses keystore yang berisi sertifikat CA lama (`rds-ca-2019-root.pem`) dan sertifikat CA baru (`rds-ca-rsa2048-g1, rds-ca-rsa4096-g1`), verifikasi bahwa keystore memilih. `global-bundle`

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Selanjutnya, perbarui aplikasi Anda untuk menggunakan paket sertifikat baru. Bundel CA baru berisi sertifikat CA lama (rds-ca-2019) dan sertifikat CA baru (2048-g1, 4096-g1). rds-ca-rsa rds-ca-rsa Dengan memiliki kedua sertifikat CA di paket CA baru, Anda dapat memperbarui aplikasi dan klaster dalam dua langkah.

Untuk memverifikasi bahwa aplikasi Anda menggunakan paket sertifikat CA terbaru, lihat Bagaimana saya bisa yakin bahwa saya menggunakan paket CA terbaru?. Jika Anda sudah menggunakan paket sertifikat CA terbaru di aplikasi Anda, Anda dapat melompat ke langkah 2.

Untuk contoh penggunaan paket CA dengan aplikasi Anda, lihat Mengenkripsi Data dalam Transit dan Menghubungkan dengan TLS yang Diaktifkan.

> **Note**
>
> Saat ini, MongoDB Go Driver 1.2.1 hanya menerima satu sertifikat server CA di `sslcertificateauthorityfile`. Silakan lihat Menghubungkan dengan TLS yang Diaktifkan untuk menghubungkan ke Amazon DocumentDB menggunakan Go ketika TLS diaktifkan.

Langkah 2: Perbarui Sertifikat Server

Setelah aplikasi telah diperbarui untuk menggunakan paket CA baru, langkah berikutnya adalah untuk memperbarui sertifikat server dengan memodifikasi setiap instans dalam sebuah klaster Amazon DocumentDB. Untuk mengubah instans untuk menggunakan sertifikat server baru, lihat petunjuk berikut.

Amazon DocumentDB menyediakan CA berikut untuk menandatangani sertifikat server DB untuk instans DB:

- rds-ca-rsa2048-g1 — Menggunakan otoritas sertifikat dengan algoritma kunci pribadi RSA 2048 dan algoritma penandatanganan SHA256 di sebagian besar wilayah. AWS CA ini mendukung rotasi sertifikat server otomatis.
- rds-ca-rsa4096-g1 —Menggunakan otoritas sertifikat dengan algoritma kunci pribadi RSA 4096 dan algoritma penandatanganan SHA384. CA ini mendukung rotasi sertifikat server otomatis.

> **Note**
>
> Jika Anda menggunakan AWS CLI, Anda dapat melihat validitas otoritas sertifikat yang tercantum di atas dengan menggunakan deskripsi-sertifikat.

Sertifikat CA ini termasuk dalam paket sertifikat regional dan global. Saat Anda menggunakan CA rds-ca-rsa 2048-g1 atau rds-ca-rsa 4096-g1 dengan database, Amazon DocumentDB mengelola sertifikat server DB pada database. Amazon DocumentDB memutar sertifikat server DB secara otomatis sebelum kedaluwarsa (mungkin memerlukan reboot).

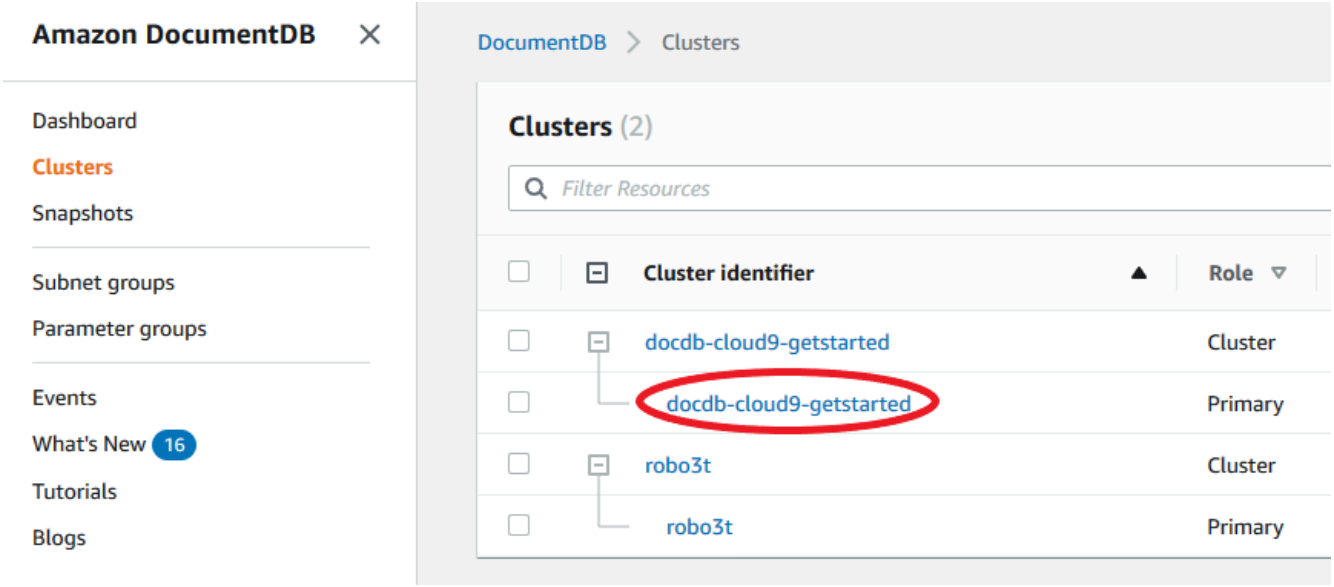
> **Note**
>
> Memperbarui instans Anda memerlukan boot ulang, yang dapat menyebabkan gangguan layanan. Sebelum memperbarui sertifikat server, pastikan Anda telah menyelesaikan Langkah 1.

Using the AWS Management Console

Selesaikan langkah-langkah berikut untuk mengidentifikasi dan memutar sertifikat server lama untuk instans Amazon DocumentDB Anda yang ada menggunakan AWS Management Console.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat klaster Anda Wilayah AWS berada.
3. Di panel navigasi di sisi kiri konsol, pilih Klaster.
4. Anda mungkin perlu mengidentifikasi instans mana yang masih di sertifikat server lama (`rds-ca-2019`). Anda dapat melakukan ini di kolom otoritas Sertifikat yang terletak di paling kanan tabel Clusters.

5. Dalam tabel Clusters, Anda akan melihat kolom Cluster identifier di paling kiri. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



6. Centang kotak di sebelah kiri instans yang Anda minati.
7. Pilih Tindakan, dan kemudian pilih Modifikasi.
8. Di bawah Otoritas sertifikat, pilih sertifikat server baru (`rds-ca-rsa2048-g1`) untuk instans ini.
9. Anda dapat melihat ringkasan perubahan pada halaman berikutnya. Perhatikan bahwa ada peringatan tambahan untuk mengingatkan Anda untuk memastikan bahwa aplikasi Anda menggunakan paket sertifikat CA terbaru sebelum memodifikasi instans untuk menghindari menyebabkan gangguan dalam konektivitas.
10. Anda dapat memilih untuk menerapkan modifikasi selama jendela pemeliharaan berikutnya atau menerapkan dengan segera. Jika maksud Anda adalah untuk mengubah sertifikat server dengan segera, gunakan opsi Terapkan Segera.
11. Pilih Modifikasi instans untuk menyelesaikan pembaruan.

Using the AWS CLI

Selesaikan langkah-langkah berikut untuk mengidentifikasi dan memutar sertifikat server lama untuk instans Amazon DocumentDB Anda yang ada menggunakan AWS CLI.

1. Untuk memodifikasi instans dengan segera, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Untuk memodifikasi instans dalam klaster Anda agar menggunakan sertifikat CA baru selama jendela pemeliharaan klaster Anda berikutnya, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Pemecahan Masalah

Jika Anda mengalami masalah saat menyambungkan ke klaster sebagai bagian dari rotasi sertifikat, kami menyarankan hal berikut:

- Reboot instance Anda. Memutar sertifikat baru mengharuskan Anda mem-boot ulang setiap instans Anda. Jika Anda menerapkan sertifikat baru untuk satu atau lebih instans tetapi tidak mem-boot ulangnya, boot ulang instans Anda untuk menerapkan sertifikat baru. Untuk informasi selengkapnya, lihat Mem-boot ulang instans Amazon DocumentDB.
- Verifikasi bahwa klien Anda menggunakan bundel sertifikat terbaru. Lihat Bagaimana saya bisa yakin bahwa saya menggunakan paket CA terbaru?.
- Verifikasi bahwa instans Anda menggunakan sertifikat terbaru. Lihat Bagaimana cara mengetahui instans Amazon DocumentDB saya yang mana yang menggunakan sertifikat server lama/baru?.
- Verifikasi bahwa sertifikat CA terbaru sedang digunakan oleh aplikasi Anda. Beberapa driver, seperti Java dan Go, memerlukan kode tambahan untuk mengimpor beberapa sertifikat dari paket sertifikat ke penyimpanan kepercayaan. Untuk informasi selengkapnya tentang menghubungkan ke Amazon DocumentDB dengan TLS, lihat Menghubungkan Secara Terprogram ke Amazon DocumentDB.
- Hubungi dukungan. Jika Anda memiliki pertanyaan atau masalah, hubungi AWS Support.

Pertanyaan yang Sering Diajukan

Berikut ini adalah jawaban atas beberapa pertanyaan umum tentang sertifikat TLS.

Bagaimana jika saya memiliki pertanyaan atau masalah?

Jika Anda memiliki pertanyaan atau masalah, hubungi AWS Support.

Bagaimana saya tahu apakah saya menggunakan TLS untuk terhubung ke klaster Amazon DocumentDB saya?

Anda dapat menentukan apakah klaster Anda menggunakan TLS dengan memeriksa parameter `tls` untuk grup parameter klaster dari klaster Anda. Jika parameter `tls` diatur ke `enabled`, Anda menggunakan sertifikat TLS untuk terhubung ke klaster Anda. Untuk informasi selengkapnya, lihat Mengelola grup parameter cluster Amazon DocumentDB.

Mengapa Anda memperbarui sertifikat CA dan server?

Amazon DocumentDB CA dan sertifikat server sedang diperbarui sebagai bagian dari pemeliharaan standar dan praktik terbaik keamanan untuk Amazon DocumentDB. Sertifikat CA dan server saat ini akan kedaluwarsa mulai Agustus 2024.

Apa yang terjadi jika saya tidak mengambil tindakan apa pun pada tanggal kedaluwarsa?

Jika Anda menggunakan TLS untuk terhubung ke cluster Amazon DocumentDB Anda dan Anda tidak membuat 2024 Agustus 2024, aplikasi Anda yang terhubung melalui TLS tidak akan lagi dapat berkomunikasi dengan cluster Amazon DocumentDB.

Amazon DocumentDB tidak akan memutar sertifikat database Anda secara otomatis sebelum kedaluwarsa. Anda harus memperbarui aplikasi dan cluster Anda untuk menggunakan sertifikat CA baru sebelum atau setelah tanggal kedaluwarsa.

Bagaimana cara mengetahui instans Amazon DocumentDB saya yang mana yang menggunakan sertifikat server lama/baru?

Untuk mengidentifikasi instans Amazon DocumentDB yang masih menggunakan sertifikat server lama, Anda dapat menggunakan Amazon DocumentDB atau file. AWS Management Console AWS CLI

Menggunakan AWS Management Console

Untuk mengidentifikasi instans di klaster Anda yang menggunakan sertifikat yang lebih lama

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat instance Anda Wilayah AWS berada.
3. Di panel navigasi di sisi kiri konsol, pilih Klaster.
4. Kolom otoritas Sertifikat (dekat paling kanan tabel) menunjukkan instance mana yang masih ada di sertifikat server lama (`rds-ca-2019`) dan sertifikat server baru (`rds-ca-rsa2048-g1`).

Menggunakan AWS CLI

Untuk mengidentifikasi instans di klaster Anda yang menggunakan sertifikat server yang lebih lama, gunakan perintah `describe-db-clusters` dengan berikut ini.

```
aws docdb describe-db-instances \
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Bagaimana cara memodifikasi instans individu di klaster Amazon DocumentDB saya untuk memperbarui sertifikat server?

Kami merekomendasikan Anda memperbarui sertifikat server untuk semua instans dalam klaster tertentu pada waktu yang sama. Untuk memodifikasi instans di klaster Anda, Anda dapat menggunakan konsol atau AWS CLI.

> **Note**
>
> Memperbarui instans Anda memerlukan boot ulang, yang dapat menyebabkan gangguan layanan. Sebelum memperbarui sertifikat server, pastikan Anda telah menyelesaikan Langkah 1.

Menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat klaster Anda Wilayah AWS berada.
3. Di panel navigasi di sisi kiri konsol, pilih Klaster.
4. Kolom otoritas Sertifikat (dekat paling kanan tabel) menunjukkan instance mana yang masih ada di sertifikat server lama (`rds-ca-2019`).
5. Dalam tabel Clusters, di bawah Cluster identifier, pilih instance untuk memodifikasi.
6. Pilih Tindakan, dan kemudian pilih Modifikasi.
7. Di bawah Otoritas sertifikat, pilih sertifikat server baru (`rds-ca-rsa2048-g1`) untuk instans ini.
8. Anda dapat melihat ringkasan perubahan pada halaman berikutnya. Perhatikan bahwa ada peringatan tambahan untuk mengingatkan Anda untuk memastikan bahwa aplikasi Anda menggunakan paket sertifikat CA terbaru sebelum memodifikasi instans untuk menghindari menyebabkan gangguan dalam konektivitas.
9. Anda dapat memilih untuk menerapkan modifikasi selama jendela pemeliharaan berikutnya atau menerapkan dengan segera.
10. Pilih Modifikasi instans untuk menyelesaikan pembaruan.

Menggunakan AWS CLI

Selesaikan langkah-langkah berikut untuk mengidentifikasi dan memutar sertifikat server lama untuk instans Amazon DocumentDB Anda yang ada menggunakan AWS CLI.

1. Untuk memodifikasi instans dengan segera, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Untuk memodifikasi instans dalam klaster Anda agar menggunakan sertifikat CA baru selama jendela pemeliharaan klaster Anda berikutnya, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Apa yang terjadi jika saya menambahkan instans baru ke klaster yang ada?

Semua instans baru yang dibuat menggunakan sertifikat server lama dan memerlukan koneksi TLS menggunakan sertifikat CA lama. Instans Amazon DocumentDB baru yang dibuat setelah 25 Januari 2024 akan default menggunakan sertifikat baru 2048-g1. rds-ca-rsa

Apa yang terjadi jika ada penggantian instans atau failover di klaster saya?

Jika ada pengganti instans di klaster Anda, instans baru yang dibuat akan terus menggunakan sertifikat server yang sama dengan yang digunakan instans sebelumnya. Kami merekomendasikan Anda memperbarui sertifikat server untuk semua instans pada waktu yang sama. Jika terjadi failover di klaster, sertifikat server pada primer baru digunakan.

Jika saya tidak menggunakan TLS untuk terhubung ke klaster saya, apakah saya masih perlu memperbarui setiap instans saya?

Jika Anda tidak menggunakan TLS untuk terhubung ke klaster Amazon DocumentDB Anda, tidak ada tindakan yang diperlukan.

Jika saya tidak menggunakan TLS untuk terhubung ke klaster saya tetapi saya berencana untuk melakukannya di masa mendatang, apa yang harus saya lakukan?

Jika Anda membuat klaster sebelum Januari 2024, ikuti Langkah 1 dan Langkah 2 di bagian sebelumnya untuk memastikan bahwa aplikasi Anda menggunakan bundel CA yang diperbarui, dan bahwa setiap instans Amazon DocumentDB menggunakan sertifikat server terbaru. Jika Anda membuat cluster setelah 25 Januari 2024, cluster Anda sudah memiliki sertifikat server terbaru (rds-ca-rsa2048-g1). Untuk memverifikasi bahwa aplikasi Anda menggunakan paket CA terbaru, lihat Jika saya tidak menggunakan TLS untuk terhubung ke klaster saya, apakah saya masih perlu memperbarui setiap instans saya?

Bisakah tenggat waktu diperpanjang melampaui Agustus 2024?

Jika aplikasi Anda terhubung melalui TLS, batas waktu tidak dapat diperpanjang.

Bagaimana saya bisa yakin bahwa saya menggunakan paket CA terbaru?

Untuk memverifikasi bahwa Anda memiliki bundel terbaru, gunakan perintah berikut. Untuk menjalankan perintah ini, Anda harus menginstal java dan alat java harus berada dalam variabel PATH shell Anda. Untuk informasi selengkapnya, lihat Menggunakan Java

macOS dan Amazon Linux

```
keytool -printcert -v -file global-bundle.pem
```

Windows

```
keytool -printcert -v -file global-bundle.p7b
```

Mengapa saya melihat “RDS” dalam nama paket CA?

Untuk fitur pengelolaan tertentu, seperti manajemen sertifikat, Amazon DocumentDB menggunakan teknologi operasional yang dibagikan dengan Amazon Relational Database Service (Amazon RDS).

Kapan sertifikat baru akan kedaluwarsa?

Sertifikat server baru akan kedaluwarsa (umumnya) sebagai berikut:

- rds-ca-rsa2048-g1 —Kedaluwarsa 2061
- rds-ca-rsa4096-g1 —Kedaluwarsa 2121

Jika saya menerapkan sertifikat server baru, dapatkah saya mengembalikannya ke sertifikat server lama?

Jika Anda perlu mengembalikan instans ke sertifikat server lama, kami merekomendasikan Anda melakukannya untuk semua instans di klaster. Anda dapat mengembalikan sertifikat server untuk setiap instance dalam klaster dengan menggunakan AWS Management Console atau. AWS CLI

Menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat klaster Anda Wilayah AWS berada.
3. Di panel navigasi di sisi kiri konsol, pilih Klaster.
4. Dalam tabel Clusters, di bawah Cluster identifier, pilih instance untuk memodifikasi. Pilih Tindakan, dan kemudian pilih Modifikasi.
5. Di bawah Otoritas sertifikat, Anda dapat memilih sertifikat server lama (`rds-ca-2019`).
6. Pilih Lanjutkan dan lihat ringkasan dari modifikasi Anda.

7. Di halaman yang dihasilkan ini, Anda dapat memilih untuk menjadwalkan modifikasi Anda untuk diterapkan di jendela pemeliharaan berikutnya atau segera menerapkan modifikasi Anda. Tentukan pilihan Anda, dan pilih Modifikasi instans.

> **Note**
>
> Jika Anda memilih untuk menerapkan modifikasi Anda dengan segera, perubahan apa pun pada antrean modifikasi yang tertunda juga akan diterapkan. Jika salah satu dari modifikasi yang tertunda memerlukan waktu henti, memilih opsi ini dapat menyebabkan waktu henti yang tidak terduga.

Menggunakan AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2019 <--apply-immediately | --no-apply-immediately>
```

Jika Anda memilih `--no-apply-immediately`, perubahan akan diterapkan selama jendela pemeliharaan klaster berikutnya.

Jika saya memulihkan dari snapshot atau pemulihan titik waktu, apakah itu akan memiliki sertifikat server baru?

Jika Anda memulihkan snapshot atau melakukan point-in-time pemulihan setelah Agustus 2024, cluster baru yang dibuat akan menggunakan sertifikat CA baru.

Bagaimana jika saya mengalami masalah saat terhubung langsung ke cluster Amazon DocumentDB saya dari Mac OS mana pun?

Mac OS telah memperbarui persyaratan untuk sertifikat tepercaya. Sertifikat tepercaya sekarang harus berlaku selama 397 hari atau kurang (lihathttps://support.apple.com/en-us/HT211025).

> **Note**
>
> Pembatasan ini diamati pada versi Mac OS yang lebih baru.

Sertifikat instans Amazon DocumentDB berlaku selama lebih dari empat tahun, lebih lama dari maksimum Mac OS. Untuk terhubung langsung ke cluster Amazon DocumentDB dari komputer yang

menjalankan Mac OS, Anda harus mengizinkan sertifikat yang tidak valid saat membuat koneksi TLS. Dalam hal ini, sertifikat yang tidak valid berarti bahwa masa berlaku lebih dari 397 hari. Anda harus memahami risikonya sebelum mengizinkan sertifikat yang tidak valid saat menghubungkan ke klaster Amazon DocumentDB Anda.

Untuk terhubung ke cluster Amazon DocumentDB dari Mac OS menggunakan, gunakan AWS CLI parameter. `tlsAllowInvalidCertificates`

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Memperbarui Sertifikat Amazon DocumentDB TLS Anda - (AS-Barat) GovCloud

> **Note**
>
> Informasi ini hanya berlaku untuk pengguna di wilayah GovCloud (AS-Barat).

Sertifikat otoritas sertifikat (CA) untuk klaster Amazon DocumentDB (dengan kompatibilitas MongoDB) akan diperbarui pada 18 Mei 2022. Jika Anda menggunakan klaster Amazon DocumentDB dengan Keamanan Lapisan Pengangkutan (TLS) diaktifkan (pengaturan default) dan Anda belum merotasi aplikasi klien dan sertifikat server, langkah-langkah berikut diperlukan untuk mengurangi masalah konektivitas antara aplikasi Anda dan klaster Amazon DocumentDB Anda.

- Langkah 1: Unduh Sertifikat CA Baru dan Perbarui Aplikasi Anda
- Langkah 2: Perbarui Sertifikat Server

Sertifikat CA dan server diperbarui sebagai bagian dari praktik terbaik pemeliharaan dan keamanan standar untuk Amazon DocumentDB. Sertifikat CA sebelumnya akan kedaluwarsa pada 18 Mei 2022. Aplikasi klien harus menambahkan sertifikat CA baru ke penyimpanan kepercayaan mereka, dan instans Amazon DocumentDB yang ada harus diperbarui untuk menggunakan sertifikat CA baru sebelum tanggal kedaluwarsa ini.

Memperbarui Aplikasi dan Klaster Amazon DocumentDB Anda

Ikuti langkah-langkah di bagian ini untuk memperbarui paket sertifikat CA aplikasi Anda (Langkah 1) dan sertifikat server klaster Anda (Langkah 2). Sebelum Anda menerapkan perubahan ke lingkungan produksi, kami sangat merekomendasikan untuk menguji langkah-langkah ini di lingkungan pengembangan atau penahapan.

> **Note**
>
> Anda harus menyelesaikan Langkah 1 dan 2 Wilayah AWS di masing-masing tempat Anda memiliki cluster Amazon DocumentDB.

Langkah 1: Unduh Sertifikat CA Baru dan Perbarui Aplikasi Anda

Unduh sertifikat CA baru dan perbarui aplikasi Anda untuk menggunakan sertifikat CA baru guna membuat koneksi TLS ke Amazon DocumentDB. Unduh paket sertifikat CA baru dari https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem. Operasi ini mengunduh file bernama `us-gov-west-1-bundle.pem`.

> **Note**
>
> Jika Anda mengakses keystore yang berisi sertifikat CA lama (`rds-ca-2017-root.pem`) dan sertifikat CA baru (`rds-ca-rsa4096-g1.pem`), verifikasi bahwa keystore memilih `CA-RSA4096-G1`.

```
wget https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem
```

Selanjutnya, perbarui aplikasi Anda untuk menggunakan paket sertifikat baru. Bundel CA baru berisi sertifikat CA lama dan sertifikat CA baru (`rds-ca-rsa4096-g1.pem`). Dengan memiliki kedua sertifikat CA di paket CA baru, Anda dapat memperbarui aplikasi dan klaster dalam dua langkah.

Setiap unduhan bundel sertifikat CA setelah 21 Desember 2021 harus menggunakan bundel sertifikat CA baru. Untuk memverifikasi bahwa aplikasi Anda menggunakan paket sertifikat CA terbaru, lihat Bagaimana saya bisa yakin bahwa saya menggunakan paket CA terbaru?. Jika Anda sudah menggunakan paket sertifikat CA terbaru di aplikasi Anda, Anda dapat melompat ke langkah 2.

Untuk contoh penggunaan paket CA dengan aplikasi Anda, lihat Mengenkripsi Data dalam Transit dan Menghubungkan dengan TLS yang Diaktifkan.

> **Note**
>
> Saat ini, MongoDB Go Driver 1.2.1 hanya menerima satu sertifikat server CA di `sslcertificateauthorityfile`. Silakan lihat Menghubungkan dengan TLS yang Diaktifkan untuk menghubungkan ke Amazon DocumentDB menggunakan Go ketika TLS diaktifkan.

Langkah 2: Perbarui Sertifikat Server

Setelah aplikasi telah diperbarui untuk menggunakan paket CA baru, langkah berikutnya adalah untuk memperbarui sertifikat server dengan memodifikasi setiap instans dalam sebuah klaster Amazon DocumentDB. Untuk mengubah instans untuk menggunakan sertifikat server baru, lihat petunjuk berikut.

> **Note**
>
> Memperbarui instans Anda memerlukan boot ulang, yang dapat menyebabkan gangguan layanan. Sebelum memperbarui sertifikat server, pastikan Anda telah menyelesaikan Langkah 1.

Using the AWS Management Console

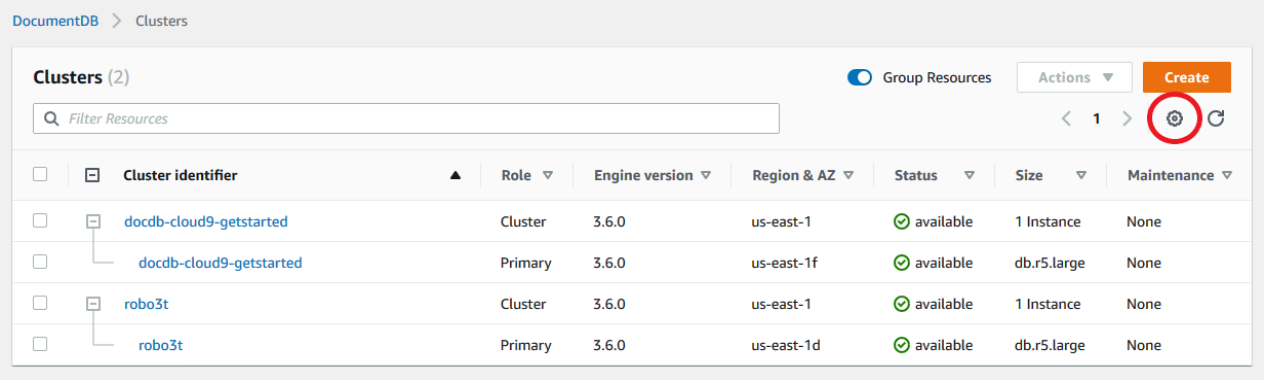
Selesaikan langkah-langkah berikut untuk mengidentifikasi dan memutar sertifikat server lama untuk instans Amazon DocumentDB Anda yang ada menggunakan AWS Management Console.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat klaster Anda Wilayah AWS berada.
3. wh

   Di panel navigasi di sisi kiri konsol, pilih Klaster.

4. Anda mungkin perlu mengidentifikasi instans mana yang masih di sertifikat server lama (`rds-ca-2017`). Anda dapat melakukan ini di kolom Otoritas sertifikat yang tersembunyi secara default. Untuk menampilkan kolom Otoritas sertifikat, lakukan hal berikut:

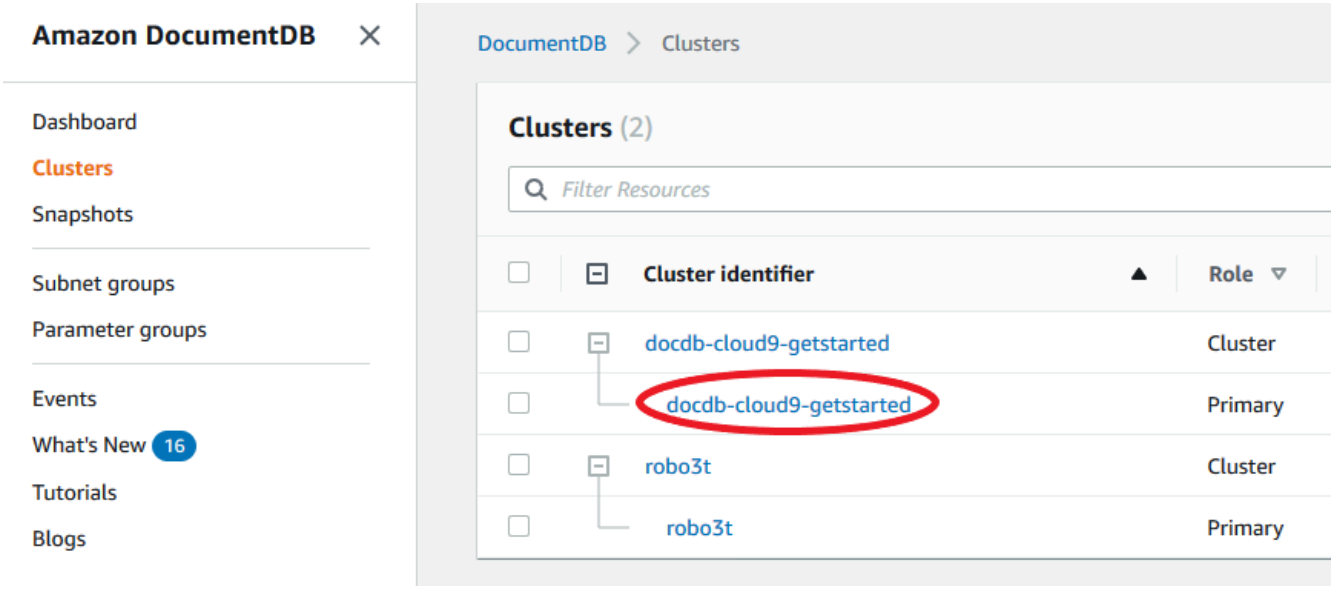
   a. Pilih ikon Pengaturan.



   b. Di bawah daftar kolom yang terlihat, pilih kolom Otoritas sertifikat.

   c. Pilih Konfirmasi untuk menyimpan perubahan Anda.

5. Sekarang kembali ke kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi Klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



6. Centang kotak di sebelah kiri instans yang Anda minati.

7. Pilih Tindakan, dan kemudian pilih Modifikasi.

8. Di bawah Otoritas sertifikat, pilih sertifikat server baru (`rds-ca-rsa4096-g1`) untuk instans ini.

9. Anda dapat melihat ringkasan perubahan pada halaman berikutnya. Perhatikan bahwa ada peringatan tambahan untuk mengingatkan Anda untuk memastikan bahwa aplikasi Anda menggunakan paket sertifikat CA terbaru sebelum memodifikasi instans untuk menghindari menyebabkan gangguan dalam konektivitas.
10. Anda dapat memilih untuk menerapkan modifikasi selama jendela pemeliharaan berikutnya atau menerapkan dengan segera. Jika maksud Anda adalah untuk mengubah sertifikat server dengan segera, gunakan opsi Terapkan Segera.
11. Pilih Modifikasi instans untuk menyelesaikan pembaruan.

Using the AWS CLI

Selesaikan langkah-langkah berikut untuk mengidentifikasi dan memutar sertifikat server lama untuk instans Amazon DocumentDB Anda yang ada menggunakan AWS CLI.

1. Untuk memodifikasi instans dengan segera, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Untuk memodifikasi instans dalam klaster Anda agar menggunakan sertifikat CA baru selama jendela pemeliharaan klaster Anda berikutnya, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Pemecahan Masalah

Jika Anda mengalami masalah saat menyambungkan ke klaster sebagai bagian dari rotasi sertifikat, kami menyarankan hal berikut:

- Reboot instance Anda. Memutar sertifikat baru mengharuskan Anda mem-boot ulang setiap instans Anda. Jika Anda menerapkan sertifikat baru untuk satu atau lebih instans tetapi tidak mem-boot ulangnya, boot ulang instans Anda untuk menerapkan sertifikat baru. Untuk informasi selengkapnya, lihat Mem-boot ulang instans Amazon DocumentDB.

- Verifikasi bahwa klien Anda menggunakan bundel sertifikat terbaru. Lihat Bagaimana saya bisa yakin bahwa saya menggunakan paket CA terbaru?.
- Verifikasi bahwa instans Anda menggunakan sertifikat terbaru. Lihat Bagaimana cara mengetahui instans Amazon DocumentDB saya yang mana yang menggunakan sertifikat server lama/baru?.
- Verifikasi bahwa sertifikat CA terbaru sedang digunakan oleh aplikasi Anda. Beberapa driver, seperti Java dan Go, memerlukan kode tambahan untuk mengimpor beberapa sertifikat dari paket sertifikat ke penyimpanan kepercayaan. Untuk informasi selengkapnya tentang menghubungkan ke Amazon DocumentDB dengan TLS, lihat Menghubungkan Secara Terprogram ke Amazon DocumentDB.
- Hubungi dukungan. Jika Anda memiliki pertanyaan atau masalah, hubungi AWS Support.

Pertanyaan yang Sering Diajukan

Berikut ini adalah jawaban atas beberapa pertanyaan umum tentang sertifikat TLS.

Bagaimana jika saya memiliki pertanyaan atau masalah?

Jika Anda memiliki pertanyaan atau masalah, hubungi AWS Support.

Bagaimana saya tahu apakah saya menggunakan TLS untuk terhubung ke klaster Amazon DocumentDB saya?

Anda dapat menentukan apakah klaster Anda menggunakan TLS dengan memeriksa parameter `tls` untuk grup parameter klaster dari klaster Anda. Jika parameter `tls` diatur ke `enabled`, Anda menggunakan sertifikat TLS untuk terhubung ke klaster Anda. Untuk informasi selengkapnya, lihat Mengelola grup parameter cluster Amazon DocumentDB.

Mengapa Anda memperbarui sertifikat CA dan server?

Sertifikat CA Amazon DocumentDB dan server diperbarui sebagai bagian dari praktik terbaik pemeliharaan dan keamanan standar untuk Amazon DocumentDB. Sertifikat CA dan server saat ini akan kedaluwarsa pada hari Rabu, 18 Mei 2022.

Apa yang terjadi jika saya tidak mengambil tindakan apa pun pada tanggal kedaluwarsa?

Jika Anda menggunakan TLS untuk terhubung ke cluster Amazon DocumentDB Anda dan Anda tidak membuat perubahan pada 18 Mei 2022, aplikasi Anda yang terhubung melalui TLS tidak akan lagi dapat berkomunikasi dengan cluster Amazon DocumentDB.

Amazon DocumentDB tidak akan memutar sertifikat database Anda secara otomatis sebelum kedaluwarsa. Anda harus memperbarui aplikasi dan cluster Anda untuk menggunakan sertifikat CA baru sebelum atau setelah tanggal kedaluwarsa.

Bagaimana cara mengetahui instans Amazon DocumentDB saya yang mana yang menggunakan sertifikat server lama/baru?

Untuk mengidentifikasi instans Amazon DocumentDB yang masih menggunakan sertifikat server lama, Anda dapat menggunakan Amazon DocumentDB atau file. AWS Management Console AWS CLI

Menggunakan AWS Management Console

Untuk mengidentifikasi instans di klaster Anda yang menggunakan sertifikat yang lebih lama

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat instance Anda Wilayah AWS berada.
3. Dalam panel navigasi di sisi kiri konsol, pilih Instans.
4. Kolom Otoritas sertifikat (tersembunyi secara default) menunjukkan instans mana yang masih di sertifikat server lama (`rds-ca-2017`) dan sertifikat server baru (`rds-ca-rsa4096-g1`). Untuk menampilkan kolom Otoritas sertifikat, lakukan hal berikut:
   a. Pilih ikon Pengaturan.
   b. Di bawah daftar kolom yang terlihat, pilih kolom Otoritas sertifikat.
   c. Pilih Konfirmasi untuk menyimpan perubahan Anda.

Menggunakan AWS CLI

Untuk mengidentifikasi instans di klaster Anda yang menggunakan sertifikat server yang lebih lama, gunakan perintah `describe-db-clusters` dengan berikut ini.

```
aws docdb describe-db-instances \
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Bagaimana cara memodifikasi instans individu di klaster Amazon DocumentDB saya untuk memperbarui sertifikat server?

Kami merekomendasikan Anda memperbarui sertifikat server untuk semua instans dalam klaster tertentu pada waktu yang sama. Untuk memodifikasi instans di klaster Anda, Anda dapat menggunakan konsol atau AWS CLI.

> **Note**
>
> Memperbarui instans Anda memerlukan boot ulang, yang dapat menyebabkan gangguan layanan. Sebelum memperbarui sertifikat server, pastikan Anda telah menyelesaikan Langkah 1.

Menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat klaster Anda Wilayah AWS berada.
3. Dalam panel navigasi di sisi kiri konsol, pilih Instans.
4. Kolom Otoritas sertifikat (tersembunyi secara default) menunjukkan instans mana yang masih di sertifikat server lama (`rds-ca-2017`). Untuk menampilkan kolom Otoritas sertifikat, lakukan hal berikut:

    a. Pilih ikon Pengaturan.

    b. Di bawah daftar kolom yang terlihat, pilih kolom Otoritas sertifikat.

    c. Pilih Konfirmasi untuk menyimpan perubahan Anda.

5. Pilih sebuah instans untuk dimodifikasi.
6. Pilih Tindakan, dan kemudian pilih Modifikasi.
7. Di bawah Otoritas sertifikat, pilih sertifikat server baru (`rds-ca-rsa4096-g1`) untuk contoh ini.
8. Anda dapat melihat ringkasan perubahan pada halaman berikutnya. Perhatikan bahwa ada peringatan tambahan untuk mengingatkan Anda untuk memastikan bahwa aplikasi Anda menggunakan paket sertifikat CA terbaru sebelum memodifikasi instans untuk menghindari menyebabkan gangguan dalam konektivitas.
9. Anda dapat memilih untuk menerapkan modifikasi selama jendela pemeliharaan berikutnya atau menerapkan dengan segera.
10. Pilih Modifikasi instans untuk menyelesaikan pembaruan.

Menggunakan AWS CLI

Selesaikan langkah-langkah berikut untuk mengidentifikasi dan memutar sertifikat server lama untuk instans Amazon DocumentDB Anda yang ada menggunakan AWS CLI.

1. Untuk memodifikasi instans dengan segera, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Untuk memodifikasi instans dalam klaster Anda agar menggunakan sertifikat CA baru selama jendela pemeliharaan klaster Anda berikutnya, jalankan perintah berikut untuk setiap instans dalam klaster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Apa yang terjadi jika saya menambahkan instans baru ke klaster yang ada?

Semua instans baru yang dibuat menggunakan sertifikat server lama dan memerlukan koneksi TLS menggunakan sertifikat CA lama. Instans Amazon DocumentDB baru yang dibuat setelah 21 Maret 2022 akan default menggunakan sertifikat baru.

Apa yang terjadi jika ada penggantian instans atau failover di klaster saya?

Jika ada pengganti instans di klaster Anda, instans baru yang dibuat akan terus menggunakan sertifikat server yang sama dengan yang digunakan instans sebelumnya. Kami merekomendasikan Anda memperbarui sertifikat server untuk semua instans pada waktu yang sama. Jika terjadi failover di klaster, sertifikat server pada primer baru digunakan.

Jika saya tidak menggunakan TLS untuk terhubung ke klaster saya, apakah saya masih perlu memperbarui setiap instans saya?

Jika Anda tidak menggunakan TLS untuk terhubung ke klaster Amazon DocumentDB Anda, tidak ada tindakan yang diperlukan.

Jika saya tidak menggunakan TLS untuk terhubung ke klaster saya tetapi saya berencana untuk melakukannya di masa mendatang, apa yang harus saya lakukan?

Jika Anda membuat klaster sebelum 21 Maret 2022, ikuti Langkah 1 dan Langkah 2 di bagian sebelumnya untuk memastikan bahwa aplikasi Anda menggunakan bundel CA yang diperbarui, dan bahwa setiap instans Amazon DocumentDB menggunakan sertifikat server terbaru. Jika Anda membuat cluster setelah 21 Maret 2022, cluster Anda sudah memiliki sertifikat server terbaru. Untuk memverifikasi bahwa aplikasi Anda menggunakan paket CA terbaru, lihat Jika saya tidak menggunakan TLS untuk terhubung ke klaster saya, apakah saya masih perlu memperbarui setiap instans saya?

Bisakah tenggat waktu diperpanjang melampaui 18 Mei 2022?

Jika aplikasi Anda terhubung melalui TLS, batas waktu tidak dapat diperpanjang melampaui 18 Mei 2022.

Bagaimana saya bisa yakin bahwa saya menggunakan paket CA terbaru?

Untuk alasan kompatibilitas, file bundel CA lama dan baru diberi nama `us-gov-west-1-bundle.pem`. Anda juga dapat menggunakan alat seperti `openssl` atau `keytool` untuk memeriksa paket CA.

Mengapa saya melihat “RDS” dalam nama paket CA?

Untuk fitur pengelolaan tertentu, seperti manajemen sertifikat, Amazon DocumentDB menggunakan teknologi operasional yang dibagikan dengan Amazon Relational Database Service (Amazon RDS).

Jika saya menerapkan sertifikat server baru, dapatkah saya mengembalikannya ke sertifikat server lama?

Jika Anda perlu mengembalikan instans ke sertifikat server lama, kami merekomendasikan Anda melakukannya untuk semua instans di klaster. Anda dapat mengembalikan sertifikat server untuk setiap instance dalam klaster dengan menggunakan AWS Management Console atau. AWS CLI

Menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dalam daftar Wilayah di sudut kanan atas layar, pilih tempat klaster Anda Wilayah AWS berada.
3. Dalam panel navigasi di sisi kiri konsol, pilih Instans.
4. Pilih sebuah instans untuk dimodifikasi. Pilih Tindakan, dan kemudian pilih Modifikasi.
5. Di bawah Otoritas sertifikat, Anda dapat memilih sertifikat server lama (`rds-ca-2017`).
6. Pilih Lanjutkan dan lihat ringkasan dari modifikasi Anda.
7. Di halaman yang dihasilkan ini, Anda dapat memilih untuk menjadwalkan modifikasi Anda untuk diterapkan di jendela pemeliharaan berikutnya atau segera menerapkan modifikasi Anda. Tentukan pilihan Anda, dan pilih Modifikasi instans.

> **Note**
>
> Jika Anda memilih untuk menerapkan modifikasi Anda dengan segera, perubahan apa pun pada antrean modifikasi yang tertunda juga akan diterapkan. Jika salah satu dari modifikasi yang tertunda memerlukan waktu henti, memilih opsi ini dapat menyebabkan waktu henti yang tidak terduga.

Menggunakan AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2017 <--apply-immediately | --no-apply-immediately>
```

Jika Anda memilih `--no-apply-immediately`, perubahan akan diterapkan selama jendela pemeliharaan klaster berikutnya.

Jika saya memulihkan dari snapshot atau pemulihan titik waktu, apakah itu akan memiliki sertifikat server baru?

Jika Anda memulihkan snapshot atau melakukan point-in-time pemulihan setelah 21 Maret 2022, cluster baru yang dibuat akan menggunakan sertifikat CA baru.

Bagaimana jika saya mengalami masalah saat menyambungkan langsung ke klaster Amazon DocumentDB saya dari Mac OS X Catalina?

Mac OS X Catalina telah memperbarui persyaratan untuk sertifikat tepercaya. Sertifikat tepercaya sekarang harus valid selama 825 hari atau kurang (lihat https://support.apple.com/en-us/HT210176). Sertifikat instans Amazon DocumentDB valid selama lebih dari empat tahun, lebih lama dari maksimum Mac OS X. Untuk terhubung langsung ke klaster Amazon DocumentDB dari komputer yang menjalankan Mac OS X Catalina, Anda harus mengizinkan sertifikat yang tidak valid saat membuat koneksi TLS. Dalam hal ini, sertifikat yang tidak valid berarti masa validnya lebih lama dari 825 hari. Anda harus memahami risikonya sebelum mengizinkan sertifikat yang tidak valid saat menghubungkan ke klaster Amazon DocumentDB Anda.

Untuk menyambung ke cluster Amazon DocumentDB dari OS X Catalina menggunakan AWS CLI, gunakan parameter. `tlsAllowInvalidCertificates`

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Validasi Kepatuhan di Amazon DocumentDB

Keamanan dan kepatuhan Amazon DocumentDB (dengan kompatibilitas MongoDB) dinilai oleh auditor pihak ke tiga sebagai bagian dari beberapa program kepatuhan AWS, termasuk sebagai berikut:

- Kontrol Sistem dan Organisasi (SOC) 1, 2, dan 3. Untuk informasi selengkapnya, lihat SOC.
- Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS). Untuk informasi lain, lihat PCI DSS.
- ISO 9001, 27001, 27017, dan 27018. Untuk informasi selengkapnya, lihat Bersertifikat ISO.
- Undang-Undang Akuntabilitas dan Portabilitas Asuransi Kesehatan Perjanjian Rekan Bisnis (HIPAA BAA). Untuk informasi lebih lanjut, lihat Kepatuhan HIPAA

AWS menyediakan daftar layanan AWS yang sering diperbarui dalam cakupan program kepatuhan tertentu di Layanan AWS dalam Cakupan menurut Program Kepatuhan.

Laporan audit pihak ke tiga tersedia untuk diunduh menggunakan AWS Artifact. Untuk informasi lebih lanjut, lihat Mengunduh Laporan di AWS Artifact.

Untuk informasi lebih lanjut tentang program kepatuhan AWS lihat Program Kepatuhan AWS.

Tanggung jawab kepatuhan Anda saat menggunakan Amazon DocumentDB ditentukan oleh sensitivitas data Anda, tujuan kepatuhan organisasi Anda, serta undang-undang dan peraturan yang berlaku. Jika penggunaan Amazon DocumentDB Anda tunduk pada kepatuhan standar seperti HIPAA atau PCI, AWS menyediakan sumber daya untuk membantu:

- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS.
- AWS Config – Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub – Pandangan komprehensif tentang status keamanan Anda di dalam AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.
- Perancangan Laporan Resmi Keamanan dan Kepatuhan HIPAA – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk menciptakan aplikasi yang sesuai dengan HIPAA.

Ketahanan di Amazon DocumentDB

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Dengan Availability Zone, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Availability Zone tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, menoleransi kegagalan, dan dapat diskalakan dibandingkan satu atau beberapa infrastruktur pusat data tradisional.

Sebuah klaster Amazon DocumentDB hanya dapat dibuat dalam Amazon VPC yang memiliki setidaknya dua subnet di setidaknya dua Availability Zone. Dengan mendistribusikan instans klaster di setidaknya dua Availability Zone, Amazon DocumentDB membantu memastikan bahwa ada instans yang tersedia di klaster Anda apabila terjadi kegagalan Availability Zone yang tidak dimungkinkan. Volume klaster untuk klaster Amazon DocumentDB Anda selalu mencakup tiga Availability Zone untuk menyediakan penyimpanan tahan lama dengan kemungkinan kehilangan data yang lebih kecil.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur Global AWS.

Selain infrastruktur global AWS, Amazon DocumentDB memberi penawaran beberapa fitur untuk membantu support ketahanan data dan kebutuhan backup Anda.

Penyimpanan toleransi kesalahan dan penyembuhan diri

Setiap porsi 10 GB dari volume penyimpanan Anda direplikasi dengan enam cara, di tiga Availability Zone. Amazon DocumentDB menggunakan penyimpanan toleran kesalahan yang secara transparan menangani hilangnya hingga dua salinan data tanpa memengaruhi ketersediaan tulis basis data, dan hingga tiga salinan tanpa memengaruhi ketersediaan baca. Penyimpanan Amazon DocumentDB juga memulihkan diri; blok data dan disk terus dipindai untuk mencari kesalahan dan diganti secara otomatis.

Backup dan pemulihan manual

Amazon DocumentDB menyediakan kemampuan untuk membuat backup penuh klaster Anda untuk retensi dan pemulihan jangka panjang. Untuk informasi selengkapnya, lihat Membuat Cadangan dan Memulihkan di Amazon DocumentDB.

Pemulihan dalam waktu

Pemulihan titik waktu (PITR) membantu melindungi klaster Amazon DocumentDB Anda dari operasi penulisan atau penghapusan yang tidak disengaja. Dengan pemulihan titik waktu, Anda tidak perlu khawatir tentang membuat, memelihara, atau menjadwalkan pencadangan sesuai permintaan. Untuk informasi selengkapnya, lihat Memulihkan ke Titik Waktu.

Keamanan Infrastruktur di Amazon DocumentDB

Sebagai layanan terkelola, Amazon DocumentDB dilindungi olehAWSkeamanan jaringan global. Untuk informasi tentangAWSlayanan keamanan dan bagaimanaAWSmelindungi infrastruktur,

lihatAWSKeamanan Cloud. Untuk mendesainAWSlingkungan menggunakan praktik terbaik untuk keamanan infrastruktur, lihatPerlindungan InfrastrukturdiPilar KeamananAWSKerangka Kerja yang Diarsiteksikan dengan Baik.

Anda menggunakan panggilan API yang diterbitkan AWS untuk mengakses Amazon DocumentDB melalui jaringan. Klien harus mendukung hal berikut:

- Transport Layer Security (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju sempurna (PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan principal IAM. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Anda dapat memanggil operasi API ini dari lokasi jaringan mana pun. Anda dapat menggunakan kebijakan Amazon DocumentDB untuk mengontrol akses dari titik akhir Amazon Virtual Private Cloud (Amazon VPC) tertentu atau VPC tertentu. Secara efektif, ini mengisolasi akses jaringan ke sumber daya Amazon DocumentDB tertentu hanya dari VPC tertentu dalam jaringan AWS.

> **Note**
>
> Amazon DocumentDB tidak mendukung kebijakan akses berbasis sumber daya.

Praktik Terbaik Keamanan untuk Amazon DocumentDB

Untuk praktik terbaik keamanan, Anda harus menggunakan akun AWS Identity and Access Management(IAM) untuk mengontrol akses ke operasi Amazon DocumentDB API, khususnya operasi yang membuat, memodifikasi, atau menghapus sumber daya Amazon DocumentDB. Sumber daya tersebut termasuk klaster, grup keamanan, dan grup parameter. Anda juga harus menggunakan IAM untuk mengontrol tindakan yang melakukan tindakan administrasi umum seperti mencadangkan pemulihan klaster. Saat membuat IAM role, terapkan prinsip hak istimewa terendah.

- Terapkan hak istimewa terendah dengan kontrol akses berbasis peran.

- Tetapkan akun IAM individu untuk setiap orang yang mengelola sumber daya Amazon DocumentDB. Jangan gunakan pengguna root Akun AWS untuk mengelola sumber daya Amazon DocumentDB. Buat pengguna IAM untuk semua orang, termasuk Anda sendiri.
- Berikan setiap pengguna set izin minimum yang diperlukan untuk melakukan tugas mereka.
- Gunakan grup IAM untuk mengelola izin secara efektif untuk beberapa pengguna. Untuk informasi lebih lanjut tentang IAM, lihat Panduan Pengguna IAM. Untuk informasi tentang praktik terbaik IAM, lihat Praktik Terbaik IAM.
- Secara rutin putar kredensial IAM Anda.
- Konfigurasi AWS Secrets Manager untuk secara otomatis memutar rahasia untuk Amazon DocumentDB. Untuk informasi selengkapnya, lihat Memutar AWS Secrets Manager Anda dan Memutar Rahasia untuk Amazon DocumentDB diPanduan Pengguna AWS Secrets Manager.
- Gunakan Keamanan Lapisan Pengangkutan (TLS) dan enkripsi yang tidak bergerak untuk mengenkripsi data Anda.

Mengaudit Peristiwa Amazon DocumentDB

Dengan Amazon DocumentDB (dengan kompatibilitas MongoDB), Anda dapat mengaudit peristiwa yang dilakukan di klaster Anda. Contoh log acara termasuk upaya autentikasi yang berhasil dan gagal, membuang koleksi dalam basis data, atau membuat indeks. Secara default, pengauditan dinonaktifkan di Amazon DocumentDB dan mengharuskan Anda untuk menggunakan fitur ini.

Saat audit diaktifkan, Amazon DocumentDB mencatat Data Definition Language (DDL), Data Manipulation Language (DHTML), otentikasi, otorisasi, dan peristiwa manajemen pengguna ke Amazon Logs. CloudWatch Saat audit diaktifkan, Amazon DocumentDB mengekspor catatan audit klaster Anda (dokumen JSON) ke Amazon Logs. CloudWatch Anda dapat menggunakan Amazon CloudWatch Logs untuk menganalisis, memantau, dan mengarsipkan peristiwa audit Amazon DocumentDB Anda.

Meskipun Amazon DocumentDB tidak membebankan biaya tambahan untuk mengaktifkan audit, Anda dikenakan tarif standar untuk penggunaan Log. CloudWatch Untuk informasi tentang harga CloudWatch Log, lihat CloudWatch harga Amazon.

Fitur audit Amazon DocumentDB jelas berbeda dari penggunaan sumber daya layanan yang dipantau. AWS CloudTrail CloudTrail merekam operasi yang dilakukan dengan AWS Command Line Interface (AWS CLI) atau AWS Management Console pada sumber daya seperti cluster, instance, grup parameter, dan snapshot. Audit sumber AWS daya aktif CloudTrail secara default dan tidak

dapat dinonaktifkan. Fitur audit Amazon DocumentDB adalah fitur keikutsertaan. Ini mencatat operasi yang terjadi dalam klaster Anda pada objek, seperti basis data, koleksi, indeks, dan pengguna.

Topik

- Peristiwa yang didukung
- Mengaktifkan Audit
- Mengaktifkan Audit
- Mengakses Acara Audit Anda

Peristiwa yang didukung

Audit Amazon DocumentDB mendukung kategori acara berikut:

- Data Definition Language (DDL) - termasuk operasi manajemen database, koneksi, manajemen pengguna, dan otorisasi.
- Data Manipulation Language read events (pembacaan DHTML) - termasuk `find()` dan berbagai operator agregasi, operator aritmatika, operator boolean, dan operator kueri baca lainnya.
- Manipulasi Data Bahasa menulis peristiwa (DHTML menulis) - termasuk `insert(), update(), delete()`, dan operator `bulkWrite()`

Jenis peristiwanya adalah sebagai berikut.

Jenis Acara	Kategori	Deskripsi
`authCheck`	Otorisasi	Kode hasil 0: Sukses
		Kode hasil 13: Upaya tidak sah untuk melakukan operasi.
`authenticate`	Koneksi	Upaya autentikasi yang berhasil atau gagal pada koneksi baru.

Jenis Acara	Kategori	Deskripsi
`createDatabase`	DDL	Pembuatan basis data baru.
`createCollection`	DDL	Pembuatan koleksi baru dalam basis data.
`createIndex`	DDL	Pembuatan indeks baru dalam koleksi.
`dropCollection`	DDL	Menghapus koleksi dalam basis data.
`dropDatabase`	DDL	Menghapus basis data.
`dropIndex`	DDL	Menghapus indeks dalam koleksi.
`modifyChangeStreams`	DDL	Ubah aliran telah dibuat.
`renameCollection`	DDL	Mengganti nama koleksi dalam database.
`createRole`	Manajemen Peran	Menciptakan peran.
`dropAllRolesFromDatabase`	Manajemen Peran	Menjatuhkan semua peran dalam database.
`dropRole`	Manajemen Peran	Menjatuhkan peran.
`grantPrivilegesToRole`	Manajemen Peran	Memberikan hak istimewa untuk suatu peran.

Jenis Acara	Kategori	Deskripsi
`grantRolesToRole`	Manajemen Peran	Memberikan peran ke peran yang ditentukan pengguna.
`revokePrivilegesFromRole`	Manajemen Peran	Mencabut hak istimewa dari suatu peran.
`revokeRolesFromRole`	Manajemen Peran	Mencabut peran dari peran yang ditentukan pengguna.
`updateRole`	Manajemen Peran	Memperbarui peran.
`createUser`	Pengelolaan Pengguna	Pembuatan pengguna baru.
`dropAllUsersFromDatabase`	Manajemen pengguna	Menghapus semua pengguna dalam basis data.
`dropUser`	Manajemen pengguna	Menghapus pengguna yang ada.
`grantRolesToUser`	Manajemen pengguna	Memberikan peran kepada pengguna.
`revokeRolesFromUser`	Manajemen pengguna	Mencabut peran dari pengguna.
`updateUser`	UserManagement	Memperbarui pengguna yang sudah ada.
`insert`	DMLTulis	Menyisipkan dokumen atau dokumen ke dalam koleksi.

Jenis Acara	Kategori	Deskripsi
`delete`	DMLTulis	Menghapus dokumen atau dokumen dari koleksi.
`update`	DMLTulis	Memodifikasi dokumen atau dokumen yang ada dalam koleksi.
`bulkWrite`	DMLTulis	Melakukan beberapa operasi tulis dengan kontrol untuk urutan eksekusi.
`count`	DMLdibaca	Mengembalikan jumlah dokumen yang akan cocok dengan query find () untuk koleksi atau tampilan.
`countDocuments`	DMLdibaca	Mengembalikan jumlah dokumen yang cocok dengan query untuk koleksi atau tampilan.
`find`	DMLdibaca	Memilih dokumen dalam koleksi atau tampilan dan mengembalikan kursor ke dokumen yang dipilih.
`findAndModify`	DMLmembaca dan menulis DML	Memodifikasi dan mengembalikan satu dokumen.

Jenis Acara	Kategori	Deskripsi
`findOneAndDelete`	DMLmembaca dan menulis DML	Menghapus satu dokumen berdasarkan filter dan kriteria pengurutan, mengembalikan dokumen yang dihapus.
`findOneAndReplace`	DMLmembaca dan menulis DML	Menggantikan satu dokumen berdasarkan filter yang ditentukan.
`findOneAndUpdate`	DMLmembaca dan menulis DML	Memperbarui satu dokumen berdasarkan filter dan kriteria pengurutan.
`aggregate`	DMLmembaca dan menulis DML	Mendukung API dalam pipeline agregasi.
`distinct`	DMLdibaca	Menemukan nilai yang berbeda untuk bidang tertentu di satu koleksi atau tampilan dan mengembalikan hasilnya dalam array.

> **Note**
>
> Nilai dalam bidang parameter dokumen peristiwa DMLmemiliki batas ukuran 1KB. Amazon DocumentDB memotong nilainya jika melebihi 1KB.

> **Note**
>
> Acara penghapusan TTL tidak diaudit saat ini.

Mengaktifkan Audit

Mengaktifkan audit pada klaster adalah proses dua langkah. Pastikan bahwa kedua langkah selesai, atau log audit tidak akan dikirim ke CloudWatch Log.

Langkah 1. Aktifkan parameter klaster audit_logs

Untuk mengaktifkan audit, Anda perlu memodifikasi `audit_logs` parameter dalam grup parameter. `audit_logs`adalah daftar peristiwa yang dibatasi koma untuk dicatat. Peristiwa harus ditentukan dalam huruf kecil dan seharusnya tidak ada spasi putih di antara elemen daftar.

Anda dapat mengatur nilai berikut untuk grup parameter:

Nilai	Deskripsi
`ddl`	Pengaturan ini akan memungkinkan audit untuk acara DDL seperti createDatabase, DropDatabase, createCollection, DropCollection, createIndex, DropIndex, AuthCheck, otentikasi, createUser, Dropuser, User, User, UpdateUser, dan grantRolesTo revokeRolesFrom dropAllUsersFromDatabase

Nilai	Deskripsi
`dml_read`	Menyetel ini akan memungkinkan auditng untuk acara baca DMLseperti find, sort count, different , group, projecta, unwind, geoInters ects, GeoInters ects, GeoWithin dan operator kueri baca MongoDB lainnya.
`dml_write`	Menyetel ini akan memungkinkan audit untuk acara penulisan DMLseperti insert (), update (), delete (), dan bulkWrite ()
`all`	Pengaturan ini akan memungkinkan audit untuk peristiwa database Anda, seperti kueri baca, kueri tulis, tindakan database, dan tindakan administr ator.
`none`	Pengaturan ini akan menonaktifkan audit

Nilai	Deskripsi	
`enabled`(warisan)	Ini adalah pengaturan parameter lama yang setara dengan 'ddl'. Pengaturan ini akan memungkinkan audit untuk acara DDL seperti createDatabase, DropDatabase, createCollection, DropCollection, createIndex, DropIndex, AuthCheck, authenticate, createUser, Dropuser, User, User, UpdateUser, dan. grantRolesTo revokeRolesFrom dropAllUsers FromDatabase Kami tidak menyarankan menggunakan pengaturan ini karena ini adalah pengaturan lama.	

Nilai	Deskripsi
`disabled` (warisan)	Ini adalah pengaturan parameter lama yang setara dengan 'tidak ada'. Kami tidak menyarankan menggunakan pengaturan ini karena ini adalah pengaturan lama.

> **Note**
>
> Nilai default untuk parameter cluster audit_logs adalah `none` (legacy ""`disabled`).

Anda juga dapat menggunakan nilai yang disebutkan di atas dalam kombinasi.

Nilai	Deskripsi
`ddl, dml_read`	Menyetel ini akan memungkinkan auding untuk acara DDL dan acara baca DML.
`ddl, dml_write`	Menyetel ini akan memungkinkan auding untuk acara DDL dan penulisan DML.
`dml_read, dml_write`	Menyetel ini akan memungkinkan audit

Nilai	Deskripsi	
	untuk semua peristiwa DML.	

> **Note**
>
> Anda tidak dapat memodifikasi grup parameter default.

Untuk informasi selengkapnya, lihat yang berikut:

- Membuat grup parameter cluster Amazon DocumentDB

  Setelah membuat grup parameter khusus, ubah dengan mengganti nilai parameter `audit_logs` menjadi `enabled`.
- Memodifikasi grup parameter cluster Amazon DocumentDB

Langkah 2. Aktifkan Ekspor CloudWatch Log Amazon

Ketika nilai parameter `audit_logs` cluster adalah`enabled`,,, atau `ddl` `dml_read``dml_write`, Anda juga harus mengaktifkan Amazon DocumentDB untuk mengekspor log ke Amazon. CloudWatch Jika Anda menghilangkan salah satu dari langkah-langkah ini, log audit tidak akan dikirim ke CloudWatch.

Saat membuat klaster, melakukan point-in-time-restore, atau memulihkan snapshot, Anda dapat mengaktifkan CloudWatch Log dengan mengikuti langkah-langkah berikut.

Using the AWS Management Console

Untuk mengaktifkan Amazon DocumentDB mengekspor log CloudWatch untuk menggunakan konsol, lihat topik berikut:

- Saat membuat klaster — Di Membuat cluster dan instance primer menggunakan AWS Management Console, lihat Membuat Klaster: Konfigurasi Tambahan (langkah 5, Ekspor log)
- Saat memodifikasi cluster yang ada — Memodifikasi cluster Amazon DocumentDB
- Saat melakukan pemulihan snapshot cluster — Memulihkan dari Snapshot Klaster

- Saat melakukan point-in-time pemulihan — Memulihkan ke Titik Waktu

Using the AWS CLI

Untuk mengaktifkan log audit saat membuat klaster baru

Kode berikut membuat cluster `sample-cluster` dan memungkinkan log CloudWatch audit.

Example

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Untuk mengaktifkan log audit ketika memodifikasi klaster yang ada

Kode berikut memodifikasi cluster `sample-cluster` dan memungkinkan log CloudWatch audit.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster \
```

```
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Untuk Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Ouput dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "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"
    }
}
```

Mengaktifkan Audit

Anda dapat menonaktifkan audit dengan menonaktifkan ekspor CloudWatch Log dan menonaktifkan parameter. `audit_logs`

Menonaktifkan Ekspor Log CloudWatch

Anda dapat menonaktifkan mengekspor log audit dengan menggunakan AWS Management Console atau AWS CLI.

Using the AWS Management Console

Prosedur berikut menggunakan AWS Management Console untuk menonaktifkan Amazon DocumentDB mengekspor log ke. CloudWatch

Untuk menonaktifkan log audit

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster. Lalu, pilih tombol di sebelah kiri nama klaster yang log ekspornya ingin Anda nonaktifkan.
3. Pilih Tindakan, lalu pilih Modifikasi.
4. Gulir ke bawah ke bagian Ekspor log dan pilih Nonaktif.
5. Pilih Continue (Lanjutkan).
6. Tinjau perubahan Anda, lalu pilih kapan Anda ingin perubahan ini diterapkan ke klaster Anda.
   - Terapkan selama jendela pemeliharaan terjadwal berikutnya
   - Terapkan segera

7. Pilih Ubah klaster.

Using the AWS CLI

Kode berikut memodifikasi cluster `sample-cluster` dan menonaktifkan log CloudWatch audit.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

Untuk Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "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"
            }
        ]
    }
}
```

Menonaktifkan Parameter audit_logs

Untuk menonaktifkan parameter `audit_logs` pada klaster, Anda dapat memodifikasi klaster agar menggunakan grup parameter ketika nilai parameter `audit_logs` adalah `disabled`. Atau Anda dapat mengubah nilai parameter `audit_logs` di grup parameter klaster sehingga menjadi `disabled`.

Untuk informasi lain, lihat topik berikut:

- Memodifikasi cluster Amazon DocumentDB
- Memodifikasi grup parameter cluster Amazon DocumentDB

Mengakses Acara Audit Anda

Gunakan langkah-langkah berikut untuk mengakses peristiwa audit Anda di Amazon CloudWatch.

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pastikan Anda berada di Wilayah yang sama dengan klaster Amazon DocumentDB Anda.

3. Di panel navigasi, pilih Log.
4. Untuk menemukan log audit pada klaster Anda, dari daftar yang tersedia, cari dan pilih **/aws/docdb/*yourClusterName*/audit**.

   Peristiwa audit untuk setiap instans Anda tersedia di bawah setiap nama instans masing-masing.

Membuat Cadangan dan Memulihkan di Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) terus mencadangkan data Anda ke Amazon Simple Storage Service (Amazon S3) selama 1–35 hari sehingga Anda dapat dengan cepat memulihkan ke titik mana pun dalam periode retensi pencadangan. Amazon DocumentDB juga mengambil snapshot otomatis data Anda sebagai bagian dari proses backup terus menerus ini.

> **Note**
>
> Ini adalah bucket Amazon S3 yang dikelola layanan dan Anda tidak akan memiliki akses ke file backup. Jika Anda ingin mengontrol backup Anda sendiri, ikuti petunjuk pada Membuang, Memulihkan, Mengimpor, dan Mengekspor Data.

Anda juga dapat mempertahankan data backup di luar periode retensi pencadangan dengan membuat snapshot manual data klaster Anda. Proses backup tidak memengaruhi performa klaster Anda.

Bagian ini membahas kasus penggunaan untuk kemampuan backup di Amazon DocumentDB dan menunjukkan cara untuk mengelola backup untuk klaster Amazon DocumentDB Anda.

Topik

- Mencadangkan dan Memulihkan: Konsep
- Memahami Penggunaan Penyimpanan Backup
- Membuang, Memulihkan, Mengimpor, dan Mengekspor Data
- Pertimbangan Snapshot Klaster
- Membandingkan Snapshot Otomatis dan Manual
- Membuat Manual Klaster Snapshot
- Menyalin Snapshot Klaster Amazon DocumentDB
- Membagikan Snapshot Klaster Amazon DocumentDB
- Memulihkan dari Snapshot Klaster
- Memulihkan ke Titik Waktu
- Menghapus Snapshot Klaster

Mencadangkan dan Memulihkan: Konsep

Kata benda	Deskripsi	API (Kata kerja)

Periode retensi cadangan	Jangka waktu antara 1 dan 35 hari di mana Anda dapat melakukan point-in-time pemulihan.	`create-db-cluster`
		`modify-db-cluster`
		`restore-db-cluster-to-point-in-time`

Volume penyimpanan Amazon DocumentDB	Volume penyimpanan yang sangat tersedia dan sangat tahan lama yang mereplikasi data enam cara di tiga Availability Zone. Sebuah klaster Amazon DocumentDB sangat tahan lama terlepas dari jumlah	`create-db-cluster` `delete-db-cluster`

Kata benda	Deskripsi	API (Kata kerja)
	instans dalam klaster.	

Periode pencadangan	Periode waktu dalam hari di mana snapshot otomatis diambil.	`create-db-cluster`
		`describe-db-cluster`
		`modify-db-cluster`

Snapshot otomatis	Snapshot harian yang merupakan backup penuh klaster dan dibuat secara otomatis oleh proses backup berkelanjutan di Amazon DocumentDB.	`restore-db-cluster-from-snapshot`
		`describe-db-cluster-snapshot-attributes`
		`describe-db-cluster-snapshots`

Kata benda	Deskripsi	API (Kata kerja)

Snapshot manual	Snapshot yang Anda buat secara manual untuk mempertahankan backup penuh klaster di luar periode backup.	`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`

Memahami Penggunaan Penyimpanan Backup

Penyimpanan backup Amazon DocumentDB terdiri dari backup berkelanjutan dalam periode retensi pencadangan dan snapshot manual di luar periode retensi. Untuk mengontrol penggunaan penyimpanan backup, Anda dapat mengurangi interval retensi backup, menghapus snapshot manual lama saat tidak lagi diperlukan, atau keduanya. Untuk informasi umum tentang backup Amazon DocumentDB, lihat Membuat Cadangan dan Memulihkan di Amazon DocumentDB. Untuk informasi harga tentang penyimpanan backup Amazon DocumentDB, lihat Harga Amazon DocumentDB.

Untuk mengontrol biaya Anda, Anda dapat memantau jumlah penyimpanan yang digunakan oleh pencadangan berkelanjutan dan snapshot manual yang terus berlanjut melampaui periode penyimpanan. Kemudian Anda dapat mengurangi interval retensi cadangan dan menghapus snapshot manual ketika tidak diperlukan lagi.

Anda dapat menggunakan CloudWatch metrik `AmazonTotalBackupStorageBilled`,`SnapshotStorageUsed`, dan `BackupRetentionPeriodStorageUsed` untuk meninjau dan memantau jumlah penyimpanan yang digunakan oleh backup Amazon DocumentDB Anda, sebagai berikut:

- `BackupRetentionPeriodStorageUsed` menunjukkan jumlah penyimpanan backup yang digunakan, untuk menyimpan backup berkelanjutan pada saat ini. Nilai metrik ini bergantung pada ukuran volume klaster dan jumlah perubahan yang Anda buat selama periode retensi. Namun, untuk tujuan penagihan, metrik tidak melebihi ukuran volume klaster kumulatif selama periode retensi. Misalnya, jika ukuran klaster Anda adalah 100 GiB dan periode retensi Anda adalah dua hari, nilai maksimum untuk `BackRetentionPeriodStorageUsed` adalah 200 GiB (100 GiB + 100 GiB).

- `SnapshotStorageUsed` mewakili jumlah penyimpanan backup yang digunakan, untuk menyimpan snapshot manual di luar periode retensi pencadangan. Snapshot manual yang diambil dalam periode retensi tidak dihitung terhadap penyimpanan backup Anda. Demikian pula, snapshot otomatis tidak dihitung terhadap penyimpanan backup Anda. Ukuran masing-masing snapshot adalah ukuran volume klaster pada saat Anda mengambil snapshot. Nilai `SnapshotStorageUsed` bergantung pada jumlah snapshot yang disimpan dan ukuran setiap snapshot. Misalnya, Anda memiliki satu snapshot di luar periode retensi dan ukuran volume klaster adalah 100 GiB saat snapshot itu diambil. Jumlah `SnapshotStorageUsed` adalah 100 GiB.

- `TotalBackupStorageBilled` mewakili jumlah dari `BackupRetentionPeriodStorageUsed` dan `SnapshotStorageUsed`, dikurangi jumlah penyimpanan backup gratis yang setara dengan ukuran volume klaster selama satu hari. Misalnya, jika ukuran klaster Anda adalah 100 GiB, Anda memiliki satu hari retensi, dan Anda memiliki satu snapshot di luar periode retensi, `TotalBackupStorageBilled`-nya adalah 100 GiB (100 GiB + 100 GiB - 100 GiB).

- Metrik ini dihitung secara independen untuk setiap klaster Amazon DocumentDB.

Anda dapat memantau klaster Amazon DocumentDB dan membuat laporan CloudWatch menggunakan metrik melalui konsol. CloudWatch Untuk informasi selengkapnya tentang cara menggunakan CloudWatch metrik, lihatMemantau Amazon DocumentDB.

Membuang, Memulihkan, Mengimpor, dan Mengekspor Data

Anda dapat menggunakan utilitas `mongodump`, `mongorestore`, `mongoexport`, dan `mongoimport` untuk memindahkan data masuk dan keluar dari klaster Amazon DocumentDB Anda. Bagian ini

membahas tujuan masing-masing alat dan konfigurasi ini untuk membantu Anda mencapai performa yang lebih baik.

Topik

- mongodump
- mongorestore
- mongoexport
- mongoimport
- Tutorial

mongodump

Utilitas `mongodump` membuat backup biner (BSON) dari basis data MongoDB. Alat `mongodump` adalah metode yang disukai untuk membuang data dari deployment MongoDB sumber Anda ketika ingin memulihkannya ke dalam klaster Amazon DocumentDB Anda karena efisiensi ukuran yang dicapai dengan menyimpan data dalam format biner.

Tergantung pada sumber daya yang tersedia pada instans atau mesin yang Anda gunakan untuk melakukan perintah, Anda dapat mempercepat `mongodump` dengan meningkatkan jumlah koneksi paralel yang dibuang dari default 1 menggunakan opsi `--numParallelCollections`. Aturan praktis yang baik adalah memulai dengan satu pekerja per vCPU pada instans utama klaster Amazon DocumentDB Anda.

> **Note**
>
> Kami merekomendasikan MongoDB Database Tools hingga dan termasuk versi 100.6.1 untuk Amazon DocumentDB. Anda dapat mengakses unduhan Alat Database MongoDB di sini.

Contoh Penggunaan

Berikut ini adalah contoh penggunaan utilitas `mongodump` dalam klaster Amazon DocumentDB, `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

Utilitas `mongorestore` memungkinkan Anda untuk memulihkan backup biner (BSON) dari basis data yang telah dibuat dengan utilitas `mongodump`. Anda dapat meningkatkan performa pemulihan dengan meningkatkan jumlah pekerja untuk setiap koleksi selama pemulihan dengan opsi `--numInsertionWorkersPerCollection` (default-nya adalah 1). Aturan praktis yang baik adalah memulai dengan satu pekerja per vCPU pada instans utama klaster Amazon DocumentDB Anda.

Contoh Penggunaan

Berikut ini adalah contoh penggunaan utilitas `mongorestore` dalam klaster Amazon DocumentDB, `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

Alat `mongoexport` mengekspor data di Amazon DocumentDB ke format file JSON, CSV, atau TSV. Alat `mongoexport` adalah metode yang disukai untuk mengekspor data yang harus dapat dibaca oleh manusia atau mesin.

> **Note**
>
> `mongoexport` tidak secara langsung mendukung ekspor paralel. Namun, dimungkinkan untuk meningkatkan performa dengan mengeksekusi beberapa tugas `mongoexport` secara bersamaan untuk koleksi yang berbeda.

Contoh Penggunaan

Berikut ini adalah contoh penggunaan alat `mongoexport` dalam klaster Amazon DocumentDB, `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`

Alat `mongoimport` mengimpor isi file JSON, CSV, atau TSV ke dalam klaster Amazon DocumentDB. Anda dapat menggunakan parameter `--numInsertionWorkers` untuk memparalelkan dan mempercepat impor (default-nya adalah 1).

Contoh Penggunaan

Berikut ini adalah contoh penggunaan alat `mongoimport` dalam klaster Amazon DocumentDB, `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

Tutorial berikut ini menjelaskan cara menggunakan utilitas mongodump, `mongorestore`, `mongoexport`, dan `mongoimport` untuk memindahkan data masuk dan keluar dari klaster Amazon DocumentDB.

1. Prasyarat — Sebelum Anda mulai, pastikan bahwa klaster Amazon DocumentDB Anda telah disediakan dan bahwa Anda memiliki akses ke Instans Amazon EC2 di VPC yang sama dengan klaster Anda. Untuk informasi selengkapnya, lihat Connect menggunakan Amazon EC2.

   Untuk dapat menggunakan alat utilitas mongo, Anda harus menginstal mongodb-org-tools paket di instans EC2 Anda, sebagai berikut.

   ```
   sudo yum install mongodb-org-tools-4.0.18
   ```

   Karena Amazon DocumentDB menggunakan enkripsi Keamanan Lapisan Pengangkutan (TLS) secara default, Anda juga harus mengunduh file otoritas sertifikat (CA) Amazon RDS untuk menggunakan shell mongo guna menghubungkan, sebagai berikut.

   ```
   wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
   ```

2. Unduh data sampel — Untuk tutorial ini, Anda akan mengunduh beberapa data sampel yang berisi informasi tentang restoran.

   ```
   wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/
   datasets/restaurant.json
   ```

3. Impor data sampel ke Amazon DocumentDB — Karena data dalam format JSON logis, Anda akan menggunakan utilitas `mongoimport` untuk mengimpor data ke dalam klaster Amazon DocumentDB Anda.

   ```
   mongoimport --ssl \
       --host="tutorialCluster.amazonaws.com:27017" \
       --collection=restaurants \
       --db=business \
       --file=restaurant.json \
       --numInsertionWorkers 4 \
       --username=<yourUsername> \
       --password=<yourPassword> \
       --sslCAFile global-bundle.pem
   ```

4. Dump data dengan **`mongodump`** — Sekarang setelah Anda memiliki data di klaster Amazon DocumentDB, Anda dapat melakukan dump biner dari data tersebut menggunakan utilitas `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. Jatuhkan koleksi **`restaurants`** — Sebelum Anda memulihkan koleksi `restaurants` di basis data `business`, Anda harus terlebih dahulu menjatuhkan koleksi yang sudah ada dalam basis data tersebut, sebagai berikut.

```
use business
```

```
db.restaurants.drop()
```

6. Pulihkan data dengan **`mongorestore`** — Dengan dump biner data dari Langkah 3, Anda sekarang dapat menggunakan utilitas `mongorestore` untuk memulihkan data Anda ke klaster Amazon DocumentDB Anda.

```
mongorestore --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem restaurantDump.bson
```

7. Ekspor data menggunakan **`mongoexport`** — Untuk menyelesaikan tutorial, ekspor data dari klaster Anda dalam format file JSON, tidak berbeda dari file yang Anda impor di Langkah 1.

```
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. Validasi — Anda dapat memvalidasi bahwa output dari Langkah 5 menghasilkan hasil yang sama seperti Langkah 1 dengan perintah berikut.

```
wc -l restaurant.json
```

Output dari perintah ini:

```
2548 restaurant.json
```

```
wc -l restaurant2.json
```

Output dari perintah ini:

```
2548 restaurant2.json
```

Pertimbangan Snapshot Klaster

Amazon DocumentDB membuat snapshot otomatis harian dari klaster Anda selama jendela backup klaster Anda. Amazon DocumentDB menyimpan snapshot otomatis dari klaster Anda sesuai periode retensi pencadangan yang Anda tentukan. Jika perlu, Anda dapat memulihkan klaster Anda ke setiap titik waktu selama periode retensi pencadangan. Snapshot otomatis tidak terjadi saat operasi penyalinan dijalankan di Wilayah yang sama untuk klaster yang sama.

Topik

- Penyimpanan Backup
- Jendela Backup
- Periode Retensi Cadangan
- Salin Enkripsi Snapshot Cluster

Selain untuk snapshot klaster otomatis, Anda juga dapat membuat snapshot klaster secara manual. Anda dapat menyalin snapshot otomatis dan manual. Untuk informasi selengkapnya, lihat Membuat Manual Klaster Snapshot dan Menyalin Snapshot Klaster Amazon DocumentDB.

> **Note**
>
> Klaster Anda harus dalam status tersedia agar snapshot otomatis diambil.
> Anda tidak dapat membagikan snapshot klaster otomatis Amazon DocumentDB. Sebagai solusinya, Anda dapat membuat snapshot manual dengan menyalin snapshot otomatis, lalu membagikan salinan tersebut. Untuk informasi selengkapnya tentang menyalin snapshot, lihat Menyalin Snapshot Klaster Amazon DocumentDB. Untuk informasi selengkapnya tentang memulihkan klaster dari snapshot, lihat Memulihkan dari Snapshot Klaster.

Penyimpanan Backup

Penyimpanan cadangan Amazon DocumentDB Anda untuk Wilayah AWS masing-masing terdiri dari penyimpanan cadangan yang diperlukan untuk periode retensi cadangan Anda, yang mencakup snapshot cluster otomatis dan manual di Wilayah tersebut. Periode retensi cadangan default adalah 1 hari. Untuk informasi selengkapnya tentang harga penyimpanan backup, lihat Harga Amazon DocumentDB.

Ketika Anda menghapus sebuah klaster, semua snapshot otomatisnya dihapus dan tidak dapat dipulihkan. Namun, snapshot manual tidak dihapus ketika Anda menghapus klaster. Jika Anda memilih agar Amazon DocumentDB membuat snapshot akhir (snapshot manual) sebelum klaster Anda dihapus, Anda dapat menggunakan snapshot akhir untuk memulihkan klaster Anda.

Untuk informasi selengkapnya tentang snapshot dan penyimpanan, lihat Memahami Penggunaan Penyimpanan Backup.

Jendela Backup

Snapshot otomatis terjadi setiap hari selama jendela backup yang dipilih. Jika snapshot memerlukan waktu lebih dari yang dialokasikan ke jendela backup, proses backup berlanjut hingga selesai, meskipun jendela backup telah berakhir. Jendela backup tidak dapat menindih jendela pemeliharaan mingguan untuk klaster.

Jika Anda tidak menentukan jendela backup yang diinginkan saat Anda membuat klaster, Amazon DocumentDB menetapkan jendela backup 30 menit default. Jendela ini dipilih secara acak dari blok waktu 8 jam yang terkait dengan Wilayah klaster Anda. Anda dapat mengubah jendela backup pilihan Anda dengan memodifikasi klaster. Untuk informasi selengkapnya, lihat Memodifikasi cluster Amazon DocumentDB.

Nama Wilayah	Wilayah	Blok Waktu UTC
AS Timur (Ohio)	us-east-2	03:00-11:00
US East (Northern Virginia)	us-east-1	03:00-11:00
AS Barat (Oregon)	us-west-2	06:00-14:00
Asia Pasifik (Hong Kong)	ap-east-1	06:00-14:00
Asia Pasifik (Hyderabad)	ap-south-2	06:30 — 14:30
Asia Pasifik (Mumbai)	ap-south-1	06:00-14:00
Asia Pasifik (Seoul)	ap-northeast-2	13:00-21:00
Asia Pasifik (Singapura)	ap-southeast-1	14:00-22:00
Asia Pasifik (Sydney)	ap-southeast-2	12:00-20:00
Asia Pasifik (Tokyo)	ap-northeast-1	13:00-21:00
Kanada (Pusat)	ca-central-1	03:00-11:00
Tiongkok (Beijing)	cn-north-1	06:00-14:00
Tiongkok (Ningxia)	cn-northwest-1	06:00-14:00
Eropa (Frankfurt)	eu-central-1	21:00-05:00
Eropa (Irlandia)	eu-west-1	22:00-06:00
Eropa (London)	eu-west-2	22:00-06:00
Eropa (Milan)	eu-south-1	02:00-10:00
Eropa (Paris)	eu-west-3	23:59-07:29
Timur Tengah (UEA)	me-central-1	05:00 — 13:00
Amerika Selatan (Sao Paulo)	sa-east-1	00:00-08:00

Nama Wilayah	Wilayah	Blok Waktu UTC
AWS GovCloud (AS-Timur)	us-gov-east-1	17:00-01:00
AWS GovCloud (AS-Barat)	us-gov-west-1	06:00-14:00

Periode Retensi Cadangan

Periode retensi cadangan adalah jumlah hari backup otomatis dipertahankan sebelum dihapus secara otomatis. Amazon DocumentDB mendukung periode retensi cadangan selama 1-35 hari.

Anda dapat mengatur periode retensi cadangan saat Anda membuat klaster. Jika Anda tidak secara eksplisit mengatur periode retensi cadangan, periode retensi cadangan default selama 1 hari ditetapkan ke klaster Anda. Setelah Anda membuat klaster, Anda dapat memodifikasi periode retensi cadangan dengan memodifikasi klaster menggunakan salah satu AWS Management Console atau. AWS CLI Untuk informasi selengkapnya, lihat Memodifikasi cluster Amazon DocumentDB.

Salin Enkripsi Snapshot Cluster

Enkripsi cluster dan snapshot didasarkan pada kunci enkripsi KMS. ID kunci KMS adalah Amazon Resource Name (ARN), pengidentifikasi kunci KMS, atau alias kunci KMS untuk kunci enkripsi KMS.

Pedoman dan batasan berikut berlaku:

- Enkripsi disimpulkan dari cluster saat membuat snapshot. Jika cluster dienkripsi, snapshot cluster itu dienkripsi dengan kunci KMS yang sama. Jika cluster tidak dienkripsi, snapshot tidak dienkripsi.
- Jika Anda menyalin snapshot kluster terenkripsi dari akun Amazon Web Services, Anda dapat menentukan nilai `KmsKeyId` untuk mengenkripsi salinan dengan kunci enkripsi KMS baru. Jika Anda tidak menentukan nilai untuk`KmsKeyId`, maka salinan snapshot cluster dienkripsi dengan kunci KMS yang sama dengan snapshot cluster sumber.
- Jika Anda menyalin snapshot kluster terenkripsi yang dibagikan dari akun Amazon Web Services lainnya, Anda harus menentukan nilainya. `KmsKeyId`
- Untuk menyalin snapshot kluster terenkripsi ke Wilayah Amazon Web Services lainnya, atur `KmsKeyId` ke ID kunci KMS yang ingin Anda gunakan untuk mengenkripsi salinan snapshot klaster di Wilayah tujuan. Kunci enkripsi KMS khusus untuk Wilayah Amazon Web Services tempat mereka dibuat, dan Anda tidak dapat menggunakan kunci enkripsi dari satu Wilayah Amazon Web Services di Wilayah Amazon Web Services lainnya.

- Jika Anda menyalin snapshot klaster yang tidak terenkripsi dan menentukan nilai untuk parameter `KmsKeyId`, kesalahan akan dikembalikan.

Membandingkan Snapshot Otomatis dan Manual

Berikut ini adalah fitur utama dari snapshot otomatis dan manual Amazon DocumentDB (dengan kompatibilitas MongoDB).

Snapshot otomatis Amazon DocumentDB memiliki fitur utama berikut:

- Penamaan snapshot otomatis — Nama snapshot otomatis mengikuti pola `rds:<cluster-name>-yyyy-mm-dd-hh-mm`, dengan `yyyy-mm-dd-hh-mm` mewakili tanggal dan waktu snapshot dibuat.
- Dibuat secara otomatis sesuai jadwal — Ketika Anda membuat atau memodifikasi klaster, Anda dapat mengatur Periode retensi cadangan ke nilai bilangan bulat dari 1 sampai 35 hari. Secara default, klaster baru memiliki periode retensi cadangan selama 1 hari. Periode retensi cadangan menentukan jumlah hari snapshot otomatis disimpan sebelum dihapus secara otomatis. Anda tidak dapat menonaktifkan backup otomatis di klaster Amazon DocumentDB.

  Selain mengatur periode retensi cadangan, Anda juga mengatur jendela backup, waktu hari dimana snapshot otomatis dibuat.
- Menghapus snapshot otomatis — Snapshot otomatis dihapus ketika Anda menghapus klaster snapshot otomatis. Anda tidak dapat menghapus snapshot otomatis secara manual.
- Tambahan — Selama periode retensi cadangan, pembaruan basis data dicatat sehingga ada catatan inkremental dari perubahan.
- Memulihkan dari snapshot otomatis — Anda dapat memulihkan dari snapshot otomatis menggunakan AWS Management Console atau AWS CLI. Saat Anda memulihkan dari snapshot menggunakan AWS CLI, Anda harus menambahkan instance secara terpisah setelah cluster tersedia.
- Membagikan — Anda tidak dapat membagikan snapshot klaster otomatis Amazon DocumentDB. Sebagai solusinya, Anda dapat membuat snapshot manual dengan menyalin snapshot otomatis, lalu membagikan salinan tersebut. Untuk informasi selengkapnya tentang menyalin snapshot, lihat Menyalin Snapshot Klaster Amazon DocumentDB. Untuk informasi selengkapnya tentang memulihkan klaster dari snapshot, lihat Memulihkan dari Snapshot Klaster.

- Anda dapat memulihkan dari titik mana pun dalam periode retensi cadangan — Karena pembaruan basis data dicatat secara bertahap, Anda dapat memulihkan klaster Anda ke setiap titik waktu dalam periode retensi cadangan.

  Saat Anda memulihkan dari snapshot otomatis atau dari point-in-time pemulihan menggunakan AWS CLI, Anda harus menambahkan instance secara terpisah setelah cluster tersedia.

Snapshot manual Amazon DocumentDB memiliki fitur utama berikut:

- Dibuat sesuai permintaan - Snapshot manual Amazon DocumentDB dibuat sesuai permintaan menggunakan Amazon DocumentDB Management Console atau. AWS CLI
- Menghapus snapshot manual — Sebuah snapshot manual dihapus hanya ketika Anda secara eksplisit menghapusnya menggunakan konsol Amazon DocumentDB atau AWS CLI. Snapshot manual tidak dihapus saat Anda menghapus klasternya.
- Backup lengkap — Ketika snapshot manual diambil, backup lengkap dari data klaster Anda dibuat dan disimpan.
- Penamaan snapshot manual — Anda menentukan nama snapshot manual. Amazon DocumentDB tidak menambahkan cap `datetime` ke nama, sehingga Anda harus menambahkan informasi itu jika Anda ingin itu disertakan dalam nama.
- Memulihkan dari snapshot manual — Anda dapat memulihkan dari snapshot manual menggunakan konsol atau AWS CLI. Saat Anda memulihkan dari snapshot menggunakan AWS CLI, Anda harus menambahkan instance secara terpisah setelah cluster tersedia.
- Service Quotas — Anda dibatasi hingga maksimal 100 snapshot manual per. Wilayah AWS
- Membagikan — Anda dapat berbagi snapshot klaster manual, yang dapat disalin oleh Akun AWS yang diotorisasi. Anda dapat berbagi snapshot manual yang terenkripsi atau tidak terenkripsi. Untuk informasi selengkapnya tentang menyalin snapshot, lihat Menyalin Snapshot Klaster Amazon DocumentDB.
- Anda memulihkan ke saat snapshot manual diambil — Saat Anda memulihkan dari snapshot manual, Anda memulihkan ke saat snapshot manual diambil.

  Saat Anda memulihkan dari snapshot menggunakan AWS CLI, Anda harus menambahkan instance secara terpisah setelah cluster tersedia.

Membuat Manual Klaster Snapshot

Anda dapat membuat snapshot manual menggunakan AWS Management Console atau AWS CLI. Jumlah waktu yang dibutuhkan untuk membuat snapshot berbeda-beda sesuai dengan ukuran dari basis data Anda. Saat Anda membuat snapshot, Anda harus melakukan hal berikut:

1. Mengidentifikasi klaster mana yang akan dicadangkan.
2. Berikan nama snapshot Anda. Hal ini memungkinkan Anda untuk memulihkan dari itu nanti.

Using the AWS Management Console

Untuk membuat snapshot manual menggunakan AWS Management Console, Anda dapat mengikuti salah satu metode di bawah ini.

1. Metode 1:

    1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
    2. Di panel navigasi, pilih Snapshot.

        > **Tip**
        >
        > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

    3. Pada halaman Snapshot, pilih Buat.
    4. Pada halaman Membuat snapshot klaster:
        a. Pengidentifikasi klaster — Dari daftar tarik-turun klaster, pilih klaster yang ingin Anda buat snapshotnya.
        b. Pengidentifikasi snapshot — Masukkan nama untuk snapshot Anda.

            Batasan penamaan snapshot:

            - Panjangnya adalah [1–255] huruf, angka, atau tanda hubung.
            - Karakter pertama harus berupa huruf.
            - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.

- Harus unik untuk semua klaster (di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB) per akun AWS , per Wilayah.

c. Pilih Buat.

2. Metode 2:

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Pada halaman Klaster, pilih tombol di sebelah kiri pada klaster yang ingin Anda snapshot.
4. Dari menu Tindakan, pilih Ambil snapshot.
5. Pada halaman Membuat snapshot klaster:

a. Pengidentifikasi snapshot — Masukkan nama untuk snapshot Anda.

Batasan penamaan snapshot:

- Panjangnya adalah [1–63] huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
- Harus unik untuk semua klaster (di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB) per akun AWS , per Wilayah.

b. Pilih Buat.

Using the AWS CLI

Untuk membuat snapshot cluster menggunakan AWS CLI, gunakan `create-db-cluster-snapshot` operasi dengan parameter berikut.

Parameter

- **`--db-cluster-identifier`** — Diperlukan. Nama klaster yang ingin Anda ambil snapshotnya. Klaster ini harus ada dan tersedia.
- **`--db-cluster-snapshot-identifier`** — Diperlukan. Nama snapshot manual yang Anda buat.

Contoh berikut membuat snapshot bernama `sample-cluster-snapshot` untuk klaster bernama `sample-cluster`.

Untuk Linux, macOS, atau Unix:

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Untuk Windows:

```
aws docdb create-db-cluster-snapshot ^
    --db-cluster-identifier sample-cluster ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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"
    }
}
```

Menyalin Snapshot Klaster Amazon DocumentDB

Di Amazon DocumentDB, Anda dapat menyalin snapshot manual dan otomatis dalam yang Wilayah AWS sama atau ke yang Wilayah AWS berbeda dalam akun yang sama. Anda juga dapat berbagi snapshot yang dimiliki oleh orang lain Akun AWS dalam hal yang sama Wilayah AWS. Namun, Anda tidak dapat menyalin snapshot cluster di seluruh Wilayah AWS dan Akun AWS dalam satu langkah. Tindakan ini harus dilakukan secara terpisah.

Sebagai alternatif untuk menyalin, Anda juga dapat berbagi foto manual dengan yang lain. Akun AWS Untuk informasi selengkapnya, lihat Membagikan Snapshot Klaster Amazon DocumentDB.

> **Note**
>
> Amazon DocumentDB menagih Anda berdasarkan jumlah backup dan data snapshot yang Anda simpan dan periode waktu Anda menyimpannya. Untuk informasi selengkapnya tentang penyimpanan yang terkait dengan backup dan snapshot Amazon DocumentDB, lihat Memahami Penggunaan Penyimpanan Backup. Untuk informasi harga tentang penyimpanan Amazon DocumentDB, lihat Harga Amazon DocumentDB.

Topik

- Menyalin Snapshot Bersama
- Menyalin Snapshot Di Seluruh Wilayah AWS
- Batasan
- Menangani Enkripsi
- Pertimbangan Grup Parameter

- Menyalin Snapshot Klaster

Menyalin Snapshot Bersama

Anda dapat menyalin snapshot yang dibagikan kepada Anda oleh orang lain Akun AWS. Jika Anda menyalin snapshot terenkripsi yang telah dibagikan dari yang lain Akun AWS, Anda harus memiliki akses ke kunci AWS KMS enkripsi yang digunakan untuk mengenkripsi snapshot.

Anda hanya dapat menyalin snapshot bersama dalam hal yang sama Wilayah AWS, apakah snapshot dienkripsi atau tidak. Untuk informasi selengkapnya, lihat Menangani Enkripsi.

Menyalin Snapshot Di Seluruh Wilayah AWS

Saat Anda menyalin snapshot ke snapshot Wilayah AWS yang berbeda dari snapshot sumber Wilayah AWS, setiap salinan adalah snapshot lengkap. Salinan snapshot lengkap berisi semua data dan metadata yang diperlukan untuk memulihkan klaster Amazon DocumentDB.

Bergantung pada yang Wilayah AWS terlibat dan jumlah data yang akan disalin, salinan snapshot lintas wilayah dapat memakan waktu berjam-jam untuk diselesaikan. Dalam beberapa kasus, mungkin ada sejumlah besar permintaan salinan snapshot lintas wilayah dari Wilayah AWS sumber tertentu. Dalam kasus ini, Amazon DocumentDB mungkin menempatkan permintaan salinan lintas wilayah baru dari Wilayah AWS sumber tersebut ke dalam antrian hingga beberapa salinan yang sedang berlangsung selesai. Tidak ada informasi kemajuan yang ditampilkan tentang penyalinan permintaan saat berada di antrean. Informasi progres akan ditampilkan saat penyalinan dimulai.

Batasan

Berikut adalah beberapa batasan saat Anda menyalin snapshot:

- Jika Anda menghapus snapshot sumber sebelum snapshot target tersedia, salinan snapshot mungkin gagal. Verifikasi bahwa snapshot target memiliki status `AVAILABLE` sebelum Anda menghapus snapshot sumber.
- Anda dapat memiliki hingga lima snapshot permintaan salinan yang sedang berlangsung ke satu Wilayah tujuan per akun.
- Tergantung pada wilayah yang terlibat dan jumlah data yang akan disalin, salinan snapshot lintas wilayah dapat memakan waktu berjam-jam untuk diselesaikan. Untuk informasi selengkapnya, lihat Menyalin Snapshot Di Seluruh Wilayah AWS.

Menangani Enkripsi

Anda dapat menyalin snapshot yang telah dienkripsi menggunakan kunci enkripsi AWS KMS . Jika Anda menyalin snapshot terenkripsi, salinan snapshot tersebut juga harus dienkripsi. Jika Anda menyalin snapshot terenkripsi dalam yang sama Wilayah AWS, Anda dapat mengenkripsi salinan dengan kunci AWS KMS enkripsi yang sama dengan snapshot asli, atau Anda dapat menentukan kunci enkripsi yang berbeda. AWS KMS Jika Anda menyalin snapshot terenkripsi di seluruh Wilayah, Anda tidak dapat menggunakan kunci AWS KMS enkripsi yang sama untuk salinan seperti yang digunakan untuk snapshot sumber, karena AWS KMS kunci khusus Wilayah. Sebagai gantinya, Anda harus menentukan AWS KMS kunci yang valid di tujuan Wilayah AWS n.

Snapshot sumber tetap terenkripsi selama proses penyalinan. Untuk informasi selengkapnya, lihat Perlindungan Data di Amazon DocumentDB.

> **Note**
>
> Untuk snapshot klaster Amazon DocumentDB, Anda tidak dapat mengenkripsi snapshot klaster yang tidak dienkripsi saat Anda menyalin snapshot.

Pertimbangan Grup Parameter

Saat Anda menyalin snapshot lintas Wilayah, salinan tersebut tidak menyertakan grup parameter yang digunakan oleh klaster Amazon DocumentDB asli. Saat Anda memulihkan snapshot untuk membuat cluster baru, cluster itu mendapatkan grup parameter default untuk Wilayah AWS itu dibuat. Untuk memberikan klaster baru parameter yang sama seperti aslinya, Anda harus melakukan hal berikut:

1. Di tujuan Wilayah AWS, buat grup parameter cluster Amazon DocumentDB dengan pengaturan yang sama dengan cluster asli. Jika sudah ada di yang baru Wilayah AWS, Anda dapat menggunakan yang itu.
2. Setelah memulihkan snapshot di tujuan Wilayah AWS, ubah cluster Amazon DocumentDB baru dan tambahkan grup parameter baru atau yang sudah ada dari langkah sebelumnya. Untuk informasi selengkapnya, lihat Memodifikasi cluster Amazon DocumentDB.

Menyalin Snapshot Klaster

Anda dapat menyalin cluster Amazon DocumentDB menggunakan atau AWS Management Console , sebagai berikut AWS CLI.

Using the AWS Management Console

Untuk membuat salinan snapshot cluster menggunakan AWS Management Console, selesaikan langkah-langkah berikut. Prosedur ini berfungsi untuk menyalin snapshot cluster terenkripsi atau tidak terenkripsi, di wilayah yang sama atau di seluruh Wilayah. Wilayah AWS

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Snapshot, lalu pilih tombol di sebelah kiri snapshot yang ingin Anda salin.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Dari menu Tindakan, pilih Salin.
4. Di halaman Buat Salinan snapshot klaster yang dihasilkan, selesaikan bagian Pengaturan.

   a. Wilayah Tujuan — Opsional. Untuk menyalin snapshot cluster ke yang lain Wilayah AWS, pilih itu Wilayah AWS untuk Wilayah Tujuan.

   b. Pengidentifikasi snapshot baru — Masukkan nama untuk snapshot baru.

      Batasan penamaan snapshot target:

      - Tidak dapat berupa nama snapshot yang sudah ada.
      - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
      - Karakter pertama harus berupa huruf.
      - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
      - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

c. Salin tag — Untuk menyalin tag apa pun yang Anda miliki di snapshot sumber ke salinan snapshot Anda, pilih Salin tag.

5. Lengkapi ncryption-at-rest bagian E.

a. Enkripsi saat istirahat — Jika snapshot Anda tidak dienkripsi, opsi ini tidak tersedia untuk Anda karena Anda tidak dapat membuat salinan terenkripsi dari snapshot yang tidak terenkripsi. Jika snapshot Anda dienkripsi, Anda dapat mengubah yang AWS KMS key digunakan selama enkripsi saat istirahat.

Untuk informasi selengkapnya tentang mengenkripsi salinan snapshot, lihat. Salin Enkripsi Snapshot Cluster

Untuk informasi selengkapnya tentang enkripsi diam, lihat Mengenkripsi Amazon DocumentDB Data at Rest.

b. AWS KMS Kunci - Dari daftar drop-down, pilih salah satu dari berikut ini:

- (default) aws/rds - Nomor akun dan ID AWS KMS kunci tercantum mengikuti opsi ini.
- **<** some-key-name **>** — Jika Anda membuat kunci, itu terdaftar dan tersedia untuk Anda pilih.
- Masukkan ARN kunci — Dalam kotak ARN, masukkan Amazon Resource Name (ARN) untuk kunci AWS KMS Anda. Format ARN adalah `arn:aws:kms:<region>:<accountID>:key/<key-id>` .

6. Untuk membuat salinan snapshot yang dipilih, pilih Salin snapshot. Sebagai alternatif, Anda dapat memilih Batalkan untuk tidak membuat salinan snapshot.

Using the AWS CLI

Untuk membuat salinan snapshot klaster yang tidak terenkripsi menggunakan AWS CLI, gunakan operasi `copy-db-cluster-snapshot` dengan parameter berikut. Jika Anda menyalin snapshot ke yang lain Wilayah AWS, jalankan perintah di mana snapshot akan disalin. Wilayah AWS

- **`--source-db-cluster-snapshot-identifier`** — Diperlukan. Pengidentifikasi snapshot klaster untuk membuat salinan. Snapshot klaster harus ada dan dalam status tersedia. Jika Anda menyalin snapshot ke yang lain Wilayah AWS, pengenal ini harus dalam format ARN untuk sumbernya. Wilayah AWS Parameter ini tidak peka huruf besar kecil.

- **`--target-db-cluster-snapshot-identifier`** — Diperlukan. Pengidentifikasi snapshot klaster baru yang akan dibuat dari snapshot klaster sumber. Parameter ini tidak peka huruf besar kecil.

  Batasan penamaan snapshot target:

  - Tidak dapat berupa nama snapshot yang sudah ada.
  - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
  - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

- **`--source-region`**— Jika Anda menyalin snapshot ke yang lain Wilayah AWS, tentukan dari Wilayah AWS mana snapshot cluster terenkripsi akan disalin.

  Jika Anda menyalin snapshot ke Wilayah AWS lain dan Anda tidak menentukan `--source-region`, Anda harus menentukan opsi `pre-signed-url` sebagai gantinya. `pre-signed-url`Nilai harus berupa URL yang berisi permintaan tanda tangan Versi 4 Tanda Tangan untuk `CopyDBClusterSnapshot` tindakan yang akan dipanggil di sumber Wilayah AWS tempat snapshot cluster disalin. Untuk mempelajari lebih lanjut tentang ini`pre-signed-url`, lihat CopyDB ClusterSnapshot.

- **`--kms-key-id`** — Pengidentifikasi kunci KMS agar kunci dapat digunakan untuk mengenkripsi salinan snapshot klaster.

  Jika Anda menyalin snapshot cluster terenkripsi ke yang lain Wilayah AWS, parameter ini diperlukan. Anda harus menentukan kunci KMS untuk tujuan Wilayah AWS.

  Jika Anda menyalin snapshot cluster terenkripsi dalam hal yang sama Wilayah AWS, parameter AWS KMS kuncinya adalah opsional. Salinan snapshot cluster dienkripsi dengan AWS KMS kunci yang sama dengan snapshot cluster sumber. Jika Anda ingin menentukan kunci AWS KMS enkripsi baru yang akan digunakan untuk mengenkripsi salinan, Anda dapat melakukannya menggunakan parameter ini.

- **`--copy-tags`** — Opsional. Tag dan nilai yang akan disalin.

Untuk membatalkan operasi penyalinan setelah ia berlangsung, Anda dapat menghapus snapshot target yang diidentifikasi oleh `--target-db-cluster-snapshot-identifier` atau

`TargetDBClusterSnapshotIdentifier` saat snapshot klaster tersebut berada dalam status menyalin.

Example

Contoh 1: Salin snapshot yang tidak terenkripsi ke Wilayah yang sama

AWS CLI Contoh berikut membuat salinan `sample-cluster-snapshot` bernama `sample-cluster-snapshot-copy` Wilayah AWS sama dengan snapshot sumber. Saat salinan dibuat, semua tag di snapshot asli disalin ke salinan snapshot.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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

Contoh 2: Salin snapshot yang tidak terenkripsi Wilayah AWS

AWS CLI Contoh berikut membuat salinan`sample-cluster-snapshot`, yang memiliki ARN`arn:aws:rds:us-east-1:123456789012:cluster-snapshot:sample-cluster-snapshot`. Salinan ini diberi nama `sample-cluster-snapshot-copy` dan berada Wilayah AWS di tempat perintah dijalankan.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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

Contoh 3: Salin snapshot terenkripsi di seluruh Wilayah AWS

AWS CLI Contoh berikut membuat salinan `sample-cluster-snapshot` dari Wilayah us-west-2 ke Wilayah us-east-1. Perintah ini dipanggil di Wilayah us-east-1.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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

Untuk informasi selengkapnya tentang mengenkripsi salinan snapshot, lihat. Salin Enkripsi Snapshot Cluster

Untuk informasi selengkapnya tentang enkripsi diam, lihat Mengenkripsi Amazon DocumentDB Data at Rest.

Membagikan Snapshot Klaster Amazon DocumentDB

Di Amazon DocumentDB, Anda dapat berbagi snapshot klaster manual, yang dapat disalin oleh Akun AWS yang diotorisasi. Anda dapat berbagi snapshot manual yang terenkripsi atau tidak terenkripsi. Saat membagikan snapshot yang tidak terenkripsi, authorized Akun AWS dapat memulihkan cluster langsung dari snapshot alih-alih membuat salinannya dan memulihkannya. Namun, Anda tidak dapat memulihkan klaster dari snapshot yang dibagikan dan dienkripsi. Sebaliknya, Anda dapat membuat salinan klaster dan memulihkan klaster dari salinan tersebut. Untuk informasi selengkapnya tentang menyalin snapshot, lihat Menyalin Snapshot Klaster Amazon DocumentDB.

> **Note**
>
> Anda tidak dapat membagikan snapshot klaster otomatis Amazon DocumentDB. Sebagai solusinya, Anda dapat membuat snapshot manual dengan menyalin snapshot otomatis, lalu membagikan salinan tersebut. Untuk informasi selengkapnya tentang menyalin snapshot, lihat Menyalin Snapshot Klaster Amazon DocumentDB. Untuk informasi selengkapnya tentang memulihkan klaster dari snapshot, lihat Memulihkan dari Snapshot Klaster.

Anda dapat berbagi snapshot manual dengan hingga 20 lainnya Akun AWS. Anda juga dapat berbagi snapshot manual yang tidak terenkripsi sebagai publik sehingga snapshot tersebut tersedia untuk semua akun . Saat membagikan snapshot sebagai publik, pastikan bahwa tidak ada informasi privat yang dimasukkan ke dalam snapshot publik Anda.

Saat berbagi snapshot manual dengan yang lain Akun AWS, dan Anda memulihkan cluster dari snapshot bersama menggunakan AWS CLI atau Amazon DocumentDB API, Anda harus menentukan Nama Sumber Daya Amazon (ARN) dari snapshot bersama sebagai pengenal snapshot.

Membagikan Snapshot Terenkripsi

Pembatasan berikut berlaku untuk berbagi snapshot terenkripsi:

- Anda tidak dapat berbagi snapshot terenkripsi sebagai publik.
- Anda tidak dapat membagikan snapshot yang telah dienkripsi menggunakan kunci AWS KMS enkripsi default akun yang membagikan snapshot.

Ikuti langkah-langkah ini untuk berbagi snapshot terenkripsi.

1. Bagikan kunci enkripsi AWS Key Management Service (AWS KMS) yang digunakan untuk mengenkripsi snapshot dengan akun apa pun yang ingin Anda akses snapshot.

   Anda dapat berbagi kunci AWS KMS enkripsi dengan AWS akun lain dengan menambahkan akun lain ke kebijakan AWS KMS kunci. Untuk detail tentang memperbarui kebijakan utama, lihat Menggunakan Kebijakan Utama di AWS KMS di Panduan AWS Key Management Service Pengembang. Untuk contoh pembuatan kebijakan kunci, lihat Membuat Kebijakan IAM untuk Memungkinkan Penyalinan Snapshot Terenkripsi nanti dalam topik ini.
2. Gunakan AWS CLI, seperti yang ditunjukkan di bawah ini, untuk membagikan snapshot terenkripsi dengan akun lain.

Mengizinkan Akses ke Kunci AWS KMS Enkripsi

Akun AWS Agar orang lain dapat menyalin snapshot terenkripsi yang dibagikan dari akun Anda, akun yang Anda bagikan snapshot Anda harus memiliki akses ke AWS KMS kunci yang mengenkripsi snapshot. Untuk mengizinkan akun lain mengakses AWS KMS kunci, perbarui kebijakan kunci untuk AWS KMS kunci dengan ARN akun yang Anda bagikan sebagai prinsipal dalam kebijakan utama AWS KMS . Kemudian izinkan tindakan `kms:CreateGrant`.

Setelah Anda memberikan akses akun ke kunci AWS KMS enkripsi Anda, untuk menyalin snapshot terenkripsi Anda, akun tersebut harus membuat pengguna AWS Identity and Access Management (IAM) jika belum memilikinya. Selain itu, akun itu juga harus melampirkan kebijakan IAM ke pengguna IAM yang memungkinkan pengguna untuk menyalin snapshot terenkripsi menggunakan kunci Anda. AWS KMS Akun harus pengguna IAM dan tidak dapat menjadi Akun AWS identitas root karena pembatasan AWS KMS keamanan.

Dalam contoh kebijakan kunci berikut, pengguna 123451234512 adalah pemilik kunci enkripsi. AWS KMS Pengguna 123456789012 adalah akun yang kuncinya sedang dibagikan dengannya. Kebijakan kunci yang diperbarui ini memberikan akses akun ke AWS KMS kunci. Ini dilakukan dengan memasukkan ARN untuk Akun AWS identitas root untuk pengguna 123456789012 sebagai prinsipal untuk kebijakan, dan dengan mengizinkan tindakan. `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}}
        }
    ]
}
```

Membuat Kebijakan IAM untuk Memungkinkan Penyalinan Snapshot Terenkripsi

Ketika eksternal Akun AWS memiliki akses ke AWS KMS kunci Anda, pemilik akun tersebut dapat membuat kebijakan untuk mengizinkan pengguna IAM yang dibuat untuk akun tersebut menyalin snapshot terenkripsi yang dienkripsi dengan kunci tersebut. AWS KMS

Contoh berikut menunjukkan kebijakan yang dapat dilampirkan ke pengguna IAM untuk Akun AWS 123456789012. Kebijakan ini memungkinkan pengguna IAM untuk menyalin snapshot bersama dari akun 123451234512 yang telah dienkripsi dengan kunci di Wilayah us-west-2. AWS KMS `c989c1dd-a3f2-4a5d-8d96-e793d082ab26`

```
{
    "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
                }
            }
        }
    ]
}
```

Untuk detail tentang memperbarui kebijakan kunci, lihat Menggunakan Kebijakan Kunci di AWS KMS dalam Panduan Developer AWS Key Management Service .

Membagikan Snapshot

Untuk berbagi snapshot, gunakan operasi `modify-db-snapshot-attribute` Amazon DocumentDB. Gunakan `--values-to-add` parameter untuk menambahkan daftar ID untuk Akun AWS yang berwenang untuk mengembalikan snapshot manual.

Contoh berikut memungkinkan dua Akun AWS pengidentifikasi, 123451234512 dan 123456789012, untuk mengembalikan snapshot bernama. `manual-snapshot1` Itu juga menghapus nilai atribut `all` untuk menandai snapshot sebagai privat.

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add '["123451234512","123456789012"]'
```

Untuk Windows:

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-add '["123451234512","123456789012"]'
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512",
                    "123456789012"
                ]
            }
        ]
    }
}
```

Untuk menghapus Akun AWS pengenal dari daftar, gunakan `--values-to-remove` parameter. Contoh berikut mencegah Akun AWS ID 123456789012 memulihkan snapshot.

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove '["123456789012"]'
```

Untuk Windows:

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-remove '["123456789012"]'
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512"
                ]
            }
        ]
    }
}
```

Memulihkan dari Snapshot Klaster

Amazon DocumentDB (dengan kompatibilitas MongoDB) membuat snapshot klaster dari volume penyimpanan Anda. Anda dapat membuat klaster baru dengan memulihkankannya dari snapshot klaster. Saat Anda memulihkan klaster, berikan nama snapshot klaster untuk memulihkannya dan nama untuk klaster baru yang dibuat oleh pemulihan. Anda tidak dapat memulihkan dari snapshot ke klaster yang ada karena klaster baru dibuat saat Anda memulihkan.

Ketika Anda memulihkan sebuah klaster dari snapshot klaster:

- Tindakan ini memulihkan hanya klaster, dan bukan instans untuk klaster tersebut. Anda harus memanggil tindakan `create-db-instance` guna membuat instans untuk klaster yang dipulihkan, menentukan pengidentifikasi klaster yang dipulihkan di `--db-cluster-identifier`. Anda dapat membuat instans hanya setelah klaster tersedia.
- Anda tidak dapat memulihkan snapshot terenkripsi ke klaster yang tidak terenkripsi. Namun, Anda dapat memulihkan snapshot yang tidak terenkripsi ke cluster terenkripsi dengan menentukan kuncinya. AWS KMS
- Untuk memulihkan cluster dari snapshot terenkripsi, Anda harus memiliki akses ke kunci. AWS KMS

> **Note**
>
> Anda tidak dapat memulihkan klaster 3.6 ke klaster 4.0 tetapi Anda dapat bermigrasi dari satu versi klaster ke versi lainnya. Untuk informasi selengkapnya, kunjungi Migrasi ke Amazon DocumentDB.

Using the AWS Management Console

Prosedur berikut menunjukkan cara untuk memulihkan klaster Amazon DocumentDB dari snapshot klaster menggunakan Konsol Manajemen Amazon DocumentDB.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Snapshot, lalu pilih tombol di sebelah kiri snapshot yang ingin Anda gunakan untuk memulihkan klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Pada menu Tindakan, pilih Pemulihan.
4. Pada halaman Pulihkan snapshot, selesaikan bagian Konfigurasi.

a. Pengidentifikasi klaster — Nama untuk klaster baru. Anda dapat menerima nama yang disediakan Amazon DocumentDB atau mengetikkan nama yang Anda inginkan. Nama yang disediakan Amazon DocumentDB adalah dalam format `docdb-` ditambah stempel waktu UTC; misalnya, `docdb-yyyy-mm-dd-hh-mm-ss`.

b. Kelas instans — Kelas instans untuk klaster baru. Anda dapat menerima kelas instans default atau memilih kelas instans dari daftar tarik-turun.

c. Jumlah instans — Jumlah instans yang ingin Anda buat dengan klaster ini. Anda dapat menerima default 3 instans (1 baca/tulis primer dan 2 replika baca-saja) atau memilih jumlah instans dari daftar tarik-turun.

5. Untuk konfigurasi penyimpanan Cluster, pilih opsi penyimpanan.

> **Note**
>
> Konfigurasi penyimpanan Amazon DocumentDB I/O-Optimized hanya tersedia di versi mesin Amazon DocumentDB 5.0.

6. Jika Anda puas dengan konfigurasi klaster, pilih Pulihkan klaster dan tunggu sementara klaster Anda dipulihkan.

7. Jika Anda lebih suka untuk mengubah beberapa konfigurasi, seperti menentukan Amazon VPC atau grup keamanan non-default, pilih Tampilkan pengaturan lanjutan dari kiri bawah halaman, dan kemudian lanjutkan dengan langkah-langkah berikut.

   a. Selesaikan bagian Pengaturan jaringan.

      - Virtual Private Cloud (VPC) — Terima VPC saat ini, atau pilih VPC dari daftar tarik-turun.
      - Grup Subnet — Terima grup subnet `default`, atau pilih salah satu dari daftar tarik-turun.
      - Grup Keamanan VPC — Terima grup keamanan `default (VPC)`, atau pilih salah satu dari daftar.

   b. Lengkapi bagian Opsi klaster.

      - Port basis data — Terima port default, `27017`, atau gunakan panah atas atau bawah untuk mengatur port yang ingin Anda gunakan untuk koneksi aplikasi.

   c. Lengkapi bagian Enkripsi.

- Enkripsi saat istirahat — Jika snapshot Anda dienkripsi, opsi ini tidak tersedia untuk Anda. Jika itu tidak dienkripsi, Anda dapat memilih salah satu hal berikut:
  - Untuk mengenkripsi semua data klaster Anda, pilih Aktifkan encryption-at-rest. Jika Anda memilih opsi ini, Anda harus menunjuk kunci KMS.
  - Untuk tidak mengenkripsi data klaster Anda, pilih Nonaktifkan encryption-at-rest. Jika Anda memilih opsi ini, Anda telah selesai dengan bagian enkripsi.
- AWS KMS Kunci - Pilih salah satu dari berikut ini dari daftar drop-down:
  - (default) aws/rds - Nomor akun dan ID AWS KMS kunci tercantum mengikuti opsi ini.
  - Kunci yang dikelola pelanggan — Opsi ini hanya tersedia jika Anda membuat kunci enkripsi IAM di konsol AWS Identity and Access Management (IAM). Anda dapat memilih kunci untuk mengenkripsi klaster Anda.
  - Masukkan kunci ARN — Di kotak ARN, masukkan Nama Sumber Daya Amazon (ARN) untuk kunci Anda. AWS KMS Format ARN adalah `arn:aws:kms:<region>:<accountID>:key/<key-id>`.

d. Lengkapi bagian Ekspor log.

- Pilih jenis log untuk dipublikasikan CloudWatch - Pilih salah satu dari berikut ini:
  - Diaktifkan - Memungkinkan cluster Anda untuk mengekspor logging DDL ke Amazon CloudWatch Logs.
  - Dinonaktifkan - Mencegah klaster Anda mengekspor log DDL ke Amazon CloudWatch Logs. Dinonaktifkan adalah default.
- IAM role — Dari daftar, pilih Peran Tertaut Layanan RDS.

e. Lengkapi bagian Tag.

- Tambahkan Tag — Di kotak Kunci, masukkan nama untuk tag untuk klaster Anda. Di kotak Nilai, secara opsional masukkan nilai tag. Tag digunakan dengan kebijakan AWS Identity and Access Management (IAM) untuk mengelola akses ke sumber daya Amazon DocumentDB dan untuk mengontrol tindakan apa yang dapat diterapkan pada sumber daya.

f. Lengkapi bagian Perlindungan penghapusan.

- Aktifkan perlindungan penghapusan — Melindungi klaster agar tidak terhapus secara tidak sengaja. Saat opsi ini diaktifkan, Anda tidak dapat menghapus klaster.

8. Pilih Pulihkan klaster.

Using the AWS CLI

Untuk mengembalikan cluster dari snapshot menggunakan AWS CLI, gunakan `restore-db-cluster-from-snapshot` operasi dengan parameter berikut. Untuk informasi selengkapnya, lihat RestoreDBClusterFromSnapshot.

- **`--db-cluster-identifier`** — Diperlukan. Nama klaster yang dibuat oleh operasi. Sebuah klaster dengan nama ini tidak boleh ada sebelum operasi ini.

  Aturan penamaan klaster:

  - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
  - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS
- **`--snapshot-identifier`** — Diperlukan. Nama snapshot yang digunakan untuk memulihkan. Snapshot dengan nama ini harus ada dan dalam status tersedia.
- **`--engine`** — Diperlukan. Harus berupa `docdb`.
- **`--storage-type standard | iopt1`** — Opsional. Default: `standard`.
- **`--kms-key-id`** — Opsional. ARN pengidentifikasi AWS KMS kunci yang akan digunakan saat memulihkan snapshot terenkripsi atau mengenkripsi cluster saat memulihkan dari snapshot yang tidak terenkripsi. Menyediakan ID AWS KMS kunci menghasilkan cluster yang dipulihkan dienkripsi dengan AWS KMS kunci, terlepas dari apakah snapshot dienkripsi atau tidak.

  Format `--kms-key-id` adalah `arn:aws:kms:<region>:<accountID>:key/<key-id>`. Jika Anda tidak menentukan nilai untuk parameter `--kms-key-id`, maka hal berikut akan terjadi:

  - Jika snapshot `--snapshot-identifier` dienkripsi, maka cluster yang dipulihkan dienkripsi menggunakan AWS KMS kunci yang sama yang digunakan untuk mengenkripsi snapshot.
  - Jika snapshot di `--snapshot-identifier` tidak dienkripsi, maka klaster yang dipulihkan tidak dienkripsi.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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
    }
}
```

Setelah status klaster tersedia, buat setidaknya satu instans untuk klaster.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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"
    }
}
```

Memulihkan ke Titik Waktu

Anda dapat memulihkan klaster ke titik waktu mana pun yang berada dalam periode retensi cadangan klaster menggunakan AWS Management Console or AWS Command Line Interface (AWS CLI).

> **Note**
>
> Anda tidak dapat melakukan point-in-time pemulihan klaster 3.6 ke kluster 4.0 tetapi Anda dapat bermigrasi dari satu versi cluster ke versi klaster lainnya. Untuk informasi selengkapnya, kunjungi Migrasi ke Amazon DocumentDB.

Ingatlah hal-hal berikut saat memulihkan klaster ke titik waktu tertentu.

- Klaster baru dibuat dengan konfigurasi yang sama seperti klaster sumber, kecuali klaster baru dibuat dengan grup parameter default. Untuk mengatur grup parameter klaster baru ke grup parameter klaster sumber, modifikasi klaster setelah tersedia. Untuk informasi selengkapnya tentang memodifikasi klaster, lihat Memodifikasi cluster Amazon DocumentDB.

Using the AWS Management Console

Anda dapat memulihkan klaster ke point-in-time dalam periode retensi cadangannya dengan menyelesaikan yang berikut ini menggunakan AWS Management Console.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Pada panel navigasi, silakan pilih Klaster. Dalam daftar klaster, pilih tombol di sebelah kiri pada klaster yang ingin Anda pulihkan.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Pada menu Tindakan, pilih Pulihkan ke titik waktu.

4. Lengkapi bagian Waktu pemulihan, yang menentukan tanggal dan waktu untuk memulihkan ke.

   a. Tanggal pemulihan — Pilih atau masukkan tanggal yang berada antara Waktu pemulihan paling awal dan Waktu pemulihan terbaru.

   b. Waktu pemulihan— Pilih atau masukkan jam, menit, dan detik yang berada antara Waktu pemulihan paling awal dan Waktu pemulihan terbaru.

5. Lengkapi bagian Konfigurasi.

   a. Pengidentifikasi klaster — Terima pengidentifikasi default, atau masukkan pengidentifikasi yang Anda inginkan.

      Aturan penamaan klaster:

      - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
      - Karakter pertama harus berupa huruf.
      - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
      - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

   b. Kelas instans — Dari daftar tarik-turun, pilih kelas instans yang Anda inginkan untuk instans klaster.

   c. Jumlah instans — Dari daftar tarik-turun, pilih jumlah instans yang Anda ingin buat ketika klaster dipulihkan.

6. Untuk konfigurasi penyimpanan Cluster, pilih opsi penyimpanan.

   > **Note**
   >
   > Konfigurasi penyimpanan Amazon DocumentDB I/O-Optimized hanya tersedia di versi mesin Amazon DocumentDB 5.0.

7. Tidak wajib. Untuk mengonfigurasi pengaturan jaringan, opsi klaster, dan mengaktifkan ekspor log, pilih Tampilkan pengaturan lanjutan, dan kemudian lengkapi bagian berikut. Jika tidak, lanjutkan dengan langkah berikutnya.

   - Pengaturan jaringan

     1. Virtual Private Cloud (VPC) — Dari daftar tarik-turun, pilih VPC yang ingin Anda gunakan untuk klaster ini.

2. Grup Subnet — Dari daftar tarik-turun, pilih grup subnet untuk klaster ini.
3. Grup keamanan VPC — Dari daftar tarik-turun, pilih grup keamanan VPC untuk klaster ini.

- Opsi cluster
  1. Port — Terima port default (27017), atau gunakan panah atas dan bawah untuk mengatur port untuk berkomunikasi dengan klaster ini.

- Ekspor log
  1. Log audit - Pilih opsi ini untuk mengaktifkan mengekspor log audit ke Amazon CloudWatch Logs. Jika Anda memilih opsi ini, Anda harus mengaktifkan `audit_logs` di grup parameter kustom klaster. Untuk informasi selengkapnya, lihat Mengaudit Peristiwa Amazon DocumentDB.
  2. Log profiler - Pilih opsi ini untuk mengaktifkan mengekspor log profiler operasi ke Amazon Logs. CloudWatch Jika Anda memilih opsi ini, Anda juga harus mengubah parameter berikut dalam grup parameter kustom klaster:
     - `profiler` — Atur ke `enabled`.
     - `profiler_threshold_ms` — Atur ke nilai `[0-INT_MAX]` untuk mengatur ambang batas untuk operasi pembuatan profil.
     - `profiler_sampling_rate` — Atur ke nilai `[0.0-1.0]` untuk mengatur persentase operasi lambat ke profil.

     Untuk informasi selengkapnya, lihat Membuat Profil Operasi Amazon DocumentDB.
  3. Log profiler - Ekspor log profiler ke Amazon CloudWatch
  4. IAM role — Dari daftar tarik-turun, pilih Peran Tertaut Layanan RDS.
- Tanda
  1. Tambahkan Tag — Di kotak Kunci, masukkan nama untuk tag untuk klaster Anda. Di kotak Nilai, secara opsional masukkan nilai tag. Tag digunakan dengan kebijakan AWS Identity and Access Management (IAM) untuk mengelola akses ke sumber daya Amazon DocumentDB dan untuk mengontrol tindakan apa yang dapat diterapkan ke sumber daya.
- Perlindungan penghapusan

1. Aktifkan perlindungan penghapusan — Melindungi klaster agar tidak terhapus secara tidak sengaja. Saat opsi ini diaktifkan, Anda tidak dapat menghapus klaster.

8. Untuk memulihkan klaster, pilih Buat klaster. Atau, Anda dapat memilih Batalkan untuk membatalkan operasi.

Using the AWS CLI

Untuk memulihkan klaster ke titik waktu menggunakan periode retensi cadangan snapshot, gunakan operasi `restore-db-cluster-to-point-in-time` dengan parameter berikut.

- **`--db-cluster-identifier`** — Diperlukan. Nama klaster baru yang akan dibuat. Klaster ini tidak boleh ada sebelum operasi. Nilai parameter harus memenuhi aturan berikut.

  Aturan penamaan klaster:

  - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
  - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

- **`--restore-to-time`** — Tanggal dan waktu UTC untuk memulihkan klaster. Misalnya, `2018-06-07T23:45:00Z`.

  Aturan Waktu:

  - Harus sebelum waktu pemulihan terbaru untuk klaster.
  - Harus ditentukan jika parameter `--use-latest-restorable-time` tidak disediakan.
  - Tidak dapat ditentukan jika parameter `--use-latest-restorable-time` adalah `true`.
  - Tidak dapat ditentukan jika nilai parameter `--restore-type` adalah `copy-on-write`.

- **`--source-db-cluster-identifier`** — Nama klaster sumber yang akan dipulihkan. Klaster ini harus ada dan tersedia.

- **`--use-latest-restorable-time`** atau **`--no-use-latest-restorable-time`** — Apakah untuk memulihkan ke waktu backup terbaru yang dapat dipulihkan. Tidak dapat ditentukan jika parameter `--restore-to-time` disediakan.

- **`--storage-type standard | iopt1`** — Opsional. Default: `standard`.

AWS CLI Operasi `restore-db-cluster-to-point-in-time` hanya mengembalikan cluster, bukan instance untuk cluster itu. Anda harus memanggil operasi `create-db-instance` guna membuat instans untuk klaster yang dipulihkan, menentukan pengidentifikasi klaster yang dipulihkan di `--db-cluster-identifier`. Anda dapat membuat instans hanya setelah operasi `restore-db-cluster-to-point-in-time` telah selesai dan klaster yang dipulihkan tersedia.

Example

Contoh berikut membuat `sample-cluster-restored` dari snapshot `sample-cluster-snapshot` ke waktu pemulihan terbaru.

Untuk Linux, macOS, atau 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
```

Untuk 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

Contoh berikut membuat `sample-cluster-restored` dari snapshot `sample-cluster-snapshot` hingga 03:15 pada 11 Desember 2018 (UTC), yang berada dalam periode retensi cadangan `sample-cluster`.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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
    }
}
```

Menghapus Snapshot Klaster

Snapshot manual adalah cadangan lengkap yang dihapus hanya ketika Anda menghapusnya secara manual menggunakan AWS Management Console atau AWS CLI. Anda tidak dapat secara manual menghapus snapshot otomatis karena snapshot otomatis dihapus hanya ketika periode retensi snapshot berakhir atau Anda menghapus klaster snapshot.

Using the AWS Management Console

Untuk menghapus snapshot cluster manual menggunakan AWS Management Console, selesaikan langkah-langkah berikut.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Snapshot.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Dalam daftar snapshot, pilih tombol di sisi kiri snapshot yang ingin Anda hapus. Jenis snapshot harus manual.

   1. Anda dapat memverifikasi bahwa jenis snapshot adalah manual dengan memeriksa apakah itu terdaftar sebagai `manual` atau `automatic` pada kolom Jenis.

4. Dari menu Tindakan, pilih Hapus. Jika opsi Hapus tidak tersedia, Anda mungkin memilih snapshot otomatis.
5. Pada layar konfirmasi penghapusan, untuk menghapus snapshot, pilih Hapus. Untuk tetap menyimpan snapshot, pilih Batalkan.

Using the AWS CLI

Sebuah snapshot klaster manual Amazon DocumentDB adalah backup lengkap yang dapat Anda hapus secara manual menggunakan AWS CLI. Anda tidak dapat menghapus snapshot otomatis secara manual.

Untuk menghapus snapshot cluster manual menggunakan AWS CLI, gunakan `delete-db-cluster-snapshot` operasi dengan parameter berikut.

Parameter

- **`--db-cluster-snapshot-identifier`** — Diperlukan. Nama snapshot manual yang akan dihapus.

Contoh berikut menghapus snapshot klaster `sample-cluster-snapshot`.

Untuk Linux, macOS, atau Unix:

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Untuk Windows:

```
aws docdb delete-db-cluster-snapshot ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Output dari operasi ini mencantumkan detail snapshot klaster yang Anda hapus.

Mengelola Sumber Daya Amazon DocumentDB

Bagian ini mencakup berbagai komponen dan tugas terkaitnya untuk mengelola implementasi Amazon DocumentDB (dengan kompatibilitas MongoDB) Anda.

Topik

- Gambaran Umum Tugas Operasi Amazon DocumentDB
- Gambaran Umum dari Klaster Global Amazon DocumentDB
- Mengelola cluster Amazon DocumentDB
- Mengelola Instans Amazon DocumentDB
- Mengelola grup subnet Amazon DocumentDB
- Ketersediaan Tinggi dan Replikasi Amazon DocumentDB
- Mengelola indeks Amazon DocumentDB
- Mengelola kompresi dokumen tingkat koleksi
- Mengelola Peristiwa Amazon DocumentDB
- Memilih Wilayah dan Availability Zone
- Mengelola grup parameter cluster Amazon DocumentDB
- Memahami Titik Akhir Amazon DocumentDB
- Memahami Amazon DocumentDB Amazon Resource Name (ARN)
- Penandaan Sumber Daya Amazon DocumentDB
- Memelihara Amazon DocumentDB
- Memahami Peran Tertaut Layanan

Gambaran Umum Tugas Operasi Amazon DocumentDB

Bagian ini mencakup tugas operasi untuk klaster Amazon DocumentDB (dengan kompatibilitas MongoDB), dan cara untuk menyelesaikan tugas ini menggunakan AWS CLI.

Topik

- Menambahkan Replika ke Klaster Amazon DocumentDB
- Menggambarkan klaster dan instans

- Membuat Snapshot Klaster
- Memulihkan dari snapshot
- Menghapus sebuah Instans dari sebuah Klaster
- Menghapus sebuah Klaster

Menambahkan Replika ke Klaster Amazon DocumentDB

Setelah Anda membuat instans primer untuk klaster Amazon DocumentDB, Anda dapat menambahkan satu atau beberapa replika. Replika adalah instans baca-saja yang melayani dua tujuan:

- Skalabilitas — Jika Anda memiliki banyak klien yang memerlukan akses bersamaan, Anda dapat menambahkan lebih banyak replika untuk penskalaan baca.
- Ketersediaan tinggi — Jika instans primer gagal, Amazon DocumentDB secara otomatis gagal ke instans replika dan menetapkannya sebagai primer baru. Jika replika gagal, instans lain di klaster masih dapat melayani permintaan hingga node yang gagal dapat dipulihkan.

Setiap klaster Amazon DocumentDB dapat mendukung hingga 15 replika.

> **Note**
>
> Untuk toleransi kesalahan maksimum, Anda harus menerapkan replika di Availability Zone yang terpisah. Ini membantu memastikan bahwa klaster Amazon DocumentDB Anda dapat terus berfungsi, bahkan jika seluruh Availability Zone menjadi tidak tersedia.

Contoh AWS CLI berikut menunjukkan cara menambahkan replika baru. Parameter `--availability-zone` menempatkan replika di Availability Zone yang ditentukan.

```
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
```

Menggambarkan klaster dan instans

Contoh AWS CLI berikut mencantumkan semua klaster Amazon DocumentDB di suatu Wilayah. Untuk fitur pengelolaan tertentu seperti pengelolaan siklus hidup klaster dan instans, Amazon DocumentDB memanfaatkan teknologi operasi yang dibagikan dengan Amazon RDS. Parameter filter `filterName=engine,Values=docdb` hanya mengembalikan klaster Amazon DocumentDB.

Untuk informasi lebih lanjut tentang mendeskripsikan dan memodifikasi klaster, lihat Siklus hidup klaster Amazon DocumentDB.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Output dari operasi ini terlihat seperti berikut ini.

```
{
    "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",
            ...
        }
    ]
}
```

Contoh AWS CLI berikut mencantumkan instans dalam klaster Amazon DocumentDB. Untuk informasi lebih lanjut tentang mendeskripsikan dan memodifikasi cluster, lihat Siklus hidup instance Amazon DocumentDB.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Output terlihat seperti di bawah ini. Dalam output ini, ada dua instans. Instans primer adalah `sample-instance-1` (`"IsClusterWriter": true`). Ada juga instans replika, `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
```

```
            }
        ]
    ]
]
```

Membuat Snapshot Klaster

Snapshot klaster adalah cadangan lengkap data di klaster Amazon DocumentDB Anda. Saat snapshot sedang dibuat, Amazon DocumentDB membaca data Anda langsung dari volume klaster. Karena ini, Anda dapat membuat snapshot meskipun klaster Anda tidak memiliki instans yang berjalan pada saat itu. Jumlah waktu yang diperlukan untuk membuat snapshot bergantung pada ukuran volume klaster Anda.

Amazon DocumentDB mendukung pencadangan otomatis, yang terjadi setiap hari selama jendela pencadangan pilihan — periode waktu 30 menit pada siang hari. Contoh AWS CLI berikut menunjukkan cara melihat jendela cadangan untuk klaster Anda:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].PreferredBackupWindow'
```

Output menunjukkan jendela cadangan (dalam UTC):

```
[
    "00:18-00:48"
]
```

Anda dapat menentukan jendela pencadangan saat membuat klaster Amazon DocumentDB Anda. Anda juga dapat mengubah jendela pencadangan, seperti yang ditunjukkan pada contoh berikut. Jika Anda tidak menentukan jendela pencadangan, Amazon DocumentDB secara otomatis menugaskan satu ke klaster Anda.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --preferred-backup-window "02:00-02:30"
```

Selain pencadangan otomatis, Anda dapat membuat snapshot klaster secara manual kapan saja. Saat melakukannya, Anda menentukan klaster mana yang ingin Anda cadangkan, dan nama unik untuk snapshot Anda sehingga Anda dapat memulihkannya nanti.

Contoh AWS CLI berikut menunjukkan cara membuat snapshot data Anda.

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Memulihkan dari snapshot

Anda dapat memulihkan snapshot klaster ke klaster Amazon DocumentDB baru. Untuk melakukannya, Anda memberikan nama snapshot dan nama klaster baru. Anda tidak dapat memulihkan dari snapshot ke klaster yang ada; sebagai gantinya, Amazon DocumentDB membuat klaster baru saat Anda memulihkan dan kemudian mengisinya dengan data snapshot Anda.

Contoh berikut menunjukkan semua snapshot untuk klaster `sample-cluster`.

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotType,Status]'
```

Outputnya terlihat seperti berikut ini. Snapshot manual adalah snapshot yang Anda buat secara manual, sedangkan snapshot otomatis dibuat oleh Amazon DocumentDB dalam jendela pencadangan klaster.

```
[
        "sample-cluster-snapshot",
        "manual",
        "available"
    ],
    [
        "rds:sample-cluster",
        "automated",
        "available"
    ]
]
```

Contoh berikut menunjukkan cara memulihkan klaster Amazon DocumentDB dari snapshot.

```
aws docdb restore-db-cluster-from-snapshot \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster \
    --snapshot-identifier sample-cluster-snapshot
```

Klaster baru tidak memiliki instans yang terkait dengannya; jadi jika Anda ingin berinteraksi dengan klaster, Anda harus menambahkan instans ke dalamnya.

```
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
```

Anda dapat menggunakan operasi AWS CLI berikut untuk memantau kemajuan pembuatan klaster dan instans. Saat status klaster dan instans tersedia, Anda dapat terhubung ke titik akhir klaster baru dan mengakses data Anda.

```
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]'
```

Menghapus sebuah Instans dari sebuah Klaster

Amazon DocumentDB menyimpan semua data Anda dalam volume klaster. Data tetap ada dalam volume klaster tersebut, meskipun Anda menghapus semua instans dari klaster Anda. Jika Anda perlu mengakses data lagi, Anda dapat menambahkan instans ke klaster kapan saja, dan melanjutkan dari bagian terakhir yang Anda tinggalkan.

Contoh berikut menunjukkan cara menghapus instans dari klaster Amazon DocumentDB Anda.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-instance
```

Menghapus sebuah Klaster

Sebelum Anda dapat menghapus klaster Amazon DocumentDB, Anda harus menghapus semua instansnya terlebih dahulu. Contoh AWS CLI berikut mengembalikan informasi tentang instans dalam sebuah klaster. Jika operasi ini mengembalikan pengidentifikasi instans apa pun, Anda harus

menghapus setiap instans. Untuk informasi selengkapnya, lihat Menghapus sebuah Instans dari sebuah Klaster.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterMembers[*].DBInstanceIdentifier'
```

Jika tidak ada lagi instans yang tersisa, Anda dapat menghapus klaster. Pada saat itu, Anda harus memilih salah satu dari opsi berikut:

- Buat snapshot akhir — Tangkap semua data klaster dalam snapshot sehingga Anda dapat membuat ulang instans baru dengan data tersebut nanti. Contoh berikut menunjukkan cara melakukannya:

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --final-db-snapshot-identifier sample-cluster-snapshot
  ```

- Lewati snapshot terakhir — Buang semua data klaster secara permanen. Ini tidak bisa dibalik. Contoh berikut menunjukkan cara melakukannya:

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --skip-final-snapshot
  ```

Gambaran Umum dari Klaster Global Amazon DocumentDB

Apa itu klaster global?

Sebuah klaster global terdiri dari satu wilayah primer dan hingga lima wilayah sekunder baca saja. Anda menerbitkan operasi tulis langsung ke klaster primer di wilayah primer dan Amazon DocumentDB secara otomatis mereplikasi data ke wilayah sekunder menggunakan infrastruktur khusus. Latensi biasanya di bawah satu detik.

Bagaimana klaster global berguna?

- Pemulihan dari pemadaman di seluruh wilayah — Jika terjadi pemadaman di seluruh wilayah, Anda dapat mempromosikan salah satu cluster sekunder ke cluster utama dalam beberapa menit,

dengan Tujuan Waktu Pemulihan (RTO) khas kurang dari satu menit. Recovery Point Objective (RPO) biasanya diukur dalam hitungan detik, tetapi ini tergantung pada lag di seluruh jaringan pada saat kegagalan.

- Pembacaan global dengan latensi lokal — Jika Anda memiliki kantor di seluruh dunia, Anda dapat menggunakan klaster global untuk terus memperbarui sumber informasi utama Anda di wilayah primer. Kantor di wilayah Anda yang lain dapat mengakses informasi di wilayah mereka sendiri, dengan latensi lokal.
- Klaster sekunder yang dapat diskalakan — Anda dapat menskalakan klaster sekunder dengan menambahkan lebih banyak instans baca-saja ke wilayah sekunder. Klaster sekunder bersifat baca-saja, sehingga dapat mendukung hingga 16 instans replika hanya-baca daripada batas biasa 15 untuk satu klaster.
- Replikasi cepat dari klaster primer ke sekunder — Replikasi yang dilakukan oleh klaster global memiliki dampak performa yang kecil pada klaster basis data primer. Sumber daya instans DB sepenuhnya dikhususkan untuk melayani beban kerja baca dan tulis aplikasi.

Apa keterbatasan terkini klaster global?

- Klaster global tidak didukung di Amazon DocumentDB v3.6.
- Cluster global tidak didukung pada jenis instans t3, t4g, dan r4.
- Cluster global tidak tersedia di wilayah berikut: Amerika Selatan (São Paulo), Eropa (Milan), China (Beijing), dan China (Ningxia).
- Jika terjadi failover wilayah, Anda harus secara manual mempromosikan klaster sekunder menjadi klaster primer, dan memodifikasi aplikasi Anda agar mengarah ke klaster primer yang baru.
- Hanya klaster primer yang melakukan operasi penulisan. Klien yang melakukan operasi tulis terhubung ke titik akhir klaster dari klaster primer.
- Anda dapat memiliki maksimal lima region sekunder dan satu region primer untuk klaster Anda.
- Klaster sekunder tidak dapat dihentikan. Sebuah klaster primer tidak dapat dihentikan jika ia memiliki klaster sekunder yang terkait dengannya. Hanya klaster wilayah yang tidak memiliki klaster sekunder yang dapat dihentikan.
- Replika yang dilampirkan ke klaster sekunder dapat dimulai ulang dalam keadaan tertentu. Jika instans wilayah primer dimulai ulang atau gagal, replika di region sekunder juga dimulai ulang. Klaster kemudian tidak tersedia sampai semua replika kembali disinkronkan dengan instans penulis klaster basis data primer. Perilaku ini yang diharapkan. Pastikan Anda memahami dampak terhadap klaster global Anda sebelum membuat perubahan pada klaster primer Anda.

- Anda tidak dapat menggunakan perubahan aliran pada klaster sekunder.

Topik

- Panduan Quick Start: Klaster Global
- Mengelola klaster global Amazon DocumentDB
- Terhubung ke Klaster Global Amazon DocumentDB
- Memantau Klaster Global Amazon DocumentDB
- Pemulihan Bencana dan Cluster Global Amazon DocumentDB

Panduan Quick Start: Klaster Global

Topik

- Konfigurasi
- Membuat klaster global Amazon DocumentDB
- Menambahkan sebuah Wilayah AWS untuk klaster global Amazon DocumentDB
- Menggunakan snapshot untuk klaster global Amazon DocumentDB Anda

Konfigurasi

Klaster global Amazon DocumentDB mencakup setidaknya dua Wilayah AWS. Wilayah primer mendukung klaster yang memiliki satu instans primer (penulis) dan hingga lima belas instans replika, sedangkan wilayah sekunder menjalankan klaster baca saja yang seluruhnya terdiri dari hingga enam belas instans replika. Klaster global dapat memiliki sampai dengan lima wilayah sekunder. Tabel mencantumkan klaster, instans, dan replika maksimum yang diizinkan dalam klaster global.

Deskripsi	Wilayah AWS Primer	Sekunder Wilayah AWS
Klaster	1	5 (maksimum)
Instans penulis	1	0
Instans baca saja (replika Amazon DocumentDB), per klaster	15 (maks)	16 (total)

Deskripsi	Wilayah AWS Primer	Sekunder Wilayah AWS
Instans read-only (maks yang diperbolehkan, mengingat jumlah aktual dari region sekunder)	15 - s	s = jumlah total sekunder Wilayah AWS

Klaster memiliki persyaratan khusus berikut:

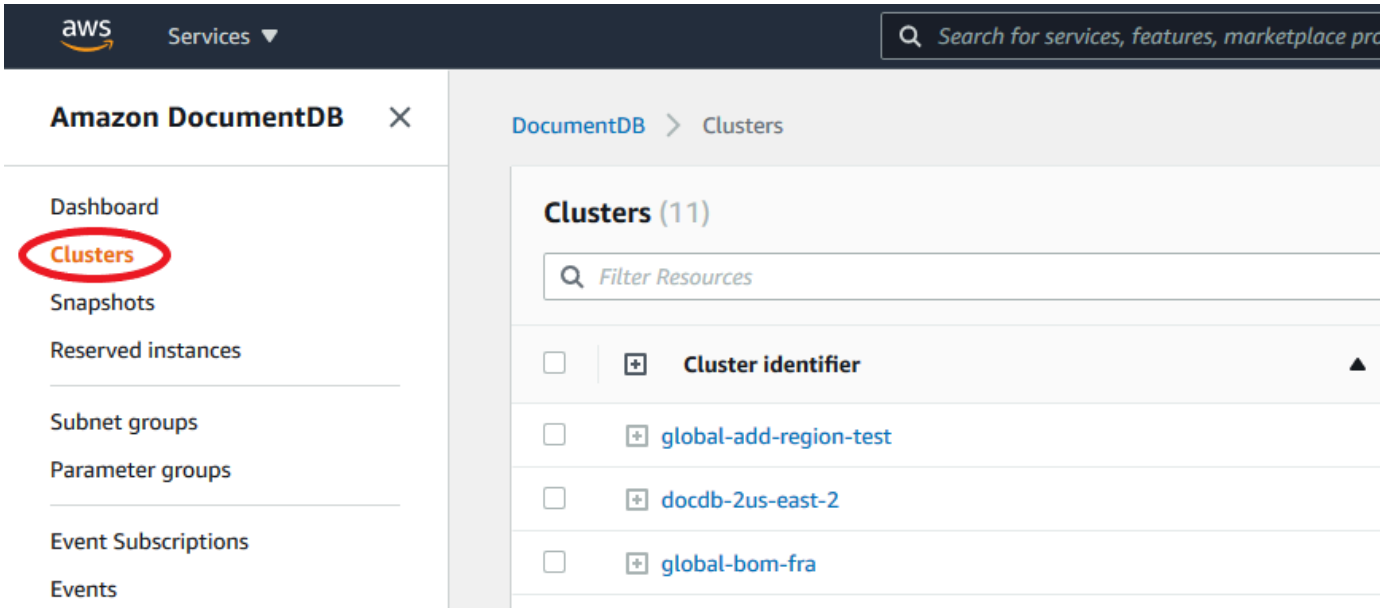
- Persyaratan kelas instance database - Anda hanya dapat menggunakan kelas `db.r5` dan `db.r6` instance.
- Wilayah AWSPersyaratan— Klaster primer harus berada di satu wilayah, dan setidaknya satu klaster sekunder harus berada di wilayah berbeda dari akun yang sama. Anda dapat membuat hingga lima klaster sekunder (baca saja), dan masing-masing klaster harus berada di wilayah yang berbeda. Dengan kata lain, tidak ada dua klaster yang bisa berada di wilayah yang sama.
- Persyaratan penamaan — Nama yang Anda pilih untuk setiap klaster harus unik, di semua wilayah. Anda tidak dapat menggunakan nama yang sama untuk klaster yang berbeda meskipun berada di wilayah yang berbeda.

Membuat klaster global Amazon DocumentDB

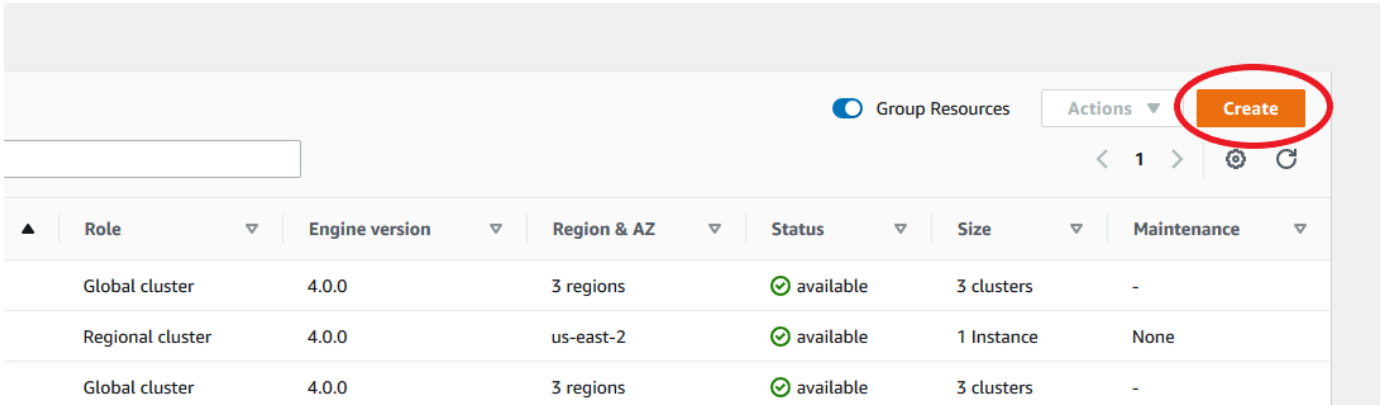
Apakah Anda siap untuk membangun klaster global pertama Anda? Di bagian ini kami akan menjelaskan cara membuat klaster global baru dengan klaster dan instans basis data baru, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

Menggunakan AWS Management Console

1. Di AWS Management Console, navigasikan ke Amazon DocumentDB.
2. Ketika Anda sampai ke konsol Amazon DocumentDB, pilih Klaster.

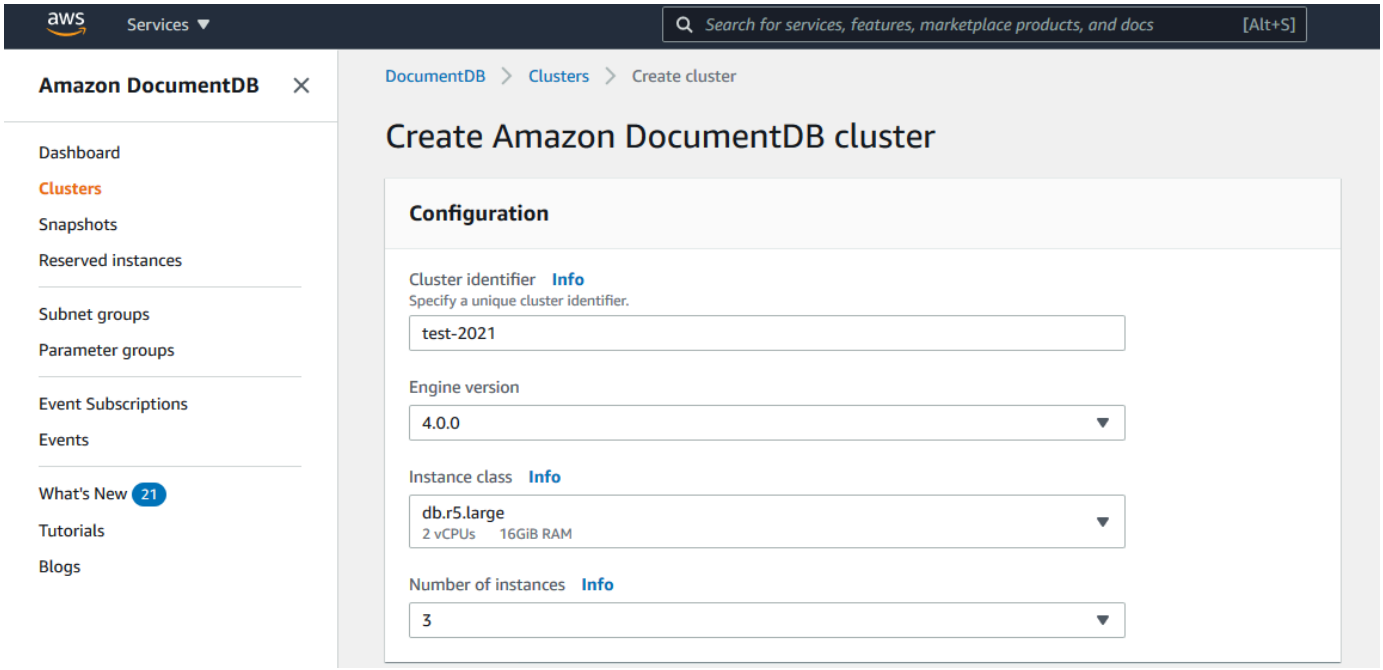


3. Pilih Buat.

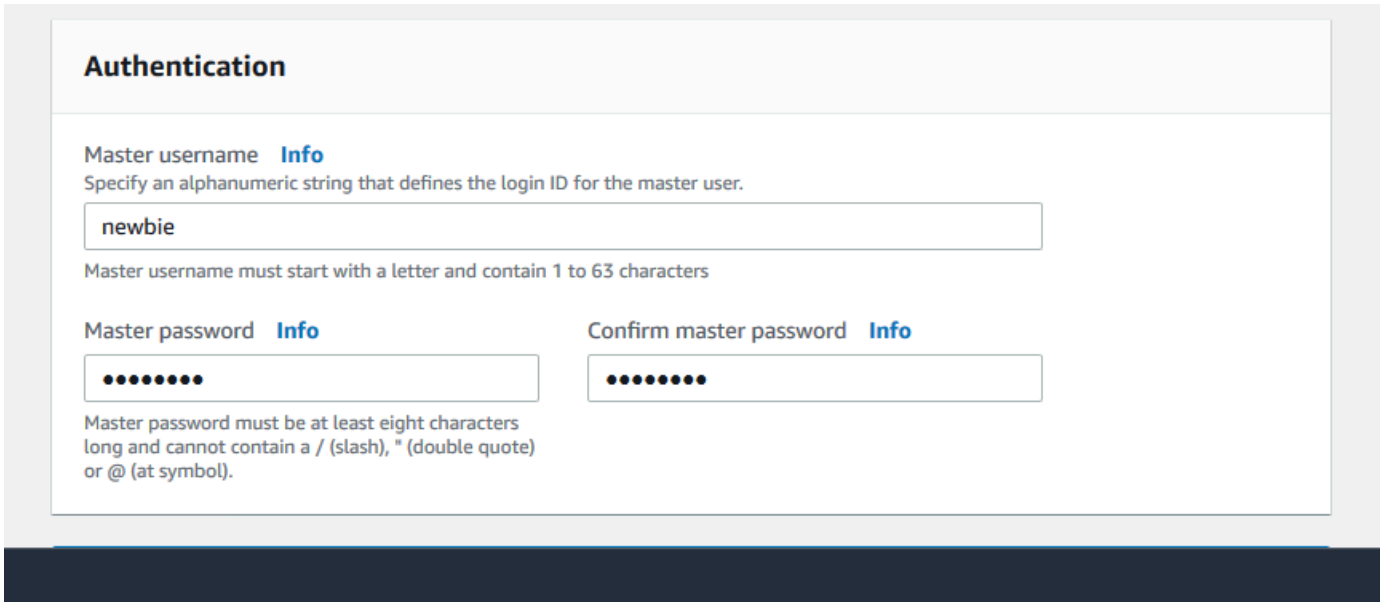


4. Mengisi bagian Konfigurasi dari formulir Buat Klaster Amazon DocumentDB dengan sesuai:

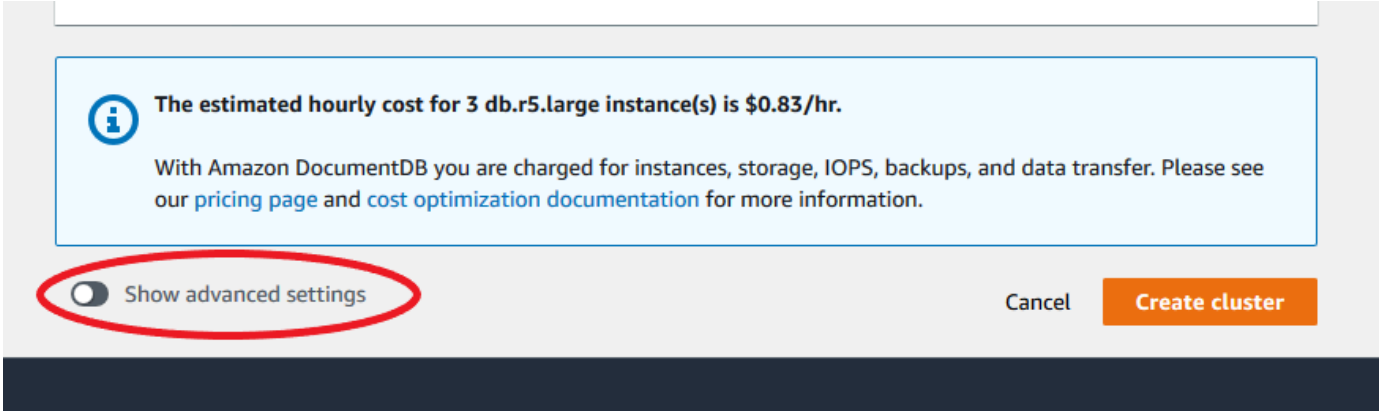
   - Pengidentifikasi klaster: Anda dapat memasukkan pengidentifikasi unik untuk instans ini atau mengizinkan Amazon DocumentDB menyediakan pengidentifikasi instans berdasarkan pengidentifikasi klaster.
   - Versi mesin: Pilih 4.0.0
   - Kelas instans: Pilih db.r5.large
   - Jumlah instans: Pilih 3.



5. Di Autentikasi, isi nama pengguna utama dan kata sandi utama.

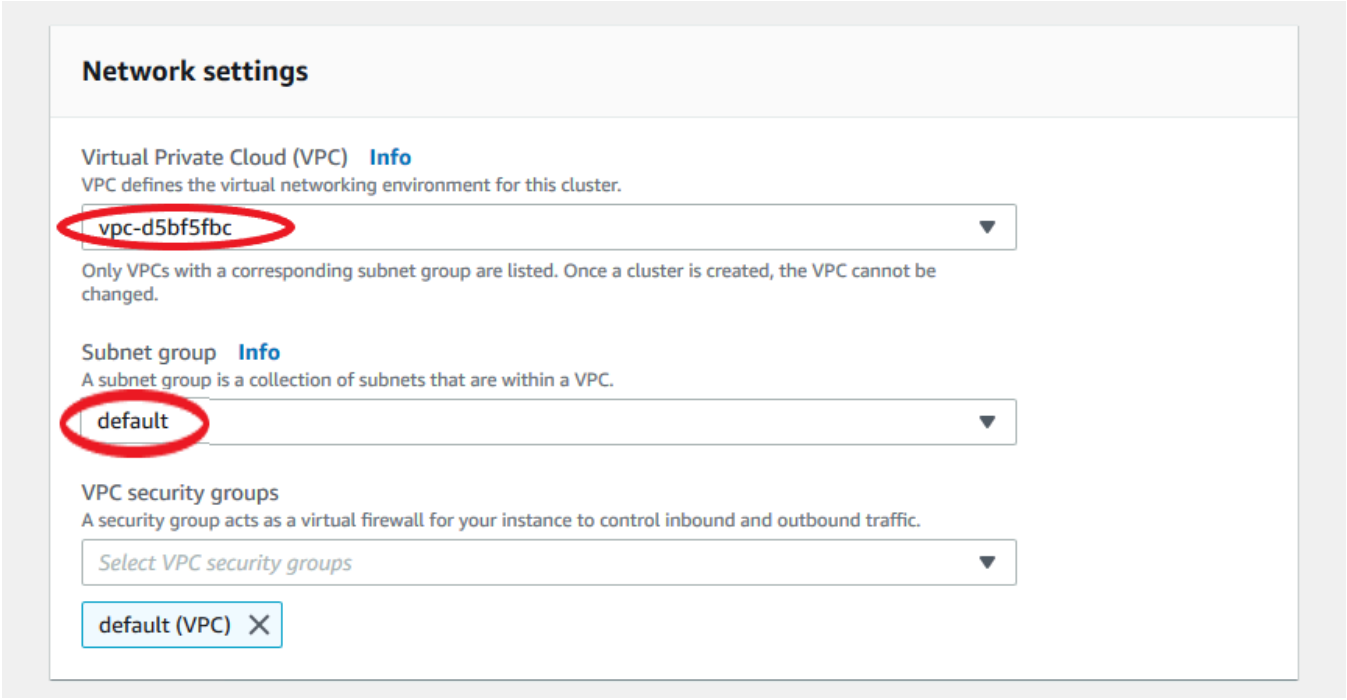


6. Pilih Pengaturan Lanjutan.

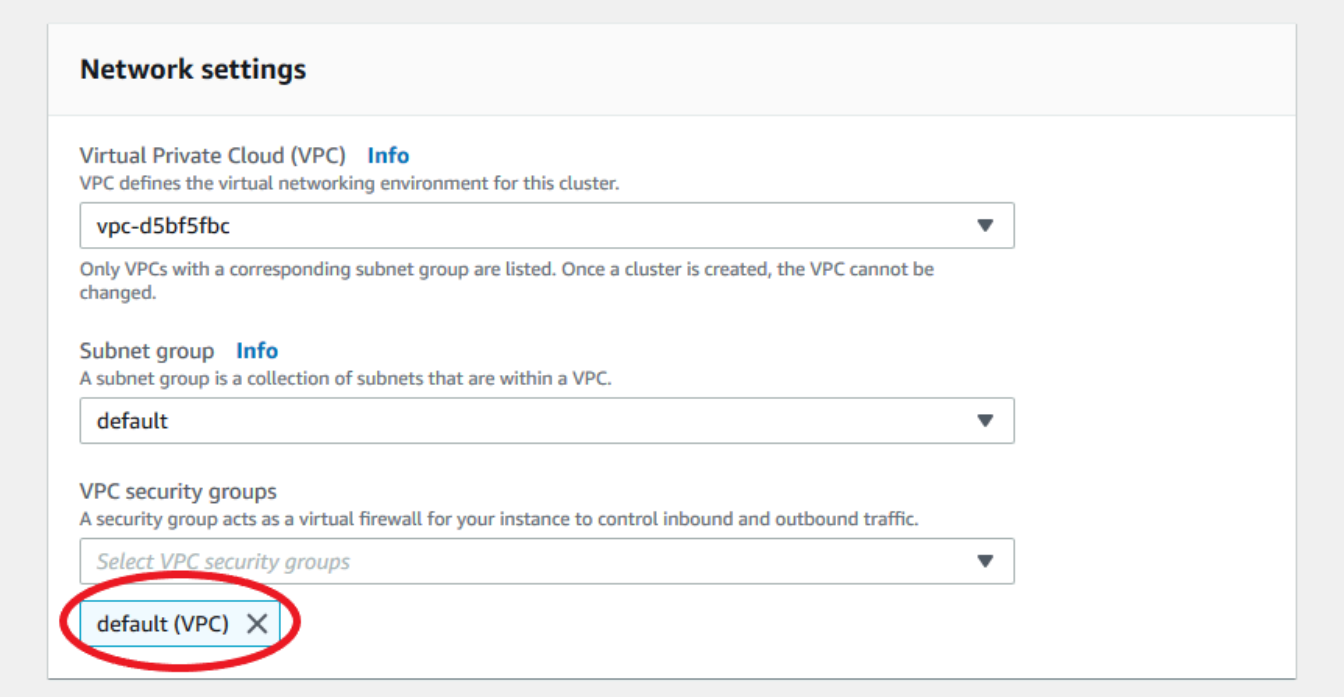


7. Di bagian Pengaturan Jaringan:

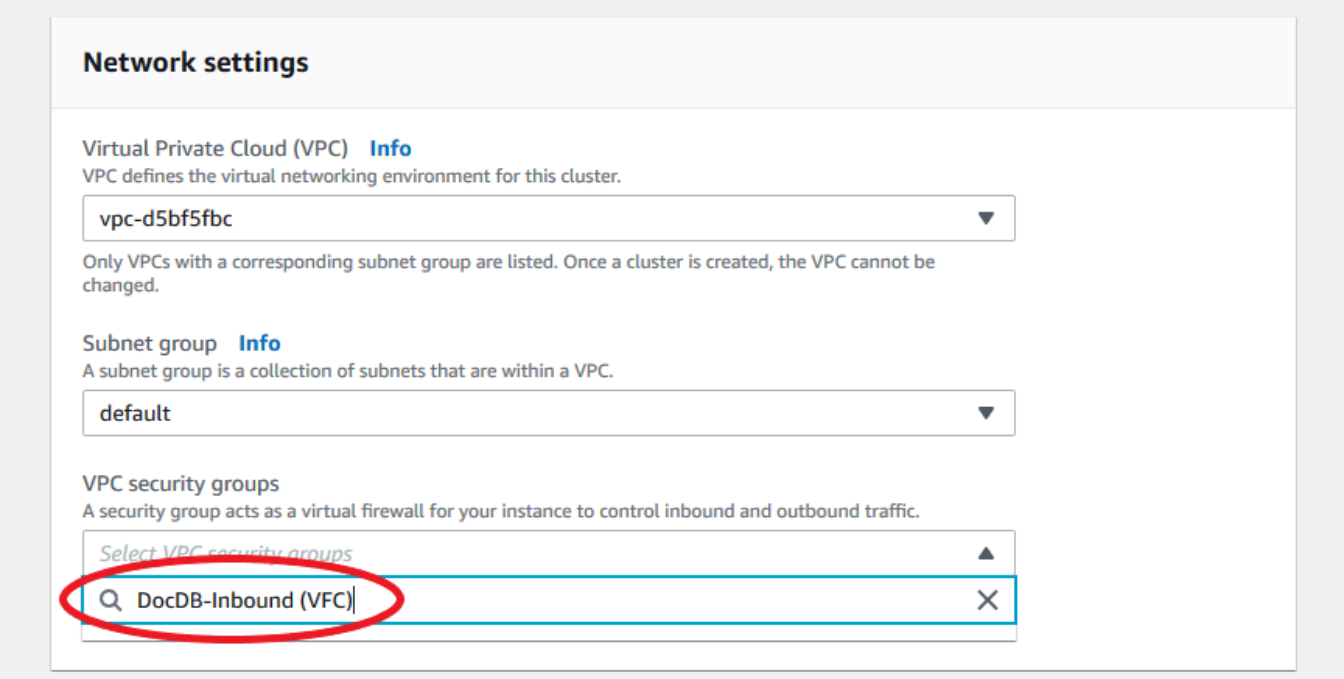
   - Simpan opsi default untuk Virtual Private Cloud dan Grup Subnet.



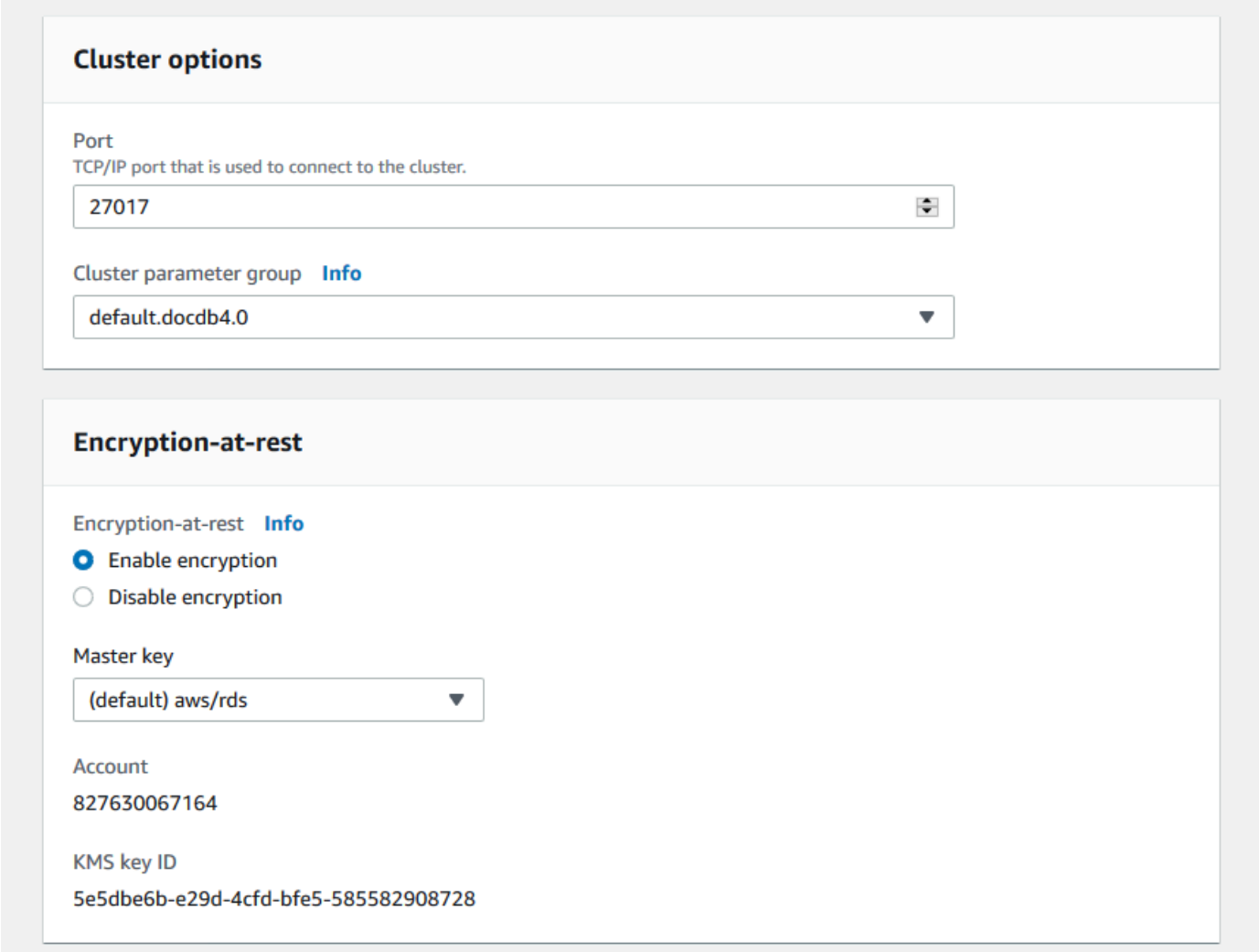
   - Untuk Grup keamanan VPC, VPC Default seharusnya sudah ditambahkan.



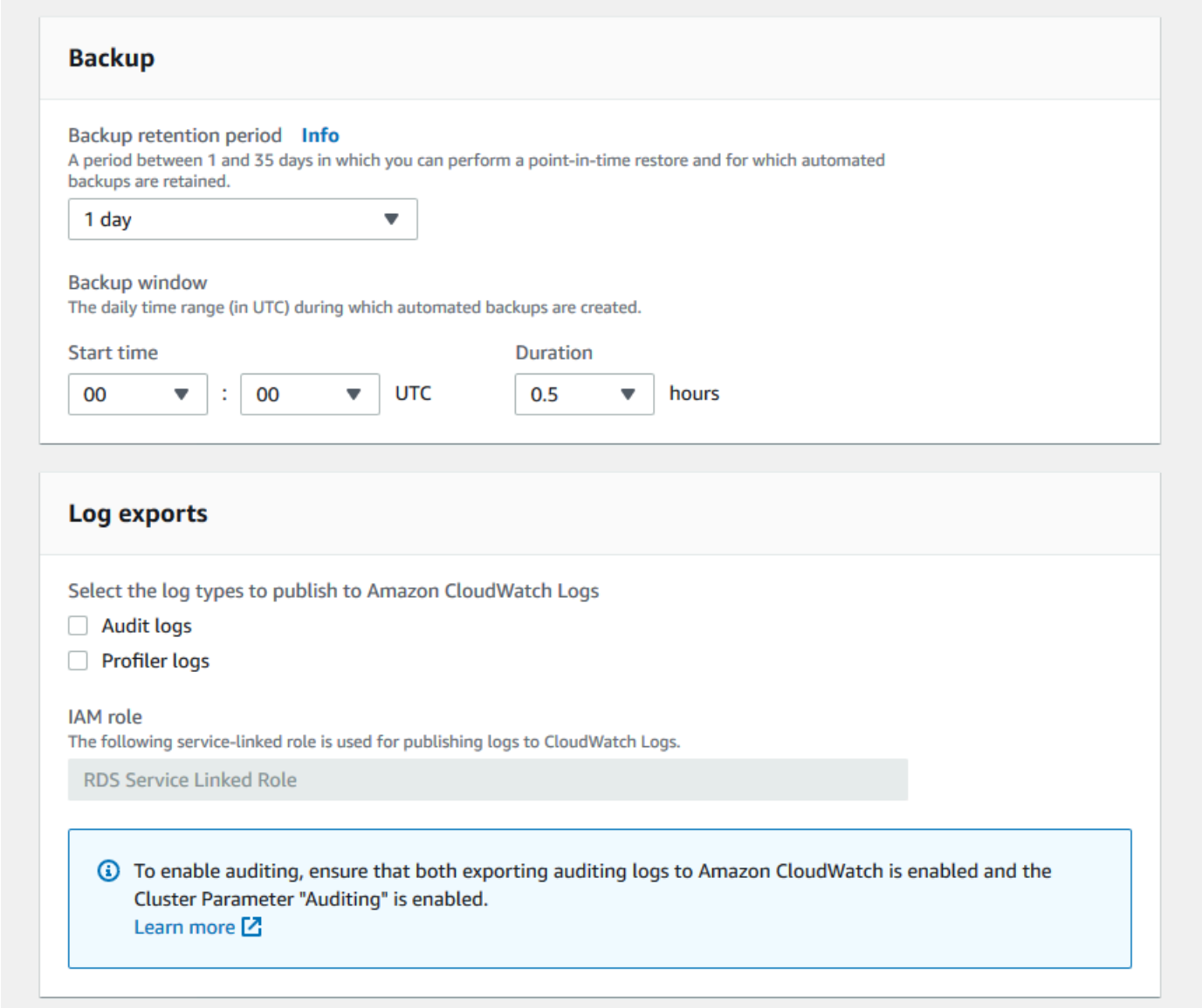
- Ketik DocDB ke dalam bidang Grup keamanan VPC dan pilih DocDB-Inbound (VPC).



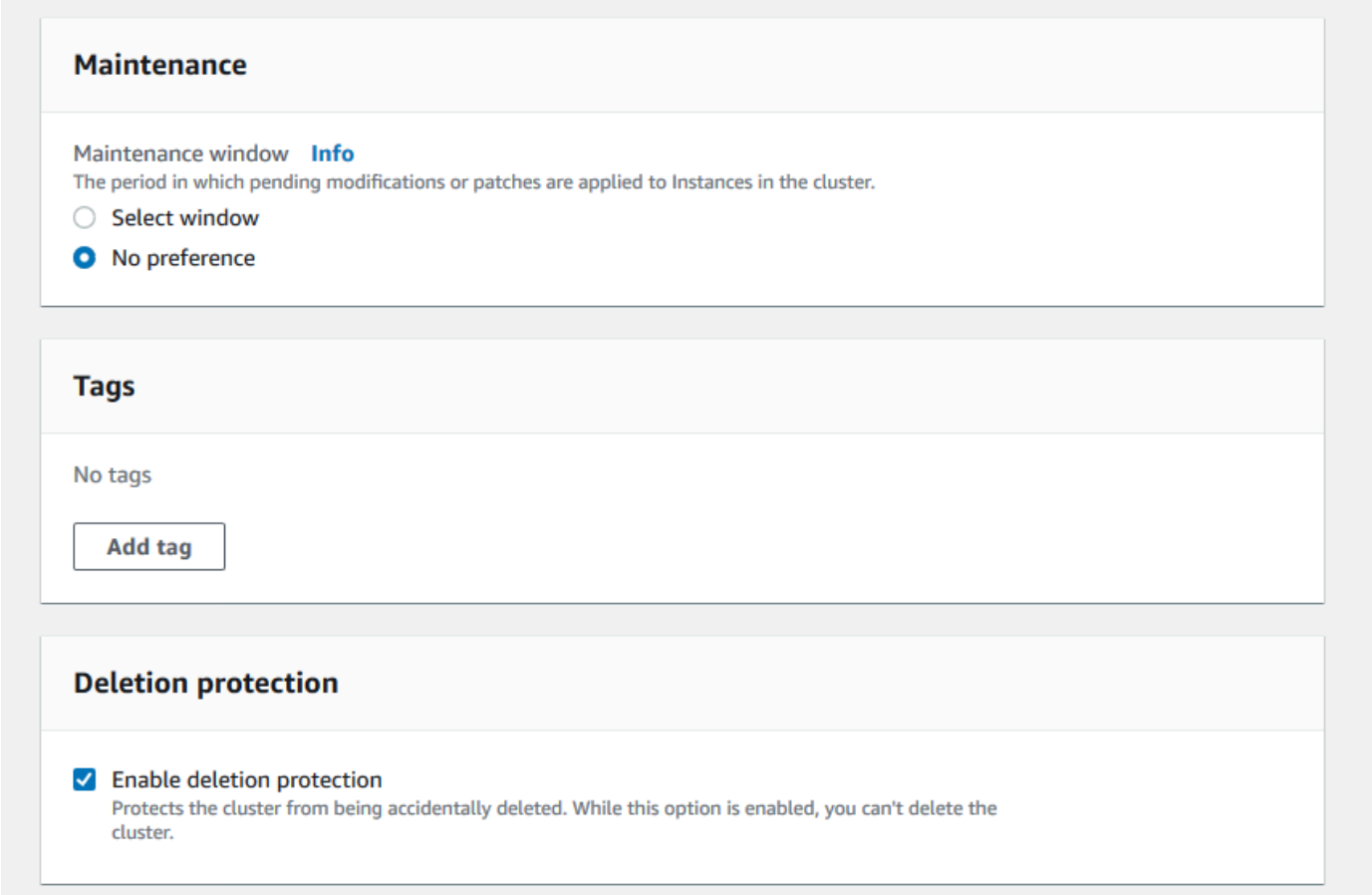
8. Untuk Cluster Options dan E ncryption-at-rest, biarkan pada pilihan default.



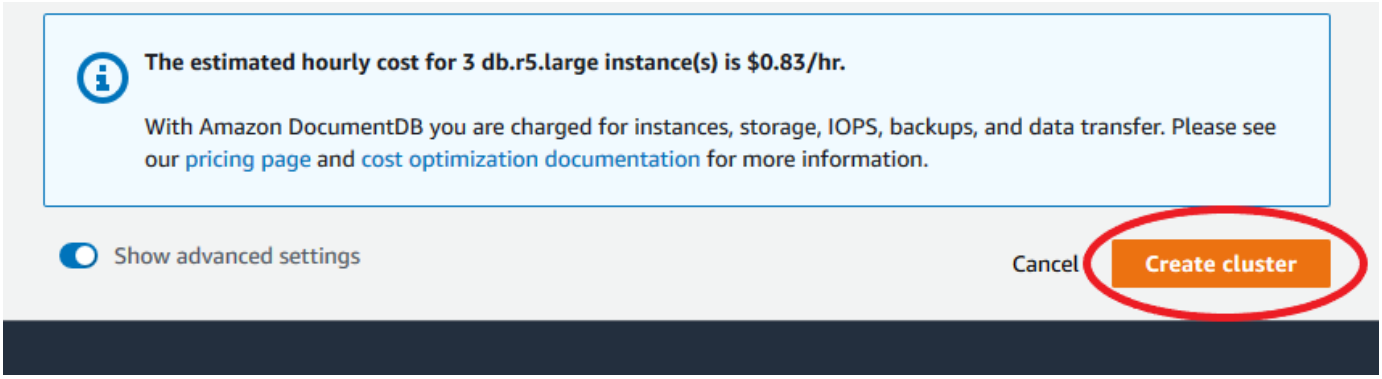
9. Untuk Cadangan dan Ekspor log, tinggalkan pada pilihan default.



10. Untuk Pemeliharaan, Tag dan Perlindungan Penghapusan, tinggalkan pilihan default.



11. Sekarang klik tombol yang mengatakan Buat.



Menggunakan AWS CLI

Untuk membuat klaster wilayah Amazon DocumentDB, panggil `create-db-cluster` AWS CLI. Perintah AWS CLI berikut menciptakan sebuah klaster Amazon DocumentDB bernama `global-cluster-id`. Untuk informasi selengkapnya tentang perlindungan penghapusan, lihat Menghapus cluster Amazon DocumentDB.

Juga, `--engine-version` adalah parameter opsional yang default ke versi mesin utama terbaru. Versi mesin utama saat ini adalah `4.0.0`. Ketika versi mesin utama baru dirilis, versi mesin default untuk `--engine-version` akan diperbarui untuk mencerminkan versi mesin utama yang terakhir. Akibatnya, untuk beban kerja produksi, dan terutama yang bergantung pada skrip, otomatisasi, atau templat AWS CloudFormation, kami menyarankan Anda secara eksplisit menentukan `--engine-version` ke versi utama yang dimaksud.

Jika `db-subnet-group-name` atau `vpc-security-group-id` tidak ditentukan, Amazon DocumentDB akan menggunakan grup subnet default dan grup keamanan Amazon VPC untuk wilayah tertentu.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "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"
    }
}
```


Dibutuhkan beberapa menit untuk membuat klaster. Anda dapat menggunakan AWS Management Console atau AWS CLI untuk memantau status klaster Anda. Untuk informasi selengkapnya, lihat Memantau Status klaster Amazon DocumentDB.

Important

Saat Anda menggunakan AWS CLI untuk membuat klaster regional Amazon DocumentDB, tidak ada instans yang dibuat. Akibatnya, Anda harus secara eksplisit membuat instans primer dan semua instans replika yang Anda butuhkan. Anda dapat menggunakan konsol tersebut atau AWS CLI untuk membuat instans. Untuk informasi selengkapnya, lihat Menambahkan instance Amazon DocumentDB ke cluster dan CreateDBCluster di Referensi API Amazon DocumentDB.

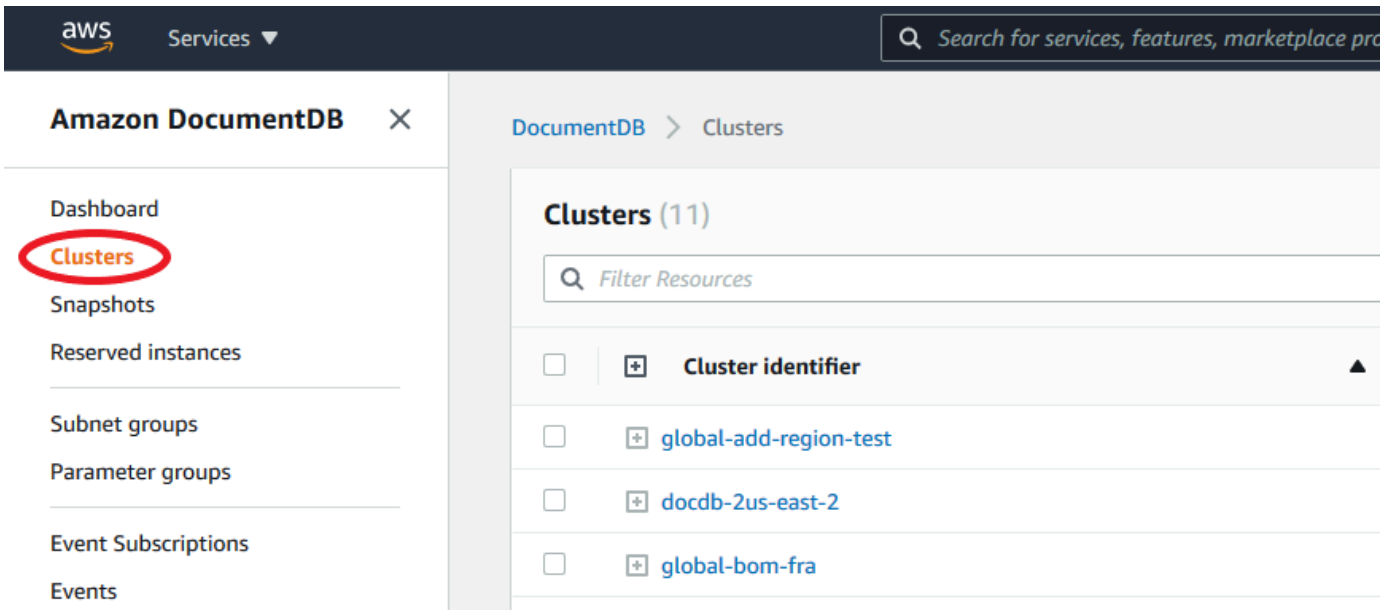
Setelah klaster wilayah Anda tersedia, Anda dapat menambahkan klaster sekunder di wilayah lain dengan petunjuk berikut: Menambahkan sebuah Wilayah AWS untuk klaster global Amazon DocumentDB. Saat Anda menambahkan wilayah, klaster wilayah Anda menjadi klaster primer Anda, dan Anda memiliki klaster sekunder baru di wilayah yang Anda pilih.

Menambahkan sebuah Wilayah AWS untuk klaster global Amazon DocumentDB

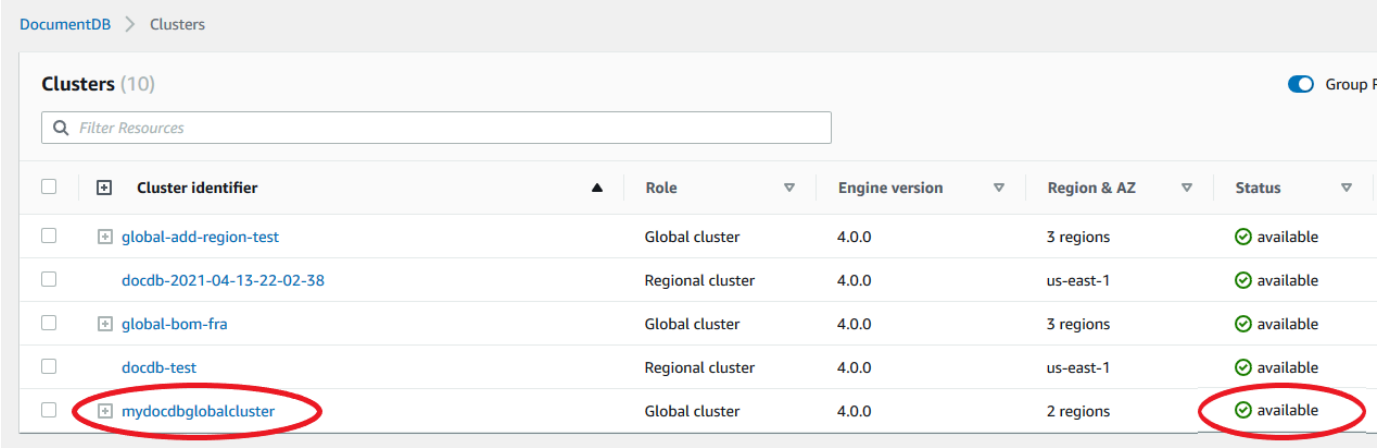
Klaster global membutuhkan setidaknya satu klaster sekunder di wilayah yang berbeda dari klaster primer, dan Anda dapat menambahkan hingga lima klaster sekunder. Perhatikan bahwa untuk setiap klaster sekunder yang Anda tambahkan, Anda harus mengurangi jumlah replika yang diizinkan di klaster primer sebanyak satu. Misalnya, jika klaster global Anda memiliki lima wilayah sekunder, klaster primer Anda hanya dapat memiliki sepuluh (bukan lima belas) replika. Untuk informasi selengkapnya, lihat Persyaratan konfigurasi klaster global Amazon DocumentDB.

Menggunakan AWS Management Console

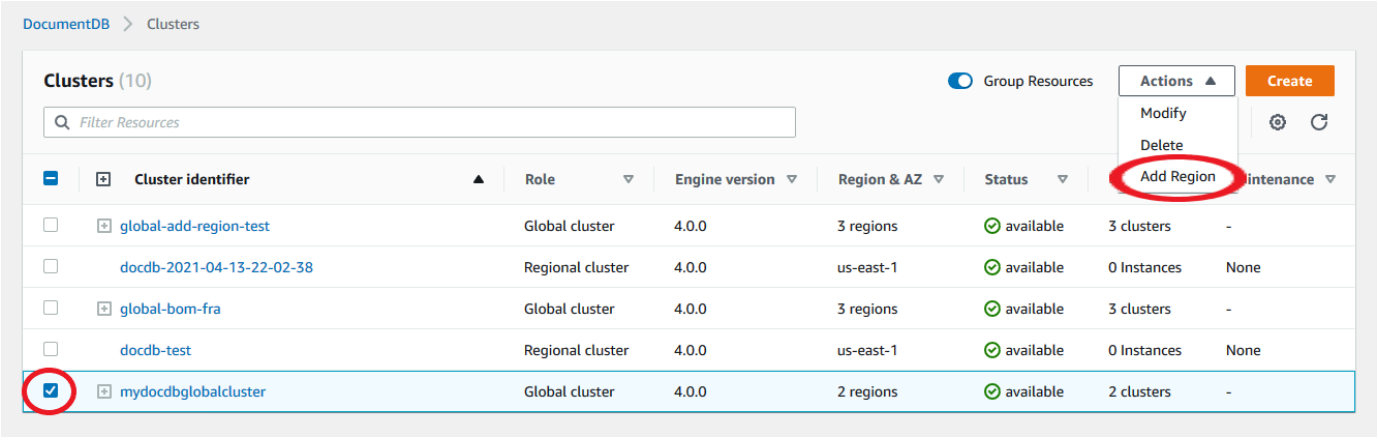
1. Masuk ke AWS Management Console dan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Klaster.



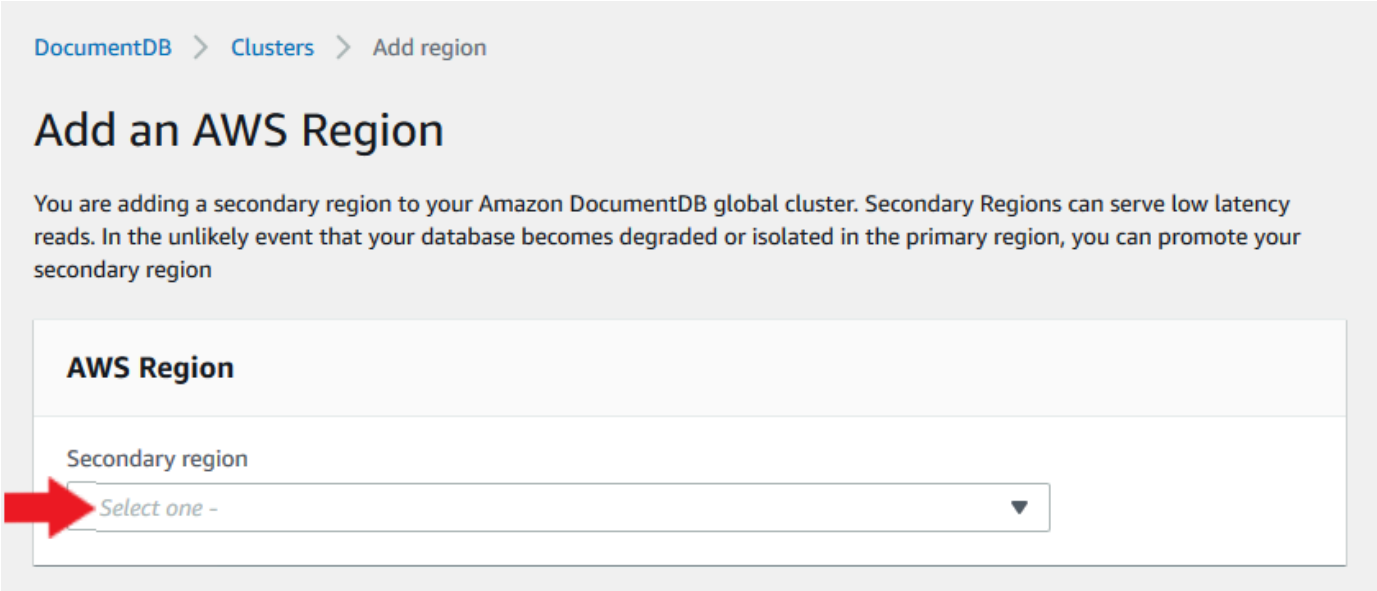
3. Pilih klaster yang ingin Anda tambahkan klaster sekundernya. Pastikan bahwa klaster adalah `Available`.



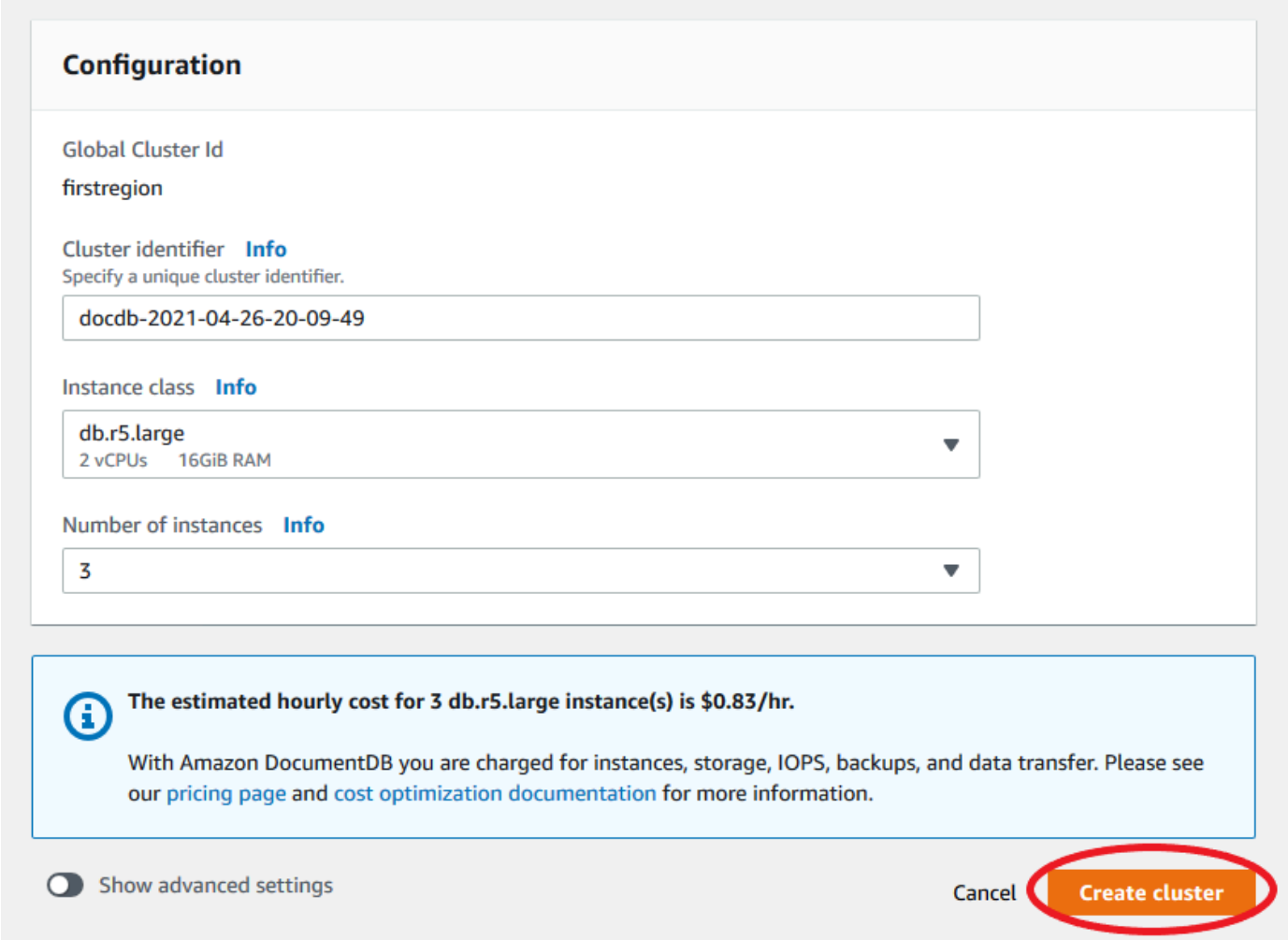
4. Pilih menu drop down untuk Tindakan dan lalu pilih Tambah wilayah.



5. Pada halaman Tambah wilayah, pilih wilayah sekunder. Perhatikan bahwa Anda tidak dapat memilih wilayah yang sudah memiliki klaster sekunder untuk klaster global yang sama. Selain itu, wilayah tersebut tidak boleh sama dengan klaster primer. Jika ini adalah wilayah pertama yang Anda tambahkan, Anda juga harus menentukan pengidentifikasi klaster global pilihan Anda.



6. Lengkapi bidang yang tersisa untuk klaster sekunder di wilayah baru, lalu pilih Buat Klaster. Setelah Anda selesai menambahkan wilayah, Anda dapat melihatnya di daftar Klaster di AWS Management Console.



Menggunakan AWS CLI

- Gunakan perintah CLI `create-db-cluster` dengan nama (`--global-cluster-identifier`) dari klaster global Anda. Untuk Parameter lainnya, lakukan hal berikut:
  - Untuk `--region`, pilih Wilayah AWS yang berbeda dari wilayah primer Anda.
  - Pilih nilai spesifik untuk parameter `--engine` dan `--engine-version`.
  - Untuk klaster terenkripsi, tentukan Wilayah AWS primer Anda sebagai `--source-region` untuk enkripsi.

Contoh berikut membuat klaster Amazon DocumentDB baru dan melampirkannya ke klaster global sebagai klaster sekunder baca-saja. Pada langkah terakhir, instans ditambahkan ke klaster baru.

Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Menggunakan snapshot untuk klaster global Amazon DocumentDB Anda

Anda dapat memulihkan snapshot klaster Amazon DocumentDB untuk digunakan sebagai titik awal klaster global Anda. Untuk melakukan ini, Anda harus memulihkan snapshot dan membuat klaster baru. Ini akan berfungsi sebagai klaster primer klaster global Anda. Anda kemudian dapat

menambahkan wilayah lain ke klaster yang dipulihkan, sehingga mengubahnya menjadi klaster global.

Mengelola klaster global Amazon DocumentDB

Anda melakukan sebagian besar operasi manajemen pada klaster individu yang membentuk klaster global. Saat Anda memilih Sumber daya terkait grup pada halaman Klaster di konsol tersebut, Anda melihat klaster primer dan klaster sekunder dikelompokkan di bawah klaster global terkait.

Tab Konfigurasi untuk klaster global menunjukkan Wilayah AWS tempat cluster berjalan, versi, dan pengidentifikasi klaster global.

Topik

- Memodifikasi klaster global Amazon DocumentDB
- Memodifikasi parameter klaster global Amazon DocumentDB
- Menghapus klaster dari klaster global Amazon DocumentDB
- Menghapus klaster dari klaster global Amazon DocumentDB
- Membuat klaster Amazon DocumentDB tanpa kepala di wilayah sekunder

Memodifikasi klaster global Amazon DocumentDB

Halaman Cluster dalam AWS Management Console daftar semua cluster global Anda, menampilkan cluster primer dan cluster sekunder untuk masing-masing cluster. Klaster global memiliki penganturan konfigurasi sendiri. Secara khusus, ia memiliki wilayah yang terkait dengan cluster primer dan sekundernya.

Saat Anda membuat perubahan pada klaster global, Anda memiliki kesempatan untuk membatalkan perubahan.

Saat Anda memilih Lanjutkan, Anda mengonfirmasi perubahan.

Memodifikasi parameter klaster global Amazon DocumentDB

Anda dapat mengonfigurasi grup parameter klaster secara independen untuk setiap klaster dalam klaster global. Sebagian besar parameter berfungsi sama seperti untuk jenis klaster Amazon DocumentDB lainnya. Kami menyarankan Anda menjaga pengaturan yang konsisten di antara semua klaster dalam basis data global. Tindakan tersebut membantu mencegah perubahan perilaku yang tidak terduga jika Anda menaikkan klaster sekunder menjadi klaster primer.

Misalnya, gunakan pengaturan zona waktu dan rangkaian karakter yang sama untuk mencegah perilaku yang tidak konsisten jika klaster yang berbeda mengambil alih peran sebagai klaster primer.

Menghapus klaster dari klaster global Amazon DocumentDB

Ada beberapa situasi ketika Anda mungkin ingin menghapus klaster dari klaster global Anda. Misalnya, Anda mungkin ingin menghapus klaster dari klaster global jika klaster primer menjadi terdegradasi atau terisolasi. Ini kemudian menjadi klaster yang disediakan mandiri yang dapat digunakan untuk membuat klaster global baru. Untuk mempelajari lebih lanjut, lihat Memulihkan klaster global secara manual dari pemadaman yang tidak direncanakan.

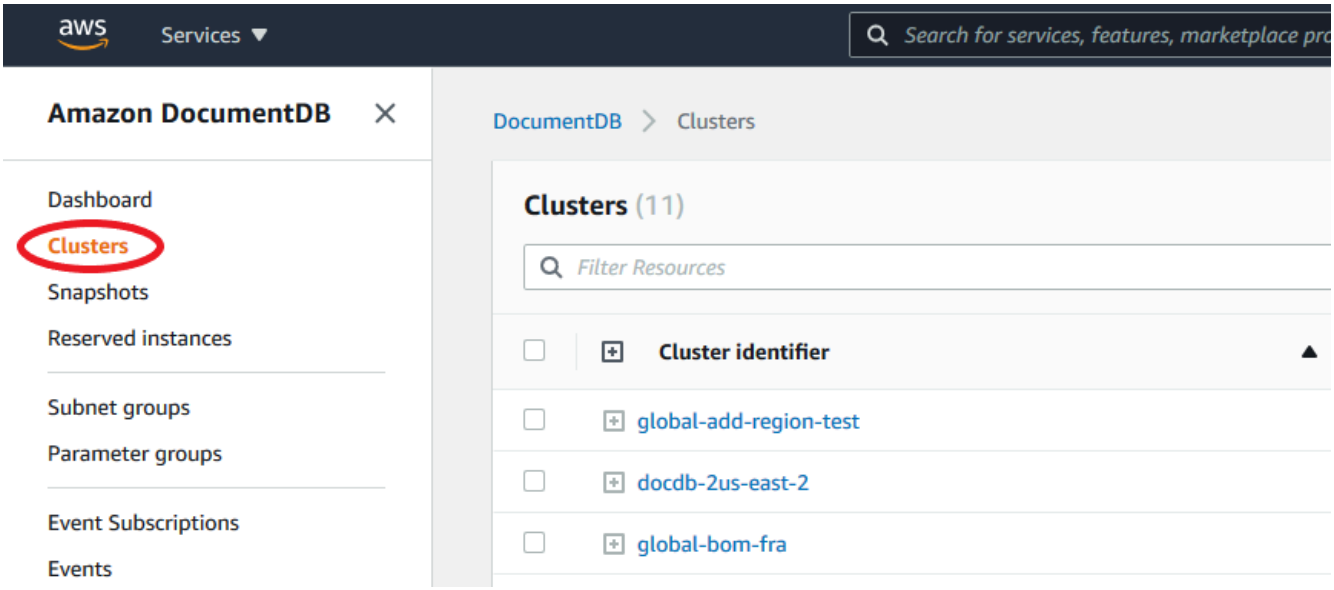
Anda juga mungkin ingin menghapus klaster karena Anda ingin menghapus klaster global yang tidak lagi Anda perlukan. Anda tidak dapat menghapus klaster global sampai setelah Anda melepaskan semua kluster terkait, meninggalkan yang primer untuk terakhir. Untuk informasi selengkapnya, lihat Menghapus klaster global Amazon DocumentDB.

> **Note**
>
> Ketika sebuah cluster terlepas dari cluster global, itu tidak lagi disinkronkan dengan yang utama. Ini menjadi klaster yang disediakan mandiri dengan kemampuan baca/tulis penuh. Selain itu, tidak lagi terlihat di konsol Amazon DocumentDB. Itu hanya terlihat ketika Anda memilih wilayah di konsol tempat cluster berada.

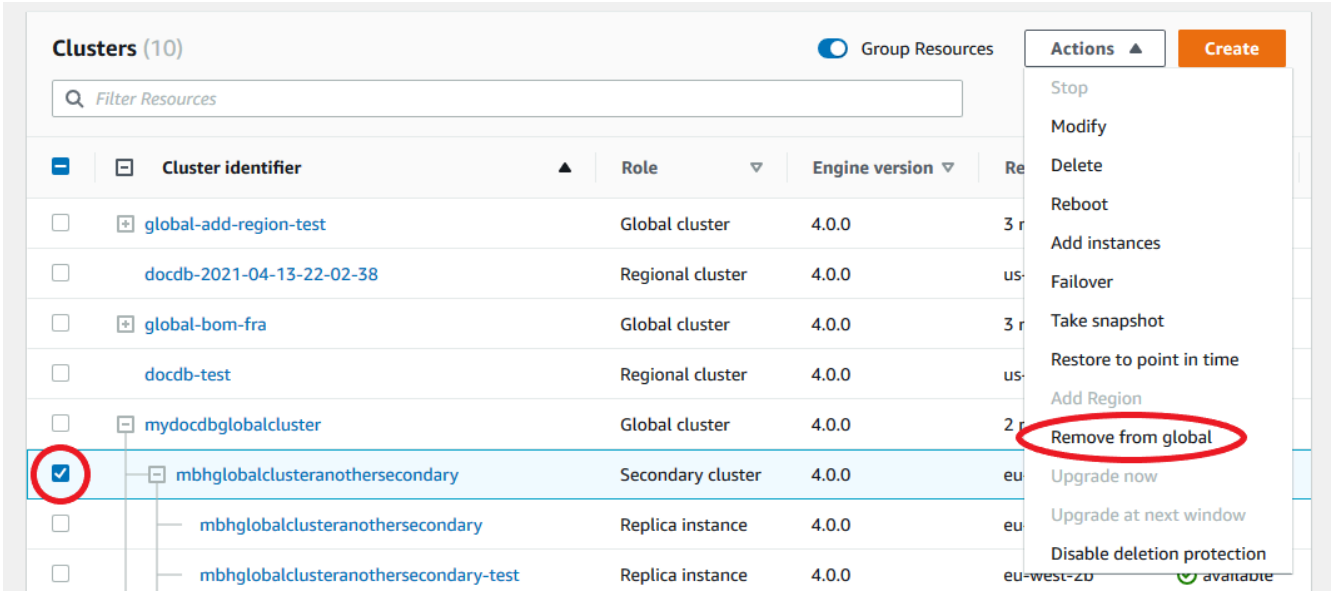
Anda dapat menghapus cluster dari cluster global Anda menggunakan AWS Management Console, the AWS CLI, atau RDS API.

Using the AWS Management Console

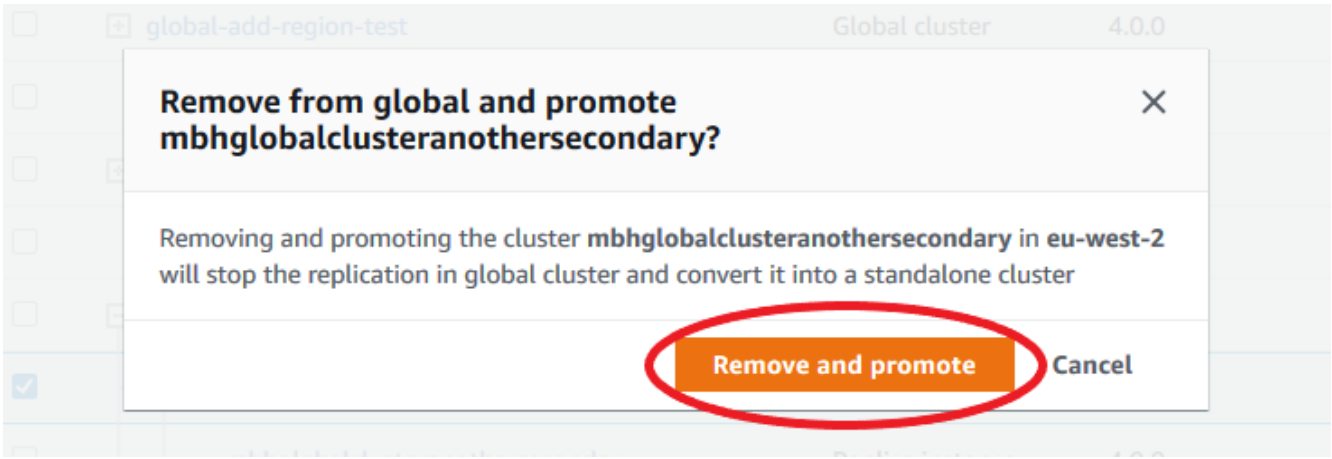
1. Masuk ke AWS Management Console dan navigasikan ke konsol Amazon DocumentDB.
2. Pilih Klaster di navigasi sisi kiri.



3. Perluas klaster global sehingga Anda dapat melihat semua klaster sekunder. Pilih klaster sekunder yang ingin Anda hapus. Pilih Tindakan, dan di menu yang turun, pilih Hapus dari Global.



4. Prompt akan muncul, meminta Anda untuk mengonfirmasi bahwa Anda ingin melepaskan sekunder dari klaster global. Pilih Hapus dan promosikan untuk menghapus klaster dari klaster global.



Sekarang klaster itu tidak lagi berfungsi sebagai sekunder dan tidak lagi disinkronkan dengan klaster primer. Ini adalah klaster mandiri dengan kemampuan baca/tulis penuh.

Setelah Anda menghapus atau menghapus semua klaster sekunder, maka Anda dapat menghapus klaster primer dengan cara yang sama. Anda tidak dapat melepaskan atau menghapus klaster primer dari klaster global sampai setelah Anda menghapus semua klaster sekunder. Klaster global mungkin tetap berada di daftar Klaster, dengan nol region dan AZ. Anda dapat menghapus jika tidak ingin lagi menggunakan klaster global ini.

Using the AWS CLI

Untuk menghapus klaster dari klaster global, jalankan perintah `remove-from-global-cluster` CLI dengan parameter berikut:

- `--global-cluster-identifier` — Nama (pengenal) klaster global Anda.
- `--db-cluster-identifier` — Nama setiap klaster untuk menghapus dari klaster global.

Contoh berikut pertama-tama menghapus klaster sekunder dan kemudian klaster primer dari klaster global.

Untuk Linux, macOS, atau 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
```

Ulangi perintah `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` untuk setiap wilayah sekunder di klaster global Anda.

Untuk 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
```

Ulangi perintah `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` untuk setiap wilayah sekunder di klaster global Anda.

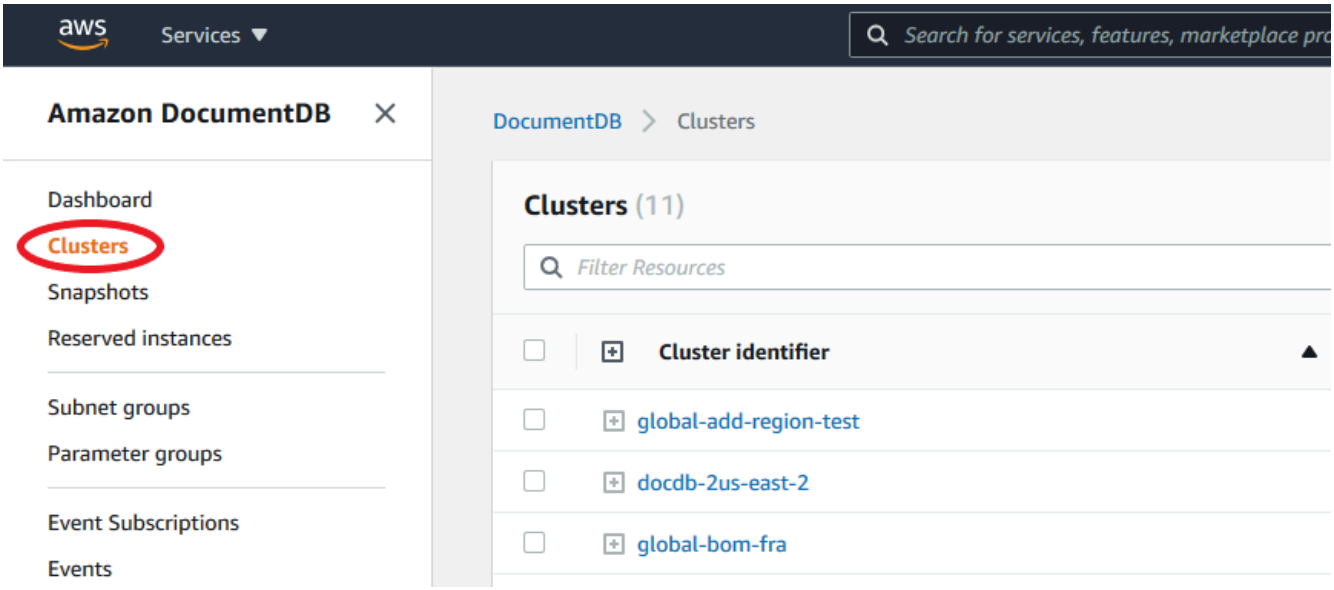
Menghapus klaster dari klaster global Amazon DocumentDB

Untuk menghapus klaster global, lakukan hal berikut:

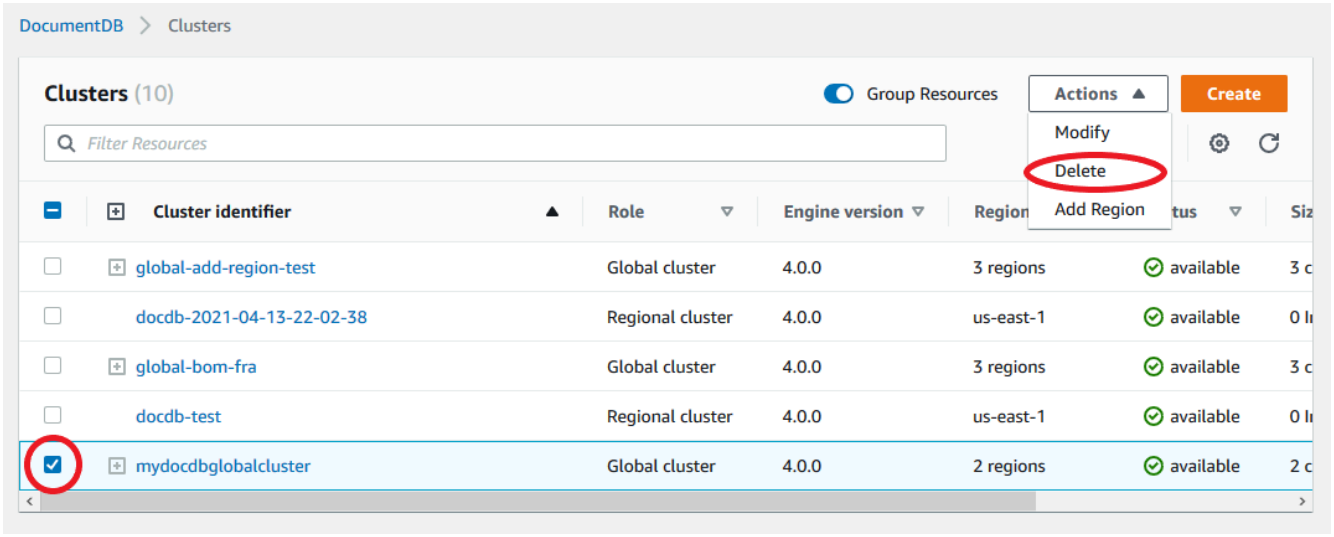
- Hapus semua klaster sekunder dari klaster global. Setiap klaster menjadi klaster yang berdiri sendiri. Lihat bagian sebelumnya, Menghapus Klaster Global.
- Dari setiap klaster mandiri, hapus semua replika.
- Hapus klaster primer dari klaster global. Ini menjadi klaster yang berdiri sendiri.
- Dari klaster primer, hapus semua replika terlebih dahulu, lalu hapus instans primer. Menghapus instans primer dari klaster mandiri yang baru juga biasanya menghapus klaster dan klaster global.

Using the AWS Management Console

1. Masuk ke AWS Management Console dan navigasikan ke konsol Amazon DocumentDB.
2. Pilih Klaster dan temukan klaster global yang ingin Anda hapus.



3. Dengan klaster global Anda dipilih, pilih Hapus dari menu Tindakan.



Konfirmasikan bahwa semua klaster dihapus dari klaster global. Klaster global harus menunjukkan nol region dan AZ dan ukuran nol klaster. Jika klaster global berisi klaster apa pun, Anda belum dapat menghapusnya. Pertama-tama Anda harus mengikuti petunjuk di langkah sebelumnya, Menghapus Klaster Global.

Using the AWS CLI

Untuk menghapus cluster global, jalankan perintah `delete-global-cluster` CLI dengan nama Wilayah AWS dan pengidentifikasi cluster global, seperti yang ditunjukkan pada contoh berikut.

Untuk Linux, macOS, atau Unix:

```
aws docdb --region primary_region delete-global-cluster \
   --global-cluster-identifier global_cluster_id
```

Untuk Windows:

```
aws docdb --region primary_region delete-global-cluster ^
   --global-cluster-identifier global_cluster_id
```

Membuat klaster Amazon DocumentDB tanpa kepala di wilayah sekunder

Meskipun klaster global Amazon DocumentDB memerlukan setidaknya satu klaster sekunder yang Wilayah AWS berbeda dari yang utama, Anda dapat menggunakan konfigurasi tanpa kepala untuk klaster sekunder. Cluster Amazon DocumentDB sekunder tanpa kepala adalah cluster tanpa instance. Jenis konfigurasi ini dapat menurunkan biaya untuk cluster global. Dalam cluster Amazon DocumentDB, komputasi dan penyimpanan dipisahkan. Tanpa instance, Anda tidak dikenakan biaya untuk komputasi, hanya untuk penyimpanan. Jika diatur dengan benar, volume penyimpanan sekunder tanpa kepala tetap sinkron dengan cluster utama.

Anda menambahkan klaster sekunder seperti biasa saat membuat klaster global Amazon DocumentDB. Namun, setelah cluster primer memulai replikasi ke sekunder, Anda menghapus instance hanya-baca dari cluster sekunder. Cluster sekunder ini sekarang dianggap “tanpa kepala” karena tidak lagi memiliki Instance. Namun, volume penyimpanan tetap sinkron dengan cluster Amazon DocumentDB utama.

> **Important**
>
> Kami hanya merekomendasikan klaster tanpa kepala untuk pelanggan yang dapat mentolerir kegagalan di seluruh wilayah selama 15+ menit. Ini karena pemulihan dari kegagalan di seluruh wilayah dengan cluster sekunder tanpa kepala akan mengharuskan pengguna untuk membuat instance baru setelah gagal. Sebuah instance baru dapat memakan waktu ~ 10-15 menit untuk tersedia.

Cara Menambahkan Cluster Sekunder Tanpa Kepala ke Cluster Global Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon DocumentDB.

2. Pilih Klaster di navigasi sisi kiri.
3. Pilih cluster global yang membutuhkan cluster sekunder. Pastikan bahwa cluster utama adalah`Available`.
4. Untuk Tindakan, pilih Tambahkan region.
5. Pada halaman Tambah wilayah, pilih wilayah sekunder.

> **Note**
>
> Anda tidak dapat memilih wilayah yang sudah memiliki cluster sekunder untuk cluster global yang sama. Selain itu, wilayah tersebut tidak boleh sama dengan klaster primer.

6. Lengkapi bidang yang tersisa untuk cluster sekunder di wilayah baru. Ini adalah opsi konfigurasi yang sama seperti untuk instance cluster apa pun.
7. Tambahkan wilayah. Setelah Anda selesai menambahkan wilayah ke cluster global Anda, Anda akan melihatnya dalam daftar `Clusters` di AWS Management Console.
8. Periksa status cluster sekunder dan instance pembacanya sebelum melanjutkan, dengan menggunakan AWS Management Console atau AWS CLI. Berikut adalah contoh perintah jika Anda menggunakan AWS CLI:

```
$ aws docdb describe-db-clusters --db-cluster-identifier secondary-cluster-id --
query '*[].[Status]' --output text
```

   Diperlukan beberapa menit agar status cluster sekunder yang baru ditambahkan berubah dari pembuatan menjadi tersedia. Ketika cluster tersedia, Anda dapat menghapus instance pembaca.
9. Pilih instance pembaca di cluster sekunder, lalu pilih Hapus.
10. Setelah menghapus instance pembaca, cluster sekunder tetap menjadi bagian dari cluster global. Seharusnya tidak ada contoh yang terkait dengannya.

> **Note**
>
> Anda dapat menggunakan klaster Amazon DocumentDB sekunder tanpa kepala ini untuk memulihkan klaster global Amazon DocumentDB Anda secara manual dari pemadaman yang tidak direncanakan di wilayah utama jika terjadi pemadaman seperti itu.

Terhubung ke Klaster Global Amazon DocumentDB

Cara Anda terhubung ke klaster global bergantung pada apakah Anda perlu menulis ke klaster atau membaca dari klaster:

- Untuk permintaan atau kueri baca-saja, Anda menyambungkan ke reader endpoint untuk klaster di Wilayah AWS.
- Untuk menjalankan pernyataan bahasa manipulasi data (DML) atau bahasa definisi data (DDL), Anda terhubung ke titik akhir klaster untuk klaster primer. Endpoint ini mungkin berbeda Wilayah AWS dari aplikasi Anda.

Saat Anda melihat klaster global di konsol, Anda dapat melihat semua titik akhir tujuan umum yang terkait dengan semua klasternya.

Bagaimana Anda terhubung ke klaster global bergantung pada apakah Anda perlu menulis ke basis data atau membaca dari basis data. Untuk DDL, DML, dan operasi baca yang ingin Anda layani dari wilayah primer, Anda harus terhubung ke klaster primer Anda. Kami menyarankan Anda menyambungkan ke klaster primer menggunakan titik akhir klaster dalam mode set replika, dengan preferensi baca `secondaryPreferred=true`. Ini akan merutekan lalu lintas tulis ke instans penulis klaster primer Anda dan lalu lintas baca ke instans replika klaster primer Anda.

Untuk lintas wilayah, lalu lintas baca saja, Anda harus terhubung ke salah satu klaster sekunder Anda. Sebaiknya sambungkan ke klaster sekunder menggunakan titik akhir klaster dalam mode set replika. Karena semua instans adalah instans replika baca-saja, Anda tidak perlu menentukan preferensi baca. Untuk meminimalkan latensi, pilih reader endpoint mana saja yang ada di wilayah Anda atau wilayah terdekat dengan Anda.

Memantau Klaster Global Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) terintegrasi CloudWatch dengan sehingga Anda dapat mengumpulkan dan menganalisis metrik operasional untuk cluster Anda. Anda dapat memantau metrik ini menggunakan CloudWatch konsol, konsol Amazon DocumentDB, () AWS Command Line Interface ,AWS CLI atau API. CloudWatch

Untuk memantau klaster global, gunakan CloudWatch metrik berikut.

Metrik	Deskripsi
`GlobalClusterReplicatedWriteIO`	Jumlah rata-rata operasi I/O tulis yang ditagih direplikasi dari volume cluster di volume primer Wilayah AWS ke volume cluster di sekunder Wilayah AWS, dilaporkan pada interval 5 menit. Banyaknya `ReplicatedWriteIOs` yang direplikasi untuk setiap wilayah sekunder sama dengan jumlah `VolumeWriteIOPs` dalam wilayah yang dilakukan oleh wilayah primer.
`GlobalClusterDataTransferBytes`	Jumlah data yang ditransfer dari cluster primer Wilayah AWS ke cluster sekunder Wilayah AWS, diukur dalam byte.
`GlobalClusterReplicationLag`	Jumlah lag, dalam milidetik, saat mereplikasi peristiwa perubahan dari cluster primer Wilayah AWS ke cluster sekunder Wilayah AWS

Untuk informasi selengkapnya tentang cara melihat metrik ini, silakan lihat Melihat CloudWatch data.

Pemulihan Bencana dan Cluster Global Amazon DocumentDB

Dengan menggunakan klaster global, Anda dapat pulih dari bencana seperti kegagalan wilayah dengan cepat. Pemulihan dari bencana biasanya diukur dengan menggunakan nilai untuk RTO dan RPO.

- Tujuan waktu pemulihan (RTO) — Waktu yang dibutuhkan sistem untuk kembali ke kondisi kerja setelah bencana. Dengan kata lain, RTO mengukur waktu henti. Untuk klaster global, RTO bisa dalam hitungan menit.
- Tujuan titik pemulihan (RPO) — Jumlah data yang dapat hilang (diukur dalam waktu). Untuk klaster global, RPO biasanya diukur dalam hitungan detik.
- Untuk memulihkan dari pemadaman yang tidak direncanakan, Anda dapat melakukan failover lintas wilayah ke salah satu sekunder di klaster global Anda. Saat klaster global Anda memiliki beberapa wilayah sekunder, pastikan Anda melepaskan semua wilayah sekunder jika Wilayah AWS primer mengalami pemadaman. Kemudian, Anda mempromosikan salah satu wilayah

sekunder tersebut menjadi Wilayah AWS primer yang baru. Terakhir, Anda membuat klaster baru di setiap wilayah sekunder lainnya dan melampirkan klaster tersebut ke klaster global Anda.

- Saat Anda mempromosikan klaster sekunder menjadi klaster primer Anda juga perlu memperbarui titik akhir yang digunakan aplikasi Anda untuk terhubung ke klaster global. Untuk mendapatkan titik akhir penulis baru dari klaster yang baru dipromosikan, Anda dapat mengonversi reader endpoint sebelumnya dengan menghapus `-ro` dari string titik akhir. Misalnya, jika reader endpoint sebelumnya adalah `global-16rr-test-cluster-1.cluster-ro-12345678901.us-west-2.docdb.amazonaws.com`, kemudian endpoint baru yang dipromosikan adalah `global-16rr-test-cluster-1.cluster-cps2igpwyrwa.us-west-2.rds.amazonaws.com`.

Failover untuk Cluster Global Amazon DocumentDB

Jika seluruh cluster dalam satu Wilayah AWS menjadi tidak tersedia, Anda dapat mempromosikan cluster lain di cluster global untuk memiliki kemampuan baca/tulis.

Anda dapat mengaktifkan mekanisme failover secara manual jika klaster di Wilayah AWS yang berbeda adalah pilihan yang lebih baik untuk menjadi klaster primer. Misalnya, Anda dapat meningkatkan kapasitas salah satu klaster sekunder dan kemudian mempromosikannya menjadi klaster primer. Atau keseimbangan aktivitas di antara Wilayah AWS mungkin berubah, sehingga mengalihkan cluster utama ke cluster lain Wilayah AWS mungkin memberikan latensi yang lebih rendah untuk operasi penulisan.

Prosedur berikut menguraikan apa yang harus dilakukan untuk mempromosikan salah satu klaster sekunder di klaster global DocumentDB.

Untuk mempromosikan cluster sekunder:

1. Berhenti mengeluarkan pernyataan DHTML dan operasi penulisan lainnya ke cluster utama di Wilayah AWS with the outage.
2. Identifikasi cluster dari sekunder Wilayah AWS untuk digunakan sebagai cluster primer baru. Jika Anda memiliki dua (atau lebih) sekunder Wilayah AWS di cluster global Anda, pilih cluster sekunder yang memiliki waktu jeda paling sedikit.
3. Lepaskan klaster sekunder yang Anda pilih dari klaster global.

   Menghapus klaster sekunder dari klaster global segera menghentikan replikasi dari primer ke sekunder ini dan mempromosikannya ke klaster klaster yang disediakan mandiri dengan kemampuan baca/tulis penuh. Klaster sekunder lainnya yang terkait dengan klaster primer di

wilayah dengan pemadaman masih tersedia dan dapat menerima panggilan dari aplikasi Anda. Klaster tersebut juga mengonsumsi sumber daya. Karena Anda membuat ulang klaster global, untuk menghindari otak-terbelah dan masalah lainnya, hapus klaster sekunder lainnya sebelum membuat klaster global baru dalam langkah-langkah berikut.

Untuk langkah-langkah rinci untuk melepaskan, lihat Menghapus klaster dari klaster global Amazon DocumentDB.

4. Konfigurasi ulang aplikasi Anda untuk mengirim semua operasi tulis ke klaster yang sekarang berdiri sendiri ini menggunakan titik akhir barunya. Jika Anda menerima nama yang diberikan saat membuat klaster global, Anda dapat mengubah titik akhir dengan menghapus -ro dari string titik akhir klaster di aplikasi Anda.

   Misalnya, titik akhir klaster sekunder `my-global.cluster-ro-aaaaaabbbbbb.us-west-1.docdb.amazonaws.com` menjadi `my-global.cluster-aaaaaabbbbbb.us-west-1.docdb.amazonaws.com` ketika klaster tersebut terlepas dari klaster global.

   Klaster ini menjadi klaster primer dari klaster global baru saat Anda mulai menambahkan Wilayah ke dalamnya, pada langkah berikutnya.
5. Tambahkan Wilayah AWS ke cluster. Saat Anda melakukannya, proses replikasi dari klaster primer ke klaster sekunder akan dimulai.
6. Tambahkan lebih banyak Wilayah AWS sesuai kebutuhan untuk membuat ulang topologi yang diperlukan untuk mendukung aplikasi Anda. Pastikan bahwa penulisan aplikasi dikirim ke klaster yang benar sebelum, selama, dan setelah membuat perubahan seperti ini, untuk menghindari inkonsistensi data di antara klaster dalam klaster global (masalah split-brain).
7. Saat pemadaman teratasi dan Anda siap menugaskan Wilayah AWS asli sebagai klaster primer lagi, lakukan langkah yang sama secara terbalik.
8. Hapus salah satu klaster sekunder dari klaster global. Ini akan memungkinkannya untuk melayani lalu lintas baca/tulis.
9. Alihkan ulang semua lalu lintas tulis ke klaster primer di Wilayah AWS asli.
10. Tambahkan Wilayah AWS untuk mengatur satu atau lebih cluster sekunder yang sama Wilayah AWS seperti sebelumnya.

Cluster global Amazon DocumentDB dapat dikelola AWS menggunakan SDK, memungkinkan Anda membuat solusi untuk mengotomatiskan proses failover klaster global untuk kasus penggunaan Disaster Recovery dan Business Continuity Planning. Salah satu solusi tersebut tersedia untuk pelanggan kami di bawah lisensi Apache 2.0 dan dapat diakses dari repositori alat kami di sini. Solusi

ini memanfaatkan Amazon Route53 untuk manajemen titik akhir dan menyediakan fungsi AWS Lambda yang dapat dipicu berdasarkan peristiwa yang sesuai.

Mengelola cluster Amazon DocumentDB

Untuk mengelola klaster Amazon DocumentDB, Anda harus memiliki kebijakan IAM dengan izin bidang kontrol Amazon DocumentDB yang sesuai. Izin ini memungkinkan Anda untuk membuat, memodifikasi, dan menghapus klaster dan instans. Kebijakan `AmazonDocDBFullAccess` menyediakan semua izin yang diperlukan untuk mengelola klaster Amazon DocumentDB.

Topik berikut menunjukkan bagaimana caranya melakukan berbagai tugas ketika bekerja dengan klaster Amazon DocumentDB, termasuk menciptakan, menghapus, memodifikasi, menghubungkan ke, dan melihat klaster.

Topik

- Memahami cluster
- Pengaturan Amazon DocumentDB Cluster
- Konfigurasi penyimpanan klaster Amazon DocumentDB
- Menentukan status klaster
- Siklus hidup klaster Amazon DocumentDB
- Menskalakan cluster Amazon DocumentDB
- Mengkloning volume untuk cluster Amazon DocumentDB
- Memahami toleransi kesalahan klaster Amazon DocumentDB

Memahami cluster

Amazon DocumentDB memisahkan perhitungan dan penyimpanan, dan replikasi data offload dan backup ke volume klaster. Volume klaster menyediakan lapisan penyimpanan tahan lama, andal, dan sangat tersedia yang mereplikasi data enam cara di tiga Zona Ketersediaan. Replika memungkinkan ketersediaan data yang lebih tinggi dan membaca penskalaan. Setiap klaster dapat diskalakan ke atas hingga 15 replika.

Kata benda	Deskripsi	Operasi API (Kata Kerja)
Klaster	Terdiri atas satu atau beberapa instans dan volume	`create-db-cluster`

Kata benda	Deskripsi	Operasi API (Kata Kerja)

	penyimpanan klaster yang mengelola data untuk instans tersebut.	`delete-db-cluster`
		`describe-db-clusters`
		`modify-db-cluster`

Instans	Membaca dan menulis data ke volume penyimpanan klaster dilakukan melalui instans. Dalam sebuah klaster tertentu, ada dua jenis instans: primer dan replika. Sebuah klaster selalu memiliki satu instans utama dan dapat memiliki 0–15 replika.	`create-db-instance`
		`delete-db-instance`
		`describe-db-instances`
		`modify-db-instance`
		`describe-orderable-db-instance-options`
		`reboot-db-instance`

Volume klaster	Volume penyimpanan basis data virtual yang mencakup tiga Availability Zone, dengan setiap Availability Zone memiliki dua salinan data klaster.	N/A
Instans utama	Mendukung operasi baca dan tulis, dan melakukan semua modifikasi data pada volume klaster. Setiap klaster memiliki satu instans utama.	N/A

Kata benda	Deskripsi	Operasi API (Kata Kerja)
Instans replika	Mendukung hanya operasi baca. Setiap klaster Amazon DocumentDB dapat memiliki hingga 15 instans replika di samping instans utama. Beberapa replika mendistri busikan beban kerja baca. Dengan menemukan replika di Availability Zone terpisah, Anda juga dapat meningkat kan ketersediaan basis data.	N/A
Titik akhir klaster	Titik akhir untuk klaster Amazon DocumentDB yang menghubungkan ke instans utama saat ini untuk klaster. Setiap klaster Amazon DocumentDB memiliki titik akhir klaster dan satu instans utama.	N/A
Reader endpoint	Titik akhir untuk klaster Amazon DocumentDB yang terhubung ke salah satu replika yang tersedia untuk klaster tersebut. Setiap klaster Amazon DocumentDB memiliki reader endpoint. Jika ada lebih dari satu replika, reader endpoint mengarahk an setiap permintaan koneksi ke salah satu replika Amazon DocumentDB.	N/A

Kata benda	Deskripsi	Operasi API (Kata Kerja)
Instans titik akhir	Titik akhir untuk instans dalam klaster Amazon DocumentDB yang terhubung ke instans tertentu. Setiap instans dalam sebuah klaster, apa pun jenis instansnya, memiliki titik akhir instans uniknya sendiri.	N/A

Pengaturan Amazon DocumentDB Cluster

Ketika Anda membuat atau memodifikasi klaster, penting untuk memahami parameter yang tetap dan yang dimodifikasi setelah klaster selesai dibuat. Tabel berikut mencantumkan semua pengaturan, atau parameter, yang khusus dibuat untuk klaster. Seperti yang ditentukan dalam tabel, beberapa dapat dimodifikasi, yang lain tidak.

> **Note**
>
> Pengaturan ini jangan sampai tertukar dengan grup parameter klaster Amazon DocumentDB dan parameternya. Untuk informasi selengkapnya tentang grup parameter klaster, lihat Mengelola grup parameter cluster Amazon DocumentDB.

Parameter	Dapat diubah	Catatan
`DBClusterIdentifier`	Ya	Aturan penamaan: • Panjangnya adalah [1—63] huruf, angka, atau tanda hubung. • Karakter pertama harus berupa huruf. • Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan. • Harus unik untuk semua cluster di Amazon Amazon RDS, Amazon Neptune, dan

Parameter	Dapat diubah	Catatan
		Amazon DocumentDB per, per Wilayah. Akun AWS
`Engine`	Tidak	Harus berupa `docdb`.
`BackupRetentionPeriod`	Ya	Harus antara [1-35] hari.
`DBClusterParameterGroupName`	Ya	Aturan penamaan: • Panjang adalah [1—255] karakter alfanumerik. • Karakter pertama harus berupa huruf. • Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.
`DBSubnetGroupName`	Tidak	Setelah klaster selesai dibuat, Anda tidak dapat mengubah subnet klaster.
`EngineVersion`	Tidak	Nilai bisa `5.0.0` (default),`4.0.0`, atau`3.6.0`.
`KmsKeyId`	Tidak	Jika Anda memilih untuk mengenkripsi klaster Anda, Anda tidak dapat mengubah AWS KMS kunci yang Anda gunakan untuk mengenkripsi klaster Anda.
`MasterUsername`	Tidak	Setelah sebuah klaster telah dibuat, Anda tidak dapat mengubah `MasterUsername` . Aturan penamaan: • Panjangnya adalah [1—63] karakter alfanumerik. • Karakter pertamanya harus berupa huruf. • Tidak bisa menjadi kata yang disimpan oleh mesin basis data.

Parameter	Dapat diubah	Catatan

MasterUserPassword	Ya	Aturan:
		• Panjang adalah [8—100] karakter ASCII yang dapat dicetak.
		• Dapat menggunakan karakter ASCII yang dapat dicetak kecuali untuk berikut ini:
		  • **/** (garis miring ke depan)
		  • **"** (tanda kutip ganda)
		  • **@** (simbol pada)

Port	Ya	Nomor port berlaku untuk semua instans dalam klaster.
PreferredBackupWindow	Ya	
PreferredMaintenanceWindow	Ya	
StorageEncrypted	Tidak	Jika Anda memilih untuk mengenkripsi klaster Anda, klaster tersebut tidak dapat terenkripsi.
StorageType	Ya	Jenis penyimpanan untuk cluster DB: Standard (`standard`) atau I/O-Optimized (`iopt1`). Default: `standard` Parameter ini dapat dikonfigurasi dengan `CreateDBCluster` dan`ModifyDBCluster` . Untuk informasi selengkapnya, lihat Konfigurasi penyimpanan klaster Amazon DocumentDB.
Tags	Ya	

Parameter	Dapat diubah	Catatan
`VpcSecurityGroupIds`	Tidak	Setelah klaster telah dibuat, Anda tidak dapat mengubah VPC tempat klaster berada.

Konfigurasi penyimpanan klaster Amazon DocumentDB

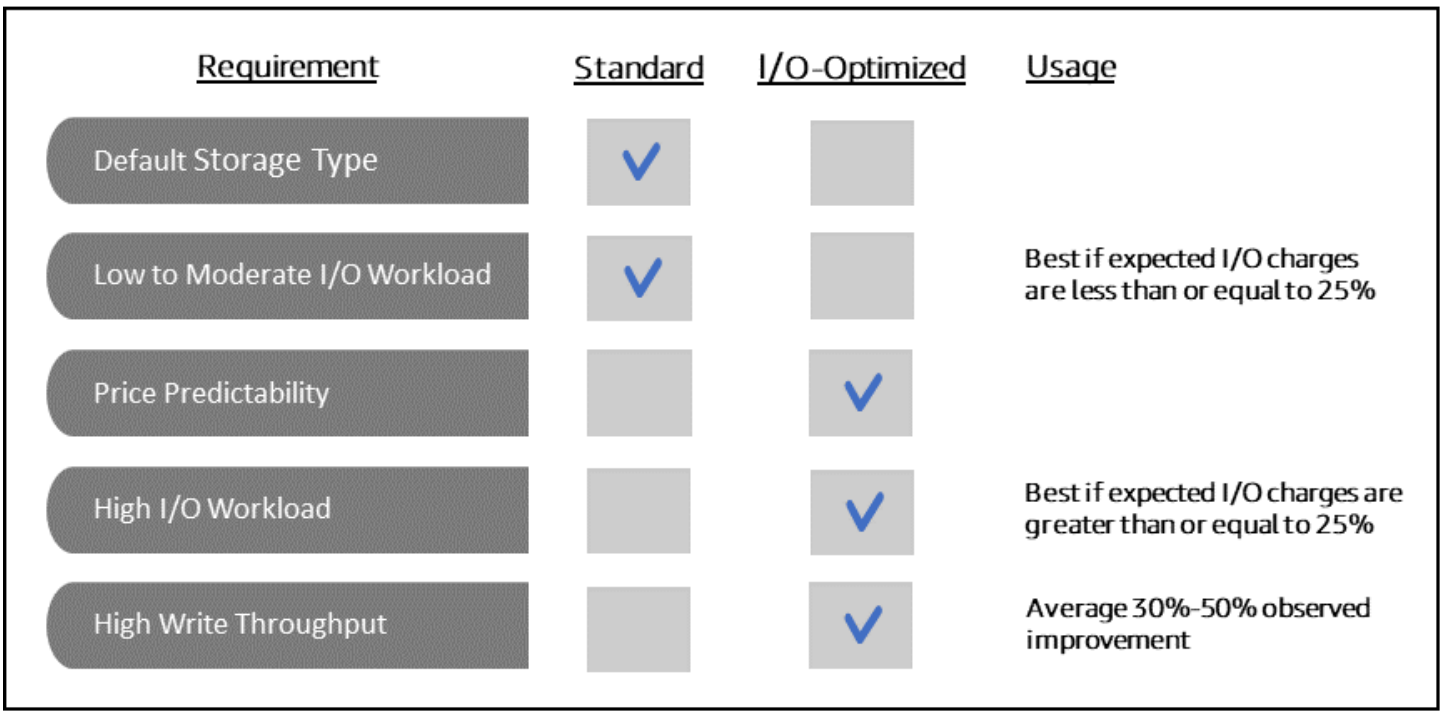
Mulai dari Amazon DocumentDB 5.0, cluster berbasis instance mendukung dua jenis konfigurasi penyimpanan:

- Penyimpanan standar Amazon DocumentDB: Dirancang untuk pelanggan dengan konsumsi I/O rendah hingga sedang. Jika Anda mengharapkan biaya I/O Anda kurang dari 25% dari total cluster Amazon DocumentDB Anda, pilihan ini mungkin ideal untuk Anda. Dengan konfigurasi penyimpanan standar Amazon DocumentDB, Anda ditagih berdasarkan I/O selain biaya instans dan penyimpanan. pay-per-request Ini berarti tagihan Anda mungkin bervariasi dari satu siklus ke siklus lainnya berdasarkan penggunaan. Konfigurasi disesuaikan untuk mengakomodasi permintaan I/O yang berfluktuasi dari aplikasi Anda.
- Amazon DocumentDB I/O penyimpanan yang dioptimalkan: Dirancang untuk pelanggan yang memprioritaskan prediktabilitas harga atau memiliki aplikasi intensif I/O. Konfigurasi I/O yang dioptimalkan menawarkan peningkatan kinerja, peningkatan throughput, dan pengurangan latensi bagi pelanggan dengan beban kerja intensif I/O. Jika Anda mengharapkan biaya I/O Anda melebihi 25% dari total biaya klaster Amazon DocumentDB Anda, opsi ini menawarkan peningkatan kinerja harga. Dengan konfigurasi penyimpanan Amazon DocumentDB I/O yang dioptimalkan, Anda tidak akan dikenakan biaya berdasarkan operasi I/O, memastikan biaya yang dapat diprediksi setiap siklus penagihan. Konfigurasi menstabilkan biaya sekaligus meningkatkan kinerja.

Anda dapat mengalihkan cluster database yang ada setiap 30 hari sekali ke Amazon DocumentDB I/O penyimpanan yang dioptimalkan. Anda dapat beralih kembali ke penyimpanan standar Amazon DocumentDB kapan saja. Tanggal berikutnya untuk memodifikasi konfigurasi penyimpanan ke I/O yang dioptimalkan dapat dilacak dengan `describe-db-clusters` perintah menggunakan AWS CLI atau melalui AWS Management Console di halaman konfigurasi cluster.

Anda dapat membuat kluster database baru termasuk konfigurasi Amazon DocumentDB I/O yang dioptimalkan atau mengonversi kluster database yang ada dengan beberapa klik di, perubahan parameter tunggal di AWS Management Console(), atau melalui SDK.AWS Command Line

InterfaceAWS CLIAWS Tidak ada downtime atau reboot instance yang diperlukan selama atau setelah memodifikasi konfigurasi penyimpanan.

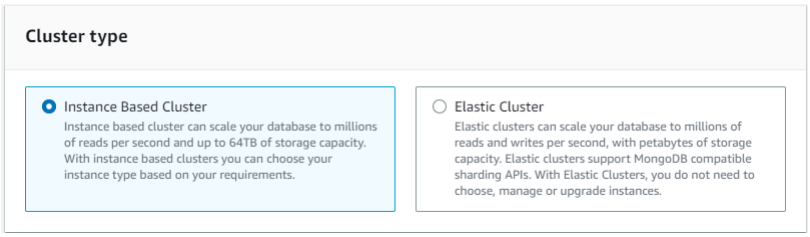


Membuat cluster yang dioptimalkan I/O

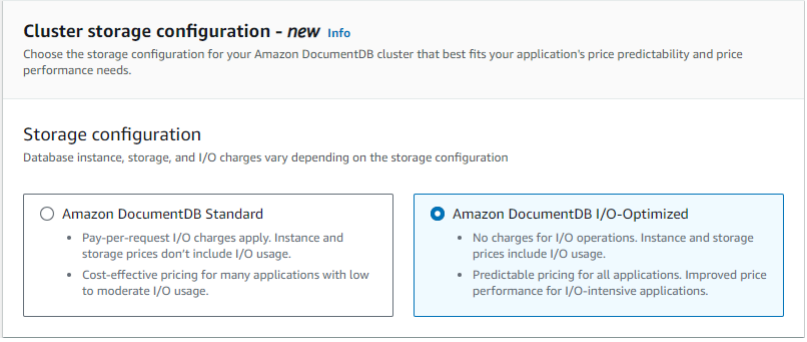
Using the AWS Management Console

Untuk membuat atau memodifikasi cluster yang dioptimalkan I/O menggunakan: AWS Management Console

1. Di konsol manajemen Amazon DocumentDB, di bawah Clusters, pilih Create atau pilih cluster dan pilih Actions, lalu pilih Modify.
2. Jika Anda membuat klaster baru, pastikan Anda memilih Instance Based Clusters di bagian tipe Cluster (ini adalah opsi default).



3. Di bagian Konfigurasi, di bawah konfigurasi penyimpanan Cluster, pilih Amazon DocumentDB I/O Optimized.



4. Selesaikan pembuatan atau modifikasi cluster Anda dan pilih Create cluster atau Modify cluster.

   Untuk proses Create cluster yang lengkap, lihatMembuat cluster dan instance primer menggunakan AWS Management Console.

   Untuk proses klaster Modify yang lengkap, lihatMemodifikasi cluster Amazon DocumentDB.

Using the AWS CLI

Untuk membuat cluster yang dioptimalkan I/O menggunakan: AWS CLI

Dalam contoh-contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Analisis biaya untuk menentukan konfigurasi penyimpanan

Dengan Amazon DocumentDB, Anda memiliki fleksibilitas untuk memilih konfigurasi penyimpanan untuk setiap cluster database yang Anda miliki. Untuk mengalokasikan cluster Anda dengan benar antara standar dan I/O yang dioptimalkan, Anda dapat melacak biaya Amazon DocumentDB Anda berdasarkan klaster. Untuk melakukannya, Anda dapat menambahkan tag ke cluster yang ada, mengaktifkan penandaan alokasi biaya di AWS Billing and Cost Management dasbor Anda, dan menganalisis biaya Anda untuk klaster tertentu di. AWS Cost Explorer Service Untuk informasi tentang analisis biaya, lihat blog kami Menggunakan tag alokasi biaya.

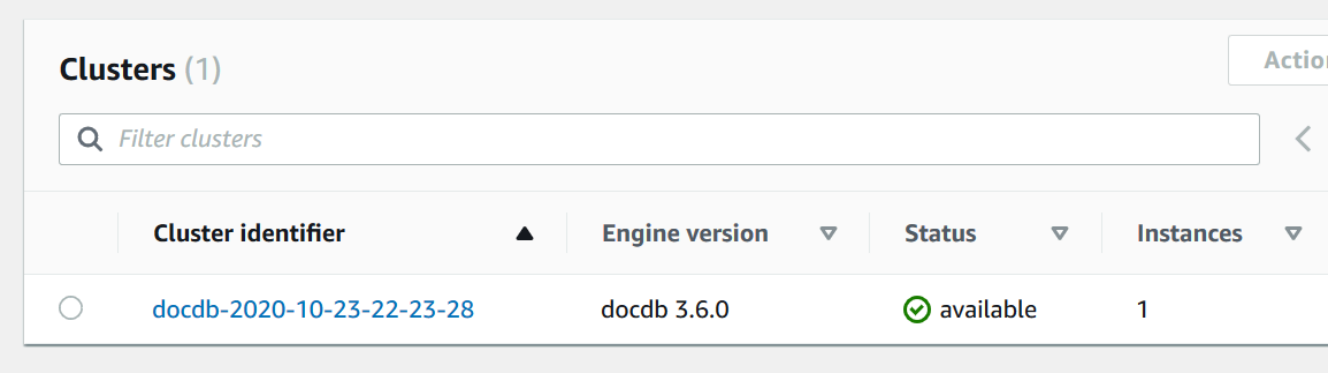
Menentukan status klaster

Anda dapat menentukan status cluster menggunakan AWS Management Console atau AWS CLI.

Using the AWS Management Console

Gunakan prosedur berikut untuk melihat status klaster Amazon DocumentDB Anda menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.
3. Dalam kolom Pengidentifikasi klaster, temukan nama klaster yang Anda minati. Kemudian, untuk menemukan status klaster, bacalah di baris tersebut pada kolom Status, seperti yang ditunjukkan di bawah ini.



Using the AWS CLI

Gunakan operasi `describe-db-clusters` untuk melihat status klaster Amazon DocumentDB Anda menggunakan AWS CLI.

Kode berikut menemukan status klaster `sample-cluster`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Siklus hidup klaster Amazon DocumentDB

Siklus hidup dari Klaster Amazon DocumentDB termasuk menciptakan, menggambarkan, memodifikasi, dan menghapus klaster. Bagian ini menyediakan informasi tentang cara menyelesaikan proses ini.

Topik

- Membuat cluster Amazon DocumentDB
- Menggambarkan cluster Amazon DocumentDB
- Memodifikasi cluster Amazon DocumentDB

- Menentukan pemeliharaan yang tertunda
- Melakukan pembaruan tambalan ke versi mesin cluster
- Menghentikan dan memulai cluster Amazon DocumentDB
- Menghapus cluster Amazon DocumentDB

Membuat cluster Amazon DocumentDB

Sebuah klaster Amazon DocumentDB terdiri dari instans dan volume yang mewakili data untuk klaster. Volume klaster direplikasi dengan enam cara di tiga Availability Zone sebagai volume virtual tunggal. Klaster berisi instans utama dan, secara opsional, hingga 15 instans replika.

Bagian berikut menunjukkan cara membuat cluster Amazon DocumentDB menggunakan salah satu atau AWS Management Console . AWS CLI Anda kemudian dapat menambahkan instans replika tambahan untuk klaster tersebut. Bila Anda menggunakan konsol untuk membuat klaster Amazon DocumentDB, instans utama secara otomatis dibuat untuk Anda pada saat yang sama. Jika Anda menggunakan AWS CLI untuk membuat klaster Amazon DocumentDB, setelah status klaster tersedia, Anda harus membuat instance utama untuk klaster tersebut.

Prasyarat

Berikut ini adalah prasyarat untuk membuat klaster Amazon DocumentDB.

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Prasyarat VPC

Anda hanya dapat membuat cluster Amazon DocumentDB di Amazon Virtual Private Cloud (Amazon VPC). VPC Amazon Anda harus memiliki setidaknya satu subnet di masing-masing setidaknya dua Availability Zone agar Anda dapat menggunakannya dengan cluster Amazon DocumentDB. Dengan mendistribusikan instance klaster di seluruh Availability Zone, Anda memastikan bahwa instance tersedia di klaster jika terjadi kegagalan Availability Zone.

Prasyarat subnet

Saat membuat klaster Amazon DocumentDB, Anda harus memilih VPC dan grup subnet yang sesuai dalam VPC itu untuk meluncurkan klaster Anda. Subnet menentukan Availability Zone dan cakupan IP dalam Availability Zone yang ingin Anda gunakan untuk meluncurkan instans. Untuk tujuan diskusi ini, istilah subnet dan Availability Zone digunakan secara bergantian. Grup subnet adalah set subnet yang telah diberi nama (atau Availability Zone). Apa yang grup subnet izinkan untuk Anda lakukan adalah menentukan Availability Zone yang ingin Anda gunakan untuk meluncurkan instans Amazon DocumentDB. Misalnya, dalam klaster dengan tiga instans, untuk ketersediaan tinggi direkomendasikan agar masing-masing instans tersebut disediakan di Availability Zone yang terpisah. Jadi, jika performa Availability Zone menurun, itu hanya memengaruhi satu instans.

Instans Amazon DocumentDB saat ini dapat disediakan dalam hingga tiga Availability Zone. Bahkan jika grup subnet memiliki lebih dari tiga subnet, Anda hanya dapat menggunakan tiga subnet tersebut untuk membuat klaster Amazon DocumentDB. Oleh karenanya, disarankan agar saat Anda membuat grup subnet, pilih hanya tiga subnet tempat Anda ingin menerapkan instans. Di US East (N. Virginia), grup subnet Anda dapat memiliki enam subnet (atau Availability Zones). Namun, ketika sebuah klaster Amazon DocumentDB disediakan, Amazon DocumentDB memilih tiga dari Availability Zones yang digunakannya untuk menyediakan instans.

Sebagai contoh, misalkan bahwa ketika Anda membuat sebuah klaster, Amazon DocumentDB memilih Availability Zone {1A, 1B, dan 1C}. Jika Anda mencoba untuk membuat sebuah instans di Availability Zone {1D}, panggilan API gagal. Namun, jika Anda memilih untuk membuat sebuah instans tanpa menentukan Availability Zone tertentu, maka Amazon DocumentDB memilih Availability Zone atas nama Anda. Amazon DocumentDB menggunakan algoritme untuk menyeimbangkan beban instans di Availability Zone untuk membantu Anda mencapai ketersediaan tinggi. Sebagai contoh, jika tiga instans disediakan, secara default, mereka disediakan di tiga Availability Zone dan tidak disediakan secara bersamaan di Availability Zone tunggal.

Rekomendasi:

- Kecuali Anda memiliki alasan tertentu, selalu buat grup subnet dengan tiga subnet. Dengan melakukannya, Anda dibantu untuk memastikan bahwa klaster dengan tiga instans atau lebih dapat mencapai ketersediaan yang lebih tinggi karena instans disediakan di tiga Availability Zone.
- Selalu sebarkan instans di beberapa Availability Zone untuk mencapai ketersediaan yang tinggi. Jangan pernah menempatkan semua instans untuk klaster dalam Availability Zone tunggal.
- Karena peristiwa failover dapat terjadi setiap saat, Anda tidak boleh berasumsi bahwa instans utama atau instans replika selalu berada di Availability Zone tertentu.

Prasyarat tambahan

Berikut ini adalah beberapa prasyarat tambahan untuk menciptakan sebuah klaster Amazon DocumentDB:

- Jika Anda terhubung ke kredensial AWS menggunakan AWS Identity and Access Management (IAM), akun IAM Anda harus memiliki kebijakan IAM yang memberikan izin yang diperlukan untuk menjalankan operasi Amazon DocumentDB.

  Jika Anda menggunakan akun IAM untuk mengakses konsol Amazon DocumentDB, Anda harus terlebih dahulu masuk AWS Management Console ke akun dengan IAM Anda. Lalu masuklah ke konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
- Jika Anda ingin menyesuaikan parameter konfigurasi untuk klaster, Anda harus menentukan grup parameter klaster dan grup parameter dengan pengaturan parameter yang diperlukan. Untuk informasi tentang membuat atau memodifikasi grup parameter klaster atau grup parameter, lihat Mengelola grup parameter cluster Amazon DocumentDB.
- Anda harus menentukan nomor port TCP/IP yang Anda inginkan untuk menentukan klaster Anda. Firewall di beberapa perusahaan memblokir koneksi ke port default untuk Amazon DocumentDB. Jika firewall perusahaan Anda memblokir port default, pilih port lain untuk klaster Anda. Semua instans dalam klaster menggunakan port yang sama.

Membuat cluster dan instance primer menggunakan AWS Management Console

Prosedur berikut menjelaskan cara menggunakan konsol untuk meluncurkan klaster Amazon DocumentDB dengan satu atau beberapa instans.

Buat cluster: menggunakan pengaturan default

Untuk membuat cluster dengan instance menggunakan pengaturan default menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Jika Anda ingin membuat klaster di wilayah Wilayah AWS lain selain Wilayah AS Timur (Virginia N.), pilih Wilayah dari daftar di bagian kanan atas konsol.
3. Di panel navigasi, pilih Klaster, lalu pilih Buat.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

4. Pada halaman Buat klaster Amazon DocumentDB, selesaikan panel Konfigurasi.

   a. Pengidentifikasi klaster—Terima nama yang disediakan oleh Amazon DocumentDB, atau masukkan nama untuk klaster Anda; misalnya, **`sample-cluster`**.

      Aturan penamaan klaster:

      - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
      - Karakter pertama harus berupa huruf.
      - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
      - Harus unik untuk semua cluster di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

   b. Versi mesin—Terima versi mesin default 4.0.0, atau pilih 3.6.0.

   c. Kelas instans—Terima default`db.r5.large`, atau pilih kelas instans yang Anda inginkan dari daftar.

   d. Jumlah instans—Dalam daftar, pilih jumlah instans yang ingin Anda buat dengan klaster ini. Instans pertama adalah instans utama, dan semua instans lain adalah instans replika hanya baca. Anda dapat menambahkan dan menghapus instans nanti jika Anda perlu. Secara default, sebuah klaster Amazon DocumentDB melakukan peluncuran dengan tiga instans (satu primer dan dua replika).

5. Lengkapi bagian konfigurasi penyimpanan Cluster.

   Pilih Amazon DocumentDB Standard (default) atau Amazon DocumentDB I/O-Optimized. Untuk informasi selengkapnya, lihat Konfigurasi penyimpanan klaster Amazon DocumentDB.

6. Menyelesaikan panel Autentikasi.

   a. Username —Masukkan nama untuk pengguna utama. Untuk masuk ke cluster Anda, Anda harus menggunakan nama pengguna utama.

      Kendala penamaan pengguna utama:

      - Panjangnya adalah [1—63] karakter alfanumerik.
      - Karakter pertamanya harus berupa huruf.
      - Tidak dapat berupa kata yang dicadangkan oleh mesin basis data.

   b. Kata sandi —Masukkan kata sandi untuk pengguna utama, lalu konfirmasikan. Untuk masuk ke cluster Anda, Anda harus menggunakan kata sandi untuk pengguna utama.

      Aturan kata sandi:

      - Panjangnya adalah [8-100] karakter ASCII yang dapat dicetak.
      - Dapat menggunakan karakter ASCII apa pun yang dapat dicetak kecuali untuk yang berikut ini:
        - **/** (garis miring ke depan)
        - **"** (tanda kutip ganda)
        - **@** (simbol pada)

7. Di bagian bawah layar, pilih salah satu dari berikut ini:

   - Untuk membuat klaster sekarang, pilih Buat klaster.
   - Untuk membatalkan pembuatan klaster, pilih Batalkan.
   - Untuk lebih mengkonfigurasi klaster sebelum membuat, pilih Menampilkan konfigurasi tambahan, dan kemudian lanjutkan di Buat cluster: konfigurasi tambahan.

     Konfigurasi yang tercakup dalam bagian Konfigurasi tambahan adalah sebagai berikut:

     - Pengaturan jaringan—Default di sini akan menggunakan grup keamanan VPC `default`.
     - Opsi klaster—Default di sini akan menggunakan port 27017 dan grup parameter default.

- Enkripsi—Default adalah untuk mengaktifkan enkripsi dengan menggunakan kunci `(default) aws/rds`.

  > **Important**
  >
  > Setelah dienkripsi, klaster tidak dapat dibuat tanpa enkripsi.

- Backup—Default adalah untuk mempertahankan backup selama 1 hari dan membiarkan Amazon DocumentDB memilih jendela backup.
- Ekspor log —Defaultnya adalah tidak mengekspor log audit ke CloudWatch Log.
- Pemeliharaan—Defaultnya adalah membiarkan Amazon DocumentDB memilih jendela pemeliharaan.
- Perlindungan penghapusan—Lindungi klaster Anda dari penghapusan yang tidak disengaja. Default untuk klaster yang dibuat dengan menggunakan konsol diaktifkan.

Jika Anda menerima pengaturan default sekarang, Anda dapat mengubah sebagian besar pengaturan tersebut nanti dengan memodifikasi klaster.

8. Mengaktifkan sambungan masuk untuk grup keamanan klaster Anda.

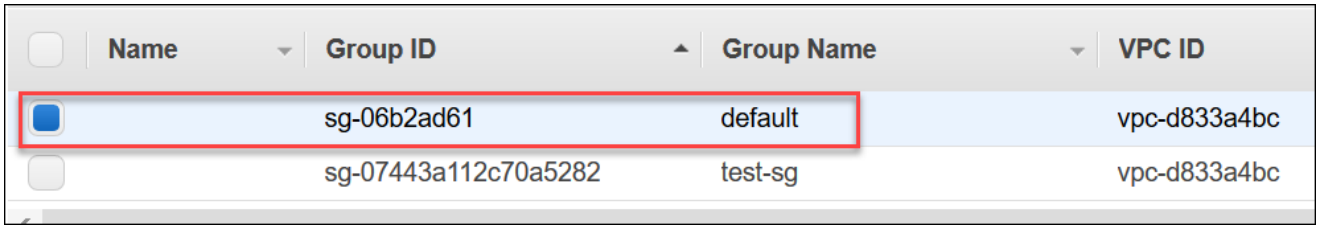
   Jika Anda tidak mengubah pengaturan default untuk klaster Anda, Anda membuat sebuah klaster menggunakan grup keamanan default untuk VPC default di wilayah tertentu. Untuk terhubung ke Amazon DocumentDB, Anda harus mengaktifkan koneksi masuk pada port 27017 (atau port pilihan Anda) untuk grup keamanan klaster Anda.

   Untuk menambahkan koneksi masuk ke grup keamanan klaster Anda

   a. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

   b. Di bagian Sumber Daya dari jendela utama, pilih Grup keamanan.

c. Dari daftar grup keamanan, temukan grup keamanan yang Anda gunakan saat membuat klaster Anda (kemungkinan besar grup keamanan default) dan pilih kotak di sebelah kiri nama grup keamanan.



d. Dari menu Tindakan, pilih Edit aturan masuk lalu pilih atau masukkan batasan aturan.

   i. Jenis—Dari daftar tersebut, pilih protokol untuk membuka lalu lintas jaringan.

   ii. Protokol—Dari daftar, pilih jenis protokol.

   iii. Cakupan Port—Untuk aturan kustom, masukkan nomor port atau cakupan port. Pastikan bahwa nomor port atau cakupan terdiri atas port yang Anda tentukan ketika membuat klaster Anda (default: 27017).

   iv. Sumber—Menentukan lalu lintas yang dapat menjangkau instans Anda. Dari daftar, pilih sumber lalu lintas. Jika Anda memilih Khusus, tentukan alamat IP tunggal atau cakupan alamat IP dalam notasi CIDR (misalnya, 203.0.113.5/32).

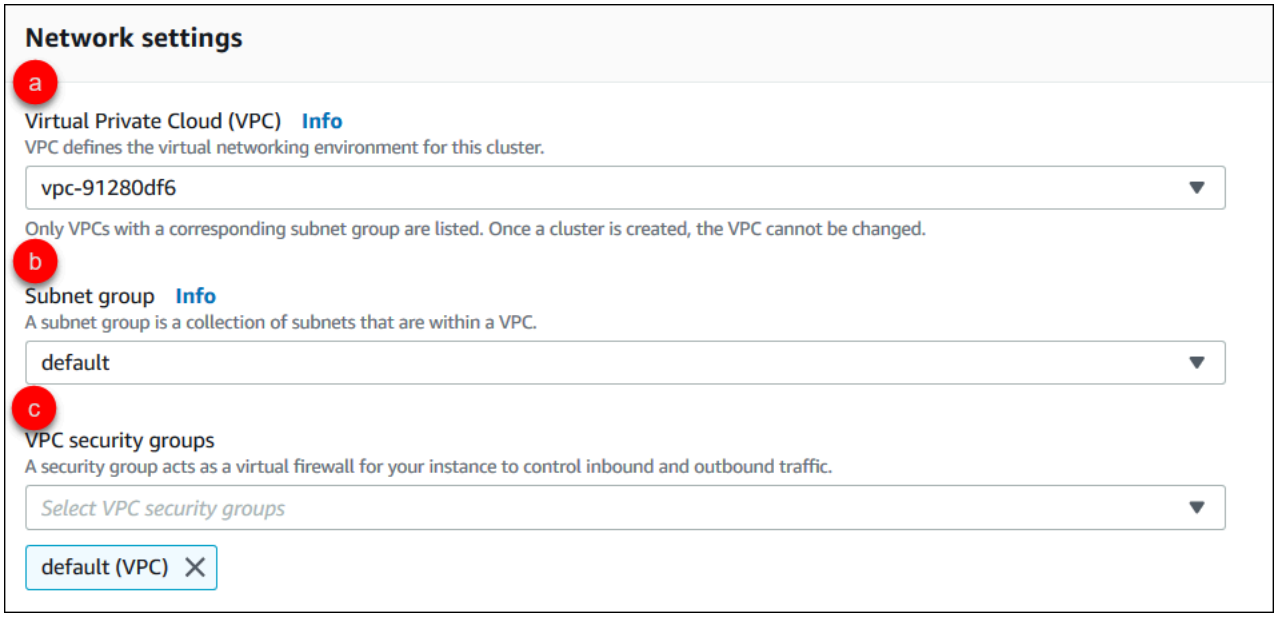
   v. Deskripsi—Masukkan deskripsi untuk aturan ini.

   vi. Setelah selesai membuat aturan, pilih Simpan.

Buat cluster: konfigurasi tambahan

Jika Anda ingin menerima pengaturan default untuk klaster Anda, Anda dapat melewatkan langkah-langkah berikut dan memilih Buat klaster.

1. Menyelesaikan panel Pengaturan jaringan.



a. Virtual Private Cloud (VPC)—Dalam daftar, pilih Amazon VPC tempat Anda ingin meluncurkan klaster ini.

b. Grup subnet—Dalam daftar, pilih grup subnet yang ingin Anda gunakan untuk klaster ini.

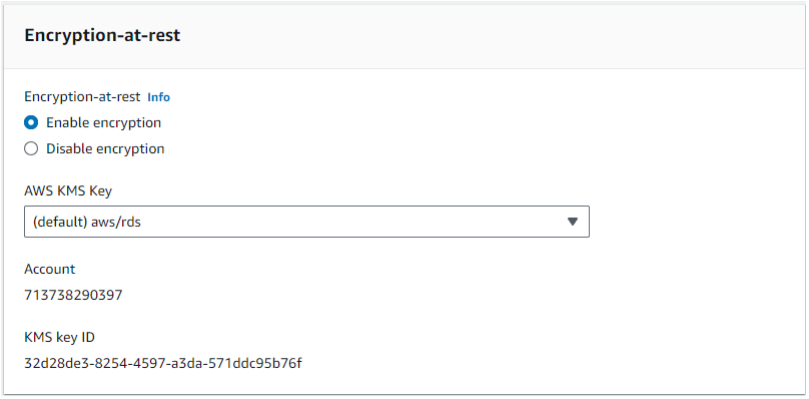
c. Grup keamanan VPC—Di daftar, pilih grup keamanan VPC untuk klaster ini.

2. Lengkapi panel Opsi klaster.

a. Port basis data—Gunakan panah atas dan bawah untuk mengatur port TCP/IP yang akan digunakan aplikasi untuk menyambung ke instans Anda.

b. Grup parameter klaster—Dalam daftar grup parameter, pilih grup parameter klaster untuk klaster ini.

3. Menyelesaikan panel Enkripsi.



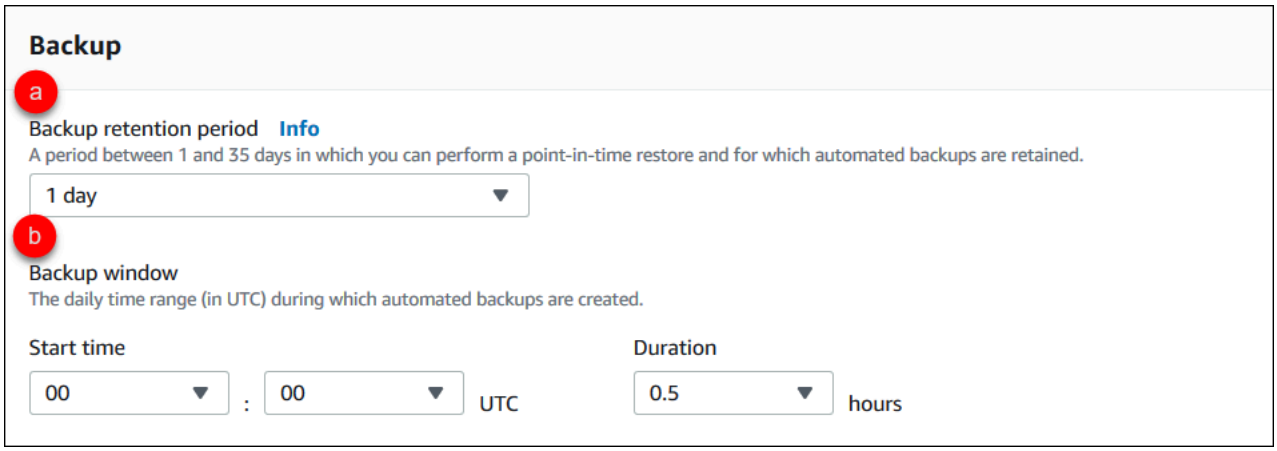
a. E ncryption-at-rest —Pilih salah satu dari berikut ini:

   - Aktifkan enkripsi—Default. Semua data at rest dienkripsi. Jika Anda memilih untuk mengenkripsi data, Anda tidak dapat membatalkan tindakan ini.
   - Menonaktifkan enkripsi—Data Anda tidak dienkripsi.

b. AWS Kunci KMS —Ini hanya tersedia jika Anda mengenkripsi data Anda. Dalam daftar, pilih kunci yang ingin Anda gunakan untuk mengenkripsi data dalam klaster ini. Nilai default-nya `(default) aws/rds`.

   Jika Anda memilih Masukkan ARN kunci, Anda harus memasukkan Amazon Resource Name (ARN) untuk kunci tersebut.

4. Menyelesaikan panel Backup.



a. Periode retensi cadangan—Dalam daftar, pilih jumlah hari untuk menyimpan backup otomatis klaster ini sebelum menghapusnya.

b. Jendela backup—Atur waktu harian dan durasi yang akan digunakan oleh Amazon DocumentDB untuk membuat backup dari klaster ini.

i. Waktu mulai—Pada daftar pertama, pilih jam waktu mulai (UTC) untuk memulai pencadangan otomatis. Dalam daftar kedua, pilih menit dari jam pilihan Anda untuk memulai backup otomatis.

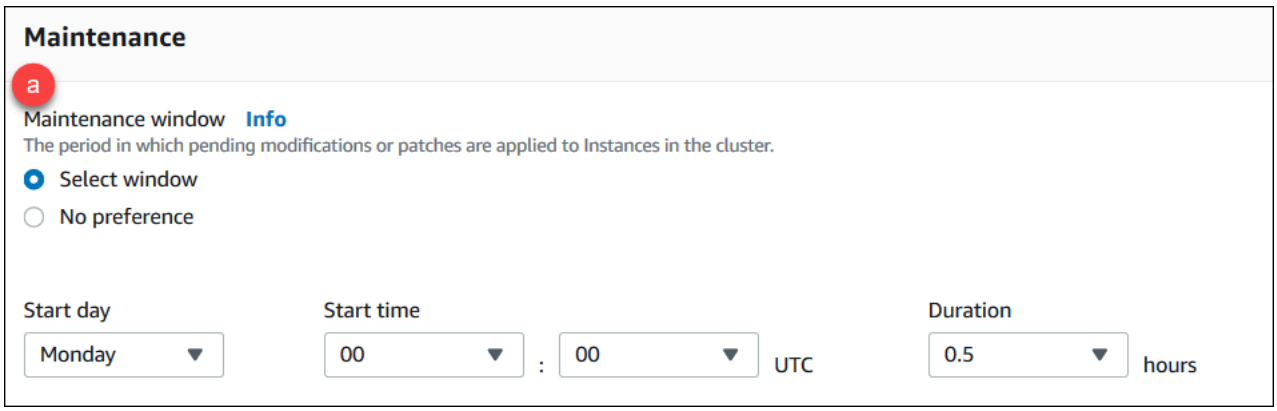
ii. Durasi—Dalam daftar, pilih jumlah jam yang akan dialokasikan untuk membuat cadangan otomatis.

5. Lengkapi panel ekspor Log dengan memilih jenis log yang ingin Anda ekspor ke CloudWatch Log.

- Log audit —Pilih opsi ini untuk mengaktifkan mengekspor log audit ke Amazon CloudWatch Logs. Jika Anda memilih Log audit, Anda harus mengaktifkan `audit_logs` di grup parameter kustom klaster. Untuk informasi selengkapnya, lihat Mengaudit Peristiwa Amazon DocumentDB.
- Log profiler —Pilih opsi ini untuk mengaktifkan mengekspor log profiler operasi ke Amazon Logs. CloudWatch Jika Anda memilih Log profil, Anda juga harus mengubah parameter berikut dalam grup parameter kustom klaster:
  - `profiler`—Setel `keenabled`.
  - `profiler_threshold_ms`—Set ke nilai `[0-INT_MAX]` untuk mengatur ambang batas pada profil operasi.
  - `profiler_sampling_rate`—Set ke nilai `[0.0-1.0]` untuk mengatur persentase operasi lambat ke profil.

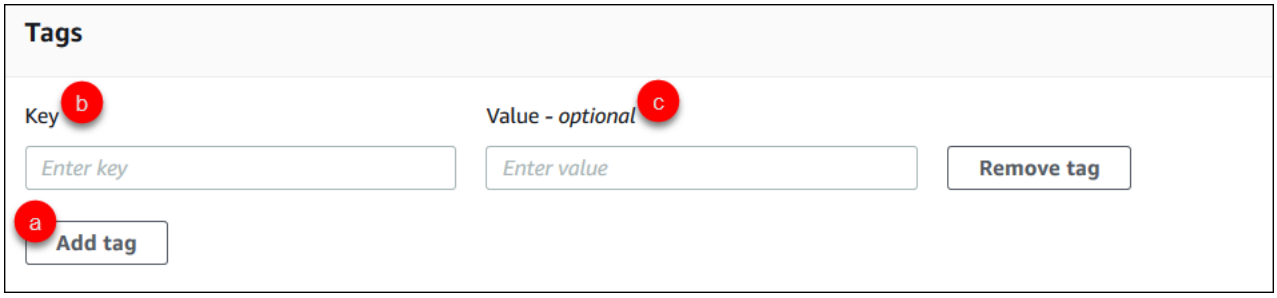
  Untuk informasi selengkapnya, lihat Membuat Profil Operasi Amazon DocumentDB.

6. Menyelesaikan panel Pemeliharaan.



- Pilih salah satu dari berikut
  - Pilih jendela—Anda dapat menentukan hari dalam seminggu, waktu mulai UTC, dan durasi untuk Amazon DocumentDB untuk melakukan pemeliharaan pada klaster Anda.
    a. Hari mulai—Dalam daftar, pilih hari dalam seminggu untuk memulai pemeliharaan klaster.
    b. Waktu mulai—Dalam daftar, pilih jam dan menit (UTC) untuk memulai pemeliharaan.
    c. Durasi—Dalam daftar, pilih berapa banyak waktu yang akan dialokasikan untuk pemeliharaan klaster. Jika pemeliharaan tidak dapat diselesaikan dalam waktu yang ditentukan, proses pemeliharaan akan berlanjut melewati waktu yang ditentukan sampai selesai.
  - Tidak Ada preferensi-Amazon DocumentDB memilih hari dalam seminggu, waktu mulai, dan durasi untuk melakukan pemeliharaan.

7. Jika Anda ingin menambahkan satu atau beberapa tag ke klaster ini, lengkapi panel Tag.



Untuk setiap tag yang ingin Anda tambahkan ke klaster, ulangi langkah-langkah berikut. Anda mungkin memiliki hingga 10 di sebuah klaster.

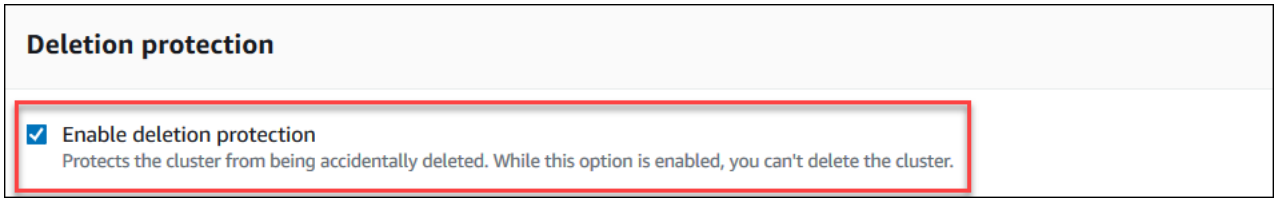
a. Pilih Tambahkan tag.

b. Ketik Kunci tag.

c. Secara opsional, ketik Nilai tag.

Untuk menghapus sebuah tag, pilih Hapus tag.

8. Perlindungan penghapusan diaktifkan secara default ketika Anda membuat klaster dengan menggunakan konsol. Untuk menonaktifkan perlindungan penghapusan, kosongkan Aktifkan perlindungan penghapusan. Ketika diaktifkan, perlindungan penghapusan mencegah agar klaster tidak terhapus. Untuk menghapus klaster yang dilindungi dari penghapusan, Anda harus terlebih dahulu memodifikasi klaster untuk menonaktifkan penghapusan perlindungan.



Untuk informasi selengkapnya tentang perlindungan penghapusan, lihat Menghapus cluster Amazon DocumentDB.

9. Untuk membuat klaster, pilih Buat klaster. Jika tidak, pilih Batalkan.

Membuat cluster menggunakan AWS CLI

Prosedur berikut menjelaskan cara menggunakan AWS CLI untuk meluncurkan cluster Amazon DocumentDB dan membuat replika Amazon DocumentDB.

Parameter

- **`--db-cluster-identifier`**—Diperlukan. Sebuah string huruf kecil yang mengidentifikasi klaster ini.

  Aturan Penamaan klaster:

  - Panjangnya adalah [1-63] huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
  - Harus unik untuk semua cluster (di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB) per akun, per Wilayah. AWS

- **`--engine`**—Diperlukan. Harus berupa **`docdb`**.

- **`--deletion-protection | --no-deletion-protection`**—Opsional. Ketika diaktifkan, perlindungan penghapusan mencegah agar klaster tidak terhapus. Saat Anda menggunakan AWS CLI, pengaturan defaultnya adalah menonaktifkan perlindungan penghapusan.

  Untuk informasi selengkapnya tentang perlindungan penghapusan, lihat Menghapus cluster Amazon DocumentDB.

- **`--storage-type standard | iopt1`**—Opsional. Default: **`standard`**. Konfigurasi penyimpanan cluster. Nilai yang valid adalah `standard` (Standar) atau `iopt1` (I/O dioptimalkan).

- **`--master-username`**—Diperlukan. Nama pengguna yang digunakan untuk mengautentikasi pengguna.

  Aturan Penamaan Pengguna Utama:

  - Panjang adalah [1-63] karakter alfanumerik.
  - Karakter pertamanya harus berupa huruf.
  - Tidak bisa menjadi kata yang disimpan oleh mesin basis data.

- **`--master-user-password`**—Diperlukan. Kata sandi pengguna yang digunakan untuk mengautentikasi pengguna.

  Aturan Kata Sandi Utama:

  - Panjang adalah [8-100] karakter ASCII yang dapat dicetak.
  - Dapat menggunakan karakter ASCII apa pun yang dapat dicetak kecuali untuk yang berikut ini:
    - **`/`** (garis miring ke depan)
    - **`"`** (tanda kutip ganda)
    - **`@`** (simbol pada)

Untuk parameter tambahan, lihat CreateDBCluster.

Untuk meluncurkan cluster Amazon DocumentDB menggunakan AWS CLI

Untuk membuat cluster Amazon DocumentDB, hubungi file. `create-db-cluster` AWS CLI AWS CLI Perintah berikut membuat klaster Amazon DocumentDB `sample-cluster` bernama

dengan perlindungan penghapusan diaktifkan. Untuk informasi selengkapnya tentang perlindungan penghapusan, lihat Menghapus cluster Amazon DocumentDB.

Juga, `--engine-version` adalah parameter opsional yang default ke versi mesin utama terbaru. Versi mesin utama saat ini adalah 4.0.0. Ketika versi mesin utama baru dirilis, versi mesin default untuk `--engine-version` akan diperbarui untuk mencerminkan versi mesin utama yang terakhir. Akibatnya, untuk beban kerja produksi, dan terutama yang bergantung pada skrip, otomatisasi, atau AWS CloudFormation templat, kami menyarankan Anda secara eksplisit menentukan `--engine-version` ke versi utama yang dimaksud.

> **Note**
>
> Jika `db-subnet-group-name` atau `vpc-security-group-id` tidak ditentukan, Amazon DocumentDB akan menggunakan grup subnet default dan grup keamanan Amazon VPC untuk wilayah tertentu.

Untuk Linux, macOS, atau Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 4.0.0 \
      --deletion-protection \
      --master-username masteruser \
      --master-user-password password
```

Untuk Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
      --engine-version 4.0.0 ^
      --deletion-protection ^
      --master-username masteruser ^
      --master-user-password password
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
```

```
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "4.0.0",
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "DBClusterIdentifier": "sample-cluster",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "sample-cluster.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "sample-cluster.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

Dibutuhkan beberapa menit untuk membuat klaster. Anda dapat menggunakan AWS Management Console atau AWS CLI untuk memantau status cluster Anda. Untuk informasi selengkapnya, lihat Memantau Status klaster Amazon DocumentDB.

> **Important**
>
> Saat Anda menggunakan AWS CLI untuk membuat klaster Amazon DocumentDB, tidak ada instance yang dibuat. Akibatnya, Anda harus secara eksplisit membuat instans primer dan semua instans replika yang Anda butuhkan. Anda dapat menggunakan konsol atau AWS CLI untuk membuat instance. Untuk informasi selengkapnya, lihat Menambahkan instance Amazon DocumentDB ke cluster.

Untuk informasi selengkapnya, lihat `CreateDBCluster` di Referensi API Amazon DocumentDB.

Menggambarkan cluster Amazon DocumentDB

Anda dapat menggunakan Amazon DocumentDB Management Console atau AWS CLI untuk melihat detail seperti titik akhir koneksi, grup keamanan, VPC, dan grup parameter yang berkaitan dengan cluster Amazon DocumentDB Anda.

Untuk informasi selengkapnya, lihat berikut ini:

- Memantau Status klaster Amazon DocumentDB
- Menemukan Titik Akhir Klaster

Using the AWS Management Console

Gunakan prosedur berikut untuk melihat perincian klaster Amazon DocumentDB yang ditentukan dengan menggunakan konsol.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Dalam daftar klaster, pilih nama klaster yang ingin Anda lihat perinciannya. Informasi tentang klaster diatur ke dalam pengelompokan berikut:

- Ringkasan — Informasi umum tentang klaster, termasuk versi mesin, status klaster, pemeliharaan tertunda, dan status grup parameter.
- Konektivitas & Keamanan —Bagian Hubungkan berisi daftar titik akhir sambungan untuk membuat sambungan ke klaster ini dengan mongo shell atau dengan aplikasi. Bagian Grup Keamanan terdiri atas daftar grup keamanan yang terkait dengan klaster ini dan ID VPC serta deskripsinya.
- Konfigurasi — Bagian Detail klaster berisi daftar perincian tentang klaster, termasuk Amazon Resource Name (ARN) klaster, titik akhir, dan grup parameter. Ini juga berisi daftar informasi backup klaster, perincian pemeliharaan, dan pengaturan keamanan dan jaringan. Bagian Instans klaster berisi daftar instans yang dimiliki klaster ini dengan masing-masing peran instans dan status grup parameter klaster.
- Monitoring — Metrik Amazon CloudWatch Logs untuk klaster ini. Untuk informasi selengkapnya, lihat Memantau Amazon DocumentDB dengan CloudWatch.
- Acara & tag — Bagian Peristiwa terbaru berisi daftar peristiwa terbaru untuk klaster ini. Amazon DocumentDB menyimpan catatan peristiwa yang terkait dengan klaster, instans, snapshot, grup keamanan, dan grup parameter klaster Anda. Informasi ini mencakup tanggal, waktu, dan pesan yang terkait dengan setiap peristiwa. Bagian Tag berisi daftar tag yang dilampirkan ke klaster ini.

Using the AWS CLI

Untuk melihat detail cluster Amazon DocumentDB Anda menggunakan, gunakan perintah seperti AWS CLI yang ditunjukkan pada contoh `describe-db-clusters` di bawah ini. Untuk informasi selengkapnya, lihat `DescribeDBClusters`di Referensi API Manajemen Sumber Daya Amazon DocumentDB.

> **Note**
>
> Untuk fitur pengelolaan tertentu seperti pengelolaan siklus hidup klaster dan instans, Amazon DocumentDB memanfaatkan teknologi operasi yang dibagikan dengan Amazon RDS. Parameter filter `filterName=engine,Values=docdb` hanya mengembalikan klaster Amazon DocumentDB.

Example

Contoh 1: Daftar semua cluster Amazon DocumentDB

AWS CLI Kode berikut mencantumkan detail untuk semua cluster Amazon DocumentDB di suatu wilayah.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
    "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

Contoh 2: Daftar semua detail untuk klaster Amazon DocumentDB tertentu

AWS CLI Kode berikut mencantumkan rincian untuk cluster`sample-cluster`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
   --filter Name=engine,Values=docdb \
   --db-cluster-identifier sample-cluster
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
   --filter Name=engine,Values=docdb ^
   --db-cluster-identifier sample-cluster
```

Output dari operasi ini terlihat seperti berikut ini.

```
{
    "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

Contoh 3: Daftar detail spesifik untuk klaster Amazon DocumentDB

Untuk mencantumkan subset detail kluster menggunakan AWS CLI, tambahkan yang menentukan anggota klaster mana yang akan dicantumkan `describe-db-clusters` operasi. `--query` Parameter `--db-cluster-identifier` adalah pengenal untuk klaster tertentu yang perinciannya ingin Anda tampilkan. Untuk informasi selengkapnya tentang kueri, lihat Cara Filter Output dengan `--query` Opsi di AWS Command Line Interface Panduan Pengguna.

Contoh berikut mencantumkan instans dalam klaster Amazon DocumentDB.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
    --filter Name=engine,Values=docdb \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
    --filter Name=engine,Values=docdb ^
    --db-cluster-identifier sample-cluster ^
    --query 'DBClusters[*].[DBClusterMembers]'
```

Output dari operasi ini terlihat seperti berikut ini.

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ]
    ]
]
```

Memodifikasi cluster Amazon DocumentDB

Untuk mengubah suatu klaster, klaster tersebut harus dalam kondisi tersedia. Anda tidak dapat mengubah klaster yang dihentikan. Jika klaster berhenti, pertama mulai klaster, tunggu hingga klaster tersedia, lalu buat modifikasi yang diinginkan. Untuk informasi selengkapnya, lihat Menghentikan dan memulai cluster Amazon DocumentDB.

Using the AWS Management Console

Gunakan prosedur berikut untuk memodifikasi klaster Amazon DocumentDB tertentu menggunakan konsol.

Cara memodifikasi klaster Amazon DocumentDB

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Tentukan klaster yang ingin Anda modifikasi dengan memilih tombol di sebelah kiri nama klaster.
4. Pilih Tindakan, lalu pilih Modifikasi.
5. Di Modifikasi klaster: panel <cluster-name>, buat perubahan yang Anda inginkan. Anda dapat membuat perubahan di bidang berikut:
   - Spesifikasi klaster—Nama klaster, grup keamanan, dan kata sandi.
   - Konfigurasi penyimpanan cluster —Mode penyimpanan data cluster. Pilih antara konfigurasi Standar dan I/O-Optimized.
   - Opsi klaster—Port klaster dan grup parameter.
   - Backup—Periode penyimpanan backup klaster dan jendela backup.
   - Ekspor log—Aktifkan atau nonaktifkan ekspor audit atau log pembuat profil.
   - Pemeliharaan—Atur jendela pemeliharaan klaster.
   - Perlindungan penghapusan—Aktifkan atau nonaktifkan perlindungan penghapusan pada klaster. Perlindungan penghapusan diaktifkan secara default.
6. Setelah selesai, pilih Lanjutkan untuk melihat ringkasan perubahan Anda.
7. Jika Anda puas dengan perubahan Anda, Anda dapat memilih Modifikasi klaster untuk memodifikasi klaster Anda. Sebagai alternatif, Anda dapat memilih Kembali atau Batalkan untuk mengedit atau membatalkan perubahan Anda, masing-masing.

Diperlukan waktu beberapa menit agar perubahan Anda diterapkan. Anda dapat menggunakan klaster hanya ketika statusnya tersedia. Anda dapat memantau status klaster menggunakan konsol atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status klaster Amazon DocumentDB.

Using the AWS CLI

Gunakan operasi `modify-db-cluster` untuk mengubah klaster yang ditentukan dengan menggunakan AWS CLI. Untuk informasi selengkapnya, lihat `ModifyDBCluster` di Referensi API Amazon DocumentDB.

Parameter

- **`--db-cluster-identifier`**—Diperlukan. Pengenal klaster Amazon DocumentDB yang akan Anda modifikasi.
- **`--backup-retention-period`**—Opsional. Jumlah hari untuk penyimpanan cadangan otomatis dipertahankan. Nilai yang valid adalah 1–35.
- **`--storage-type`**—Opsional. Konfigurasi penyimpanan cluster. Nilai yang valid adalah `standard` (Standar) atau `iopt1` (I/O dioptimalkan).
- **`--db-cluster-parameter-group-name`**—Opsional. Nama grup parameter klaster yang digunakan untuk klaster.
- **`--master-user-password`**—Opsional. Kata sandi baru untuk pengguna basis data utama.

  Aturan kata sandi:

  - Panjang adalah [8—100] karakter ASCII yang dapat dicetak.
  - Dapat menggunakan karakter ASCII yang dapat dicetak kecuali untuk berikut ini:
    - **`/`** (garis miring ke depan)
    - **`"`** (tanda kutip ganda)
    - **`@`** (simbol pada)
- **`--new-db-cluster-identifier`**—Opsional. Pengenal klaster baru untuk klaster ketika mengubah nama klaster. Nilai ini disimpan sebagai string huruf kecil.

  Aturan penamaan:

  - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
  - Harus unik untuk semua cluster di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB per, per wilayah. Akun AWS
- **`--preferred-backup-window`**—Opsional. Dalam rentang waktu harian selama pencadangan otomatis dibuat, dalam Waktu Terkoordinasi Universal (UTC).

  - Format: `hh24:mm-hh24:mm`
- **`--preferred-maintenance-window`**—Opsional. Rentang waktu mingguan untuk melakukan pemeliharaan sistem dapat terjadi, dalam UTC.
  - Format: `ddd:hh24:mm-ddd:hh24:mm`
  - Hari yang sah: `Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, dan `Sat`.
- **`--deletion-protection`** atau **`--no-deletion-protection`**—Opsional. Apakah perlindungan penghapusan harus diaktifkan pada klaster ini. Perlindungan penghapusan mencegah sebuah klaster dari yang sengaja dihapus sampai klaster dimodifikasi untuk menonaktifkan perlindungan penghapusan. Untuk informasi selengkapnya, lihat Menghapus cluster Amazon DocumentDB.
- **`--apply-immediately`** atau **`--no-apply-immediately`**—Gunakan `--apply-immediately` untuk segera membuat perubahan. Gunakan `--no-apply-immediately` untuk membuat perubahan selama jendela pemeliharaan berikutnya pada klaster Anda.

Example

Kode berikut mengubah periode penyimpanan cadangan untuk klaster `sample-cluster`.

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --apply-immediately \
       --backup-retention-period 7
```

Untuk Windows:

```
aws docdb modify-db-cluster ^
       --db-cluster-identifier sample-cluster ^
       --apply-immediately ^
       --backup-retention-period 7
```

Output dari operasi ini terlihat seperti berikut ini.

```
{
    "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
```

```
    }
}
```

Diperlukan waktu beberapa menit agar perubahan Anda diterapkan. Anda dapat menggunakan klaster hanya ketika statusnya tersedia. Anda dapat memantau status klaster menggunakan konsol atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status klaster Amazon DocumentDB.

Menentukan pemeliharaan yang tertunda

Anda dapat menentukan apakah Anda memiliki versi mesin Amazon DocumentDB terbaru dengan menentukan apakah Anda memiliki pemeliharaan klaster tertunda.

Using the AWS Management Console

Anda dapat menggunakan AWS Management Console untuk menentukan apakah klaster memiliki pemeliharaan yang tertunda.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika panel navigasi tidak terlihat di sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Temukan kolom Pemeliharaan untuk menentukan apakah sebuah klaster memiliki pemeliharaan tertunda.

Tidak ada menunjukkan bahwa klaster menjalankan versi mesin terbaru. Tersedia menunjukkan bahwa klaster memiliki pemeliharaan tertunda, yang mungkin berarti bahwa peningkatan mesin diperlukan.

4. Jika klaster Anda memiliki pemeliharaan tertunda, lanjutkan dengan langkah-langkah di Melakukan pembaruan tambalan ke versi mesin cluster.

Using the AWS CLI

Anda dapat menggunakan AWS CLI untuk menentukan apakah sebuah cluster memiliki versi mesin terbaru dengan menggunakan `describe-pending-maintenance-actions` operasi dengan parameter berikut.

Parameter

- **`--resource-identifier`**—Opsional. ARN untuk sumber daya (klaster). Jika parameter ini dihilangkan, tindakan pemeliharaan tertunda untuk semua klaster didaftarkan.
- **`--region`**—Opsional. Wilayah AWS tempat Anda ingin menjalankan operasi ini, misalnya, `us-east-1`.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-pending-maintenance-actions \
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster \
   --region us-east-1
```

Untuk Windows:

```
aws docdb describe-pending-maintenance-actions ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --region us-east-1
```

Output dari operasi ini terlihat seperti berikut ini.

```
{
```

```
    "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"
                }
            ]
        }
    ]
}
```

Jika klaster Anda memiliki pemeliharaan tertunda, lanjutkan dengan langkah-langkah di Melakukan pembaruan tambalan ke versi mesin cluster.

Melakukan pembaruan tambalan ke versi mesin cluster

Di bagian ini, kami akan menjelaskan cara menerapkan pembaruan tambalan menggunakan AWS Management Console atau. AWS CLI Pembaruan patch adalah pembaruan dalam versi mesin yang sama (misalnya, memperbarui versi mesin 3.6 ke versi mesin 3.6 yang lebih baru). Anda dapat memperbaruinya segera atau selama jendela pemeliharaan berikutnya pada klaster Anda. Untuk menentukan apakah mesin Anda memerlukan pembaruan, lihat Menentukan pemeliharaan yang tertunda. Silakan dicatat bahwa ketika Anda menerapkan pembaruan, klaster Anda akan mengalami waktu henti selama beberapa saat.

> **Note**
>
> Jika Anda mencoba untuk meng-upgrade dari versi mesin utama ke yang lain, seperti 3.6 ke 5.0, lihat salah satu Peningkatan versi utama Amazon DocumentDB di tempat atauMemutakhirkan cluster Amazon DocumentDB Anda menggunakan AWS Database Migration Service. Upgrade versi utama di tempat hanya mendukung docdb 5.0 sebagai versi mesin target.

Ada dua persyaratan konfigurasi untuk mendapatkan pembaruan patch terbaru untuk sebuah versi mesin pada suatu klaster:

- Status klaster harus tersedia.
- Klaster harus menjalankan versi mesin sebelumnya.

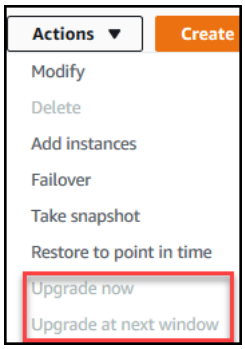
Using the AWS Management Console

Prosedur berikut menerapkan pembaruan patch untuk versi mesin pada klaster dengan menggunakan konsol. Anda memiliki pilihan untuk memperbarui segera atau selama jendela pemeliharaan berikutnya pada klaster Anda.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Pada panel navigasi, silakan pilih Klaster. Dalam daftar klaster, pilih tombol di sebelah kiri pada klaster yang ingin Anda tingkatkan. Status klaster harus tersedia.

   > **Tip**
   >
   > Jika panel navigasi tidak terlihat di sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Dari menu Tindakan, pilih salah satu dari berikut ini. Opsi menu ini dapat dipilih hanya jika klaster yang Anda pilih tidak menjalankan versi mesin terbaru.



   - Tingkatkan sekarang—Segera memulai proses peningkatan. Klaster Anda akan offline untuk sementara waktu, sedangkan klaster ditingkatkan ke versi mesin terbaru.

- Tingkatkan pada jendela berikutnya—Memulai proses peningkatan selama jendela pemeliharaan berikutnya pada klaster. Klaster Anda akan offline untuk sementara waktu, sedangkan klaster ditingkatkan ke versi mesin terbaru.

4. Jika jendela konfirmasi terbuka, pilih salah satu hal berikut:
   - Meningkatkan—Untuk meningkatkan klaster Anda ke versi mesin terbaru sesuai dengan jadwal yang dipilih pada langkah sebelumnya.
   - Batalkan—Untuk membatalkan peningkatan mesin klaster dan melanjutkan dengan versi mesin saat ini pada klaster.

Using the AWS CLI

Anda dapat menerapkan pembaruan tambalan ke cluster Anda menggunakan AWS CLI dan `apply-pending-maintenance-action` operasi dengan parameter berikut.

Parameter

- **`--resource-identifier`**—Diperlukan. ARN dari klaster Amazon DocumentDB yang akan Anda tingkatkan.
- **`--apply-action`**—Diperlukan. Nilai berikut diizinkan. Untuk meningkatkan versi mesin klaster Anda, gunakan `db-upgrade`.
  - **`db-upgrade`**
  - **`system-update`**
- **`--opt-in-type`**—Diperlukan. Nilai berikut diizinkan.
  - `immediate`—Terapkan tindakan pemeliharaan segera.
  - `next-maintenance`—Terapkan tindakan pemeliharaan selama jendela pemeliharaan berikutnya.
  - `undo-opt-in`—Membatalkan permintaan keikutsertaan `next-maintenance` yang ada.

Example

Patch contoh berikut memperbarui versi mesin `sample-cluster` ke versi 4.0.0.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut ini.

```
{
    "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"
            }
        ]
    }
}
```

Menghentikan dan memulai cluster Amazon DocumentDB

Menghentikan dan memulai klaster Amazon DocumentDB membantu Anda mengelola biaya untuk pengembangan dan pengujian lingkungan. Alih-alih menciptakan dan menghapus klaster dan instans setiap kali Anda menggunakan Amazon DocumentDB, untuk sementara, Anda dapat menghentikan semua instans pada klaster ketika tidak diperlukan. Anda kemudian dapat memulainya lagi saat melanjutkan pengujian.

Topik

- Ikhtisar menghentikan dan memulai klaster
- Operasi yang dapat Anda lakukan pada cluster yang dihentikan

Ikhtisar menghentikan dan memulai klaster

Selama periode di mana Anda tidak memerlukan klaster Amazon DocumentDB, Anda dapat menghentikan semua instans dalam klaster itu sekaligus. Anda lalu dapat memulai klaster lagi kapan saja Anda memerlukannya. Memulai dan menghentikan menyederhanakan proses pengaturan dan perombakan pada klaster yang digunakan untuk pengembangan, pengujian, atau aktifitas serupa yang tidak memerlukan ketersediaan berkelanjutan. Anda dapat menghentikan dan memulai cluster menggunakan AWS Management Console atau AWS CLI dengan satu tindakan, terlepas dari berapa banyak instance yang ada di cluster.

Sementara klaster Anda berhenti, volume penyimpanan klaster tetap tidak berubah. Anda hanya dikenakan biaya untuk penyimpanan, snapshot manual, dan penyimpanan cadangan otomatis dalam jendela penyimpanan yang Anda tentukan. Anda tidak dikenakan biaya untuk jam instans apa pun. Amazon DocumentDB secara otomatis memulai klaster Anda setelah tujuh hari sehingga tidak tertinggal pembaruan pemeliharaan yang diperlukan. Ketika klaster Anda dimulai setelah tujuh hari, Anda akan mulai dikenakan biaya untuk instans di klaster lagi. Sementara klaster Anda berhenti, Anda tidak dapat mengajukan kueri untuk volume penyimpanan Anda karena kueri mengharuskan instans dalam keadaan tersedia.

Ketika sebuah klaster Amazon DocumentDB dihentikan, baik klaster maupun instans dapat dimodifikasi dengan cara apa pun. Ini termasuk menambahkan atau menghapus instans, atau menghapus klaster.

Using the AWS Management Console

Prosedur berikut menunjukkan kepada Anda caranya menghentikan sebuah klaster dengan satu atau beberapa instans dalam keadaan tersedia, atau caranya memulai klaster yang telah dihentikan.

Untuk menghentikan atau memulai klaster Amazon DocumentDB

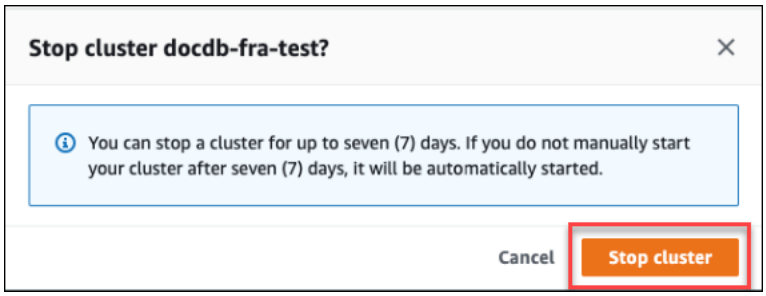
1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika panel navigasi tidak terlihat di sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

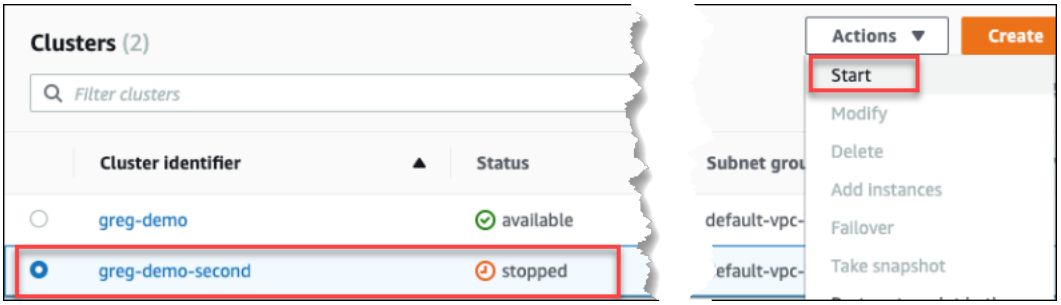
3. Dalam daftar klaster, pilih tombol di sebelah kiri pada klaster yang ingin Anda mulai atau tingkatkan.
4. Pilih Tindakan, lalu pilih tindakan yang ingin Anda lakukan pada klaster.
   - Jika Anda ingin menghentikan klaster dan klaster tersedia:
     a. Pilih Berhenti.

     Untuk menghindari pengaktifan mekanisme failover, operasi stop menghentikan instans replika terlebih dahulu, lalu instans utama.

     b. Pada dialog konfirmasi, konfirmasikan bahwa Anda ingin menghentikan klaster dengan memilih Hentikan klaster, atau untuk mempertahankan klaster agar terus berjalan, pilih Batalkan.



   - Jika Anda ingin memulai klaster, dan klaster berhenti, pilih Mulai.



5. Pantau status klaster dan instansnya. Jika Anda memulai klaster, Anda dapat melanjutkan penggunaan klaster ketika klaster dan instansnya tersedia. Untuk informasi selengkapnya, lihat Menentukan status klaster.

Using the AWS CLI

Contoh kode berikut menunjukkan kepada Anda caranya menghentikan sebuah klaster dengan satu atau beberapa instans dalam keadaan tersedia, atau caranya memulai klaster yang telah dihentikan.

Untuk menghentikan cluster dengan satu atau lebih instance yang tersedia menggunakan AWS CLI, gunakan `stop-db-cluster` operasi. Untuk memulai klaster yang telah dihentikan, gunakan operasi `start-db-cluster`. Kedua operasi menggunakan parameter `--db-cluster-identifier`.

Parameter:

- **`--db-cluster-identifier`**—Diperlukan. Nama klaster untuk berhenti atau mulai.

Example — Untuk menghentikan cluster menggunakan AWS CLI

Kode berikut menghentikan klaster `sample-cluster`. Klaster harus memiliki satu atau beberapa instans dalam keadaan tersedia.

Untuk Linux, macOS, atau Unix:

```
aws docdb stop-db-cluster \
   --db-cluster-identifier sample-cluster
```

Untuk Windows:

```
aws docdb stop-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Example — Untuk memulai cluster menggunakan AWS CLI

Kode berikut memulai klaster `sample-cluster`. Klaster saat ini harus dihentikan.

Untuk Linux, macOS, atau Unix:

```
aws docdb start-db-cluster \
   --db-cluster-identifier sample-cluster
```

Untuk Windows:

```
aws docdb start-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Operasi yang dapat Anda lakukan pada cluster yang dihentikan

Saat klaster Amazon DocumentDB dihentikan, Anda dapat melakukan point-in-time pemulihan ke titik mana pun dalam jendela retensi cadangan otomatis yang Anda tentukan. Untuk detail tentang melakukan point-in-time pemulihan, lihatMemulihkan ke Titik Waktu.

Anda tidak dapat memodifikasi konfigurasi klaster Amazon DocumentDB, atau salah satu instansnya saat klaster dihentikan. Anda juga tidak dapat menambahkan atau menghapus instans dari klaster, atau menghapus klaster tersebut jika masih memiliki instans terkait. Anda harus memulai klaster sebelum melakukan tindakan administratif tersebut.

Amazon DocumentDB menerapkan pemeliharaan terjadwal pada klaster yang dihentikan hanya setelah dimulai lagi. Setelah tujuh hari, Amazon DocumentDB secara otomatis memulai klaster yang telah dihentikan sehingga tidak jauh tertinggal dalam status pemeliharaan. Ketika klaster dinyalakan kembali, Anda akan mulai kembali dikenakan biaya untuk instans dalam klaster.

Sementara klaster dihentikan, Amazon DocumentDB tidak melakukan backup otomatis atau memperpanjang periode retensi cadangan.

Menghapus cluster Amazon DocumentDB

Anda dapat menghapus cluster Amazon DocumentDB menggunakan AWS Management Console atau file. AWS CLI Untuk menghapus sebuah klaster, klaster harus dalam kondisi tersedia dan tidak harus memiliki instans apa pun yang terkait dengannya. Jika klaster berhenti, pertama mulai klaster, tunggu sampai klaster tersedia, kemudian hapus klaster tersebut. Untuk informasi selengkapnya, lihat Menghentikan dan memulai cluster Amazon DocumentDB.

Perlindungan penghapusan

Untuk melindungi klaster Anda dari penghapusan tidak disengaja, Anda dapat mengaktifkan perlindungan penghapusan. Perlindungan penghapusan diaktifkan secara default ketika Anda membuat klaster dengan menggunakan konsol. Namun, perlindungan penghapusan dinonaktifkan secara default jika Anda membuat klaster menggunakan AWS CLI.

Amazon DocumentDB memberlakukan perlindungan penghapusan untuk klaster, tidak soal apakah Anda melakukan operasi hapus dengan menggunakan konsol atau AWS CLI. Jika perlindungan penghapusan diaktifkan, Anda tidak dapat menghapus klaster. Untuk menghapus klaster yang mengaktifkan perlindungan penghapusan, Anda harus terlebih dahulu mengubah klaster dan menonaktifkan perlindungan penghapusan.

Saat menggunakan konsol dengan perlindungan penghapusan yang diaktifkan di sebuah klaster, Anda tidak dapat menghapus instans terakhir pada klaster karena hal itu juga akan menghapus klaster. Anda dapat menghapus instans terakhir dari klaster dengan perlindungan penghapusan menggunakan AWS CLI. Namun, klaster itu sendiri masih ada, dan data Anda disimpan. Anda dapat mengakses data dengan membuat instans baru untuk klaster. Untuk informasi selengkapnya tentang mengaktifkan dan menonaktifkan perlindungan penghapusan, lihat:

- Membuat cluster Amazon DocumentDB
- Memodifikasi cluster Amazon DocumentDB

Using the AWS Management Console

Untuk menghapus cluster menggunakan AWS Management Console, perlindungan penghapusan harus dinonaktifkan.

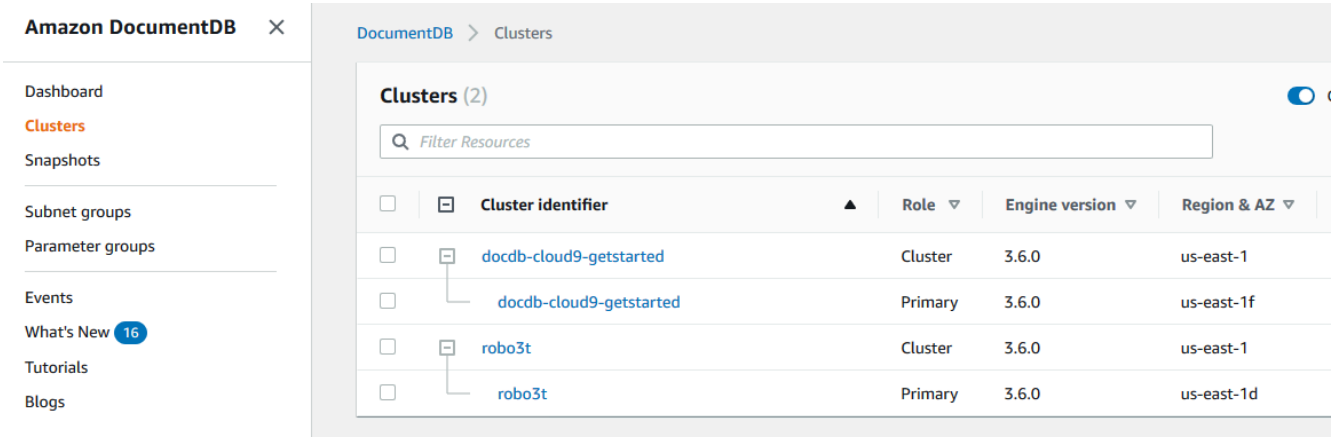
Untuk menentukan apakah sebuah klaster memiliki perlindungan penghapusan yang diaktifkan:

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.

2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Perhatikan bahwa dalam kotak navigasi Klaster, kolom Pengidentifikasi klaster menunjukkan klaster dan instans sekaligus. Instans tercantum di bawah klaster, mirip dengan gambar di bawah ini.



4. Pilih nama klaster, dan pilih tab Konfigurasi. Di bagian Detail klaster, temukan Perlindungan penghapusan. Jika perlindungan penghapusan diaktifkan, modifikasi klaster untuk menonaktifkan perlindungan penghapusan. Untuk informasi tentang memodifikasi klaster, lihat Memodifikasi cluster Amazon DocumentDB.

   Setelah Perlindungan penghapusan dinonaktifkan, Anda siap untuk menghapus klaster.

Untuk menghapus klaster:

1. Pada panel navigasi, silakan pilih Klaster.
2. Tentukan apakah klaster memiliki instans dengan memeriksa kolom Instans. Sebelum dapat menghapus klaster, Anda harus menghapus semua instans. Untuk informasi selengkapnya, lihat Menghapus instans Amazon DocumentDB .
3. Tergantung pada apakah klaster Anda memiliki instans, lakukan salah satu langkah berikut.

- Jika klaster tidak memiliki instans, pilih tombol di sebelah kiri nama klaster dan pilih Tindakan. Dari menu pilihan menurun, pilih Hapus. Lengkapi kotak dialog Hapus <cluster-name>, lalu pilih Hapus.
- Jika klaster memiliki satu atau beberapa instans, lakukan hal berikut:

  a. Di panel navigasi, pilih Instans.

  b. Menghapus setiap instans klaster. Ketika Anda menghapus instans terakhir, klaster juga dihapus. Untuk informasi tentang menghapus instans, lihat Menghapus instans Amazon DocumentDB .

Dibutuhkan beberapa menit untuk menghapus klaster. Untuk memantau status klaster, lihat Memantau Status klaster Amazon DocumentDB.

Using the AWS CLI

Anda tidak dapat menghapus klaster yang memiliki instans yang berhubungan dengan itu. Untuk menentukan instans mana yang terkait dengan klaster Anda, jalankan perintah `describe-db-clusters` dan hapus semua instans klaster. Kemudian, jika diperlukan, nonaktifkan perlindungan penghapusan pada klaster Anda, dan terakhir, hapus klaster.

1. Pertama, hapus semua instance cluster.

   Untuk menentukan instans mana yang perlu Anda hapus, jalankan perintah berikut.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

   Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    [
        "sample-cluster",
        [
            "sample-instance-1",
            "sample-instance-2"
        ]
    ]
]
```

Jika klaster yang Anda ingin hapus memiliki suatu instans, hapuslah dengan cara yang ditunjukkan di bawah ini.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-instance
```

2. Kedua, nonaktifkan perlindungan penghapusan.

Menggunakan AWS CLI untuk menghapus semua instance cluster tidak menghapus cluster. Anda juga harus menghapus klaster, tetapi Anda dapat melakukan ini hanya jika perlindungan penghapusan dinonaktifkan.

Untuk menentukan apakah klaster memiliki perlindungan penghapusan yang diaktifkan, jalankan perintah berikut.

> **Tip**
>
> Untuk melihat status perlindungan penghapusan di seluruh klaster Amazon DocumentDB Anda, hilangkan parameter `--db-cluster-identifier`.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterIdentifier,DeletionProtection]'
```

Output dari operasi ini terlihat seperti berikut ini.

```
[
    [
        "sample-cluster",
        "true"
    ]
]
```

Jika klaster memiliki perlindungan penghapusan yang diaktifkan, modifikasi klaster dan nonaktifkan perlindungan penghapusan. Untuk menonaktifkan perlindungan penghapusan pada klaster, jalankan perintah berikut.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --no-deletion-protection \
    --apply-immediately
```

3. Akhirnya, hapus cluster.

   Setelah perlindungan penghapusan dinonaktifkan, Anda siap untuk menghapus klaster. Untuk menghapus klaster, gunakan operasi `delete-db-cluster` dengan parameter berikut.

   - **`--db-cluster-identifier`**—Diperlukan. Pengenal klaster yang ingin Anda hapus.
   - **`--final-db-snapshot-identifier`**—Opsional. Jika Anda ingin snapshot akhir, Anda harus menyertakan parameter ini dengan nama untuk snapshot akhir. Anda harus menyertakan `--final-db-snapshot-identifier` atau `--skip-final-snapshot`.

     Aturan penamaan:

     - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
     - Karakter pertama harus berupa huruf.
     - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
     - Harus unik untuk semua cluster di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB per, per Wilayah. Akun AWS
   - **`--skip-final-snapshot`**—Opsional. Gunakan parameter ini hanya jika Anda tidak ingin mengambil snapshot akhir sebelum menghapus klaster Anda. Pengaturan default adalah untuk mengambil snapshot akhir. Anda harus menyertakan `--final-db-snapshot-identifier` atau `--skip-final-snapshot`.

   AWS CLI Kode berikut menghapus cluster `sample-cluster` dengan snapshot akhir. Operasi gagal jika ada setiap instans yang terkait dengan klaster atau jika perlindungan penghapusan diaktifkan.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Untuk Windows:

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Example

AWS CLI Kode berikut menghapus cluster `sample-cluster` tanpa mengambil snapshot akhir.

Untuk Linux, macOS, atau Unix:

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

Untuk Windows:

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --skip-final-snapshot
```

Output dari operasi `delete-db-cluster` adalah klaster Anda akan Anda hapus.

Dibutuhkan beberapa menit untuk menghapus klaster. Untuk memantau status klaster, lihat Memantau status klaster.

Menskalakan cluster Amazon DocumentDB

Amazon DocumentDB memungkinkan Anda untuk menskalakan penyimpanan dan melakukan komputasi dalam klaster Anda berdasarkan kebutuhan Anda. Bagian ini menjelaskan bagaimana Anda dapat menggunakan penskalaan untuk penyimpanan, penskalaan untuk instans, dan penskalaan untuk pembacaan guna mengelola performa dan penskalaan untuk klaster dan instans Amazon DocumentDB.

Topik

- Penskalaan penyimpanan
- Penskalaan instans
- Penskalaan baca
- Tulis penskalaan

Penskalaan penyimpanan

Penyimpanan Amazon DocumentDB secara otomatis diskalakan dengan data dalam volume klaster Anda. Seiring pertumbuhan data Anda, penyimpanan volume cluster Anda tumbuh dalam peningkatan 10 GiB, hingga 128 TiB.

Penskalaan instans

Anda dapat menskalakan klaster Amazon DocumentDB sesuai kebutuhan dengan memodifikasi kelas instans untuk setiap instans dalam klaster. Amazon DocumentDB mendukung beberapa kelas instans yang dioptimalkan untuk Amazon DocumentDB.

Untuk informasi selengkapnya, lihat Memodifikasi instance Amazon DocumentDB.

Penskalaan baca

Anda dapat mencapai penskalaan baca untuk klaster Amazon DocumentDB dengan membuat hingga 15 replika Amazon DocumentDB dalam klaster. Setiap replika Amazon DocumentDB menghasilkan data yang sama dari volume klaster dengan sedikit penundaan replika—biasanya kurang dari 100 milidetik setelah instans utama menulis pembaruan. Saat lalu lintas baca meningkat, Anda dapat membuat replika Amazon DocumentDB tambahan dan terhubung langsung ke sana untuk mendistribusikan beban baca pada klaster Anda. Replika Amazon DocumentDB tidak harus dari kelas instans DB yang sama dengan instans utama.

Untuk informasi selengkapnya, lihat Menambahkan instance Amazon DocumentDB ke cluster.

Untuk membaca penskalaan dengan Amazon DocumentDB, Anda sebaiknya membuat sambungan ke klaster Anda sebagai set replika dan mendistribusikan pembacaan ke instans replika menggunakan kemampuan preferensi baca bawaan driver Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke Amazon DocumentDB sebagai Set Replika

Tulis penskalaan

Anda dapat menskalakan kapasitas tulis pada klaster Amazon DocumentDB Anda dengan meningkatkan ukuran instans utama klaster Anda. Bagian ini menyediakan dua metode untuk menskalakan instans utama klaster Anda berdasarkan kebutuhan. Opsi pertama berusaha untuk meminimalkan dampak aplikasi tetapi membutuhkan lebih banyak langkah yang perlu diselesaikan. Opsi kedua mengoptimalkan kesederhanaan karena memiliki langkah yang lebih sedikit, tetapi dilengkapi dengan tradeoff yang memiliki dampak yang lebih potensial untuk aplikasi Anda.

Bergantung pada aplikasi Anda, Anda bisa memilih pendekatan apa di bawah ini yang terbaik untuk Anda. Untuk informasi selengkapnya tentang ukuran dan biaya instans yang tersedia, lihat halaman Harga Amazon DocumentDB.

1. Optimalkan ketersediaan dan performa tinggi — Jika Anda terhubung ke klaster Anda di mode set replika (disarankan), Anda dapat menggunakan proses berikut untuk meminimalkan dampak aplikasi Anda saat menskalakan instans primer Anda. Metode ini meminimalkan dampak karena mempertahankan klaster Anda pada atau di atas ketersediaan tinggi Anda, dan target penskalaan baca ditambahkan ke klaster sebagai instans, alih-alih diperbarui pada tempatnya.

    a. Tambahkan satu atau beberapa replika dari jenis instans yang lebih besar ke klaster Anda (lihat ???). Semua replika sebaiknya memiliki jenis instans yang sama dengan atau lebih besar dari yang utama. Hal ini untuk menghindari pengurangan yang tidak disengaja dalam performa tulis akibat kegagalan terhadap jenis instans yang lebih kecil. Untuk sebagian besar pelanggan, ini berarti penggandaan jumlah instans sementara di klaster mereka, kemudian penghapusan replika yang lebih kecil setelah penskalaan selesai.

    b. Atur tingkat failover pada semua replika baru ke prioritas nol, hal ini memastikan replika jenis instans yang lebih kecil memiliki prioritas failover tertinggi. Untuk informasi selengkapnya, lihat ???.

    c. Memulai failover manual, yang akan mempromosikan salah satu replika baru menjadi instans utama. Untuk informasi selengkapnya, lihat ???.

> **Note**
>
> Ini akan menimbulkan ~30 detik waktu henti untuk klaster Anda. Silakan buat rencana yang sesuai.

d. Hapus semua replika dengan jenis instans yang lebih kecil daripada intans utama baru Anda dari klaster.

e. Kembalikan tingkat failover semua intans ke prioritas yang sama (biasanya, ini berarti pengaturan kembali ke 1).

Sebagai contoh, anggaplah Anda memiliki klaster yang saat ini berisi tiga instans `r5.large` (satu primer dan dua replika), dan Anda ingin melakukan penskalaan ke tipe instans `r5.xlarge`. Untuk melakukannya, pertama Anda akan menambahkan tiga instans replika `r5.xlarge` ke klaster Anda, lalu tetapkan tingkat failover replika `r5.xlarge` baru ke nol. Selanjutnya, Anda akan memulai failover manual (diketahui bahwa aplikasi Anda akan mengalami ~ 30 detik waktu henti). Setelah failover selesai, Anda akan menghapus ketiga instans `r5.large` dari klaster Anda, sehingga klaster tersebut diskalakan ke instans `r5.xlarge`.

Untuk membantu mengoptimalkan biaya, instans Amazon DocumentDB ditagih dalam kenaikan satu detik, dengan biaya minimum sepuluh menit setelah perubahan status yang dapat ditagih seperti membuat, memodifikasi, atau menghapus instans. Untuk informasi selengkapnya lihat Pengoptimalan Biaya dalam dokumentasi praktik terbaik.

2. Optimalisasi untuk kesederhanaan — Pendekatan ini dioptimalkan untuk menyederhanakan prosesnya. Cara ini tidak memperluas dan menyempitkan klaster, tetapi mungkin mengurangi kapasitas baca Anda untuk sementara.

   Ada kemungkinan bahwa mengubah kelas instans replika akan menghasilkan instans yang tidak melayani permintaan untuk jangka waktu singkat, dari beberapa detik hingga kurang dari 30 detik. Jika Anda terhubung ke klaster Anda di mode set replika (direkomendasikan), maka ini akan mengurangi kapasitas baca Anda pada satu replika (misalnya, untuk 66% kapasitas dalam klaster 3-node, atau 75% kapasitas dalam klaster 4-node, dll) selama operasi penskalaan.

   a. Skalakan salah satu contoh instans di klaster Anda. Untuk informasi selengkapnya, lihat Mengelola Kelas Instans.

   b. Tunggu sampai instans tersedia (lihatMemantau Status instans Amazon DocumentDB).

> **Note**
>
> Ini akan menimbulkan ~30 detik waktu henti untuk klaster Anda. Silakan buat rencana yang sesuai.

c. Lanjutkan menjalankan langkah 1 dan 2 sampai semua instans replika telah diskalakan, satu per satu.

d. Memulai failover manual. Ini akan mempromosikan salah satu replika menjadi instans utama. Untuk informasi selengkapnya, lihat Failover Amazon DocumentDB.

> **Note**
>
> Tindakan ini akan menyebabkan waktu henti hingga 30 detik untuk klaster Anda, tetapi seringkali membutuhkan waktu lebih singkat. Silakan buat rencana yang sesuai.

e. Skalakan instans primer sebelumnya (sekarang replika).

Mengkloning volume untuk cluster Amazon DocumentDB

Dengan menggunakan kloning Amazon DocumentDB, Anda dapat membuat cluster baru yang menggunakan volume cluster Amazon DocumentDB yang sama dan memiliki data yang sama dengan aslinya. Prosesnya dirancang agar cepat dan hemat biaya. Klaster baru dengan volume data terkaitnya dikenal sebagai Klon. Membuat klon lebih cepat dan lebih hemat ruang daripada menyalin data secara fisik menggunakan teknik lain, seperti memulihkan snapshot.

Amazon DocumentDB mendukung pembuatan klon yang disediakan Amazon DocumentDB dari klaster Amazon DocumentDB yang disediakan. Saat Anda membuat klon menggunakan konfigurasi penyebaran yang berbeda dari sumbernya, klon dibuat menggunakan versi terbaru dari mesin Amazon DocumentDB sumber.

Saat Anda membuat klon dari kluster Amazon DocumentDB Anda, klon dibuat AWS di akun Anda—akun yang sama yang memiliki cluster Amazon DocumentDB sumber.

Topik

- Ikhtisar kloning Amazon DocumentDB
- Keterbatasan kloning Amazon DocumentDB

- Cara kerja kloning Amazon DocumentDB
- Membuat klon Amazon DocumentDB

Ikhtisar kloning Amazon DocumentDB

Amazon DocumentDB menggunakan protokol untuk copy-on-write membuat klon. Mekanisme ini menggunakan ruang tambahan minimal untuk membuat klon awal. Saat klon pertama kali dibuat, Amazon DocumentDB menyimpan satu salinan data yang digunakan oleh cluster DB sumber dan cluster Amazon DocumentDB baru (kloning). Penyimpanan tambahan dialokasikan hanya ketika perubahan dilakukan pada data (pada volume penyimpanan Amazon DocumentDB) oleh cluster Amazon DocumentDB sumber atau klon klon Amazon DocumentDB. Untuk mempelajari lebih lanjut tentang copy-on-write protokol, lihatCara kerja kloning Amazon DocumentDB.

Kloning Amazon DocumentDB sangat berguna untuk menyiapkan lingkungan pengujian dengan cepat menggunakan data produksi Anda, tanpa risiko kerusakan data. Anda dapat menggunakan klon untuk berbagai jenis aplikasi, seperti berikut:

- Bereksperimen dengan potensi perubahan (misalnya, perubahan skema dan perubahan grup parameter) untuk menilai semua dampak.
- Menjalankan operasi sarat beban kerja, seperti mengekspor data atau menjalankan kueri analitis.
- Membuat salinan dari klaster DB produksi Anda untuk pengembangan, pengujian, atau tujuan lainnya.

Anda dapat membuat lebih dari satu klon dari cluster Amazon DocumentDB yang sama. Anda juga dapat membuat beberapa klon dari klon lain.

Setelah membuat klon Amazon DocumentDB, Anda dapat mengonfigurasi instans Amazon DocumentDB secara berbeda dari cluster Amazon DocumentDB sumber. Misalnya, Anda mungkin tidak memerlukan klon untuk tujuan pengembangan agar memenuhi persyaratan ketersediaan tinggi yang sama dengan klaster Amazon DocumentDB produksi sumber. Dalam hal ini, Anda dapat mengonfigurasi klon dengan satu instance Amazon DocumentDB daripada beberapa instans DB yang digunakan oleh cluster Amazon DocumentDB.

Setelah selesai menggunakan klon untuk pengujian, pengembangan, atau tujuan lainnya, Anda dapat menghapusnya.

Keterbatasan kloning Amazon DocumentDB

Amazon DocumentDB; kloning saat ini memiliki batasan sebagai berikut:

- Anda dapat membuat klon sebanyak yang Anda inginkan, hingga jumlah maksimum klaster DB yang diizinkan di Wilayah AWS. Namun, setelah Anda membuat 15 klon, klon berikutnya adalah salinan lengkap. Operasi kloning bertindak seperti point-in-time pemulihan.
- Anda tidak dapat membuat klon di AWS Wilayah yang berbeda dari cluster Amazon DocumentDB sumber.
- Anda tidak dapat membuat klon dari cluster Amazon DocumentDB yang tidak memiliki instans DB. Anda hanya dapat mengkloning cluster Amazon DocumentDB yang memiliki setidaknya satu instans DB.
- Anda dapat membuat klon di cloud pribadi virtual (VPC) yang berbeda dari klaster Amazon DocumentDB. Jika Anda melakukannya, subnet VPC harus dipetakan ke Zona Ketersediaan yang sama.

Cara kerja kloning Amazon DocumentDB

Kloning Amazon DocumentDB bekerja pada lapisan penyimpanan cluster Amazon DocumentDB. Ini menggunakan copy-on-writeprotokol yang cepat dan hemat ruang dalam hal media tahan lama yang mendasarinya yang mendukung volume penyimpanan Amazon DocumentDB. Anda dapat mempelajari lebih lanjut tentang volume klaster Amazon DocumentDB di. Mengelola cluster Amazon DocumentDB

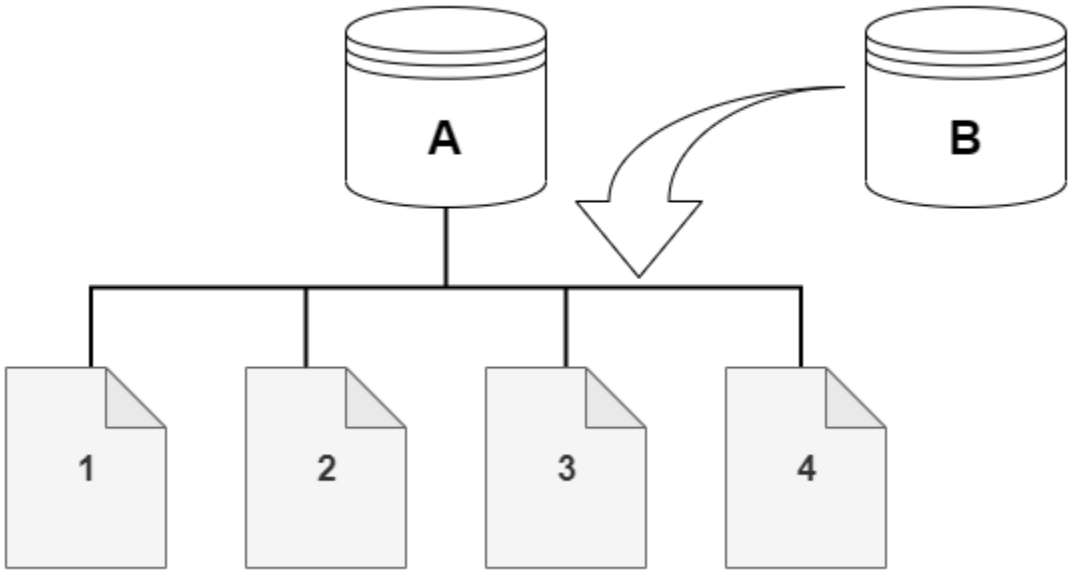
Topik

- Memahami copy-on-write protokol
- Menghapus volume klaster sumber

Memahami copy-on-write protokol

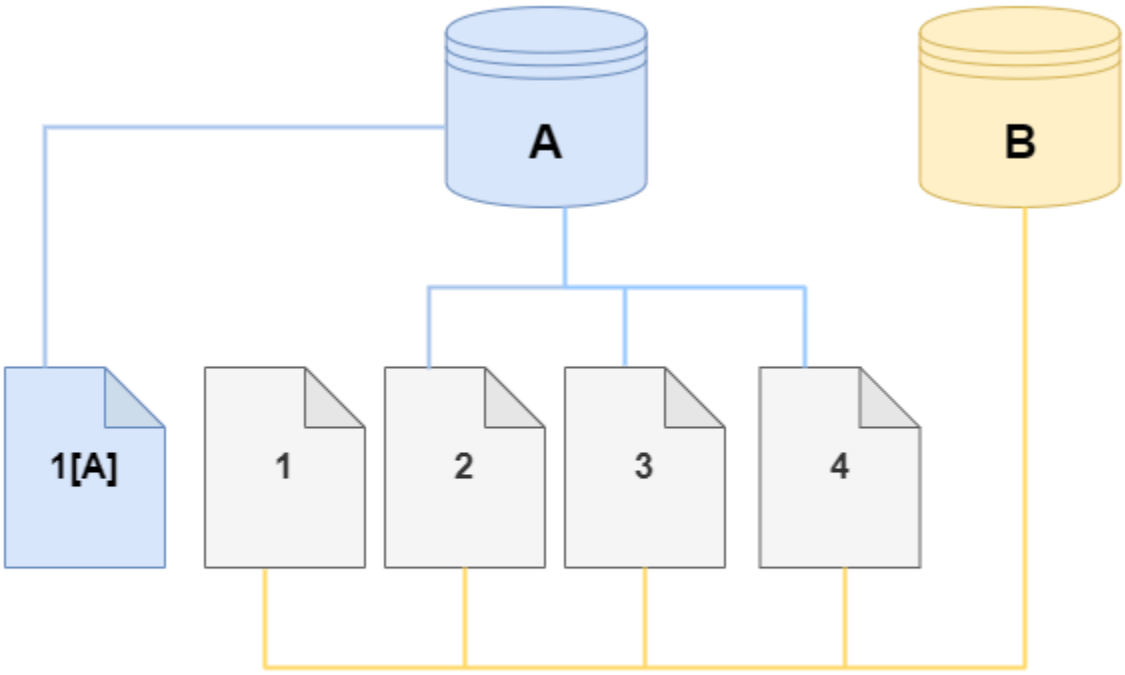
Cluster Amazon DocumentDB menyimpan data di halaman dalam volume penyimpanan Amazon DocumentDB yang mendasarinya.

Misalnya, dalam diagram berikut Anda dapat menemukan cluster Amazon DocumentDB (A) yang memiliki empat halaman data, 1, 2, 3, dan 4. Bayangkan bahwa klon, B, dibuat dari cluster Amazon DocumentDB. Saat klon dibuat, tidak ada data yang disalin. Sebaliknya, klon menunjuk ke kumpulan halaman yang sama dengan cluster Amazon DocumentDB sumber.

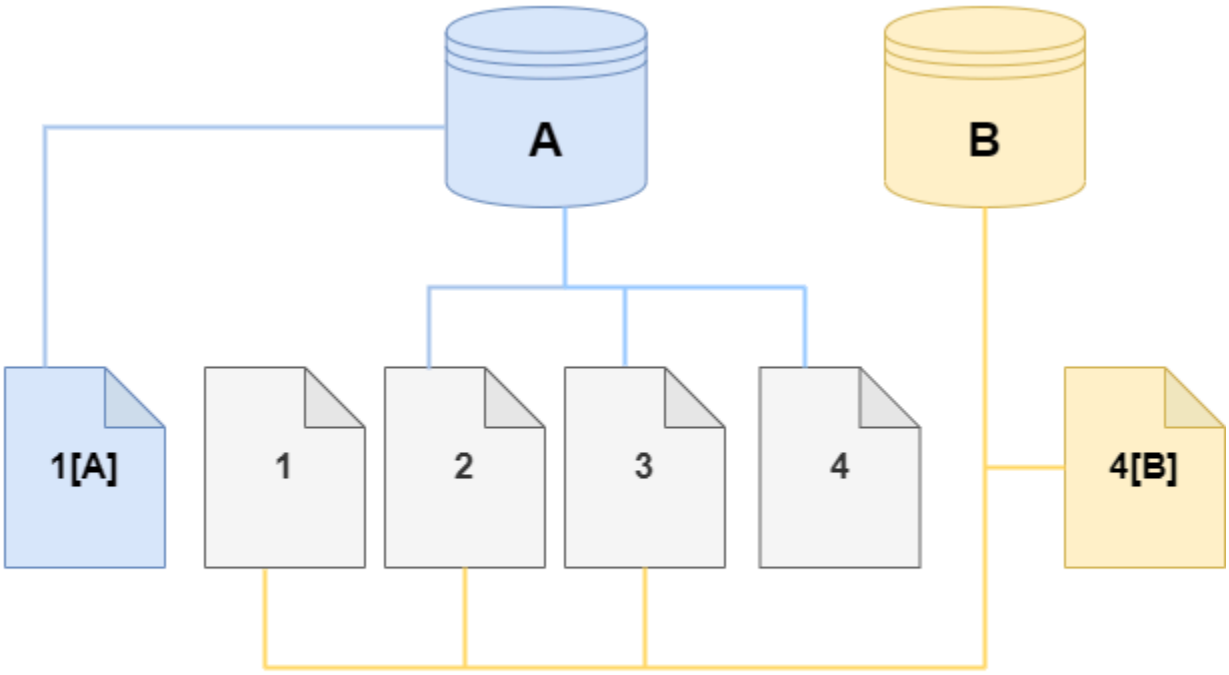


Saat klon dibuat, tidak ada penyimpanan tambahan yang biasanya diperlukan. copy-on-write Protokol menggunakan segmen yang sama pada media penyimpanan fisik sebagai segmen sumber. Penyimpanan tambahan hanya diperlukan jika kapasitas segmen sumber tidak cukup untuk seluruh segmen klon. Jika demikian, segmen sumber disalin ke perangkat fisik lain.

Dalam diagram berikut, Anda dapat menemukan contoh copy-on-write protokol dalam tindakan menggunakan cluster A yang sama dan klonnya, B, seperti yang ditunjukkan sebelumnya. Katakanlah Anda membuat perubahan pada klaster Amazon DocumentDB (A) yang menghasilkan perubahan pada data yang disimpan di halaman 1. Alih-alih menulis ke halaman asli 1, Amazon DocumentDB membuat halaman baru 1 [A]. Volume cluster Amazon DocumentDB untuk cluster (A) sekarang menunjuk ke halaman 1 [A], 2, 3, dan 4, sedangkan klon (B) masih mereferensikan halaman asli.



Pada klon, perubahan dibuat pada halaman 4 di volume penyimpanan. Alih-alih menulis ke halaman asli 4, Amazon DocumentDB membuat halaman baru, 4 [B]. Klon sekarang menunjuk ke halaman 1, 2, 3, dan halaman 4[B], sementara klaster (A) terus menunjuk ke 1[A], 2, 3, dan 4.



Karena semakin banyak perubahan terjadi dari waktu ke waktu di volume cluster Amazon DocumentDB sumber dan klon, lebih banyak penyimpanan diperlukan untuk menangkap dan menyimpan perubahan.

Menghapus volume klaster sumber

Saat Anda menghapus volume klaster sumber yang memiliki satu atau beberapa klon yang berkaitan dengannya, klon ini tidak akan terpengaruh. Klon terus menunjuk ke halaman yang sebelumnya dimiliki oleh volume klaster sumber.

Membuat klon Amazon DocumentDB

Anda dapat membuat klon di AWS akun yang sama dengan cluster Amazon DocumentDB sumber. Untuk melakukannya, Anda dapat menggunakan AWS Management Console atau AWS CLI dan prosedur berikut.

Dengan menggunakan kloning Amazon DocumentDB, Anda dapat membuat klon klon Amazon DocumentDB yang disediakan dari klon Amazon DocumentDB yang disediakan.

Using the AWS Management Console

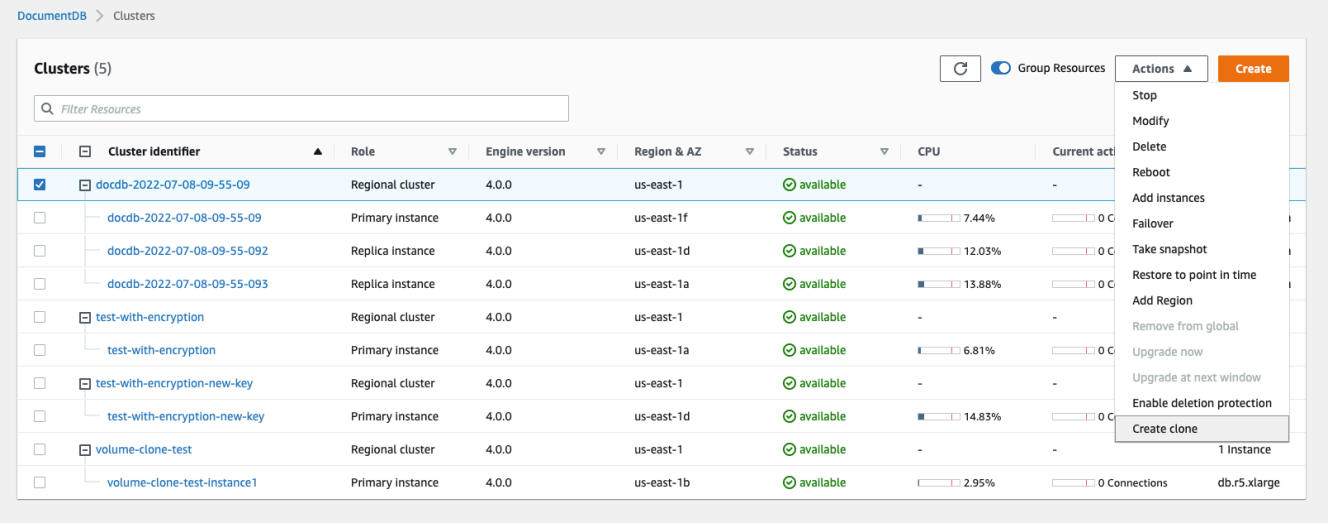
Prosedur berikut menjelaskan cara mengkloning cluster Amazon DocumentDB menggunakan. AWS Management Console

Membuat klon menggunakan AWS Management Console hasil di cluster Amazon DocumentDB dengan satu instance Amazon DocumentDB.

Instruksi ini berlaku untuk cluster DB yang dimiliki oleh AWS akun yang sama yang membuat klon. Cluster DB harus dimiliki oleh AWS akun yang sama karena kloning lintas akun tidak didukung di Amazon DocumentDB.

Untuk membuat tiruan dari cluster DB yang dimiliki oleh AWS akun Anda menggunakan AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Pada panel navigasi, silakan pilih Klaster.
3. Pilih klaster Amazon DocumentDB Anda dari daftar, dan untuk Tindakan, pilih Buat klon.

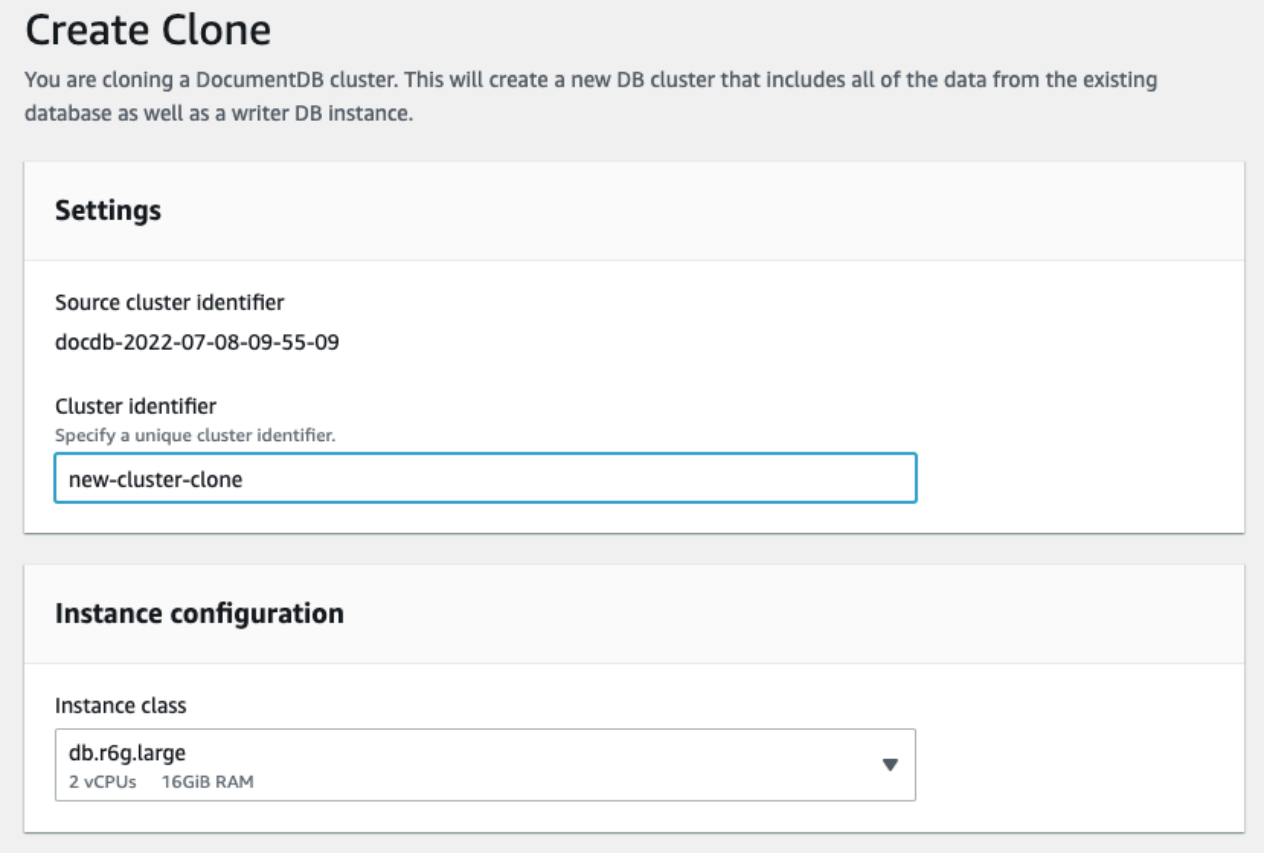


Halaman Create clone terbuka, di mana Anda dapat mengonfigurasi pengenal Cluster dan kelas Instance, dan opsi lain untuk klon klon Amazon DocumentDB.

4. Di bagian Pengaturan, lakukan hal berikut:

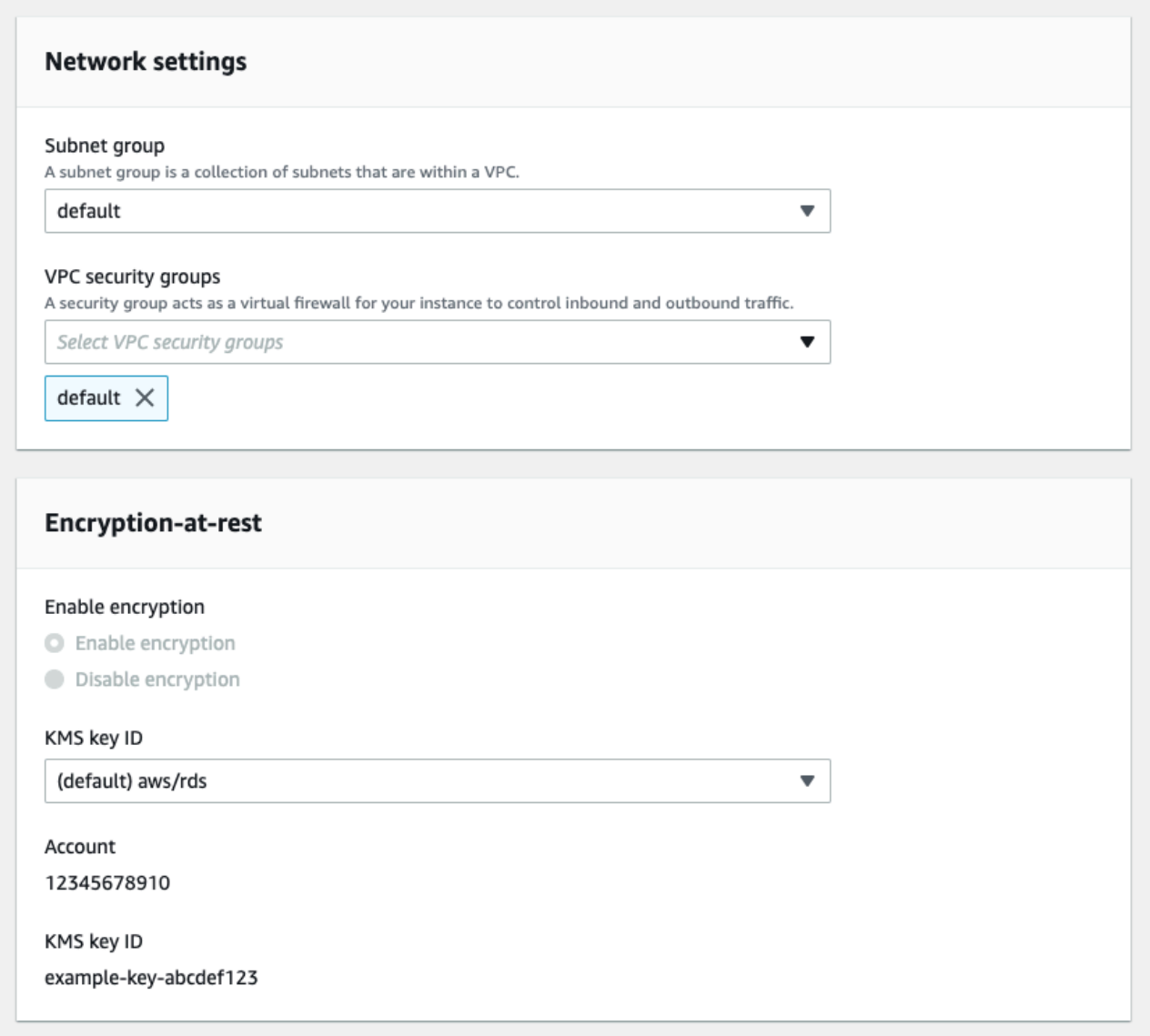
   a. Untuk pengidentifikasi Cluster, masukkan nama yang ingin Anda berikan ke klaster Amazon DocumentDB kloning Anda.

b. Untuk konfigurasi Instance, pilih class Instance yang sesuai untuk klaster Amazon DocumentDB kloning Anda.

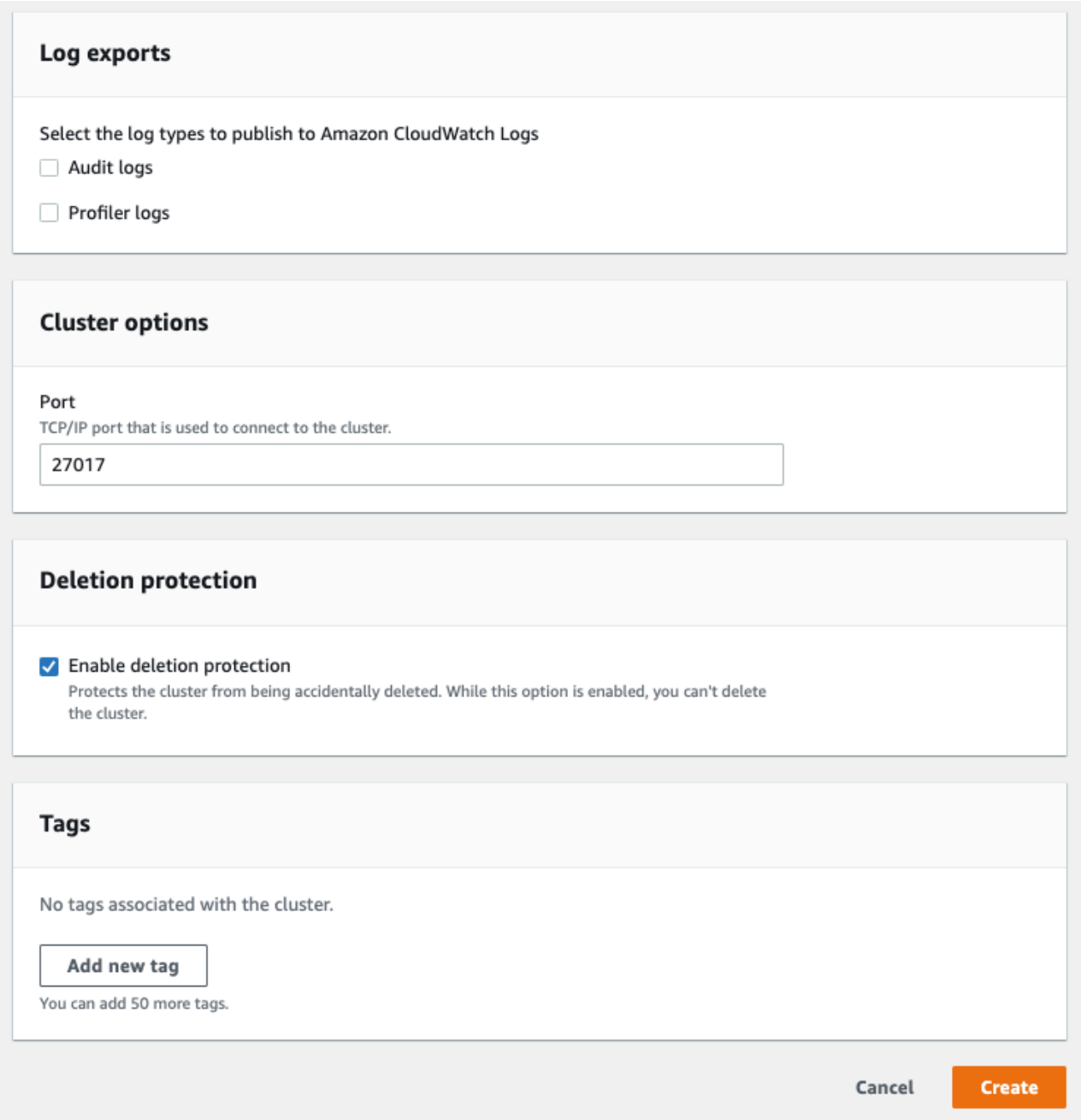


c. Untuk pengaturan Jaringan, pilih grup Subnet untuk kasus penggunaan Anda dan grup keamanan VPC terkait.

d. Untuk E ncryption-at-rest, jika cluster sumber (cluster yang sedang dikloning) mengaktifkan enkripsi, klaster kloning juga harus mengaktifkan enkripsi. Jika skenario ini benar, maka opsi Enable encryption berwarna abu-abu (dinonaktifkan) tetapi dengan pilihan Enable encryption yang dipilih. Sebaliknya, jika cluster sumber tidak mengaktifkan enkripsi, opsi Enable encryption tersedia dan Anda dapat memilih untuk mengaktifkan atau menonaktifkan enkripsi.



e. Selesaikan konfigurasi klon cluster baru dengan memilih jenis log yang akan diekspor (opsional), memasukkan port tertentu yang digunakan untuk terhubung ke cluster, dan mengaktifkan perlindungan dari penghapusan cluster secara tidak sengaja (diaktifkan secara default).



f. Selesai memasukkan semua pengaturan untuk klon klon Amazon DocumentDB Anda. Untuk mempelajari selengkapnya tentang setelan cluster dan instans Amazon DocumentDB, lihat. Mengelola cluster Amazon DocumentDB

5. Pilih Buat klon untuk meluncurkan klon Amazon DocumentDB dari klaster Amazon DocumentDB pilihan Anda.

Saat klon dibuat, klon tersebut terdaftar dengan kluster Amazon DocumentDB Anda yang lain di bagian Database konsol dan menampilkan statusnya saat ini. Klon Anda siap digunakan ketika statusnya Tersedia.

Using the AWS CLI

Menggunakan AWS CLI untuk mengkloning cluster Amazon DocumentDB Anda melibatkan beberapa langkah.

`restore-db-cluster-to-point-in-time` AWS CLI Perintah yang Anda gunakan menghasilkan klaster Amazon DocumentDB kosong dengan 0 instance Amazon DocumentDB. Artinya, perintah hanya mengembalikan cluster Amazon DocumentDB, bukan instance DB untuk cluster itu. Anda melakukannya secara terpisah setelah klon tersedia. Dua langkah dalam proses ini adalah sebagai berikut:

1. Buat klon dengan menggunakan perintah restore-db-cluster-to- point-in-time CLI. Parameter yang Anda gunakan dengan perintah ini mengontrol jenis kapasitas dan detail lain dari klaster Amazon DocumentDB kosong (klon) yang sedang dibuat.
2. Buat instance Amazon DocumentDB untuk klon dengan menggunakan perintah create-db-instanceCLI untuk membuat ulang instans Amazon DocumentDB di cluster Amazon DocumentDB yang dipulihkan.

Perintah berikut mengasumsikan bahwa AWS CLI diatur dengan AWS Wilayah Anda sebagai default. Dengan pendekatan ini, Anda tidak perlu meneruskan nama `--region` di setiap perintah. Untuk informasi selengkapnya, lihat Mengonfigurasi AWS CLI. Anda juga dapat menentukan `--region` di setiap perintah CLI selanjutnya.

Membuat klon

Parameter tertentu yang Anda teruskan ke perintah CLI `restore-db-cluster-to-point-in-time` akan bervariasi. Apa yang Anda lewati tergantung pada jenis klon yang ingin Anda buat.

Gunakan prosedur berikut untuk membuat klon Amazon DocumentDB yang disediakan dari klaster Amazon DocumentDB yang disediakan.

Untuk membuat tiruan dari mode mesin yang sama dengan sumber cluster Amazon DocumentDB

- Gunakan perintah CLI `restore-db-cluster-to-point-in-time` dan tentukan nilai untuk parameter berikut:

- `--db-cluster-identifier` – Pilih nama yang bermakna untuk klon anda. Anda memberi nama klon saat Anda menggunakan perintah restore-db-cluster-to- point-in-time CLI.
- `--restore-type` – Gunakan `copy-on-write` untuk membuat klon dari klaster DB sumber. Tanpa parameter ini, `restore-db-cluster-to-point-in-time` mengembalikan cluster Amazon DocumentDB daripada membuat klon. Default untuk `restore-type` adalah`full-copy`.
- `--source-db-cluster-identifier`— Gunakan nama cluster Amazon DocumentDB sumber yang ingin Anda kloning.
- `--use-latest-restorable-time` – Nilai ini menunjuk ke data volume terbaru yang dapat dipulihkan untuk klon. Parameter ini diperlukan untuk`restore-type copy-on-write`, bagaimanapun, Anda tidak dapat menggunakan `restore-to-time parameter` dengan itu.

Contoh berikut membuat klon bernama `my-clone` dari klaster bernama `my-source-cluster`.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Perintah tersebut mengembalikan objek JSON yang berisi rincian dari klon. Periksa untuk memastikan bahwa klaster DB yang Anda kloning tersedia sebelum mencoba membuat instans DB untuk klon Anda. Untuk informasi selengkapnya, lihat Memeriksa status dan mendapatkan detail klon di bawah ini:

Memeriksa status dan mendapatkan detail klon

Anda dapat menggunakan perintah berikut untuk memeriksa status klaster DB kosong yang baru dibuat.

```
$ aws docdb describe-db-clusters --db-cluster-identifier my-clone --query '*[].
[Status]' --output text
```

Atau Anda dapat memperoleh status dan nilai-nilai lain yang Anda butuhkan untuk membuat instance DB untuk klon Anda dengan menggunakan AWS CLI kueri berikut:

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone \
  --query '*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}'
```

Untuk Windows:

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone ^
  --query "*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}"
```

Kueri ini menghasilkan output yang serupa dengan berikut ini:

```
[
  {
        "Status": "available",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
    }
]
```

Membuat instance Amazon DocumentDB untuk klon Anda

Gunakan perintah create-db-instanceCLI untuk membuat instance DB untuk klon Anda.

`--db-instance-class`Parameter ini digunakan hanya untuk klaster Amazon DocumentDB yang disediakan.

Untuk Linux, macOS, atau 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
```

Untuk 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 yang digunakan untuk kloning

Tabel berikut merangkum berbagai parameter yang digunakan `restore-db-cluster-to-point-in-time` untuk mengkloning cluster Amazon DocumentDB.

Parameter	Deskripsi
--source-db-cluster-identifier	Gunakan nama cluster Amazon DocumentDB sumber yang ingin Anda kloning.
--db-cluster-identifier	Gunakan parameter ini untuk memilih nama yang bermakna untuk klon anda. Anda menamai klon Anda dengan perintah `restore-db-cluster-to-point-in-time` . Kemudian, Anda meneruskan nama ini ke perintah `create-db-instance` .
--restore-type	Tentukan `copy-on-write` sebagai `--restore-type` untuk membuat klon dari cluster DB sumber daripada memulihkan sumber Amazon DocumentDB cluster.
--use-latest-restorable-time	Nilai ini menunjuk ke data volume terbaru yang dapat dipulihkan untuk klon.

Memahami toleransi kesalahan klaster Amazon DocumentDB

Klaster Amazon DocumentDB menurut desainnya toleran terhadap kesalahan. Setiap volume klaster mencakup beberapa Availability Zone dalam satu Wilayah AWS, dan setiap Availability Zone berisi salinan data volume cluster. Dengan fungsionalitas ini, klaster Anda dapat menoleransi kegagalan Availability Zone tanpa kehilangan data dan hanya gangguan layanan singkat.

Jika instans utama dalam sebuah klaster gagal, Amazon DocumentDB secara otomatis melakukan failover ke instans utama baru dengan salah satu dari dua cara berikut:

- Dengan mempromosikan replika Amazon DocumentDB yang ada ke instans utama baru yang dipilih berdasarkan pengaturan Tingkat Promosi dari setiap replika, lalu membuat pengganti untuk instans primer sebelumnya. Failover untuk instans replika biasanya membutuhkan waktu kurang dari 30 detik. Operasi baca dan tulis mungkin mengalami gangguan singkat selama periode ini. Untuk meningkatkan ketersediaan klaster Anda, kami sarankan Anda membuat setidaknya satu replika Amazon DocumentDB atau lebih di dua Availability Zone yang berbeda.
- Dengan membuat instans primer baru. Ini hanya terjadi jika Anda tidak memiliki instans replika di klaster Anda dan penyelesaiannya dapat memakan waktu beberapa menit.

Jika klaster memiliki satu atau beberapa replika Amazon DocumentDB, replika Amazon DocumentDB akan dipromosikan ke instans primer selama peristiwa kegagalan. Peristiwa failover mengakibatkan interupsi singkat, selama operasi baca dan tulis gagal dengan pengecualian. Namun, layanan biasanya dipulihkan dalam waktu kurang dari 120 detik, dan sering kali kurang dari 60 detik. Untuk meningkatkan ketersediaan klaster Anda, kami sarankan Anda membuat setidaknya satu replika Amazon DocumentDB atau lebih di dua Availability Zone yang berbeda.

Anda dapat membuat urutan tertentu agar replika Amazon DocumentDB Anda dipromosikan ke instans primer setelah kegagalan dengan menetapkan prioritas pada setiap replika. Prioritas berkisar dari 0 untuk prioritas tertinggi hingga 15 untuk prioritas terendah. Jika instans primer gagal, replika Amazon DocumentDB dengan prioritas tertinggi akan dipromosikan ke instans primer baru. Anda dapat mengubah prioritas replika Amazon DocumentDB kapan saja. Memodifikasi prioritas tidak memicu failover. Anda dapat menggunakan operasi `modify-db-instance` dengan parameter `--promotion-tier`. Untuk informasi lebih lanjut tentang menyesuaikan prioritas failover instans, lihat Failover Amazon DocumentDB.

Lebih dari satu replika Amazon DocumentDB dapat berbagi prioritas yang sama, dan menghasilkan tingkat promosi. Jika dua replika Amazon DocumentDB atau lebih berbagi prioritas yang sama, maka replika dengan ukuran terbesar akan dipromosikan menjadi yang primer. Jika dua replika Amazon DocumentDB atau lebih memiliki prioritas dan ukuran yang sama, replika arbitrer di tingkat promosi yang sama akan dipromosikan.

Jika klaster tidak berisi replika Amazon DocumentDB, instans primer akan dibuat ulang selama peristiwa kegagalan. Peristiwa failover mengakibatkan gangguan di mana selama operasi baca dan tulis gagal dengan pengecualian. Layanan dipulihkan ketika instans primer baru dibuat, yang

biasanya memakan waktu kurang dari 10 menit. Mempromosikan replika Amazon DocumentDB ke instans utama jauh lebih cepat daripada membuat instans utama baru.

Mengelola Instans Amazon DocumentDB

Topik berikut memberikan informasi untuk membantu Anda mengelola instans Amazon DocumentDB Anda. Topik-topik tersebut menyertakan detail tentang kelas dan status instans, serta cara membuat, menghapus, dan memodifikasi instans.

Topik

- Mengelola Kelas Instans
- Menentukan Status Instans
- Siklus hidup instance Amazon DocumentDB

Mengelola Kelas Instans

Kelas instans menentukan komputasi dan kapasitas memori dari instans Amazon DocumentDB (dengan kompatibilitas MongoDB). Kelas instans yang Anda butuhkan tergantung pada kebutuhan daya dan memori pemrosesan Anda.

Amazon DocumentDB mendukung keluarga kelas instans R4, R5, R6G, T3, dan T4G. Kelas-kelas ini adalah kelas instans generasi saat ini yang dioptimalkan untuk aplikasi yang memakan memori. Untuk spesifikasi pada kelas-kelas ini, lihat Spesifikasi Kelas Instans.

Topik

- Menentukan Kelas Instans
- Mengubah Kelas Instans
- Kelas Instans yang Didukung berdasarkan Wilayah
- Spesifikasi Kelas Instans

Menentukan Kelas Instans

Untuk menentukan kelas sebuah instance, Anda dapat menggunakan AWS Management Console atau `describe-db-instances` AWS CLI operasi.

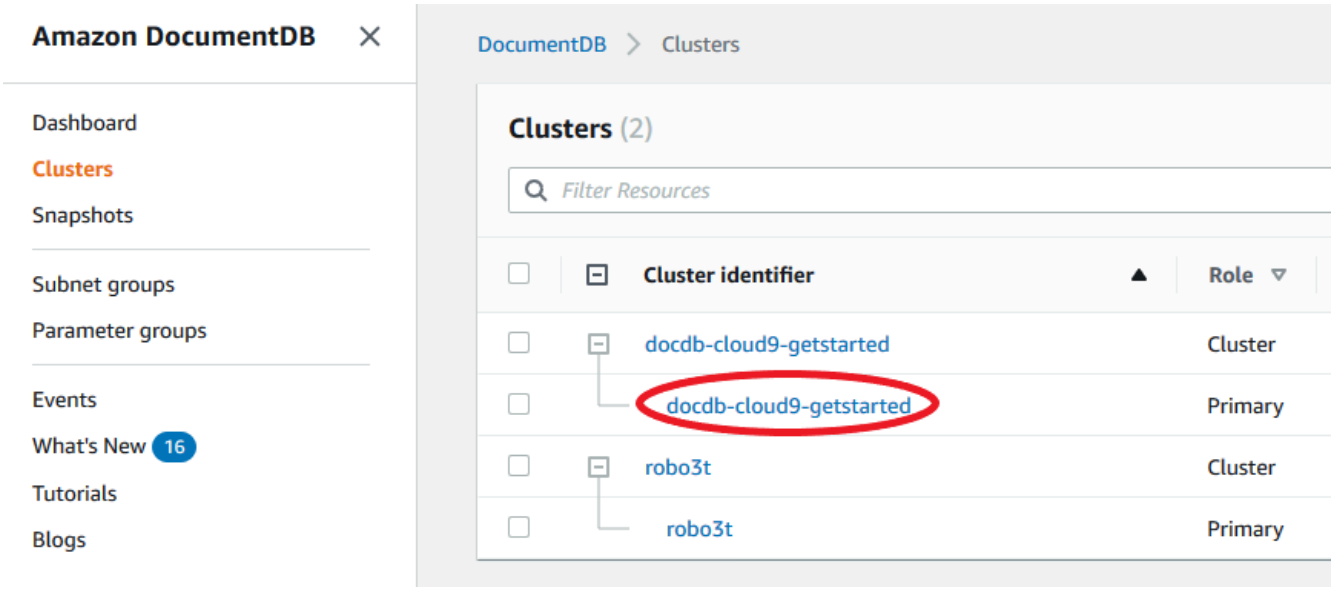
Using the AWS Management Console

Untuk menentukan kelas instans bagi instans klaster Anda, selesaikan langkah-langkah berikut di konsol.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster untuk menemukan instans yang Anda minati.

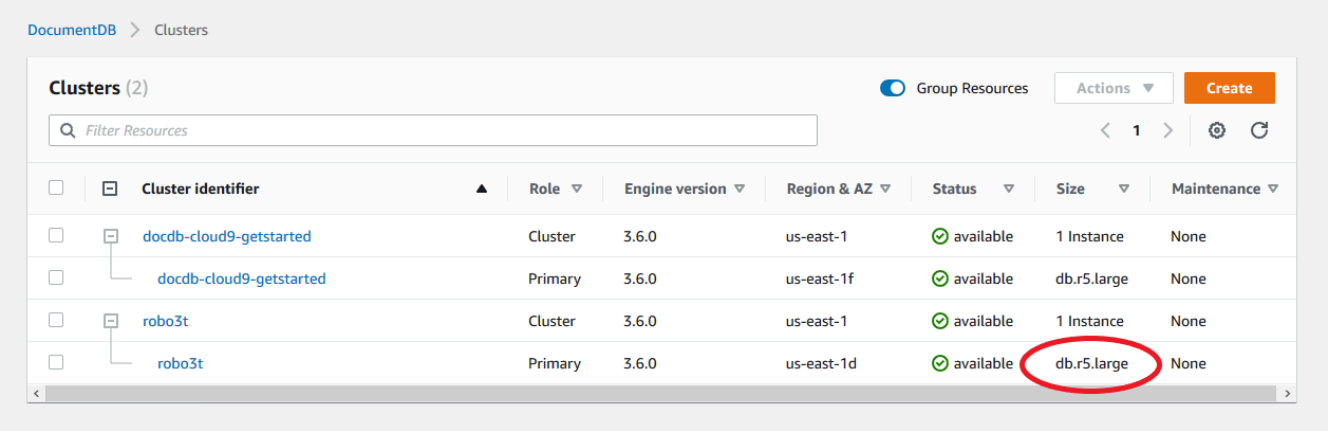
   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Dalam daftar instans, perluas klaster untuk menemukan instans Anda minati. Temukan instans yang Anda inginkan. Kemudian, lihat pada kolom Ukuran dari baris instans untuk melihat kelas instansnya.

   Pada gambar berikut, kelas instans untuk instans `robo3t` adalah `db.r5.4xlarge`.



Using the AWS CLI

Untuk menentukan kelas instance menggunakan AWS CLI, gunakan `describe-db-instances` operasi dengan parameter berikut.

- **`--db-instance-identifier`** — Opsional. Menentukan instans yang ingin Anda temukan kelas instansnya. Jika parameter ini dihilangkan, `describe-db-instances` mengembalikan deskripsi hingga 100 instans Anda.
- **`--query`** — Opsional. Menentukan anggota instans untuk disertakan dalam hasil. Jika parameter ini dihilangkan, semua anggota instans dikembalikan.

Example

Contoh berikut menemukan nama dan kelas instans untuk instans `sample-instance-1`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --db-instance-identifier sample-instance-1
```

Untuk Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --db-instance-identifier sample-instance-1
```

Output dari operasi ini terlihat seperti berikut.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ]
```

Example

Contoh berikut menemukan nama dan kelas instans hingga 100 instans Amazon DocumentDB.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --filter Name=engine,Values=docdb
```

Untuk Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --filter Name=engine,Values=docdb
```

Output dari operasi ini terlihat seperti berikut.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ],
    [
        "sample-instance-2",
        "db.r5.large"
    ],
    [
        "sample-instance-3",
        "db.r5.4xlarge"
    ],
    [
        "sample-instance-4",
        "db.r5.4xlarge"
    ]
]
```

Untuk informasi selengkapnya, lihat Menjelaskan instans Amazon DocumentDB.

Mengubah Kelas Instans

Anda dapat mengubah kelas instance instance Anda menggunakan AWS Management Console atau AWS CLI. Untuk informasi selengkapnya, lihat Memodifikasi instance Amazon DocumentDB.

Kelas Instans yang Didukung berdasarkan Wilayah

Amazon DocumentDB mendukung kelas instans berikut:

- R6G—Generasi terbaru dari instans yang dioptimalkan memori yang didukung oleh prosesor AWS Graviton2 berbasis ARM yang memberikan kinerja hingga 30% lebih baik dibandingkan instans R5 dengan biaya 5% lebih murah.
- R5—Instans yang dioptimalkan untuk memori yang memberikan kinerja hingga 100% lebih baik dibandingkan instans R4 dengan biaya instans yang sama.
- R4 — Generasi sebelumnya dari instans dengan memori yang dioptimalkan.
- T4G—Jenis instans tujuan umum burstable berbiaya rendah generasi terbaru yang ditenagai oleh prosesor AWS Graviton2 berbasis ARM yang memberikan tingkat kinerja CPU dasar, memberikan kinerja harga hingga 35% lebih baik dibandingkan instans T3 dan ideal untuk menjalankan aplikasi dengan penggunaan CPU moderat yang mengalami lonjakan sementara dalam penggunaan.
- T3—Jenis instans tujuan umum burstable dengan biaya rendah yang memberikan tingkat kinerja CPU dasar dengan kemampuan untuk meledakkan penggunaan CPU kapan saja selama diperlukan.

Untuk spesifikasi mendetail tentang kelas instans, lihat Spesifikasi Kelas Instans.

Sebuah kelas instans tertentu mungkin atau mungkin tidak didukung di Wilayah tertentu. Tabel berikut menentukan kelas instans yang didukung oleh Amazon DocumentDB di setiap Wilayah.

Kelas instans yang didukung berdasarkan Wilayah

Wilayah	R6G	R5	R4	T4G	T3
AS Timur (Ohio)	Didukung	Didukung	Didukung	Didukung	Didukung

Wilayah	R6G	R5	R4	T4G	T3
US East (N. Virginia)	Didukung	Didukung	Didukung	Didukung	Didukung
US West (Oregon)	Didukung	Didukung	Didukung	Didukung	Didukung
Amerika Selatan (Sao Paulo)	Didukung	Didukung		Didukung	Didukung
Asia Pasifik (Hong Kong)	Didukung	Didukung		Didukung	Didukung
Asia Pasifik (Hyderabad)		Didukung			Didukung
Asia Pasifik (Mumbai)	Didukung	Didukung		Didukung	Didukung
Asia Pasifik (Seoul)	Didukung	Didukung		Didukung	Didukung
Asia Pacific (Sydney)	Didukung	Didukung		Didukung	Didukung
Asia Pacific (Singapore)	Didukung	Didukung		Didukung	Didukung
Asia Pacific (Tokyo)	Didukung	Didukung		Didukung	Didukung
Kanada (Pusat)	Didukung	Didukung		Didukung	Didukung
Eropa (Frankfurt)	Didukung	Didukung		Didukung	Didukung
Eropa (Irlandia)	Didukung	Didukung	Didukung	Didukung	Didukung

Wilayah	R6G	R5	R4	T4G	T3
Eropa (London)	Didukung	Didukung		Didukung	Didukung
Eropa (Milan)	Didukung	Didukung		Didukung	Didukung
Eropa (Paris)	Didukung	Didukung		Didukung	Didukung
Timur Tengah (UEA)	Didukung	Didukung		Didukung	Didukung
Wilayah Tiongkok (Beijing)	Didukung	Didukung		Didukung	Didukung
Tiongkok (Ningxia)	Didukung	Didukung		Didukung	Didukung
AWS GovCloud (AS-Barat)	Didukung	Didukung		Didukung	Didukung
AWS GovCloud (AS-Timur)	Didukung	Didukung		Didukung	Didukung

Spesifikasi Kelas Instans

Tabel berikut memberikan detail dari kelas instans Amazon DocumentDB. Anda dapat menemukan penjelasan untuk setiap kolom tabel di bawah tabel.

Kelas instans Amazon DocumentDB yang didukung

Kelas instans	vCPU[1]	Memori (GiB) 2	Maks. temp. penyimpanan (GiB) 3	Bandwidth maksimum (Mbps)[4]	Performa jaringan[5]	Mesin Pendukung 6
R6G - Kelas Instance yang Dioptimalkan Memori Generasi Saat Ini berdasarkan Graviton2						

Kelas instans	vCPU[1]	Memori (GiB) 2	Maks. temp. penyimpanan (GiB) 3	Bandwidth maksimum (Mbps)[4]	Performa jaringan[5]	Mesin Pendukung 6
db.r6g.large	2	16	32	Hingga 4.750	Hingga 10 Gbps	4.0.0 dan 5.0.0
db.r6g.xlarge	4	32	63	Hingga 4.750	Hingga 10 Gbps	4.0.0 dan 5.0.0
db.r6g.2xlarge	8	64	126	Hingga 4.750	Hingga 10 Gbps	4.0.0 dan 5.0.0
db.r6g.4xlarge	16	128	252	4,750	Hingga 10 Gbps	4.0.0 dan 5.0.0
db.r6g.8xlarge	32	256	504	9.000	12 Gbps	4.0.0 dan 5.0.0
db.r6g.12xlarge	48	384	756	13.500	20 Gbps	4.0.0 dan 5.0.0
db.r6g.16xlarge	64	512	1008	19.000	25 Gbps	4.0.0 dan 5.0.0
R5 - Kelas Instance Dioptimalkan Memori Generasi Sebelumnya						
db.r5.large	2	16	31	Hingga 3.500	Hingga 10 Gbps	3.6.0, 4.0.0, dan 5.0.0
db.r5.xlarge	4	32	62	Hingga 3.500	Hingga 10 Gbps	3.6.0, 4.0.0, dan 5.0.0
db.r5.2xlarge	8	64	124	Hingga 3.500	Hingga 10 Gbps	3.6.0, 4.0.0, dan 5.0.0

Kelas instans	vCPU[1]	Memori (GiB) 2	Maks. temp. penyimpanan (GiB) 3	Bandwidth maksimum (Mbps)[4]	Performa jaringan[5]	Mesin Pendukung 6
db.r5.4xlarge	16	128	249	3.500	Hingga 10 Gbps	3.6.0, 4.0.0, dan 5.0.0
db.r5.8xlarge	32	256	504	6,800	10 Gbps	3.6.0, 4.0.0, dan 5.0.0
db.r5.12xlarge	48	384	748	7.000	10 Gbps	3.6.0, 4.0.0, dan 5.0.0
db.r5.16xlarge	64	512	1008	13.600	20 Gbps	3.6.0, 4.0.0, dan 5.0.0
db.r5.24xlarge	96	768	1500	14.000	25 Gbps	3.6.0, 4.0.0, dan 5.0.0
R4 - Kelas Instance Dioptimalkan Memori Generasi Sebelumnya						
db.r4.large	2	15.25	30	437	Hingga 10 Gbps	3.6.0 saja
db.r4.xlarge	4	30,5	60	875	Hingga 10 Gbps	3.6.0 saja
db.r4.2xlarge	8	61	120	875	Hingga 10 Gbps	3.6.0 saja
db.r4.4xlarge	16	122	240	875	Hingga 10 Gbps	3.6.0 saja

Kelas instans	vCPU[1]	Memori (GiB) 2	Maks. temp. penyimpanan (GiB) 3	Bandwidth maksimum (Mbps)[4]	Performa jaringan[5]	Mesin Pendukung 6
db.r4.8xlarge	32	244	480	875	10 Gbps	3.6.0 saja
db.r4.16xlarge	64	488	960	14.000	25 Gbps	3.6.0 saja
T4G — Kelas Instans Kinerja Burstable Generasi Terbaru berdasarkan Graviton2						
db.t4g.medium	2	4	8.13	Hingga 2.085	Hingga 5 Gbps	4.0.0 dan 5.0.0
T3 — Kelas Instans Kinerja Burstable Generasi Sebelumnya						
db.t3.medium	2	4	7.5	Hingga 1.536	Hingga 5 Gbps	3.6.0, 4.0.0, dan 5.0.0

1. vCPU — Jumlah unit pemrosesan pusat (CPU) virtual. CPU virtual adalah unit kapasitas yang dapat Anda gunakan untuk membandingkan kelas instans. Alih-alih membeli atau menyewa prosesor tertentu untuk digunakan selama beberapa bulan atau tahun, Anda menyewa kapasitas per jam. Tujuan kami adalah menyediakan jumlah kapasitas CPU yang konsisten, apa pun perangkat keras yang mendasarinya.
2. Memori (GiB) — RAM, dalam gigabyte, yang dialokasikan ke instans. Sering kali ada rasio yang konsisten antara memori dan vCPU.
3. Maks. temp. penyimpanan (GiB) - RAM, dalam gigabyte, yang dialokasikan ke instance untuk penyimpanan file sementara yang tidak persisten.
4. Bandwidth maks. (Mbps) — Bandwidth maksimum dalam megabit per detik. Bagi dengan 8 untuk mendapatkan throughput yang diharapkan dalam megabyte per detik.
5. Performa jaringan — Kecepatan jaringan relatif dengan kelas instans lainnya.
6. Mesin Pendukung - Enigines Amazon DocumentDB yang mendukung kelas instance.

Menentukan Status Instans

Untuk melihat status instans yang valid, artinya, dan cara menentukan status instans Anda, lihat Memantau Status instans Amazon DocumentDB.

Siklus hidup instance Amazon DocumentDB

Siklus hidup dari instans Amazon DocumentDB termasuk menciptakan, memodifikasi, memelihara dan meningkatkan, melakukan backup dan pemulihan, mem-boot ulang, dan menghapus instans. Bagian ini menyediakan informasi tentang cara menyelesaikan proses ini.

Topik

- Menambahkan instance Amazon DocumentDB ke cluster
- Menjelaskan instans Amazon DocumentDB
- Memodifikasi instance Amazon DocumentDB
- Mem-boot ulang instans Amazon DocumentDB
- Menghapus instans Amazon DocumentDB

Anda dapat membuat instance Amazon DocumentDB baru menggunakan AWS Management Console atau. AWS CLI Untuk menambahkan instans ke klaster, klaster harus dalam status tersedia. Anda tidak dapat menambahkan instans ke klaster yang dihentikan. Jika klaster berhenti, pertama mulai klaster, tunggu sampai klaster tersedia, kemudian tambahkan instans. Untuk informasi selengkapnya, lihat Menghentikan dan memulai cluster Amazon DocumentDB.

> **Note**
>
> Jika Anda membuat klaster Amazon DocumentDB menggunakan konsol, sebuah instans secara otomatis dibuat untuk Anda pada saat yang sama. Jika Anda ingin membuat instans tambahan, gunakan salah satu prosedur berikut.

Menambahkan instance Amazon DocumentDB ke cluster

Using the AWS Management Console

Gunakan prosedur berikut untuk membuat instans untuk klaster Anda menggunakan konsol Amazon DocumentDB.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Untuk memilih klaster yang ingin Anda tambahi instans, pilih tombol di sebelah kiri nama klaster.
4. Pilih Tindakan, dan kemudian pilih Tambahkan instans.
5. Di halaman Tambahkan instans ke: <cluster-name>, ulangi langkah berikut untuk setiap instans yang ingin Anda tambahkan ke klaster. Anda dapat memiliki hingga 15.

   a. Pengidentifikasi instans — Anda dapat memasukkan pengidentifikasi unik untuk instans ini atau mengizinkan Amazon DocumentDB menyediakan pengidentifikasi instans berdasarkan pengidentifikasi klaster.

      Kendala penamaan instans:

      - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
      - Karakter pertama harus berupa huruf.
      - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
      - Harus unik untuk semua instans di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS

   b. Kelas instans — Dari daftar tarik-turun, pilih tipe instans yang Anda inginkan untuk instans ini.

   c. Tingkat promosi — Dari daftar tarik-turun, pilih tingkat promosi untuk instans Anda atau pilih Tidak ada preferensi guna mengizinkan Amazon DocumentDB untuk mengatur tingkat promosi untuk instans Anda. Angka yang lebih rendah berarti prioritas yang lebih tinggi. Untuk informasi selengkapnya, lihat Mengontrol Target Failover.

   d. Untuk menambahkan lebih banyak instans, pilih Tambahkan instans tambahan dan ulangi langkah a, b, dan c.

6. Selesaikan operasi.

- Untuk menambahkan instans ke klaster Anda, pilih Buat.
- Untuk membatalkan operasi, pilih Batal.

Dibutuhkan beberapa menit untuk membuat sebuah instans. Anda dapat menggunakan konsol atau AWS CLI untuk melihat status instans. Untuk informasi selengkapnya, lihat Memantau status instans.

Using the AWS CLI

Gunakan `create-db-instance` AWS CLI operasi dengan parameter berikut untuk membuat instance utama untuk cluster Anda.

- **`--db-instance-class`** — Diperlukan. Kapasitas komputasi dan memori instans, misalnya, `db.m4.large`. Tidak semua kelas instans tersedia di semua Wilayah AWS.
- **`--db-instance-identifier`** — Diperlukan. Sebuah string yang mengidentifikasi instans.

  Kendala Penamaan Instans:
  - Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
  - Harus unik untuk semua instans di Amazon RDS, Neptunus, dan Amazon DocumentDB per, per Wilayah. Akun AWS
- **`--engine`** — Diperlukan. Harus berupa `docdb`.
- **`--availability-zone`** — Opsional. Availability Zone tempat Anda ingin membuat instans ini. Gunakan parameter ini untuk menemukan instans Anda di Availability Zone yang berbeda untuk meningkatkan toleransi kesalahan. Untuk informasi selengkapnya, lihat Ketersediaan Tinggi dan Replikasi Amazon DocumentDB.
- **`--promotion-tier`** — Opsional. Tingkat prioritas failover untuk instans ini. Harus antara 0 dan 15 dengan angka yang lebih rendah menjadi prioritas yang lebih tinggi. Untuk informasi selengkapnya, lihat Mengontrol Target Failover.

1. Pertama, tentukan Availability Zone tempat Anda dapat membuat instance.

   Jika Anda ingin menentukan Availability Zone sebelum membuat instans, jalankan perintah berikut untuk menentukan Availability Zone mana yang tersedia untuk klaster Amazon DocumentDB Anda.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
       --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
       --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

Output dari operasi ini terlihat seperti berikut.

```
[
    [
        "sample-cluster",
        [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ]
    ]
]
```

2. Kedua, tentukan kelas instance apa yang dapat Anda buat di Wilayah Anda.

   Untuk menentukan kelas instans mana yang tersedia untuk Anda di Wilayah Anda, jalankan perintah berikut. Dari output, pilih kelas instans untuk instans yang ingin Anda tambahkan ke klaster Amazon DocumentDB Anda.

   Untuk Linux, macOS, atau Unix:

   ```
   aws docdb describe-orderable-db-instance-options \
           --engine docdb \
           --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
   ```

   Untuk Windows:

   ```
   aws docdb describe-orderable-db-instance-options ^
           --engine docdb ^
   ```

```
        --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

Output dari operasi ini terlihat seperti berikut.

```
[
    "db.r5.16xlarge",
    "db.r5.2xlarge",
    "db.r5.4xlarge",
    "db.r5.8xlarge",
    "db.r5.large",
    "db.r5.xlarge"
]
```

3. Terakhir, tambahkan instance ke cluster Amazon DocumentDB Anda.

   Untuk menambahkan instans ke klaster Amazon DocumentDB Anda, jalankan perintah berikut..

   Untuk Linux, macOS, atau 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
```

   Untuk 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
```

   Output dari operasi ini terlihat seperti berikut.

```
{
    "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"
    }
}
```

Dibutuhkan beberapa menit untuk membuat instans. Anda dapat menggunakan konsol atau AWS CLI untuk melihat status instans. Untuk informasi selengkapnya, lihat Memantau Status instans Amazon DocumentDB.

Menjelaskan instans Amazon DocumentDB

Anda dapat menggunakan Konsol Manajemen Amazon DocumentDB atau AWS CLI untuk melihat detail seperti titik akhir koneksi, VPC grup keamanan, otoritas sertifikat, dan grup parameter yang berkaitan dengan instans Amazon DocumentDB Anda.

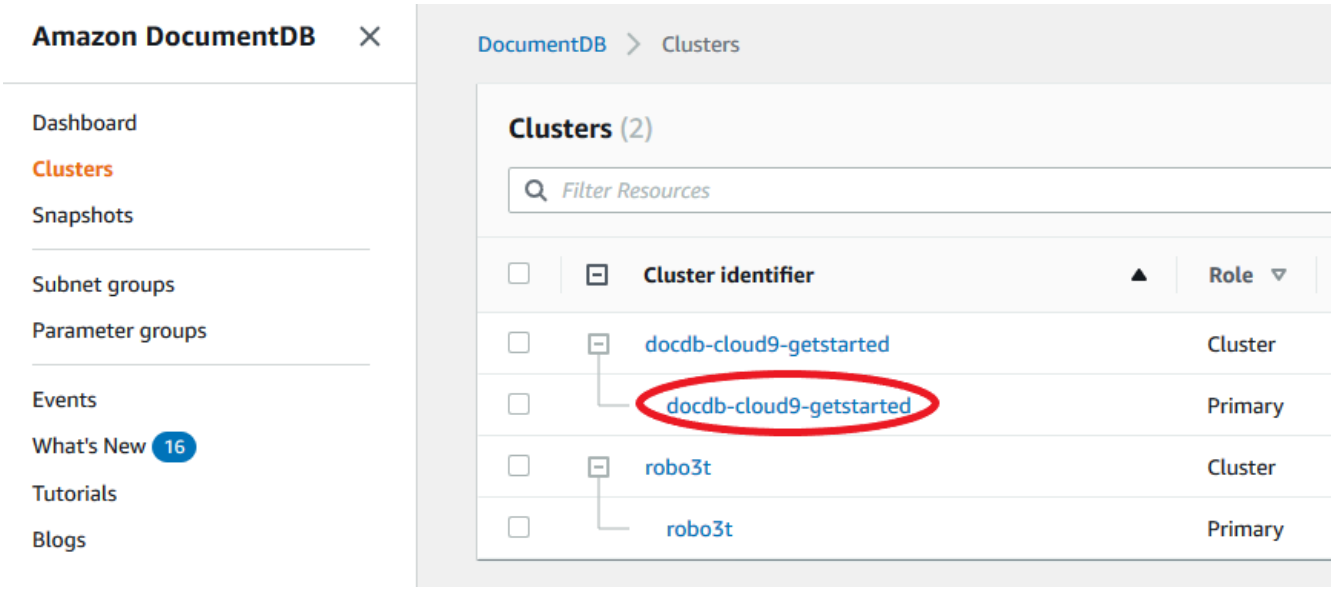
Using the AWS Management Console

Untuk melihat detail instans Anda menggunakan AWS Management Console, ikuti langkah-langkah di bawah ini.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Dalam daftar instans, pilih nama instans yang ingin Anda lihat detailnya. Informasi tentang instans diatur ke dalam pengelompokan berikut:

   - Ringkasan — Informasi umum tentang instans, termasuk versi mesin, kelas, status, dan setiap pemeliharaan tertunda.
   - Konektivitas & Keamanan — Bagian Hubungkan berisi daftar titik akhir koneksi untuk membuat sambungan ke instans ini dengan mongo shell atau dengan aplikasi. Bagian Grup Keamanan terdiri atas daftar grup keamanan yang terkait dengan instans ini dan ID VPC serta deskripsinya.
   - Konfigurasi — Bagian Detail berisi daftar konfigurasi dan status instans, termasuk Amazon Resource Name (ARN), titik akhir, peran, kelas, dan otoritas sertifikat instans. Ini juga berisi daftar pengaturan keamanan dan jaringan, serta informasi backup instans. Bagian Detail klaster berisi daftar detail klaster tempat instans ini berada. Bagian Instans klaster berisi daftar semua instans yang dimiliki klaster Anda dengan masing-masing peran instans dan status grup parameter klaster.

     > **Note**
     >
     > Anda dapat memodifikasi klaster yang terkait dengan instans Anda dengan memilih Modifikasi di samping header Detail klaster. Untuk informasi selengkapnya, lihat Memodifikasi cluster Amazon DocumentDB.

   - Monitoring —Metrik CloudWatch Log untuk contoh ini. Untuk informasi selengkapnya, lihat Memantau Amazon DocumentDB dengan CloudWatch.

- Peristiwa & tag — Bagian Peristiwa terbaru berisi daftar peristiwa terbaru untuk instans ini. Amazon DocumentDB menyimpan catatan peristiwa yang terkait dengan klaster, instans, snapshot, grup keamanan, dan grup parameter klaster Anda. Informasi ini mencakup tanggal, waktu, dan pesan yang terkait dengan setiap peristiwa. Bagian Tag berisi daftar tag yang dilampirkan ke klaster ini. Untuk informasi selengkapnya, lihat Penandaan Sumber Daya Amazon DocumentDB.

Using the AWS CLI

Untuk melihat detail instance Amazon DocumentDB Anda menggunakan, gunakan perintah seperti AWS CLI yang ditunjukkan pada contoh `describe-db-clusters` di bawah ini. Untuk informasi selengkapnya, lihat `DescribeDBInstances`di Referensi API Manajemen Sumber Daya Amazon DocumentDB.

> **Note**
>
> Untuk fitur pengelolaan tertentu seperti pengelolaan siklus hidup klaster dan instans, Amazon DocumentDB memanfaatkan teknologi operasi yang dibagikan dengan Amazon RDS. Parameter filter `filterName=engine,Values=docdb` hanya mengembalikan klaster Amazon DocumentDB.

1. Daftar semua instance Amazon DocumentDB.

   AWS CLI Kode berikut mencantumkan detail untuk semua instance Amazon DocumentDB di suatu wilayah.

   Untuk Linux, macOS, atau Unix:

   ```
   aws docdb describe-db-instances \
       --filter Name=engine,Values=docdb
   ```

   Untuk Windows:

   ```
   aws docdb describe-db-instances \
       --filter Name=engine,Values=docdb
   ```

2. Cantumkan semua detail untuk instance Amazon DocumentDB tertentu

Kode berikut mengeluarkan daftar detail untuk `sample-cluster-instance`. Termasuk parameter `--db-instance-identifier` dengan nama instans membatasi output ke informasi pada instans tertentu.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Untuk Windows:

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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"
        }
    ]
}
```

Memodifikasi instance Amazon DocumentDB

Anda dapat memodifikasi instans Amazon DocumentDB menggunakan instans atau AWS Management Console file. AWS CLI Untuk memodifikasi sebuah instans, instans harus dalam status tersedia. Anda tidak dapat memodifikasi instans yang dihentikan. Jika klaster berhenti, pertama mulai klaster, tunggu hingga instans tersedia, lalu buat modifikasi yang diinginkan. Untuk informasi selengkapnya, lihat Menghentikan dan memulai cluster Amazon DocumentDB.

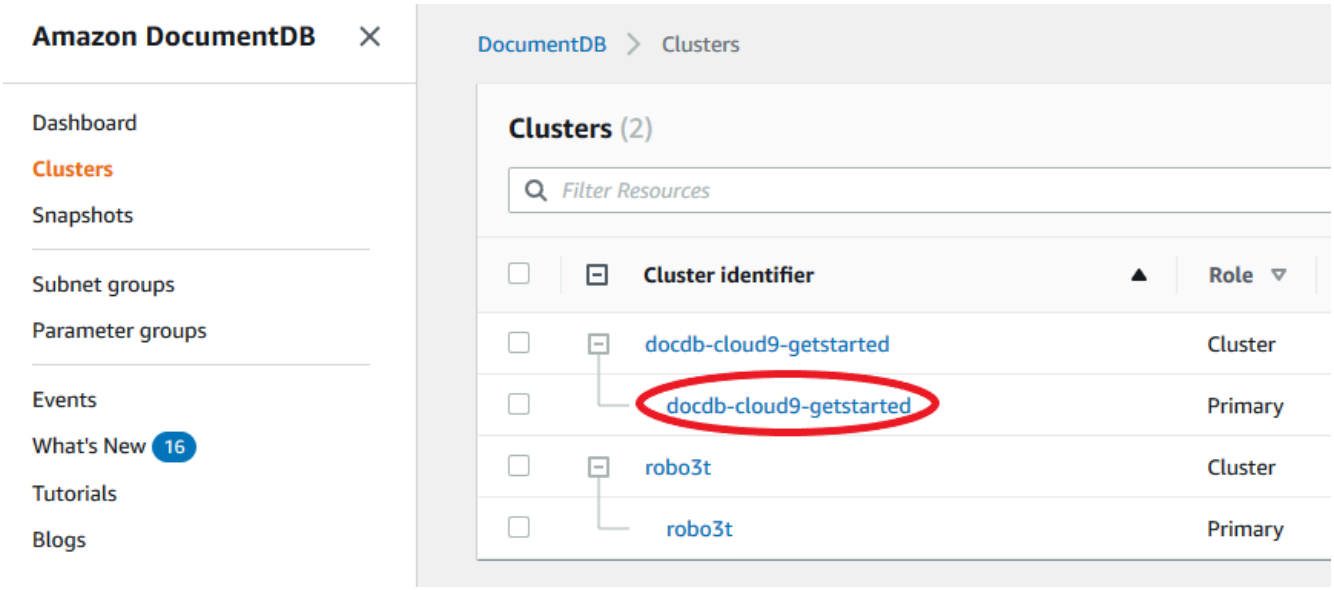
Using the AWS Management Console

Untuk memodifikasi instans Amazon DocumentDB tertentu menggunakan konsol, selesaikan langkah berikut.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Centang kotak di sebelah kiri instans yang ingin Anda modifikasi.
5. Pilih Tindakan, dan kemudian pilih Modifikasi.
6. Di panel Modifikasi instans: <instance-name>, buat perubahan yang Anda inginkan. Anda dapat membuat perubahan berikut:

   - Spesifikasi instans — Pengidentifikasi dan kelas instans. Kendala penamaan pengidentifikasi instans:
     - Instance identifier — Masukkan nama yang unik untuk semua instans yang dimiliki oleh Anda Akun AWS di wilayah saat ini. Pengidentifikasi instans harus berisi [1-63] karakter alfanumerik atau tanda hubung, memiliki huruf sebagai karakter pertama, dan tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berturut-turut.
     - Kelas instans — Dari menu tarik-turun, pilih kelas instans untuk instans Amazon DocumentDB Anda. Untuk informasi selengkapnya, lihat Mengelola Kelas Instans.
   - Otoritas sertifikat — Sertifikat server untuk instans ini. Untuk informasi selengkapnya, lihat Memperbarui Sertifikat TLS Amazon DocumentDB Anda.
   - Failover — Selama failover, instans dengan tingkat promosi tertinggi akan dipromosikan ke primer. Untuk informasi selengkapnya, lihat Failover Amazon DocumentDB.
   - Pemeliharaan — Jendela pemeliharaan di mana modifikasi atau patch yang tertunda diterapkan ke instans di klaster.
7. Setelah selesai, pilih Lanjutkan untuk melihat ringkasan perubahan Anda.
8. Setelah memverifikasi perubahan Anda, Anda dapat menerapkannya segera atau selama jendela pemeliharaan berikutnya di bawah Penjadwalan modifikasi. Pilih Modifikasi instans

untuk menyimpan perubahan Anda. Atau, Anda dapat memilih Batal untuk membuang perubahan Anda.

Diperlukan beberapa menit agar perubahan Anda diterapkan. Anda dapat menggunakan instans hanya ketika statusnya tersedia. Anda dapat memantau status instans menggunakan konsol atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status instans Amazon DocumentDB.

Using the AWS CLI

Untuk memodifikasi instance Amazon DocumentDB tertentu menggunakan AWS CLI, gunakan dengan parameter `modify-db-instance` berikut. Untuk informasi selengkapnya, lihat ModifyDBInstance. Kode berikut memodifikasi kelas instans menjadi `db.r5.large` untuk instans `sample-instance`.

Parameter

- **`--db-instance-identifier`** — Diperlukan. Pengidentifikasi untuk instans yang akan dimodifikasi.
- **`--db-instance-class`** — Opsional. Kapasitas komputasi dan memori baru dari instans; misalnya, `db.r5.large`. Tidak semua kelas instans tersedia di semua Wilayah AWS. Jika Anda memodifikasi kelas instans, pemadaman terjadi selama perubahan. Perubahan diterapkan selama jendela pemeliharaan berikutnya, kecuali `ApplyImmediately` ditentukan sebagai true untuk permintaan ini.
- **`--apply-immediately`** atau **`--no-apply-immediately`** — Opsional. Menentukan apakah modifikasi ini harus diterapkan segera atau menunggu hingga jendela pemeliharaan berikutnya. Jika parameter ini dihilangkan, modifikasi dilakukan selama jendela pemeliharaan berikutnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-instance \
       --db-instance-identifier sample-instance \
       --db-instance-class db.r5.large \
       --apply-immediately
```

Untuk Windows:

```
aws docdb modify-db-instance ^
```

```
    --db-instance-identifier sample-instance ^
    --db-instance-class db.r5.large ^
    --apply-immediately
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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"
            ]
        }
    ]
}
```

Diperlukan beberapa menit agar modifikasi Anda diterapkan. Anda dapat menggunakan instans hanya ketika statusnya tersedia. Anda dapat memantau status instans menggunakan AWS Management Console atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status instans Amazon DocumentDB.

Mem-boot ulang instans Amazon DocumentDB

Terkadang, Anda mungkin perlu mem-boot ulang instans Amazon DocumentDB Anda, biasanya karena alasan pemeliharaan. Jika Anda melakukan perubahan tertentu, seperti mengubah grup parameter klaster yang terkait dengan klaster, Anda harus mem-boot ulang instans di klaster agar

perubahan diterapkan. Anda dapat me-reboot instance tertentu menggunakan AWS Management Console atau AWS CLI.

Mem-boot ulang instans akan memulai ulang layanan mesin basis data. Mem-boot ulang menyebabkan pemadaman sementara, selama status instans diatur ke `rebooting`. Peristiwa Amazon DocumentDB dibuat saat boot ulang selesai.

Mem-boot ulang instans tidak mengakibatkan failover. Untuk melakukan failover cluster Amazon DocumentDB, AWS Management Console gunakan atau operasinya. AWS CLI `failover-db-cluster` Untuk informasi selengkapnya, lihat Failover Amazon DocumentDB.

Anda tidak dapat mem-boot ulang instans Anda jika tidak berada dalam status tersedia. Basis data Anda dapat tidak tersedia karena beberapa alasan, seperti modifikasi yang diminta sebelumnya, atau tindakan jendela pemeliharaan. Untuk informasi lebih lanjut tentang status instans, lihat Memantau Status instans Amazon DocumentDB.

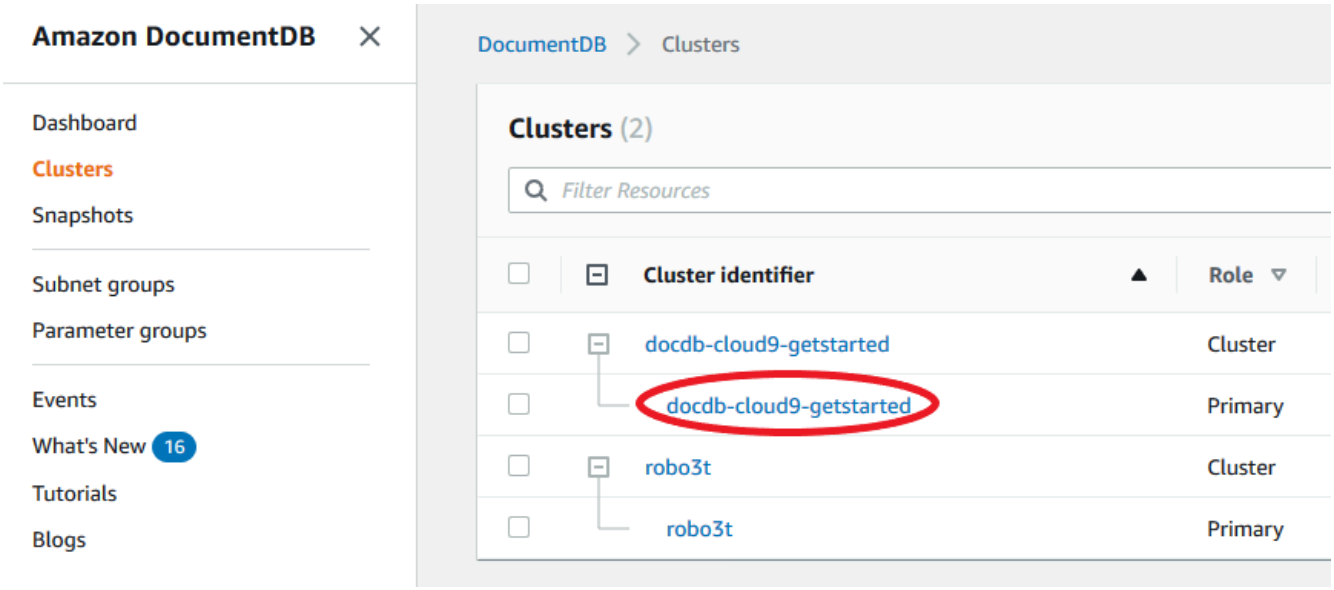
Using the AWS Management Console

Prosedur berikut mem-boot ulang instans yang Anda tentukan menggunakan konsol.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Centang kotak di sebelah kiri instans yang ingin Anda boot ulang.
5. Pilih Tindakan, pilih Boot ulang, lalu pilih Boot ulang untuk mengonfirmasi boot ulang Anda.

Diperlukan waktu beberapa menit agar instans di-boot ulang. Anda dapat menggunakan instans hanya ketika statusnya tersedia. Anda dapat memantau status instans menggunakan konsol atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status instans Amazon DocumentDB.

Using the AWS CLI

Untuk mem-boot ulang instans Amazon DocumentDB, gunakan operasi `reboot-db-instance` dengan parameter `--db-instance-identifier`. Parameter ini menentukan pengidentifikasi untuk instans yang akan di-boot ulang.

Kode berikut mem-boot ulang instans `sample-instance`.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-instance
```

Untuk Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-instance
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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"
        ]
    }
}
```

Diperlukan waktu beberapa menit agar instans di-boot ulang. Anda dapat menggunakan instans hanya ketika statusnya tersedia. Anda dapat memantau status instans menggunakan konsol atau AWS CLI. Untuk informasi selengkapnya, lihat Memantau Status instans Amazon DocumentDB.

Menghapus instans Amazon DocumentDB

Anda dapat menghapus instans Amazon DocumentDB Anda menggunakan salah satu atau AWS Management Console . AWS CLI Untuk menghapus sebuah instans, instans harus dalam status tersedia. Anda tidak dapat menghapus instans yang dihentikan. Jika klaster Amazon DocumentDB yang berisi instans Anda berhenti, pertama mulai klaster, tunggu instans menjadi tersedia, dan kemudian hapus instans tersebut. Untuk informasi selengkapnya, lihat Menghentikan dan memulai cluster Amazon DocumentDB.

> **Note**
>
> Amazon DocumentDB menyimpan semua data Anda dalam volume klaster. Data tetap ada dalam volume klaster tersebut, meskipun Anda menghapus semua instans dari klaster Anda. Jika Anda perlu mengakses data lagi, Anda dapat menambahkan instans ke klaster kapan saja dan melanjutkan dari bagian terakhir yang Anda tinggalkan.

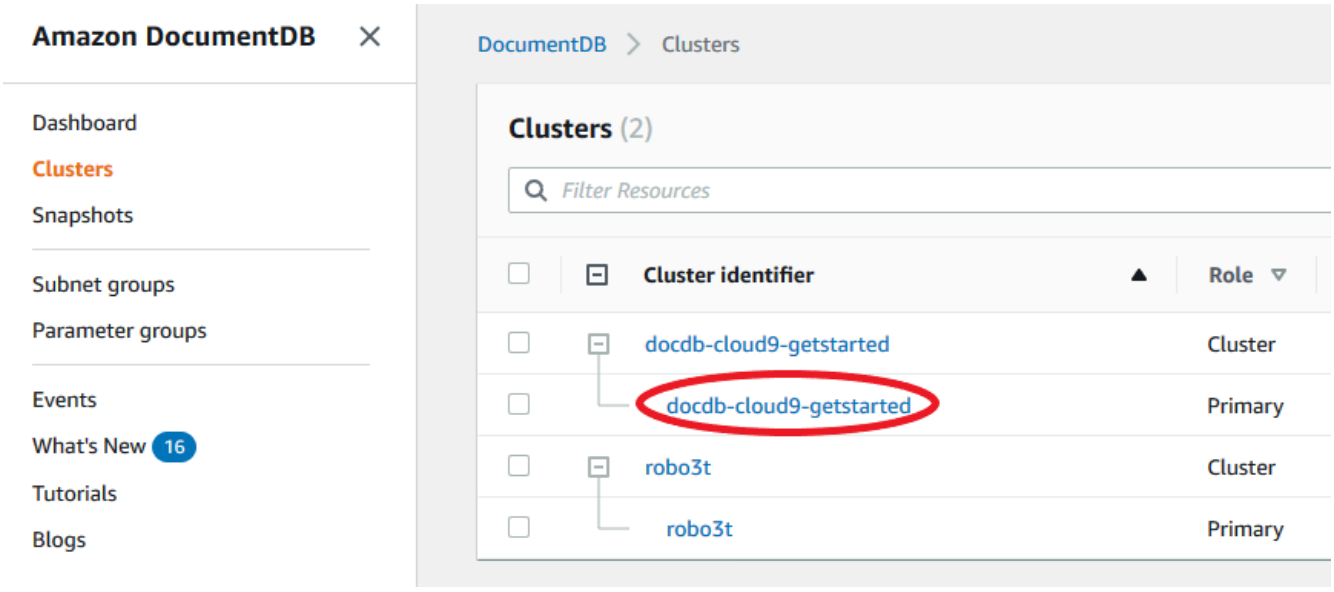
Using the AWS Management Console

Prosedur berikut menghapus instans Amazon DocumentDB yang ditentukan dengan menggunakan konsol.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Centang kotak di sebelah kiri instans yang ingin Anda hapus.
5. Pilih Tindakan, lalu pilih Hapus.

    1. Jika Anda menghapus instans terakhir di klaster Anda:

        - Buat snapshot cluster akhir? — Pilih Ya jika Anda ingin membuat snapshot akhir sebelum klaster dihapus. Jika tidak, pilih Tidak.

- Nama snapshot akhir — Jika Anda memilih untuk membuat snapshot akhir, masukkan pengidentifikasi snapshot klaster dari snapshot klaster baru yang dibuat.
- Hapus contoh? <instance-name> — Masukkan frasa hapus seluruh klaster ke dalam bidang untuk mengonfirmasi penghapusan.

2. Jika Anda tidak menghapus instans terakhir di klaster Anda:
   - Hapus contoh? <instance-name> — Masukkan frasa hapus saya ke dalam bidang untuk mengonfirmasi penghapusan.

6. Pilih Hapus untuk menghapus instans.

Diperlukan beberapa menit untuk menghapus instans. Untuk memantau status instans, lihat Memantau Status instans Amazon DocumentDB.

Using the AWS CLI

Prosedur berikut menghapus instans Amazon DocumentDB dengan menggunakan AWS CLI.

1. Pertama, tentukan berapa banyak instance yang ada di cluster Amazon DocumentDB Anda:

   Untuk menentukan berapa banyak instans di klaster Anda, jalankan perintah `describe-db-clusters`, sebagai berikut.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

   Output dari operasi ini terlihat seperti berikut.

```
[
    [
        "sample-cluster",
        [
            "sample-instance-1",
            "sample-instance-2"
        ]
    ]
]
```

2. Jika ada lebih dari satu instance di cluster Amazon DocumentDB Anda:

Untuk menghapus instans Amazon DocumentDB tertentu, gunakan perintah `delete-db-instance` dengan parameter `--db-instance-identifier`, seperti yang ditunjukkan di bawah ini. Diperlukan beberapa menit untuk menghapus instans. Untuk memantau status instans, lihat Memantau Status instans Amazon DocumentDB.

```
aws docdb delete-db-instance \
       --db-instance-identifier sample-instance-2
```

Output dari operasi ini akan terlihat seperti berikut.

```
{
    "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. Jika instance yang ingin Anda hapus adalah instance terakhir di cluster Amazon DocumentDB Anda:

   Jika Anda menghapus instans terakhir di klaster Amazon DocumentDB, Anda juga menghapus klaster tersebut serta snapshot otomatis dan backup terus menerus yang terkait dengan klaster tersebut.

   Untuk menghapus instans terakhir di klaster Anda, Anda dapat menghapus klaster dan secara opsional membuat snapshot akhir. Untuk informasi selengkapnya, lihat Menghapus cluster Amazon DocumentDB.

Perlindungan penghapusan

Menghapus instans terakhir dari klaster Amazon DocumentDB juga akan menghapus klaster tersebut, serta snapshot otomatis dan backup terus menerus yang terkait dengan klaster tersebut. Amazon DocumentDB memberlakukan perlindungan penghapusan untuk klaster apakah Anda melakukan operasi penghapusan menggunakan atau. AWS Management Console AWS CLI Jika perlindungan penghapusan diaktifkan, Anda tidak dapat menghapus klaster.

Untuk menghapus klaster yang mengaktifkan perlindungan penghapusan, Anda harus terlebih dahulu mengubah klaster dan menonaktifkan perlindungan penghapusan. Lihat informasi yang lebih lengkap di Menghapus cluster Amazon DocumentDB.

Mengelola grup subnet Amazon DocumentDB

Virtual private cloud (VPC) adalah jaringan virtual yang dikhususkan untuk Akun AWS Anda. VPC diisolasi secara logis dari jaringan virtual lain di AWS Cloud. Anda dapat meluncurkan sumber daya AWS Anda, seperti klaster Amazon DocumentDB, ke Amazon VPC Anda. Anda dapat menentukan kisaran alamat IP untuk VPC, menambahkan subnet, grup keamanan yang berkaitan, dan mengonfigurasi tabel rute.

Subnet adalah serangkaian alamat IP di Amazon VPC Anda. Anda dapat meluncurkan sumber daya AWS menjadi subnet yang ditetapkan. Gunakan subnet publik untuk sumber daya yang harus dihubungkan ke internet. Gunakan subnet privat untuk sumber daya yang tidak akan dihubungkan ke internet. Untuk informasi selengkapnya tentang subnet publik dan privat, lihat Basic VPC dan Subnet di Panduan Pengguna Amazon Virtual Private Cloud.

Grup subnet DB adalah kumpulan subnet yang Anda buat di VPC dan kemudian Anda tetapkan untuk klaster Anda. Grup subnet memungkinkan Anda untuk menentukan VPC tertentu saat membuat klaster. Jika Anda menggunakan grup subnet `default`, ini mencakup semua subnet di VPC.

Setiap grup subnet DB harus memiliki subnet di setidaknya dua Availability Zone di Wilayah tertentu. Saat membuat klaster DB dalam VPC, Anda harus memilih grup subnet DB. Amazon DocumentDB menggunakan grup subnet DB dan Availability Zone pilihan Anda untuk memilih subnet dan alamat IP di dalam subnet tersebut untuk dikaitkan dengan klaster Anda. Jika instans primer gagal, Amazon DocumentDB dapat mempromosikan instans replika yang sesuai menjadi primer baru. Hal ini kemudian dapat membuat instans replika baru menggunakan alamat IP subnet di mana primer sebelumnya terletak.

Saat Amazon DocumentDB membuat instans dalam VPC, jaringan ini menetapkan antarmuka jaringan ke klaster Anda menggunakan alamat IP yang dipilih dari grup subnet DB Anda. Kami sangat menyarankan agar Anda menggunakan nama DNS karena alamat IP dapat berubah selama failover. Untuk informasi selengkapnya, lihat Titik Akhir Amazon DocumentDB.

Untuk informasi tentang membuat VPC dan subnet Anda sendiri, lihat Bekerja dengan VPC dan Subnet di Panduan Pengguna Amazon Virtual Private Cloud.

Topik

- Membuat Grup Subnet Amazon DocumentDB
- Menjelaskan grup subnet Amazon DocumentDB
- Mengubah grup subnet Amazon DocumentDB
- Menghapus Grup Subnet Amazon DocumentDB

Membuat Grup Subnet Amazon DocumentDB

Saat membuat klaster Amazon DocumentDB, Anda harus memilih Amazon VPC dan grup subnet yang sesuai di dalam Amazon VPC tersebut untuk meluncurkan klaster Anda. Subnet menentukan availability zone dan kisaran IP di dalam availability zone yang ingin Anda gunakan untuk meluncurkan instans.

Grup subnet adalah serangkaian subnet (atau AZ) yang diberi nama yang mengizinkan Anda untuk menentukan availability zone yang ingin Anda gunakan untuk meluncurkan instans Amazon DocumentDB. Sebagai contoh, dalam klaster dengan tiga instans, disarankan bahwa setiap instans tersebut disediakan dalam AZ terpisah—sehingga mengoptimalkan ketersediaan tinggi. Dengan demikian, jika AZ tunggal gagal, maka hanya akan memengaruhi instans tunggal.

Saat ini, instans Amazon DocumentDB dapat disediakan dalam hingga tiga AZ. Bahkan jika grup subnet memiliki lebih dari tiga subnet, Anda hanya akan dapat menggunakan tiga subnet tersebut untuk membuat klaster Amazon DocumentDB. Oleh karena itu, kami menyarankan bahwa ketika Anda membuat grup subnet di mana Anda hanya memilih tiga subnet tempat Anda akan menyebarkan instans Anda.

Sebagai contoh: Klaster dibuat dan Amazon DocumentDB memilih AZ {1A, 1B, dan 1C}. Jika Anda mencoba untuk membuat instans di AZ {1D} panggilan API akan gagal. Namun demikian, jika Anda memilih untuk membuat instans, tanpa menentukan AZ tertentu, maka Amazon DocumentDB akan memilih AZ atas nama Anda. Amazon DocumentDB menggunakan algoritme untuk penyeimbangan beban instans di seluruh AZ untuk membantu Anda mencapai ketersediaan tinggi. Jika tiga instans

disediakan, secara default, mereka akan disediakan di tiga AZ dan tidak akan disediakan semuanya dalam AZ tunggal.

Praktik Terbaik

- Kecuali Anda memiliki alasan tertentu, selalu buat grup subnet dengan tiga subnet. Ini memastikan bahwa klaster dengan tiga instans atau lebih dapat mencapai ketersediaan yang lebih tinggi karena instans akan disediakan di tiga instans.
- Selalu sebarkan instans di beberapa AZ untuk mencapai ketersediaan yang tinggi. Jangan pernah menempatkan semua instans untuk klaster dalam AZ tunggal.
- Karena peristiwa failover dapat terjadi setiap saat, Anda tidak boleh berasumsi bahwa instans utama atau instans replika akan selalu berada di AZ tertentu.

Cara untuk membuat grup subnet

Anda dapat menggunakanAWS Management Console atauAWS CLI untuk membuat grup subnet Amazon DocumentDB:

Using the AWS Management Console

Gunakan langkah-langkah berikut ini untuk membuat grup subnet Amazon DocumentDB.

Untuk membuat grup subnet Amazon DocumentDB

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup subnet, kemudian pilih Buat.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Pada halaman Buat grup subnet:

   a. Di bagian Detail grup subnet:

      i. Nama—Masukkan nama yang bermakna untuk grup subnet.

ii. Deskripsi — Masukkan deskripsi untuk grup subnet.

b. Di bagian Tambahkan subnet:

i. VPC—Dalam daftar, pilih VPC untuk grup subnet ini.

ii. Lakukan salah satu dari berikut:

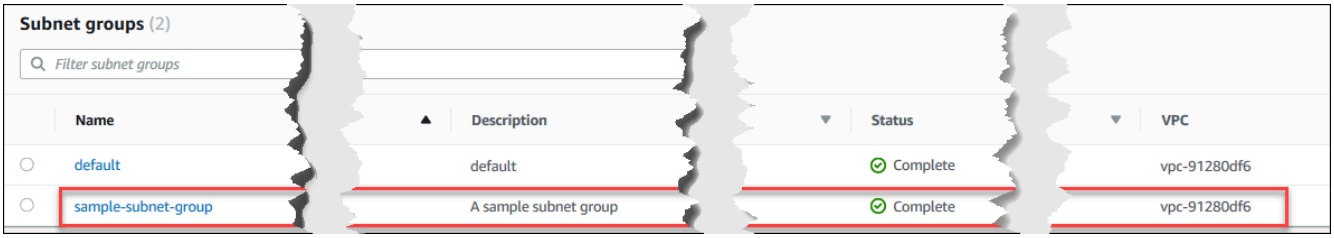
- Untuk memasukkan semua subnet dalam VPC yang dipilih, pilih Tambahkan semua subnet yang berkaitan dengan VPC ini.
- Untuk menentukan subnet untuk grup subnet ini, lakukan hal berikut ini untuk setiap Availability Zone tempat Anda ingin menyertakan subnet. Anda harus menyertakan setidaknya dua Availability Zone.

A. Availability Zone—Dalam daftar, pilih Availability Zone.

B. Subnet—Dalam daftar, pilih subnet dari Availability Zone yang yang dipilih untuk grup subnet ini.

C. Pilih Tambahkan subnet.

4. Pilih Create (Buat). Ketika grup subnet dibuat, maka akan dicantumkan dengan grup subnet Anda lainnya.



Using the AWS CLI

Sebelum Anda dapat membuat grup subnet menggunakan AWS CLI, Anda harus terlebih dahulu menentukan subnet mana yang tersedia. Jalankan operasi AWS CLI berikut ini untuk daftar Availability Zone dan subnet mereka.

Parameter:

- **`--db-subnet-group`**—Opsional. Menentukan grup subnet tertentu akan mencantumkan Availability Zone dan subnet untuk grup tersebut. Menghilangkan parameter ini akan mencantumkan Availability Zone dan subnet untuk semua grup subnet Anda. Menentukan grup subnet `default` akan mencantumkan semua subnet VPC.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default \
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

Untuk Windows:

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name default ^
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
[
    [
        "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"
            ]
        ]
    ]
]
```

Dengan menggunakan keluaran dari operasi sebelumnya, Anda dapat membuat grup subnet baru. Grup subnet baru harus mencakup subnet dari setidaknya dua Availability Zone.

Parameter:

- **`--db-subnet-group-name`**—Wajib. Nama untuk grup subnet ini.
- **`--db-subnet-group-description`**—Wajib. Deskripsi dari grup subnet ini.
- **`--subnet-ids`**—Wajib. Daftar subnet untuk disertakan dalam grup subnet ini. Contoh:`subnet-53ab3636`.
- --Tanda—Opsional. Daftar tanda (pasangan nilai kunci) untuk dilampirkan ke grup subnet ini.

Kode berikut ini membuat `sample-subnet-group` grup subnet dengan tiga subnet, `subnet-4e26d263`, `subnet-afc329f4`, dan `subnet-b3806e8f`.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
```

```
    "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"
    }
}
```

Menjelaskan grup subnet Amazon DocumentDB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mendapatkan detail dari grup subnet Amazon DocumentDB.

Using the AWS Management Console

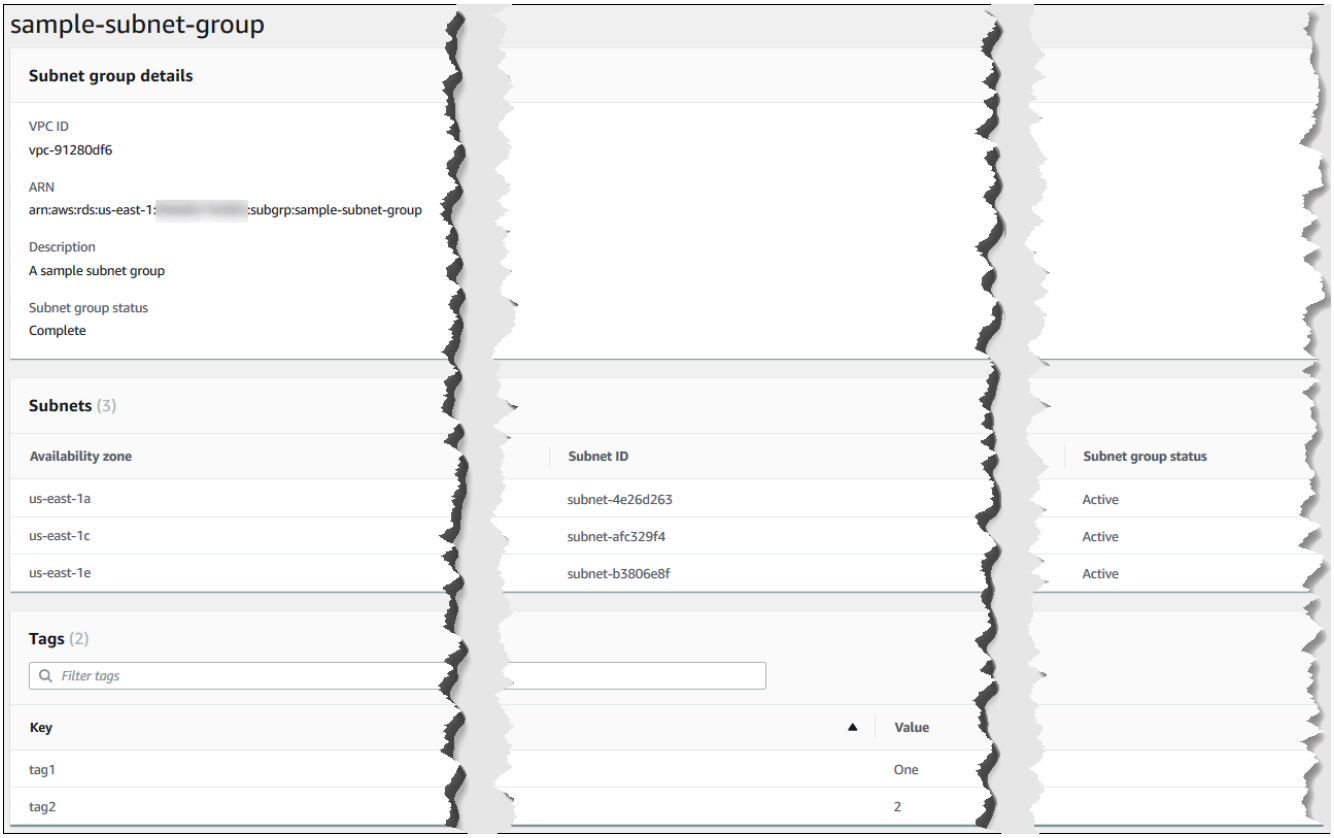
Prosedur berikut ini menunjukkan kepada Anda cara untuk mendapatkan detail grup subnet Amazon DocumentDB.

Untuk menemukan detail grup subnet

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Subnet.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Untuk melihat detail grup subnet, pilih nama grup subnet tersebut.



Using the AWS CLI

Untuk menemukan detail grup subnet Amazon DocumentDB, gunakan operasi `describe-db-subnet-groups` dengan parameter berikut ini.

Parameter

- `--db-subnet=group-name`—Opsional. Jika disertakan, detail untuk grup subnet yang diberi nama dicantumkan. Jika dihilangkan, detail untuk hingga 100 grup subnet dicantumkan.

Example

Kode berikut ini mencantumkan detail untuk grup subnet `sample-subnet-group` yang kita buat di bagian Membuat Grup Subnet Amazon DocumentDB.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name sample-subnet-group
```

Untuk Windows:

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name sample-subnet-group
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "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"
    }
}
```

Mengubah grup subnet Amazon DocumentDB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk memodifikasi deskripsi grup subnet atau untuk menambahkan atau menghapus subnet dari grup subnet Amazon DocumentDB. Namun demikian, Anda tidak dapat mengubah grup subnet `default`.

Using the AWS Management Console

Anda dapat menggunakan AWS Management Console untuk mengubah deskripsi grup subnet atau menambahkan dan menghapus subnet. Ingat bahwa setelah selesai, Anda harus memiliki setidaknya dua Availability Zone yang berkaitan dengan grup subnet Anda.

Untuk memodifikasi grup subnet Anda

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Subnet. Kemudian pilih tombol di sebelah kiri nama grup subnet. Ingat bahwa Anda tidak dapat memodifikasi grup subnet `default`.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Pilih Tindakan, dan kemudian pilih Modifikasi.
4. Deskripsi—Untuk mengubah deskripsi grup subnet Anda, masukkan deskripsi baru.
5. Untuk mengubah subnet yang berkaitan dengan grup subnet Anda, di bagian Tambahkan subnet, lakukan salah satu atau lebih dari hal berikut ini:
   - Untuk menghapus semua subnet dari grup subnet ini, pilih Hapus semua.
   - Untuk menghapus subnet tertentu dari grup subnet ini, pilih Hapus untuk setiap subnet yang ingin Anda hapus.
   - Untuk menambahkan semua subnet yang berkaitan dengan VPC ini, pilih Tambahkan semua subnet yang berkaitan dengan VPC ini.
   - Untuk menambahkan subnet tertentu ke grup subnet ini, lakukan hal berikut ini untuk setiap Availability Zone tempat Anda ingin menambahkan subnet.

     a. Availability zone—Dalam daftar, pilih Availability Zone baru.

     b. Subnet—Dalam daftar, pilih subnet dari Availability Zone yang yang dipilih untuk grup subnet ini.

     c. Pilih Tambahkan subnet.
6. Di kotak dialog konfirmasi:
   - Untuk membuat perubahan tersebut ke grup subnet, pilih Modifikasi.
   - Untuk menyimpan grup subnet agar tidak berubah, pilih Batal.

Using the AWS CLI

Anda dapat menggunakan AWS CLI untuk mengubah deskripsi grup subnet atau menambahkan dan menghapus subnet. Ingat bahwa setelah selesai, Anda harus memiliki setidaknya dua Availability Zone yang berkaitan dengan grup subnet Anda. Anda tidak dapat memodifikasi grup subnet `default`.

Parameter:

- `--db-subnet-group-name`—Wajib. Nama grup subnet Amazon DocumentDB yang Anda modifikasi.
- `--subnet-ids`—Wajib. Daftar semua subnet yang Anda inginkan dalam grup subnet setelah perubahan ini dilakukan.

> **Important**
>
> Setiap subnet yang saat ini berada dalam grup subnet yang tidak disertakan dalam daftar ini akan dihapus dari grup subnet. Jika Anda ingin menyimpan salah satu subnet yang saat ini dalam grup subnet, Anda harus menyertakannya dalam daftar ini.

- `--db-subnet-group-description`—Opsional. Deskripsi grup subnet.

Example

Kode berikut memodifikasi deskripsi dan menggantikan subnet yang ada dengan subnet `subnet-991cb8d0`, `subnet-53ab3636`, dan `subnet-29ab1025`.

Untuk Linux, macOS, atau 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"
```

Untuk 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"
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON). Perhatikan bahwa ini adalah grup subnet yang sama yang dibuat di bagian Membuat Grup Subnet Amazon DocumentDB. Namun demikian, subnet dalam grup subnet diganti dengan yang tercantum dalam operasi `modify-db-subnet-group`.

```
{
    "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"
    }
}
```

Menghapus Grup Subnet Amazon DocumentDB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menghapus grup subnet Amazon DocumentDB. Namun demikian, Anda tidak dapat menghapus grup subnet `default`.

Using the AWS Management Console

Anda dapat menggunakan AWS Management Console untuk menghapus grup subnet. Namun Anda tidak dapat menghapus grup subnet `default`.

Untuk menghapus grup subnet

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.

2. Di panel navigasi, pilih Grup Subnet. Kemudian pilih tombol di sebelah kiri nama grup subnet. Ingat bahwa Anda tidak dapat menghapus grup subnet `default`.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Pilih Actions (Tindakan), lalu pilih Delete (Hapus).
4. Di kotak dialog konfirmasi:
   - Untuk menghapus grup subnet, pilih Hapus.
   - Untuk menyinmpan grup subnet, pilih Batal.

Using the AWS CLI

Untuk menghapus grup subnet Amazon DocumentDB menggunakan AWS CLI, gunakan operasi `delete-db-subnet-group` dengan parameter berikut ini.

Parameter

- `--db-subnet-group-name`—Wajib. Nama grup subnet Amazon DocumentDB yang akan dihapus. Ingat bahwa Anda tidak dapat menghapus grup subnet `default`.

Example

Kode berikut ini akan menghapus `sample-subnet-group`.

Untuk Linux, macOS, atau Unix:

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

Untuk Windows:

```
aws docdb delete-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group
```

Operasi ini tidak menghasilkan keluaran.

Ketersediaan Tinggi dan Replikasi Amazon DocumentDB

Anda dapat mencapai ketersediaan tinggi dan membaca penskalaan di Amazon DocumentDB (dengan kompatibilitas MongoDB) dengan menggunakan instans replika. Klaster Amazon DocumentDB tunggal mendukung instans primer tunggal dan hingga 15 instans replika. Instans tersebut dapat didistribusikan di seluruh Availability Zone di dalam Wilayah klaster. Instans primer menerima lalu lintas baca dan tulis, dan instans replika hanya menerima permintaan baca.

Volume kluster dibuat dari beberapa salinan data untuk kluster. Namun demikian, data dalam volume klaster direpresentasikan sebagai volume tunggal yang logis ke instans primer dan replika Amazon DocumentDB dalam klaster. Instans replika pada akhirnya konsisten. Mereka mengembalikan hasil kueri dengan sedikit penundaan replika—biasanya kurang dari 100 milidetik setelah instans primer menulis pembaruan. Lager replika bervariasi tergantung pada laju perubahan basis data. Artinya, selama periode di mana sejumlah besar operasi tulis terjadi untuk basis data, Anda mungkin melihat peningkatan lag replika.

Penskalaan Baca

Replika Amazon DocumentDB berfungsi dengan baik untuk penskalaan baca karena didedikasikan sepenuhnya untuk membaca operasi pada volume klaster Anda. Operasi tulis dikelola oleh instans primer. Volume klaster dibagikan di semua instans di klaster Anda. Oleh karena itu, Anda tidak perlu mereplikasi dan mempertahankan salinan data untuk setiap replika Amazon DocumentDB.

Ketersediaan Yang Tinggi

Saat Anda membuat klaster Amazon DocumentDB, bergantung pada jumlah Availability Zone dalam grup subnet (harus ada setidaknya dua), Amazon DocumentDB menyediakan instans di seluruh Availability Zone. Ketika Anda membuat instans dalam klaster, Amazon DocumentDB secara otomatis mendistribusikan instans di seluruh Availability Zone dalam grup subnet untuk menyeimbangkan klaster. Tindakan ini juga mencegah semua instans diletakkan di Availability Zone yang sama.

Contoh

Untuk mengilustrasikan intinya, pertimbangkan contoh di mana Anda membuat cluster yang memiliki grup subnet dengan tiga Availability Zone: AZ1,AZ2, danAZ3.

Ketika instans pertama dalam klaster dibuat, ini adalah instans primer dan terletak di salah satu Availability Zone. Dalam contoh ini, instans berada di AZ1. Instans kedua yang dibuat adalah instans

replika dan terletak di salah satu dari dua Availability Zone lainnya, yaitu AZ2. Instans ketiga yang dibuat adalah instans replika dan terletak di Availability Zone yang tersisa, AZ3. Jika Anda membuat lebih banyak instans, mereka didistribusikan di seluruh Availability Zone sehingga Anda mencapai keseimbangan dalam klaster.

Jika terjadi kegagalan dalam instans primer (AZ1), failover dipicu, dan salah satu replika yang ada dipromosikan ke primer. Ketika primer yang lama pulih, maka menjadi replika di Availability Zone yang sama di mana replika tersebut disediakan (AZ1). Ketika Anda menyediakan tiga klaster instans, Amazon DocumentDB terus menjaga tiga instans klaster tersebut. Amazon DocumentDB secara otomatis menangani deteksi, failover, dan pemulihan kegagalan instans tanpa intervensi manual apa pun.

Ketika Amazon DocumentDB melakukan failover dan memulihkan instans, instans yang dipulihkan tetap di Availability Zone di mana instans tersebut disediakan pada awalnya. Namun demikian, peran instans mungkin berubah dari primer ke replika. Melakukan hal ini akan mencegah skenario di mana serangkaian failover dapat mengakibatkan semua instans berada di Availability Zone yang sama.

Anda dapat menentukan replika Amazon DocumentDB sebagai target failover. Artinya, jika instans primer gagal, replika Amazon DocumentDB yang ditentukan atau replika dari tingkat dipromosikan menjadi instans primer. Terdapat gangguan singkat selama permintaan baca dan tulis dibuat ke instans primer gagal dengan pengecualian. Jika klaster Amazon DocumentDB Anda tidak mencakup replika Amazon DocumentDB apa pun, ketika instans primer gagal, replika tersebut kembali dibuat. Mempromosikan replika Amazon DocumentDB jauh lebih cepat daripada membuat ulang instans primer.

Untuk skenario dengan ketersediaan tinggi, kami sarankan Anda membuat satu atau lebih replika Amazon DocumentDB. Replika tersebut harus berasal dari kelas instans yang sama dengan instans primer dan dalam Availability Zone yang berbeda untuk klaster Amazon DocumentDB Anda.

Untuk informasi selengkapnya, lihat yang berikut:

- Memahami toleransi kesalahan klaster Amazon DocumentDB
- Failover Amazon DocumentDB
  - Mengontrol Target Failover

Ketersediaan Tinggi dengan Cluster Global

Untuk ketersediaan tinggi di beberapaWilayah AWS, Anda dapat mengaturKlaster global Amazon DocumentDB. Setiap klaster global mencakup beberapa wilayah, yang memungkinkan pembacaan

global dengan latensi rendah dan pemulihan bencana dari pemadaman di seluruhWilayah AWS. Amazon DocumentDB secara otomatis menangani replikasi semua data dan pembaruan dari wilayah primer ke setiap wilayah sekunder.

Menambahkan Replika

Instans pertama yang ditambahkan ke klaster adalah instans primer. Setiap instans yang ditambahkan setelah instans pertama adalah instans replika. Klaster dapat memiliki hingga 15 instans replika di samping instans primer.

Ketika Anda membuat kluster menggunakan AWS Management Console, instans primer secara otomatis dibuat pada waktu yang sama. Untuk membuat replika pada saat yang sama seperti saat Anda membuat klaster dan instans primer, pilih Membuat replika di zona yang berbeda. Untuk informasi lebih lanjut, lihat langkah 4.d. di Membuat cluster Amazon DocumentDB. Untuk menambahkan lebih banyak replika untuk klaster Amazon DocumentDB, lihat Menambahkan instance Amazon DocumentDB ke cluster.

Saat menggunakan AWS CLI untuk membuat klaster Anda, Anda harus secara eksplisit membuat instans primer dan instans replika Anda. Untuk informasi selengkapnya, lihat bagian "Menggunakan AWS CLI" dalam topik berikut ini:

- Membuat cluster Amazon DocumentDB
- Menambahkan instance Amazon DocumentDB ke cluster

Failover Amazon DocumentDB

Dalam kasus tertentu, seperti jenis tertentu dari pemeliharaan yang direncanakan, atau peristiwa simpul primer yang tidak dimungkinkan node utama atau kegagalan Availability Zone, Amazon DocumentDB (dengan kompatibilitas MongoDB) mendeteksi kegagalan dan menggantikan simpul primer. Selama failover, penulisan waktu diminimalkan. Hal ini karena peran simpul primer tidak berhasil ke salah satu replika baca alih-alih harus membuat dan menyediakan simpul primer baru. Deteksi kegagalan dan promosi replika ini memastikan bahwa Anda dapat melanjutkan penulisan ke primer baru segera setelah promosi selesai.

Agar failover berfungsi, klaster Anda harus memiliki setidaknya dua instans — primer dan setidaknya satu instans replika.

Mengontrol Target Failover

Amazon DocumentDB menyediakan Anda dengan tingkatan failover sebagai sarana untuk mengontrol instans replika mana yang dipromosikan ke primer ketika terjadi failover.

Tingkatan Failover

Setiap instans replika berkaitan dengan tingkatan failover (0-15). Ketika failover terjadi akibat pemeliharaan atau kegagalan perangkat keras yang tidak dimungkinkan, instan utama tidak berhasil menjadi replika dengan prioritas tertinggi (tingkatan bernomor terendah). Jika beberapa replika memiliki tingkatan prioritas yang sama, primer tidak berhasil menjadi replika tingkatan tersebut yang paling dekat dalam ukuran primer sebelumnya.

Dengan menetapkan tingkatan failover untuk grup pilih replika menjadi `0` (prioritas tertinggi), Anda dapat memastikan bahwa failover akan mempromosikan salah satu replika dalam grup tersebut. Anda dapat secara efektif mencegah replika spesifik yang dipromosikan ke primer dalam kasus failover dengan menetapkan tingkatan prioritas rendah (nomor tinggi) untuk replika tersebut. Hal ini berguna dalam kasus di mana replika spesifik menerima penggunaan berat oleh aplikasi dan ketidakberhasilan untuk salah satu dari mereka akan berdampak negatif pada aplikasi kritis.

Anda dapat mengatur tingkatan failover instans ketika Anda membuatnya atau kemudian dengan memodifikasinya. Menetapkan tingkatan failover instans dengan memodifikasi instans tidak memicu failover. Untuk informasi selengkapnya lihat topik berikut:

- Menambahkan instance Amazon DocumentDB ke cluster
- Memodifikasi instance Amazon DocumentDB

Ketika secara manual menginisiasi failover, Anda memiliki dua cara untuk mengontrol instans replika yang dipromosikan ke primer: tingkatan failover seperti yang diuraiakn sebelumnya, dan parameter `--target-db-instance-identifier`.

--target-db-instance-identifier

Untuk pengujian, Anda dapat memaksa peristiwa failover menggunakan operasi `failover-db-cluster`. Anda dapat menggunakan parameter `--target-db-instance-identifier` untuk menentukan replika mana yang akan dipromosikan ke primer. Menggunakan parameter `--target-db-instance-identifier` akan menggantikan tingkatan prioritas failover. Jika Anda tidak menentukan parameter `--target-db-instance-identifier`, failover primer adalah sesuai dengan tingkatan prioritas failover.

Apa yang Terjadi Selama Failover

Failover secara otomatis ditangani oleh Amazon DocumentDB sehingga aplikasi Anda dapat melanjutkan operasi basis data secepat mungkin tanpa intervensi administratif.

- Jika Anda memiliki instans replika Amazon DocumentDB di Availability Zone yang sama atau berbeda saat gagal: Amazon DocumentDB membalik catatan nama kanonik (CNAME) agar instans Anda menunjukkan replika sehat, yang, pada gilirannya, dipromosikan menjadi primer baru. Failover biasanya selesai dalam waktu 30 detik dari awal sampai akhir.
- Jika Anda tidak memiliki instans replika Amazon DocumentDB (misalnya, klaster instans tunggal): Amazon DocumentDB akan mencoba membuat instans baru di Availability Zone yang sama dengan instans asli. Penggantian instans asli ini dilakukan atas dasar upaya terbaik dan mungkin tidak berhasil jika, sebagai contoh, terdapat masalah yang secara luas memengaruhi Availability Zone.

Aplikasi Anda harus mencoba kembali koneksi basis data dalam peristiwa kehilangan koneksi.

Failover Pengujian

Failover untuk klaster mempromosikan salah satu replika Amazon DocumentDB (instans baca-saja) di klaster menjadi instans primer (penulis klaster).

Ketika instans primer gagal, Amazon DocumentDB secara otomatis melakukan failover ke replika Amazon DocumentDB, jika ada. Anda dapat memaksa failover saat Anda ingin menyimulasikan kegagalan instans primer untuk pengujian. Setiap instans dalam klaster memiliki alamat titik akhir sendiri. Oleh karena itu, Anda perlu membersihkan dan membentuk kembali koneksi yang sudah ada yang menggunakan titik akhir tersebut yang ditujukan saat failover selesai.

Untuk memaksa failover, gunakan operasi `failover-db-cluster` dengan parameter tersebut.

- `--db-cluster-identifier`—Wajib. Nama kluster yang akan di-failover.
- `--target-db-instance-identifier`—Opsional. Nama instans yang akan dipromosikan ke instans primer.

Example

Operasi berikut ini memaksa failover kluster `sample-cluster`. Operasi tersebut tidak menentukan instans mana yang akan menjadi instans primer baru, sehingga Amazon DocumentDB memilih instans sesuai dengan prioritas tingkatan failover.

Untuk Linux, macOS, atau Unix:

```
aws docdb failover-db-cluster \
   --db-cluster-identifier sample-cluster
```

Untuk Windows:

```
aws docdb failover-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Operasi berikut ini memaksa failover klaster `sample-cluster`, yang menentukan bahwa `sample-cluster-instance` akan dipromosikan menjadi peran primer. (Perhatikan `"IsClusterWriter": true` dalam keluaran.)

Untuk Linux, macOS, atau Unix:

```
aws docdb failover-db-cluster \
   --db-cluster-identifier sample-cluster \
   --target-db-instance-identifier sample-cluster-instance
```

Untuk Windows:

```
aws docdb failover-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --target-db-instance-identifier sample-cluster-instance
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "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"
    }
}
```

Lag Replikasi

Lag replikasi biasanya 50ms atau kurang. Alasan paling umum untuk peningkatan lag replika adalah:

- Tingkat tulis yang tinggi pada primer yang menyebabkan replika baca jatuh di belakang primer.
- Pertikaian pada replika baca antara kueri yang berjalan lama (misalnya, pindaian sekuensial besar, kueri agregasi) dan replikasi tulis yang masuk.
- Jumlah kueri bersamaan yang sangat besar pada replika baca.

Untuk meminimalkan lag replikasi, cobalah teknik pemecahan masalah ini:

- Jika Anda memiliki tingkat tulis tinggi atau utilisasi CPU yang tinggi, kami sarankan Anda meningkatkan instans di klaster Anda.
- Jika terdapat kueri yang berjalan lama pada replika baca Anda, dan sangat sering terdapat pembaruan untuk dokumen yang dikuerikan, pertimbangkan mengubah kueri yang berjalan lama Anda, atau jalankan mereka terhadap replika primer/tulis untuk menghindari perdebatan pada replika baca.
- Jika terdapat jumlah kueri bersamaan yang sangat besar atau utilisasi CPU yang tinggi hanya pada replika baca, pilihan lainnya adalah untuk menskalakan keluar jumlah replika baca untuk menyebarkan beban kerja.
- Karena lag replikasi adalah hasil dari throughput tulis yang tinggi dan kueri yang berjalan lama, kami sarankan untuk memecahkan masalah lag replikasi dengan memanfaatkan metrik DBClusterReplicaLagMaximum CW dalam kombinasi dengan logger kueri yang lambat dan metrik `WriteThroughput`/`WriteIOPS`.

Secara umum, kami menyarankan bahwa semua replika Anda adalah dari tipe instans yang sama, sehingga failover klaster tidak akan menyebabkan penurunan performa.

Jika Anda memilih antara menaikkan skala dan menskalakan keluar (misalnya enam instans yang lebih kecil vs tiga instans yang lebih besar), kami umumnya merekomendasikan untuk pertama-tama mencoba menaikkan skala (instans yang lebih besar) sebelum menskalakan keluar, karena Anda akan mendapatkan cache buffer yang lebih besar per instans DB.

Secara proaktif, Anda harus mengatur alarm lag replikasi dan mengatur ambang batasnya untuk nilai yang Anda rasa merupakan batas atas untuk seberapa jauh dapat meninggalkan (atau “menahan”) data Anda pada instans replika sebelum mulai memengaruhi fungsionalitas aplikasi Anda. Secara umum, kami akan menyarankan bahwa ambang batas lag replikasi terlampaui untuk beberapa titik data sebelum menciptakan alarm, akibat beban kerja sementara.

> **Note**
>
> Selain itu, kami sarankan Anda mengatur alarm lainnya untuk lag replikasi yang melebihi 10 detik. Jika Anda melampaui ambang batas ini untuk beberapa titik data, kami sarankan Anda menaikkan skala instans Anda atau mengurangi throughput tulis Anda pada instans primer.

Mengelola indeks Amazon DocumentDB

Pembuatan indeks Amazon DocumentDB

Membangun indeks di Amazon DocumentDB memerlukan sejumlah keputusan yang harus dibuat:

- Seberapa cepat itu harus diselesaikan?
- Bisakah koleksi tidak dapat diakses saat build sedang berlangsung?
- Berapa banyak daya komputasi instans yang dapat dialokasikan ke build?
- Jenis indeks apa yang harus dibuat?

Bagian ini membantu Anda menjawab pertanyaan-pertanyaan ini dan memberikan perintah dan contoh pemantauan untuk membuat dan indeks Amazon DocumentDB pada koleksi cluster berbasis instans Anda.

Pedoman

Pedoman berikut mencakup batasan dasar dan pengorbanan konfigurasi saat membuat indeks baru:

- Dukungan versi Amazon DocumentDB - Meskipun pengindeksan pekerja tunggal didukung di semua versi Amazon DocumentDB, beberapa pengindeksan pekerja hanya didukung di Amazon DocumentDB versi 4.0 dan 5.0.
- Trade-off kinerja - Meningkatkan jumlah pekerja dalam proses pembuatan indeks meningkatkan pemanfaatan CPU dan membaca IO pada instance utama database Amazon DocumentDB Anda.

Sumber daya yang dibutuhkan untuk membuat indeks baru tidak akan tersedia untuk beban kerja Anda yang sedang berjalan.

- Cluster elastis - Pengindeksan paralel tidak didukung pada cluster elastis Amazon DocumentDB.
- Pekerja maksimum - Jumlah maksimum pekerja yang dapat Anda konfigurasi tergantung pada ukuran instance utama Anda di cluster database Anda. Ini adalah setengah dari jumlah total vCPU pada instance utama cluster database Anda. Misalnya, Anda dapat menjalankan maksimal 32 pekerja pada instance db.r6g.16xlarge yang memiliki 64 vCPU.

  > **Note**
  >
  > Pekerja paralel tidak didukung pada kelas instance 2xlarge dan lebih rendah.

- Pekerja minimum - Jumlah minimum pekerja yang dapat Anda konfigurasi adalah satu. Pengaturan default untuk pembuatan indeks pada cluster berbasis instance adalah dua pekerja. Namun, Anda dapat mengurangi jumlah pekerja menjadi satu dengan menggunakan opsi “utas pekerja”. Ini akan menjalankan proses dengan satu pekerja.
- Kompresi indeks - Amazon DocumentDB tidak mendukung kompresi indeks. Ukuran data untuk indeks mungkin lebih besar daripada saat Anda menggunakan opsi lain.
- Mengindeks beberapa koleksi - Setengah dari vCPU pada instance utama cluster database Anda dapat digunakan untuk pekerja yang dikonfigurasi yang melakukan pembuatan indeks pada beberapa koleksi.
- Jenis indeks - Lihat posting blog ini untuk penjelasan lengkap tentang jenis indeks yang didukung di Amazon DocumentDB.

Memulai

Untuk memulai pembuatan indeks pada koleksi, gunakan `createIndexes` perintah. Secara default, perintah akan menjalankan dua pekerja paralel yang meningkatkan kecepatan proses pembuatan indeks sebanyak dua kali.

Misalnya, proses perintah berikut menunjukkan cara membuat indeks untuk bidang “user_name” dalam dokumen dan meningkatkan kecepatan proses pengindeksan menjadi empat pekerja:

1. Buat indeks menggunakan dua pekerja paralel di cluster:

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
 "name":"username_idx"}]})
```

2. Untuk mengoptimalkan kecepatan proses pembuatan indeks, Anda dapat menentukan jumlah pekerja dengan menggunakan opsi “thread pekerja” (`"workers":<number>`) dalam `db.runCommand createIndexes` perintah.

   Tingkatkan kecepatan proses menjadi empat pekerja paralel:

   ```
   db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
    "name":"username_idx", "workers":4}]})
   ```

   > **Note**
   >
   > Semakin tinggi jumlah pekerja, semakin cepat pembuatan indeks berlangsung. Namun, semakin tinggi jumlah pekerja meningkat, semakin tinggi beban meningkat pada vCPU dan membaca IO dari instance utama Anda. Pastikan klaster Anda cukup disediakan untuk menangani beban yang meningkat tanpa menurunkan beban kerja lainnya.

Status kemajuan pengindeksan

Proses pembuatan indeks bekerja dengan menginisialisasi, memindai koleksi, menyortir kunci, dan, akhirnya, memasukkan kunci melalui pembuat indeks. Prosesnya memiliki hingga enam tahap saat Anda menjalankannya di latar depan, dan hingga sembilan tahap saat Anda menjalankannya di latar belakang. Anda dapat melihat metrik status seperti penyelesaian persentase, jumlah total blok penyimpanan yang dipindai, kunci yang diurutkan, dan kunci yang disisipkan berdasarkan tahap demi tahap.

Pantau kemajuan proses pengindeksan dengan menggunakan `db.currentOp()` perintah di shell mongo. Penyelesaian 100% dari tahap terakhir menunjukkan bahwa semua indeks telah berhasil dibuat:

```
db.currentOp({"command.createIndexes": { $exists : true } })
```

Jenis build indeks

Empat jenis build indeks adalah:

- Foreground - Build indeks latar depan memblokir semua operasi database lainnya sampai indeks dibuat. Build latar depan Amazon DocumentDB terdiri dari lima tahap.

- Foreground (unique) - Build indeks latar depan dokumen tunggal (unik) memblokir operasi database lain seperti build latar depan biasa. Berbeda dengan build latar depan dasar, build unik menggunakan tahap tambahan (kunci pengurutan 2) untuk mencari kunci duplikat. Build latar depan (unik) terdiri dari enam tahap.
- Latar Belakang - Build indeks latar belakang memungkinkan operasi database lainnya berjalan di latar depan saat indeks sedang dibuat. Build latar belakang Amazon DocumentDB terdiri dari delapan tahap.
- Latar Belakang (unik) - Build indeks latar belakang dokumen tunggal (unik) memungkinkan operasi database lainnya berjalan di latar depan saat indeks sedang dibuat. Berbeda dengan build latar belakang dasar, build unik menggunakan tahap tambahan (kunci pengurutan 2) untuk mencari kunci duplikat. Build latar belakang (unik) terdiri dari sembilan tahap.

Tahapan pembuatan indeks

Stage	Latar depan	Latar depan (unik)	Latar Belakang	Latar Belakang (unik)
Inisialisasi	1	1	1	1
indeks bangunan: menginisialisasi	2	2	2	2
indeks bangunan: koleksi pemindaian	3	3	3	3
indeks bangunan: kunci penyortiran 1	4	4	4	4
indeks bangunan: kunci penyortiran 2		5		5

Stage	Latar depan	Latar depan (unik)	Latar Belakang	Latar Belakang (unik)
indeks bangunan: memasukkan kunci	5	6	5	6
memvalidasi: indeks pemindaian			6	7
memvalidasi: menyortir tupel			7	8
memvalidasi: koleksi pemindaian			8	9

- inisialisasi - createIndex sedang mempersiapkan pembuat indeks. Fase ini harus sangat singkat.
- indeks bangunan: menginisialisasi - Pembangun indeks sedang mempersiapkan untuk membuat indeks. Fase ini harus sangat singkat.
- indeks bangunan: koleksi pemindaian - Pembuat indeks melakukan pemindaian koleksi untuk mengumpulkan kunci indeks. Satuan ukuran adalah “blok”.

> **Note**
>
> Jika lebih dari satu pekerja dikonfigurasi untuk build indeks, itu akan ditampilkan di tahap ini. Tahap “pengumpulan pemindaian” adalah satu-satunya tahap yang menggunakan banyak pekerja selama proses pembuatan indeks. Semua tahapan lainnya akan menampilkan satu pekerja.

- indeks bangunan: kunci penyortiran 1 - Pembuat indeks menyortir kunci indeks yang dikumpulkan. Satuan ukuran adalah “kunci”.
- indeks bangunan: kunci penyortiran 2 - Pembuat indeks menyortir kunci indeks yang dikumpulkan yang sesuai dengan tupel mati. Fase ini hanya ada untuk bangunan indeks yang unik. Satuan ukuran adalah “kunci”.

- indeks bangunan: memasukkan kunci - Pembuat indeks memasukkan kunci indeks ke dalam indeks baru. Satuan ukuran adalah “kunci”.
- memvalidasi: indeks pemindaian - createIndex memindai indeks untuk menemukan kunci yang perlu divalidasi. Satuan ukuran adalah “blok”.
- memvalidasi: menyortir tupel - createIndex menyortir output dari fase pemindaian indeks.
- memvalidasi: koleksi pemindaian - createIndex memindai koleksi untuk memvalidasi kunci indeks yang ditemukan dalam dua fase sebelumnya. Satuan ukuran adalah “blok”.

Contoh keluaran pembuatan indeks

Pada contoh keluaran di bawah ini (pembuatan indeks latar depan), status pembuatan indeks ditampilkan. Bidang “msg” merangkum kemajuan build dengan menunjukkan tahapan dan persentase penyelesaian build. Bidang “pekerja” menunjukkan jumlah pekerja yang digunakan selama tahap pembuatan indeks tersebut. Bidang “kemajuan” menunjukkan angka aktual yang digunakan untuk menghitung persentase penyelesaian.

> **Note**
>
> Bidang “currentIndexBuildNama”, “msg”, dan “kemajuan” tidak didukung di Amazon DocumentDB versi 4.0.

```
{
    "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
}
```

Mengelola kompresi dokumen tingkat koleksi

Kompresi dokumen tingkat pengumpulan Amazon DocumentDB memungkinkan Anda menurunkan biaya penyimpanan dan IO dengan mengompresi dokumen dalam koleksi Anda. Anda dapat mengaktifkan kompresi dokumen pada tingkat pengumpulan dan melihat metrik kompresi sesuai kebutuhan dengan mengukur keuntungan penyimpanan melalui metrik kompresi seperti ukuran penyimpanan dokumen terkompresi dan status kompresi. Amazon DocumentDB menggunakan algoritme kompresi LZ4 untuk mengompres dokumen.

Pedoman

Pedoman berikut berlaku untuk kompresi dokumen tingkat pengumpulan:

- Kompresi dokumen dinonaktifkan secara default
- Kompresi dokumen tidak dapat diterapkan ke koleksi yang ada.
- Kompresi dokumen hanya didukung di Amazon DocumentDB versi 5.0 dan yang lebih tinggi.
- Amazon DocumentDB hanya memampatkan dokumen dengan ukuran 2KB dan lebih besar.

Mengaktifkan kompresi dokumen

Aktifkan kompresi dokumen saat membuat koleksi di Amazon DocumentDB dengan menggunakan `db.createCollection()` metode:

```
db.createCollection( sample_collection,{
```

```
    storageEngine : {
        documentDB: {
            compression:{
                enable: <true | false>
            }
        }
    }
})
```

Memantau kompresi dokumen

Anda dapat memeriksa apakah koleksi dikompresi dan menghitung rasio kompresi itu sebagai berikut.

Lihat statistik kompresi dengan menjalankan `db.collection.stats()` perintah `db.printCollectionStats()` atau dari shell mongo. Output menunjukkan ukuran asli dan ukuran terkompresi yang dapat Anda bandingkan untuk menganalisis keuntungan penyimpanan dari kompresi dokumen. Dalam contoh ini, statistik untuk koleksi bernama “sample_collection” ditampilkan:

```
db.sample_collection.stats(1024*1024)

{
    "ns" : "test.sample_collection",
    "count" : 1000000,
    "size" : 3906.3,
    "avgObjSize" : 4096,
    "storageSize" : 1953.1,
    compression:{
        "enabled" : true,
        "threshold" : 2032
    }
    ...
}
```

- ukuran - Ukuran asli dari koleksi dokumen.
- avgObjSize- Ukuran dokumen rata-rata sebelum kompresi dibulatkan ke desimal pertama. Satuan ukuran adalah byte.
- StorageSize - Ukuran penyimpanan koleksi setelah kompresi. Satuan ukuran adalah byte.
- enabled - Menunjukkan apakah kompresi diaktifkan atau dinonaktifkan.

Untuk menghitung rasio kompresi aktual, bagi ukuran koleksi dengan ukuran penyimpanan (Size/StorageSize). Untuk contoh di atas, perhitungannya adalah 3906.3/1953.1 yang diterjemahkan menjadi rasio kompresi 2:1.

Mengelola koleksi yang ada

Meskipun Anda tidak dapat mengompres koleksi yang ada, Anda dapat mengonversi dokumen yang tidak terkompresi atau terkompresi. Untuk menyimpan dokumen yang tidak terkompresi yang ada dalam format terkompresi, salin dokumen ke koleksi berkemampuan kompresi. Untuk mengonversi dokumen terkompresi ke format yang tidak terkompresi, salin dokumen ke koleksi yang dinonaktifkan kompresi.

Mengelola Peristiwa Amazon DocumentDB

Amazon DocumentDB (dengan kompabilitas MongoDB) menyimpan catatan peristiwa yang terkait dengan klaster, instans, snapshot, grup keamanan, dan grup parameter klaster Anda. Informasi ini mencakup tanggal dan waktu acara, nama sumber dan jenis sumber acara, dan pesan yang terkait dengan acara tersebut.

> **Important**
>
> Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon RDS dan Amazon Neptune. Batas wilayah, batas yang diatur di tingkat Wilayah, dibagi antara Amazon DocumentDB, Amazon RDS, dan Amazon Neptune. Untuk informasi selengkapnya, lihat Kuota Wilayah.

Topik

- Menampilkan Kategori Peristiwa Amazon DocumentDB
- Menampilkan Peristiwa Amazon DocumentDB

Menampilkan Kategori Peristiwa Amazon DocumentDB

Setiap jenis sumber daya Amazon DocumentDB memiliki jenis kejadian tertentu yang dapat dikaitkan dengannya. Anda dapat menggunakan operasi AWS CLI `describe-event-categories` untuk menampilkan pemetaan antara jenis peristiwa dan jenis sumber daya Amazon DocumentDB.

Parameter

- **`--source-type`**—Opsional. Gunakan parameter `--source-type` untuk melihat kategori kejadian pada jenis sumber tertentu. Nilai berikut diizinkan:
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`
- **`--filters`**—Opsional. Untuk menampilkan kategori peristiwa hanya untuk Amazon DocumentDB, gunakan filter `--filter Name=engine,Values=docdb`.

Example

Kode berikut mencantumkan kategori peristiwa yang terkait dengan klaster.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-event-categories \
    --filter Name=engine,Values=docdb \
    --source-type db-cluster
```

Untuk Windows:

```
aws docdb describe-event-categories ^
    --filter Name=engine,Values=docdb ^
    --source-type db-cluster
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "EventCategoriesMapList": [
        {
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
```

```
            ],
            "SourceType": "db-cluster"
        }
    ]
}
```

Kode berikut mencantumkan kategori peristiwa yang terkait dengan setiap jenis sumber Amazon DocumentDB.

```
aws docdb describe-event-categories
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "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"
            ]
        }
    ]
}
```

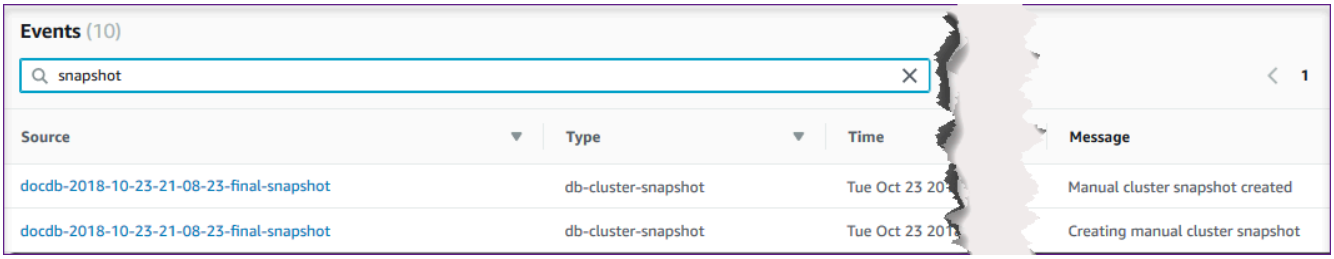
Menampilkan Peristiwa Amazon DocumentDB

Anda dapat mengambil kejadian untuk sumber daya Amazon DocumentDB Anda melalui konsol Amazon DocumentDB, yang menampilkan kejadian selama 24 jam terakhir. Anda juga dapat mengambil kejadian untuk sumber daya Amazon DocumentDB Anda dengan menggunakan AWS CLIperintah gambarkan-kejadian, atau operasi API DescribeEventsAmazon DocumentDB. Jika Anda menggunakan AWS CLI atau API Amazon DocumentDB untuk menampilkan kejadian, Anda dapat mengambil kejadian hingga 14 hari terakhir.

Using the AWS Management Console

Untuk menampilkan semua acara instans Amazon DocumentDB selama 24 jam terakhir

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih kejadian. Kejadian yang tersedia muncul dalam daftar.
3. Gunakan daftar Filter untuk menyelekasi peristiwa menurut jenisnya. Masukkan suatu istilah dalam kotak teks untuk menyeleksi hasil Anda lebih lanjut. Misalnya, cuplikan layar berikut menunjukkan pemfilteran semua peristiwa Amazon DocumentDB untuk peristiwa snapshot.



Using the AWS CLI

Untuk menampilkan semua peristiwa instans Amazon DocumentDB selama 7 hari terakhir

Anda dapat menampilkan semua peristiwa instans Amazon DocumentDB selama 7 hari terakhir dengan menjalankan operasi AWS CLI deskripsi kejadian dengan parameter `--duration` yang disetel ke `10080` (10.080 menit).

```
aws docdb describe-events --duration 10080
```

Penyaringan untuk Peristiwa Amazon DocumentDB

Untuk menampilkan peristiwa Amazon DocumentDB tertentu, gunakan operasi `describe-events` dengan parameter berikut.

Parameter

- **`--filter`**—Diperlukan untuk membatasi nilai yang ditampilkan ke peristiwa Amazon DocumentDB. Gunakan **`Name=engine,Values=docdb`** untuk memfilter semua peristiwa untuk Amazon DocumentDB saja.
- **`--source-identifier`**—Opsional. Pengidentifikasi sumber peristiwa yang akan menerima peristiwa tersebut. Jika dihilangkan, peristiwa dari semua sumber disertakan ke dalam hasilnya.
- **`--source-type`**—Opsional, jika `--source-identifier` tidak disediakan, maka ini diperlukan. Jika `--source-identifier` disediakan, `--source-type` harus setuju dengan jenis `--source-identifier`. Nilai berikut diizinkan:
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`

Contoh berikut mencantumkan semua persitiwa Amazon DocumentDB Anda.

```
aws docdb describe-events --filters Name=engine,Values=docdb
```

Output dari operasi ini terlihat seperti berikut (format JSON).

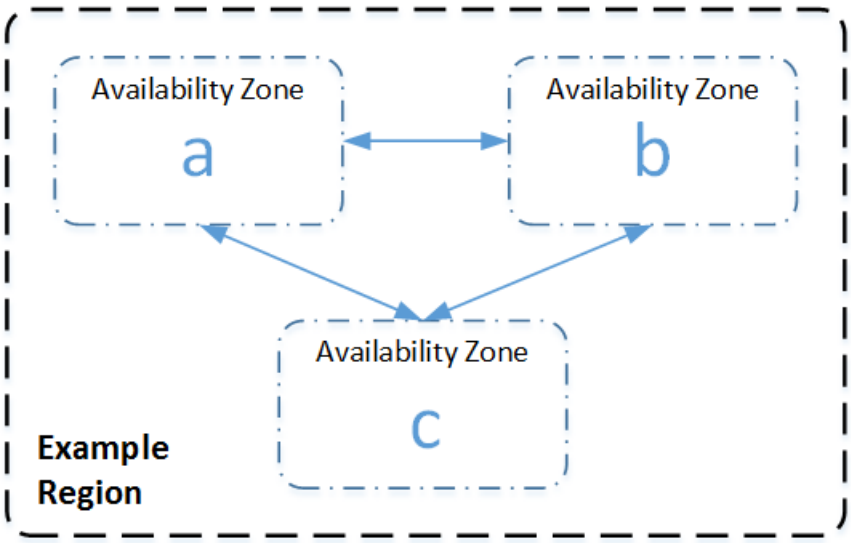
```
{
    "Events": [
        {
            "SourceArn": "arn:aws:rds:us-east-1:123SAMPLE012:db:sample-cluster-
instance3",
            "Message": "instance created",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:17:40.023Z",
            "SourceIdentifier": "sample-cluster-instance3",
            "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"
            ]
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengaudit Peristiwa Amazon DocumentDB.

Memilih Wilayah dan Availability Zone

Sumber daya komputasi cloud Amazon di-host di beberapa lokasi di seluruh dunia. Lokasi ini terdiri dari Wilayah AWS dan Availability Zones. Setiap Wilayah AWS adalah wilayah geografis yang terpisah. Setiap Wilayah memiliki beberapa lokasi terisolasi yang dikenal sebagai Availability Zone. Amazon DocumentDB memberi Anda kemampuan untuk menempatkan sumber daya, seperti instans, dan data di beberapa lokasi. Sumber daya tidak direplikasi Wilayah AWS kecuali Anda melakukannya secara khusus.

Amazon mengoperasikan pusat data yang canggih dan sangat tersedia. Meskipun jarang terjadi, kegagalan dapat terjadi yang memengaruhi ketersediaan instans yang berada di lokasi yang sama. Jika Anda menghosting semua instans Anda di satu lokasi yang terpengaruh oleh kegagalan tersebut, tidak ada instans Anda yang akan tersedia. Diagram berikut menunjukkan Wilayah AWS dengan tiga Availability Zones.



Penting untuk diingat bahwa setiap Wilayah adalah independen. Aktivitas Amazon DocumentDB apa pun yang Anda mulai (misalnya, membuat instans atau membuat daftar instans yang tersedia) hanya berjalan di Wilayah AWS default Anda saat ini. Anda dapat mengubah Wilayah default di konsol tersebut dengan mengatur variabel lingkungan `EC2_REGION`. Atau Anda dapat menimpanya dengan menggunakan parameter `--region` di AWS CLI. Untuk informasi selengkapnya, lihat Mengkonfigurasi AWS Command Line Interface, secara khusus, bagian tentang variabel lingkungan dan opsi baris perintah.

Saat Anda membuat klaster menggunakan konsol Amazon DocumentDB, dan Anda memilih untuk membuat replika di Availability Zone yang berbeda, Amazon DocumentDB membuat dua instans. Ini membuat instans primer di satu Availability Zone dan instans replika di Availability Zone yang berbeda. Volume klaster selalu direplikasi di tiga Availability Zone.

Untuk membuat atau bekerja dengan instans Amazon DocumentDB secara Wilayah AWS spesifik, gunakan titik akhir layanan regional yang sesuai.

Ketersediaan Wilayah

Amazon DocumentDB tersedia di Wilayah berikut. AWS

Wilayah yang Didukung oleh Amazon DocumentDB

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
AS Timur (Ohio)	`us-east-2`	3
AS Timur (Virginia Utara)	`us-east-1`	6
AS Barat (Oregon)	`us-west-2`	4
Amerika Selatan (Sao Paulo)	`sa-east-1`	3
Asia Pasifik (Hong Kong)	`ap-east-1`	3
Asia Pasifik (Hyderabad)	`ap-south-2`	3
Asia Pasifik (Mumbai)	`ap-south-1`	3
Asia Pasifik (Seoul)	`ap-northeast-2`	4
Asia Pasifik (Singapura)	`ap-southeast-1`	3
Asia Pasifik (Sydney)	`ap-southeast-2`	3

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
Asia Pasifik (Tokyo)	`ap-northeast-1`	3
Kanada (Pusat)	`ca-central-1`	3
Wilayah Tiongkok (Beijing)	`cn-north-1`	3
Tiongkok (Ningxia)	`cn-northwest-1`	3
Eropa (Frankfurt)	`eu-central-1`	3
Eropa (Irlandia)	`eu-west-1`	3
Eropa (London)	`eu-west-2`	3
Eropa (Milan)	`eu-south-1`	3
Eropa (Paris)	`eu-west-3`	3
Timur Tengah (UEA)	`me-central-1`	3
AWS GovCloud (AS-Barat)	`us-gov-west-1`	3
AWS GovCloud (AS-Timur)	`us-gov-east-1`	3

Secara default, zona waktu untuk klaster Amazon DocumentDB adalah Universal Time Coordinated (UTC).

Untuk informasi tentang menemukan titik akhir koneksi untuk klaster dan instans di wilayah tertentu, lihat Memahami Titik Akhir Amazon DocumentDB.

Mengelola grup parameter cluster Amazon DocumentDB

Anda dapat mengelola konfigurasi mesin Amazon DocumentDB dengan menggunakan parameter dalam grup parameter klaster. Grup parameter klaster adalah kumpulan nilai konfigurasi Amazon DocumentDB yang mempermudah pengelolaan parameter klaster Amazon DocumentDB Anda.

Grup parameter klaster bertindak sebagai kontainer untuk nilai konfigurasi mesin yang diterapkan ke semua instans dalam klaster.

Bagian ini menjelaskan cara membuat, melihat, dan mengubah grup parameter klaster. Ini juga menunjukkan bagaimana Anda dapat menentukan grup parameter klaster mana yang dikaitkan dengan klaster tertentu.

Topik

- Menjelaskan kelompok parameter Custer Amazon DocumentDB
- Membuat grup parameter cluster Amazon DocumentDB
- Memodifikasi grup parameter cluster Amazon DocumentDB
- Memodifikasi cluster Amazon DocumentDB untuk menggunakan grup parameter cluster kustom
- Menyalin grup parameter cluster Amazon DocumentDB
- Menyetel ulang grup parameter cluster Amazon DocumentDB
- Menghapus grup parameter cluster Amazon DocumentDB
- Referensi parameter cluster Amazon DocumentDB

Menjelaskan kelompok parameter Custer Amazon DocumentDB

Grup parameter `default` cluster dibuat secara otomatis saat Anda membuat klaster Amazon DocumentDB pertama di wilayah baru atau menggunakan mesin baru. Cluster berikutnya, yang dibuat di wilayah yang sama dan memiliki versi mesin yang sama, dibuat dengan grup parameter `default` cluster.

Topik

- Menjelaskan detail grup parameter cluster Amazon DocumentDB
- Menentukan grup parameter cluster Amazon DocumentDB

Menjelaskan detail grup parameter cluster Amazon DocumentDB

Untuk menjelaskan detail grup parameter klaster tertentu, selesaikan langkah-langkah berikut menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI).

Using the AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di panel Grup parameter klaster , pilih nama grup parameter yang ingin Anda lihat detailnya.
4. Halaman yang dihasilkan menunjukkan parameter grup parameter, aktivitas terbaru, dan tag.

   - Di bawah Parameter klaster, Anda dapat melihat nama parameter, nilai saat ini, nilai yang diizinkan, apakah parameter dapat dimodifikasi, jenis penerapannya, tipe data, dan deskripsinya. Anda dapat mengubah parameter individual dengan memilih parameter lalu memilih Edit di bagian Parameter klaster. Untuk informasi selengkapnya, lihat Memodifikasi parameter cluster Amazon DocumentDB.
   - Di bawah Peristiwa terbaru, Anda dapat melihat peristiwa terbaru untuk grup parameter ini. Anda dapat memfilter acara ini menggunakan bar pencarian di bagian ini. Untuk informasi selengkapnya, lihat Mengelola Peristiwa Amazon DocumentDB.
   - Di bawah Tag, Anda dapat melihat tag yang ada di grup parameter klaster ini. Anda dapat menambah atau menghapus tag dengan memilih Edit di bagian Tag. Untuk informasi selengkapnya, lihat Penandaan Sumber Daya Amazon DocumentDB.

Using the AWS CLI

Anda dapat menggunakan perintah `describe-db-cluster-parameter-groups` AWS CLI untuk melihat Amazon Resource Name (ARN), keluarga, deskripsi, dan nama grup parameter klaster tunggal atau semua grup parameter klaster yang Anda miliki untuk Amazon DocumentDB. Anda juga dapat menggunakan perintah `describe-db-cluster-parameters` AWS CLI untuk melihat parameter dan detailnya di dalam satu grup parameter klaster.

- **`--describe-db-cluster-parameter-groups`** — Untuk melihat daftar semua grup parameter klaster dan detailnya.

- **`--db-cluster-parameter-group-name`** – Opsional. Nama grup parameter klaster yang ingin Anda jelaskan. Jika parameter ini dihilangkan, semua grup parameter klaster akan dijelaskan.

- **`--describe-db-cluster-parameters`** — Untuk membuat daftar semua parameter di dalam grup parameter dan nilainya.
  - **`--db-cluster-parameter-group name`** — Diperlukan. Nama grup parameter klaster yang ingin Anda jelaskan.

Example

Kode berikut mencantumkan hingga 100 grup parameter klaster dan ARN, keluarga, deskripsi, dan namanya.

```
aws docdb describe-db-cluster-parameter-groups
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
          "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

Kode berikut mencantumkan ARN, keluarga, deskripsi, dan nama untuk `sample-parameter-group`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameter-groups \
          --db-cluster-parameter-group-name sample-parameter-group
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
          --db-cluster-parameter-group-name sample-parameter-group
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
          "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

Kode berikut mencantumkan nilai parameter dalam `sample-parameter-group`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
   "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"
         }
   ]
}
```

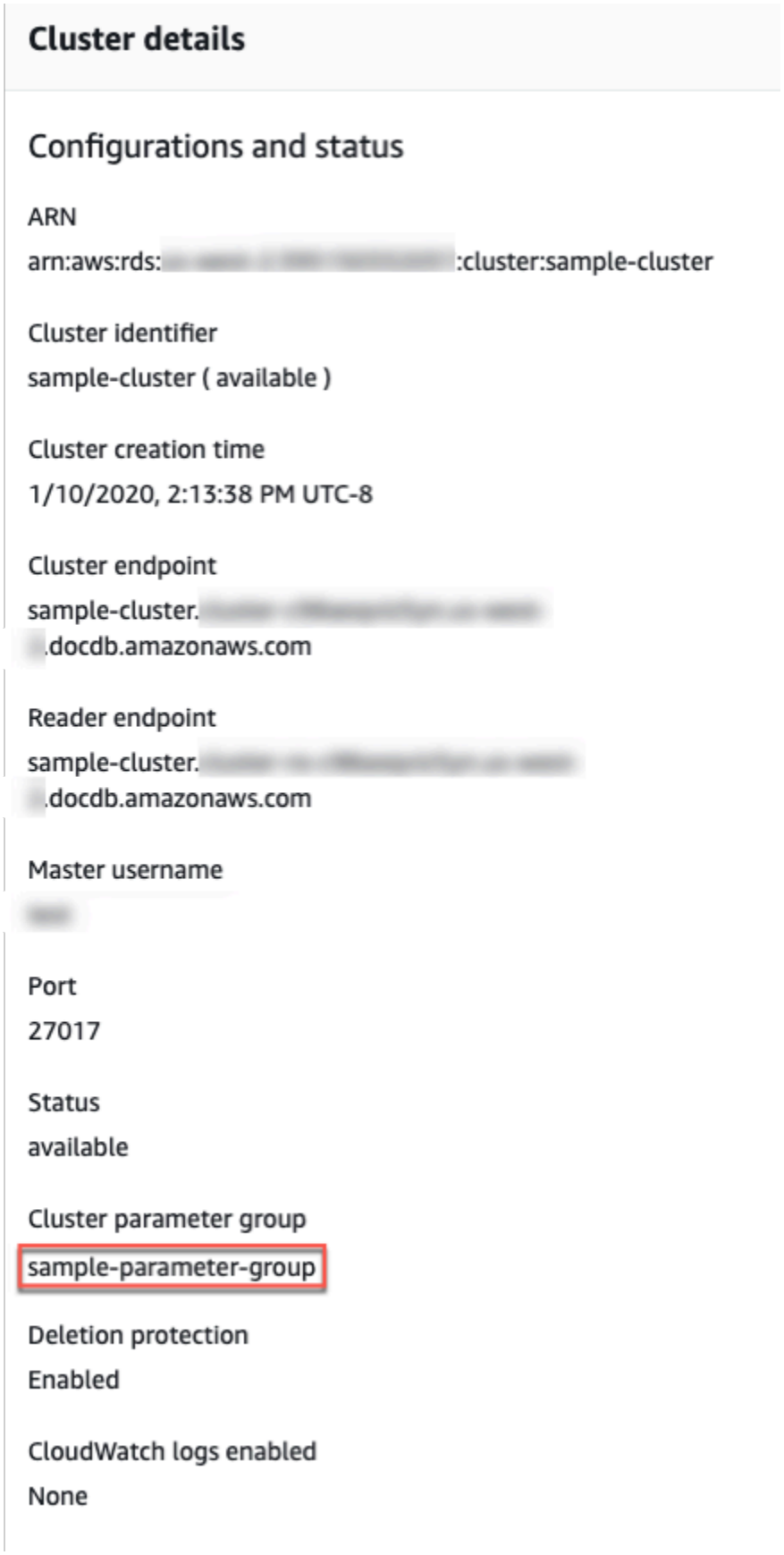
Menentukan grup parameter cluster Amazon DocumentDB

Untuk menentukan grup parameter mana yang terkait dengan klaster tertentu, selesaikan langkah-langkah berikut menggunakan AWS Management Console atau AWS CLI.

Using the AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi kiri, pilih Klaster.
3. Dalam daftar klaster, pilih nama klaster yang Anda minati.

4. Halaman yang dihasilkan menunjukkan detail klaster yang Anda pilih. Gulir ke bawah ke Detail klaster. Di bagian bawah bagian tersebut, temukan nama grup parameter di bawah Grup parameter klaster.


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

Kode AWS CLI berikut menentukan grup parameter mana yang mengatur klaster `sample-cluster`.

```
aws docdb describe-db-clusters \
      --db-cluster-identifier sample-cluster \
      --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
       [
           "sample-cluster",
           "sample-parameter-group"
       ]
]
```

Membuat grup parameter cluster Amazon DocumentDB

Grup parameter cluster default sepertidefault.docdb5.0,default.docdb4.0, ataudefault.docdb3.6, dibuat saat Anda membuat klaster dengan versi mesin baru dan di wilayah baru. Cluster berikutnya dibuat di wilayah ini dan dengan versi mesin yang sama mewarisi grup parameter `default` cluster. Setelah dibuat, grup `default` parameter tidak dapat dihapus atau diganti namanya. Anda dapat memodifikasi perilaku engine instance cluster dengan membuat grup parameter kustom dengan nilai parameter pilihan dan melampirkannya ke cluster Amazon DocumentDB Anda.

Prosedur berikut memandu Anda melalui pembuatan grup parameter cluster kustom. Anda kemudian dapat mengubah parameter dalam grup parameter tersebut.

> **Note**
>
> Setelah Anda membuat grup parameter klaster, Anda harus menunggu setidaknya 5 menit sebelum menggunakan grup parameter tersebut. Ini memungkinkan Amazon DocumentDB untuk menyelesaikan tindakan `create` sepenuhnya sebelum grup parameter klaster digunakan untuk klaster baru. Anda dapat menggunakan AWS Management Console atau operasi `describe-db-cluster-parameter-groups` AWS CLI untuk memverifikasi

bahwa grup parameter klaster Anda telah dibuat. Untuk informasi selengkapnya, lihat Menjelaskan kelompok parameter Custer Amazon DocumentDB.

Using the AWS Management Console

Untuk membuat grup parameter klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di panel Grup parameter klaster, pilih Buat.
4. Di panel Buat grup parameter klaster, masukkan berikut ini:

   a. Nama grup — Masukkan nama untuk grup parameter klaster. Misalnya, `sample-parameter-group`. Grup parameter klaster memiliki batasan penamaan berikut:

      - Panjangnya [1–255] karakter alfanumerik.
      - Karakter pertama harus berupa huruf.
      - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.

   b. Deskripsi — Memberikan deskripsi untuk grup parameter klaster ini.
5. Untuk membuat grup parameter klaster, pilih Buat. Untuk membatalkan operasi, pilih Batal.
6. Setelah Anda memilih Buat, teks berikut akan muncul di bagian atas halaman untuk memverifikasi bahwa grup parameter klaster Anda telah berhasil dibuat:

```
Successfully created cluster parameter group 'sample-parameter-group'.
```

Using the AWS CLI

Untuk membuat grup parameter klaster baru untuk klaster Amazon DocumentDB 4.0, gunakan operasi AWS CLI `create-db-cluster-parameter-group` dengan parameter berikut:

- **`--db-cluster-parameter-group-name`** — Nama grup parameter klaster kustom. Misalnya, `sample-parameter-group`.
- **`--db-cluster-parameter-group-family`** — Keluarga grup parameter klaster yang digunakan sebagai templat untuk grup parameter klaster kustom. Saat ini, ini harus `docdb4.0`.
- **`--description`** — Deskripsi yang diberikan pengguna untuk grup parameter klaster ini. Contoh berikut menggunakan "`Custom docdb4.0 parameter group`".

Untuk Linux, macOS, atau 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"
```

Untuk 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"
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "sample-parameter-group",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Custom docdb4.0 parameter group",
        "DBClusterParameterGroupArn": "sample-parameter-group-arn"
    }
}
```

Memodifikasi grup parameter cluster Amazon DocumentDB

Bagian ini menjelaskan cara memodifikasi grup parameter Amazon DocumentDB kustom. Di Amazon DocumentDB, Anda tidak dapat memodifikasi `default` grup parameter cluster yang dibuat saat pertama kali membuat klaster dengan versi engine baru di wilayah baru. Jika klaster Amazon DocumentDB Anda menggunakan grup parameter klaster default dan Anda ingin mengubah nilai di dalamnya, Anda harus terlebih dahulu membuat grup parameter baru atau menyalin grup parameter yang ada, memodifikasinya, lalu menerapkan grup parameter yang dimodifikasi ke klaster Anda.

Selesaikan langkah-langkah berikut untuk memodifikasi grup parameter cluster kustom. Memodifikasi tindakan bisa memakan waktu beberapa saat untuk disebarkan. Harap tunggu grup paramater cluster yang dimodifikasi tersedia sebelum melampirkannya ke cluster Anda. Anda dapat menggunakan AWS Management Console atau operasi AWS CLI `describe-db-cluster-parameters` untuk memverifikasi bahwa grup parameter klaster Anda telah dimodifikasi. Untuk informasi selengkapnya, lihat Menjelaskan kelompok parameter cluster.

Using the AWS Management Console

Ikuti langkah-langkah ini untuk mengubah grup parameter Amazon DocumentDB kustom. Anda tidak dapat mengubah grup parameter `default`. Jika Anda ingin mengubah nilai dalam grup parameter `default`, Anda dapat menyalin grup parameter klaster default, memodifikasinya, lalu menerapkan grup parameter yang dimodifikasi ke klaster Anda. Untuk informasi selengkapnya tentang menerapkan grup parameter ke klaster Anda, lihat Memodifikasi cluster Amazon DocumentDB.

Untuk mengubah grup parameter klaster kustom

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi di sisi kiri konsol tersebut, pilih Grup parameter. Dalam daftar grup parameter, pilih nama grup parameter yang ingin Anda ubah.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Untuk setiap parameter dalam grup parameter yang ingin Anda ubah, lakukan hal berikut:

a. Temukan parameter yang ingin Anda ubah, dan verifikasi bahwa parameter tersebut dapat dimodifikasi dengan memeriksa apakah parameter tersebut terdaftar sebagai `true` di bawah kolom yang Dapat dimodifikasi.

b. Jika dapat dimodifikasi, pilih parameter dan pilih Edit dari kanan atas halaman konsol tersebut.

c. Dalam kotak dialog Modifikasi **<parameter-name>**, buat perubahan yang Anda inginkan. Kemudian pilih Ubah parameter klaster, atau pilih Batal untuk membuang perubahan.

Using the AWS CLI

Anda dapat memodifikasi `ParameterValue`, `Description`, atau `ApplyMethod` dari parameter apa pun yang dapat dimodifikasi di dalam grup parameter klaster Amazon DocumentDB kustom menggunakan AWS CLI. Anda tidak dapat membuat modifikasi langsung ke grup parameter klaster default.

Untuk mengubah parameter grup parameter klaster kustom, gunakan operasi `modify-db-cluster-parameter-group` dengan parameter berikut.

- **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang Anda modifikasi.
- **`--parameters`** — Diperlukan. Parameter yang Anda modifikasi. Untuk daftar parameter yang berlaku untuk semua instans di klaster Amazon DocumentDB, lihat Referensi parameter cluster Amazon DocumentDB. Setiap entri parameter harus menyertakan hal berikut:
  - **`ParameterName`** — Nama parameter yang Anda ubah.
  - **`ParameterValue`** — Nilai baru untuk parameter ini.
  - **`ApplyMethod`** — Bagaimana Anda ingin perubahan pada parameter ini diterapkan. Nilai yang diizinkan adalah `immediate` dan `pending-reboot`.

> **Note**
>
> Parameter dengan `ApplyType` dari `static` harus memiliki `ApplyMethod` dari `pending-reboot`.

Example - Memodifikasi nilai parameter

Dalam contoh ini, Anda mencantumkan nilai parameter `sample-parameter-group` dan memodifikasi parameter `tls`. Kemudian, setelah menunggu 5 menit, Anda kembali mencantumkan nilai parameter `sample-parameter-group` untuk melihat nilai parameter yang diubah.

1. Buat daftar parameter dan nilai `sample-parameter-group` mereka.

   Untuk Linux, macOS, atau Unix:

   ```
   aws docdb describe-db-cluster-parameters \
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   Untuk Windows:

   ```
   aws docdb describe-db-cluster-parameters ^
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   Output dari operasi ini terlihat seperti berikut (format JSON).

   ```
   {
          "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. Ubah parameter `tls` sehingga nilainya menjadi `disabled` .

   Anda tidak dapat mengubah `ApplyMethod` karena `ApplyType` adalah `static`.

   Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --parameters
 "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
```

   Untuk Windows:

```
aws docdb modify-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --parameters
 "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
```

   Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
```

3. Tunggu paling tidak 5 menit.
4. Buat daftar nilai parameter `sample-parameter-group` untuk memverifikasi bahwa parameter `tls` telah dimodifikasi.

   Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
```

   Untuk Windows:

```
aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
       "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"
           }
       ]
}
```

Memodifikasi cluster Amazon DocumentDB untuk menggunakan grup parameter cluster kustom

Saat Anda membuat klaster Amazon DocumentDB, grup parameter `default.docdb4.0` secara otomatis dibuat untuk klaster tersebut. Anda tidak dapat mengubah grup parameter klaster `default`. Sebagai gantinya, Anda dapat memodifikasi klaster Amazon DocumentDB untuk mengaitkan grup parameter baru yang disesuaikan dengan itu.

Bagian ini menjelaskan cara memodifikasi klaster Amazon DocumentDB yang ada untuk menggunakan grup parameter klaster kustom menggunakan AWS Management Console dan AWS Command Line Interface (AWS CLI).

Using the AWS Management Console

Untuk memodifikasi klaster Amazon DocumentDB untuk menggunakan grup parameter klaster bukan-default yang baru

1. Sebelum Anda mulai, pastikan Anda telah membuat klaster Amazon DocumentDB dan grup parameter klaster. Lihat Membuat cluster Amazon DocumentDB dan Membuat grup parameter cluster Amazon DocumentDB untuk instruksi lebih lanjut.
2. Setelah membuat grup parameter klaster Anda, buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb. Di panel navigasi, pilih Klaster untuk menambahkan grup parameter baru Anda ke sebuah klaster.
3. Pilih klaster yang ingin Anda kaitkan dengan grup parameter Anda. Pilih Tindakan, lalu pilih Ubah untuk mengubah klaster Anda.
4. Di bawah Opsi klaster, pilih grup parameter baru yang ingin Anda kaitkan dengan klaster Anda.
5. Pilih Lanjutkan untuk melihat ringkasan dari modifikasi Anda.
6. Setelah memverifikasi perubahan Anda, Anda dapat menerapkannya segera atau selama jendela pemeliharaan berikutnya di bawah Penjadwalan modifikasi.
7. Pilih Ubah klaster untuk memperbarui klaster Anda dengan grup parameter baru Anda.

Using the AWS CLI

Sebelum Anda memulai, pastikan Anda telah membuat klaster Amazon DocumentDB dan grup parameter klaster. Anda dapat membuat klaster Amazon DocumentDB menggunakan operasi AWS CLI `create-db-cluster`. Anda dapat membuat grup parameter klaster menggunakan operasi AWS CLI `create-db-cluster-parameter-group`.

Untuk menambahkan grup parameter klaster baru ke klaster Anda, gunakan operasi AWS CLI `modify-db-cluster` dengan parameter berikut.

- -- db-cluster-identifier — Nama cluster Anda (misalnya,`sample-cluster`).
- -- db-cluster-parameter-group -name — Nama grup parameter yang ingin Anda kaitkan dengan cluster Anda (misalnya,`sample-parameter-group`).

Example

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster
    --db-cluster-parameter-group-name sample-parameter-group
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
"DBCluster": {
    "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster",
    "DBClusterParameterGroup": "sample-parameter-group",
    "DBSubnetGroup": "default",
   ...
}
```

Menyalin grup parameter cluster Amazon DocumentDB

Anda dapat membuat salinan grup parameter klaster di Amazon DocumentDB menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI).

Using the AWS Management Console

Prosedur berikut memandu Anda membuat grup parameter klaster baru dengan membuat salinan grup parameter klaster yang ada.

Untuk menyalin grup parameter klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .
3. Di panel Grup parameter klaster, pilih nama grup parameter klaster yang ingin Anda salin.
4. Pilih Tindakan, lalu pilih Salin untuk menyalin grup parameter tersebut.

5. Di bawah Opsi salin, masukkan nama dan deskripsi untuk grup parameter klaster baru. Kemudian pilih Salin untuk menyimpan perubahan Anda.

Using the AWS CLI

Untuk membuat salinan grup parameter klaster, gunakan operasi `copy-db-cluster-parameter-group` dengan parameter berikut.

- **`--source-db-cluster-parameter-group-identifier`** — Diperlukan. Nama atau Amazon Resource Name (ARN) dari grup parameter klaster yang ingin Anda buat salinannya.

  Jika grup parameter klaster sumber dan target berada di Wilayah AWS yang sama, pengidentifikasi dapat berupa nama atau ARN.

  Jika grup parameter klaster sumber dan target berada di Wilayah AWS yang berbeda, pengidentifikasi harus berupa ARN.
- **`--target-db-cluster-parameter-group-identifier`** — Diperlukan. Nama atau ARN salinan grup parameter klaster.

  Batasan:
  - Tidak dapat berupa null, kosong, atau blank.
  - Harus berisi 1-255 huruf, angka, atau tanda hubung.
  - Karakter pertama harus berupa huruf.
  - Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.
- **`--target-db-cluster-parameter-group-description`** — Diperlukan. Deskripsi yang diberikan pengguna untuk salinan grup parameter klaster.

Example

Kode berikut membuat salinan `sample-parameter-group`, menamai salinan `sample-parameter-group-copy`.

Untuk Linux, macOS, atau 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"
```

Untuk 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"
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "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"
    }
}
```

Menyetel ulang grup parameter cluster Amazon DocumentDB

Anda dapat mengatur ulang beberapa atau semua nilai parameter grup parameter klaster Amazon DocumentDB ke nilai defaultnya dengan menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI) untuk mengatur ulang grup parameter klaster.

Using the AWS Management Console

Ikuti langkah-langkah ini untuk mengatur ulang beberapa atau semua nilai parameter grup parameter klaster ke nilai defaultnya.

Untuk mengatur ulang nilai parameter grup parameter klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi di sisi kiri konsol tersebut, pilih Grup parameter.
3. Di panel Grup parameter klaster, pilih nama grup parameter klaster yang ingin Anda atur ulang.
4. Pilih Tindakan, lalu pilih Atur Ulang untuk mengatur ulang grup parameter tersebut.

5. Pada halaman Konfirmasi reset grup parameter klaster yang dihasilkan, konfirmasi bahwa Anda ingin mengatur ulang semua parameter klaster untuk grup parameter tersebut ke defaultnya. Kemudian pilih Atur ulang untuk mengatur ulang grup parameter Anda. Anda juga dapat memilih Batal untuk membuang perubahan Anda.

Using the AWS CLI

Untuk mereset beberapa atau semua nilai parameter grup parameter klaster ke nilai defaultnya, gunakan operasi `reset-db-cluster-parameter-group` dengan parameter berikut.

- **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang akan diatur ulang.
- **`--parameters`** – Opsional. Daftar `ParameterName` dan `ApplyMethod` dalam grup parameter klaster untuk diatur ulang ke nilai defaultnya. Parameter statis harus disetel ke `pending-reboot` agar berlaku pada mulai ulang instans atau permintaan `reboot-db-instance` berikutnya. Anda harus memanggil `reboot-db-instance` untuk setiap instans di klaster yang Anda inginkan untuk menerapkan parameter statis yang diperbarui.

  Parameter ini dan `--reset-all-parameters` saling eksklusif: Anda dapat menggunakan salah satunya tetapi tidak keduanya.
- **`--reset-all-parameters`** atau **`--no-reset-all-parameters`** — Opsional. Menentukan apakah akan mengatur ulang semua parameter (`--reset-all-parameters` atau hanya beberapa parameter (`--no-reset-all-parameters`) ke nilai defaultnya. Parameter `--reset-all-parameters` dan `--parameters` saling eksklusif: Anda dapat menggunakan salah satunya tetapi tidak keduanya.

  Saat Anda mengatur ulang seluruh grup, parameter dinamis segera diperbarui. Parameter statis diatur ke `pending-reboot` untuk diterapkan pada mulai ulang instans atau permintaan `reboot-db-instance` berikutnya. Anda harus memanggil `reboot-db-instance` untuk setiap instans di klaster yang Anda inginkan untuk menerapkan parameter statis yang diperbarui.

Example

Contoh 1: Mengatur ulang semua parameter ke nilai defaultnya

Kode berikut mengatur ulang semua parameter di grup parameter klaster `sample-parameter-group` nilai defaultnya.

Untuk Linux, macOS, atau Unix:

```
aws docdb reset-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --reset-all-parameters
```

Untuk Windows:

```
aws docdb reset-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --reset-all-parameters
```

Contoh 2: Mengatur ulang parameter yang ditentukan ke nilai defaultnya

Kode berikut mengatur ulang parameter `tls` di grup parameter klaster `sample-parameter-group` ke nilai defaultnya.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
```

Melakukan reboot instans klaster

Sebelum nilai parameter statis diubah, instans klaster harus di-boot ulang. Reboot setiap instans di klaster Anda yang Anda inginkan untuk menerapkan parameter statis yang diperbarui.

Untuk Linux, macOS, atau Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-cluster-instance
```

Untuk Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-cluster-instance
```

Menghapus grup parameter cluster Amazon DocumentDB

Anda dapat menghapus grup parameter klaster Amazon DocumentDB kustom menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI). Anda tidak dapat menghapus grup parameter klaster `default.docdb4.0`.

Using the AWS Management Console

Untuk menghapus grup parameter klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di panel Grup parameter, pilih tombol radio di sebelah kiri grup parameter klaster yang ingin Anda hapus.
4. Pilih Tindakan, lalu pilih Hapus.
5. Di panel konfirmasi Hapus, pilih Hapus untuk menghapus grup parameter klaster. Untuk mempertahankan grup parameter klaster, pilih Batal.

Using the AWS CLI

Untuk menghapus grup parameter klaster, gunakan operasi `delete-db-cluster-parameter-group` dengan parameter berikut.

- **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang akan dihapus. Ini harus menjadi grup parameter klaster yang sudah ada. Anda tidak dapat menghapus grup parameter *`default.docdb4.0`* cluster.

Example - Menghapus grup parameter klaster

Contoh berikut memandu Anda menjalani tiga langkah untuk menghapus grup parameter klaster:

1. Menemukan nama grup parameter klaster yang ingin Anda hapus.
2. Menghapus grup parameter klaster yang ditentukan.
3. Memverifikasi bahwa grup parameter klaster telah dihapus.

1. Temukan nama grup parameter klaster yang ingin Anda hapus.

Kode berikut mencantumkan nama semua grup parameter klaster.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Output dari operasi sebelumnya adalah daftar nama grup parameter klaster yang mirip dengan berikut ini (format JSON).

```
[
       [
           "default.docdb4.0"
       ],
       [
```

```
            "sample-parameter-group"
        ],
        [
            "sample-parameter-group-copy"
        ]
    ]
```

2. Menghapus grup parameter klaster tertentu.

Kode berikut menghapus grup parameter klaster `sample-parameter-group-copy`.

Untuk Linux, macOS, atau Unix:

```
aws docdb delete-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Untuk Windows:

```
aws docdb delete-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Tidak ada output dari operasi ini.

3. Verifikasi bahwa grup parameter klaster yang ditentukan telah dihapus.

Kode berikut mencantumkan nama semua grup parameter klaster yang tersisa.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Output dari operasi sebelumnya adalah daftar grup parameter klaster yang mirip dengan berikut ini (format JSON). Grup parameter klaster yang baru saja Anda hapus tidak boleh ada dalam daftar.

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
       [
           "default.docdb4.0"
       ],
       [
           "sample-parameter-group"
       ]
   ]
```

Referensi parameter cluster Amazon DocumentDB

Saat Anda mengubah parameter dinamis dan menyimpan grup parameter klaster, perubahan akan segera diterapkan terlepas dari pengaturan Terapkan segera. Saat Anda mengubah parameter statis dan menyimpan grup parameter klaster, perubahan parameter akan berlaku setelah Anda melakukan reboot instans secara manual. Anda dapat melakukan reboot instans menggunakan konsol Amazon DocumentDB atau dengan memanggil `reboot-db-instance` secara eksplisit.

Tabel berikut menunjukkan parameter yang berlaku untuk semua instans di klaster Amazon DocumentDB.

Parameter tingkat klaster Amazon DocumentDB

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
`audit_logs`	dinonaktifkan	diaktifkan, dinonaktifkan, ddl, dml_read, dml_write, semua, tidak ada	Ya	Dinamis	String	Menentukan apakah log CloudWatch audit Amazon diaktifkan. • **`enabled`**—log CloudWatch audit

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
						diaktifkan. • **disabled**—log CloudWatch audit dinonaktifkan. • **ddl**—audit untuk acara DDL diaktifkan. • **dml_read**—auditng untuk acara baca DMLdiaktifkan. • **dml_write**—audit untuk acara penulisan DMLdiaktifkan. • **all**—audit untuk semua

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
						peristiwa database diaktifkan. • **none**— audit dinonaktifkan.
`change_stream_log_retention_duration`	10800	3600-604800	Ya	Dinamis	Bilangan Bulat	Menentukan durasi waktu (dalam detik) bahwa log aliran perubahan dipertahankan dan dapat digunakan.

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
`profiler`	dinonaktifkan	diaktifkan, dinonaktifkan	Ya	Dinamis	String	Mengaktifkan pembuatan profil untuk operasi yang lambat. • **`enabled`**— operasi yang memakan waktu lebih lama dari nilai ambang batas yang ditentukan pelanggan (misalnya, 100 md) dicatat ke Amazon Logs. CloudWatch • **`disabled`**— operasi lambat

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
						tidak dicatat ke CloudWatch Log.
`profiler_sampling_rate`	1.0	0,0-1,0	Ya	Dinamis	Desimal	Menentukan laju pengambilan sampel untuk operasi yang dicatat.

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
`profiler_threshold_ms`	100	50-2147483646	Ya	Dinamis	Bilangan Bulat	Mendefinisikan ambang batas untuk `profiler`. • Semua operasi yang lebih besar dari `profiler_threshold_ms` yang dicatat ke CloudWatch Log.

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
`tls`	diaktifkan	diaktifkan, dinonaktifkan, fips-140-3	Ya	Statis	String	Menentukan apakah koneksi Transport Layer Security (TLS) diperlukan. • **`enabled`**— Koneksi TLS diperlukan untuk terhubung. • **`disabled`**— Koneksi TLS tidak dapat digunakan untuk terhubung. • **`fips-140-3`** Koneksi TLS dengan atribut Federal Information

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
						Processing Standards (FIPS) diperlukan untuk terhubung. Cluster hanya menerima koneksi aman per Publikasi FIPS 140-3. Ini hanya didukung mulai dengan cluster Amazon DocumentDB 5.0 (engine versi 3.0.3727) di wilayah ini: ca-central-1, us-west-2, us-

Parameter	Nilai default	Nilai yang Valid	Dapat diubah	Tipe Penerapan	Tipe Data	Deskripsi
						east-1 , us-east-2 , us-east-2 , -1. us-gov-east us-gov-west
`ttl_monitor`	diaktifkan	diaktifkan, dinonaktifkan	Ya	Dinamis	String	Menentukan apakah pemantauan Time to Live (TTL) diaktifkan untuk klaster. • **`enabled`**— Pemantauan TTL diaktifkan. • **`disabled`**— Pemantauan TTL dinonaktifkan.

Memodifikasi parameter cluster Amazon DocumentDB

Di Amazon DocumentDB, grup parameter klaster terdiri dari parameter yang berlaku untuk semua instans yang Anda buat di klaster. Untuk grup parameter klaster kustom, Anda dapat mengubah nilai

parameter kapan saja atau mengatur ulang semua nilai parameter ke default untuk grup parameter yang Anda buat. Bagian ini menjelaskan cara melihat parameter yang membentuk grup parameter klaster Amazon DocumentDB dan nilainya, dan bagaimana Anda dapat mengubah atau memperbarui nilai ini.

Parameter bisa dinamis atau statis. Saat Anda mengubah parameter dinamis dan menyimpan grup parameter klaster, perubahan akan segera diterapkan terlepas dari pengaturan `Apply Immediately`. Saat Anda mengubah parameter statis dan menyimpan grup parameter klaster, perubahan parameter hanya berlaku setelah Anda melakukan reboot instans secara manual.

Melihat parameter grup parameter cluster Amazon DocumentDB

Anda dapat melihat parameter klaster Amazon DocumentDB dan nilainya menggunakan AWS Management Console atau AWS CLI.

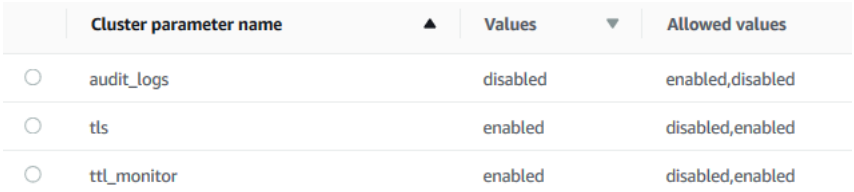
Using the AWS Management Console

Untuk melihat detail grup parameter klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di panel Grup parameter, pilih nama grup parameter klaster yang ingin Anda lihat detailnya.
4. Halaman yang dihasilkan menunjukkan nilai berikut untuk setiap parameter: nama parameter, nilai saat ini, nilai yang diizinkan, apakah parameter dapat dimodifikasi, jenis penerapan, tipe data, dan deskripsi.



	Cluster parameter name	Values	Allowed values
○	audit_logs	disabled	enabled,disabled
○	tls	enabled	disabled,enabled
○	ttl_monitor	enabled	disabled,enabled

Using the AWS CLI

Untuk melihat parameter grup parameter klaster dan nilainya, gunakan operasi `describe-db-cluster-parameters` dengan parameter berikut.

- **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang Anda inginkan daftar parameternya secara mendetail.
- **`--source`** – Opsional. Jika disediakan, mengembalikan hanya parameter untuk sumber tertentu. Sumber parameter dapat berupa `engine-default`, `system`, atau `user`.

Example

Kode berikut mencantumkan parameter dan nilainya untuk grup parameter `custom3-6-param-grp`. Untuk informasi lebih lanjut tentang grup parameter, hilangkan garis `--query`. Untuk informasi tentang semua grup parameter, hilangkan garis `--db-cluster-parameter-group-name`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameters \
   --db-cluster-parameter-group-name custom3-6-param-grp \
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameters ^
   --db-cluster-parameter-group-name custom3-6-param-grp ^
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    [
        "audit_logs",
        "disabled"
    ],
    [
        "tls",
        "enabled"
    ],
    [
```

```
        "ttl_monitor",
        "enabled"
    ]
]
```

Memodifikasi parameter grup parameter cluster Amazon DocumentDB

Anda dapat mengubah parameter grup parameter menggunakan AWS Management Console atau AWS CLI.

Using the AWS Management Console

Untuk memperbarui parameter grup parameter klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di panel Grup parameter, pilih grup parameter klaster yang ingin Anda perbarui parameternya.
4. Halaman yang dihasilkan menunjukkan parameter dan detail yang sesuai untuk grup parameter klaster ini. Pilih parameter untuk diperbarui.
5. Di kanan atas halaman, pilih Edit untuk mengubah nilai parameter. Untuk informasi selengkapnya tentang tipe parameter klaster, lihat Referensi parameter cluster Amazon DocumentDB.
6. Buat perubahan Anda, lalu pilih Ubah parameter klaster untuk menyimpan perubahan. Untuk membuang perubahan Anda, pilih Batal.

Using the AWS CLI

Untuk mengubah parameter grup parameter klaster, gunakan operasi `modify-db-cluster-parameter-group` dengan parameter berikut:

- **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang Anda modifikasi.
- **`--parameters`** — Diperlukan. Parameter atau parameter yang Anda modifikasi. Setiap entri parameter harus menyertakan hal berikut:
  - **`ParameterName`** — Nama parameter yang Anda ubah.
  - **`ParameterValue`** — Nilai baru untuk parameter ini.
  - **`ApplyMethod`** — Bagaimana Anda ingin perubahan pada parameter ini diterapkan. Nilai yang diizinkan adalah `immediate` dan `pending-reboot`.

    > **Note**
    >
    > Parameter dengan `ApplyType` dari `static` harus memiliki `ApplyMethod` dari `pending-reboot`.

Untuk mengubah nilai parameter grup parameter klaster (AWS CLI)

Contoh berikut mengubah parameter `tls`.

1. Buat daftar parameter dan nilainya **`sample-parameter-group`**

   Untuk Linux, macOS, atau Unix:

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Untuk Windows:

   ```
   aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Output dari operasi ini terlihat seperti berikut (format JSON).

   ```
   {
       "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. Ubah parameter **`tls`** sehingga nilainya menjadi **`disabled`**. Anda tidak dapat mengubah `ApplyMethod` karena `ApplyType` adalah `static`.

   Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 "ParameterName"=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

   Untuk Windows:

```
aws docdb modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

   Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
```

```
    "DBClusterParameterGroupName": "sample-parameter-group"
}
```

3. Tunggu setidaknya 5 menit.
4. Buat daftar nilai parameter dari**`sample-parameter-group`**.

   Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

   Untuk Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

   Output dari operasi ini terlihat seperti berikut (format JSON).

```
{
    "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"
        }
```

```
    ]
}
```

Memahami Titik Akhir Amazon DocumentDB

Anda dapat menggunakan titik akhir Amazon DocumentDB (dengan kompatibilitas MongoDB) untuk terhubung ke klaster atau instans. Amazon DocumentDB memiliki tiga jenis titik akhir yang berbeda, masing-masing dengan tujuannya sendiri.

Topik

- Menemukan Titik Akhir Klaster
- Menemukan Titik Akhir Instans
- Menyambung ke Titik Akhir

Titik akhir klaster

Titik akhir klaster adalah titik akhir untuk klaster Amazon DocumentDB yang tersambung ke instans utama saat ini untuk klaster tersebut. Setiap klaster Amazon DocumentDB memiliki titik akhir klaster tunggal dan satu instans utama. Jika terjadi failover, titik akhir klaster dipetakan ulang ke instans utama baru.

Reader endpoint

Reader endpoint adalah titik akhir untuk klaster Amazon DocumentDB yang terhubung ke salah satu replika yang tersedia untuk klaster tersebut. Setiap klaster Amazon DocumentDB memiliki reader endpoint. Jika ada lebih dari satu replika, reader endpoint mengarahkan setiap permintaan koneksi ke salah satu replika Amazon DocumentDB.

Instans titik akhir

Titik akhir instans adalah titik akhir yang terhubung ke instans tertentu. Setiap instans dalam sebuah klaster, terlepas dari apakah itu instans primer atau replika, memiliki titik akhir instans uniknya sendiri. Yang terbaik adalah tidak menggunakan titik akhir instans di aplikasi Anda. Ini karena mereka dapat mengubah peran jika terjadi failover, sehingga memerlukan perubahan kode dalam aplikasi Anda.

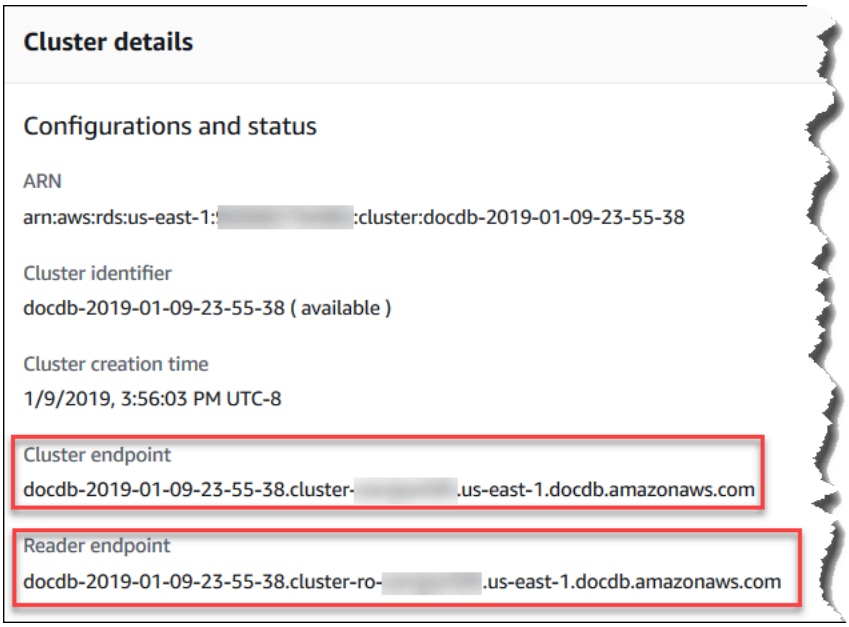
Menemukan Titik Akhir Klaster

Anda dapat menemukan titik akhir klaster dan reader endpoint pada suatu klaster dengan menggunakan konsol Amazon DocumentDB atau AWS CLI.

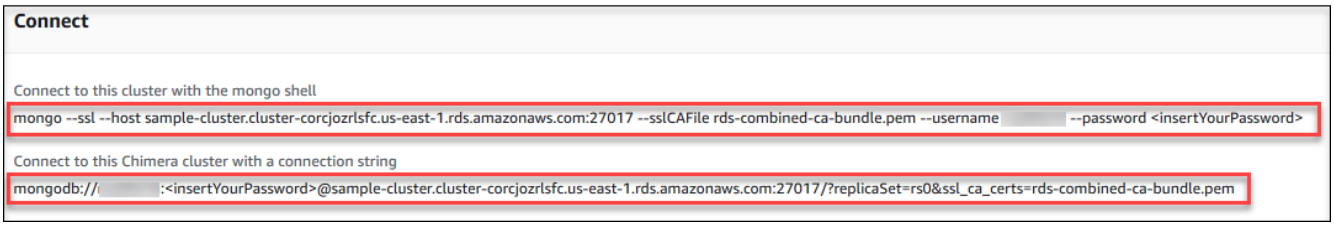
Using the AWS Management Console

Untuk menemukan titik akhir klaster dengan menggunakan konsol

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih klaster.
3. Dari daftar klaster, pilih nama klaster yang Anda minati.
4. Gulir ke bawah ke bagian Detail dan temukan titik akhir klaster dan reader endpoint.



5. Untuk terhubung ke klaster ini, gulir ke atas ke bagian Koneksi. Temukan string koneksi untuk shell `mongo` dan string koneksi yang dapat digunakan dalam kode aplikasi untuk terhubung ke klaster Anda.



Using the AWS CLI

Untuk menemukan klaster dan reader endpoint untuk klaster Anda menggunakan AWS CLI, jalankan perintah `describe-db-clusters` dengan parameter ini.

Parameter

- **`--db-cluster-identifier`**—Opsional. Menentukan klaster untuk mengembalikan titik akhir. Jika dihilangkan, menampilkan titik akhir hingga 100 klaster Anda.
- **`--query`**—Opsional. Menentukan bidang yang akan menampilkan. Bermanfaat dengan mengurangi jumlah data yang Anda butuhkan untuk melihat serta menemukan titik akhir. Jika dihilangkan, semua informasi tentang klaster dikembalikan.
- **`--region`**—Opsional. Gunakan parameter `--region` untuk menentukan Wilayah tempat Anda ingin menerapkan perintah. Jika dihilangkan, Wilayah default Anda digunakan.

Example

Contoh berikut mengembalikan `DBClusterIdentifier`, titik akhir (titik akhir klaster), dan `ReaderEndpoint` untuk `sample-cluster`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
   --region us-east-1 \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
   --region us-east-1 ^
   --db-cluster-identifier sample-cluster ^
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
  [
```

```
    "sample-cluster",
    27017,
    "sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com",
    "sample-cluster.cluster-ro-corlsfccjozr.us-east-1.docdb.amazonaws.com"
  ]
]
```

Sekarang setelah Anda memiliki titik akhir klaster, Anda dapat terhubung ke klaster menggunakan salah satu `mongo` atau `mongodb`. Untuk informasi selengkapnya, lihat Menyambung ke Titik Akhir.

Menemukan Titik Akhir Instans

Anda dapat menemukan titik akhir untuk instans dengan menggunakan konsol Amazon DocumentDB atau AWS CLI.

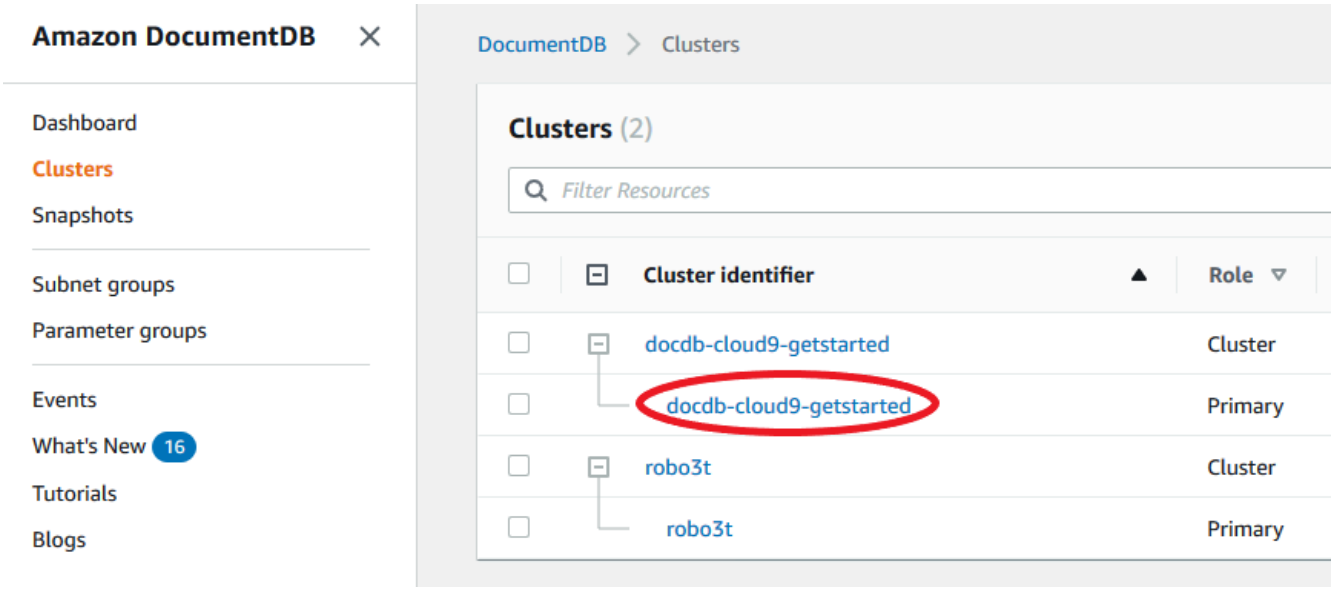
Using the AWS Management Console

Untuk menemukan titik akhir instans dengan menggunakan konsol

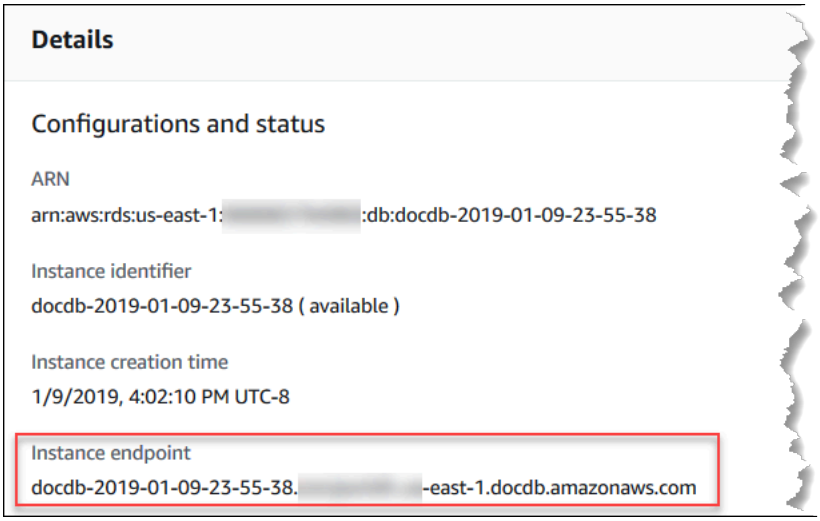
1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Centang kotak di sebelah kiri instans yang Anda minati.
5. Gulir ke bawah ke bagian Detail lalu temukan titik akhir instans.



6. Untuk terhubung ke instans ini, gulir ke atas ke bagian Koneksi. Temukan string koneksi untuk shell mongo dan string koneksi yang dapat digunakan dalam kode aplikasi Anda untuk terhubung ke instans Anda.

Using the AWS CLI

Untuk menemukan titik akhir instans menggunakan AWS CLI, jalankan perintah berikut dengan argumen ini.

Pendapat

- **`--db-instance-identifier`**—Opsional. Menentukan instans untuk menampilkan titik akhir. Jika dihilangkan, mengembalikan titik akhir hingga 100 instans Anda.
- **`--query`**—Opsional. Menentukan bidang yang akan menampilkan. Bermanfaat dengan mengurangi jumlah data yang Anda butuhkan untuk melihat serta menemukan titik akhir. Jika dihilangkan, semua informasi pada sebuah instans dikembalikan. Bidang `Endpoint` memiliki tiga anggota, jadi mencantumkannya dalam kueri seperti pada contoh berikut mengembalikan ketiga anggota. Jika Anda hanya tertarik pada beberapa anggota `Endpoint`, ganti `Endpoint` pada kueri dengan anggota yang Anda minati, seperti pada contoh kedua.
- **`--region`**—Opsional. Gunakan parameter `--region` untuk menentukan Wilayah tempat Anda ingin menerapkan perintah. Jika dihilangkan, Wilayah default Anda digunakan.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

Untuk Windows:

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    [
```

```
        "sample-cluster-instance",
        {
            "Port": 27017,
            "Address": "sample-cluster-instance.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "HostedZoneId": "Z2R2ITUGPM61AM"
        }
    ]
]
```

Mengurangi output untuk menghilangkan `HostedZoneId` pada titik akhir, Anda dapat memodifikasi kueri Anda dengan menentukan `Endpoint.Port` dan `Endpoint.Address`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

Untuk Windows:

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    [
        "sample-cluster-instance",
        27017,
        "sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com"
    ]
]
```

Sekarang setelah Anda memiliki titik akhir instans, Anda dapat terhubung ke instans dengan menggunakan `mongo` atau `mongodb`. Untuk informasi selengkapnya, lihat Menyambung ke Titik Akhir.

Menyambung ke Titik Akhir

Setelah Anda memiliki titik akhir, baik klaster atau instans, Anda dapat menghubungkannya dengan menggunakan shell mongo atau string koneksi.

Membuat Koneksi dengan Menggunakan mongo Shell

Gunakan struktur berikut untuk membuat string yang perlu Anda sambungkan ke klaster atau instans Anda dengan menggunakan shell mongo:

```
mongo \
    --ssl \
    --host Endpoint:Port \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Contoh shell **mongo**

Sambungkan ke klaster:

```
mongo \
    --ssl \
    --host sample-cluster.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Sambungkan ke instans:

```
mongo \
    --ssl \
    --host sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Membuat Koneksi dengan Menggunakan String Koneksi

Gunakan struktur berikut untuk membuat string koneksi yang Anda perlukan untuk terhubung ke klater atau instans Anda.

```
mongodb://UserName:Password@endpoint:port?replicaSet=rs0&ssl_ca_certs=global-
bundle.pem
```

Contoh string koneksi

Sambungkan ke klaster:

```
mongodb://UserName:Password@sample-cluster.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Sambungkan ke instans:

```
mongodb://UserName:Password@sample-cluster-instance.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Memahami Amazon DocumentDB Amazon Resource Name (ARN)

Sumber daya yang Anda buat masing-masing AWS diidentifikasi secara unik dengan Nama Sumber Daya Amazon (ARN). Untuk operasi Amazon DocumentDB (dengan kompatibilitas MongoDB) tertentu, Anda harus secara unik mengidentifikasi sumber daya Amazon DocumentDB dengan menentukan ARN-nya. Misalnya, saat Anda menambahkan tag ke sumber daya, Anda harus menyediakan ARN sumber daya.

Topik

- Membangun ARN untuk Sumber Daya Amazon DocumentDB
- Menemukan ARN Sumber Daya Amazon DocumentDB

Membangun ARN untuk Sumber Daya Amazon DocumentDB

Anda dapat membangun ARN untuk sumber daya Amazon DocumentDB menggunakan sintaks berikut. Amazon DocumentDB berbagi format Amazon Relational Database Service (Amazon RDS) ARNS. ARN Amazon DocumentDB terdiri atas `rds` dan bukan `docdb`.

`arn:aws:rds:region:account_number:resource_type:resource_id`

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
AS Timur (Ohio)	us-east-2	3
AS Timur (Virginia Utara)	us-east-1	6
AS Barat (Oregon)	us-west-2	4
Amerika Selatan (Sao Paulo)	sa-east-1	3
Asia Pasifik (Hong Kong)	ap-east-1	3
Asia Pasifik (Hyderabad)	ap-south-2	3
Asia Pasifik (Mumbai)	ap-south-1	3
Asia Pasifik (Seoul)	ap-northeast-2	4
Asia Pasifik (Singapura)	ap-southeast-1	3
Asia Pasifik (Sydney)	ap-southeast-2	3
Asia Pasifik (Tokyo)	ap-northeast-1	3
Kanada (Pusat)	ca-central-1	3
Wilayah Tiongkok (Beijing)	cn-north-1	3
Tiongkok (Ningxia)	cn-northwest-1	3
Eropa (Frankfurt)	eu-central-1	3
Eropa (Irlandia)	eu-west-1	3
Eropa (London)	eu-west-2	3

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
Eropa (Milan)	`eu-south-1`	3
Eropa (Paris)	`eu-west-3`	3
Timur Tengah (UEA)	`me-central-1`	3
AWS GovCloud (AS-Barat)	`us-gov-west-1`	3
AWS GovCloud (AS-Timur)	`us-gov-east-1`	3

> **Note**
>
> Arsitektur Amazon DocumentDB memisahkan penyimpanan dan komputasi. Untuk lapisan penyimpanan, Amazon DocumentDB mereplikasi enam salinan data Anda di AWS tiga Availability Zones (AZ). AZ yang tercantum dalam tabel di atas adalah jumlah AZ yang dapat Anda gunakan di region tertentu untuk menyediakan instans komputasi. Sebagai contoh, jika Anda meluncurkan klaster Amazon DocumentDB di ap-northeast-1, penyimpanan Anda akan direplikasi enam cara di tiga AZ tetapi instans komputasi Anda hanya akan tersedia di dua AZ.

Tabel berikut menunjukkan format yang harus Anda gunakan saat membangun ARN untuk jenis sumber daya Amazon DocumentDB tertentu. Amazon DocumentDB berbagi format Amazon RDS ARNS. ARN Amazon DocumentDB terdiri atas `rds` dan bukan `docdb`.

Jenis Sumber Daya	Format/Contoh ARN
Contoh (db)	`arn:aws:rds:` *`region`*`:`*`account_number`*`:db:`*`resource_id`* `arn:aws:rds:us-east-1:` *`1234567890`* `:db:`*`sample-db-instance`*

Jenis Sumber Daya	Format/Contoh ARN
Klaster (`cluster`)	`arn:aws:rds:` *`region`*`:`*`account_number`* `:cluster:` *`resource_id`* `arn:aws:rds:us-east-1:` *`1234567890`* `:cluster:` *`sample-db-cluster`*
Grup parameter klaster (`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`*
Grup keamanan (`secgrp`)	`arn:aws:rds:` *`region`*`:`*`account_number`* `:secgrp:`*`resource_id`* `arn:aws:rds:us-east-1:` *`1234567890`* `:secgrp:`*`sample-public-secgrp`*
Snapshot klaster (`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`*
Grup subnet (`subgrp`)	`arn:aws:rds:` *`region`*`:`*`account_number`* `:subgrp:`*`resource_id`* `arn:aws:rds:us-east-1:` *`1234567890`* `:subgrp:`*`sample-subnet-10`*

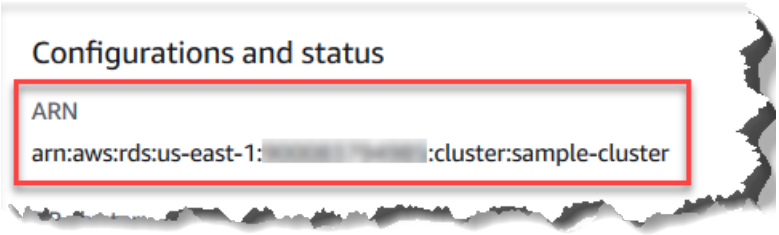
Menemukan ARN Sumber Daya Amazon DocumentDB

Anda dapat menemukan ARN dari sumber daya Amazon DocumentDB menggunakan atau file. AWS Management Console AWS CLI

Using the AWS Management Console

Untuk menemukan ARN dengan menggunakan konsol, arahkan navigasi ke sumber daya yang Anda inginkan untuk ARN, dan lihat detail untuk sumber daya tersebut.

Misalnya, Anda bisa mendapatkan ARN untuk klaster di panel Detail untuk klaster tersebut, seperti yang ditunjukkan pada cuplikan layar berikut.



Using the AWS CLI

Untuk mendapatkan ARN menggunakan sumber daya Amazon DocumentDB tertentu, gunakan `describe` operasi AWS CLI untuk sumber daya tersebut. Tabel berikut menunjukkan setiap AWS CLI operasi dan properti ARN yang digunakan dengan operasi untuk mendapatkan ARN.

AWS CLI Perintah	Properti ARN
`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 - Menemukan ARN untuk klaster

AWS CLI Operasi berikut menemukan ARN untuk cluster. `sample-cluster`

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster"
]
```

Example - Menemukan ARN untuk beberapa grup parameter

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-cluster-parameter-groups \
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

```
[
    "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"
]
```

Penandaan Sumber Daya Amazon DocumentDB

Anda dapat menggunakan tanda Amazon DocumentDB (dengan kompatibilitas MongoDB) untuk menambahkan metadata ke sumber daya Amazon DocumentDB Anda. Tanda tersebut juga dapat digunakan dengan kebijakan AWS Identity and Access Management (IAM) untuk mengelola akses ke sumber daya Amazon DocumentDB dan untuk mengontrol tindakan apa yang dapat diterapkan ke sumber daya. Anda juga dapat menggunakan tanda untuk melacak biaya dengan mengelompokkan pengeluaran untuk sumber daya serupa yang ditandai.

Anda dapat menandai pada sumber Amazon DocumentDB berikut ini:

- Klaster
- Instans
- Snapshot
- Snapshot klaster
- Grup parameter
- Grup parameter klaster
- Grup keamanan
- Grup subnet

Ikhtisar Tag Sumber Daya Amazon DocumentDB

Tanda Amazon DocumentDB adalah pasangan nama dan nilai yang Anda tentukan dan berkaitan dengan sumber daya Amazon DocumentDB. Nama ini disebut sebagai kunci. Memberikan nilai untuk kunci bersifat opsional. Anda dapat menggunakan tanda untuk menetapkan informasi tambahan ke sumber daya Amazon DocumentDB. Anda dapat menggunakan kunci tag, misalnya, untuk menentukan kategori, dan nilai tag mungkin merupakan item dalam kategori tersebut. Sebagai contoh, Anda dapat menentukan kunci tanda `project` dan nilai tanda `Salix`, yang menunjukkan bahwa sumber daya Amazon DocumentDB ditugaskan ke proyek Salix. Anda juga bisa menggunakan tanda untuk menunjuk domain Amazon DocumentDB agar digunakan untuk uji atau produksi dengan menggunakan kunci seperti `environment=test` atau

`environment=production`. Kami menyarankan Anda untuk menggunakan kumpulan kunci tanda yang konsisten guna mempermudah untuk melacak metadata yang berkaitan dengan sumber daya Amazon DocumentDB.

Anda juga dapat menggunakan tanda untuk mengorganisasi tagihan AWS Anda untuk merefleksikan struktur biaya Anda sendiri. Untuk melakukannya, daftar untuk mendapatkan tagihan Akun AWS Anda dengan menyertakan nilai kunci tanda. Lalu, untuk melihat biaya sumber daya gabungan, organisasikan informasi penagihan Anda sesuai dengan sumber daya Anda dengan nilai kunci tanda yang sama. Misalnya, Anda dapat menandai beberapa sumber daya dengan nama aplikasi tertentu, kemudian organisasikan informasi penagihan Anda untuk melihat biaya total aplikasi tersebut pada beberapa layanan. Untuk informasi selengkapnya, lihat Menggunakan Tanda Alokasi Biaya dalam Panduan Pengguna Manajemen Penagihan dan Biaya AWS.

Setiap sumber daya Amazon DocumentDB memiliki satu rangkaian tanda, yang berisi semua tanda yang ditetapkan ke sumber daya tersebut. Rangkaian tanda dapat berisi 10 tanda, atau dapat kosong. Jika Anda menambahkan tanda ke sumber daya Amazon DocumentDB yang memiliki kunci yang sama dengan tanda yang ada pada sumber daya, nilai lama akan ditimpa dengan nilai baru.

AWS tidak menerapkan makna semantik pada tanda Anda; tanda ditafsirkan secara ketat sebagai string karakter. Amazon DocumentDB dapat menetapkan tanda pada instans atau sumber daya Amazon DocumentDB lainnya, bergantung pada pengaturan yang Anda gunakan saat Anda membuat sumber daya. Sebagai contoh, Amazon DocumentDB dapat menambahkan tanda yang menunjukkan bahwa instans digunakan untuk produksi atau untuk pengujian.

Anda dapat menambahkan tanda ke snapshot, namun tagihan Anda tidak akan mencerminkan pengelompokan ini.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menambah, mencantumkan, dan menghapus tanda pada sumber daya Amazon DocumentDB. Saat menggunakan AWS CLI, Anda harus menyediakan Amazon Resource Name (ARN) untuk sumber daya yang ingin Anda gunakan. Untuk informasi selengkapnya tentang Amazon DocumentDB ARN, lihat Memahami Amazon DocumentDB Amazon Resource Name (ARN).

Kendala Tanda

Kendala berikut ini berlaku untuk tanda Amazon DocumentDB:

- Jumlah maksimum tanda per sumber daya - 10
- Panjang Kunci maksimum – 128 karakter Unicode

- Panjang Nilai maksimum – 256 karakter Unicode
- Karakter valid untuk Kunci dan Nilai - huruf besar dan huruf kecil dalam set karakter UTF-8, digit, spasi, dan karakter berikut ini: `_ . : / = + -` dan `@` (Java regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`)
- Kunci dan nilai tag peka huruf besar dan kecil.
- Awalan `aws:` tidak dapat digunakan untuk kunci atau nilai tanda; awalan tersebut disediakan untuk AWS.

Menambahkan dan Memperbarui Tanda pada Sumber Daya Amazon DocumentDB

Anda dapat menambahkan hingga 10 tanda ke sumber daya menggunakan AWS Management Console atau AWS CLI.

Using the AWS Management Console

Proses untuk menambahkan tanda ke sumber daya serupa terlepas dari sumber daya apa yang Anda tambahkan tanda. Dalam contoh ini, Anda menambahkan tanda ke klaster.

Untuk menambah atau memperbarui tanda ke klaster menggunakan konsol

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dari panel navigasi, pilih klaster.
3. Pilih nama klaster yang ingin Anda tambahkan tanda.
4. Gulir ke bawah ke bagian Tanda, dan kemudian pilih Edit.
5. Untuk setiap tanda yang ingin Anda tambahkan ke sumber daya ini, lakukan hal berikut ini:
    a. Untuk menambahkan tanda baru, masukkan nama tanda di kotak Kunci. Untuk mengubah nilai tanda, temukan nama tanda di kolom Kunci.
    b. Untuk memberikan nilai baru atau nilai yang diperbarui pada tanda, masukkan nilai untuk tanda di kotak Nilai.
    c. Jika Anda memiliki lebih banyak tanda untuk ditambahkan, pilih Tambahkan. Jika tidak, saat selesai, pilih Simpan.

Using the AWS CLI

Proses untuk menambahkan tanda ke sumber daya serupa terlepas dari apa sumber daya yang Anda tambahkan tanda. Dalam contoh ini, Anda menambahkan tiga tanda ke klaster. Tanda kedua, `key2`, tidak memiliki nilai.

Gunakan operasi AWS CLI `add-tags-to-resource` dengan parameter tersebut.

Parameter

- **`--resource-name`**—ARN dari sumber daya Amazon DocumentDB yang ingin Anda tambahkan tanda.
- **`--tags`**—Daftar tanda (pasangan kunci-nilai) yang ingin Anda tambahkan ke sumber daya ini dalam format `Key=`*`key-name`*`,Value=`*`tag-value`*.

Example

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Operasi `add-tags-to-resource` tidak menghasilkan keluaran. Untuk melihat hasil operasi, gunakan operasi `list-tags-for-resource`.

Mencantumkan Tanda pada Sumber Daya Amazon DocumentDB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mendapatkan daftar tanda untuk sumber daya Amazon DocumentDB.

Using the AWS Management Console

Proses untuk mencantumkan tanda pada sumber daya serupa terlepas dari sumber daya yang Anda tambahkan tanda. Dalam contoh ini, Anda mencantumkan tanda untuk klaster.

Untuk mencantumkan tanda pada klaster menggunakan konsol

1. Buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dari panel navigasi, pilih klaster.
3. Pilih nama klaster yang ingin Anda cantumkan tanda.
4. Untuk melihat daftar tanda pada sumber daya ini, gulir ke bawah ke bagian Tanda.

Using the AWS CLI

Proses untuk mencantumkan tanda pada sumber daya serupa terlepas dari sumber daya yang Anda cantumkan tanda. Dalam contoh ini, Anda mencantumkan tanda pada klaster.

Gunakan operasi AWS CLI `list-tags-for-resource` dengan parameter tersebut.

Parameter

- **`--resource-name`**—Wajib. ARN dari sumber daya Amazon DocumentDB yang ingin Anda cantumkan tanda.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

Untuk Windows:

```
aws docdb list-tags-for-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
```

```
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": ""
        },
        {
            "Key": "key3",
            "Value": "value3"
        }
    ]
}
```

Menghapus Tanda dari Sumber Daya Amazon DocumentDB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menghapus tanda dari sumber daya Amazon DocumentDB.

Using the AWS Management Console

Proses untuk menghapus tanda dari sumber daya serupa terlepas dari sumber daya apa yang Anda tambahkan tanda. Dalam contoh ini, Anda menghapus tanda dari klaster.

Untuk menghapus tanda dari klaster menggunakan konsol

1. Buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Dari panel navigasi, pilih klaster.
3. Pilih nama klaster yang ingin Anda hapus tandanya.
4. Gulir ke bawah ke bagian Tanda, dan kemudian pilih Edit.
5. Jika Anda ingin menghapus semua tanda dari sumber daya ini, pilih Hapus semua. Jika tidak, untuk setiap tanda yang ingin Anda hapus dari sumber daya ini, lakukan hal berikut ini:
    a. Cari nama tanda di kolom Kunci.
    b. Pilih Hapus pada baris yang sama dengan kunci tanda.
    c. Setelah selesai, pilih Simpan.

Using the AWS CLI

Proses untuk menghapus tanda dari sumber daya serupa terlepas dari sumber daya apa yang Anda hapus tandanya. Dalam contoh ini, Anda menghapus tanda dari klaster.

Gunakan operasi AWS CLI `remove-tags-from-resource` dengan parameter tersebut.

- **`--resource-name`**—Wajib. ARN dari sumber daya Amazon DocumentDB yang ingin Anda hapus tandanya.
- **`--tag-keys`**—Wajib. Daftar kunci tanda yang ingin Anda hapus dari sumber daya ini.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Untuk Windows:

```
aws docdb remove-tags-from-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Operasi `removed-tags-from-resource` tidak menghasilkan keluaran. Untuk melihat hasil operasi, gunakan operasi `list-tags-for-resource`.

Memelihara Amazon DocumentDB

Secara berkala, Amazon DocumentDB melakukan pemeliharaan pada sumber daya Amazon DocumentDB. Pemeliharaan paling sering melibatkan pembaruan ke mesin basis data (pemeliharaan klaster) atau sistem operasi (OS) yang mendasari instans (pemeliharaan instans). Pembaruan mesin database adalah tambalan yang diperlukan dan mencakup perbaikan keamanan, perbaikan bug, dan penyempurnaan pada mesin database. Pembaruan untuk sistem operasi sering kali mencakup perbaikan keamanan. Meskipun tambalan sistem operasi bersifat opsional, kami menyarankan Anda menerapkannya ke instans Amazon DocumentDB Anda segera setelah tersedia.

Patch mesin database mengharuskan Anda membuat cluster Amazon DocumentDB offline untuk waktu yang singkat. Setelah tersedia, tambalan ini secara otomatis dijadwalkan untuk diterapkan selama jendela pemeliharaan terjadwal yang akan datang dari cluster Amazon DocumentDB Anda.

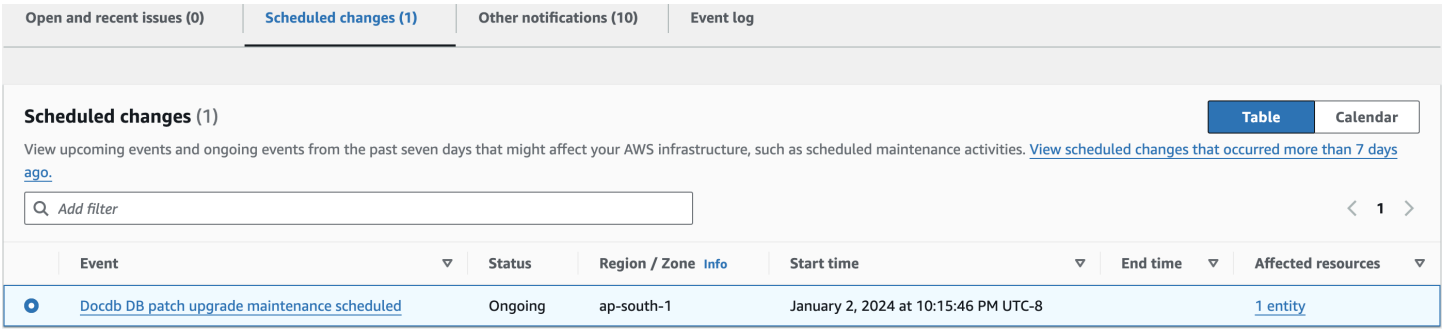
Pemeliharaan klaster dan instans memiliki jendela pemeliharaan masing-masing. Modifikasi cluster dan instance yang telah Anda pilih untuk tidak segera diterapkan, juga diterapkan selama jendela pemeliharaan. Secara default, saat Anda membuat klaster, Amazon DocumentDB menetapkan jendela pemeliharaan untuk klaster dan setiap instans tunggal. Anda dapat memilih jendela pemeliharaan saat membuat klaster atau instans. Anda juga dapat mengubah jendela pemeliharaan kapan saja agar sesuai dengan jadwal atau praktik bisnis Anda. Umumnya disarankan untuk memilih jendela pemeliharaan yang meminimalkan dampak pemeliharaan pada aplikasi Anda (misalnya, pada malam hari atau akhir pekan). Panduan ini sangat kontekstual pada jenis aplikasi dan pola penggunaan yang Anda alami.

Topik

- Pemberitahuan untuk patch mesin Amazon DocumentDB
- Melihat tindakan pemeliharaan Amazon DocumentDB yang tertunda
- Menerapkan pembaruan mesin Amazon DocumentDB
- Pembaruan yang diprakarsai pengguna
- Mengelola jendela pemeliharaan Amazon DocumentDB Anda
- Bekerja dengan pembaruan sistem operasi

Pemberitahuan untuk patch mesin Amazon DocumentDB

Anda akan menerima pemberitahuan pemeliharaan untuk patch mesin database yang diperlukan melalui acara kesehatan di AWS Health Dashboard (AHD) di AWS konsol dan melalui email. Ketika patch pemeliharaan mesin Amazon DocumentDB tersedia di wilayah AWS tertentu, semua akun pengguna Amazon DocumentDB yang terkena dampak di wilayah tersebut akan menerima pemberitahuan AHD dan email untuk setiap versi Amazon DocumentDB yang terpengaruh oleh tambalan. Anda dapat melihat pemberitahuan ini di bawah bagian Perubahan terjadwal AHD di AWS konsol. Pemberitahuan akan memiliki detail tentang waktu ketersediaan patch, jadwal penerapan otomatis, daftar cluster yang terkena dampak, dan catatan rilis. Pemberitahuan ini juga akan dikirimkan melalui e-mail ke alamat email pengguna root AWS akun.



Setelah menerima pemberitahuan ini, Anda dapat memilih untuk menerapkan sendiri patch mesin ini ke cluster Amazon DocumentDB Anda sebelum tanggal penerapan otomatis yang dijadwalkan. Atau Anda dapat menunggu patch mesin diterapkan secara otomatis selama jendela pemeliharaan yang akan datang (opsi default).

> **Note**
>
> Status untuk notifikasi di AHD akan diatur ke 'Sedang Berlangsung' hingga patch mesin Amazon DocumentDB baru dengan versi patch mesin baru dirilis.
> Setelah patch mesin diterapkan ke cluster Amazon DocumentDB Anda, versi patch mesin cluster akan diperbarui untuk mencerminkan versi dalam notifikasi. Anda dapat menjalankan `db.runCommand({getEngineVersion: 1})` perintah untuk memverifikasi pembaruan ini.

AWS Health juga terintegrasi dengan Amazon EventBridge yang menggunakan peristiwa untuk membangun aplikasi berbasis peristiwa yang dapat diskalakan dan terintegrasi dengan lebih dari 20 target, termasuk, Amazon Simple Queue Service (SQS) AWS Lambda, dan lainnya. Anda dapat menggunakan kode `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_SCHEDULED` acara untuk mengatur Amazon EventBridge sebelum tambalan mesin tersedia. Anda dapat mengatur EventBridge untuk menanggapi peristiwa dan melakukan tindakan otomatis seperti menangkap informasi peristiwa, memulai acara tambahan, mengirim pemberitahuan melalui saluran tambahan seperti pemberitahuan push ke, dan mengambil tindakan korektif atau tindakan lainnya AWS Console Mobile Application, ketika patch mesin Amazon DocumentDB tersedia di wilayah Anda.

Dalam skenario langka Amazon DocumentDB membatalkan patch mesin, Anda akan menerima pemberitahuan AHD serta email yang memberi tahu Anda tentang pembatalan. Dengan demikian, Anda dapat menggunakan kode `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_CANCELLED` acara untuk mengatur Amazon EventBridge untuk menanggapi acara ini. Lihat Panduan EventBridge Pengguna Amazon untuk mempelajari lebih lanjut tentang menggunakan EventBridge aturan Amazon.

Melihat tindakan pemeliharaan Amazon DocumentDB yang tertunda

Anda dapat melihat apakah pembaruan pemeliharaan tersedia untuk klaster Anda dengan menggunakan AWS Management Console atau file AWS CLI.

Jika pembaruan tersedia, Anda dapat melakukan salah satu hal berikut:

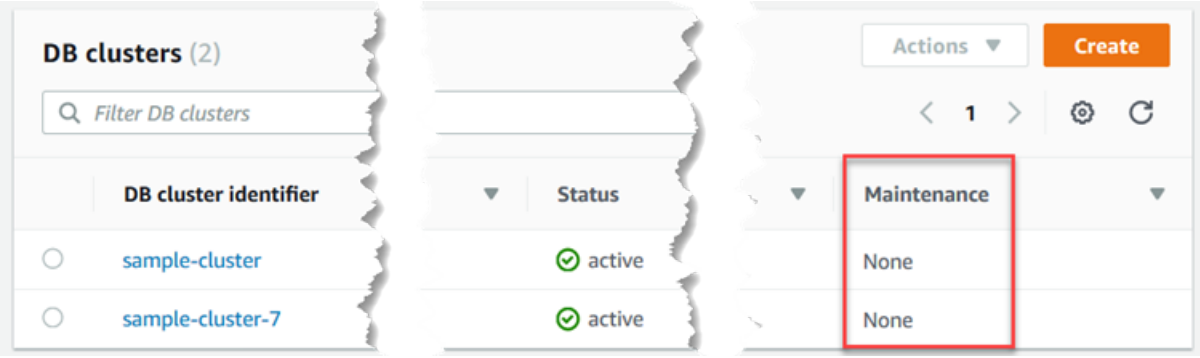
- Tunda tindakan pemeliharaan yang saat ini dijadwalkan untuk jendela pemeliharaan berikutnya (hanya untuk patch OS).
- Terapkan tindakan pemeliharaan segera.
- Jadwalkan tindakan pemeliharaan untuk dimulai selama periode pemeliharaan berikutnya.

> **Note**
>
> Jika Anda tidak mengambil tindakan, tindakan pemeliharaan yang diperlukan seperti tambalan engine akan diterapkan secara otomatis di jendela pemeliharaan terjadwal yang akan datang.

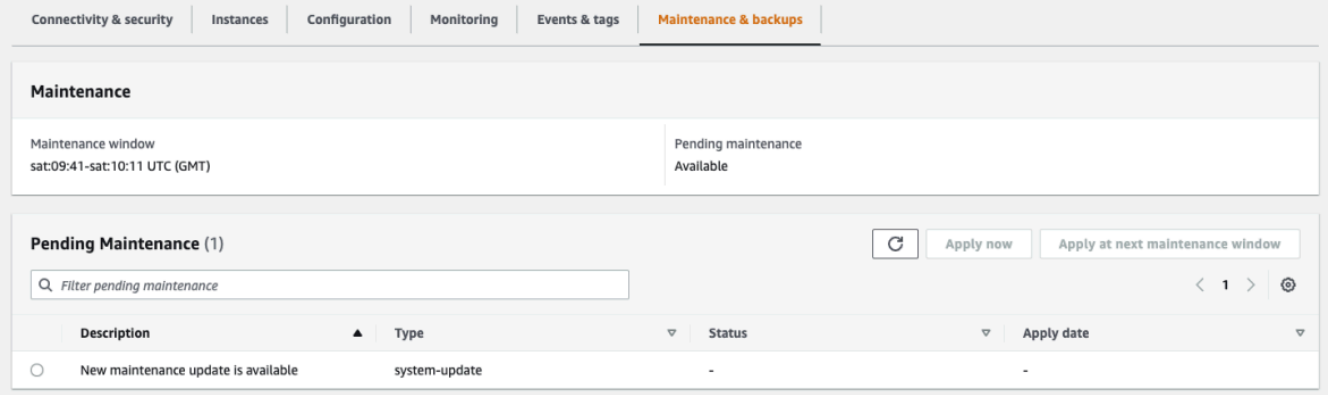
Jendela pemeliharaan menentukan kapan operasi yang tertunda dimulai, tetapi tidak membatasi total waktu pelaksanaan operasi ini.

Using the AWS Management Console

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Pada panel navigasi, silakan pilih Klaster.
3. Jika pembaruan tersedia, ini ditunjukkan dengan kata Tersedia, Diperlukan, atau Jendela Berikutnya di kolom Pemeliharaan untuk cluster di konsol Amazon DocumentDB, seperti yang ditunjukkan di sini:



4. Untuk mengambil tindakan, pilih cluster untuk menampilkan detailnya, lalu pilih Maintenance & backup. Item Pemeliharaan Tertunda muncul.



Using the AWS CLI

Gunakan AWS CLI operasi berikut untuk menentukan tindakan pemeliharaan apa yang tertunda. Output di sini menunjukkan tidak ada tindakan pemeliharaan yang tertunda.

```
aws docdb describe-pending-maintenance-actions
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "PendingMaintenanceActions": []
}
```

Menerapkan pembaruan mesin Amazon DocumentDB

Dengan Amazon DocumentDB, Anda dapat memilih kapan menerapkan operasi pemeliharaan. Anda dapat memutuskan kapan Amazon DocumentDB menerapkan pembaruan menggunakan atau. AWS Management Console AWS CLI

Gunakan prosedur dalam topik ini untuk segera meningkatkan atau menjadwalkan peningkatan untuk klaster Anda.

Using the AWS Management Console

Anda dapat menggunakan konsol untuk mengelola pembaruan untuk klaster Amazon DocumentDB Anda.

Untuk mengelola pembaruan untuk klaster

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Pada panel navigasi, silakan pilih Klaster.
3. Dalam daftar klaster, pilih tombol di samping nama klaster yang ingin Anda terapkan operasi pemeliharaannya.
4. Pada menu Tindakan, pilih salah satu dari berikut ini:
   - Perbarui sekarang untuk segera melakukan tugas pemeliharaan yang tertunda.
   - Perbarui di jendela berikutnya untuk melakukan tugas pemeliharaan yang tertunda selama jendela pemeliharaan klaster berikutnya.

   Atau, Anda dapat mengklik Terapkan sekarang atau Terapkan di jendela pemeliharaan berikutnya di bagian Pemeliharaan Tertunda pada tab Pemeliharaan & pencadangan cluster (lihat Menggunakan bagian sebelumnya AWS Management Console).

   > **Note**
   >
   > Jika tidak ada tugas pemeliharaan yang tertunda, semua opsi sebelumnya tidak aktif.

Using the AWS CLI

Untuk menerapkan pembaruan yang tertunda ke cluster, gunakan `apply-pending-maintenance-action` AWS CLI operasi.

Parameter-parameter

- **`--resource-identifier`**—Amazon DocumentDB Amazon Resource Name (ARN) dari sumber daya tempat tindakan pemeliharaan tertunda diterapkan.
- **`--apply-action`**—Tindakan pemeliharaan yang tertunda untuk diterapkan ke sumber daya ini.

  Nilai yang valid: `system-update` dan `db-upgrade`.
- **`--opt-in-type`**—Nilai yang menentukan jenis permintaan keikutsertaan, atau membatalkan permintaan keikutsertaan. Permintaan keikutsertaan tipe `immediate` tidak dapat diurungkan.

  Nilai yang valid:
  - `immediate`—Terapkan tindakan pemeliharaan segera.
  - `next-maintenance`—Terapkan tindakan pemeliharaan selama jendela pemeliharaan berikutnya untuk sumber daya.
  - `undo-opt-in`—Membatalkan permintaan keikutsertaan `next-maintenance` yang ada.

Example

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Untuk mengembalikan daftar sumber daya yang memiliki setidaknya satu pembaruan yang tertunda, gunakan `describe-pending-maintenance-actions` AWS CLI operasi.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

Untuk Windows:

```
aws docdb describe-pending-maintenance-actions ^
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "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"
                }
            ]
        }
    ]
}
```

Anda juga dapat mengembalikan daftar sumber daya untuk cluster dengan menentukan `--filters` parameter `describe-pending-maintenance-actions` AWS CLI operasi. Format untuk operasi `--filters` adalah `Name=`*`filter-name`*`,Values=`*`resource-id`*`,...`.

`db-cluster-id`adalah nilai yang dapat diterima untuk `Name` parameter filter. Nilai ini menerima daftar pengidentifikasi cluster atau ARN. Daftar yang dikembalikan hanya mencakup tindakan pemeliharaan yang tertunda untuk klaster yang diidentifikasi oleh pengidentifikasi atau ARN ini.

Contoh berikut mengembalikan tindakan pemeliharaan tertunda untuk klaster `sample-cluster1` dan `sample-cluster2`.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-pending-maintenance-actions \
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Untuk Windows:

```
aws docdb describe-pending-maintenance-actions ^
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Terapkan tanggal

Setiap tindakan pemeliharaan memiliki tanggal penerapan masing-masing yang dapat Anda temukan saat menjelaskan tindakan pemeliharaan yang tertunda. Saat Anda membaca output tindakan pemeliharaan yang tertunda dari AWS CLI, tiga tanggal dicantumkan:

- **`CurrentApplyDate`**—Tanggal tindakan pemeliharaan akan diterapkan baik segera atau selama jendela pemeliharaan berikutnya. Jika pemeliharaan bersifat opsional, nilai ini dapat berupa `null`.
- **`ForcedApplyDate`**—Tanggal saat pemeliharaan akan diterapkan secara otomatis, terlepas dari jangka waktu pemeliharaan Anda.
- **`AutoAppliedAfterDate`**—Tanggal setelah pemeliharaan akan diterapkan selama jendela pemeliharaan klaster.

Pembaruan yang diprakarsai pengguna

Sebagai pengguna Amazon DocumentDB, Anda dapat memulai pembaruan untuk klaster atau instans Anda. Misalnya, Anda dapat mengubah kelas instans menjadi kelas dengan memori lebih banyak atau lebih sedikit, atau Anda dapat mengubah grup parameter klaster. Amazon DocumentDB melihat perubahan ini secara berbeda dari pembaruan yang dimulai Amazon DocumentDB. Untuk informasi selengkapnya tentang memodifikasi klaster atau instans, lihat berikut ini:

- Memodifikasi cluster Amazon DocumentDB

- Memodifikasi instance Amazon DocumentDB

Untuk melihat daftar modifikasi yang dimulai oleh pengguna yang tertunda, jalankan perintah berikut.

Example

Untuk melihat perubahan yang dimulai oleh pengguna yang tertunda untuk instans Anda

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

Untuk Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

Output dari operasi ini terlihat seperti berikut (format JSON).

Dalam hal ini, `sample-cluster-instance` memiliki perubahan yang tertunda ke kelas instans `db.r5.xlarge`, sedangkan `sample-cluster-instance-2` tidak memiliki perubahan yang tertunda.

```
[
    [
        "sample-cluster",
        "sample-cluster-instance",
        {
            "DBInstanceClass": "db.r5.xlarge"
        }
    ],
    [
        "sample-cluster",
        "sample-cluster-instance-2",
        {}
    ]
]
```

Mengelola jendela pemeliharaan Amazon DocumentDB Anda

Setiap instans dan klaster memiliki jendela pemeliharaan mingguan selama perubahan yang tertunda diterapkan. Jendela pemeliharaan adalah kesempatan untuk mengendalikan kapan modifikasi dan patch perangkat lunak terjadi, jika diminta atau diperlukan. Jika peristiwa pemeliharaan dijadwalkan untuk minggu tertentu, itu dilakukan selama periode pemeliharaan 30 menit yang Anda identifikasi. Sebagian besar peristiwa pemeliharaan juga selesai selama jendela pemeliharaan 30 menit, meskipun peristiwa pemeliharaan yang lebih besar mungkin memerlukan waktu lebih dari 30 menit untuk diselesaikan.

Jendela pemeliharaan 30 menit dipilih secara acak dari blok waktu 8 jam per Wilayah. Jika Anda tidak menentukan jendela pemeliharaan pilihan saat membuat instans atau klaster, Amazon DocumentDB menugaskan jendela pemeliharaan 30 menit pada hari yang dipilih secara acak dalam seminggu.

Tabel berikut mencantumkan blok waktu untuk setiap Wilayah dari mana jendela pemeliharaan default ditugaskan.

Nama Wilayah	Wilayah	Blok Waktu UTC
AS Timur (Ohio)	us-east-2	03:00-11:00
AS Timur (Virginia Utara)	us-east-1	03:00-11:00
AS Barat (Oregon)	us-west-2	06:00-14:00
Asia Pasifik (Hong Kong)	ap-east-1	06:00-14:00
Asia Pasifik (Hyderabad)	ap-south-2	06:30 — 14:30
Asia Pasifik (Mumbai)	ap-south-1	06:00-14:00
Asia Pasifik (Seoul)	ap-northeast-2	13:00-21:00
Asia Pasifik (Singapura)	ap-southeast-1	14:00-22:00
Asia Pasifik (Sydney)	ap-southeast-2	12:00-20:00
Asia Pasifik (Tokyo)	ap-northeast-1	13:00-21:00
Kanada (Pusat)	ca-central-1	03:00-11:00

Nama Wilayah	Wilayah	Blok Waktu UTC
Tiongkok (Beijing)	cn-north-1	06:00-14:00
Tiongkok (Ningxia)	cn-northwest-1	06:00-14:00
Eropa (Frankfurt)	eu-central-1	21:00-05:00
Eropa (Irlandia)	eu-west-1	22:00-06:00
Eropa (London)	eu-west-2	22:00-06:00
Eropa (Milan)	eu-south-1	02:00-10:00
Eropa (Paris)	eu-west-3	23:59-07:29
Timur Tengah (UEA)	me-central-1	05:00 — 13:00
Amerika Selatan (Sao Paulo)	sa-east-1	00:00-08:00
AWS GovCloud (AS-Timur)	us-gov-east-1	17:00-01:00
AWS GovCloud (AS-Barat)	us-gov-west-1	06:00-14:00

Mengubah jendela pemeliharaan Amazon DocumentDB Anda

Waktu pemeliharaan harus jatuh pada saat penggunaan terendah dan dengan demikian mungkin perlu diubah dari waktu ke waktu. Klaster atau instans Anda tidak tersedia selama waktu ini hanya jika perubahan sistem (seperti operasi penyimpanan skala atau perubahan kelas instans) sedang diterapkan dan memerlukan penghentian. Dan kemudian itu tidak tersedia hanya untuk jumlah waktu minimum yang diperlukan untuk membuat perubahan yang diperlukan.

Untuk pembaruan ke mesin basis data, Amazon DocumentDB menggunakan jendela pemeliharaan pilihan klaster dan bukan jendela pemeliharaan untuk instans individual.

Mengubah waktu pemeliharaan

- Untuk sebuah klaster, lihat Memodifikasi cluster Amazon DocumentDB.
- Untuk sebuah instans, lihat Memodifikasi instance Amazon DocumentDB.

Bekerja dengan pembaruan sistem operasi

Instans di cluster Amazon DocumentDB terkadang memerlukan pembaruan sistem operasi. Amazon DocumentDB meningkatkan sistem operasi ke versi yang lebih baru untuk meningkatkan kinerja database dan postur keamanan pelanggan secara keseluruhan. Pembaruan sistem operasi tidak mengubah versi mesin cluster atau kelas instans dari instans Amazon DocumentDB.

Sebaiknya Anda memperbarui instance pembaca di klaster terlebih dahulu, lalu instance penulis untuk memaksimalkan ketersediaan klaster Anda. Kami tidak menyarankan untuk memperbarui instance pembaca dan penulis secara bersamaan, karena Anda mungkin mengalami downtime yang lebih lama jika terjadi failover.

Pembaruan sistem operasi tidak memiliki tanggal berlaku dan dapat diterapkan kapan saja. Kami menyarankan Anda menerapkannya secara berkala untuk memperbarui basis data Amazon DocumentDB Anda. Amazon DocumentDB tidak menerapkan pembaruan ini secara otomatis. Untuk diberi tahu saat pembaruan opsional baru tersedia, Anda dapat berlangganan RDS-EVENT-0230 dalam kategori acara penambalan keamanan. Untuk informasi tentang berlangganan acara Amazon DocumentDB, lihat Berlangganan Langganan Acara Amazon DocumentDB.

Anda harus mengharapkan bahwa ketika pemeliharaan dilakukan pada klaster atau instans Anda, jika instans adalah instans primer, itu akan gagal. Untuk meningkatkan ketersediaan, sebaiknya gunakan lebih dari satu instans untuk klaster Amazon DocumentDB Anda. Untuk informasi selengkapnya, lihat Failover Amazon DocumentDB.

> **Note**
>
> Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS).

> **Important**
>
> Instans Amazon DocumentDB Anda akan offline selama peningkatan sistem operasi.

> **Note**
>
> Tetap mengikuti semua pembaruan opsional dan wajib mungkin diperlukan untuk memenuhi berbagai kewajiban kepatuhan. Kami menyarankan Anda menerapkan semua pembaruan yang disediakan oleh Amazon DocumentDB secara rutin selama jendela pemeliharaan Anda.

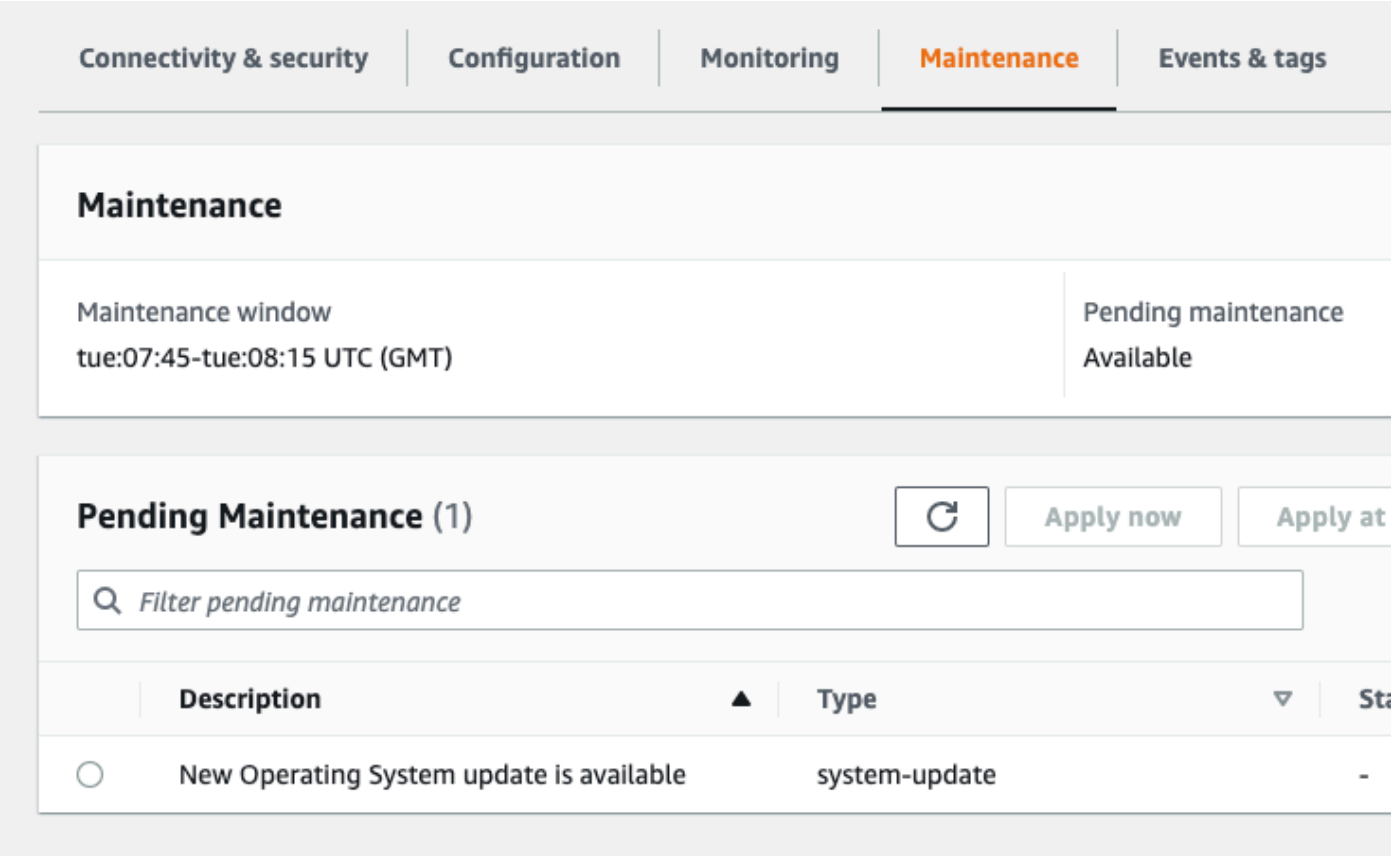
Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menentukan apakah pembaruan bersifat opsional atau wajib.

Using the AWS Management Console

Untuk menentukan apakah pembaruan bersifat opsional atau wajib menggunakan AWS Management Console:

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Clusters, lalu pilih instance.
3. Pilih Pemeliharaan.
4. Di bagian Pending Maintenance, temukan pembaruan sistem operasi, dan periksa nilai Status.

Di dalam AWS Management Console, pembaruan sistem operasi memiliki Status pemeliharaan disetel ke tersedia dan tidak memiliki tanggal Terapkan, seperti yang ditunjukkan pada gambar berikut:



Anda dapat memilih pembaruan sistem operasi dan klik Terapkan sekarang atau Terapkan di jendela pemeliharaan berikutnya di bagian Pemeliharaan Tertunda. Jika nilai pemeliharaan adalah jendela berikutnya, tunda item pemeliharaan dengan memilih Defer upgrade. Anda tidak dapat menunda tindakan pemeliharaan jika sudah dimulai.

Atau, Anda dapat memilih instance dari daftar cluster dengan mengklik Cluster di panel navigasi dan pilih Terapkan sekarang atau Terapkan di jendela maintentnace berikutnya dari menu Tindakan.

Using the AWS CLI

Untuk menentukan apakah pembaruan bersifat opsional atau wajib menggunakan AWS CLI, panggil describe-pending-maintenance-actions perintah:

```
aws docdb describe-pending-maintenance-actions
```

Pembaruan sistem operasi wajib mencakup nilai `AutoAppliedAfterDate` dan nilai `CurrentApplyDate`. Pembaruan sistem operasi opsional tidak mencakup nilai-nilai ini.

Output berikut menunjukkan pembaruan sistem operasi wajib:

```
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
The following output shows an optional operating system update.
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Ketersediaan pembaruan sistem operasi

Pembaruan sistem operasi khusus untuk versi mesin Amazon DocumentDB dan kelas instance. Oleh karena itu, instans Amazon DocumentDB menerima atau memerlukan pembaruan pada waktu yang berbeda. Ketika pembaruan sistem operasi tersedia untuk instans Anda berdasarkan versi mesin dan kelas instans, pembaruan muncul di konsol. Hal ini juga dapat dilihat dengan menjalankan AWS CLI `describe-pending-maintenance-actions` perintah atau dengan memanggil operasi `DescribePendingMaintenanceActions` API. Jika pembaruan tersedia untuk instans Anda, Anda dapat memperbarui sistem operasi Anda dengan mengikuti petunjuk dalam Menerapkan Pembaruan Amazon DocumentDB.

Memahami Peran Tertaut Layanan

Amazon DocumentDB (dengan kompatibilitas MongoDB) menggunakan peran tartaut layanan AWS Identity and Access Management (IAM). Peran tertaut layanan adalah jenis IAM role unik yang tertaut secara langsung ke Amazon DocumentDB. Peran tertaut layanan ditentukan sebelumnya oleh

Amazon DocumentDB dan mencakup semua izin yang diperlukan oleh layanan untuk memanggil layanan AWS lainnya atas nama Anda.

Peran tertaut layanan membuat penggunaan Amazon DocumentDB lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Amazon DocumentDB menentukan izin peran tertaut layanannya, dan kecuali ditentukan lain, hanya Amazon DocumentDB yang dapat menjalankan perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran hanya setelah menghapus sumber daya yang tertaut lebih dahulu. Ini melindungi sumber daya Amazon DocumentDB karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran tertaut layanan, lihat Layanan yang Bekerja dengan IAM AWS dan mencari layanan yang memiliki opsi Ya di kolom Peran Tertaut Layanan. Pilih Yes (Ya) bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin Peran Tertaut Layanan untuk Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) menggunakan peran terkait layanan bernama AWSServiceRoleForRDSuntuk memungkinkan Amazon DocumentDB memanggil AWS layanan atas nama klaster Anda.

AWSServiceRoleForRDS peran terkait layanan memercayakan layanan berikut untuk menjalankan peran tersebut:

- `docdb.amazonaws.com`

Kebijakan izin peran mengizinkan Amazon DocumentDB untuk menyelesaikan tindakan berikut ini pada sumber daya yang ditentukan:

- Tindakan pada `ec2`:
  - `AssignPrivateIpAddresses`
  - `AuthorizeSecurityGroupIngress`
  - `CreateNetworkInterface`
  - `CreateSecurityGroup`
  - `DeleteNetworkInterface`

  - `DeleteSecurityGroup`
  - `DescribeAvailabilityZones`
  - `DescribeInternetGateways`
  - `DescribeSecurityGroups`
  - `DescribeSubnets`
  - `DescribeVpcAttribute`
  - `DescribeVpcs`
  - `ModifyNetworkInterfaceAttribute`
  - `RevokeSecurityGroupIngress`
  - `UnassignPrivateIpAddresses`
- Tindakan pada `sns`:
  - `ListTopic`
  - `Publish`
- Tindakan pada `cloudwatch`:
  - `PutMetricData`
  - `GetMetricData`
  - `CreateLogStream`
  - `PullLogEvents`
  - `DescribeLogStreams`
  - `CreateLogGroup`

> **Note**
>
> Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Anda mungkin menemukan pesan kesalahan berikut ini:
> Tidak dapat membuat sumber daya. Verifikasi bahwa Anda memiliki izin untuk membuat peran layanan terhubung. Jika tidak, tunggulah dan coba lagi nanti.
> Jika Anda melihat kesalahan ini, pastikan Anda mengaktifkan izin berikut ini:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
> ```

```
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"rds.amazonaws.com"
        }
    }
}
```

Untuk informasi lebih lanjut, lihat Izin Peran Tertaut Layanan di Panduan Pengguna IAM.

Membuat Peran Tertaut Layanan Amazon DocumentDB

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat klaster, Amazon DocumentDB membuat peran tertaut layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, lalu ingin membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuat ulang peran tersebut di akun Anda. Saat Anda membuat klaster, Amazon DocumentDB membuat peran tertaut layanan untuk Anda.

Memodifikasi Peran Tertaut Layanan Amazon DocumentDB

Amazon DocumentDB tidak memungkinkan Anda memodifikasi peran terkait AWSServiceRoleForRDS layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun demikian, Anda dapat memodifikasi penjelasan peran menggunakan IAM. Untuk informasi lebih lanjut, lihat Mengedit Peran Tertaut Layanan di Panduan Pengguna IAM.

Menghapus Peran Tertaut Layanan Amazon DocumentDB

Jika Anda tidak lagi perlu menggunakan fitur atau layanan yang memerlukan peran tertaut layanan, sebaiknya hapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Namun demikian, Anda harus menghapus semua klaster Anda sebelum Anda dapat menghapus peran tertaut layanan.

Membersihkan Peran Tertaut Layanan Amazon DocumentDB

Sebelum Anda dapat menggunakan IAM untuk menghapus peran terkait layanan, Anda harus mengonfirmasi terlebih dahulu bahwa peran tersebut tidak memiliki sesi aktif dan menghapus sumber daya yang digunakan oleh peran tersebut.

Untuk memeriksa apakah peran tertaut layanan memiliki sesi aktif menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol IAM pada https://console.aws.amazon.com/iam/.
2. Di panel navigasi konsol IAM, pilih Peran, lalu pilih nama (bukan kotak centang) AWSServiceRoleForRDSperan.
3. Pada halaman Ringkasan untuk peran yang dipilih, pilih tab Penasihat Akses.
4. Di tab Penasihat Akses, tinjau aktivitas terbaru untuk peran tertaut layanan.

> **Note**
>
> Jika Anda tidak yakin apakah Amazon DocumentDB menggunakan AWSServiceRoleForRDS peran tersebut, Anda dapat mencoba menghapus peran tersebut. Jika layanan menggunakan peran tersebut, maka penghapusan akan gagal dan Anda dapat melihat Region tempat peran tersebut digunakan. Jika peran tersebut sedang digunakan, Anda harus menunggu hingga sesi ini berakhir sebelum dapat menghapus peran tersebut. Anda tidak dapat mencabut sesi untuk peran terkait layanan.

Jika Anda ingin menghapus AWSServiceRoleForRDS peran, Anda harus menghapus semua instans dan cluster Anda terlebih dahulu. Untuk informasi tentang menghapus instans dan klaster, lihat topik berikut ini:

- Menghapus instans Amazon DocumentDB
- Menghapus cluster Amazon DocumentDB

Wilayah yang Didukung untuk Peran Tertaut Layanan Amazon DocumentDB

Amazon DocumentDB mendukung menggunakan peran tertaut layanan di semua Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html#regions-and-azs-availability.

Menggunakan cluster elastis Amazon DocumentDB

Cluster elastis Amazon DocumentDB mendukung beban kerja dengan jutaan baca/tulis per detik dan kapasitas penyimpanan petabyte. Cluster elastis juga menyederhanakan cara pengembang berinteraksi dengan Amazon DocumentDB dengan menghilangkan kebutuhan untuk memilih, mengelola, atau meningkatkan instance.

Cluster elastis Amazon DocumentDB dibuat untuk:

- Berikan solusi bagi pelanggan yang mencari database yang menyediakan skala hampir tak terbatas dengan kemampuan kueri yang kaya dan kompatibilitas API MongoDB.
- Berikan pelanggan batas koneksi yang lebih tinggi, dan untuk mengurangi downtime dari patching.
- Terus berinvestasi dalam arsitektur cloud-native, elastis, dan terkemuka di kelas untuk beban kerja JSON.

Topik

- Kasus penggunaan cluster elastis
- Keuntungan dari cluster elastis
- Wilayah cluster elastis dan ketersediaan versi
- Batasan
- Cluster elastis Amazon DocumentDB: cara kerjanya
- Memulai dengan cluster elastis Amazon DocumentDB
- Praktik terbaik
- Mengelola cluster elastis
- Enkripsi data saat istirahat untuk klaster elastis Amazon DocumentDB
- Peran terkait layanan dalam kelompok elastis

Kasus penggunaan cluster elastis

Basis data dokumen berguna untuk beban kerja yang memerlukan skema fleksibel untuk pengembangan yang cepat dan berulang. Misalnya kasus penggunaan Amazon DocumentDB, lihat. Kasus Penggunaan Basis Data Dokumen

Berikut ini adalah beberapa contoh kasus penggunaan di mana cluster elastis dapat memberikan keuntungan yang signifikan:

Profil pengguna

Karena database dokumen memiliki skema yang fleksibel, mereka dapat menyimpan dokumen yang memiliki atribut dan nilai data yang berbeda dalam skala besar. Cluster elastis adalah solusi praktis untuk profil online di mana pengguna yang berbeda memberikan berbagai jenis informasi. Misalkan aplikasi Anda mendukung ratusan juta profil pengguna. Anda dapat menggunakan cluster elastis untuk mendukung aplikasi tersebut karena mereka dapat ditingkatkan dan keluar untuk mendukung jutaan penulisan dan pembacaan ke profil pengguna ini. Anda juga dapat menurunkan skala untuk jam sibuk untuk mengurangi biaya.

Manajemen konten dan catatan sejarah

Untuk mengelola konten secara efektif, Anda harus dapat mengumpulkan dan mengagregasikan konten dari berbagai sumber, dan kemudian mengirimkannya ke pelanggan. Akibat skema fleksibel mereka, basis data dokumen sempurna untuk mengumpulkan dan menyimpan jenis data apa pun. Anda dapat menggunakannya untuk membuat dan menggabungkan jenis konten baru, termasuk konten buatan pengguna seperti gambar, komentar, dan video. Seiring waktu, database Anda mungkin memerlukan lebih banyak penyimpanan. Dengan cluster elastis, Anda dapat mendistribusikan data Anda ke lebih banyak volume penyimpanan yang memungkinkan Anda menyimpan petabyte data dalam satu cluster.

Keuntungan dari cluster elastis

AWS Integrasi layanan

Cluster elastis Amazon DocumentDB terintegrasi dengan layanan AWS lain dengan cara yang sama seperti Amazon DocumentDB:

- Migrasi - Anda dapat menggunakan AWS Database Migration Service (DMS) untuk bermigrasi dari MongoDB dan database relasional lainnya ke cluster elastis Amazon DocumentDB.
- Pemantauan - Anda dapat memantau kesehatan dan kinerja cluster elastis Anda menggunakan Amazon CloudWatch.

- Keamanan - Anda dapat mengatur otentikasi dan otorisasi melalui AWS Identity and Access Management (IAM) untuk mengelola cluster elastis Anda dan menggunakan Amazon VPC untuk koneksi VPC yang aman saja.
- Manajemen data - Anda dapat menggunakan AWS Glue untuk mengimpor dan mengekspor data dari/ke AWS layanan lain seperti Amazon S3, Amazon Redshift, dan Amazon Service. OpenSearch

Wilayah cluster elastis dan ketersediaan versi

Ketersediaan wilayah

Tabel berikut menunjukkan AWS wilayah tempat cluster elastis Amazon DocumentDB saat ini tersedia dan titik akhir untuk setiap wilayah.

Nama Wilayah	Wilayah	Zona ketersediaan
AS Timur (Virginia Utara)	`us-east-1`	5
AS Timur (Ohio)	`us-east-2`	3
AS Barat (Oregon)	`us-west-2`	3
Asia Pasifik (Mumbai)	`ap-south-1`	3
Asia Pasifik (Seoul)	`ap-northeast-2`	3
Asia Pasifik (Singapura)	`ap-southeast-1`	3
Asia Pasifik (Sydney)	`ap-southeast-2`	3
Asia Pasifik (Tokyo)	`ap-northeast-1`	3
Amerika Selatan (Sao Paulo)	`sa-east-1`	3
Eropa (Frankfurt)	`eu-central-1`	3
Eropa (Irlandia)	`eu-west-1`	3
Eropa (London)	`eu-west-2`	3

Ketersediaan versi

Cluster elastis mendukung protokol kabel MongoDB 5.0-compatable. Untuk perbedaan antara cluster berbasis instans DocumentDB 4.0 dan cluster elastis, lihat. Perbedaan fungsional antara Amazon DocumentDB 4.0 dan cluster elastis

Batasan

Manajemen cluster elastis

Fitur dan kemampuan manajemen klaster berikut tidak didukung dalam rilis ini:

- Kemampuan untuk membuat cluster global
- Acara Amazon DocumentDB yang ada dan berlangganan acara
- Rentang sharding
- Koleksi Shard yang ada
- Kunci pecahan multi-bidang
- Ubah kunci pecahan
- oint-in-time Pulihkan P
- Kloning
- Wawasan Performa

> **Note**
>
> Untuk informasi tentang batas cluster elastis, lihatKuota dan Batas Amazon DocumentDB.

Operasi kueri dan tulis

Perintah dan kemampuan operasi query dan write berikut tidak didukung dalam rilis ini:

- Perintah DDL selama operasi penskalaan
- Profiler
- Grup parameter

- AWS Config
- AWS Backup

Pengumpulan dan manajemen indeks

Fitur manajemen koleksi dan indeks berikut tidak didukung dalam rilis ini:

- Indeks geospasial
- Indeks latar belakang membuat

Administrasi dan diagnostik

Perintah dan kemampuan administrasi dan diagnostik berikut tidak didukung dalam rilis ini:

- AWS Secrets Manager
- R ole-based-access-control (RBAC) peran khusus.
- Saat menghubungkan, tulis kekhawatiran 0 tidak didukung.
- Mengubah subnet milik VPC yang saat ini tidak ditetapkan ke cluster elastis yang ada.

Fitur keikutsertaan

Fitur keikutsertaan Amazon DocumentDB berikut tidak didukung dalam rilis ini:

- Transaksi ACID
- Audit DDL/DML
- Mengelola aliran
- Perintah sesi

Cluster elastis Amazon DocumentDB: cara kerjanya

Topik di bagian ini memberikan informasi tentang mekanisme dan fungsi yang mendukung cluster elastis Amazon DocumentDB.

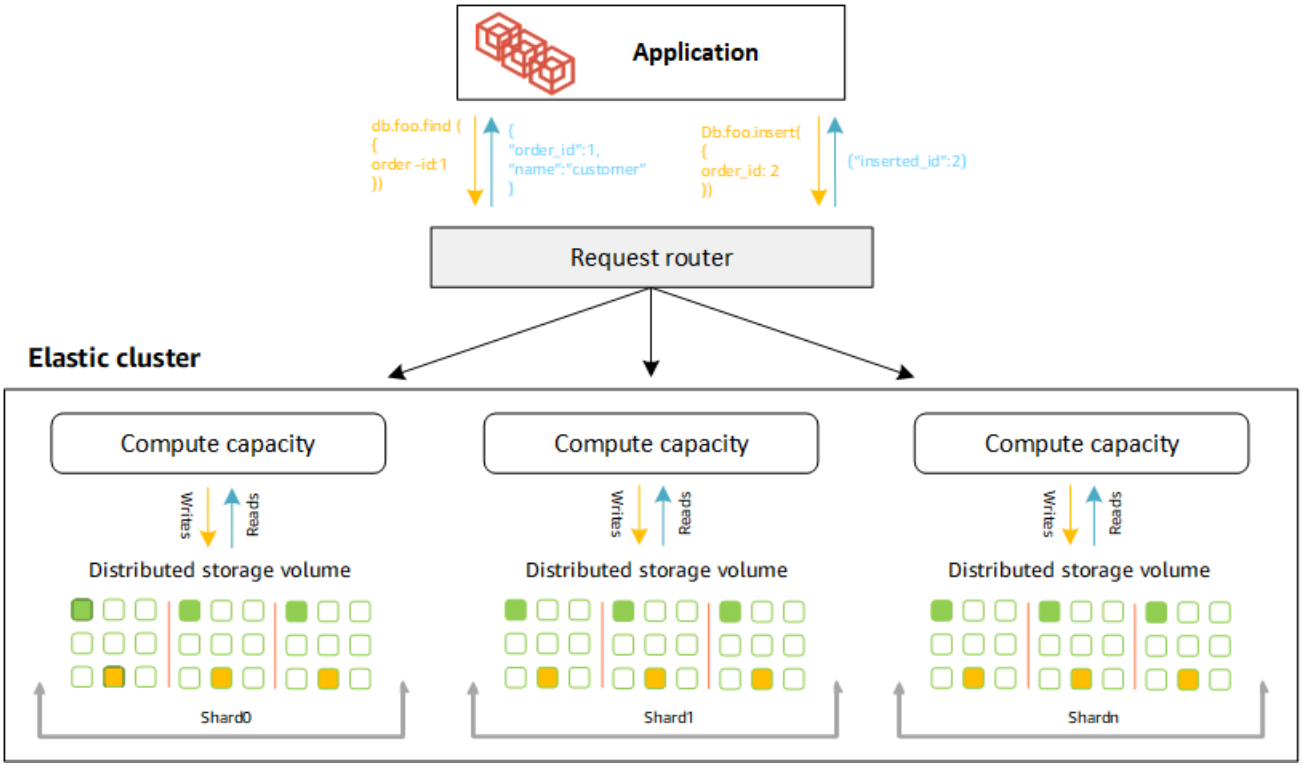
Topik

- Sharding cluster elastis Amazon DocumentDB

- Migrasi cluster elastis
- Penskalaan cluster elastis
- Keandalan cluster elastis
- Penyimpanan dan ketersediaan cluster elastis
- Perbedaan fungsional antara Amazon DocumentDB 4.0 dan cluster elastis

Sharding cluster elastis Amazon DocumentDB

Cluster elastis Amazon DocumentDB menggunakan sharding berbasis hash untuk mempartisi data di seluruh sistem penyimpanan terdistribusi. Sharding, juga dikenal sebagai partisi, membagi kumpulan data besar menjadi kumpulan data kecil di beberapa node yang memungkinkan Anda untuk skala database Anda di luar batas penskalaan vertikal. Cluster elastis menggunakan pemisahan, atau “decoupling”, komputasi dan penyimpanan di Amazon DocumentDB yang memungkinkan Anda untuk menskalakan secara independen satu sama lain. Alih-alih mempartisi ulang koleksi dengan memindahkan potongan kecil data antara node komputasi, cluster elastis menyalin data secara efisien dalam sistem penyimpanan terdistribusi.



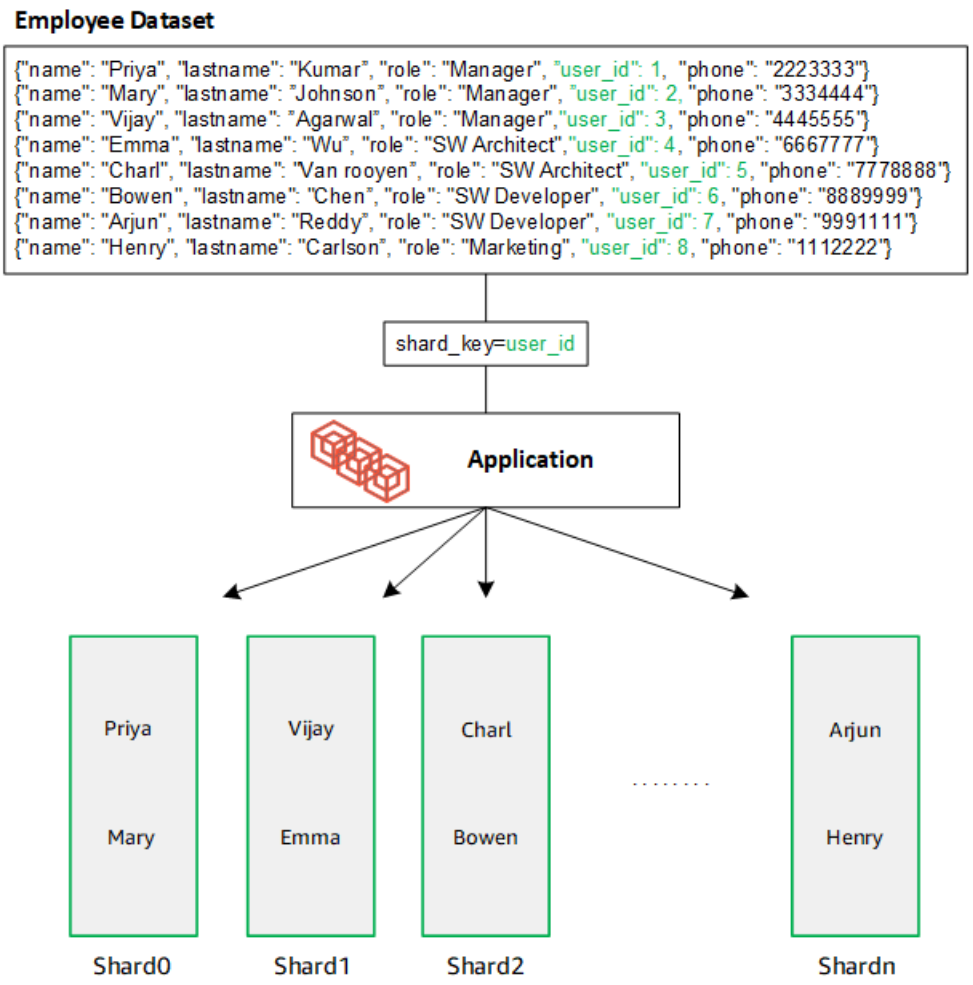
Definisi pecahan

Definisi nomenklatur pecahan:

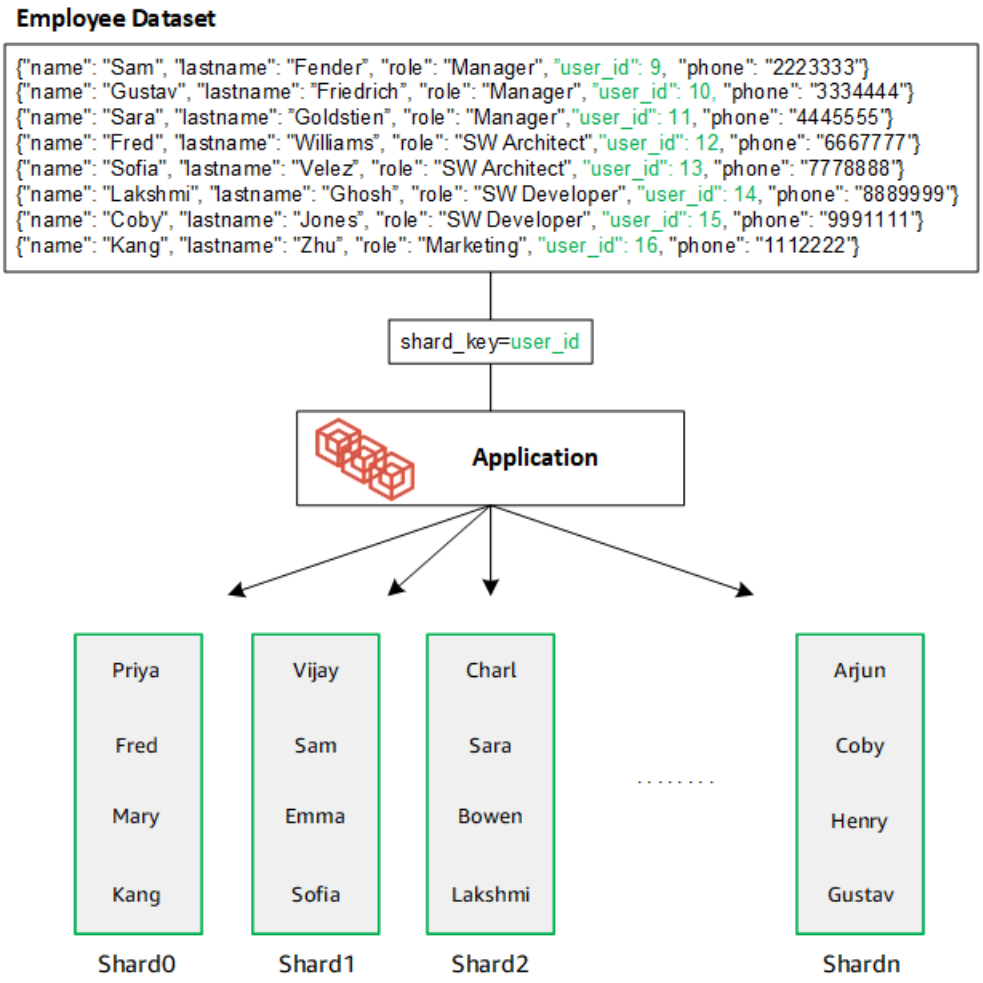
- Pecahan — Pecahan menyediakan komputasi untuk cluster elastis. Sebuah shard secara default akan memiliki dua node. Anda dapat mengonfigurasi maksimum 32 pecahan dan setiap pecahan dapat memiliki maksimum 64 vCPU.
- Kunci pecahan — Kunci pecahan adalah bidang wajib dalam dokumen JSON Anda dalam koleksi sharded yang digunakan cluster elastis untuk mendistribusikan lalu lintas baca dan tulis ke pecahan yang cocok.
- Koleksi pecahan — Koleksi pecahan adalah koleksi yang datanya didistribusikan di seluruh cluster elastis di partisi data.
- Partisi — Partisi adalah bagian logis dari data sharded. Saat Anda membuat koleksi sharded, data diatur ke dalam partisi dalam setiap shard secara otomatis berdasarkan kunci shard. Setiap pecahan memiliki beberapa partisi.

Mendistribusikan data di seluruh pecahan yang dikonfigurasi

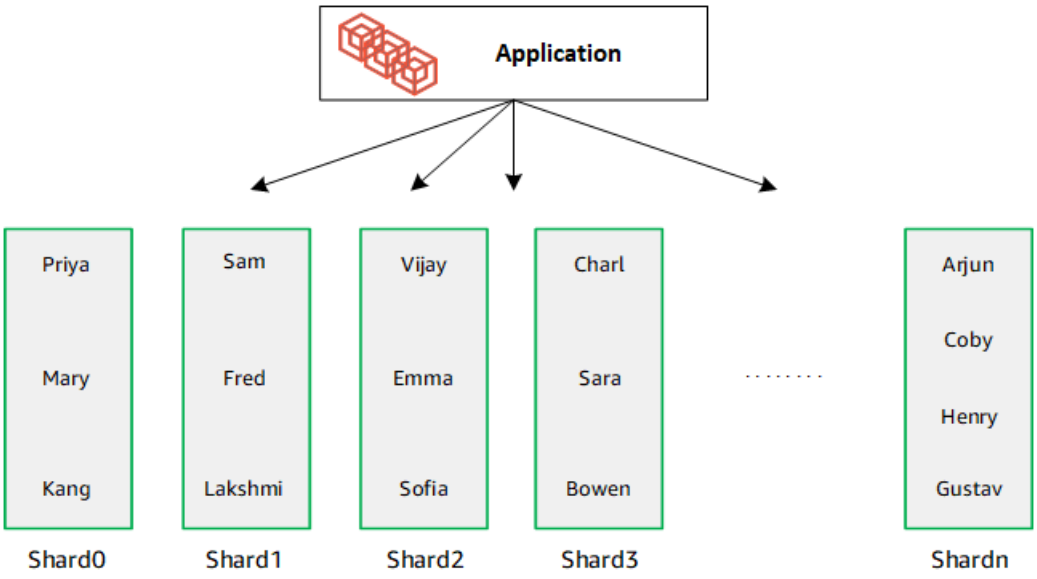
Buat kunci pecahan yang memiliki banyak nilai unik. Kunci shard yang baik akan mempartisi data Anda secara merata di seluruh pecahan yang mendasarinya, memberikan beban kerja Anda throughput dan kinerja terbaik. Contoh berikut adalah data nama karyawan yang menggunakan kunci shard bernama “user_id”:



DocumentDB menggunakan hash sharding untuk mempartisi data Anda di seluruh pecahan yang mendasarinya. Data tambahan dimasukkan dan didistribusikan dengan cara yang sama:



Saat Anda menskalakan database dengan menambahkan pecahan tambahan, Amazon DocumentDB secara otomatis mendistribusikan ulang data:



Migrasi cluster elastis

Amazon DocumentDB mendukung migrasi data sharded MongoDB ke cluster elastis. Metode migrasi offline, online, dan hibrida didukung. Untuk informasi selengkapnya, lihat Migrasi ke Amazon DocumentDB.

Penskalaan cluster elastis

Amazon DocumentDB Elastic Clusters menyediakan kemampuan untuk meningkatkan jumlah pecahan (skala keluar) di cluster elastis Anda, dan jumlah vCPU yang diterapkan pada setiap pecahan (skala naik). Anda juga dapat mengurangi jumlah pecahan dan kapasitas komputasi (vCPU) sesuai kebutuhan.

Untuk praktik terbaik penskalaan, lihatMenskalakan klaster.

> **Note**
>
> Penskalaan tingkat cluster juga tersedia. Untuk informasi selengkapnya, lihat Menskalakan cluster Amazon DocumentDB.

Keandalan cluster elastis

Amazon DocumentDB dirancang agar andal, tahan lama, dan toleran terhadap kesalahan. Untuk meningkatkan ketersediaan, cluster elastis menyebarkan dua node per shard yang ditempatkan di Availability Zone yang berbeda. Amazon DocumentDB menyertakan beberapa fitur otomatis yang menjadikannya solusi basis data yang andal. Untuk informasi selengkapnya, lihat Keandalan Amazon DocumentDB.

Penyimpanan dan ketersediaan cluster elastis

Data Amazon DocumentDB disimpan dalam volume klaster, yang merupakan volume virtual tunggal yang menggunakan solid state drive (SSD). Volume cluster terdiri dari enam salinan data Anda, yang direplikasi secara otomatis di beberapa Availability Zone dalam satu AWS Region. Replikasi ini membantu memastikan bahwa data Anda sangat berdaya tahan, dengan kemungkinan kehilangan data yang lebih kecil. Ini juga membantu memastikan bahwa klaster Anda lebih tersedia selama failover karena salinan data Anda sudah ada di Availability Zone lainnya. Untuk detail lebih lanjut tentang penyimpanan, ketersediaan tinggi dan replikasi lihatAmazon DocumentDB: Cara Kerjanya.

Perbedaan fungsional antara Amazon DocumentDB 4.0 dan cluster elastis

Perbedaan fungsional berikut ada antara Amazon DocumentDB 4.0 dan cluster elastis.

- Hasil dari `top` dan `collStats` dipartisi oleh pecahan. Untuk koleksi sharded, data didistribusikan di antara beberapa partisi dan `collStats` laporan yang dikumpulkan `collScans` dari partisi.
- Statistik koleksi dari `top` dan `collStats` untuk koleksi sharded diatur ulang saat jumlah pecahan cluster diubah.
- Peran bawaan cadangan sekarang mendukung`serverStatus`. Tindakan - Pengembang dan aplikasi dengan peran cadangan dapat mengumpulkan statistik tentang keadaan cluster Amazon DocumentDB.
- `SecondaryDelaySecs`Bidang menggantikan `slaveDelay` `replSetGetConfig` output.
- `hello`Perintah menggantikan `isMaster` - `hello` mengembalikan dokumen yang menggambarkan peran cluster elastis.
- `$elemMatch`Operator dalam cluster elastis hanya cocok dengan dokumen di tingkat bersarang pertama dari sebuah array. Di Amazon DocumentDB 4.0, operator melintasi semua tingkatan sebelum mengembalikan dokumen yang cocok. Sebagai contoh:

```
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 ] } ] ] }
```

- Proyeksi “$” di Amazon DocumentDB 4.0 mengembalikan semua dokumen dengan semua bidang. Dengan cluster elastis, `find` perintah dengan proyeksi “$” mengembalikan dokumen yang cocok dengan parameter kueri yang hanya berisi bidang yang cocok dengan proyeksi “$”.
- Dalam cluster elastis, `find` perintah dengan `$regex` dan parameter `$options` kueri mengembalikan kesalahan: “Tidak dapat mengatur opsi di $regex dan $options”.

- Dengan cluster elastis, `$indexOfCP` sekarang mengembalikan “-1" ketika:
  - substring tidak ditemukan di`string expression`, atau
  - `start`adalah angka yang lebih besar dari`end`, atau
  - `start`adalah angka yang lebih besar dari panjang byte string.

  Di Amazon DocumentDB 4.0`$indexOfCP`, mengembalikan “0" ketika `start` posisi adalah angka yang `end` lebih besar dari atau panjang byte string.
- Dengan cluster elastis, operasi proyeksi di`_id fields`, misalnya:`{"_id.nestedField" : 1}`, mengembalikan dokumen yang hanya menyertakan bidang yang diproyeksikan. Sedangkan di Amazon DocumentDB 4.0, perintah proyeksi bidang bersarang tidak memfilter dokumen apa pun.

Memulai dengan cluster elastis Amazon DocumentDB

Bagian memulai ini memandu Anda tentang bagaimana Anda dapat membuat dan menanyakan cluster elastis pertama Anda. Ada banyak cara untuk terhubung dan memulai dengan cluster elastis. Panduan ini menggunakan AWS Cloud9, terminal berbasis web untuk menghubungkan dan menanyakan cluster elastis Anda menggunakan cangkang mongo langsung dari. AWS Management Console

Topik

- Mengatur
- Langkah 1: Buat cluster elastis
- Langkah 2: Buat AWS Cloud9 lingkungan
- Langkah 3: Instal cangkang mongo

- Langkah 4: Connect ke cluster elastis baru Anda
- Langkah 5: Pecahkan koleksi Anda; masukkan dan kueri data

Mengatur

Jika Anda lebih suka terhubung ke Amazon DocumentDB dari komputer lokal Anda dengan membuat koneksi SSH ke instans Amazon EC2, silakan lihat Menghubungkan dengan Amazon EC2.

Prasyarat

Sebelum membuat klaster Amazon DocumentDB pertama, Anda harus melakukan hal berikut:

Buatlah akun Amazon Web Services (AWS)

Sebelum Anda dapat mulai menggunakan Amazon DocumentDB, Anda harus memiliki akun Amazon Web Services (AWS). AWS Akun ini gratis. Anda hanya membayar untuk layanan dan sumber daya yang Anda gunakan.

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Siapkan izin yang diperlukan AWS Identity and Access Management (IAM).

Akses untuk mengelola sumber daya Amazon DocumentDB seperti cluster, instance, dan grup parameter cluster memerlukan kredensil yang dapat digunakan untuk AWS mengautentikasi permintaan Anda. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon DocumentDB.

1. Di bilah pencarian AWS Management Console, ketik IAM dan pilih IAM di menu tarik-turun.
2. Setelah Anda berada di konsol IAM, pilih Pengguna dari panel navigasi.
3. Pilih nama pengguna Anda.
4. Klik tombol Tambahkan izin.
5. Pilih Lampirkan kebijakan yang ada secara langsung.
6. Ketik `AmazonDocDBFullAccess` di bilah pencarian dan pilih itu setelah muncul di hasil pencarian.
7. Klik tombol biru di bagian bawah yang bertuliskan Berikutnya: Tinjau.
8. Klik tombol biru di bagian bawah yang bertuliskan Tambahkan izin.

Buat Amazon Virtual Private Cloud (Amazon VPC)

Langkah ini diperlukan hanya jika Anda belum memiliki Amazon VPC default. Jika tidak, selesaikan langkah 1 dari Memulai Amazon VPC di Panduan Pengguna Amazon VPC. Ini akan memakan waktu kurang dari lima menit.

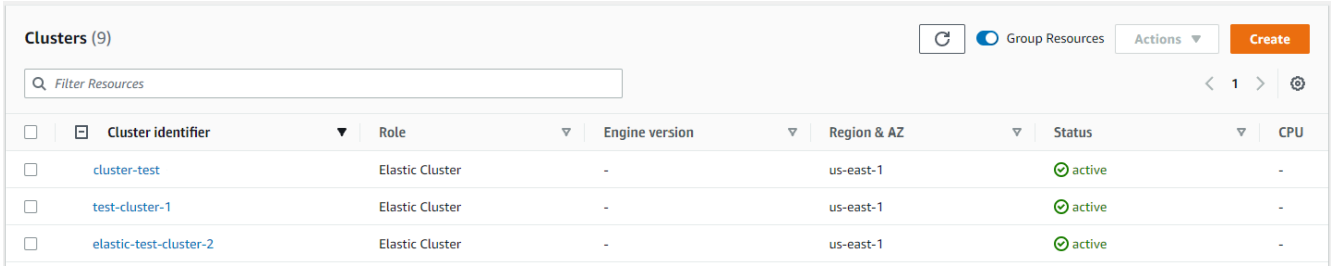
Langkah 1: Buat cluster elastis

Pada bagian ini kami menjelaskan cara membuat cluster elastis baru, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

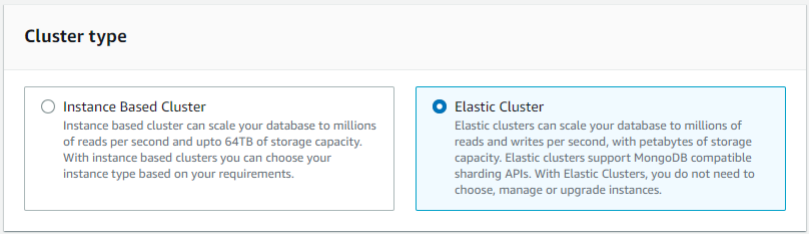
Using the AWS Management Console

Untuk membuat konfigurasi cluster elastis menggunakan AWS Management Console:

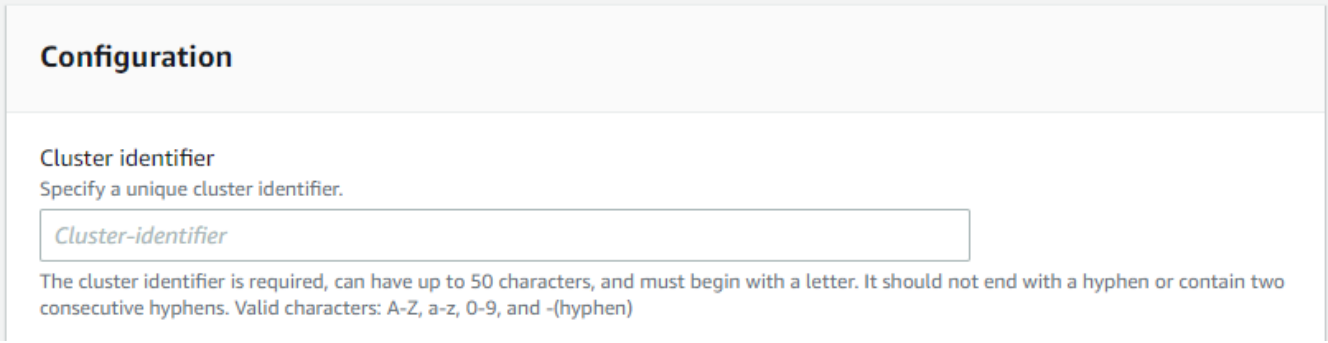
1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di Amazon DocumentDB Management Console, di bawah Clusters, pilih Create.



3. Pada halaman cluster Create Amazon DocumentDB, di bagian tipe Cluster, pilih Elastic Cluster.



4. Pada halaman cluster Create Amazon DocumentDB, di bagian Konfigurasi, masukkan pengidentifikasi cluster unik (mengikuti persyaratan penamaan di bawah bidang).



5. Untuk bidang konfigurasi shard:

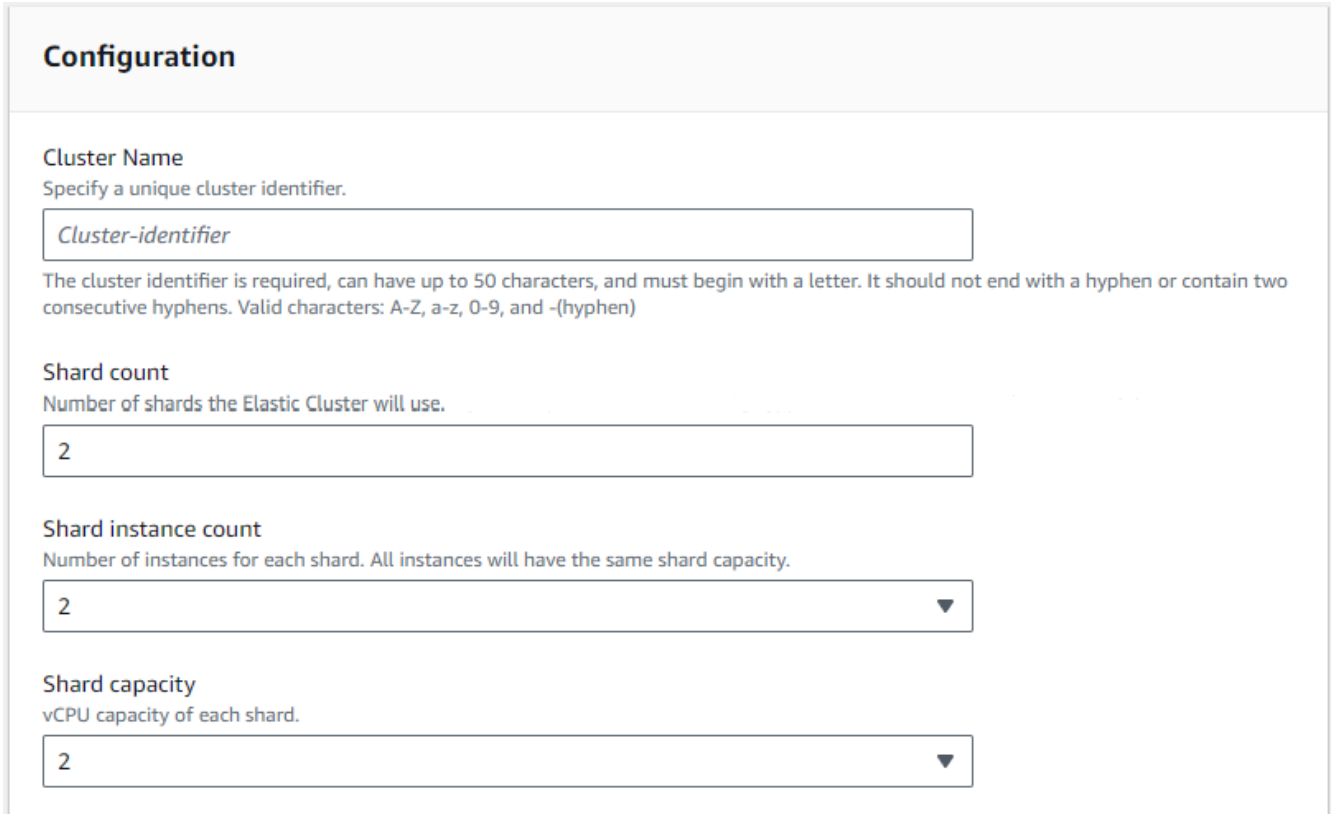
   a. Di bidang Shard count, masukkan jumlah pecahan yang Anda inginkan di cluster Anda. Jumlah maksimum pecahan per cluster adalah 32.

   > **Note**
   >
   > Dua node akan digunakan untuk setiap pecahan. Kedua node akan memiliki kapasitas shard yang sama.

   b. Di bidang hitungan instance Shard, pilih jumlah instance replika yang ingin Anda kaitkan dengan setiap pecahan. Jumlah maksimum instance shard adalah 16, dengan penambahan 1. Semua contoh replika memiliki kapasitas pecahan yang sama seperti yang didefinisikan dalam bidang berikut.

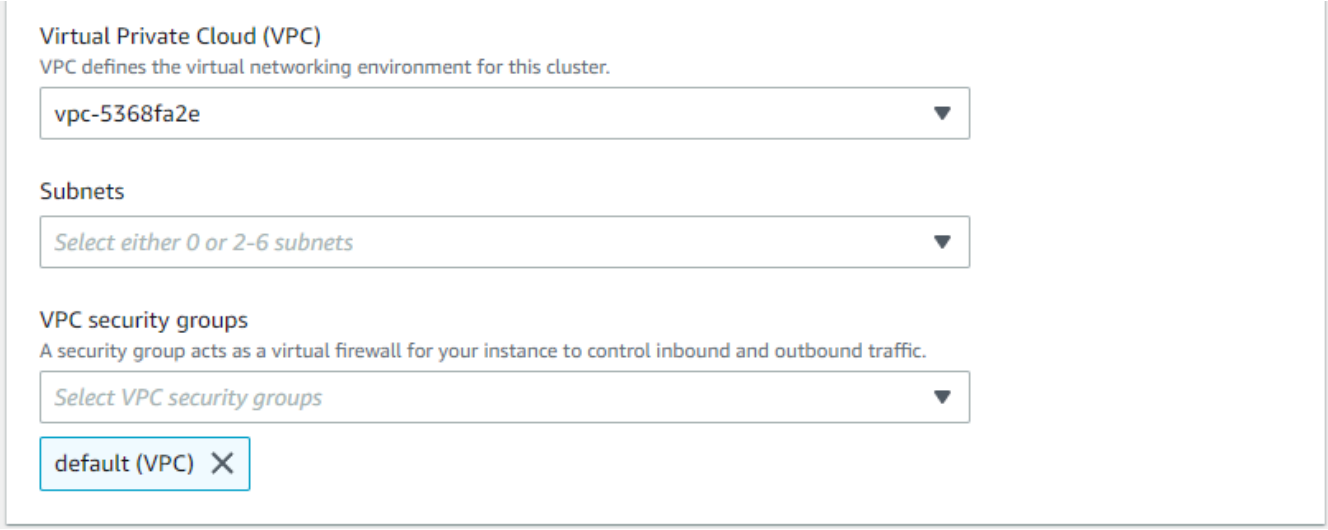
   > **Note**
   >
   > Jumlah contoh replika berlaku untuk semua pecahan di cluster elastis. Nilai hitungan instans pecahan 1 berarti ada satu instance penulis, dan setiap contoh tambahan adalah replika yang dapat digunakan untuk membaca dan untuk meningkatkan ketersediaan.

c. Di bidang kapasitas Shard, pilih jumlah CPU virtual (vCPU) yang ingin dikaitkan dengan setiap instance shard. Jumlah maksimum vCPU per instance shard adalah 64. Nilai yang diizinkan adalah 2, 4, 8, 16, 32, 64.



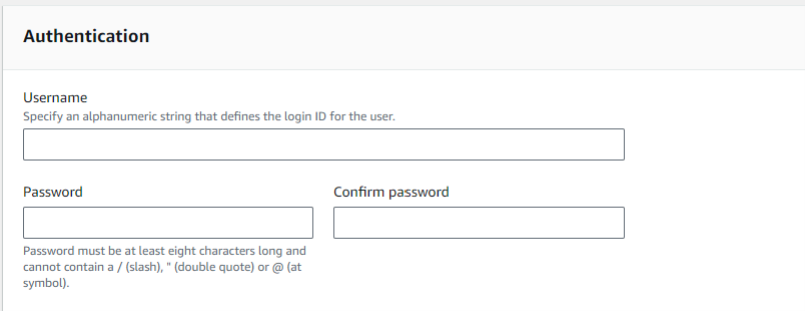
6. Di bidang Virtual Private Cloud (VPC), pilih VPC dari daftar drop-down.

   Untuk grup keamanan Subnet dan VPC, Anda dapat menggunakan default atau memilih tiga subnet pilihan Anda dan hingga tiga grup keamanan VPC (minimal satu).



7. Di bagian Otentikasi, masukkan string yang mengidentifikasi nama login pengguna utama di bidang Nama Pengguna.

   Di bidang Kata Sandi, masukkan kata sandi unik yang sesuai dengan instruksi.

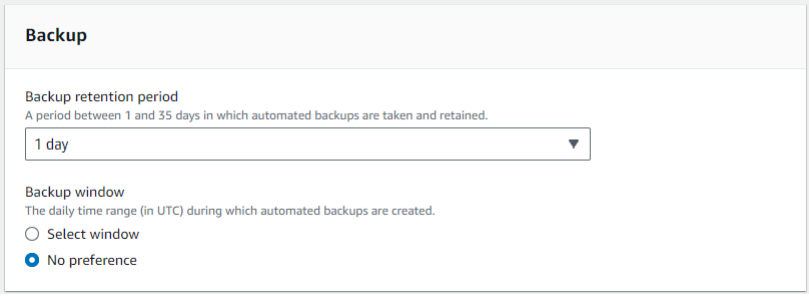


8. Di bagian Enkripsi, pertahankan pengaturan default.

   Secara opsional, Anda dapat memasukkan AWS KMS key ARN yang Anda buat. Untuk informasi selengkapnya, lihat Enkripsi data saat istirahat untuk klaster elastis Amazon DocumentDB.

   > **Important**
   >
   > Enkripsi harus diaktifkan untuk cluster elastis.

9. Di bagian Backup, edit bidang sesuai dengan kebutuhan cadangan Anda.



a. Periode retensi cadangan—Dalam daftar, pilih jumlah hari untuk menyimpan backup otomatis klaster ini sebelum menghapusnya.

b. Jendela backup—Atur waktu harian dan durasi yang akan digunakan oleh Amazon DocumentDB untuk membuat backup dari klaster ini.

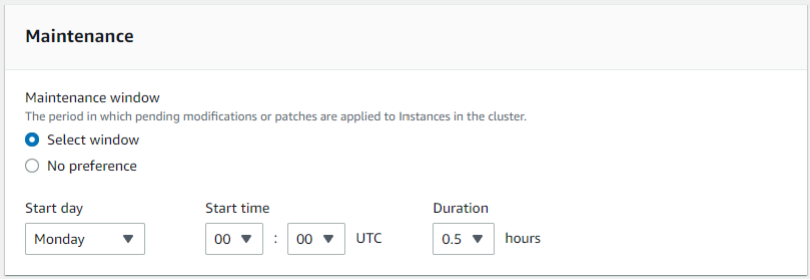
   i. Pilih Pilih jendela jika Anda ingin mengonfigurasi waktu dan durasi saat cadangan dibuat.

      Waktu mulai—Pada daftar pertama, pilih jam waktu mulai (UTC) untuk memulai pencadangan otomatis. Dalam daftar kedua, pilih menit dari jam pilihan Anda untuk memulai backup otomatis.

      Durasi—Dalam daftar, pilih jumlah jam yang akan dialokasikan untuk membuat cadangan otomatis.

   ii. Pilih Tidak ada preferensi jika Anda ingin Amazon DocumentDB memilih waktu dan durasi saat backup dibuat.

10. Di bagian Maintenance, pilih hari, waktu, dan durasi saat modifikasi atau patch diterapkan ke cluster Anda.



11. Pilih Buat klaster.

Cluster elastis sekarang sedang disediakan. Ini bisa memakan waktu hingga beberapa menit untuk menyelesaikannya. Anda dapat terhubung ke cluster Anda ketika status cluster elastis ditampilkan seperti **`active`** dalam daftar Clusters.

Using the AWS CLI

Untuk membuat cluster elastis menggunakan AWS CLI, gunakan `create-cluster` operasi dengan parameter berikut:

- --cluster-name—Wajib. Nama saat ini dari cluster skala elastis seperti yang dimasukkan selama pembuatan atau terakhir dimodifikasi.
- --shard-capacity—Wajib. Jumlah vCPU yang ditetapkan untuk setiap pecahan. Maksimal adalah 64. Nilai yang diizinkan adalah 2, 4, 8, 16, 32, 64.
- --shard-count—Wajib. Jumlah pecahan yang ditugaskan ke cluster. Maksimal 32.
- --shard-instance-count—Opsional. Jumlah contoh replika yang diterapkan ke semua pecahan di cluster ini. Maksimal adalah 16.
- --admin-user-name—Wajib. Nama pengguna yang terkait dengan pengguna admin.
- --admin-user-password—Wajib. Kata sandi yang terkait dengan pengguna admin.
- --auth-type—Wajib. Jenis otentikasi yang digunakan untuk menentukan di mana untuk mengambil password yang digunakan untuk mengakses cluster elastis. Jenis yang valid adalah `PLAIN_TEXT` atau`SECRET_ARN`.
- --vpc-security-group-ids—Opsional. Konfigurasikan daftar grup keamanan VPC EC2 untuk dikaitkan dengan cluster ini.
- --preferred-maintenance-window—Opsional. Konfigurasikan rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, di Universal Coordinated Time (UTC).

  Formatnya adalah:`ddd:hh24:mi-ddd:hh24:mi`. Hari yang berlaku (ddd): Senin, Sel, Rabu, Kam, Jum, Sabtu, Minggu

  Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk setiap Wilayah Amazon Web Services, yang terjadi pada hari acak dalam seminggu.

  Jendela minimal 30 menit.
- --kms-key-id—Opsional. Konfigurasikan pengidentifikasi kunci KMS untuk cluster terenkripsi.

  Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN) untuk kunci enkripsi. AWS KMS Jika Anda membuat klaster menggunakan akun Amazon Web Services yang sama yang memiliki kunci enkripsi KMS yang digunakan untuk mengenkripsi cluster baru, Anda dapat menggunakan alias kunci KMS alih-alih ARN untuk kunci enkripsi KMS.

Jika kunci enkripsi tidak ditentukan KmsKeyId dan jika `StorageEncrypted` parameternya benar, Amazon DocumentDB menggunakan kunci enkripsi default Anda.

- --preferred-backup-window—Opsional. Rentang waktu harian yang disukai selama pencadangan otomatis dibuat. Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing. Wilayah AWS
- --backup-retention-period—Opsional. Jumlah hari penyimpanan cadangan otomatis. Nilai default adalah 1.
- --storage-encrypted—Opsional. Mengkonfigurasikan apakah cluster dienkripsi atau tidak dienkripsi.

  `--no-storage-encrypted`menentukan bahwa cluster tidak dienkripsi.
- --subnet-ids—Opsional. Konfigurasikan ID subnet jaringan.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

> **Note**
>
> Contoh berikut termasuk pembuatan kunci KMS tertentu. Untuk menggunakan kunci KMS default, jangan sertakan `--kms-key-id` parameter.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

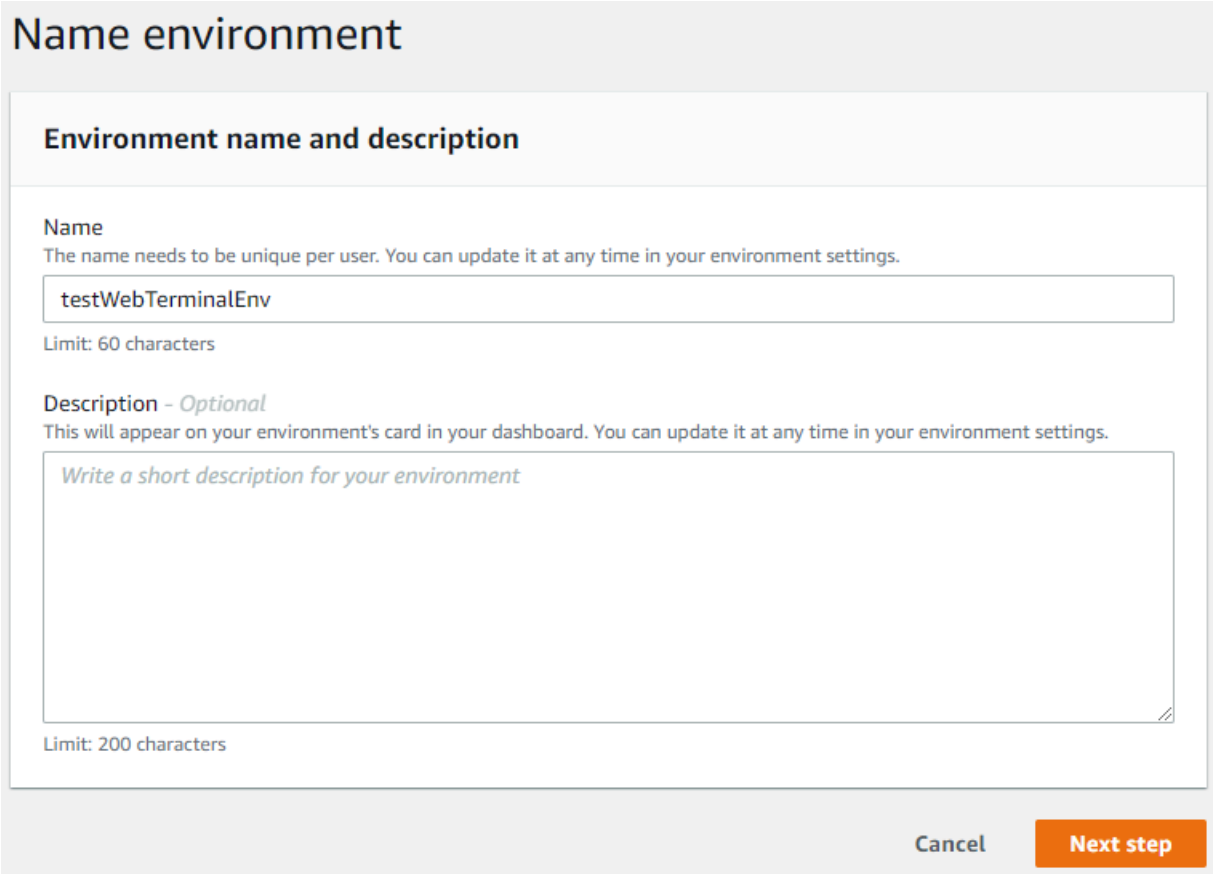
Langkah 2: Buat AWS Cloud9 lingkungan

AWS Cloud9 menyediakan terminal berbasis web yang dapat Anda gunakan untuk menghubungkan dan menanyakan cluster elastis Amazon DocumentDB Anda menggunakan cangkang mongo.

> **Note**
>
> Catatan: AWS Cloud9 Lingkungan Anda harus berada dalam grup keamanan yang sama dengan instans Anda. Anda dapat mengubah grup keamanan di konsol Amazon EC2.

1. Gunakan AWS akun Anda dan akses AWS Management Console.
2. Arahkan ke AWS Cloud9 Konsol. Anda dapat mengetik “Cloud9" di bidang Pencarian untuk menemukannya.
3. Di halaman beranda AWS Cloud9lingkungan, pilih Buat lingkungan.
4. Pada halaman Lingkungan nama, di bidang Nama, masukkan nama yang Anda pilih.

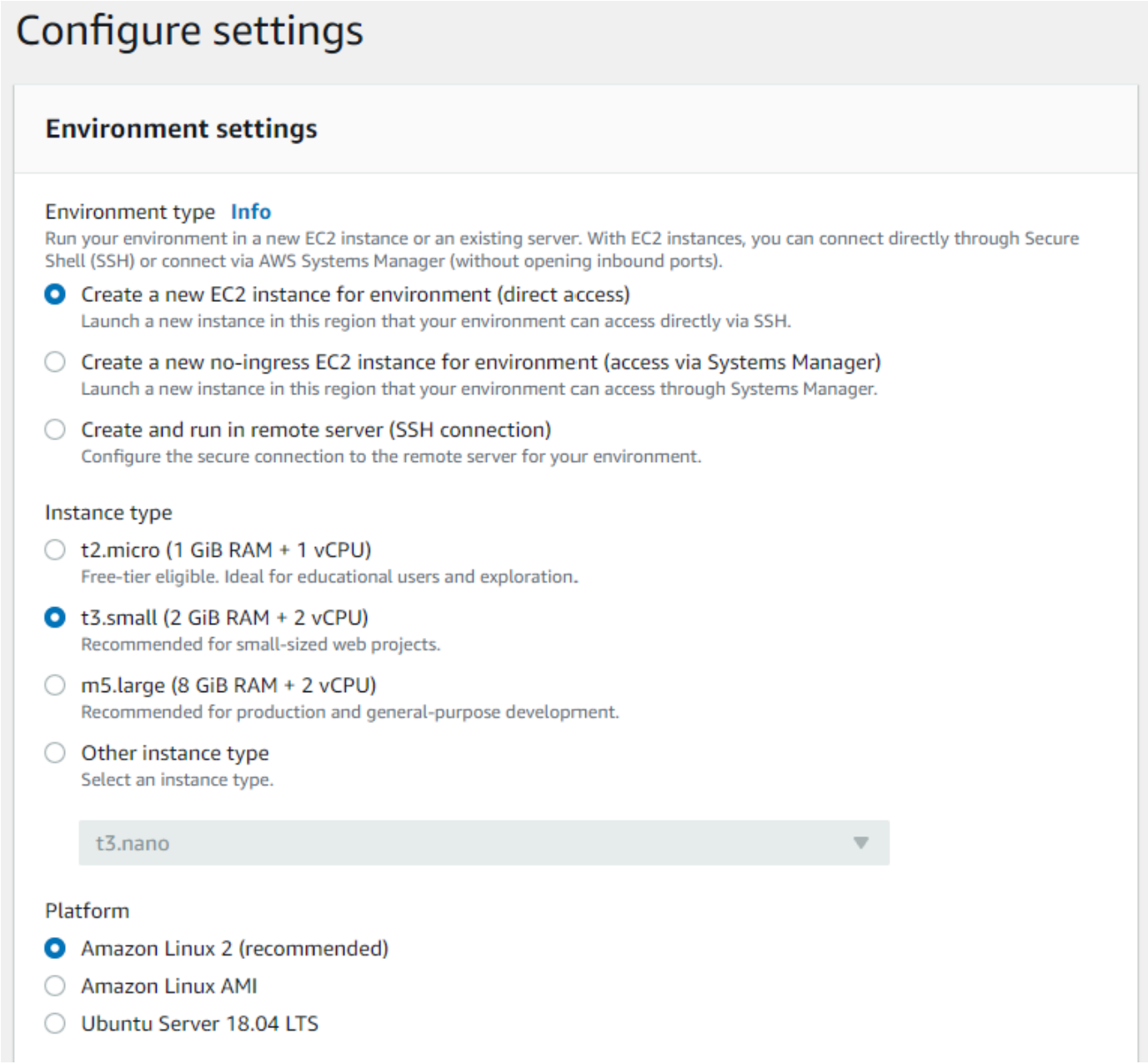
   Pilih Langkah selanjutnya.



5. Di Pengaturan lingkungan di bawah bagian Jenis lingkungan, pilih, Buat instans EC2 baru untuk lingkungan (akses langsung).

   Di bawah bagian Jenis instans, pilih jenis instans yang sesuai untuk jaringan Anda.

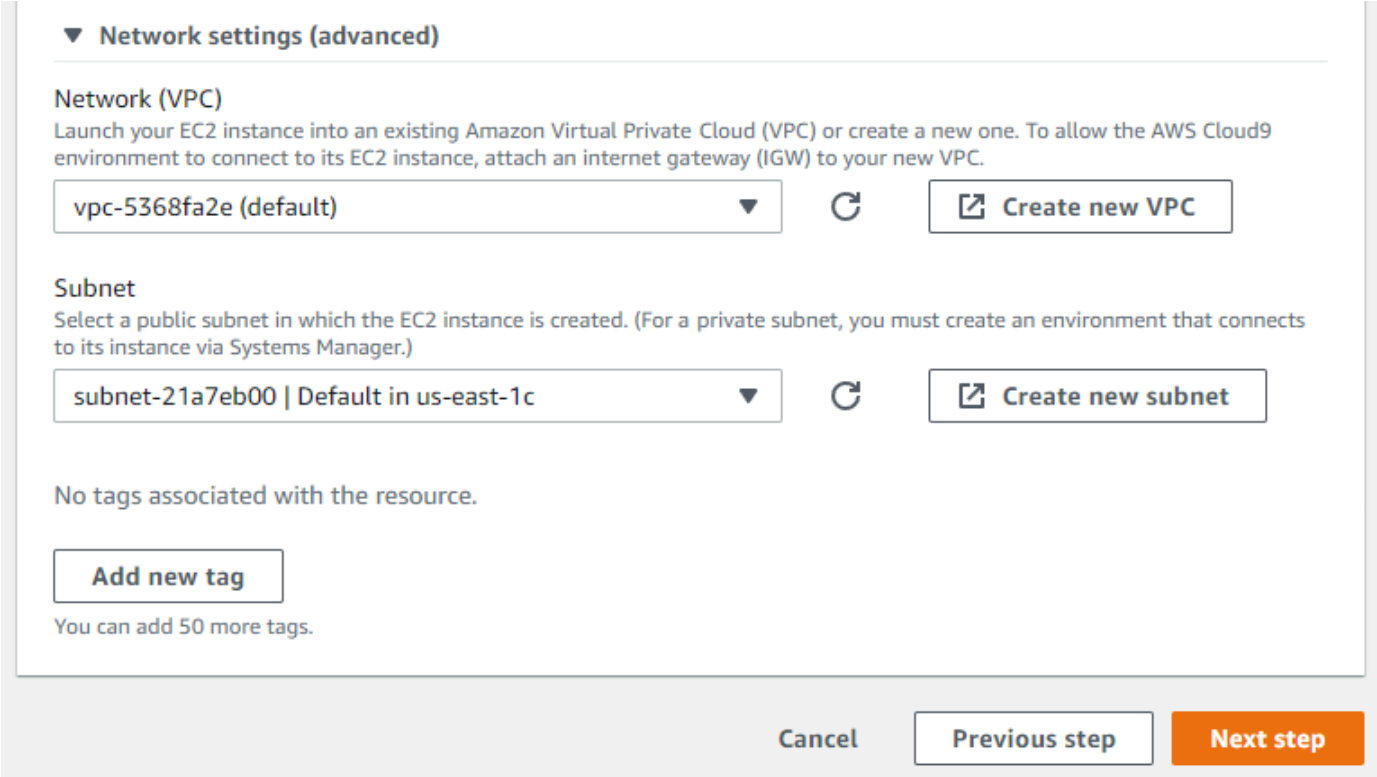
   Di bawah bagian Platform, pilih Amazon Linux 2 (disarankan).



6. Perluas Pengaturan jaringan (lanjutan).

   Pilih VPC dan salah satu subnet yang Anda gunakan saat membuat cluster elastis Anda.

   Pilih Langkah selanjutnya.



7. Tinjau AWS Cloud9 konfigurasi Anda.

   Jika konfigurasi Anda benar, pilih Buat lingkungan.

Langkah 3: Instal cangkang mongo

Setelah AWS Cloud9 lingkungan Anda siap, Anda siap untuk terhubung ke cluster Anda. Selanjutnya, instal shell mongo di AWS Cloud9 lingkungan Anda yang Anda buat di Langkah 3. Mongo shell adalah utilitas baris perintah yang Anda gunakan untuk menghubungkan dan menanyakan cluster elastis Anda.

Jika AWS Cloud9 lingkungan Anda masih terbuka dari Langkah 3, kembali ke lingkungan itu dan lompat ke instruksi 3. Jika Anda menavigasi jauh dari AWS Cloud9 lingkungan Anda, di AWS Cloud9 konsol, di bawah lingkungan Anda, temukan lingkungan berlabel dengan nama yang Anda tetapkan di langkah sebelumnya. Pilih Buka IDE.

1. Pada prompt perintah, buat file repositori dengan perintah berikut:

Example

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

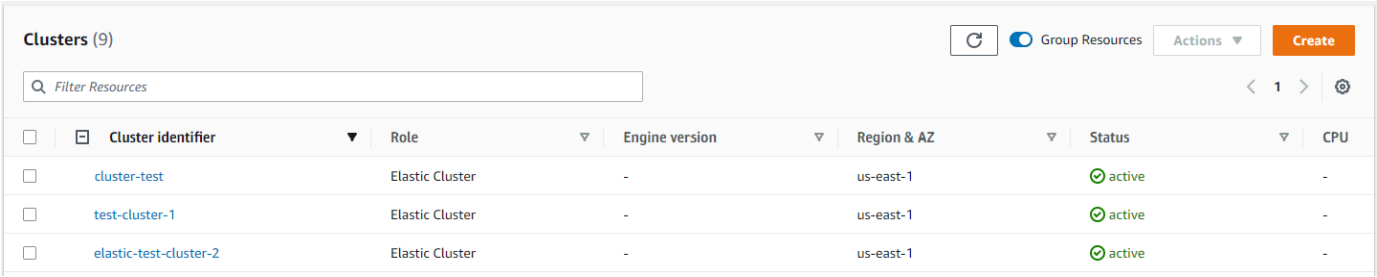
2. Setelah selesai, instal mongo shell dengan perintah berikut:

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

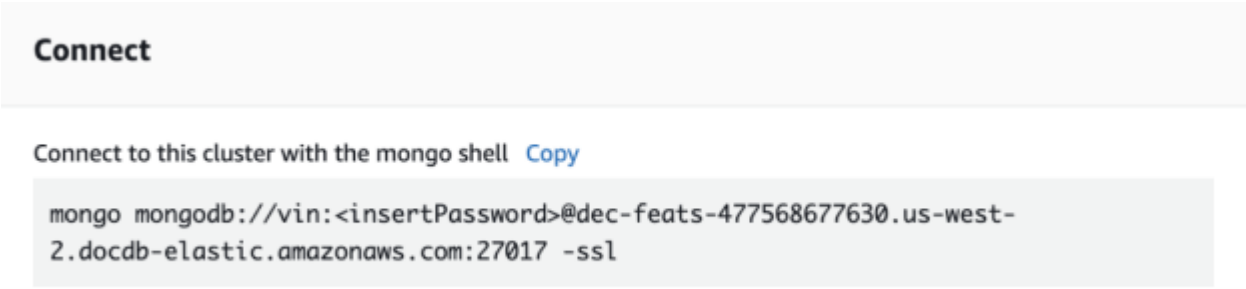
Langkah 4: Connect ke cluster elastis baru Anda

Connect ke cluster Anda menggunakan shell mongo yang Anda instal di Langkah 4.

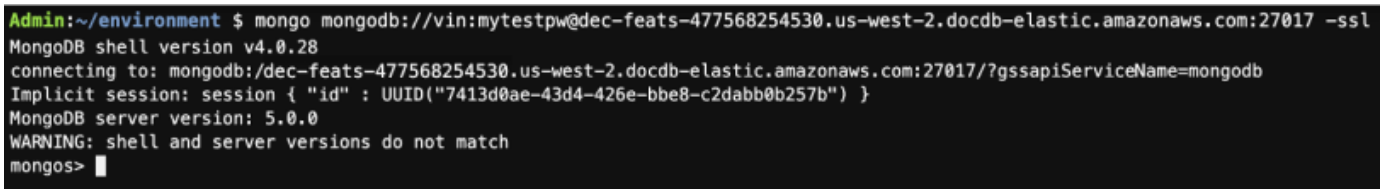
1. Di Amazon DocumentDB Management Console, di bawah Clusters, temukan klaster Anda. Urutkan berdasarkan peran untuk menampilkan semua cluster dengan peran Elastic Cluster.



2. Pilih cluster yang Anda buat dengan memilih pengidentifikasi cluster. Dari Konektivitas dan Keamanan, salin titik akhir Anda dan tempelkan ke AWS Cloud9 lingkungan Anda.



3. Setelah terhubung, Anda akan melihat output berikut:



Langkah 5: Pecahkan koleksi Anda; masukkan dan kueri data

Cluster elastis menambahkan dukungan untuk sharding di Amazon DocumentDB. Sekarang Anda terhubung ke cluster Anda, Anda dapat menghancurkan cluster, menyisipkan data dan menjalankan beberapa kueri.

1. Untuk menghancurkan koleksi, masukkan yang berikut ini:

   ```
   sh.shardCollection("db.Employee1" , { "Employeeid" : "hashed" })
   ```

2. Untuk memasukkan satu dokumen, masukkan hal berikut ini:

   ```
   db.Employee1.insert({"Employeeid":1, "Name":"Joe", "LastName": "Bruin",
   "level": 1 })
   ```

   Output berikut ditampilkan:

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

3. Untuk membaca dokumen yang Anda tulis, masukkan `findOne()` perintah (mengembalikan satu dokumen):

   ```
   db.Employee1.findOne()
   ```

   Output berikut ditampilkan:

   Example

   ```
   {
   "_id" : ObjectId("61f344e0594fe1a1685a8151"),
   "EmployeeID" : 1,
   "Name" : "Joe",
   "LastName" : "Bruin",
   "level" : 1
   }
   ```

4. Untuk melakukan beberapa pertanyaan lagi, pertimbangkan kasus penggunaan profil game. Pertama, masukkan beberapa entri ke dalam koleksi berjudul “Karyawan”. Masukkan yang berikut ini:

   Example

   ```
   db.Employee1.insertMany([
       { "Employeeid" : 1, "name" : "Matt", "lastname": "Winkle", "level": 12},
       { "Employeeid" : 2, "name" : "Frank", "lastname": "Chen", "level": 2},
       { "Employeeid" : 3, "name" : "Karen", "lastname": "William", "level": 7},
       { "Employeeid" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}
   ])
   ```

   Output berikut ditampilkan:

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

5. Untuk mengembalikan semua dokumen dalam koleksi profil, masukkan perintah `find ()`:

   ```
   db.Employee1.find()
   ```

   Data yang Anda masukkan pada langkah 4 ditampilkan.

6. Untuk menanyakan satu dokumen, sertakan filter (misalnya: “Katie”). Masukkan yang berikut ini:

   ```
   db.Employee1.find({name: "Katie"})
   ```

   Output berikut ditampilkan:

   ```
   { "_id" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}
   ```

7. Untuk menemukan profil dan memodifikasinya, masukkan `findAndModify` perintah. Dalam contoh ini, karyawan “Matt” diberi level “14" yang lebih tinggi:

   Example

   ```
   db.Employee1.findAndModify({
       query: { "Employeeid" : 1, "name" : "Matt"},
       update: { "Employeeid" : 1, "name" : "Matt", "lastname" : "Winkle", "level" :
    14 }
   })
   ```

   Output berikut ditampilkan (perhatikan bahwa levelnya belum berubah):

Example

```
{
    "_id" : 1,
    "name" : "Matt",
    "lastname" : "Winkle",
    "level" : 12,
}
```

8. Untuk memverifikasi peningkatan level, masukkan kueri berikut:

   ```
   db.Employee1.find({name: "Matt"})
   ```

   Output berikut ditampilkan:

   ```
   { "_id" : 1, "name" : "Matt", "lastname" : "winkle", "level" : 14 }
   ```

Praktik terbaik

Pelajari praktik terbaik untuk bekerja dengan klaster Amazon DocumentDB. Semua praktik terbaik untuk klaster Amazon DocumentDB berbasis instans juga berlaku untuk klaster elastis. Bagian ini terus diperbarui saat praktik terbaik baru diidentifikasi.

Topik

- Memilih Shard Keys
- Manajemen Koneksi
- Koleksi Unsharded
- Menskalakan klaster
- Memantau klaster

Memilih Shard Keys

Daftar berikut menjelaskan pedoman untuk membuat kunci pecahan.

- Gunakan kunci hash yang didistribusikan secara merata untuk mendistribusikan data Anda di semua pecahan di klaster Anda (hindari tombol pintas).

- Gunakan kunci pecahan Anda di semua permintaan baca/perbarui/hapus untuk menghindari permintaan pencar mengumpulkan.
- Hindari tombol beling bersarang saat melakukan operasi baca/update/delete.
- Saat melakukan operasi batch, atur`ordered` ke false sehingga semua pecahan dapat berjalan secara parallel dan meningkatkan latensi.

Manajemen Koneksi

Daftar berikut menjelaskan pedoman untuk mengelola koneksi Anda ke database Anda.

- Pantau jumlah koneksi Anda dan seberapa sering koneksi baru dibuka ditutup.
- Bagikan koneksi Anda di semua subnet dalam konfigurasi aplikasi Anda. Jika klaster Anda dikonfigurasi dalam beberapa subnet tetapi Anda hanya menggunakan subset subnet, Anda mungkin mengalami hambatan pada koneksi maksimum Anda.

Koleksi Unsharded

Berikut ini menjelaskan pedoman untuk koleksi unsharded.

- Ketika bekerja dengan koleksi unsharded, untuk mendistribusikan beban, cobalah menjaga koleksi unsharded sangat dimanfaatkan pada database yang berbeda. Klaster elastis Amazon DocumentDB menempatkan database di seluruh pecahan yang berbeda dan menempatkan bersama koleksi tanpa sharded untuk database yang sama pada pecahan yang sama.

Menskalakan klaster

Daftar berikut menjelaskan pedoman untuk menskalakan klaster elastis Anda.

- Operasi penskalaan dapat menyebabkan periode singkat database intermiten dan kesalahan jaringan. Bila memungkinkan, hindari penskalaan selama jam sibuk. Cobalah untuk skala skala selama jendela pemeliharaan.
- Scaling kapasitas shard naik dan turun (mengubah jumlah vCPU per shard) untuk meningkatkan komputasi lebih disukai daripada menambah atau mengurangi shard-count karena lebih cepat dan memiliki durasi yang lebih pendek dari database intermiten dan kesalahan jaringan.

- Ketika mengantisipasi pertumbuhan, mendukung peningkatan jumlah pecahan bukannya skala kapasitas pecahan. Ini memungkinkan Anda untuk menskalakan klaster Anda dengan meningkatkan kapasitas pecahan untuk skenario di mana Anda perlu menskalakan dengan cepat.
- Pantau kebijakan percobaan ulang sisi klien Anda dan coba lagi dengan backoff dan jitter eksponensial untuk menghindari overloading database Anda saat mendapatkan kesalahan saat penskalaan.

Memantau klaster

Daftar berikut menjelaskan pedoman untuk memantau klaster elastis Anda.

- Lacakpeak-to-average rasio metrik per-shard Anda untuk menentukan apakah Anda mengendarai lalu lintas yang tidak genap (memiliki hot-key/hot-spot). Metrik utama untuk melacakpeak-to-average rasio adalah:
  - `PrimaryInstanceCPUUtilization`
    - Ini dapat dipantau pada tingkat per-shard.
    - Pada tingkat cluster Anda dapat memantau rata-rata untuk p99 condong.
  - `PrimaryInstanceFreeableMemory`
    - Ini dapat dipantau pada tingkat per-shard.
    - Pada tingkat cluster Anda dapat memantau rata-rata untuk p99 condong.
  - `DatabaseCursorsMax`
    - Ini harus dipantau pada tingkat per-pecahan untuk menentukan kemiringan.
  - `Documents-Inserted/Updated/Returned/Deleted`
    - Ini harus dipantau pada tingkat per-pecahan untuk menentukan kemiringan.

Mengelola cluster elastis

Untuk mengelola cluster elastis Amazon DocumentDB, Anda harus memiliki kebijakan IAM dengan izin bidang kontrol Amazon DocumentDB yang sesuai. Izin ini memungkinkan Anda untuk membuat, memodifikasi, dan menghapus cluster. Kebijakan Amazon FullAccess DocumentDB menyediakan semua izin yang diperlukan untuk mengelola klaster elastis Amazon DocumentDB.

Topik berikut menunjukkan cara melakukan berbagai tugas saat bekerja dengan cluster elastis Amazon DocumentDB.

Topik

- Memodifikasi konfigurasi cluster elastis
- Memantau cluster elastis
- Menghapus cluster elastis
- Mengelola snapshot cluster elastis
- Menghentikan dan memulai cluster elastis Amazon DocumentDB

Memodifikasi konfigurasi cluster elastis

Pada bagian ini kami menjelaskan cara memodifikasi cluster elastis, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

Penggunaan utama memodifikasi cluster adalah untuk menskalakan pecahan dengan meningkatkan atau mengurangi jumlah pecahan dan/atau kapasitas komputasi pecahan.

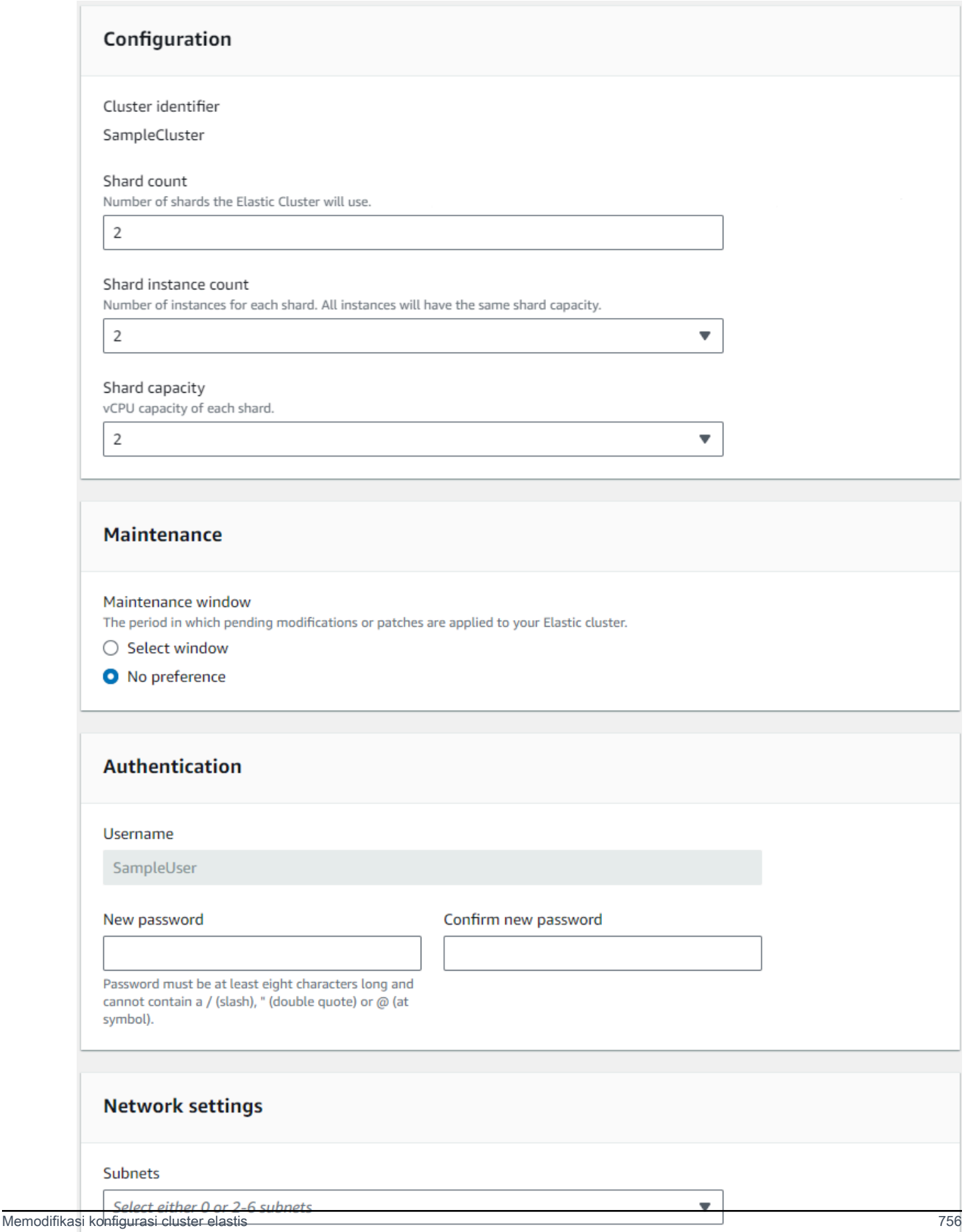
Using the AWS Management Console

Untuk memodifikasi konfigurasi cluster elastis menggunakan AWS Management Console:

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Pada panel navigasi, silakan pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pilih nama cluster yang ingin Anda ubah di kolom pengenal Cluster.
4. Pilih Ubah.
5. Edit bidang yang ingin diubah lalu pilih Modify cluster.


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**
>
> Atau, Anda dapat mengakses dialog Modify cluster dengan membuka halaman Clusters, mencentang kotak di sebelah cluster Anda, memilih Actions, lalu Modify.

Using the AWS CLI

Untuk memodifikasi konfigurasi cluster elastis menggunakan AWS CLI, gunakan `update-cluster` operasi dengan parameter berikut:

- **`--cluster-arn`**—Wajib. Pengidentifikasi ARN dari cluster yang ingin Anda modifikasi.
- **`--shard-capacity`**—Opsional. Jumlah vCPU yang ditetapkan untuk setiap pecahan. Maksimal 64. Nilai yang diizinkan adalah 2, 4, 8, 16, 32, 64.
- **`--shard-count`**—Opsional. Jumlah pecahan yang ditugaskan ke cluster. Maksimal 32.
- **`--shard-instance`**-Hitung—opsional. Jumlah contoh replika yang diterapkan ke semua pecahan di cluster ini. Maksimal adalah 16.
- **`--auth-type`**—Opsional. Jenis otentikasi yang digunakan untuk menentukan di mana untuk mengambil password yang digunakan untuk mengakses cluster elastis. Jenis yang valid adalah `PLAIN_TEXT` atau`SECRET_ARN`.
- **`--admin-user-password`**—Opsional. Kata sandi yang terkait dengan pengguna admin.
- **`--vpc-security-group-ids`**—Opsional. Konfigurasikan daftar grup keamanan Amazon EC2 dan Amazon Virtual Private Cloud (VPC) untuk diasosiasikan dengan klaster ini.
- **`--preferred-maintenance-window`**—Opsional. Konfigurasikan rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, di Universal Coordinated Time (UTC)

  Formatnya adalah:`ddd:hh24:mi-ddd:hh24:mi`. Hari yang berlaku (ddd): Senin, Sel, Rabu, Kam, Jum, Sabtu, Minggu

  Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk setiap Wilayah Amazon Web Services, yang terjadi pada hari acak dalam seminggu.

  Jendela minimal 30 menit.
- **`--subnet-ids`**—Opsional. Konfigurasikan ID subnet jaringan.

Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Untuk memantau status cluster elastis setelah modifikasi Anda, lihat Memantau cluster elastis.

Memantau cluster elastis

Pada bagian ini, kami menjelaskan cara memantau cluster elastis Anda, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

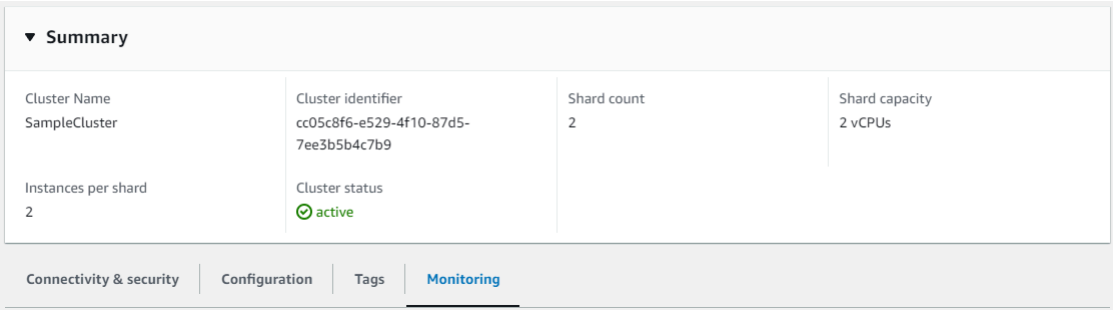
Using the AWS Management Console

Untuk memantau konfigurasi cluster elastis menggunakan AWS Management Console:

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Pada panel navigasi, silakan pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pilih nama cluster yang ingin Anda pantau di kolom pengenal Cluster.
4. Pilih tab Pemantauan.



Sejumlah grafik dari Amazon CloudWatch ditampilkan untuk kategori pemantauan berikut:

- Pemanfaatan Sumber Daya
- Throughput
- Latensi
- Operasi
- Sistem

Anda juga dapat mengakses Amazon CloudWatch melalui AWS Management Console untuk mengatur lingkungan pemantauan Anda sendiri untuk cluster elastis Anda.

Using the AWS CLI

Untuk memantau konfigurasi cluster elastis tertentu menggunakan AWS CLI, gunakan `get-cluster` operasi dengan parameter berikut:

- **`--cluster-arn`**—Wajib. Pengidentifikasi ARN dari cluster yang Anda inginkan informasinya.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic get-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Untuk Windows:

```
aws docdb-elastic get-cluster ^
    --cluster-arn arn:aws:docdb:-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Output dari operasi ini terlihat seperti berikut:

```
"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",
        ...
 }
```

Untuk informasi selengkapnya, lihat `DescribeClusterSnapshot` di Referensi API Manajemen Sumber Daya Amazon DocumentDB.

Untuk melihat rincian semua cluster elastis menggunakan AWS CLI, gunakan `list-clusters` operasi dengan parameter berikut:

- **`--next-token`**—Opsional. Jika jumlah item output (`--max-results`) lebih sedikit dari jumlah total item yang dikembalikan oleh panggilan API yang mendasarinya, outputnya menyertakan perintah `NextToken` yang dapat Anda teruskan ke perintah berikutnya untuk mengambil set item berikutnya.

- **`--max-results`**—Opsional. Jumlah total item yang akan dikembalikan dalam output perintah. Jika lebih banyak hasil yang ada daripada `max-results` nilai yang ditentukan, token pagination (`next-token`) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.
  - Default: 100
  - Minimum 20, maksimum 100

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic list-clusters \
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== \
    --max-results 2
```

Untuk Windows:

```
aws docdb-elastic list-clusters ^
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== ^
    --max-results 2
```

Output dari operasi ini terlihat seperti berikut:

```
{
   "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"
```

```
        }
    ]
}
```

Menghapus cluster elastis

Pada bagian ini kami menjelaskan cara menghapus cluster elastis, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

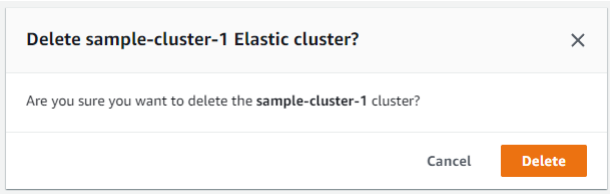
Using the AWS Management Console

Untuk menghapus konfigurasi cluster elastis menggunakan AWS Management Console:

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Pada panel navigasi, silakan pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Dalam tabel daftar cluster, pilih kotak centang di sebelah kiri nama cluster yang ingin Anda hapus lalu pilih Tindakan. Dari menu pilihan menurun, pilih Hapus.
4. Di cluster elastis Delete “cluster-name”? kotak dialog, pilih Hapus.



Dibutuhkan beberapa menit untuk menghapus klaster. Untuk memantau status klaster, lihat Memantau Status Cluster Amazon DocumentDB.

Using the AWS CLI

Untuk menghapus cluster elastis menggunakan AWS CLI, gunakan `delete-cluster` operasi dengan parameter berikut:

- **`--cluster-arn`**—Wajib. Pengidentifikasi ARN dari cluster yang ingin Anda hapus.

- **`--no-skip-final-backup`**—Opsional. Jika Anda menginginkan cadangan akhir, Anda harus menyertakan parameter ini dengan nama untuk cadangan akhir. Anda harus menyertakan `--final-backup-identifier` atau `--skip-final-backup`.
- **`--skip-final-backup`**—Opsional. Gunakan parameter ini hanya jika Anda tidak ingin mengambil cadangan akhir sebelum menghapus klaster Anda. Pengaturan default adalah untuk mengambil snapshot akhir.

Contoh AWS CLI kode berikut menghapus cluster dengan ARN arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster dengan cadangan akhir.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna dengan informasi`* Anda sendiri..

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Contoh AWS CLI kode berikut menghapus cluster dengan ARN arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster tanpa mengambil cadangan akhir.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic delete-cluster \
```

```
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster \
    --skip-final-backup \
```

Untuk Windows:

```
aws docdb-elastic delete-cluster ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster ^
    --skip-final-backup ^
```

Output dari `delete-cluster` operasi adalah tampilan cluster yang Anda hapus.

Dibutuhkan beberapa menit untuk menghapus klaster. Untuk memantau status klaster, lihat Memantau Status Cluster Amazon DocumentDB.

Mengelola snapshot cluster elastis

Snapshot manual dapat diambil setelah cluster elastis dibuat. Pencadangan otomatis dibuat saat snapshot cluster elastis dibuat.

> **Note**
>
> Cluster elastis Anda harus dalam `Available` keadaan agar snapshot manual diambil.

Bagian ini menjelaskan bagaimana Anda dapat membuat, melihat, memulihkan dari, dan menghapus snapshot cluster elastis.

Topik berikut menunjukkan cara melakukan berbagai tugas saat bekerja dengan snapshot cluster elastis Amazon DocumentDB.

Topik

- Membuat snapshot cluster elastis manual
- Melihat snapshot cluster elastis
- Memulihkan cluster elastis dari snapshot
- Menyalin snapshot cluster elastis

- Menghapus snapshot cluster elastis
- Mengelola cadangan otomatis snapshot cluster elastis

Membuat snapshot cluster elastis manual

Pada bagian ini kami menjelaskan cara membuat snapshot cluster elastis manual, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

Using the AWS Management Console

Untuk membuat snapshot cluster elastis manual menggunakan: AWS Management Console

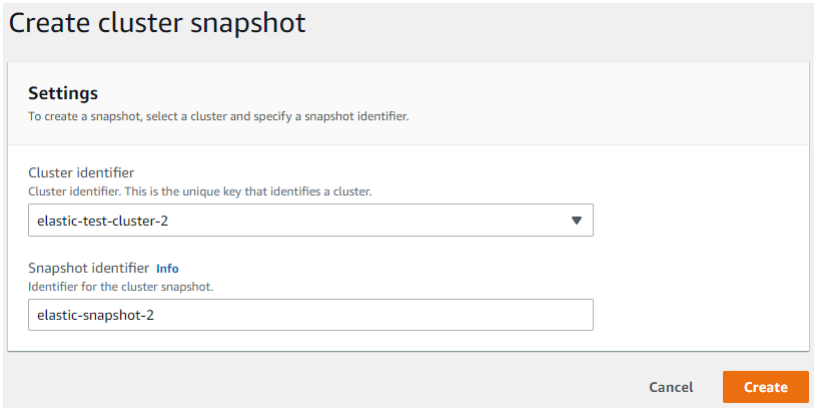
1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Snapshot.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pada halaman Snapshot, pilih Buat.
4. Pada halaman Create cluster snapshot, di kolom Cluster identifier, pilih cluster elastis Anda dari daftar drop-down.

   Di bidang pengenal Snapshot, masukkan pengenal unik untuk cluster elastis Anda.

   Pilih Buat.



> **Note**
>
> Atau, Anda dapat mengakses dialog Buat snapshot cluster dengan membuka halaman Clusters, mencentang kotak di sebelah cluster Anda, lalu memilih Tindakan, lalu Ambil snapshot.

Snapshot cluster elastis Anda sekarang sedang disediakan. Ini bisa memakan waktu hingga beberapa menit untuk menyelesaikannya. Anda dapat melihat dan memulihkan dari snapshot Anda ketika status ditampilkan seperti `Available` dalam daftar Snapshots.

Using the AWS CLI

Untuk membuat snapshot cluster elastis manual menggunakan AWS CLI, gunakan `create-cluster-snapshot` operasi dengan parameter berikut:

- **`--snapshot-name`**—Wajib. Nama snapshot cluster yang ingin Anda buat.
- **`--cluster-arn`**—Wajib. Pengidentifikasi ARN dari cluster yang ingin Anda buat snapshot.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Melihat snapshot cluster elastis

Pada bagian ini kami menjelaskan cara melihat informasi snapshot cluster elastis, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

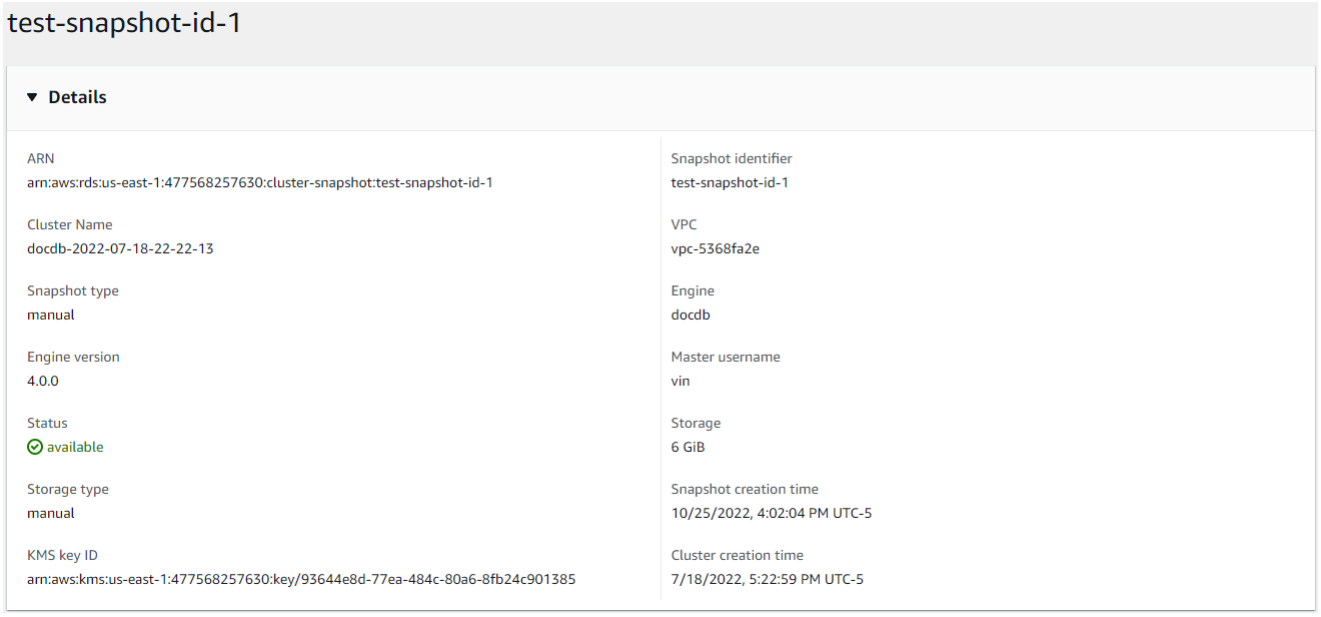
Using the AWS Management Console

Untuk melihat informasi tentang snapshot cluster elastis tertentu menggunakan: AWS Management Console

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Snapshot.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pada halaman Snapshots, pilih snapshot Anda dari daftar dengan mengklik nama di kolom pengenal Snapshot.
4. Lihat informasi snapshot Anda di Detail.



Using the AWS CLI

Untuk melihat informasi tentang snapshot cluster elastis tertentu menggunakan AWS CLI, gunakan `get-cluster-snapshot` operasi dengan parameter berikut:

- **`--snapshot-arn`**—Wajib. Pengidentifikasi ARN dari snapshot yang Anda inginkan informasinya.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Untuk Windows:

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Untuk melihat informasi tentang snapshot cluster elastis tertentu menggunakan AWS CLI, gunakan `get-cluster-snapshot` operasi dengan parameter berikut:

- **`--snapshot-arn`**—Wajib. Pengidentifikasi ARN dari snapshot yang Anda inginkan informasinya.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Untuk Windows:

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Untuk melihat informasi tentang semua snapshot cluster elastis menggunakan AWS CLI, gunakan `list-cluster-snapshots` operasi dengan parameter berikut:

- **`--snapshot-type`**—Opsional. Jenis snapshot klaster untuk dikembalikan. Anda dapat menentukan salah satu nilai berikut:
  - `automated`- Kembalikan semua snapshot cluster yang Amazon DocumentDB telah secara otomatis dibuat untuk akun Anda. AWS
  - `manual`- Kembalikan semua snapshot cluster yang telah Anda buat secara manual untuk AWS akun Anda.
  - `shared`- Kembalikan semua snapshot cluster manual yang telah dibagikan ke AWS akun Anda.
  - `public` - Mengembalikan semua snapshot klaster yang telah ditandai sebagai publik.
- **`--next-token`**—Opsional. Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya mencakup catatan di luar token ini, hingga nilai yang ditentukan oleh`max-results`.
- **`--max-results`**—Opsional. Jumlah maksimum hasil untuk dimasukkan dalam respons. Jika lebih banyak hasil yang ada daripada `max-results` nilai yang ditentukan, token pagination (`next-token`) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.
  - Default: 100
  - Minimum 20, maksimum 100

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic list-cluster-snapshots \
    --snapshot-type  value \
    --next-token  value \
    --max-results 50
```

Untuk Windows:

```
aws docdb-elastic list-cluster-snapshots ^
    --snapshot-type  value ^
```

```
    --next-token  value ^
    --max-results 50
```

Memulihkan cluster elastis dari snapshot

Pada bagian ini kami menjelaskan cara mengembalikan cluster elastis dari snapshot, menggunakan AWS Management Console atau AWS CLI dengan instruksi berikut.

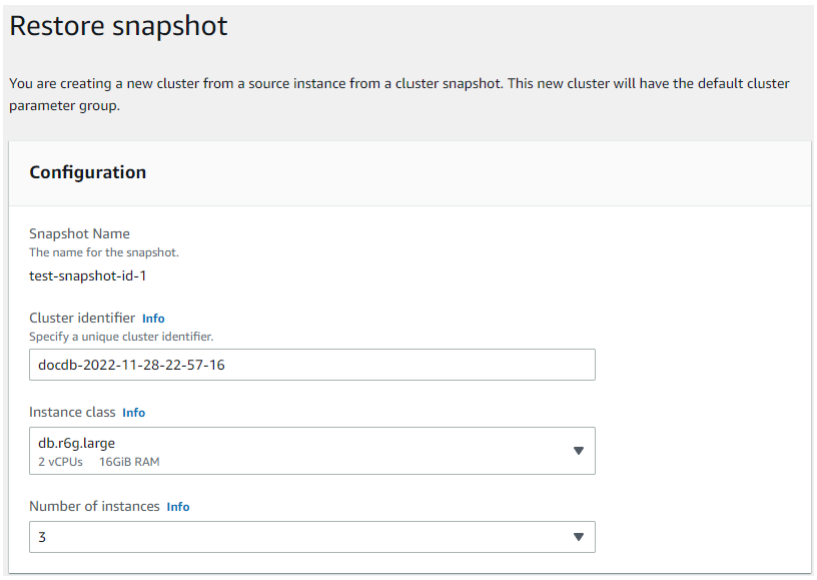
Using the AWS Management Console

Untuk mengembalikan cluster elastis dari snapshot menggunakan: AWS Management Console

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Snapshot.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pilih tombol di sebelah kiri snapshot, yang ingin Anda gunakan untuk memulihkan cluster, di kolom pengenal Snapshot.
4. Pilih Tindakan, lalu Pulihkan.



5. Pada halaman Restore snapshot, masukkan nama untuk cluster baru di kolom Cluster identifier.

> **Note**
>
> Untuk restorasi snapshot manual, Anda harus membuat cluster baru.

6. Di bidang Virtual Private Cloud (VPC), pilih VPC dari daftar drop-down.
7. Untuk grup keamanan Subnet dan VPC, Anda dapat menggunakan default atau memilih tiga subnet pilihan Anda dan hingga tiga grup keamanan VPC (minimal satu).
8. Jika Anda puas dengan konfigurasi klaster, pilih Pulihkan klaster dan tunggu sementara klaster Anda dipulihkan.

Using the AWS CLI

Untuk mengembalikan cluster elastis dari snapshot menggunakan AWS CLI, gunakan `restore-cluster-from-snapshot` operasi dengan parameter berikut:

- **`--cluster-name`**—Wajib. Nama cluster elastis saat ini seperti yang dimasukkan selama pembuatan atau terakhir dimodifikasi.
- **`--snapshot-arn`**—Wajib. Pengidentifikasi ARN dari snapshot yang digunakan untuk memulihkan cluster.
- **`--vpc-security-group-ids`**—Opsional. Satu atau beberapa grup keamanan Amazon EC2 dan Amazon Virtual Private Cloud (VPC) untuk diasosiasikan dengan cluster.
- **`--kms-key-id`**—Opsional. Konfigurasikan pengenal kunci KMS untuk cluster terenkripsi.

  Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN) untuk kunci enkripsi. AWS KMS Jika Anda membuat cluster menggunakan akun Amazon Web Services yang sama yang memiliki kunci enkripsi KMS yang digunakan untuk mengenkripsi cluster baru, Anda dapat menggunakan alias kunci KMS alih-alih ARN untuk kunci enkripsi KMS.

  Jika kunci enkripsi tidak ditentukan KmsKeyId dan jika `StorageEncrypted` parameternya benar, Amazon DocumentDB menggunakan kunci enkripsi default Anda.
- **`--subnet-ids`**—Opsional. Id subnet jaringan.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna dengan informasi`* Anda sendiri..

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Menyalin snapshot cluster elastis

Di Amazon DocumentDB, Anda dapat menyalin snapshot cluster elastis manual dan otomatis dalam wilayah yang sama dan dalam akun yang sama. Pada bagian ini kami menjelaskan cara menyalin snapshot cluster elastis, menggunakan AWS Management Console atau AWS CLI.

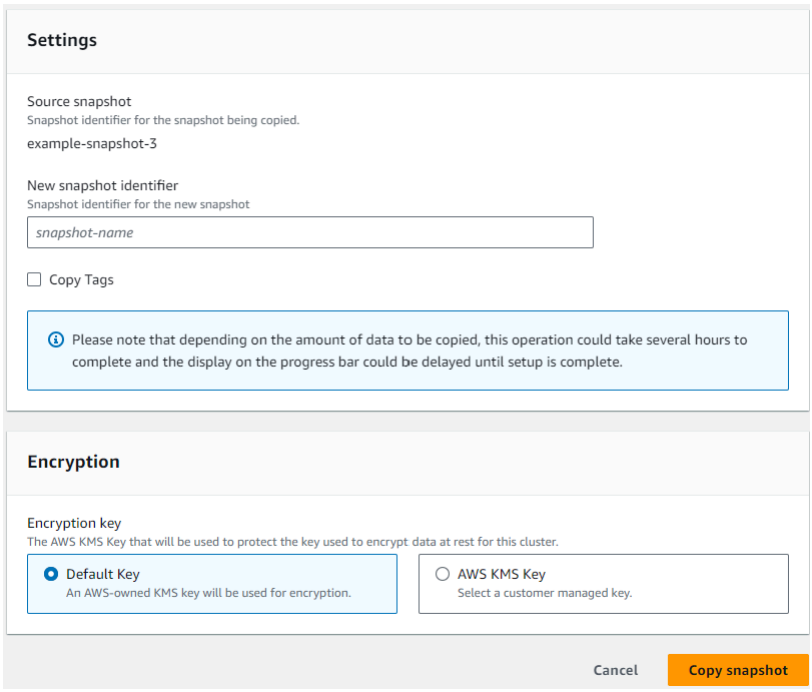
Using the AWS Management Console

Untuk menyalin snapshot cluster elastis menggunakan: AWS Management Console

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Snapshot.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pilih tombol di sebelah kiri snapshot, yang ingin Anda salin, di kolom pengenal Snapshot.
4. Pilih Tindakan, lalu Salin.



5. Untuk pengenal snapshot baru, masukkan nama snapshot baru.
6. Untuk Copy Tags, centang kotak jika Anda ingin menyalin semua tag dari snapshot cluster elastis sumber ke snapshot cluster elastis target.
7. Untuk Enkripsi, pilih kunci AWS KMS default atau kunci KMS yang Anda pilih. Opsi kedua memungkinkan Anda untuk memilih kunci KMS yang sudah Anda buat atau memungkinkan Anda membuat yang baru.
8. Pilih Salin snapshot saat selesai.

Using the AWS CLI

Untuk menyalin snapshot cluster elastis menggunakan AWS CLI, gunakan `copy-cluster-snapshot` operasi dengan parameter berikut:

- **`--source-db-cluster-snapshot-identifier`**—Wajib. Pengidentifikasi snapshot cluster elastis yang ada sedang disalin. Snapshot cluster elastis harus ada dan berada dalam keadaan tersedia. Jika Anda menyalin snapshot ke yang lain Wilayah AWS, pengenal ini harus dalam format ARN untuk sumbernya. Wilayah AWS Parameter ini tidak peka huruf besar kecil.
- **`--target-db-cluster-snapshot-identifier`**—Wajib. Pengidentifikasi snapshot cluster elastis baru untuk dibuat dari snapshot cluster yang ada. Parameter ini tidak peka huruf besar kecil.

Kendala nama snapshot target:

- Tidak dapat berupa nama snapshot yang sudah ada.
- Panjangnya adalah [1—63] huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.

Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic copy-cluster-snapshot \
     --source-cluster-snapshot-arn <sample ARN> \
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Untuk Windows:

```
aws docdb-elastic copy-cluster-snapshot ^
     --source-cluster-snapshot-arn <sample ARN> ^
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Menghapus snapshot cluster elastis

Pada bagian ini kami menjelaskan cara menghapus snapshot cluster elastis, menggunakan AWS Management Console atau AWS CLI.

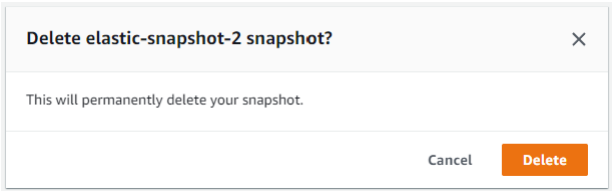
Using the AWS Management Console

Untuk mengembalikan cluster elastis dari snapshot menggunakan: AWS Management Console

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Snapshot.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pilih tombol di sebelah kiri snapshot, yang ingin Anda gunakan untuk memulihkan cluster, di kolom pengenal Snapshot.
4. Pilih Actions (Tindakan), lalu Delete (Hapus).



5. Pada dialog Hapus snapshot “snapshot-name”, pilih Hapus.

Using the AWS CLI

Untuk menghapus snapshot cluster elastis menggunakan AWS CLI, gunakan `delete-cluster-snapshot` operasi dengan parameter berikut:

- **`--snapshot-arn`**—Wajib. Pengidentifikasi ARN dari snapshot yang digunakan untuk memulihkan cluster.

Dalam contoh berikut, ganti setiap *`placeholder input pengguna dengan informasi`* Anda sendiri..

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic delete-cluster-snapshot \
     --snapshot-arn sampleResourceName
```

Untuk Windows:

```
aws docdb-elastic delete-cluster-snapshot ^
     --snapshot-arn sampleResourceName
```

Mengelola cadangan otomatis snapshot cluster elastis

Amazon DocumentDB mengambil snapshot harian dari cluster elastis Anda. Anda dapat menentukan jendela cadangan pilihan dan periode retensi cadangan dalam konfigurasi snapshot cluster elastis baru atau yang sudah ada. Pada bagian ini kami menjelaskan cara mengatur parameter cadangan otomatis dalam snapshot cluster elastis, menggunakan AWS Management Console atau AWS CLI.

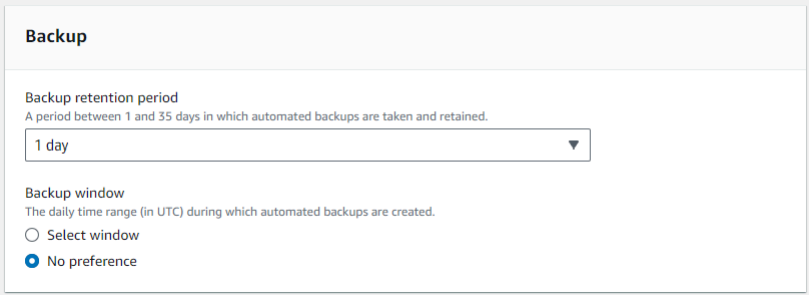
Using the AWS Management Console

Untuk mengatur cadangan otomatis untuk snapshot cluster elastis baru menggunakan: AWS Management Console

1. Masuk ke AWS Management Consoledan buka konsol Amazon DocumentDB.
2. Pada panel navigasi, silakan pilih Klaster.

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu di sudut kiri atas panel navigasi.

3. Pilih tombol di sebelah kiri cluster, yang ingin Anda ubah pengaturan cadangan, di kolom pengenal Cluster.
4. Pilih Tindakan, lalu Ubah.
5. Di bagian Backup, edit bidang sesuai dengan kebutuhan cadangan Anda.



   a. Periode retensi cadangan—Dalam daftar, pilih jumlah hari untuk menyimpan backup otomatis klaster ini sebelum menghapusnya.
   b. Jendela backup—Atur waktu harian dan durasi yang akan digunakan oleh Amazon DocumentDB untuk membuat backup dari klaster ini.
      i. Pilih Pilih jendela jika Anda ingin mengonfigurasi waktu dan durasi saat pencadangan dibuat.

Waktu mulai—Pada daftar pertama, pilih jam waktu mulai (UTC) untuk memulai pencadangan otomatis. Dalam daftar kedua, pilih menit dari jam pilihan Anda untuk memulai backup otomatis.

Durasi—Dalam daftar, pilih jumlah jam yang akan dialokasikan untuk membuat cadangan otomatis.

ii. Pilih Tidak ada preferensi jika Anda ingin Amazon DocumentDB memilih waktu dan durasi saat pencadangan dibuat.

6. Pilih Modify cluster saat selesai.

Using the AWS CLI

Untuk mengatur cadangan otomatis untuk snapshot cluster elastis baru menggunakan AWS CLI, gunakan `create-cluster-snapshot` operasi dengan parameter berikut:

- **`--preferred-backup-window`**—Opsional. Rentang waktu harian yang disukai selama pencadangan otomatis dibuat. Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing. Wilayah AWS

  Batasan:

  - Harus dalam format `hh24:mi-hh24:mi`.
  - Harus dalam Waktu Universal Terkoordinasi (UTC).
  - Tidak boleh bertentangan dengan jendela pemeliharaan yang diinginkan.
  - Harus setidaknya 30 menit.
- **`--backup-retention-period`**—Opsional. Jumlah hari penyimpanan cadangan otomatis. Nilai default adalah 1.

  Batasan:

  - Harus menentukan nilai minimum 1.
  - Rentang adalah dari 1 hingga 35.

> **Note**
>
> Pencadangan otomatis hanya dilakukan saat klaster berada dalam keadaan 'aktif'.

> **Note**
>
> Anda juga dapat memodifikasi `preferred-backup-window` dan `backup-retention-period` parameter cluster elastis yang ada menggunakan `aws docdb-elastic update-cluster` perintah.

Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

`create-cluster`*Contoh berikut membuat cluster sampel cluster elastis Amazon DocumentDB dengan periode retensi untuk pencadangan otomatis 7 hari, dan jendela pencadangan pilihan 18:00-18:30 UTC.*

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Menghentikan dan memulai cluster elastis Amazon DocumentDB

Menghentikan dan memulai cluster elastis Amazon DocumentDB dapat membantu Anda mengelola biaya untuk lingkungan pengembangan dan pengujian. Alih-alih membuat dan menghapus cluster elastis setiap kali Anda menggunakan Amazon DocumentDB, Anda dapat menghentikan sementara klaster Anda saat tidak diperlukan. Anda kemudian dapat memulainya lagi ketika Anda melanjutkan pengujian Anda.

Topik

- Ikhtisar menghentikan dan memulai cluster elastis
- Operasi yang dapat Anda lakukan pada cluster elastis yang berhenti

Ikhtisar menghentikan dan memulai cluster elastis

Selama periode di mana Anda tidak memerlukan kluster elastis Amazon DocumentDB, Anda dapat menghentikan cluster. Anda lalu dapat memulai klaster lagi kapan saja Anda memerlukannya. Memulai dan menghentikan menyederhanakan proses penyiapan dan pembongkaran untuk cluster elastis yang digunakan untuk pengembangan, pengujian, atau aktivitas serupa yang tidak memerlukan ketersediaan berkelanjutan. Anda dapat berhenti dan memulai cluster elastis menggunakan AWS Management Console atau AWS CLI dengan satu tindakan.

Sementara cluster elastis Anda dihentikan, volume penyimpanan cluster tetap tidak berubah. Anda hanya dikenakan biaya untuk penyimpanan, snapshot manual, dan penyimpanan cadangan otomatis dalam jendela penyimpanan yang Anda tentukan. Amazon DocumentDB secara otomatis memulai cluster elastis Anda setelah tujuh hari sehingga tidak ketinggalan pembaruan pemeliharaan yang diperlukan. Ketika cluster Anda dimulai setelah tujuh hari, Anda akan mulai dikenakan biaya untuk penggunaan cluster elastis lagi. Saat klaster dihentikan, Anda tidak dapat melakukan kueri volume penyimpanan karena kueri mengharuskan klaster berada dalam status yang tersedia.

Ketika cluster elastis Amazon DocumentDB dihentikan, cluster tidak dapat dimodifikasi dengan cara apa pun. Ini termasuk menghapus cluster.

Using the AWS Management Console

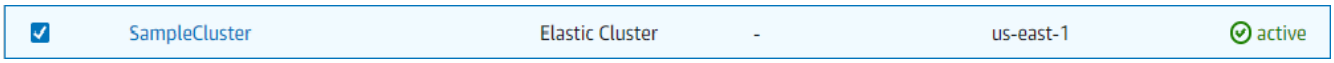
Prosedur berikut menunjukkan kepada Anda cara menghentikan gugus elastis dalam keadaan tersedia, atau memulai gugus elastis yang berhenti.

Untuk menghentikan atau memulai cluster elastis Amazon DocumentDB

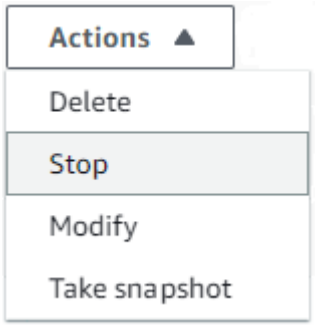
1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Tip**
>
> Jika panel navigasi tidak terlihat di sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

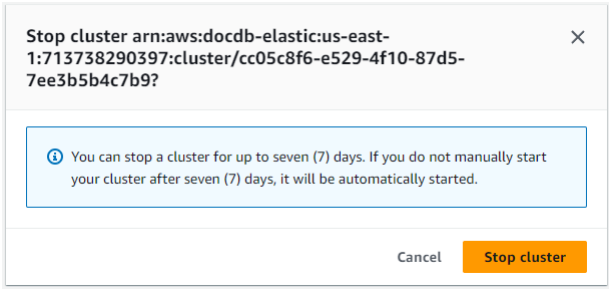
3. Dalam daftar klaster, pilih tombol di sebelah kiri pada klaster yang ingin Anda mulai atau tingkatkan.



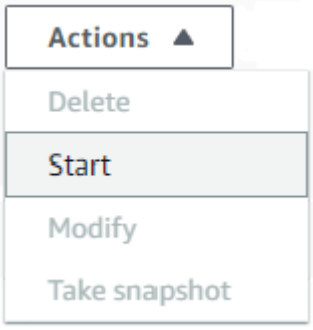
4. Pilih Tindakan, lalu pilih tindakan yang ingin Anda lakukan pada klaster.
   - Jika Anda ingin menghentikan klaster dan klaster tersedia:
     a. Pilih Berhenti.



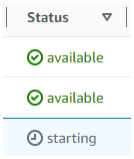
     b. Pada dialog konfirmasi, konfirmasikan bahwa Anda ingin menghentikan cluster elastis dengan memilih Stop cluster, atau agar cluster tetap berjalan, pilih Batal.



   - Jika Anda ingin memulai klaster, dan klaster berhenti, pilih Mulai.



5. Pantau status cluster elastis. Jika Anda memulai cluster, Anda dapat melanjutkan menggunakan cluster saat cluster tersedia. Untuk informasi selengkapnya, lihat Menentukan status klaster.



Using the AWS CLI

Contoh kode berikut menunjukkan cara menghentikan gugus elastis dalam keadaan aktif atau tersedia, atau memulai gugus elastis yang berhenti.

Untuk menghentikan cluster elastis menggunakan AWS CLI, gunakan `stop-cluster` operasi. Untuk memulai klaster yang telah dihentikan, gunakan operasi `start-cluster`. Kedua operasi menggunakan parameter `--cluster-arn`.

Parameter:

- **`--cluster-arn`**—Diperlukan. Pengidentifikasi ARN dari cluster elastis yang ingin Anda hentikan atau mulai.

Example — Untuk menghentikan cluster elastis menggunakan AWS CLI

Dalam contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.

Kode berikut menghentikan cluster elastis dengan ARN dari. `arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`

> **Note**
>
> Cluster elastis harus dalam keadaan aktif atau tersedia.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic stop-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Untuk Windows:

```
aws docdb-elastic stop-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Example — Untuk memulai cluster elastis menggunakan AWS CLI

Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

Kode berikut memulai cluster elastis dengan ARN dari. `arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`

> **Note**
>
> Cluster elastis saat ini harus dihentikan.

Untuk Linux, macOS, atau Unix:

```
aws docdb-elastic start-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Untuk Windows:

```
aws docdb-elastic start-cluster ^
```

```
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Operasi yang dapat Anda lakukan pada cluster elastis yang berhenti

Anda tidak dapat mengubah konfigurasi klaster elastis Amazon DocumentDB saat cluster dihentikan. Anda harus memulai klaster sebelum melakukan tindakan administratif tersebut.

Amazon DocumentDB menerapkan pemeliharaan terjadwal apa pun ke cluster elastis Anda yang berhenti hanya setelah dimulai lagi. Setelah tujuh hari, Amazon DocumentDB secara otomatis memulai cluster elastis yang berhenti sehingga tidak tertinggal terlalu jauh dalam status pemeliharaannya. Saat cluster elastis dimulai ulang, Anda akan mulai dikenakan biaya untuk pecahan di cluster lagi.

Sementara cluster elastis dihentikan, Amazon DocumentDB tidak melakukan pencadangan otomatis apa pun juga tidak memperpanjang periode retensi cadangan.

Enkripsi data saat istirahat untuk klaster elastis Amazon DocumentDB

Topik berikut membantu Anda mempelajari, membuat, dan memantau kunci AWS Key Management Service enkripsi untuk klaster elastis Amazon DocumentDB:

Topik

- Bagaimana klaster elastis Amazon DocumentDB menggunakan hibah di AWS KMS
- Buat kunci terkelola pelanggan
- Memantau kunci enkripsi Anda untuk klaster elastis Amazon DocumentDB
- Pelajari selengkapnya

Klaster elastis Amazon DocumentDB secara otomatis terintegrasi dengan AWS Key Management Service (AWS KMS) untuk manajemen kunci dan menggunakan metode yang dikenal sebagai enkripsi amplop untuk melindungi data Anda. Untuk informasi lebih lanjut tentang enkripsi amplop, lihat Enkripsi amplop dalam Panduan Developer AWS Key Management Service.

An AWS KMS key adalah representasi logis dari kunci. Kunci KMS mencakup metadata, seperti ID kunci, tanggal pembuatan, deskripsi, dan status kunci. Kunci KMS juga berisi bahan utama yang

digunakan untuk mengenkripsi dan mendekripsi data. Untuk informasi selengkapnya tentang kunci KMS, lihat AWS KMS keys dalam Panduan Developer AWS Key Management Service.

Cluster elastis Amazon DocumentDB mendukung enkripsi dengan dua jenis kunci:

- AWSkunci yang dimiliki — cluster elastis Amazon DocumentDB menggunakan kunci ini secara default untuk secara otomatis mengenkripsi data yang dapat diidentifikasi secara pribadi. Anda tidak dapat melihat, mengelola, atau menggunakan kunci AWS yang dimiliki, atau audit penggunaannya. Namun, Anda tidak perlu mengambil tindakan atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda. Untuk informasi selengkapnya, lihat kunci yang AWS dimiliki di Panduan AWS Key Management Service Pengembang.
- Kunci yang dikelola pelanggan — Simetris AWS KMS keys yang Anda buat, miliki, dan kelola. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:
  - Menetapkan dan memelihara kebijakan utama
  - Menetapkan dan memelihara kebijakan dan hibah IAM
  - Mengaktifkan dan menonaktifkan kebijakan utama
  - Memutar bahan kriptografi kunci
  - Menambahkan tanda
  - Membuat alias kunci
  - Kunci penjadwalan untuk penghapusan

  Untuk informasi selengkapnya, lihat Kunci yang dikelola pelanggan di Panduan AWS Key Management Service Developer.

> **Important**
>
> Anda harus menggunakan kunci KMS enkripsi simetris untuk mengenkripsi klaster Anda karena Amazon DocumentDB mendukung hanya kunci KMS enkripsi simetris. Jangan gunakan kunci KMS asimetris untuk mencoba mengenkripsi data di klaster elastis Amazon DocumentDB Anda. Untuk informasi selengkapnya, lihat Kunci asimetris di AWS KMS dalam Panduan AWS Key Management Service Pengembang.
> Jika Amazon DocumentDB tidak bisa lagi mendapatkan akses ke kunci enkripsi untuk sebuah klaster — misalnya, saat akses ke kunci dicabut — klaster terenkripsi masuk ke status terminal. Dalam hal ini, Anda hanya dapat memulihkan klaster dari backup. Untuk Amazon DocumentDB, backup selalu diaktifkan selama 1 hari. Selain itu, jika Anda menonaktifkan

kunci untuk klaster Amazon DocumentDB terenkripsi, Anda akhirnya akan kehilangan akses baca dan tulis ke klaster tersebut. Ketika Amazon DocumentDB menemukan klaster yang dienkripsi dengan kunci yang tidak dapat diaksesnya, klaster akan dimasukkan ke status terminal. Dalam kondisi ini, klaster DB tidak lagi tersedia, dan status basis data saat ini tidak dapat dipulihkan. Untuk memulihkan klaster, Anda harus mengaktifkan kembali akses ke kunci enkripsi untuk Amazon DocumentDB, lalu memulihkan klaster dari backup.

Important

Anda tidak dapat mengubah kunci KMS untuk klaster terenkripsi setelah Anda membuatnya. Pastikan Anda menentukan persyaratan kunci enkripsi sebelum membuat klaster elastis terenkripsi Anda.

Bagaimana klaster elastis Amazon DocumentDB menggunakan hibah di AWS KMS

Klaster elastis Amazon DocumentDB memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan Anda.

Saat Anda membuat klaster yang dienkripsi dengan kunci yang dikelola pelanggan, klaster elastis Amazon DocumentDB membuat hibah atas nama Anda dengan mengirimkan permintaan ke. `CreateGrant` AWS KMS Hibah AWS KMS digunakan untuk memberikan akses cluster elastis Amazon DocumentDB ke kunci KMS di akun pelanggan.

Klaster elastis Amazon DocumentDB memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan Anda untuk operasi internal berikut:

- Kirim `DescribeKey` permintaan AWS KMS untuk memverifikasi bahwa ID kunci KMS yang dikelola pelanggan simetris, yang dimasukkan saat membuat pelacak atau pengumpulan geofence, valid.
- Kirim `GenerateDataKey` permintaan AWS KMS untuk menghasilkan kunci data yang dienkripsi oleh kunci yang dikelola pelanggan Anda.
- Kirim `Decrypt` permintaan AWS KMS untuk mendekripsi kunci data terenkripsi sehingga mereka dapat digunakan untuk mengenkripsi data Anda.

- Anda dapat mencabut akses ke hibah, atau menghapus akses layanan ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, klaster elastis Amazon DocumentDB tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, yang memengaruhi operasi yang bergantung pada data tersebut.

Buat kunci terkelola pelanggan

Anda dapat membuat kunci yang dikelola pelanggan simetris dengan menggunakan AWS Management Console atau AWS KMS API.

Pembuatan kunci yang dikelola pelanggan simetris

Ikuti langkah-langkah untuk Membuat kunci yang dikelola pelanggan simetris di Panduan AWS Key Management Service Pengembang.

Kebijakan kunci

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki satu kebijakan kunci, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya.
Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan utama. Untuk informasi selengkapnya, lihat informasi akses kunci KMS yang terletak di AWS Key Management Serviceikhtisar Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci yang dikelola pelanggan Anda dengan sumber daya klaster elastis Amazon DocumentDB, operasi API berikut harus diizinkan dalam kebijakan utama:

- `kms:CreateGrant`- Menambahkan hibah ke kunci yang dikelola pelanggan. Memberikan akses kontrol ke kunci KMS tertentu, yang memungkinkan akses untuk memberikan operasi yang diperlukan Amazon Location Service. Untuk informasi selengkapnya tentang penggunaan hibah, lihat Hibah di AWS KMS dalam Panduan AWS Key Management ServicePengembang.
- `kms:DescribeKey`- Menyediakan pelanggan dikelola rincian kunci untuk memungkinkan Docdb Elastic untuk memvalidasi kunci.
- `kms:Decrypt`- Memungkinkan Docdb Elastic untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.
- `kms:GenerateDataKey`- Memungkinkan Docdb Elastic untuk menghasilkan kunci data terenkripsi dan menyimpannya karena kunci data tidak segera digunakan untuk mengenkripsi.

Untuk informasi selengkapnya, lihat Izin untuk AWS layanan dalam kebijakan utama dan Pemecahan Masalah akses kunci di Panduan AWS Key Management ServiceDeveloper.

Membatasi akses kunci yang dikelola pelanggan melalui kebijakan IAM

Selain kebijakan utama KMS, Anda juga dapat membatasi izin kunci KMS dalam kebijakan IAM.

Anda dapat membuat kebijakan IAM lebih ketat dalam berbagai cara. Misalnya, untuk mengizinkan kunci yang dikelola pelanggan digunakan hanya untuk permintaan yang berasal di klaster elastis Amazon DocumentDB, Anda dapat menggunakan kunci `kms:ViaService` syarat dengan nilai tersebut. `docdb-elastic.<region-name>.amazonaws.com`

Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS di AWS Key Management Service Panduan Developer.

Memantau kunci enkripsi Anda untuk klaster elastis Amazon DocumentDB

Saat Anda menggunakan kunci yang dikelola AWS KMS key pelanggan dengan sumber daya Docdb Elastic, Anda dapat menggunakan AWS CloudTrail atau Amazon CloudWatch Logs untuk melacak permintaan yang dikirim Docdb Elastic. AWS KMS

Contoh berikut adalah AWS CloudTrail peristiwa untuk`CreateGrant`, `GenerateDataKeyWithoutPlainTextDecrypt`, dan `DescribeKey` untuk memantau AWS KMS key operasi yang dipanggil oleh klaster elastis Amazon DocumentDB untuk mengakses data yang dienkripsi oleh kunci yang dikelola pelanggan Anda:

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"
}
```

Pelajari selengkapnya

Sumber daya berikut memberikan informasi lebih lanjut tentang enkripsi data saat istirahat:

- Untuk informasi selengkapnya tentang AWS KMS konsep, lihat konsep AWS Key Management Service dasar di Panduan AWS Key Management Service Pengembang.
- Untuk informasi selengkapnya tentang AWS KMS keamanan, lihat Praktik terbaik keamanan AWS Key Management Service dalam Panduan AWS Key Management Service Developer.

Peran terkait layanan dalam kelompok elastis

Cluster elastis Amazon DocumentDB AWS Identity and Access Management menggunakan peran terkait layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke cluster elastis Amazon DocumentDB. Peran terkait layanan telah ditentukan sebelumnya oleh klaster elastis Amazon DocumentDB dan menyertakan semua izin yang diperlukan layanan untuk memanggil layanan lain atas nama Anda. AWS

Peran terkait layanan membuat penggunaan cluster elastis Amazon DocumentDB lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Cluster elastis Amazon DocumentDB mendefinisikan izin peran terkait layanannya, dan kecuali ditentukan lain, hanya cluster elastis Amazon DocumentDB yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya. Anda dapat menghapus peran hanya setelah terlebih dahulu menghapus sumber daya terkaitnya. Ini melindungi sumber daya cluster elastis Amazon DocumentDB karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat AWS layanan yang bekerja dengan IAM dan cari layanan yang ditandai dengan Ya di kolom Peran Tertaut Layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk cluster elastis

Cluster elastis Amazon DocumentDB menggunakan peran terkait layanan yang diberi nama untuk `AWS ServiceRoleForDocDB-Elastic` memungkinkan klaster elastis Amazon DocumentDB memanggil layanan atas nama kluster Anda. AWS

Peran terkait layanan ini memiliki kebijakan izin yang menyertainya bernama `AmazonDocDB-ElasticServiceRolePolicy` yang memberikannya izin untuk beroperasi di akun Anda. Kebijakan

izin peran memungkinkan klaster elastis Amazon DocumentDB menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Jika Anda menemukan pesan galat berikut: “Tidak dapat membuat sumber daya. Verifikasi bahwa Anda memiliki izin untuk membuat peran terkait layanan. Kalau tidak, tunggu dan coba lagi nanti.” , pastikan Anda mengaktifkan izin berikut:

```
{
"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"
        }
    }
}
```

Untuk informasi selengkapnya, lihat Izin peran terkait layanan di Panduan Pengguna AWS Identity and Access Management.

Membuat peran terkait layanan untuk cluster elastis Amazon DocumentDB

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat instans DB, cluster elastis Amazon DocumentDB membuat peran terkait layanan untuk Anda.

Mengedit peran terkait layanan untuk cluster elastis Amazon DocumentDB

Cluster elastis Amazon DocumentDB tidak memungkinkan Anda mengedit peran terkait layanan. `AWS ServiceRoleForDocDB-Elastic` Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran terkait layanan di Panduan Pengguna AWS Identity and Access Management.

Menghapus peran terkait layanan untuk cluster elastis Amazon DocumentDB

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami menyarankan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas tak terpakai yang tidak dipantau atau dipelihara secara aktif. Namun demikian, Anda harus menghapus semua klaster Anda sebelum Anda dapat menghapus peran tertaut layanan.

Membersihkan peran terkait layanan

Sebelum dapat menggunakan IAM untuk menghapus peran tertaut layanan, Anda harus mengonfirmasi terlebih dahulu bahwa peran tersebut tidak memiliki sesi aktif dan menghapus sumber daya yang digunakan oleh peran tersebut.

Untuk memeriksa apakah peran terkait layanan memiliki sesi aktif di konsol IAM:

1. Masuk ke AWS Management Consoledan buka konsol IAM.
2. Di panel navigasi konsol IAM, pilih Peran. Lalu pilih nama (bukan kotak centang) dari peran `AWS ServiceRoleForDocDB-Elastic`.

3. Di halaman Ringkasan untuk peran yang dipilih, pilih tab Penasihat Akses.

> **Note**
>
> Jika Anda tidak yakin apakah cluster elastis Amazon DocumentDB `AWS ServiceRoleForDocDB-Elastic` menggunakan peran tersebut, Anda dapat mencoba menghapus peran tersebut. Jika layanan menggunakan peran, maka penghapusan gagal dan Anda dapat melihat Wilayah AWS di mana peran tersebut digunakan. Jika peran tersebut sedang digunakan, Anda harus menunggu hingga sesi ini berakhir sebelum dapat menghapus peran tersebut. Anda tidak dapat mencabut sesi untuk peran terkait layanan. Jika Anda ingin menghapus `AWS ServiceRoleForDocDB-Elastic` peran, Anda harus terlebih dahulu menghapus semua cluster Anda.

Menghapus semua klaster Anda

Untuk menghapus cluster di konsol Amazon DocumentDB:

1. Masuk ke AWS Management Console dan buka konsol Amazon DocumentDB.
2. Di panel navigasi, pilih Klaster.
3. Pilih cluster yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Jika Anda diminta untuk Membuat Snapshot akhir? , pilih Ya atau Tidak.
6. Jika Anda memilih Ya pada langkah sebelumnya, untuk Nama snapshot akhir masukkan nama snapshot akhir Anda.
7. Pilih Hapus.

> **Note**
>
> Anda dapat menggunakan konsol IAM, IAM CLI, atau IAM API untuk menghapus `AWS ServiceRoleForDocDB-Elastic`-peran terkait layanan. Untuk informasi selengkapnya, lihat Menghapus peran terkait layanan di Panduan Pengguna AWS Identity and Access Management.

Memantau Amazon DocumentDB

Memantau layanan AWS Anda adalah bagian penting untuk menjaga sistem Anda tetap sehat dan berfungsi secara optimal. Sebaiknya kumpulkan data pemantauan dari semua bagian solusi AWS Anda sehingga Anda dapat lebih mudah men-debug dan memperbaiki kegagalan atau degradasi, jika itu terjadi. Sebelum Anda mulai memantau solusi AWS Anda, kami merekomendasikan Anda mempertimbangkan dan merumuskan jawaban untuk pertanyaan berikut:

- Apa tujuan pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang bertanggung jawab untuk melakukan pemantauan?
- Siapa yang akan diberitahu dan dengan cara apa jika terjadi kesalahan?

Untuk memahami pola performa Anda saat ini, mengidentifikasi anomali performa, dan merumuskan metode untuk mengatasi masalah, Anda harus menetapkan metrik performa dasar untuk berbagai waktu dan dalam kondisi beban yang berbeda. Saat Anda memantau solusi AWS Anda, kami merekomendasikan Anda menyimpan data pemantauan historis Anda untuk referensi di masa mendatang dan untuk menetapkan garis dasar Anda.

Secara umum, nilai yang dapat diterima untuk metrik performa bergantung pada seperti apa garis dasar Anda dan apa yang dilakukan aplikasi Anda. Selidiki varian yang konsisten atau sedang tren dari garis dasar Anda. Berikut ini adalah saran tentang jenis metrik tertentu:

- Penggunaan CPU atau RAM yang tinggi — Nilai tinggi untuk penggunaan CPU atau RAM mungkin tepat, asalkan sesuai dengan tujuan Anda untuk aplikasi Anda (seperti throughput atau konkurensi) dan diharapkan.
- Konsumsi volume penyimpanan — Selidiki konsumsi penyimpanan (`VolumeBytesUsed`) jika ruang yang digunakan secara konsisten pada atau di atas 85 persen dari total ruang volume penyimpanan. Tentukan apakah Anda dapat menghapus data dari volume penyimpanan atau mengarsipkan data ke sistem lain untuk mengosongkan ruang. Untuk informasi selengkapnya, lihat Penyimpanan Amazon DocumentDB dan Kuota dan Batas Amazon DocumentDB.

- Lalu lintas jaringan — Untuk lalu lintas jaringan, bicarakan dengan administrator sistem Anda untuk memahami throughput yang diharapkan untuk jaringan domain dan koneksi internet Anda. Selidiki lalu lintas jaringan jika throughput secara konsisten lebih rendah dari yang diharapkan.
- Koneksi basis data — Pertimbangkan untuk membatasi koneksi basis data jika Anda melihat jumlah koneksi pengguna yang tinggi bersamaan dengan penurunan performa instans dan waktu respons. Jumlah koneksi pengguna terbaik untuk instans Anda akan bervariasi berdasarkan kelas instans Anda dan kompleksitas operasi yang dilakukan.
- Metrik IOPS—Nilai yang diharapkan untuk metrik IOPS bergantung pada spesifikasi disk dan konfigurasi server, jadi gunakan garis dasar Anda untuk mengetahui apa yang khusus. Selidiki apakah nilainya secara konsisten berbeda dari garis dasar Anda. Untuk kinerja IOPS terbaik, pastikan pekerjaan Anda yang biasa cocok dengan memori untuk meminimalkan operasi baca dan tulis.

Amazon DocumentDB (dengan kompatibilitas MongoDB) menyediakan berbagaiCloudWatch metrik Amazon DocumentDB yang dapat Anda pantau untuk menentukan kondisi dan performa klaster dan instans Amazon DocumentDB Anda. Anda dapat melihat metrik Amazon DocumentDB menggunakan berbagai alat, termasuk konsol Amazon DocumentDB,,AWS CLI,CloudWatch dan Performance Insights.

Topik

- Memantau Status klaster Amazon DocumentDB
- Memantau Status instans Amazon DocumentDB
- Menampilkan Rekomendasi Amazon DocumentDB
- Menggunakan Langganan Peristiwa Amazon DocumentDB
- Memantau Amazon DocumentDB dengan CloudWatch
- Mencatat Panggilan API Amazon DocumentDB dengan AWS CloudTrail
- Membuat Profil Operasi Amazon DocumentDB
- Memantau dengan Wawasan Kinerja

Memantau Status klaster Amazon DocumentDB

Status klaster menunjukkan kondisi klaster. Anda dapat melihat status klaster dengan menggunakan konsol Amazon DocumentDB atau perintah AWS CLI `describe-db-clusters`.

Topik

- Nilai Status Klaster
- Memantau status klaster

Nilai Status Klaster

Tabel berikut mencantumkan nilai yang valid untuk status klaster.

status klaster	Deskripsi
`active`	Cluster aktif. Status ini hanya berlaku untuk cluster elastis.
`available`	Klaster sehat dan tersedia. Status ini hanya berlaku untuk klaster berbasis instans.
`backing-up`	Klaster saat ini sedang dicadangkan.
`creating`	Klaster sedang dibuat. Klaster tidak dapat diakses ketika sedang dibuat.
`deleting`	Klaster sedang dihapus. Klaster tidak dapat diakses ketika sedang dihapus.
`failing-over`	Sebuah failover dari instans primer ke replika Amazon DocumentDB sedang dilakukan .
`inaccessible-encryption-credentials`	Kunci AWS KMS yang digunakan untuk mengenkripsi atau mendekripsi klaster tidak dapat diakses.

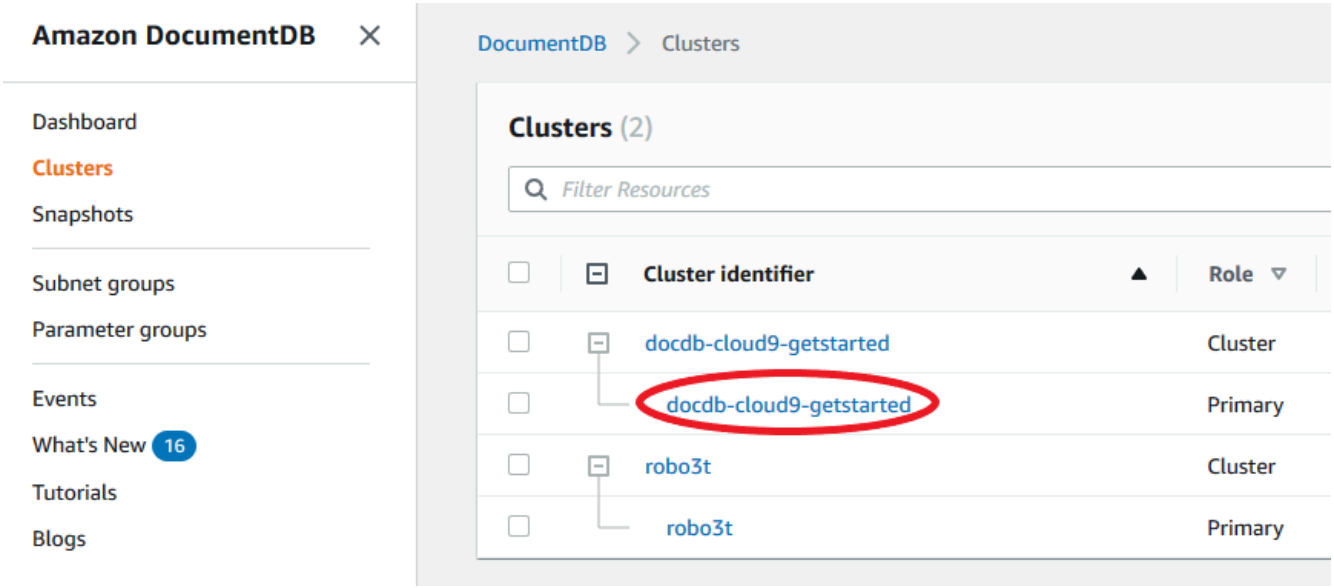
status klaster	Deskripsi
`maintenance`	Pembaruan pemelihar aan sedang diterapkan ke klaster. Status ini digunakan untuk pemeliharaan tingkat klaster yang dijadwalkan jauh sebelumnya oleh Amazon DocumentDB.
`migrating`	Snapshot klaster sedang dipulihkan ke sebuah klaster.
`migration-failed`	Migrasi gagal.
`modifying`	Klaster sedang dimodifikasi karena permintaan pelanggan untuk memodifikasi klaster.
`renaming`	Klaster sedang diganti namanya karena permintaan pelanggan untuk mengganti namanya.
`resetting-master-credentials`	Kredensial utama untuk klaster sedang diatur ulang karena permintaan pelanggan untuk mengatur ulang.
`upgrading`	Versi mesin klaster sedang ditingkatkan.

Memantau status klaster

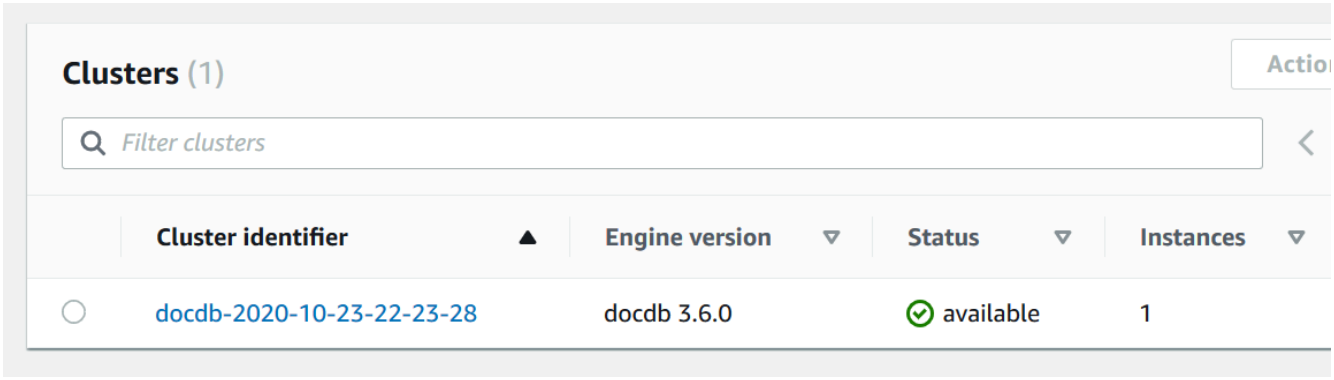
Using the AWS Management Console

Saat menggunakan AWS Management Console untuk menentukan status klaster, gunakan prosedur berikut.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.
3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Dalam kolom Pengidentifikasi klaster, temukan nama instans yang Anda minati. Kemudian, untuk menemukan status instans, baca baris tersebut ke kolom Status, seperti yang ditunjukkan di bawah ini.



Using the AWS CLI

Saat menggunakan AWS CLI untuk menentukan status klaster, gunakan operasi `describe-db-clusters`. Kode berikut menemukan status klaster `sample-cluster`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Untuk Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Output dari operasi ini terlihat seperti berikut ini.

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Memantau Status instans Amazon DocumentDB

Amazon DocumentDB memberikan informasi tentang kondisi saat ini dari setiap instans yang dikonfigurasi dalam database.

Ada tiga jenis status yang dapat Anda lihat untuk instans Amazon DocumentDB:

- Status instans: Status ini ditampilkan di kolom Status tabel klaster diAWS Management Console dan menunjukkan kondisi siklus hidup saat ini dari instance. Nilai yang ditampilkan di kolom Status berasal dari`Status` bidang respons`DescribeDBCluster` API.
- Status kesehatan instans: Status ini ditampilkan di kolom kesehatan Instance dari tabel klaster diAWS Management Console dan menunjukkan apakah mesin database, komponen yang bertanggung jawab untuk mengelola dan mengambil data, sedang berjalan. Nilai yang ditampilkan di kolom kesehatan Instans didasarkan pada metrik`EngineUptime` sistem Amazon CloudWatch.
- Status pemeliharaan: Status ini ditampilkan di kolom Pemeliharaan tabel klaster diAWS Management Console dan menunjukkan status setiap peristiwa pemeliharaan yang perlu diterapkan ke instans. Status pemeliharaan tidak tergantung pada status instans lain dan berasal dari`PendingMaintenanceAction` API. Untuk informasi lebih lanjut tentang status pemeliharaan, lihat Menjaga Amazon DocumentDB.

Topik

- Nilai status status status status status status status status
- Memantau status instans menggunakanAWS Management Console atauAWS CLI
- Nilai status status status status status status status status status kesehatan
- Memantau status kesehatan instans menggunakanAWS Management Console

Nilai status status status status status status status status

Tabel berikut mencantumkan kemungkinan nilai status untuk instans dan cara Anda ditagih untuk setiap status. Ini menunjukkan apakah Anda akan ditagih untuk instans dan penyimpanan, hanya penyimpanan, atau tidak ditagih. Untuk semua status instans, Anda selalu ditagih untuk penggunaan pencadangan.

Status instans	Ditagih	Deskripsi
`available`	Ditagih	Instans sehat dan tersedia.
`backing-up`	Ditagih	Instans saat ini sedang dicadangkan.
`configuring-log-exports`	Ditagih	Menerbitkan file log ke AmazonCloudWatch Logs diaktifkan atau dinonaktifkan untuk instans ini.
`creating`	Tidak ditagih	Instans sedang dibuat. Instans tidak dapat diakses saat sedang dibuat.
`deleting`	Tidak ditagih	Instans sedang dihapus.
`failed`	Tidak ditagih	Instans gagal dan Amazon DocumentDB tidak dapat memulihkannya. Untuk memulihkan data, lakukanpoint-in-time pemulihan ke waktu terbaru yang dapat dipulihkan dari instans.
`inaccessible-encryption-credentials`	Tidak ditagih	Kunci AWS KMS yang digunakan untuk mengenkripsi atau mendekripsi instans tidak dapat diakses.

Status instans	Ditagih	Deskripsi
`incompatible-network`	Tidak ditagih	Amazon DocumentDB mencoba melakukan tindakan pemulihan pada instans tetapi tidak dapat melakukannya karena VPC berada dalam status yang mencegah penyelesaian tindakan. Status ini dapat terjadi jika, misalnya, semua alamat IP yang tersedia di subnet sedang digunakan dan Amazon DocumentDB tidak bisa mendapatkan alamat IP untuk instans tersebut.
`maintenance`	Ditagih	Amazon DocumentDB menerapkan pembaruan pemeliharaan ke instans. Status ini digunakan untuk pemeliharaan tingkat instans yang dijadwalkan jauh sebelumnya oleh Amazon DocumentDB. Kami sedang mengevaluasi cara untuk mengekspos tindakan pemeliharaan tambahan kepada pelanggan melalui status ini.
`modifying`	Ditagih	Instans sedang dimodifikasi karena permintaan untuk memodifikasi instans.
`rebooting`	Ditagih	Instans sedang dinyalakan ulang karena permintaan atau proses Amazon DocumentDB yang memerlukan penyalaan ulang instans.
`renaming`	Ditagih	Instans sedang diganti namanya karena ada permintaan untuk mengganti namanya.
`resetting-master-credentials`	Ditagih	Kredensial utama untuk instans sedang diatur ulang karena permintaan untuk mengatur ulang.

Status instans	Ditagih	Deskripsi
`restore-error`	Ditagih	Instans mengalami kesalahan saat mencoba memulihkan kepoint-in-time atau dari snapshot.
`starting`	Ditagih untuk penyimpanan	Instans dimulai.
`stopped`	Ditagih untuk penyimpanan	Instans dihentikan.
`stopping`	Ditagih untuk penyimpanan	Instans sedang dihentikan.
`storage-full`	Ditagih	Instans telah mencapai alokasi kapasitas penyimpanannya. Ini adalah status kritis dan harus segera diperbaiki; tingkatkan penyimpanan Anda dengan memodifikasi instans. AturCloudWatch alarm Amazon untuk memperingatkan Anda saat ruang penyimpanan hampir habis agar Anda tidak mengalami situasi ini.

Memantau status instans menggunakanAWS Management Console atauAWS CLI

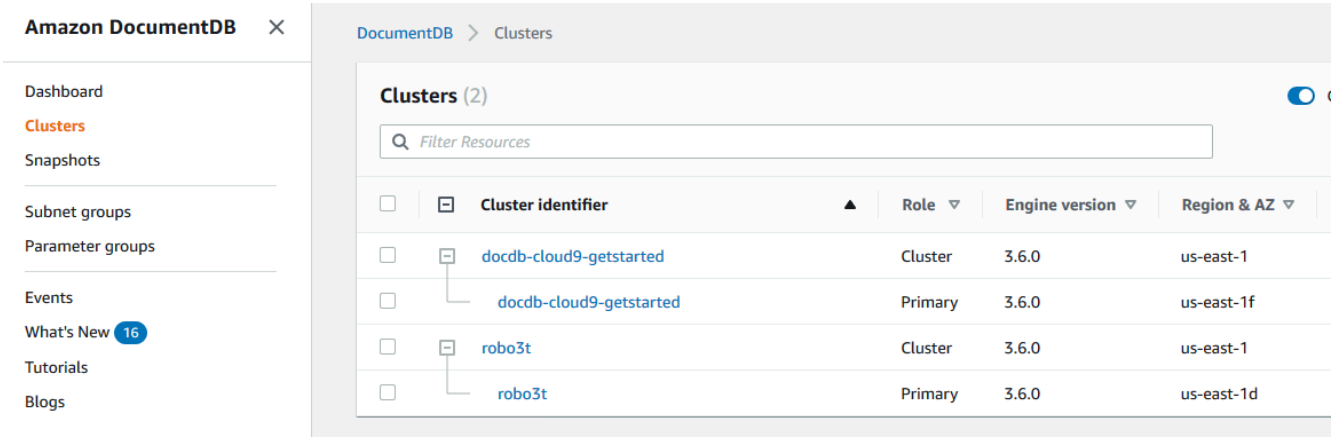
GunakanAWS Management Console atauAWS CLI untuk memantau status contoh Anda.

Using the AWS Management Console

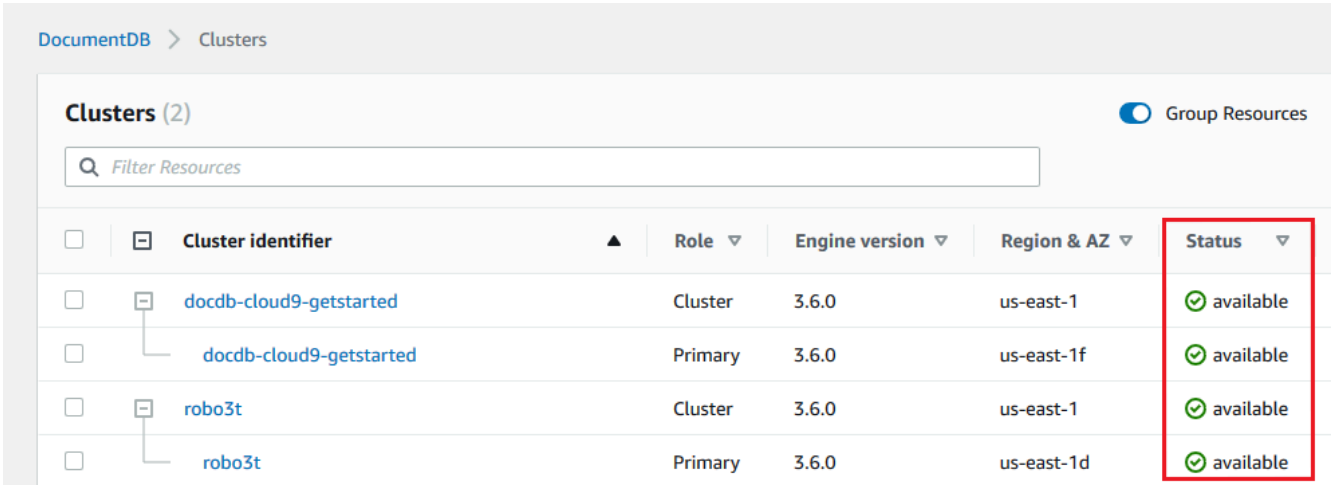
Saat menggunakan AWS Management Console untuk menentukan status klaster, gunakan prosedur berikut.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Note**
>
> Perhatikan bahwa di kotak navigasi Klaster, kolom Pengidentifikasi Klaster dan instans. Instans tercantum di bawah klaster, mirip dengan gambar di bawah ini.



3. Temukan nama instans yang Anda minati. Kemudian, untuk menemukan status instans, baca sepanjang baris tersebut ke kolom Status, seperti yang ditunjukkan berikut ini.



Using the AWS CLI

Saat menggunakan AWS CLI untuk menentukan status klaster, gunakan operasi `describe-db-instances`. Kode berikut menemukan status instans `sample-cluster-instance-01`.

Untuk Linux, macOS, atau Unix:

```
aws docdb describe-db-instances \
          --db-instance-identifier sample-cluster-instance-01  \
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Untuk Windows:

```
aws docdb describe-db-instances ^
          --db-instance-identifier sample-cluster-instance-01  ^
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Output dari operasi ini terlihat seperti berikut.

```
[
          [
              "sample-cluster-instance-01",
              "available"
          ]
      ]
```

Nilai status status status status status status status status status kesehatan

Tabel berikut mencantumkan kemungkinan nilai status kesehatan untuk instans. Kolom kesehatan Instance, yang terletak di tabel Cluster diAWS Management Console, menunjukkan apakah mesin database, komponen yang bertanggung jawab untuk menyimpan, mengelola, dan mengambil data, beroperasi secara normal. Kolom ini juga menunjukkan apakah metrik`EngineUptime` sistem, tersedia diCloudWatch, menunjukkan status kesehatan setiap instans.

Status kesehatan instance	Deskripsi
sehat	Mesin database berjalan di instans Amazon DocumentDB.
tidak sehat	Mesin database tidak berjalan atau telah restart kurang dari satu menit yang lalu.

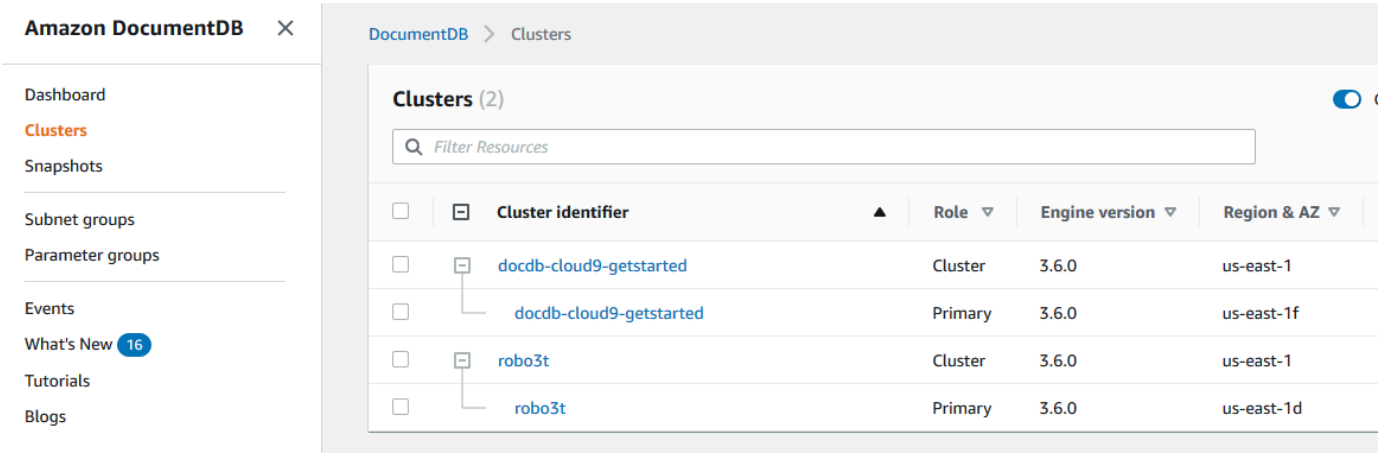
Memantau status kesehatan instans menggunakanAWS Management Console

GunakanAWS Management Console untuk memantau status kesehatan instans Anda.

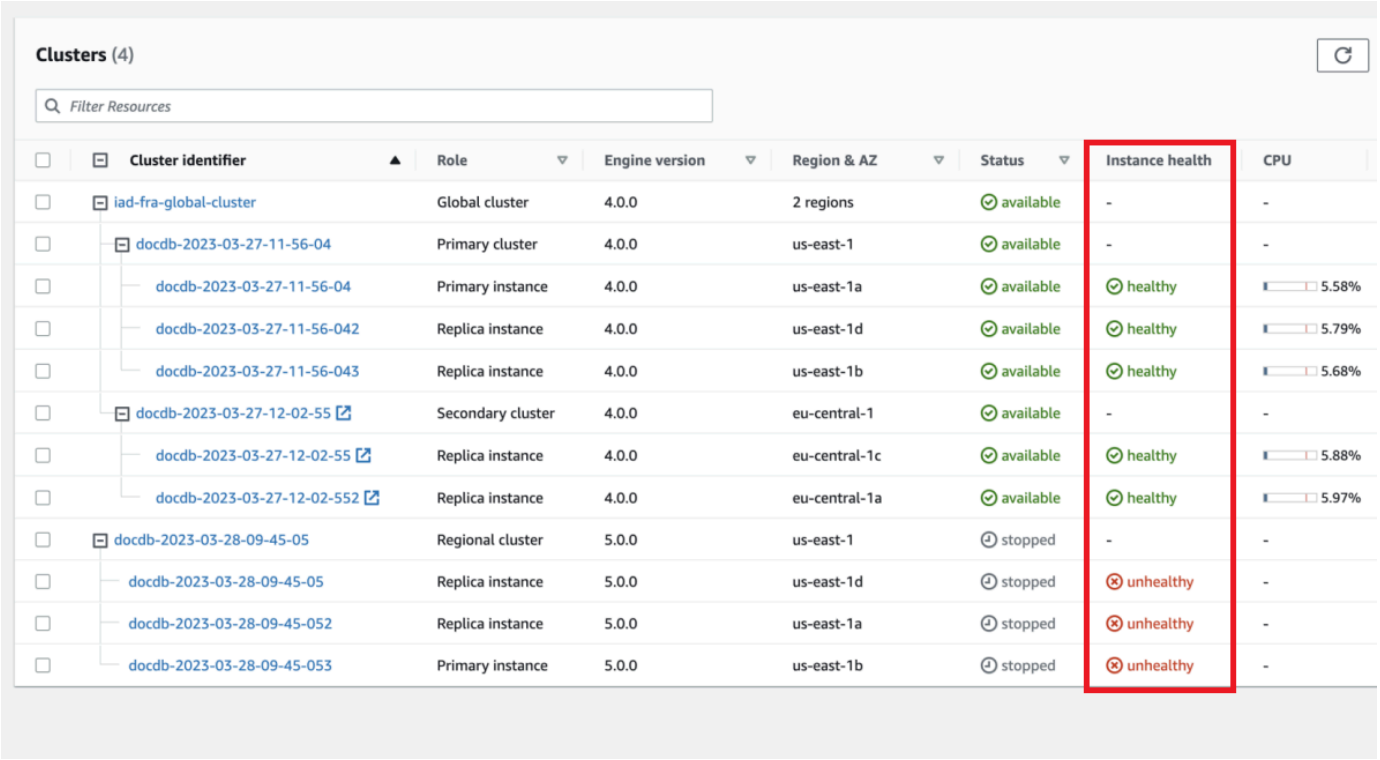
Saat menggunakanAWS Management Console, gunakan langkah-langkah berikut untuk memahami status kesehatan instans.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

> **Note**
>
> Di kotak navigasi Klaster, kolom Pengidentifikasi Klaster dan instans. Instans tercantum di bawah klaster, mirip dengan gambar di bawah ini.



3. Temukan nama instans yang Anda minati. Kemudian, untuk menemukan status instans, baca baris tersebut ke kolom kesehatan instance, seperti yang ditunjukkan pada gambar berikut:



> **Note**
>
> Instance status kesehatan polling terjadi setiap 60 detik dan didasarkan pada metrikCloudWatch`EngineUptime` sistem. Nilai dalam kolom Kesehatan Instans diperbarui secara otomatis.

Menampilkan Rekomendasi Amazon DocumentDB

Amazon DocumentDB menyediakan daftar rekomendasi otomatis untuk sumber daya database, seperti instans dan klaster. Rekomendasi ini memberikan panduan praktik terbaik dengan menganalisis konfigurasi klaster dan instans Anda.

Sebagai contoh rekomendasi ini, lihat berikut:

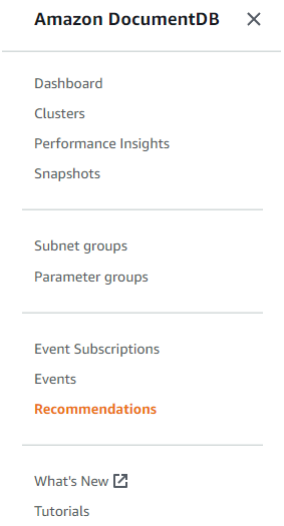
Tipe	Deskripsi	Rekomendasi	Informasi tambahan
Satu instans	Cluster hanya berisi satu instance	Performa dan ketersediaan: kami menyarankan	Ketersediaan Tinggi dan Replikasi Amazon DocumentDB

Tipe	Deskripsi	Rekomendasi	Informasi tambahan
		penambahan instans lain dengan kelas instans yang sama di Availability Zone yang berbeda.	

Amazon DocumentDB memberikan rekomendasi untuk sumber daya saat sumber daya diciptakan atau diubah. Amazon DocumentDB juga secara berkala memindai sumber daya Anda dan memberikan rekomendasi.

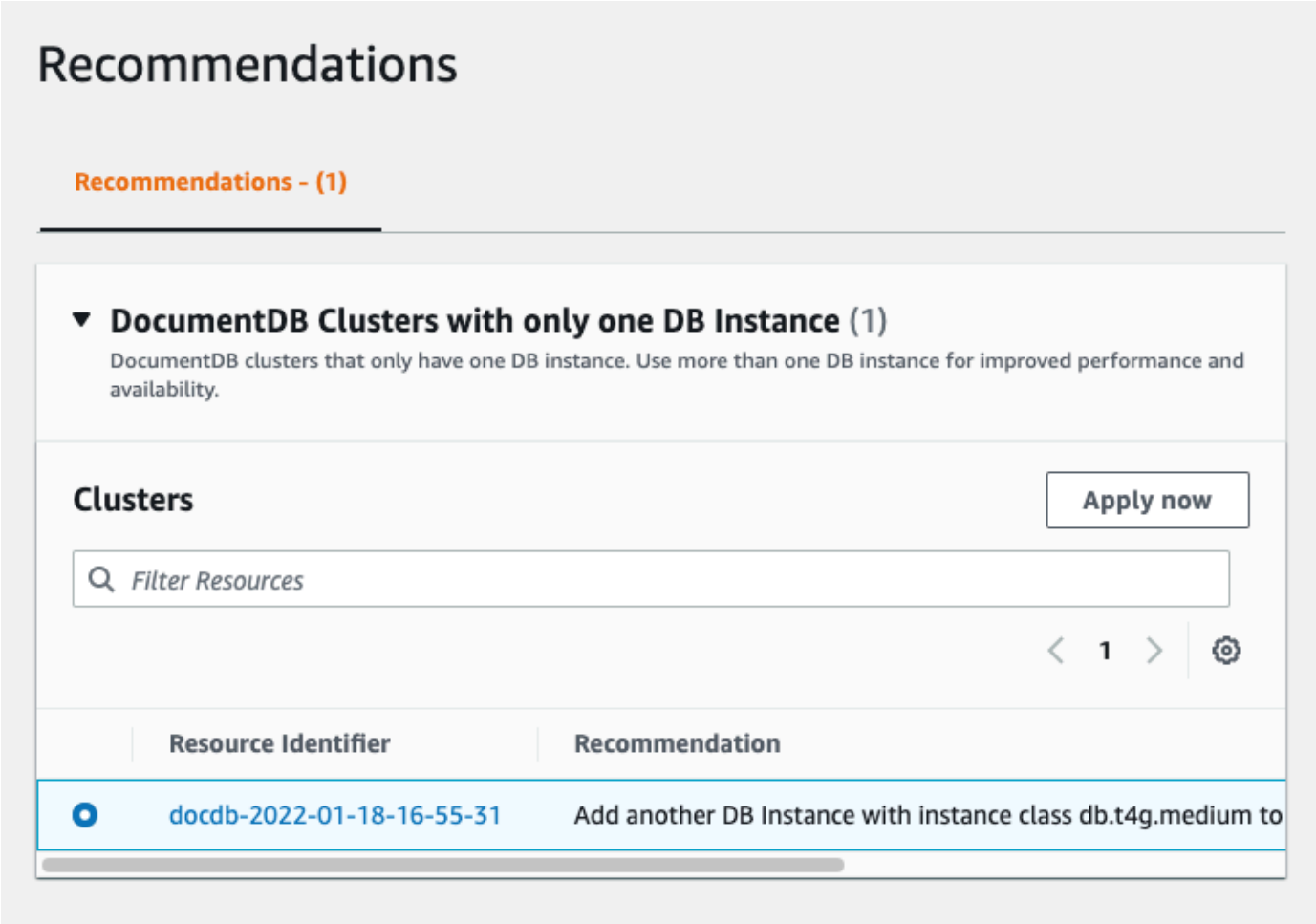
Untuk melihat dan mengambil tindakan atas rekomendasi Amazon DocumentDB

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Rekomendasi:



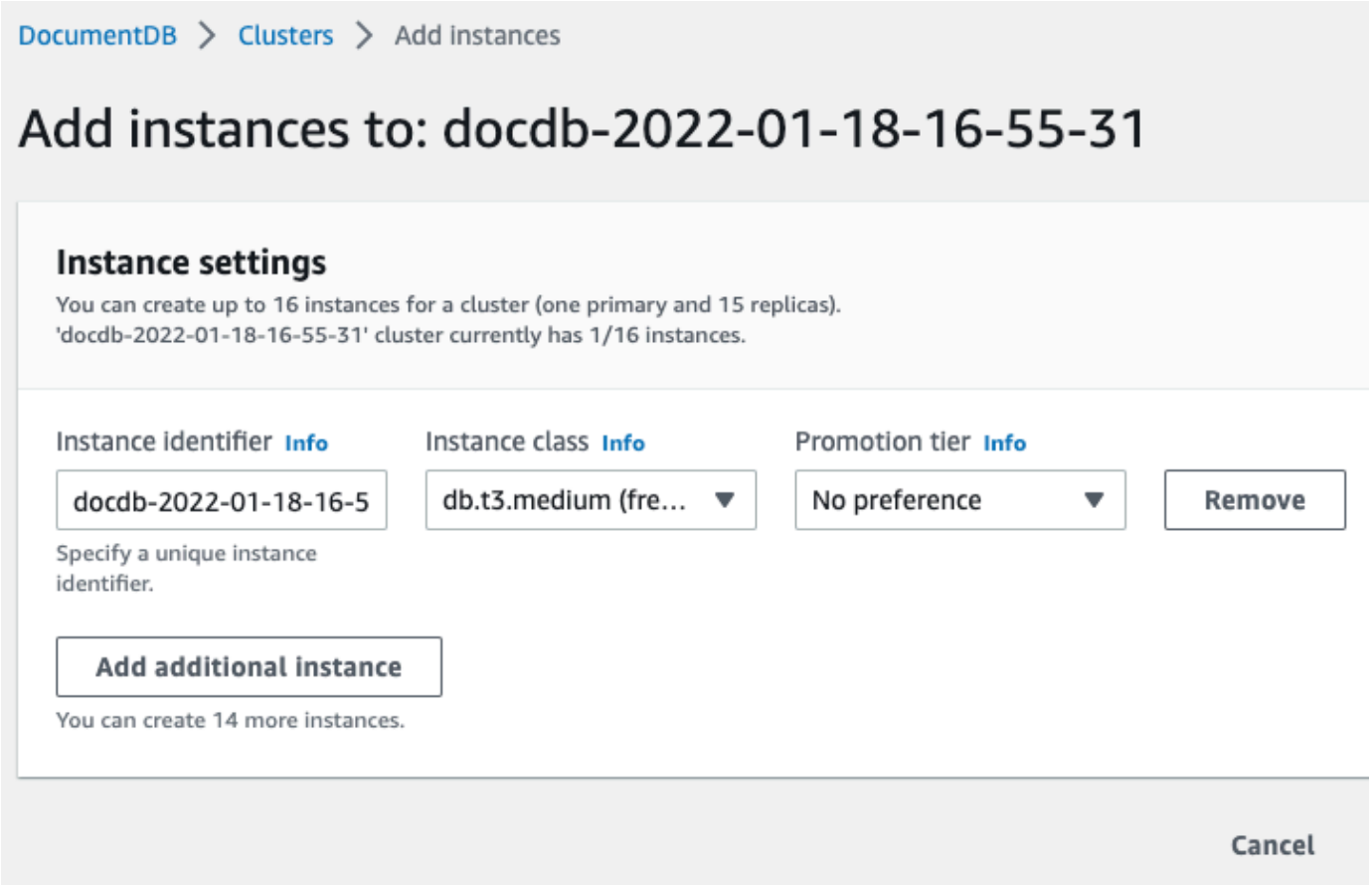
3. Dalam dialog Rekomendasi, perluas bagian yang menarik dan pilih tugas yang disarankan.

   Pada contoh di bawah ini, tugas yang direkomendasikan berlaku untuk klaster Amazon DocumentDB dengan hanya satu instans. Rekomendasi adalah untuk menambahkan contoh lain untuk meningkatkan kinerja dan ketersediaan.



4. Klik Terapkan sekarang.

   Untuk contoh ini, dialog Add instance muncul:



5. Ubah pengaturan instans baru Anda dan klik Buat.

Menggunakan Langganan Peristiwa Amazon DocumentDB

Amazon DocumentDB menggunakan Amazon Simple Notification Service (Amazon SNS) untuk memberikan pemberitahuan saat peristiwa Amazon DocumentDB terjadi. Pemberitahuan ini dapat berada dalam bentuk yang didukung oleh Amazon SNS untuk Wilayah AWS, seperti email, pesan teks, atau panggilan ke titik akhir HTTP.

Amazon DocumentDB mengelompokkan kejadian ini ke dalam kategori yang dapat Anda langgani sehingga Anda dapat menerima pemberitahuan saat suatu kejadian dalam kategori tersebut terjadi. Anda dapat berlangganan kategori peristiwa untuk instans, klaster, snapshot, snapshot klaster, atau untuk grup parameter. Misalnya, jika Anda berlangganan kategori Backup untuk instans tertentu, Anda akan diberi tahu setiap kali ada kejadian terkait pencadangan yang memengaruhi instans. Anda juga menerima notifikasi saat langganan peristiwa berubah.

Peristiwa terjadi di tingkat klaster dan instans, sehingga Anda dapat menerima peristiwa jika berlangganan ke klaster atau instans.

Langganan peristiwa dikirim ke alamat yang Anda berikan saat membuat langganan. Anda mungkin ingin membuat beberapa langganan berbeda, seperti satu langganan yang menerima semua notifikasi peristiwa dan langganan lain yang hanya mencakup peristiwa penting untuk instans produksi Anda. Anda dapat dengan mudah mematikan notifikasi tanpa menghapus langganan. Untuk melakukannya, atur tombol radio Diaktifkan ke pilihan Tidak di konsol Amazon DocumentDB.

> **Important**
>
> Amazon DocumentDB tidak menjamin urutan acara yang dikirim dalam alur kejadian. Urutan kejadian dapat berubah.

Amazon DocumentDB menggunakan Amazon Resource Name (ARN) dari topik Amazon SNS untuk mengidentifikasi setiap langganan. Konsol Amazon DocumentDB membuat ARN untuk Anda saat Anda membuat langganan.

Penagihan untuk langganan peristiwa Amazon DocumentDB dilakukan melalui Amazon SNS. Biaya Amazon SNS berlaku saat menggunakan pemberitahuan kejadian. Untuk informasi selengkapnya, lihat Harga Amazon Simple Notification Service. Selain biaya Amazon SNS, Amazon DocumentDB tidak menagih langganan peristiwa.

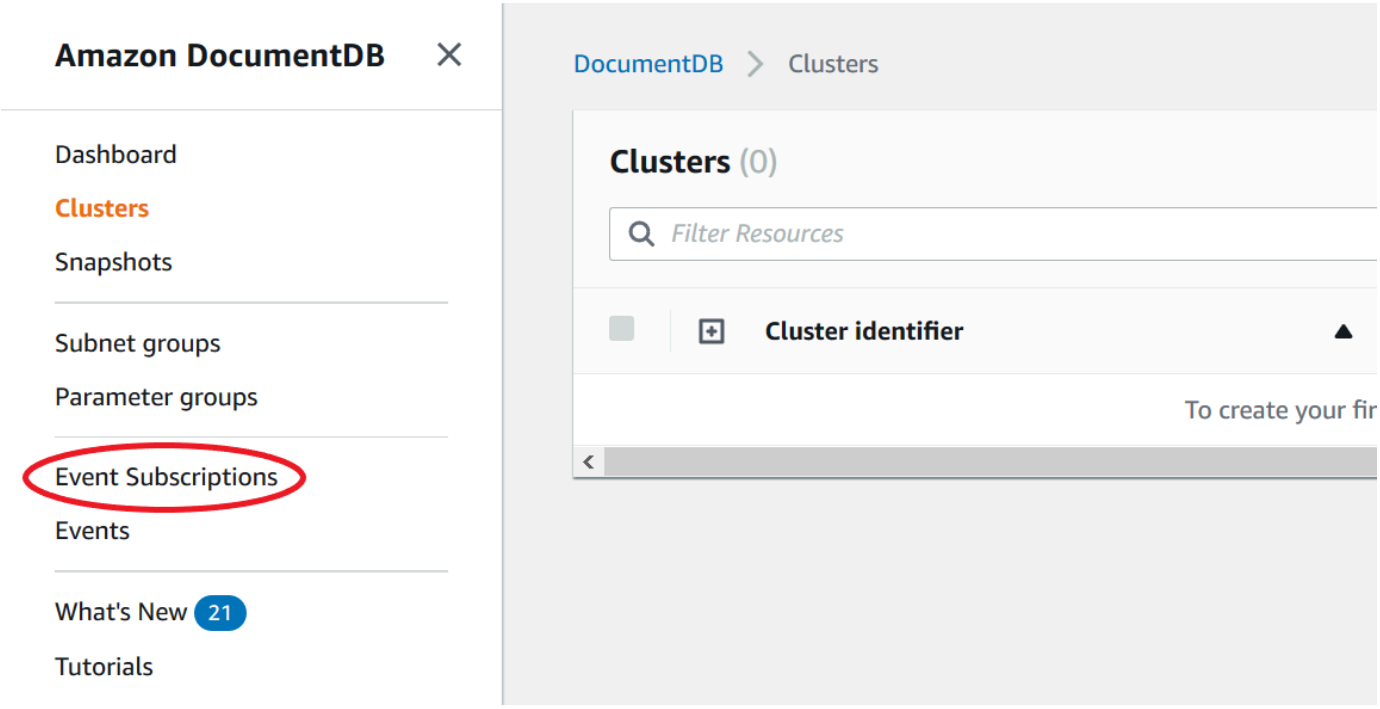
Topik

- Berlangganan Langganan Peristiwa Amazon DocumentDB
- Mengelola Langganan Notifikasi Peristiwa Amazon DocumentDB
- Kategori dan Pesan Kejadian Amazon DocumentDB

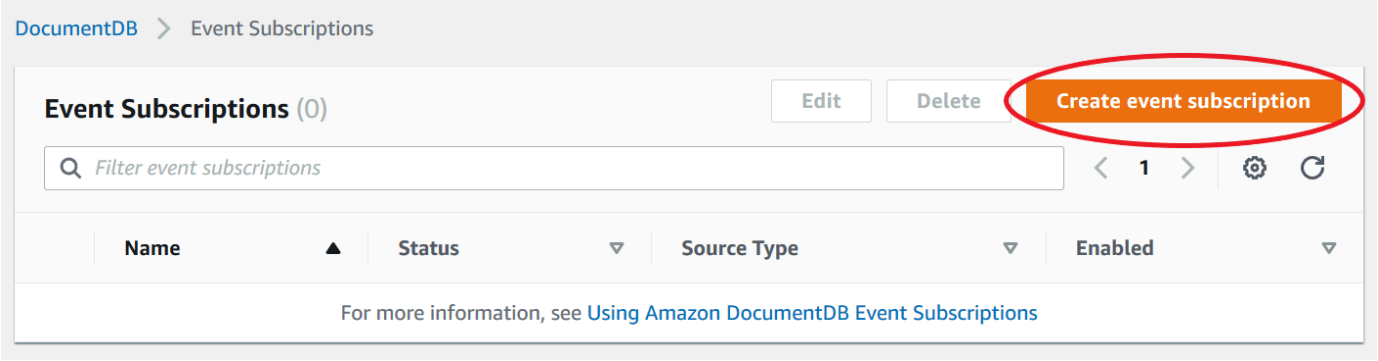
Berlangganan Langganan Peristiwa Amazon DocumentDB

Anda dapat menggunakan konsol Amazon DocumentDB untuk berlangganan langganan peristiwa, sebagai berikut:

1. Masuklah ke AWS Management Console di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Berlangganan peristiwa.

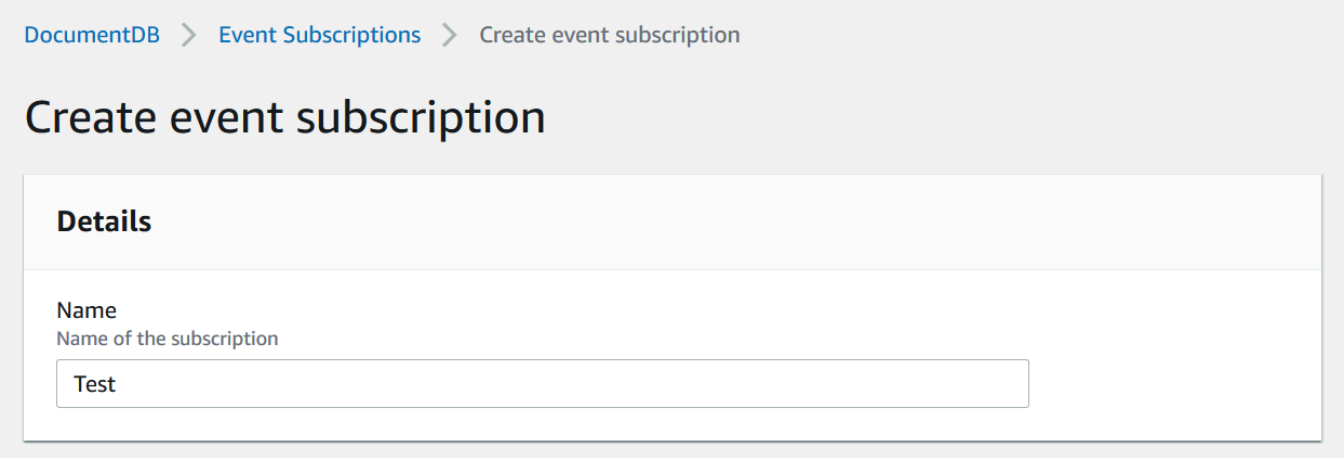


3. Di panel Berlangganan kejadian, pilih Buat langganan kejadian.

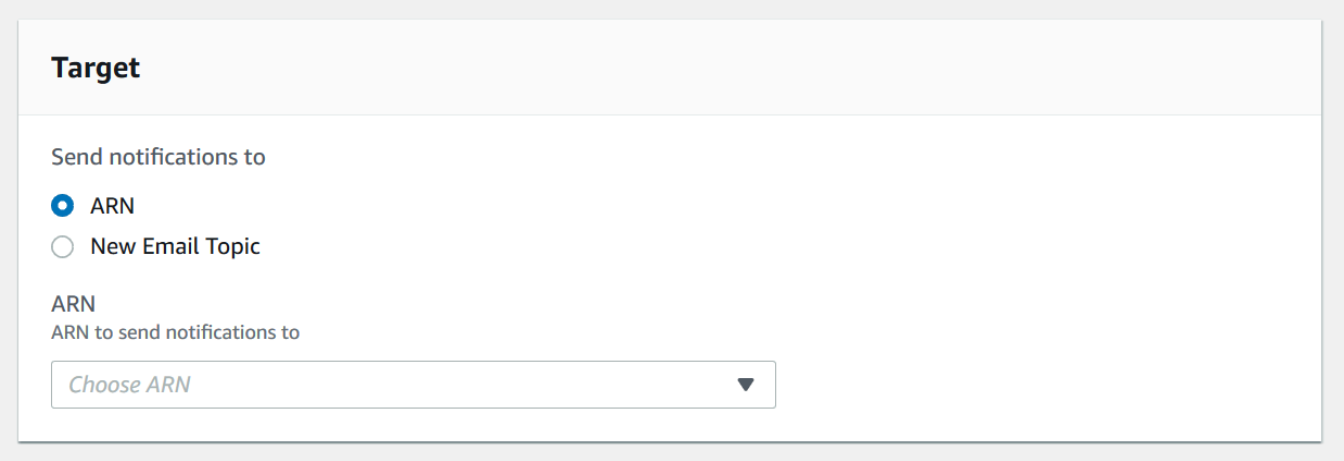


4. Di kotak dialog Buat langganan kejadian, lakukan hal berikut:

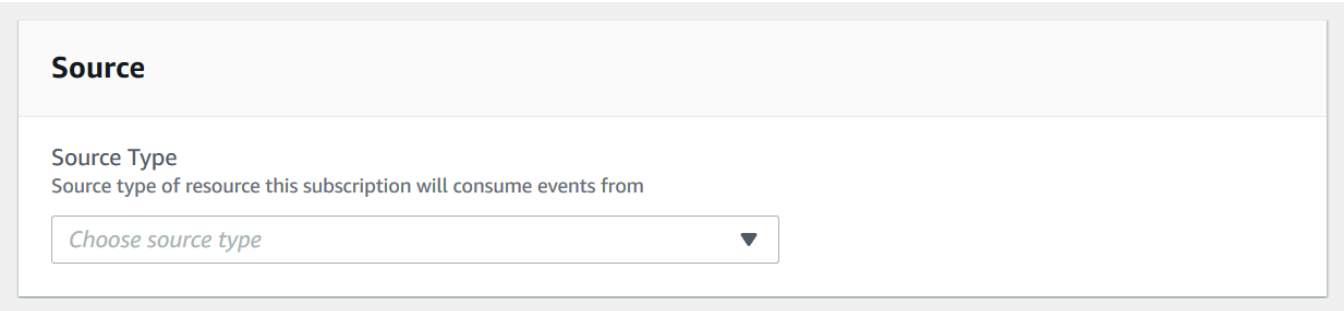
   - Untuk Nama, masukkan nama untuk langganan pemberitahuan kejadian.



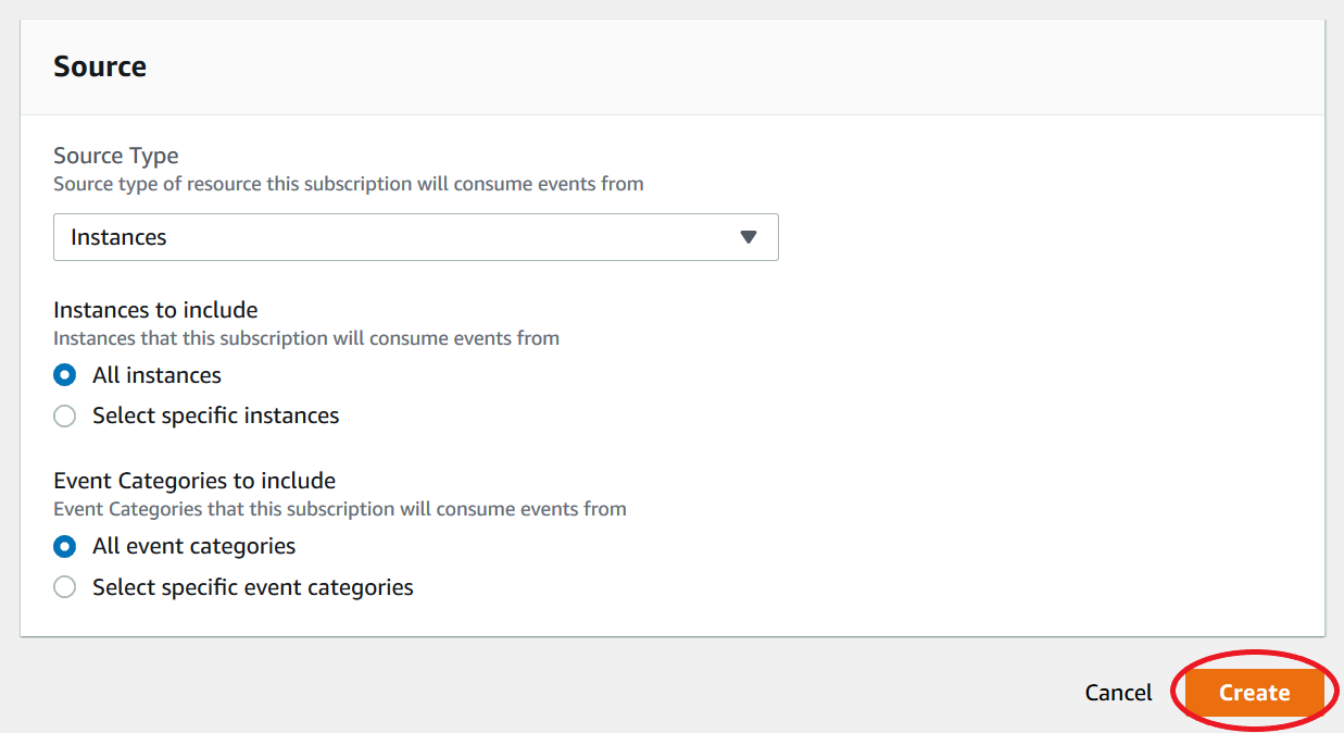
- Untuk Target, pilih target yang Anda tuju untuk mengirimkan notifikasi. Anda dapat memilih ARN yang ada atau memilih Topik Email Baru untuk memasukkan nama topik dan daftar penerima.



- Untuk Sumber, pilih jenis sumber. Tergantung pada jenis sumber yang Anda pilih, tentukan kategori kejadian dan sumber yang Anda inginkan untuk menjadi pangkal notifikasi kejadian.



- Pilih Create (Buat).

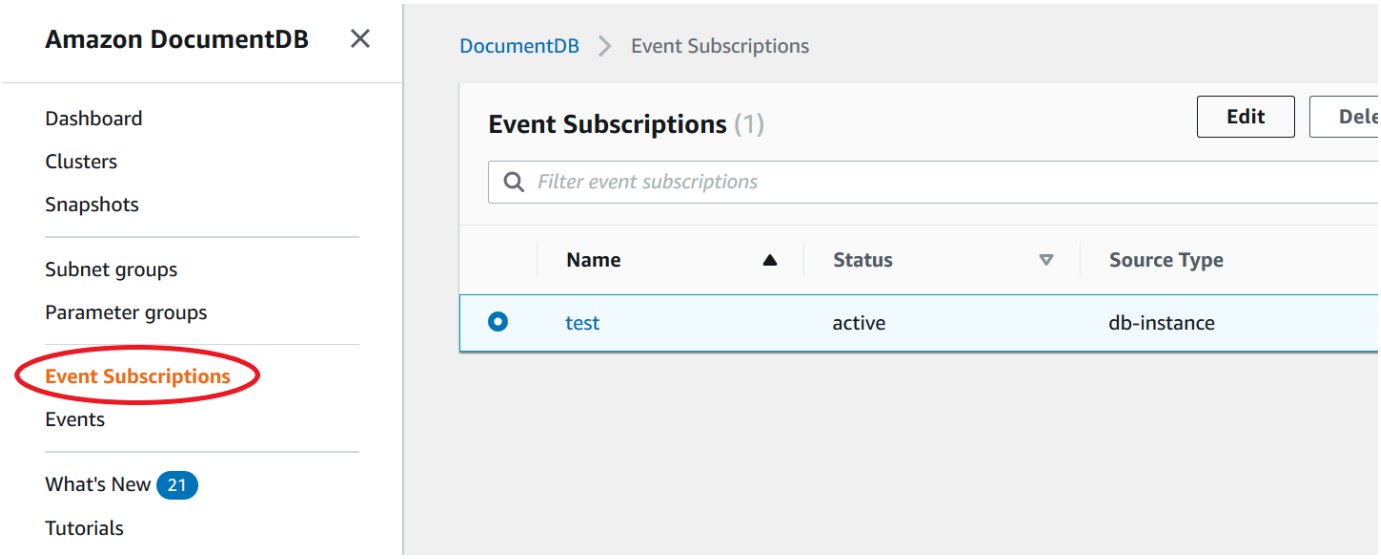


Mengelola Langganan Notifikasi Peristiwa Amazon DocumentDB

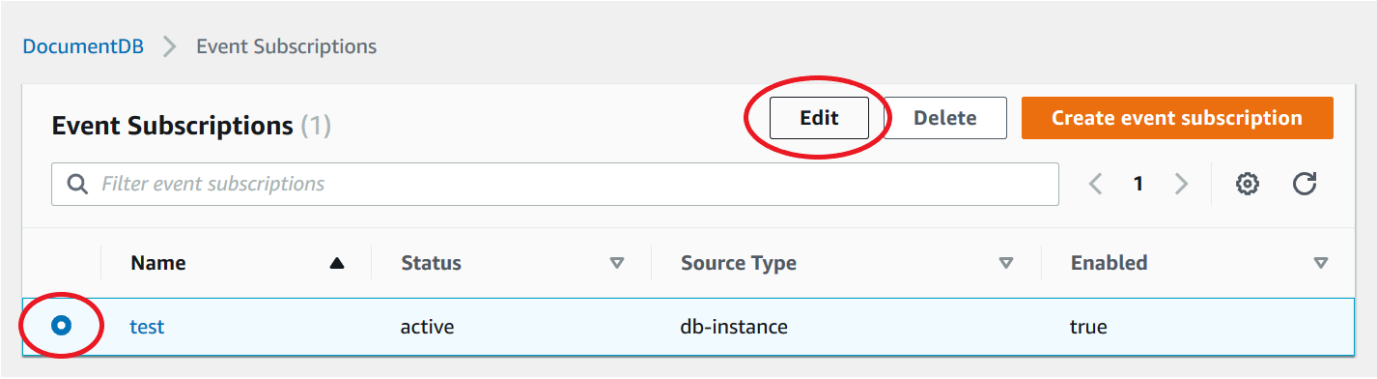
Jika Anda memilih Berlangganan acara di panel navigasi konsol Amazon DocumentDB, Anda dapat menampilkan kategori berlangganan dan daftar langganan Anda saat ini. Anda juga dapat mengubah atau menghapus langganan tertentu.

Untuk mengubah langganan notifikasi kejadian Amazon DocumentDB Anda saat ini

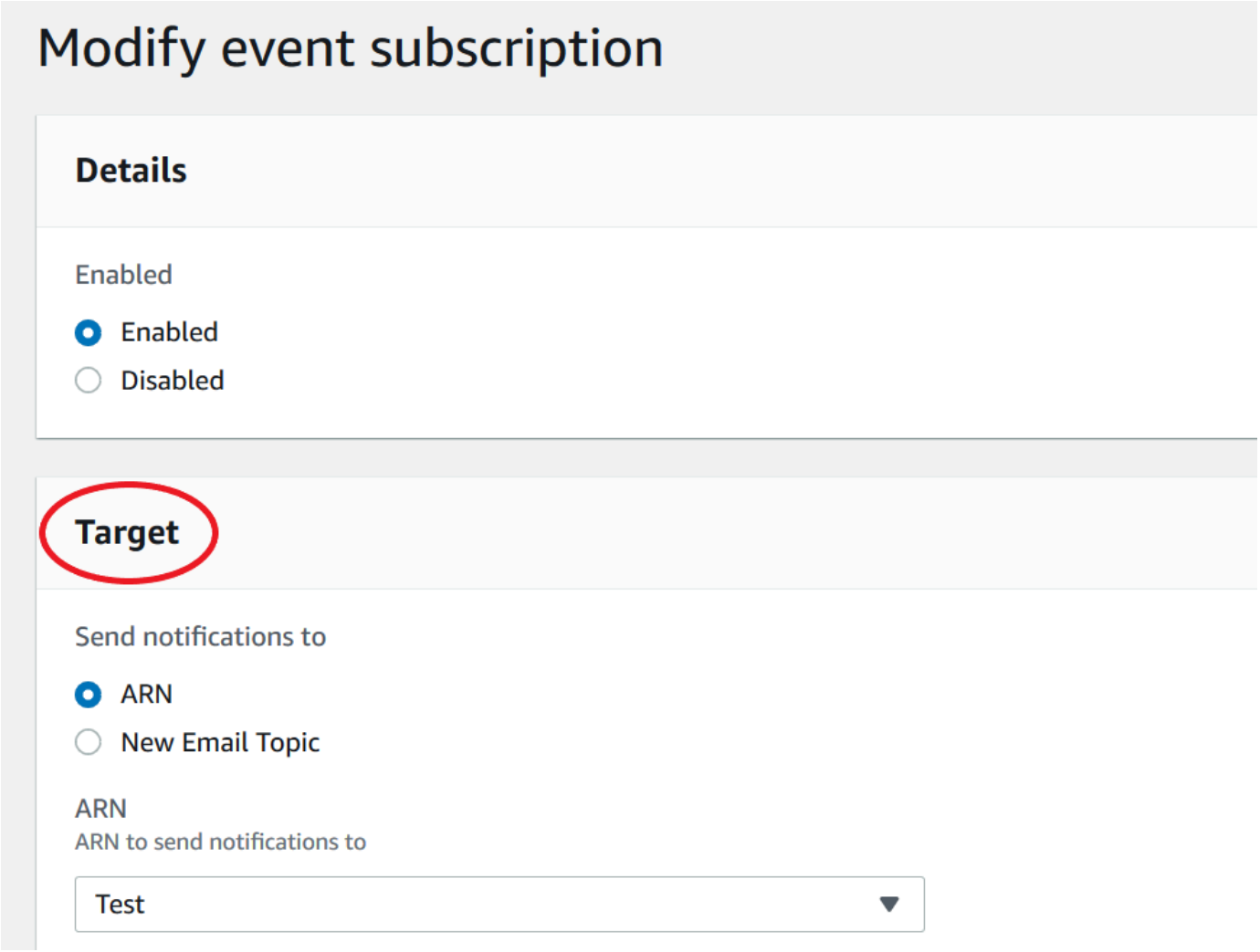
1. Masuklah ke AWS Management Console di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Berlangganan peristiwa. Panel Berlangganan kejadian menampilkan semua langganan pemberitahuan kejadian Anda.



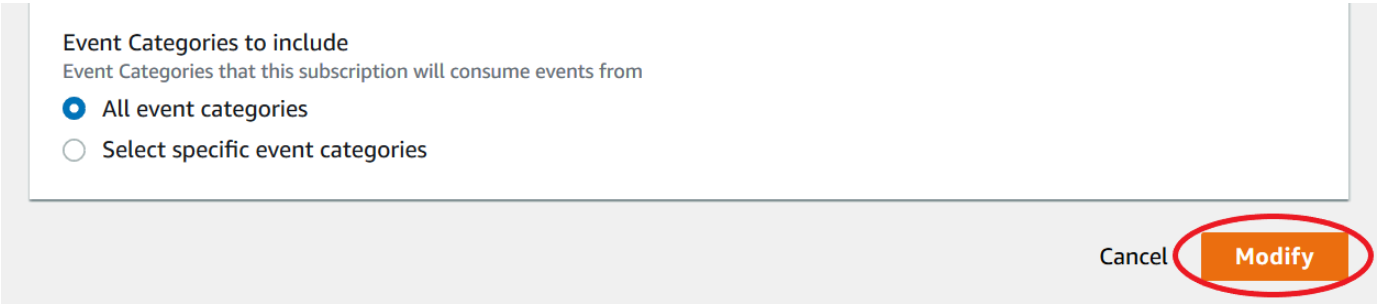
3. Di panel Berlangganan kejadian, pilih langganan yang ingin Anda ubah dan pilih Edit.



4. Buat perubahan pada langganan di bagian Target atau Sumber. Anda dapat menambahkan atau menghapus pengidentifikasi sumber dengan memilih atau membatalkan pilihan pada bagian Sumber.



5. Pilih Ubah. Konsol Amazon DocumentDB menunjukkan bahwa langganan sedang dimodifikasi.

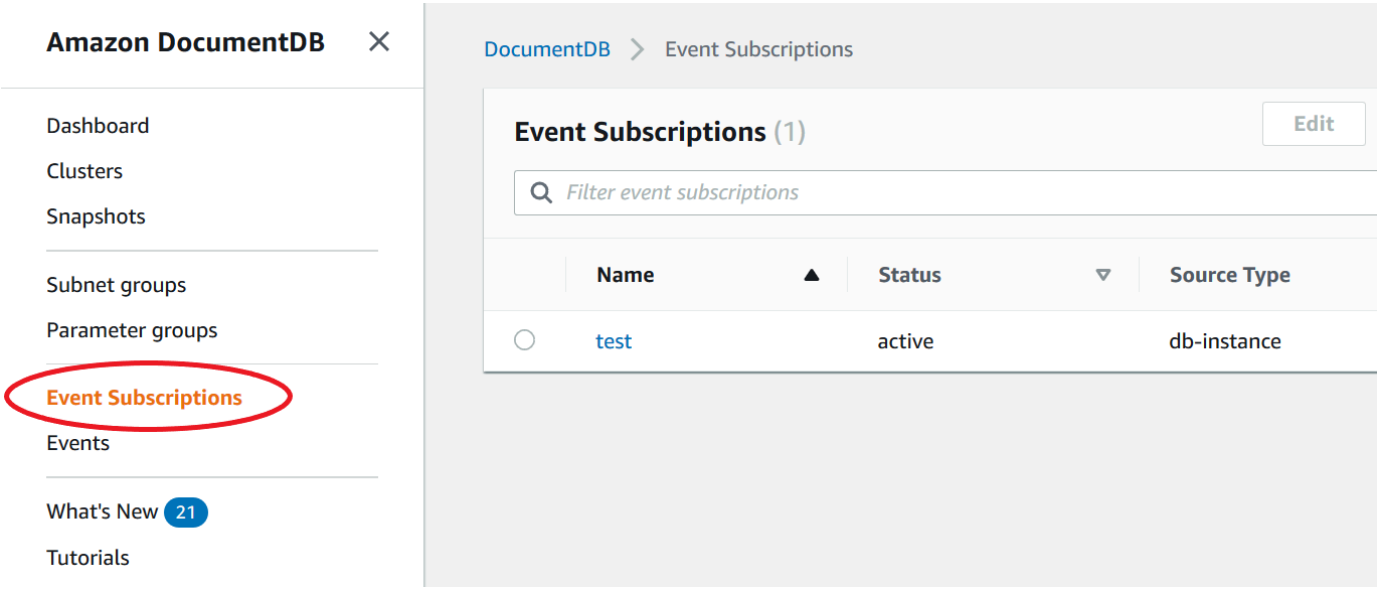


Menghapus langganan notifikasi kejadian Amazon DocumentDB

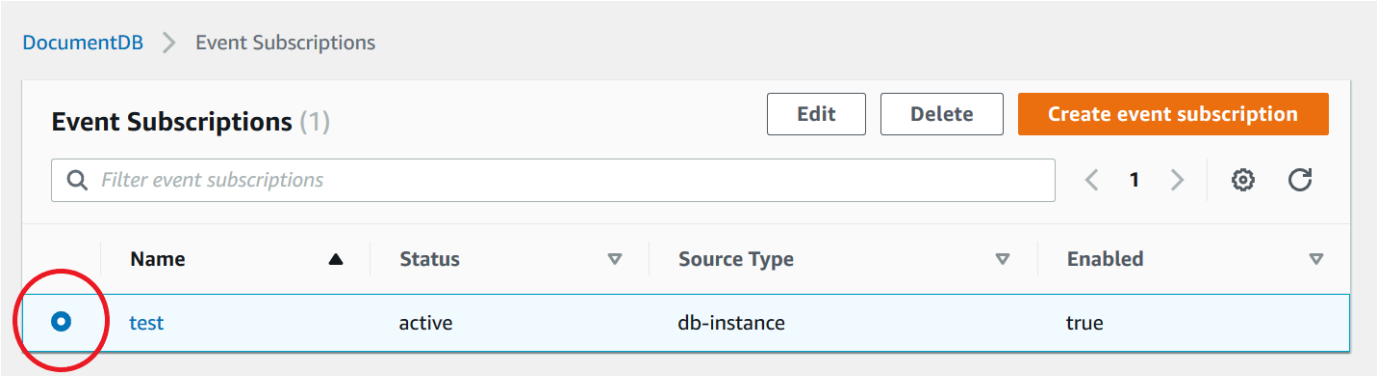
Anda dapat menghapus langganan saat tidak lagi membutuhkannya. Semua pelanggan topik tidak akan lagi menerima pemberitahuan kejadian yang ditentukan oleh langganan.

1. Masuklah ke AWS Management Console di https://console.aws.amazon.com/docdb.

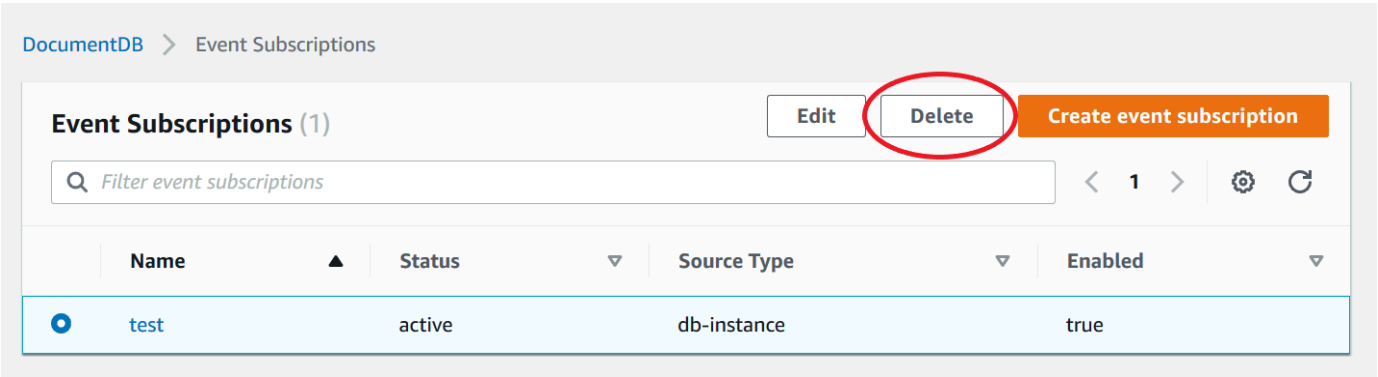
2. Di panel navigasi, pilih Berlangganan peristiwa.



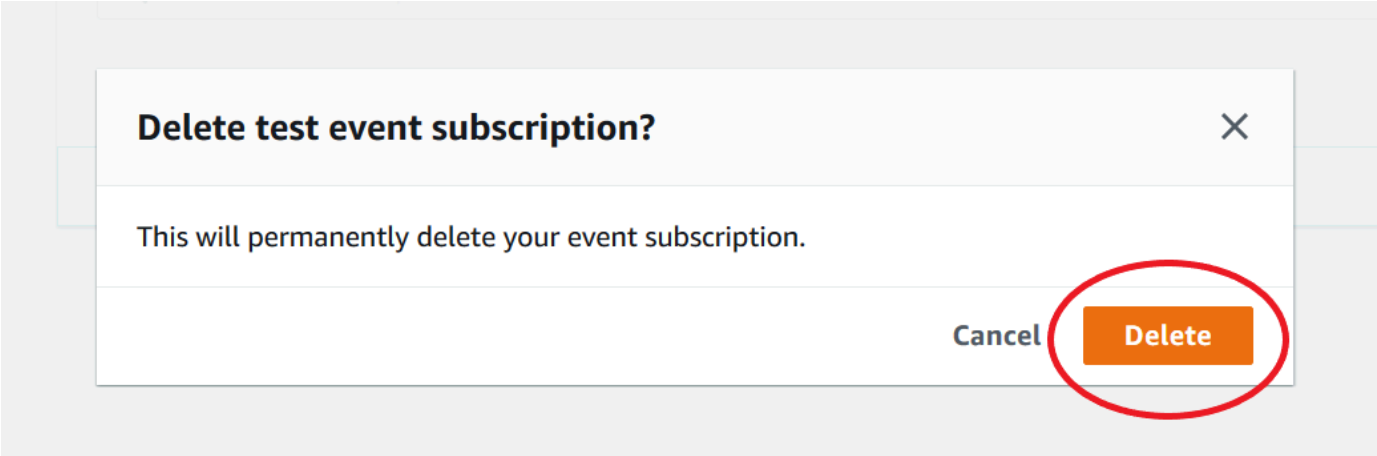
3. Di panel Langganan Kejadian, pilih langganan yang ingin Anda hapus.



4. Pilih Delete (Hapus).



5. Jendela pop-up akan muncul menanyakan apakah Anda ingin menghapus notifikasi ini secara permanen. Pilih Delete (Hapus).



Kategori dan Pesan Kejadian Amazon DocumentDB

Amazon DocumentDB menghasilkan sejumlah besar kejadian dalam kategori yang dapat Anda langgani menggunakan konsol. Setiap kategori berlaku untuk jenis sumber, yang dapat berupa instans, snapshot, atau grup parameter.

> **Note**
>
> Amazon DocumentDB menggunakan definisi dan ID kejadian Amazon RDS yang ada.

Kejadian Amazon DocumentDB yang berasal dari instans

Kategori	Deskripsi
ketersediaan	Instans dimulai ulang.
ketersediaan	Pematian instans DB.
perubahan konfigurasi	Menerapkan modifikasi pada kelas instans.
perubahan konfigurasi	Selesai menerapkan modifikasi pada kelas instans.
perubahan konfigurasi	Atur ulang kredensial utama.
pembuatan	Instans dibuat.

Kategori	Deskripsi
penghapusan	Instans dihapus
kegagalan	Instans gagal karena konfigurasi yang tidak kompatibel atau masalah penyimpanan yang mendasari. Mulailah point-in-time-restore untuk contoh.
notifikasi	Instans berhenti.
notifikasi	Instans dimulai.
notifikasi	Instans sedang dimulai karena melebihi waktu maksimum penghentian yang diizinkan.
pemulihan	Pemulihan instans telah dimulai. Waktu pemulihan akan beragam dengan jumlah data yang akan dipulihkan.
pemulihan	Pemulihan instans selesai.
keamanan patching	Pembaruan sistem operasi tersedia untuk instans Anda. Untuk informasi tentang menerapkan pembaruan, lihat Mempertah ankan Amazon DocumentDB.

Kejadian Amazon DocumentDB yang berasal dari klaster

Kategori	Deskripsi
pembuatan	Klaster dibuat
penghapusan	Klaster dihapus.
failover	Mempromosikan ulang primer sebelumnya.
failover	Failover ke instans selesai.

Kategori	Deskripsi
failover	Failover ke instans DB dimulai: %s
failover	Failover AZ yang sama ke instans DB dimulai: %s
failover	Failover AZ melintang ke instans DB dimulai: %s
pemeliharaan	Klaster telah diberi patch.
pemeliharaan	Klaster basis data berada dalam status yang tidak dapat ditingkatkan: %s
notifikasi	Klaster berhenti.
notifikasi	Klaster dimulai.
notifikasi	Perhentian klaster DB gagal.
notifikasi	klaster sedang dimulai karena melebihi waktu maksimal yang diperbolehkan untuk dihentikan.
notifikasi	Klaster berganti nama dari %s menjadi %s.

Kejadian Amazon DocumentDB yang berasal dari snapshot klaster

Tabel berikut menunjukkan kategori kejadian dan daftar kejadian saat snapshot klaster Amazon DocumentDB merupakan jenis sumber.

Kategori	Deskripsi
pencadangan	Membuat snapshot klaster manual.
pencadangan	Snapshot klaster manual dibuat.
pencadangan	Membuat snapshot klaster otomatis.
pencadangan	Snapshot klaster otomatis dibuat.

Kejadian Amazon DocumentDB yang berasal dari grup parameter

Tabel berikut menunjukkan kategori kejadian dan daftar kejadian saat grup parameter merupakan jenis sumber.

Kategori	Deskripsi
perubahan konfigurasi	Parameter diperbarui %s ke %s dengan metode aplikasi %s

Memantau Amazon DocumentDB dengan CloudWatch

Amazon DocumentDB (dengan kompatibilitas MongoDB) terintegrasi dengan Amazon CloudWatch sehingga Anda dapat mengumpulkan dan menganalisis metrik operasional untuk cluster Anda. Anda dapat memantau metrik ini menggunakan CloudWatch konsol, konsol Amazon DocumentDB,AWS Command Line Interface(AWS CLI), atau CloudWatchAPI.

CloudWatch juga memungkinkan Anda mengatur alarm sehingga Anda dapat diberi tahu jika nilai metrik melanggar ambang batas yang Anda tentukan. Anda bahkan dapat mengatur Amazon CloudWatch Peristiwa untuk mengambil tindakan korektif jika terjadi pelanggaran. Untuk informasi lebih lanjut tentang penggunaan CloudWatch dan alarm, lihatAmazon CloudWatch dokumentasi.

Topik

- Metrik Amazon DocumentDB
- Melihat CloudWatch Data
- Dimensi Amazon DocumentDB
- Pemantauan Opcounters
- Pemantauan Koneksi Basis Data

Metrik Amazon DocumentDB

Untuk memantau kesehatan dan kinerja klaster dan instans Amazon DocumentDB Anda, Anda dapat melihat metrik berikut di konsol Amazon DocumentDB.

> **Note**
>
> Metrik dalam tabel berikut berlaku untuk cluster berbasis instance dan elastis.

Pemanfaatan Sumber Daya

Metrik	Deskripsi
`BackupRetentionPeriodStorageUsed`	Jumlah total penyimpanan cadangan di GiB yang digunakan untuk mendukung point-in-time pengembalikan fitur dalam jendela retensi Amazon DocumentDB. Termasuk dalam total yang dilaporkan oleh metrik `TotalBackupStorageBilled` . Dihitung secara terpisah untuk setiap klaster Amazon DocumentDB.
`ChangeStreamLogSize`	Jumlah penyimpanan yang digunakan oleh klaster Anda untuk menyimpan log aliran perubahan dalam megabyte. Nilai ini adalah subset dari total penyimpanan untuk klaster (`VolumeBytesUsed`) dan memengaruhi biaya klaster. Untuk informasi harga penyimpanan, lihat Halaman produk Amazon DocumentDB. Ukuran log aliran perubahan adalah fungsi dari seberapa banyak perubahan yang terjadi di klaster Anda dan durasi

Metrik	Deskripsi
	retensi aliran perubahan yang lama. Untuk informasi lebih lanjut tentang aliran perubahan, lihat Menggunakan Aliran Perubahan dengan Amazon DocumentDB.
`CPUUtilization`	Persentase CPU yang digunakan oleh instans.
`DatabaseConnections`	Jumlah koneksi yang terbuka pada instans yang diambil pada frekuensi satu menit.
`DatabaseConnectionsMax`	Jumlah maksimum koneksi basis data terbuka pada instans dalam periode satu menit.
`DatabaseCursors`	Jumlah kursor yang terbuka pada instans yang diambil pada frekuensi satu menit.
`DatabaseCursorsMax`	Jumlah maksimum kursor terbuka pada instans dalam periode satu menit.
`DatabaseCursorsTimedOut`	Jumlah kursor yang habis waktunya dalam periode satu menit.
`FreeableMemory`	Jumlah memori akses acak yang tersedia, dalam byte.

Metrik	Deskripsi
FreeLocalStorage	Metrik ini melaporkan jumlah penyimpanan yang tersedia untuk setiap instans untuk tabel dan log sementara . Nilai ini tergantung pada kelas instans. Anda dapat meningkatkan jumlah ruang penyimpanan gratis untuk sebuah instans dengan memilih kelas instans yang lebih besar untuk instans Anda.
LowMemThrottleQueueDepth	Kedalaman antrean untuk permintaan yang dibatasi karena rendahnya memori yang tersedia yang diambil pada frekuensi satu menit.
LowMemThrottleMaxQueueDepth	Kedalaman antrean maksimum untuk permintaan yang dibatasi karena rendahnya memori yang tersedia dalam periode satu menit.
LowMemNumOperationsThrottled	Jumlah permintaan yang dibatasi karena rendahnya memori yang tersedia dalam periode satu menit.

Metrik	Deskripsi
SnapshotStorageUsed	Jumlah total penyimpanan backup dalam GiB yang digunakan oleh semua snapshot untuk klaster Amazon DocumentDB tertentu di luar jendela penyimpanan backup-nya. Termasuk dalam total yang dilaporkan oleh metrik TotalBackupStorageBilled. Dihitung secara terpisah untuk setiap klaster Amazon DocumentDB.
SwapUsage	Jumlah ruang tukar yang digunakan pada instans.
TotalBackupStorageBilled	Jumlah total penyimpanan backup dalam GiB yang akan ditagihkan untuk klaster Amazon DocumentDB tertentu. Termasuk penyimpanan backup yang diukur oleh metrik BackupRetentionPeriodStorageUsed dan SnapshotStorageUsed. Dihitung secara terpisah untuk setiap klaster Amazon DocumentDB.
TransactionsOpen	Jumlah transaksi yang terbuka pada instans yang diambil pada frekuensi satu menit.

Metrik	Deskripsi
`TransactionsOpenMax`	Jumlah maksimum transaksi terbuka pada instans dalam periode satu menit.
`VolumeBytesUsed`	Jumlah penyimpanan yang digunakan oleh klaster Anda dalam byte. Nilai ini memengaruhi biaya klaster. Untuk informasi harga, lihat Halaman produk Amazon DocumentDB.

Latensi

Metrik	Deskripsi
`DBClusterReplicaLagMaximum`	Jumlah maksimum lag, dalam milidetik, antara instans primer dan setiap instans Amazon DocumentDB dalam klaster.
`DBClusterReplicaLagMinimum`	Jumlah minimum lag, dalam milidetik, antara instans primer dan setiap instans replika dalam klaster.
`DBInstanceReplicaLag`	Jumlah lag, dalam milidetik, saat mereplikasi pembaruan dari instans primer ke instans replika.
`ReadLatency`	Jumlah waktu rata-rata yang diambil per operasi I/O disk.

Metrik	Deskripsi
WriteLatency	Jumlah waktu rata-rata, dalam milidetik, yang diambil per operasi I/O disk.

Operasi

Metrik	Deskripsi
DocumentsDeleted	Jumlah dokumen yang dihapus dalam periode satu menit.
DocumentsInserted	Jumlah dokumen yang dimasukkan dalam periode satu menit.
DocumentsReturned	Jumlah dokumen yang dikembalikan dalam periode satu menit.
DocumentsUpdated	Jumlah dokumen yang diperbarui dalam periode satu menit.
OpcountersCommand	Jumlah perintah yang dikeluarkan dalam periode satu menit.
OpcountersDelete	Jumlah operasi penghapus an yang dikeluarkan dalam periode satu menit.
OpcountersGetmore	Jumlah getmore yang dikeluarkan dalam periode satu menit.

Metrik	Deskripsi
OpcountersInsert	Jumlah operasi penyisipan yang dikeluarkan dalam periode satu menit.
OpcountersQuery	Jumlah kueri yang dikeluarkan dalam periode satu menit.
OpcountersUpdate	Jumlah operasi pembaruan yang dikeluarkan dalam periode satu menit.
TransactionsStarted	Jumlah transaksi yang dimulai pada instans dalam periode satu menit.
TransactionsCommitted	Jumlah transaksi yang dilakukan pada instans dalam periode satu menit.
TransactionsAborted	Jumlah transaksi yang dibatalkan pada instans dalam periode satu menit.
TTLDeletedDocuments	Jumlah dokumen yang dihapus oleh TTLMonitor dalam periode satu menit.

Throughput

Metrik	Deskripsi
NetworkReceiveThroughput	Jumlah throughput jaringan, dalam byte per detik, yang diterima dari klien oleh setiap instans dalam klaster. Throughput ini tidak mencakup

Metrik	Deskripsi
	lalu lintas jaringan antar instans dalam klaster dan volume klaster.
NetworkThroughput	Jumlah throughput jaringan, dalam byte per detik, yang diterima dari dan ditransmi sikan ke klien oleh setiap instans dalam klaster Amazon DocumentDB. Throughput ini tidak mencakup lalu lintas jaringan antar instans dalam klaster dan volume klaster.
NetworkTransmitThr oughput	Jumlah throughput jaringan, dalam byte per detik, yang dikirim ke klien oleh setiap instans dalam klaster. Throughput ini tidak mencakup lalu lintas jaringan antar instans dalam klaster dan volume klaster.
ReadIOPS	Jumlah rata-rata operasi I/O baca disk per detik. Laporan Amazon DocumentD B membaca dan menulis IOPS secara terpisah, dan dalam interval satu menit.
ReadThroughput	Jumlah byte rata-rata yang dibaca dari disk per detik.

Metrik	Deskripsi
VolumeReadIOPs	Jumlah rata-rata operasi I/O baca yang ditagih dari volume klaster, dilaporkan pada interval 5 menit. Operasi baca yang ditagih dihitung pada tingkat volume klaster, diagregasikan dari semua instans dalam klaster, dan kemudian dilaporkan pada interval 5 menit. Nilai dihitung dengan mengambil nilai dari metrik operasi baca selama periode 5 menit. Anda dapat menentukan jumlah operasi pembacaan yang ditagih per detik dengan mengambil nilai metrik operasi baca yang ditagih dan membaginya dengan 300 detik. Misalnya, jika VolumeReadIOPs menghasilkan 13.686, kemudian operasi baca yang ditagih per detik adalah 45 (13.686 / 300 = 45,62). Anda mengakumulasi operasi baca tertagih untuk kueri yang meminta halaman basis data yang tidak ada dalam cache buffer dan oleh karena itu harus dimuat dari penyimpanan. Anda mungkin melihat lonjakan dalam operasi baca yang ditagih karena hasil kueri

Metrik	Deskripsi	
	dibaca dari penyimpanan, lalu dimuat ke dalam cache buffer.	

Metrik	Deskripsi
VolumeWriteIOPs	Jumlah rata-rata operasi I/O tulis yang ditagih dari volume klaster, dilaporkan pada interval 5 menit. Operasi tulis yang ditagih dihitung pada tingkat volume klaster, diagregasikan dari semua instans dalam klaster, dan kemudian dilaporkan pada interval 5 menit. Nilai dihitung dengan mengambil nilai dari metrik operasi tulis selama periode 5 menit. Anda dapat menentukan jumlah operasi tulis yang ditagih per detik dengan mengambil nilai metrik operasi tulis yang ditagih dan membaginya dengan 300 detik. Misalnya, jika VolumeWriteIOPs menghasilkan 13.686, kemudian operasi tulis yang ditagih per detik adalah 45 (13.686 / 300 = 45,62). Perhatikan bahwa metrik VolumeReadIOPs dan VolumeWriteIOPs dihitung oleh lapisan penyimpanan DocumentDB dan itu termasuk IO yang dilakukan oleh instans primer dan replika. Data dikumpulkan setiap 20-30 menit dan kemudian dilaporka

Metrik	Deskripsi
	n pada interval 5 menit, sehingga memancarkan titik data yang sama untuk metrik dalam periode waktu. Jika Anda mencari metrik untuk dikorelasikan dengan operasi penyisipan selama interval 1 menit, Anda dapat menggunakan metrik WriteIOPS tingkat instans. Metrik ini tersedia di tab pemantauan instans primer Amazon DocumentDB Anda.
`WriteIOPS`	Jumlah rata-rata operasi I/O tulis disk per detik. Ketika digunakan pada tingkat cluster,`WriteIOPs` dievaluasi di semua instance di cluster. IOPS baca dan tulis dilaporkan secara terpisah, dengan interval 1 menit.
`WriteThroughput`	Jumlah byte rata-rata byte yang ditulis dari disk per detik.

Sistem

Metrik	Deskripsi
`BufferCacheHitRatio`	Persentase permintaan yang dilayani oleh cache buffer.

Metrik	Deskripsi
DiskQueueDepth	jumlah permintaan tulis bersamaan ke volume penyimpanan terdistribusi.
EngineUptime	Jumlah waktu, dalam detik, instans telah berjalan.
IndexBufferCacheHitRatio	Persentase permintaan indeks yang dilayani oleh cache buffer. Anda mungkin melihat lonjakan lebih besar dari 100% untuk metrik tepat setelah Anda menghapus indeks, koleksi, atau basis data. Ini akan secara otomatis dikoreksi setelah 60 detik. Batasan ini akan diperbaiki dalam pembaruan patch di masa mendatang.

Metrik Instans T3

Metrik	Deskripsi
CPUCreditUsage	Jumlah kredit CPU yang digunakan selama periode pengukuran.
CPUCreditBalance	Jumlah kredit CPU yang diperoleh instans. Saldo ini habis saat CPU meledak dan kredit CPU dihabiskan lebih cepat daripada yang diperoleh.

Metrik	Deskripsi
`CPUSurplusCreditBalance`	Jumlah kredit CPU surplus yang dihabiskan untuk mempertahankan kinerja CPU saat CPUCreditBalance nilainya adalah nol.
`CPUSurplusCreditsCharged`	Jumlah kredit CPU surplus yang melebihi jumlah kredit CPU maksimum yang dapat diperoleh dalam periode 24 jam, dan dengan demikian menarik biaya tambahan. Untuk informasi selengkapnya, lihat Memantau kredit CPU Anda.

Melihat CloudWatch Data

Anda dapat melihat Amazon CloudWatch data menggunakan CloudWatch konsol, konsol Amazon DocumentDB,AWS Command Line Interface(AWS CLI), atau CloudWatch API.

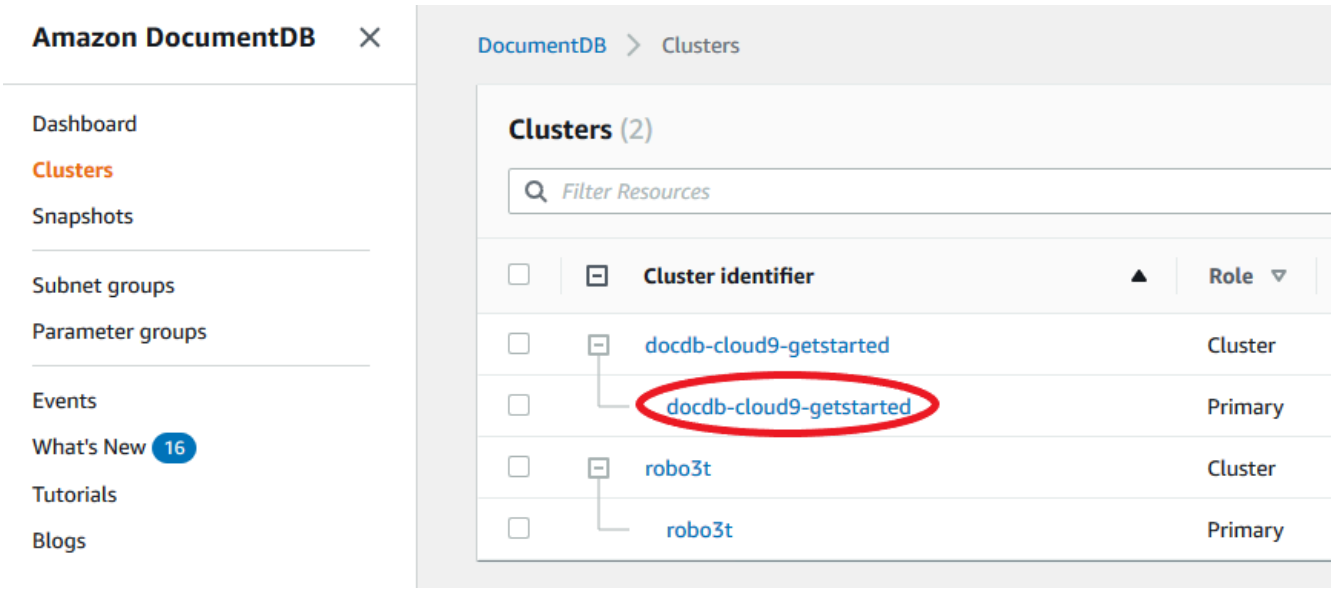
Using the AWS Management Console

Untuk melihat CloudWatch metrik menggunakan Amazon DocumentDB Management Console, selesaikan langkah-langkah berikut.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster.

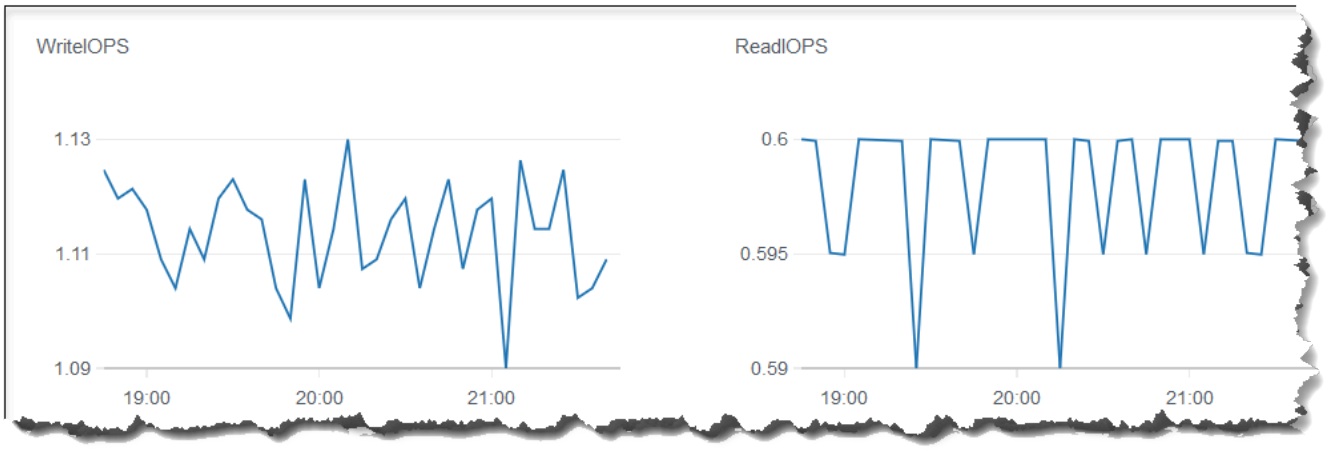
> **Tip**
>
> Jika Anda tidak melihat panel navigasi pada sisi kiri layar Anda, pilih ikon menu (≡) di sudut kiri atas halaman.

3. Di kotak navigasi Klaster, Anda akan melihat kolom Pengidentifikasi klaster. Instans Anda tercantum di bawah klaster, mirip dengan tangkapan layar di bawah.



4. Dari daftar instans, pilih nama instans yang Anda inginkan metriknya.
5. Di halaman ringkasan instans yang dihasilkan, pilih tab Pemantauan untuk melihat representasi grafis dari metrik instans Amazon DocumentDB Anda. Karena grafik harus dibuat untuk setiap metrik, mungkin diperlukan beberapa menit untukCloudWatchgrafik untuk diisi.

   Gambar berikut menunjukkan representasi grafis dari dua CloudWatch metrik di konsol Amazon DocumentDB,`WriteIOPS`dan`ReadIOPS`.



Using the CloudWatch Management Console

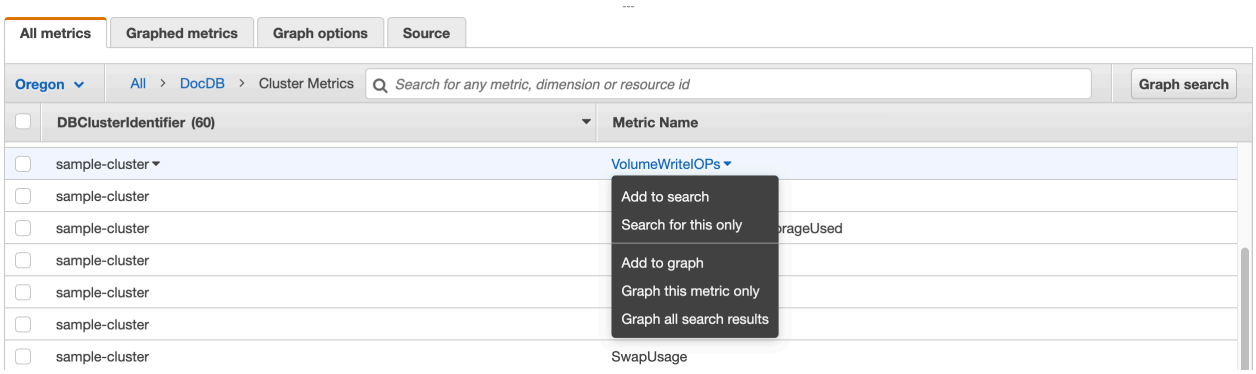
Untuk melihat CloudWatch metrik menggunakan CloudWatch Management Console, selesaikan langkah-langkah berikut.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/cloudwatch.
2. Di panel navigasi, pilih Metrik. Kemudian, dari daftar nama layanan, pilih DocDB.
3. Pilih dimensi metrik (misalnya, Metrik Klaster).
4. Tab Semua metrik menampilkan semua metrik untuk dimensi tersebut di DocDB.

   a. Untuk menyortir tabel, gunakan judul kolom.

   b. Untuk membuat grafik metrik, pilih kotak centang di samping metrik. Untuk memilih semua metrik, pilih kotak centang di baris judul tabel.

   c. Untuk memfilter menurut metrik, arahkan kursor ke nama metrik dan pilih panah tarik-turun di samping nama metrik. Lalu, pilih Tambahkan ke pencarian, seperti yang ditunjukkan pada gambar di bawah ini.



Using the AWS CLI

Untuk melihat CloudWatch data untuk Amazon DocumentDB, gunakan CloudWatch `get-metric-statistics`operasi dengan parameter berikut.

Parameter

- **`--namespace`** — Diperlukan. Namespace layanan yang Anda inginkan CloudWatch metrik. Untuk Amazon DocumentDB, ini harus `AWS/DocDB`.
- **`--metric-name`** — Diperlukan. Nama metrik yang Anda inginkan datanya.

- **`--start-time`** — Diperlukan. Stempel waktu yang menentukan titik data pertama untuk dikembalikan.

  Nilai yang ditentukan bersifat inklusif; hasil termasuk titik data dengan stempel waktu yang ditentukan. Stempel waktu harus dalam format ISO 8601 UTC (misalnya, 2016-10-03T23:00:00Z).

- **`--end-time`** — Diperlukan. Stempel waktu yang menentukan titik data terakhir untuk dikembalikan.

  Nilai yang ditentukan bersifat inklusif; hasil termasuk titik data dengan stempel waktu yang ditentukan. Stempel waktu harus dalam format ISO 8601 UTC (misalnya, 2016-10-03T23:00:00Z).

- **`--period`** — Diperlukan. Perincian, dalam detik, dari titik data yang dikembalikan. Untuk metrik dengan resolusi reguler, periode dapat sesingkat satu menit (60 detik) dan harus dalam kelipatan 60. Untuk metrik resolusi tinggi yang dikumpulkan dengan interval kurang dari satu menit, periode dapat selama 1, 5, 10, 30, 60, atau kelipatan 60.

- **`--dimensions`** — Opsional. Jika metrik berisi beberapa dimensi, Anda harus menyertakan nilai untuk setiap dimensi. CloudWatch memperlakukan setiap kombinasi dimensi yang unik sebagai metrik terpisah. Jika kombinasi dimensi tertentu tidak dipublikasikan, Anda tidak dapat mengambil statistiknya. Anda harus menentukan dimensi yang sama yang digunakan saat metrik dibuat.

- **`--statistics`** — Opsional. Statistik metrik, selain persentil. Untuk statistik persentil, gunakan `ExtendedStatistics`. Ketika memanggil `GetMetricStatistics`, Anda harus menentukan salah satu `Statistics` atau `ExtendedStatistics`, tapi tidak keduanya.

  Nilai yang diizinkan:

  - `SampleCount`
  - `Average`
  - `Sum`
  - `Minimum`
  - `Maximum`

- **`--extended-statistics`** — Opsional. Statistik `percentile`. Tentukan nilai antara p0.0 dan p100. Ketika memanggil `GetMetricStatistics`, Anda harus menentukan salah satu `Statistics` atau `ExtendedStatistics`, tapi tidak keduanya.

- **`--unit`** — Opsional. Unit untuk metrik tertentu. Metrik dapat dilaporkan dalam beberapa unit. Tidak menyediakan sebuah unit mengakibatkan semua unit dikembalikan. Jika Anda hanya menentukan unit yang tidak dilaporkan oleh metrik, hasil panggilan adalah null.

  Kemungkinan nilai:

  - `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

Contoh berikut menemukan `CPUUtilization` maksimum untuk periode 2 jam dengan mengambil sampel setiap 60 detik.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Output dari operasi ini terlihat seperti berikut.

```
{
       "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"
            }
        ]
    }
```

Dimensi Amazon DocumentDB

Metrik untuk Amazon DocumentDB dikualifikasikan berdasarkan nilai untuk akun atau operasi. Anda dapat menggunakan CloudWatch konsol untuk mengambil data Amazon DocumentDB yang difilter oleh salah satu dimensi dalam tabel berikut.

Dimensi	Deskripsi
`DBClusterIdentifier`	Memfilter data yang Anda minta untuk klaster Amazon DocumentDB tertentu.
`DBClusterIdentifier, Role`	Memfilter data yang Anda minta untuk klaster Amazon DocumentDB tertentu, menggabun gkan metrik menurut peran instans (PENULIS/ PEMBACA). Misalnya, Anda dapat menggabun gkan metrik untuk semua instans PEMBACA yang termasuk dalam suatu klaster.
`DBInstanceIdentifier`	Memfilter data yang Anda minta untuk instans basis data tertentu.

Pemantauan Opcounters

Metrik opcounter memiliki nilai bukan nol (biasanya ~50) untuk klaster siaga. Hal ini karena Amazon DocumentDB melakukan pemeriksaan kesehatan berkala, operasi internal, dan tugas pengumpulan metrik.

Pemantauan Koneksi Basis Data

Ketika Anda melihat jumlah koneksi dengan menggunakan perintah mesin database seperti`db.runCommand( { serverStatus: 1 })`, Anda mungkin melihat hingga 10 koneksi lebih banyak daripada yang Anda lihat di`DatabaseConnections`lewat CloudWatch. Hal ini terjadi karena Amazon DocumentDB melakukan pemeriksaan kesehatan berkala dan tugas pengumpulan metrik yang tidak diperhitungkan dalam `DatabaseConnections`. `DatabaseConnections` mewakili koneksi yang diprakarsai pelanggan saja.

Mencatat Panggilan API Amazon DocumentDB dengan AWS CloudTrail

Amazon DocumentDB (dengan kompatibilitas MongoDB) terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWS layanan di Amazon DocumentDB (dengan kompatibilitas MongoDB). CloudTrail menangkap semua panggilanAWS CLI API untuk Amazon DocumentDB sebagai peristiwa, termasuk panggilan dari konsol Amazon DocumentDB dan dari panggilan kode ke Amazon DocumentDB SDK. Jika membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa berkelanjutan ke bucket Amazon S3, termasuk peristiwa untuk Amazon DocumentDB. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat kejadian terbaru di CloudTrail konsol di Riwayat peristiwa. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Amazon DocumentDB (dengan kompatibilitas MongoDB), alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan rincian lainnya.

> **Important**
>
> Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS). Konsol Amazon DocumentDB, AWS CLI, dan panggilan API dicatat sebagai panggilan yang dibuat ke API Amazon RDS.

Untuk mempelajari selengkapnya tentang AWS CloudTrail, lihat Panduan Pengguna AWS CloudTrail.

Informasi Amazon DocumentDB di CloudTrail

CloudTrail diaktifkan pada AndaAkun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Amazon DocumentDB (dengan kompatibilitas MongoDB), aktivitas tersebut dicatat di CloudTrail peristiwa bersama peristiwaAWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihat Melihat Kejadian dengan Riwayat CloudTrail Kejadian.

Untuk catatan peristiwa yang sedang berlangsung di Akun AWS Anda, termasuk peristiwa untuk Amazon DocumentDB (dengan kompatibilitas MongoDB), buatlah jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasiAWS layanan lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi lebih lanjut, lihat topik berikut di Panduan Pengguna AWS CloudTrail:

- Ikhtisar untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima Berkas CloudTrail Log dari Beberapa Wilayah
- Menerima Berkas CloudTrail Log dari Beberapa Akun

Setiap entri kejadian atau log menyertakan informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau .
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail.

Membuat Profil Operasi Amazon DocumentDB

Anda dapat menggunakan profiler di Amazon DocumentDB (dengan kompatibilitas MongoDB) untuk mencatat waktu eksekusi dan detail operasi yang dilakukan di klaster Anda. Profiler berguna untuk memantau operasi paling lambat di klaster Anda untuk membantu Anda meningkatkan performa kueri individual dan performa klaster secara keseluruhan.

Secara default, fitur profiler dinonaktifkan. Saat diaktifkan, profiler mencatat operasi yang memakan waktu lebih lama dari nilai ambang batas yang ditentukan pelanggan (misalnya, 100 ms) ke Amazon CloudWatch Log. Detail yang dicatat termasuk perintah yang diprofilkan, waktu, ringkasan rencana, dan metadata klien. Setelah operasi dicatat CloudWatch Log, Anda dapat menggunakan CloudWatch Log Wawasan untuk menganalisis, memantau, dan mengarsipkan data profil Amazon DocumentDB Anda. Kueri umum disediakan di bagian Kueri umum.

Saat diaktifkan, profiler menggunakan sumber daya tambahan di klaster Anda. Kami merekomendasikan Anda memulai dengan nilai ambang batas yang tinggi (misalnya, 500 milidetik) dan secara bertahap menurunkan nilai tersebut untuk mengidentifikasi operasi yang lambat. Memulai dengan nilai ambang 50 milidetik dapat menyebabkan masalah performa pada klaster Anda untuk aplikasi dengan throughput tinggi. Profiler diaktifkan di tingkat klaster dan berfungsi pada semua instans dan basis data dalam klaster. Amazon DocumentDB mencatat operasi ke Amazon CloudWatch Log atas dasar upaya terbaik.

Meskipun Amazon DocumentDB tidak mengenakan biaya tambahan untuk mengaktifkan profiler, Anda dikenakan tarif standar untuk penggunaan CloudWatch Log. Untuk informasi tentang CloudWatch Harga log, lihatAmazon CloudWatch harga.

Topik

- Operasi yang Didukung
- Keterbatasan:
- Mengaktifkan Amazon DocumentDB Profiler
- Menonaktifkan Amazon DocumentDB Profiler
- Menonaktifkan Ekspor Log Profiler
- Mengakses Log Amazon DocumentDB Profiler Anda
- Kueri umum

Operasi yang Didukung

Profiler Amazon DocumentDB mendukung operasi berikut:

- `aggregate`
- `count`
- `delete`
- `distinct`
- `find` (OP_QUERY dan perintah)
- `findAndModify`
- `insert`
- `update`

Keterbatasan:

Profiler kueri lambat hanya dapat memancarkan log profiler jika seluruh set hasil kueri dapat ditampung dalam satu batch, dan jika set hasil di bawah 16 MB (ukuran BSON maksimum). Set hasil yang lebih besar dari 16MB secara otomatis dibagi menjadi beberapa batch.

Sebagian besar driver atau shell dapat menetapkan ukuran batch default yang kecil. Anda dapat menentukan ukuran batch sebagai bagian dari kueri Anda. Untuk tujuan menangkap log kueri lambat, kami merekomendasikan ukuran batch yang melebihi ukuran set hasil yang Anda harapkan. Jika Anda tidak yakin dengan ukuran set hasil, atau jika bervariasi, Anda juga dapat mengatur ukuran batch ke jumlah besar (mis., 100rb).

Namun, menggunakan ukuran batch yang lebih besar berarti lebih banyak hasil yang harus diambil dari basis data sebelum respons dikirim ke klien. Untuk beberapa kueri, hal itu dapat membuat penundaan lebih lama sebelum Anda mendapatkan hasil. Jika Anda tidak berencana untuk menggunakan seluruh set hasil, mungkin Anda akan menghabiskan lebih banyak I/O untuk memproses kueri dan membuang hasilnya.

Mengaktifkan Amazon DocumentDB Profiler

Mengaktifkan profiler pada sebuah klaster adalah proses tiga langkah. Pastikan bahwa semua langkah selesai, atau log profil tidak akan dikirim ke CloudWatch Log. Profiler diatur pada tingkat klaster dan dilakukan pada semua basis data dan intans klaster.

Untuk mengaktifkan profiler di klaster

1. Karena Anda tidak dapat mengubah grup parameter klaster default, pastikan Anda memiliki grup parameter klaster kustom yang tersedia. Untuk informasi selengkapnya, lihat Membuat grup parameter cluster Amazon DocumentDB.
2. Dengan menggunakan grup parameter klaster kustom yang tersedia, ubah parameter berikut: `profiler`, `profiler_threshold_ms`, dan `profiler_sampling_rate`. Untuk informasi selengkapnya, lihat Memodifikasi grup parameter cluster Amazon DocumentDB.
3. Membuat atau memodifikasi cluster Anda untuk menggunakan grup parameter cluster kustom dan untuk mengaktifkan ekspor`profiler`log ke CloudWatch Log.

Bagian berikut menunjukkan bagaimana menerapkan langkah-langkah ini menggunakan AWS Management Console dan AWS Command Line Interface (AWS CLI).

Using the AWS Management Console

1. Sebelum Anda mulai, buat klaster Amazon DocumentDB dan grup parameter klaster kustom jika Anda belum memilikinya. Untuk informasi selengkapnya, lihat Membuat grup parameter cluster Amazon DocumentDB dan Membuat cluster Amazon DocumentDB.
2. Dengan menggunakan grup parameter klaster kustom yang tersedia, ubah parameter berikut. Untuk informasi selengkapnya, lihat Memodifikasi grup parameter cluster Amazon DocumentDB.

   - `profiler` — Mengaktifkan atau menonaktifkan pembuatan profil kueri. Nilai yang diizinkan adalah `enabled` dan `disabled`. Nilai default-nya adalah `disabled`. Untuk mengaktifkan pembuatan profil, atur nilainya ke `enabled`.
   - `profiler_threshold_ms`— Kapan`profiler`diatur ke`enabled`, semua perintah yang memakan waktu lebih lama dari `profiler-threshold-ms`dicatat ke CloudWatch. Nilai yang diizinkan adalah `[50-INT_MAX]`. Nilai default-nya adalah `100`.
   - `profiler_sampling_rate` — Bagian dari operasi lambat yang harus diprofilkan atau dicatat. Nilai yang diizinkan adalah `[0.0-1.0]`. Nilai default-nya adalah `1.0`.
3. Ubah klaster Anda untuk menggunakan grup parameter klaster kustom dan atur ekspor log profiler untuk dipublikasikan ke Amazon CloudWatch.

   a. Di panel navigasi, pilih Klaster untuk menambahkan grup parameter kustom Anda ke sebuah klaster.

b. Pilih tombol di sebelah kiri nama klaster yang ingin Anda kaitkan dengan grup parameter Anda. Pilih Tindakan, lalu Modifikasi untuk memodifikasi klaster Anda.

c. Di bawah Opsi klaster, pilih grup parameter kustom dari langkah di atas untuk menambahkannya ke klaster Anda.

d. Di bawahEkspor log, pilihLog profileruntuk mempublikasikan ke Amazon CloudWatch.

e. Pilih Lanjutkan dan lihat ringkasan dari modifikasi Anda.

f. Setelah memverifikasi perubahan Anda, Anda dapat menerapkannya segera atau selama jendela pemeliharaan berikutnya di bawah Penjadwalan modifikasi .

g. Pilih Ubah klaster untuk memperbarui klaster Anda dengan grup parameter baru Anda.

Using the AWS CLI

Prosedur berikut mengaktifkan profiler pada semua operasi yang didukung untuk klaster `sample-cluster`.

1. Sebelum Anda mulai, pastikan bahwa Anda memiliki grup parameter klaster kustom yang tersedia dengan menjalankan perintah berikut, dan meninjau output untuk grup parameter klaster yang tidak memiliki `default` dalam nama dan memiliki `docdb3.6` sebagai keluarga grup parameter. Jika Anda tidak memiliki grup parameter klaster bukan-default, lihat Membuat grup parameter cluster Amazon DocumentDB.

```
aws docdb describe-db-cluster-parameter-groups \
    --query 'DBClusterParameterGroups[*].
[DBClusterParameterGroupName,DBParameterGroupFamily]'
```

Pada output berikut, hanya `sample-parameter-group` yang memenuhi kedua kriteria.

```
[
       [
           "default.docdb3.6",
           "docdb3.6"
       ],
       [
           "sample-parameter-group",
           "docdb3.6"
       ]
]
```

2. Dengan menggunakan grup parameter klaster kustom Anda, ubah parameter berikut:

   - `profiler` — Mengaktifkan atau menonaktifkan pembuatan profil kueri. Nilai yang diizinkan adalah `enabled` dan `disabled`. Nilai default-nya adalah `disabled`. Untuk mengaktifkan pembuatan profil, atur nilainya ke `enabled`.
   - `profiler_threshold_ms`— Kapan`profiler`diatur ke`enabled`, semua perintah membutuhkan waktu lebih lama dari`profiler -threshold-ms`dicatat ke CloudWatch. Nilai yang diizinkan adalah `[0-INT_MAX]`. Mengatur nilai ini ke profil `0` semua operasi yang didukung. Nilai default-nya adalah `100`.
   - `profiler_sampling_rate` — Bagian dari operasi lambat yang harus diprofilkan atau dicatat. Nilai yang diizinkan adalah `[0.0-1.0]`. Nilai default-nya adalah `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. Ubah klaster Amazon DocumentDB Anda sehingga menggunakan grup parameter klaster kustom `sample-parameter-group` dari langkah sebelumnya dan atur parameter `--enable-cloudwatch-logs-exports` ke `profiler`.

   Kode berikut memodifikasi clusters`sample-cluster`untuk menggunakan`sample-parameter-group`dari langkah sebelumnya, dan menambahkan`profiler`ke yang diaktifkan CloudWatch Log ekspor.

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --db-cluster-parameter-group-name sample-parameter-group \
       --cloudwatch-logs-export-configuration '{"EnableLogTypes":["profiler"]}'
```

   Output dari operasi ini akan terlihat seperti berikut.

```
{
    "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
    }
}
```

Menonaktifkan Amazon DocumentDB Profiler

Untuk menonaktifkan profiler, Anda menonaktifkan keduanya`profiler`parameter dan ekspor`profiler`log ke CloudWatch Log.

Menonaktifkan Profiler

Anda dapat menonaktifkan parameter `profiler` menggunakan AWS Management Console atau AWS CLI, sebagai berikut.

Using the AWS Management Console

Prosedur berikut menggunakan AWS Management Console untuk menonaktifkan Amazon DocumentDB `profiler`.

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter . Kemudian pilih nama grup parameter klaster yang ingin Anda nonaktifkan profilernya.
3. Di halaman Parameter klaster yang dihasilkan, pilih tombol di sebelah kiri parameter `profiler` dan pilih Edit.
4. Di kotak dialog Ubah profiler, pilih `disabled` di dalam daftar.
5. Pilih Ubah parameter klaster.

Using the AWS CLI

Untuk menonaktifkan `profiler` pada klaster menggunakan AWS CLI, modifikasi klaster sebagai berikut.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=disabled,ApplyMethod=immediate
```

Menonaktifkan Ekspor Log Profiler

Anda dapat menonaktifkan ekspor`profiler`log ke CloudWatch Log dengan menggunakan salah satuAWS Management ConsoleatauAWS CLI, sebagai berikut.

Using the AWS Management Console

Prosedur berikut menggunakanAWS Management Consoleuntuk menonaktifkan log ekspor Amazon DocumentDB ke CloudWatch.

1. Buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Klaster. Pilih tombol di sebelah kiri nama klaster yang log ekspornya ingin Anda nonaktifkan.
3. Pada menu Tindakan, pilih Modifikasi.
4. Gulir ke bawah ke bagian Ekspor log dan batalkan pilihan Log profiler.
5. Pilih Continue (Lanjutkan).
6. Tinjau perubahan Anda, lalu pilih kapan Anda ingin perubahan ini diterapkan ke klaster Anda:
   - Terapkan selama jendela pemeliharaan terjadwal berikutnya
   - Terapkan segera
7. Pilih Ubah klaster.

Using the AWS CLI

Kode berikut memodifikasi clustersample-cluster dan menonaktifkan CloudWatch log profiler.

Example

Untuk Linux, macOS, atau Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Untuk Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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
    }
}
```

Mengakses Log Amazon DocumentDB Profiler Anda

Ikuti langkah-langkah ini untuk mengakses log profil Anda di Amazon CloudWatch.

1. Buka CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/.
2. Pastikan Anda berada di Wilayah yang sama dengan klaster Amazon DocumentDB Anda.
3. Di panel navigasi, pilih Log.
4. Untuk menemukan log profiler untuk klaster Anda, dalam daftar, pilih `/aws/docdb/`*`yourClusterName`*`/profiler`.

   Log profil untuk setiap instans Anda tersedia di bawah setiap nama instans masing-masing.

Kueri umum

Berikut ini adalah beberapa kueri umum yang dapat Anda gunakan untuk menganalisis perintah yang diprofilkan. Untuk informasi lebih lanjut tentang CloudWatch Wawasan Log, lihatMenganalisis Data Log dengan CloudWatch Wawasan LogdanContoh Pertanyaan.

Dapatkan 10 Operasi Paling Lambat pada Koleksi Tertentu

```
filter ns="test.foo" | sort millis desc | limit 10
```

Dapatkan Semua Operasi Pembaruan pada Koleksi yang Membutuhkan Lebih dari 60 milidetik

```
filter millis > 60 and op = "update"
```

Dapatkan 10 Operasi Paling Lambat Dalam Sebulan Terakhir

```
sort millis desc | limit 10
```

Dapatkan Semua Kueri dengan Ringkasan Paket COLLSCAN

```
filter planSummary="COLLSCAN"
```

Memantau dengan Wawasan Kinerja

Performance Insights menambahkan fitur pemantauan Amazon DocumentDB yang ada untuk menggambarkan kinerja klaster Anda dan membantu Anda menganalisis masalah apa pun yang mempengaruhinya. Dengan dasbor Performance Insights, Anda dapat memvisualisasikan pemuatan database dan memfilter beban berdasarkan menunggu, pernyataan kueri, host, atau aplikasi.

> **Note**
>
> Performance Insights hanya tersedia untuk klaster berbasis instans Amazon DocumentDB 3.6, 4.0, dan 5.0.

Bagaimana ini berguna?

- Visualisasikan kinerja database - Visualisasikan beban untuk menentukan kapan dan di mana beban berada di database
- Tentukan apa yang menyebabkan beban pada database - Tentukan kueri, host, dan aplikasi mana yang berkontribusi terhadap pemuatan instans Anda
- Tentukan kapan ada pemuatan di database Anda — Perbesar dasbor Performance Insights untuk fokus pada peristiwa tertentu atau memperkecil untuk melihat tren di rentang waktu yang lebih besar
- Peringatan tentang pemuatan basis data - Akses metrik pemuatan basis data baru secara otomatis dari CloudWatch mana Anda dapat memantau metrik pemuatan DB bersama metrik DocumentDB lainnya dan mengatur peringatan pada mereka

Apa batasan Amazon DocumentDB Performance Insights?

- Performance Insights di wilayah AWS GovCloud (AS-Barat) belum tersedia
- Performance Insights untuk DocumentDB mempertahankan data kinerja hingga 7 hari
- Kueri dengan panjang lebih dari 1024kb tidak digabungkan dalam Performance Insights

Topik

- Konsep Performance Insights
- Mengaktifkan dan menonaktifkan Wawasan Kinerja
- Konfigurasikan kebijakan akses untuk Wawasan Kinerja
- Menganalisis metrik dengan dasbor Performance Insights
- Mengambil metrik dengan API Wawasan Kinerja
- CloudWatch Metrik Amazon untuk Performance Insights
- Performance Insights untuk metrik penghitung

Konsep Performance Insights

Topik

- Sesi Aktif Rata-Rata
- Dimensi
- Maks vCPU

Sesi Aktif Rata-Rata

Beban basis data (beban DB) mengukur tingkat aktivitas dalam database Anda. Metrik kunci dalam Performance Insights adalah`DB Load`, yang dikumpulkan setiap detik. Unit untuk `DBLoad` metrik adalah Average Active Sessions (AAS) untuk instance DocumentDB.

Sesi aktif adalah koneksi yang telah mengirimkan pekerjaan ke instance DocumentDB dan sedang menunggu tanggapan. Misalnya, jika Anda mengirimkan kueri ke instance DocumentDB, sesi database aktif saat instance memproses kueri.

Untuk mendapatkan sesi aktif rata-rata, Wawasan Kinerja mengambil sampel jumlah sesi yang secara bersamaan menjalankan kueri. AAS adalah jumlah total sesi dibagi dengan jumlah total sampel. Tabel berikut menunjukkan lima sampel berturut-turut dari query berjalan.

Sampel	Jumlah sesi yang menjalankan kueri	AAS	Perhitungan
1	2	2	2 sesi / 1 sampel
2	0	1	2 sesi / 2 sampel
3	4	2	6 sesi / 3 sampel
4	0	1.5	6 sesi / 4 sampel
5	4	2	10 sesi / 5 sampel

Pada contoh sebelumnya, Beban DB untuk interval waktu dari 1-5 adalah 2 AAS. Peningkatan beban DB berarti, rata-rata, lebih banyak sesi berjalan di database.

Dimensi

Metrik `DB Load` berbeda dari metrik seri waktu lainnya karena Anda dapat membaginya menjadi subkomponen yang disebut dimensi. Anda dapat menganggap dimensi sebagai kategori untuk karakteristik `DB Load` metrik yang berbeda. Saat Anda mendiagnosis masalah kinerja, dimensi yang paling berguna adalah status tunggu dan kueri teratas.

tunggu negara

Status tunggu menyebabkan pernyataan kueri menunggu peristiwa tertentu terjadi sebelum dapat terus berjalan. Misalnya, pernyataan kueri mungkin menunggu hingga sumber daya yang terkunci dibuka. Dengan menggabungkan `DB Load` dengan status tunggu, Anda bisa mendapatkan gambaran lengkap tentang status sesi. Berikut adalah berbagai status tunggu DocumentDB:

Status tunggu DocumentDB	Tunggu Deskripsi Negara
Kait	Status tunggu Latch terjadi saat sesi menunggu halaman kumpulan buffer. Sering paging masuk dan keluar dari buffer pool dapat terjadi lebih sering ketika sering ada kueri besar yang diproses oleh sistem, pemindaian koleksi, atau ketika kumpulan buffer terlalu kecil untuk menangani set kerja.
CPU	Status menunggu CPU terjadi ketika sesi sedang menunggu pada CPU.
CollectionLock	Status CollectionLock tunggu terjadi ketika sesi menunggu untuk mendapatkan kunci pada koleksi. Peristiwa ini terjadi ketika ada operasi DDL pada koleksi.
DocumentLock	Status DocumentLock tunggu terjadi ketika sesi menunggu untuk memperoleh kunci pada dokumen. Jumlah penulisan bersamaan yang tinggi ke dokumen yang sama akan berkontri busi pada lebih banyak status DocumentLock menunggu pada dokumen itu.
SystemLock	Status SystemLock tunggu terjadi ketika sesi sedang menunggu di sistem. Hal ini dapat terjadi ketika sering ada query berjalan lama, transaksi berjalan lama, atau konkurensi tinggi pada sistem.
IO	Status tunggu IO terjadi ketika sesi menunggu IO selesai.

Status tunggu DocumentDB	Tunggu Deskripsi Negara
BufferLock	Status BufferLock tunggu terjadi ketika sesi menunggu untuk memperoleh kunci pada halaman bersama di buffer. BufferLockstatus tunggu dapat diperpanjang jika proses lain menahan kursor terbuka pada halaman yang diminta.
LowMemThrottle	Status LowMemThrottle tunggu terjadi ketika sesi menunggu karena tekanan memori yang berat pada instance Amazon DocumentDB. Jika status ini berlanjut untuk waktu yang lama, pertimbangkan untuk meningkatkan instance untuk menyediakan memori tambahan. Untuk informasi lebih lanjut, lihat Gubernur Sumber Daya.
BackgroundActivity	Status BackgroundActivity tunggu terjadi ketika sesi sedang menunggu proses sistem internal.
Lainnya	Status tunggu lainnya adalah keadaan tunggu internal. Jika status ini bertahan lama, pertimbangkan untuk menghentikan kueri ini. Untuk informasi selengkapnya, lihat Bagaimana Cara Menemukan dan Menghentikan Kueri Berjalan Lama atau Diblokir?

Kueri teratas

Sementara status tunggu menunjukkan kemacetan, kueri teratas menunjukkan kueri mana yang paling berkontribusi pada pemuatan DB. Misalnya, saat ini mungkin ada banyak kueri yang berjalan di database, tetapi kueri tunggal mungkin menggunakan 99% dari muatan DB. Dalam hal ini, muatan tinggi mungkin menunjukkan masalah dalam kueri.

Maks vCPU

Di dasbor, bagan Muatan database mengumpulkan, menggabungkan, dan menampilkan informasi sesi. Untuk mengetahui jika sesi aktif melebihi CPU maksimum, lihat hubungannya dengan baris vCPU Maks. Nilai Max vCPU ditentukan oleh jumlah inti vCPU (virtual CPU) untuk instance DocumentDB Anda.

Jika beban DB sering di atas garis vCPU Max, dan status tunggu utama adalah CPU, CPU kelebihan beban. Dalam hal ini, Anda mungkin ingin membatasi koneksi ke instance, menyetel kueri apa pun dengan beban CPU yang tinggi, atau mempertimbangkan kelas instance yang lebih besar. Instans yang tinggi dan konsisten dari setiap status tunggu menunjukkan bahwa mungkin terjadi kemacetan atau masalah ketidakcocokan sumber daya yang perlu diselesaikan. Ini bisa benar bahkan jika beban DB tidak melewati garis vCPU Max.

Mengaktifkan dan menonaktifkan Wawasan Kinerja

Untuk menggunakan Wawasan Kinerja, aktifkan di instans DB Anda. Anda dapat menonaktifkannya nanti jika perlu. Mengaktifkan dan menonaktifkan Wawasan Kinerja tidak menyebabkan waktu henti, boot ulang, atau kegagalan.

Agen Wawasan Kinerja menggunakan CPU dan memori terbatas di host DB. Ketika muatan DB tinggi, agen membatasi dampak kinerja dengan mengurangi frekuensi pengumpulan data.

Mengaktifkan Performance Insights saat membuat klaster

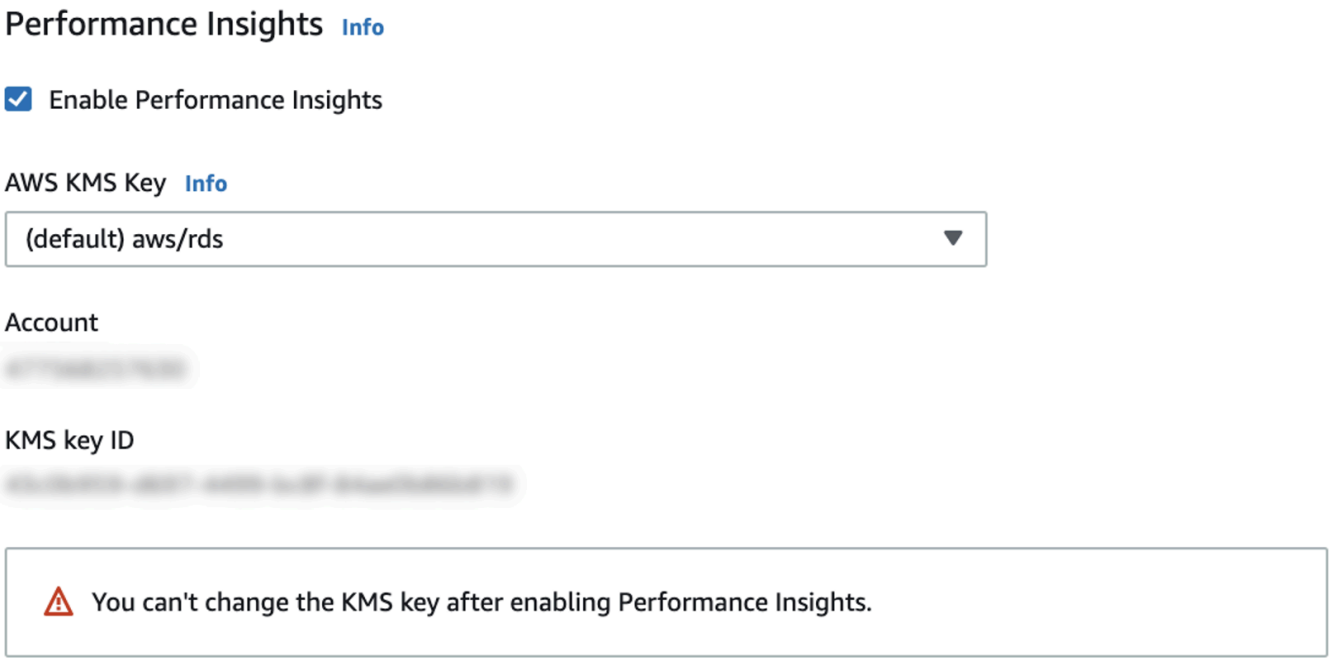
Di konsol, Anda dapat mengaktifkan atau menonaktifkan Wawasan Kinerja saat membuat atau mengubah instans DB baru.

Menggunakan AWS Management Console

Di konsol, Anda dapat mengaktifkan Performance Insights saat membuat cluster DocumentDB. Saat Anda membuat cluster DocumentDB baru, aktifkan Performance Insights dengan memilih Aktifkan Performance Insights di bagian Performance Insights.

Instruksi konsol

1. Untuk membuat cluster, ikuti petunjuk untuk Membuat cluster Amazon DocumentDB.
2. Pilih Aktifkan Performance Insights di bagian Performance Insights.



> **Note**
>
> Periode retensi data Performance Insights akan berlangsung selama tujuh hari.

AWS KMSkunci — Tentukan kunci AWS KMS Anda. Performance Insights mengenkripsi semua data yang berpotensi sensitif menggunakan kunci Anda. AWS KMS Data dienkripsi saat dipindahkan dan saat tidak digunakan. Untuk selengkapnya, lihat Mengonfigurasi AWS AWS KMS kebijakan untuk Performance Insights.

Mengaktifkan dan menonaktifkan saat memodifikasi instance

Anda dapat mengubah instans DB untuk mengaktifkan atau menonaktifkan Performance Insights menggunakan konsol atau. AWS CLI

Using the AWS Management Console

Instruksi konsol

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Pilih Klaster.

3. Pilih instans DB, lalu pilih Modifikasi.
4. Di bagian Performance Insights, pilih Aktifkan Performance Insights atau Nonaktifkan Performance Insights.

> **Note**
>
> Jika memilih Aktifkan Performance Insights, Anda dapat menentukan kunci. AWS AWS KMS Performance Insights mengenkripsi semua data yang berpotensi sensitif menggunakan kunci Anda. AWS KMS Data dienkripsi saat dipindahkan dan saat tidak digunakan. Untuk informasi selengkapnya, lihat Mengenkripsi Data Amazon DocumentDB saat Istirahat.

5. Pilih Lanjutkan.
6. Untuk Penjadwalan Modifikasi, pilih Langsung terapkan. Jika Anda memilih Terapkan selama jendela pemeliharaan terjadwal berikutnya, instans Anda akan mengabaikan setelan ini dan segera mengaktifkan Performance Insights.
7. Pilih Ubah instans.

Using the AWS CLI

Saat menggunakan `modify-db-instance` AWS AWS CLI perintah `create-db-instance` atau, Anda dapat mengaktifkan Performance Insights dengan menentukan`--enable-performance-insights`, atau menonaktifkannya dengan menentukan. `--no-enable-performance-insights`

Prosedur berikut menjelaskan cara mengaktifkan atau menonaktifkan Performance Insights untuk instans DB menggunakan. AWS AWS CLI

AWSAWS CLIinstruksi

Panggil `modify-db-instance` AWS AWS CLI perintah dan berikan nilai-nilai berikut:

- `--db-instance-identifer`— Nama instans DB
- `--enable-performance-insights` untuk mengaktifkan atau `--no-enable-performance-insights` untuk menonaktifkan

Example

Contoh berikut memungkinkan Performance Insights untuk: `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
```

Konfigurasikan kebijakan akses untuk Wawasan Kinerja

Untuk mengakses Wawasan Kinerja, Anda harus memiliki izin yang sesuai dari AWS Identity and Access Management (IAM). Anda memiliki opsi berikut untuk memberikan akses:

- Lampirkan kebijakan `AmazonRDSPerformanceInsightsReadOnly` terkelola ke set izin atau peran.
- Buat kebijakan IAM kustom dan lampirkan ke set izin atau peran.

Selain itu, jika Anda menetapkan kunci terkelola pelanggan saat mengaktifkan Performance Insights, pastikan pengguna di akun memiliki izin `kms:Decrypt` dan `kms:GenerateDataKey` izin pada kunci KMS.

> **Note**
>
> Untuk encryption-at-rest dengan manajemen AWS KMS kunci dan grup keamanan, Amazon DocumentDB memanfaatkan teknologi operasional yang dibagikan dengan Amazon RDS.

Melampirkan PerformanceInsightsReadOnly kebijakan AmazonRDS ke kepala sekolah IAM

`AmazonRDSPerformanceInsightsReadOnly`adalah kebijakan AWS terkelola yang memberikan akses ke semua operasi hanya-baca dari Amazon DocumentDB Performance Insights API. Saat ini, semua operasi di API ini hanya-baca. Jika Anda melampirkan `AmazonRDSPerformanceInsightsReadOnly` ke set izin atau peran, penerima dapat menggunakan Performance Insights dengan fitur konsol lainnya.

Membuat kebijakan IAM khusus untuk Wawasan Kinerja

Untuk pengguna yang tidak memiliki `AmazonRDSPerformanceInsightsReadOnly` kebijakan, Anda dapat memberikan akses ke Performance Insights dengan membuat atau memodifikasi kebijakan IAM yang dikelola pengguna. Jika Anda melampirkan kebijakan ke set izin atau peran, penerima dapat menggunakan Performance Insights.

Untuk membuat kebijakan khusus

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di halaman Buat Kebijakan, pilih tab JSON.
5. Salin dan tempel teks berikut, ganti *us-east-1* dengan nama AWS Wilayah Anda dan *111122223333* dengan nomor rekening pelanggan Anda.

```
{
    "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. Pilih Tinjau kebijakan.
7. Berikan nama untuk kebijakan tersebut dan secara opsional untuk deskripsinya, lalu pilih Buat kebijakan.

Anda sekarang dapat melampirkan kebijakan ke set izin atau peran. Prosedur berikut mengasumsikan bahwa Anda sudah memiliki pengguna yang tersedia untuk tujuan ini.

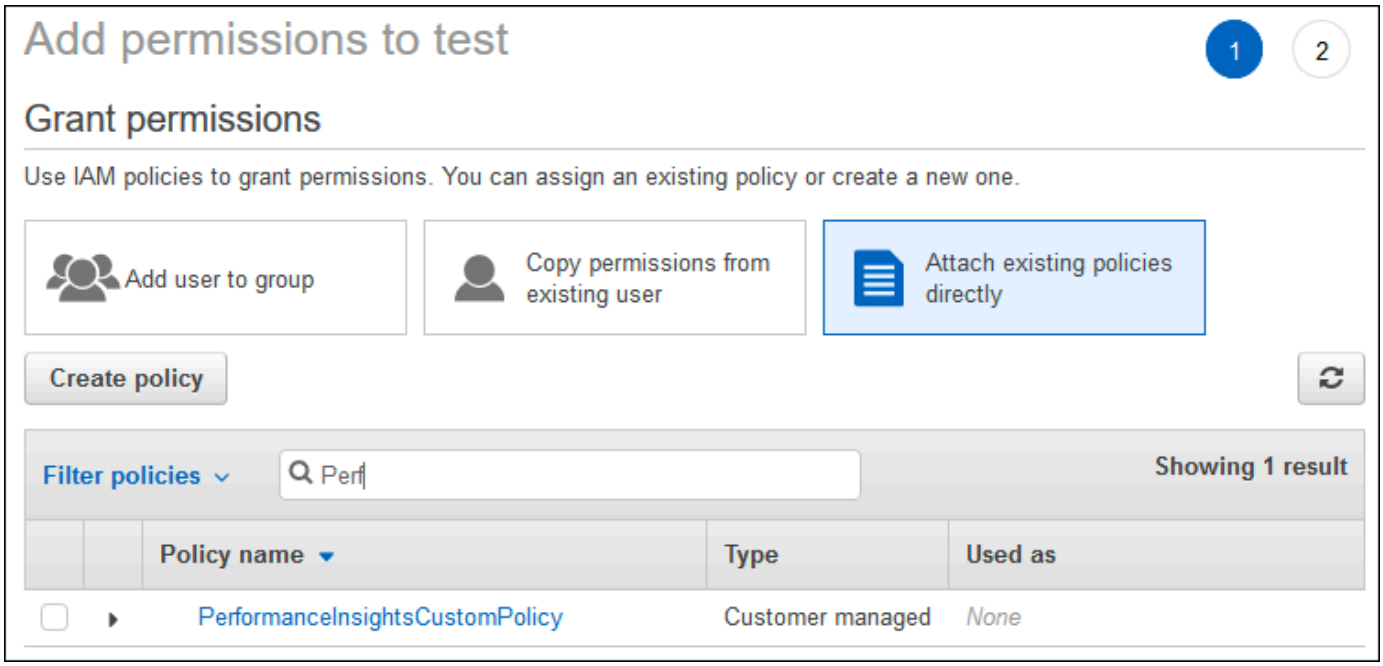
Untuk melampirkan kebijakan ke pengguna

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.

2. Di panel navigasi, pilih Pengguna.
3. Pilih pengguna yang ada dari daftar.

> **Important**
>
> Untuk menggunakan Performance Insights, pastikan Anda memiliki akses ke Amazon DocumentDB selain kebijakan kustom. Misalnya, kebijakan AmazonDocDB yang ReadOnlyAccess telah ditentukan menyediakan akses hanya-baca ke Amazon docdb.Untuk informasi selengkapnya, lihat Mengelola akses menggunakan kebijakan.

4. Di halaman Ringkasan, pilih Tambahkan izin.
5. Pilih Lampirkan kebijakan yang ada secara langsung. Untuk Pencarian, ketik beberapa karakter pertama dari nama kebijakan Anda, seperti yang ditampilkan di bawah ini.



6. Pilih kebijakan Anda, kemudian pilih Berikutnya: Tinjauan.
7. Pilih Tambahkan izin.

Mengonfigurasi AWS KMS kebijakan untuk Performance Insights

Performance Insights menggunakan perangkat AWS KMS key untuk mengenkripsi data sensitif. Saat mengaktifkan Wawasan Kinerja melalui API atau konsol, Anda akan memiliki opsi berikut:

- Pilih defaultKunci yang dikelola AWS.

Amazon DocumentDB menggunakan Kunci yang dikelola AWS instans DB baru Anda. Amazon DocumentDB membuat untuk Kunci yang dikelola AWS akun Anda. AWS AWSAkun Anda memiliki Amazon DocumentDB yang berbeda Kunci yang dikelola AWS untuk setiap Wilayah. AWS

- Pilih kunci yang dikelola pelanggan.

  Jika Anda menentukan kunci yang dikelola pelanggan, pengguna di akun yang memanggil API Performance Insights memerlukan izin `kms:Decrypt` dan `kms:GenerateDataKey` izin pada kunci KMS. Anda dapat mengkonfigurasi izin ini melalui kebijakan IAM. Namun, kami menyarankan Anda mengelola izin ini melalui kebijakan kunci KMS Anda. Untuk informasi lebih lanjut, lihat Menggunakan kebijakan kunci dalam KMS AWS.

Example

Contoh kebijakan kunci berikut menunjukkan cara menambahkan pernyataan ke kebijakan kunci KMS Anda. Pernyataan ini memungkinkan akses ke Wawasan Kinerja. Tergantung pada bagaimana Anda menggunakanAWS KMS, Anda mungkin ingin mengubah beberapa batasan. Sebelum menambahkan pernyataan ke kebijakan Anda, hapus semua komentar.

```
{
 "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"
        }
    }
}
```

Menganalisis metrik dengan dasbor Performance Insights

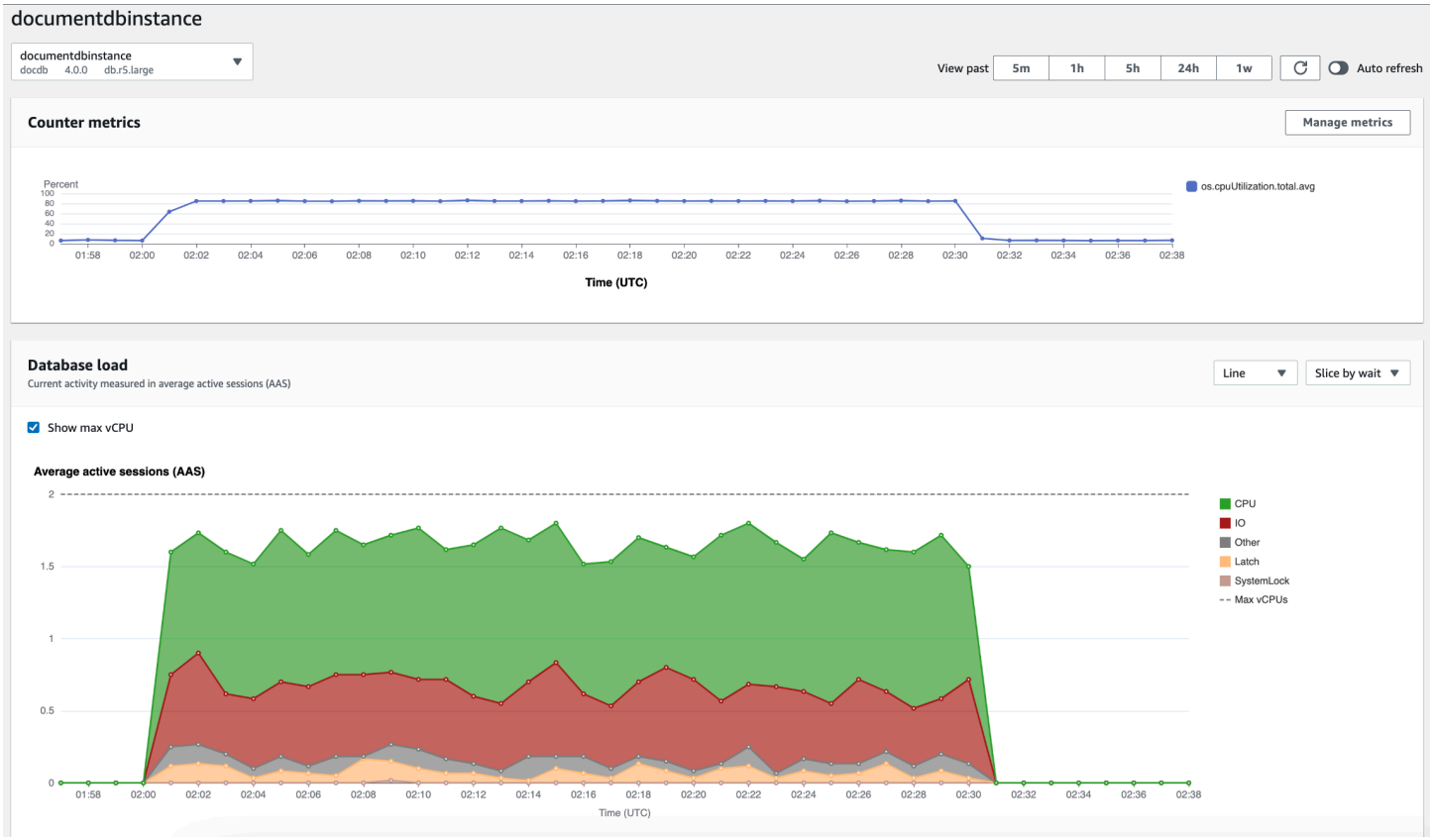
Dasbor Wawasan Kinerja berisi informasi kinerja database untuk membantu Anda menganalisis dan memecahkan masalah kinerja. Pada halaman dasbor utama, Anda dapat melihat informasi tentang beban database (beban DB). Anda dapat "mengiris" beban DB berdasarkan dimensi seperti status tunggu atau kueri.

Topik

- Ikhtisar dasbor Performance Insights
- Membuka dasbor Wawasan Kinerja
- Menganalisis beban basis data dengan status tunggu
- Ikhtisar tab Kueri teratas
- Memperbesar bagan beban database

Ikhtisar dasbor Performance Insights

Dasbor adalah cara termudah untuk berinteraksi dengan Performance Insights. Contoh berikut menunjukkan dasbor untuk instance Amazon DocumentDB. Secara default, dasbor Wawasan Kinerja menunjukkan data selama satu jam terakhir.



Dasbor dibagi menjadi beberapa bagian berikut:

1. Metrik penghitung — Menampilkan data untuk metrik penghitung kinerja tertentu.
2. Beban basis data - Menunjukkan bagaimana beban DB dibandingkan dengan kapasitas instans DB seperti yang diwakili oleh baris vCPU Max.
3. Dimensi teratas - Menunjukkan dimensi teratas yang berkontribusi terhadap beban DB. Dimensi ini meliputi`waits`,`queries`,`hosts`,`databases`, dan`applications`.

Topik

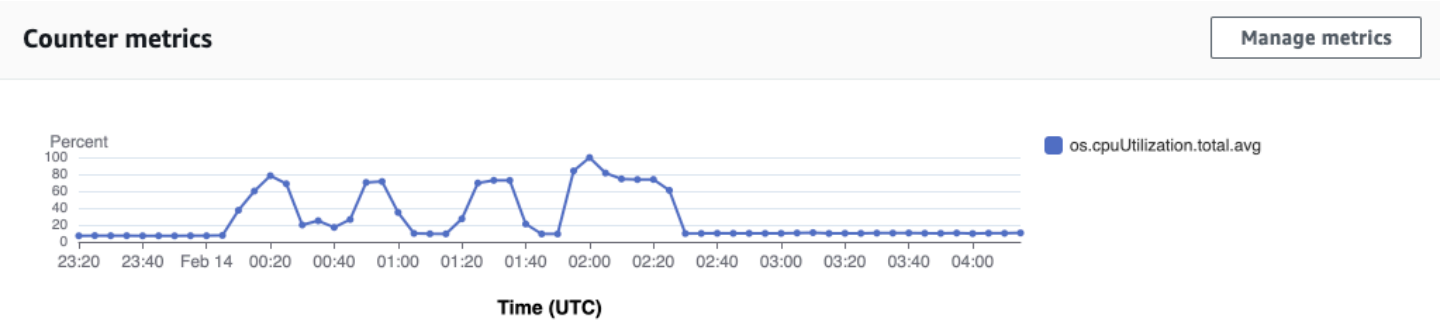
- Bagan metrik kontra
- Bagan beban basis data
- Tabel dimensi atas

Bagan metrik kontra

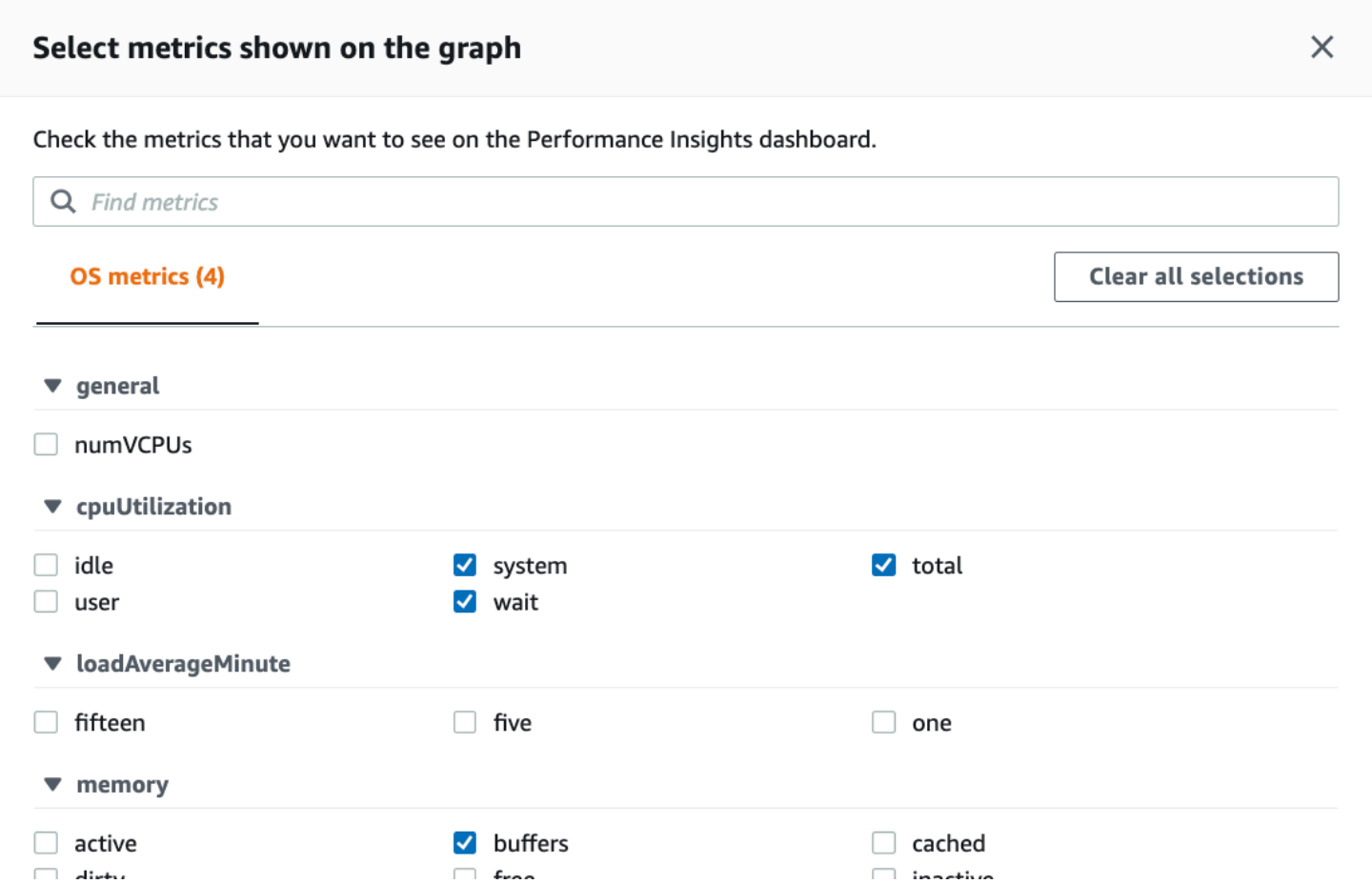
Dengan metrik penghitung, Anda dapat menyesuaikan dasbor Wawasan Kinerja untuk menyertakan hingga 10 bagan tambahan. Grafik ini menunjukkan pilihan lusinan metrik sistem operasi. Anda dapat

menghubungkan informasi ini dengan beban DB untuk membantu mengidentifikasi dan menganalisis masalah kinerja.

Bagan metrik Penghitung menampilkan data untuk penghitung kinerja.



Untuk mengubah penghitung kinerja, pilih Kelola metrik. Anda dapat memilih beberapa metrik OS seperti yang ditunjukkan pada gambar berikut. Untuk melihat detail setiap metrik, arahkan kursor ke nama metrik.



Bagan beban basis data

Bagan beban Database menunjukkan bagaimana aktivitas database dibandingkan dengan kapasitas instance seperti yang diwakili oleh baris Max vCPU. Secara default, bagan garis bertumpuk mewakili beban DB sebagai sesi aktif rata-rata per unit waktu. Beban DB diiris (dikelompokkan) berdasarkan status tunggu.

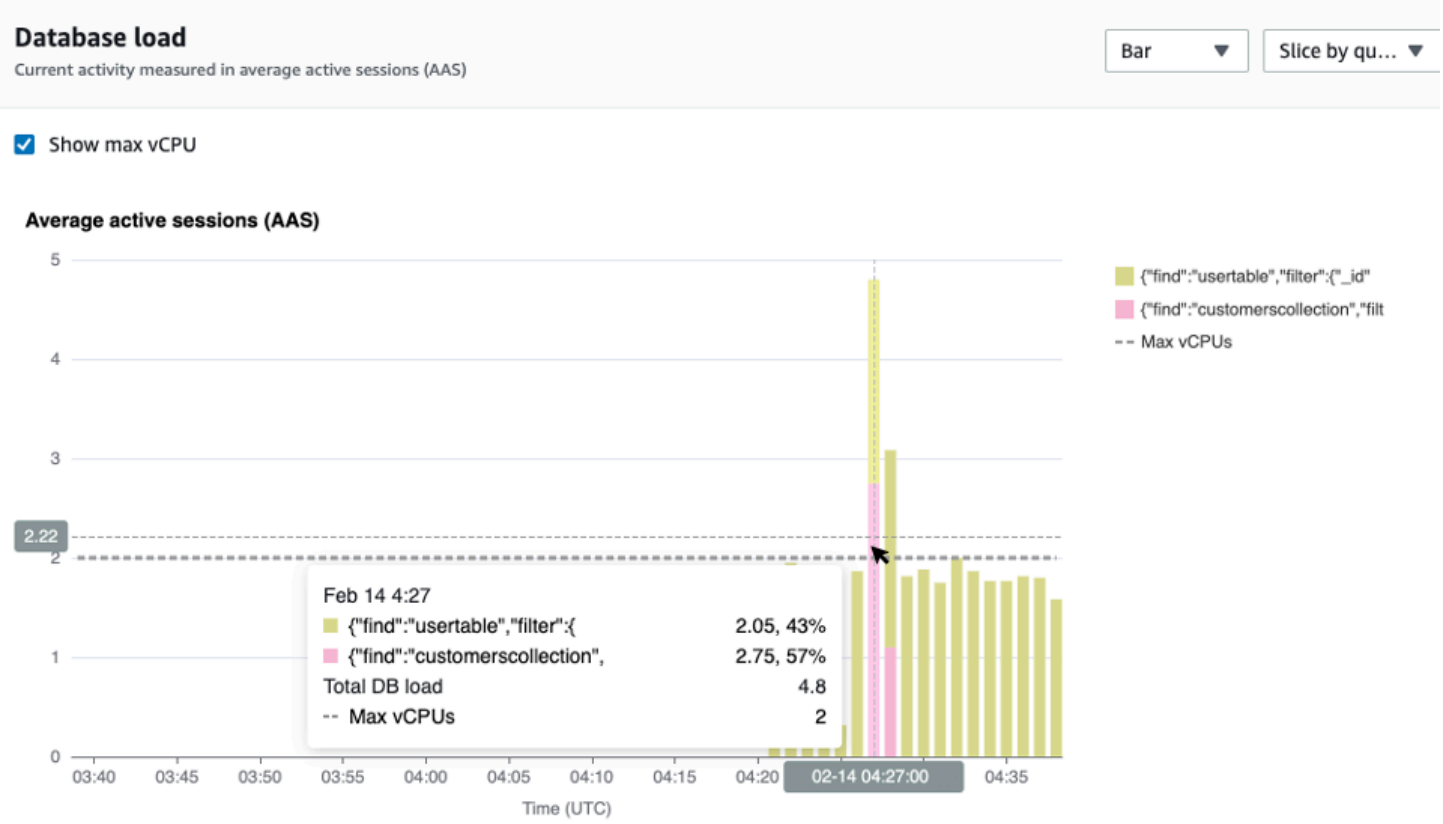
Beban DB diiris oleh dimensi

Anda dapat memilih untuk menampilkan beban sebagai sesi aktif yang dikelompokkan berdasarkan dimensi yang didukung. Gambar berikut menunjukkan dimensi untuk instance Amazon DocumentDB.

Detail beban DB untuk item dimensi

Untuk melihat detail tentang item pemuatan DB dalam dimensi, arahkan kursor ke nama item. Gambar berikut menunjukkan rincian untuk pernyataan query.

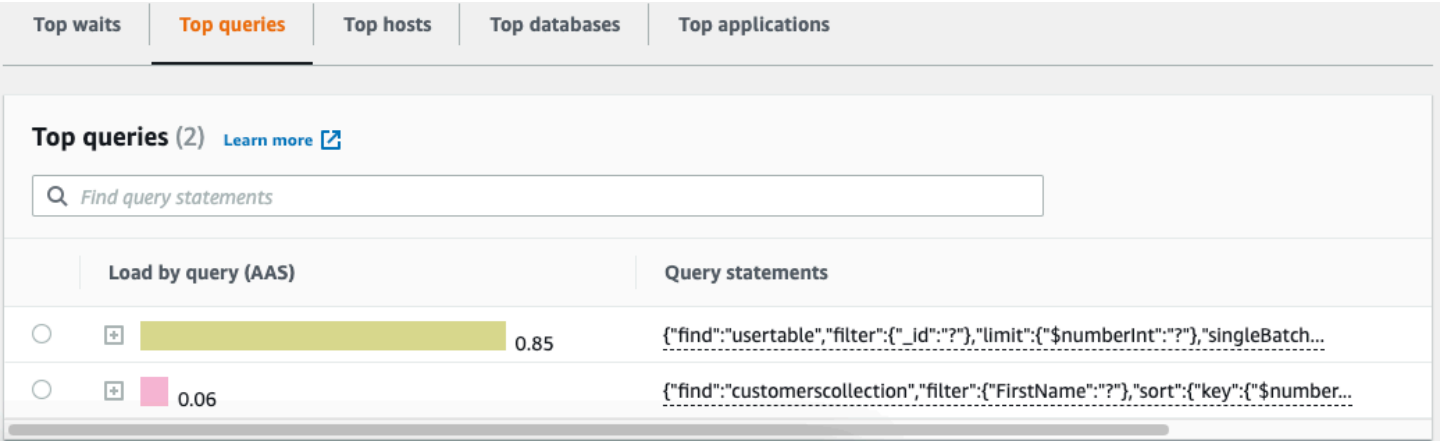
Untuk melihat detail setiap item untuk periode waktu yang dipilih dalam legenda, arahkan kursor ke item tersebut.



Tabel dimensi atas

Tabel dimensi atas mengiris beban DB dengan dimensi yang berbeda. Dimensi adalah kategori atau “slice by” untuk karakteristik beban DB yang berbeda. Jika dimensinya adalah kueri, Kueri teratas menunjukkan pernyataan kueri yang paling berkontribusi pada beban DB.

Pilih salah satu tab dimensi berikut.



Tabel berikut memberikan deskripsi singkat dari setiap tab.

Deskripsi
Acara mana backend database sedang menunggu
Pernyataan teratas kueri yang sedang berjalan
Jumlah teratas port klien yang terhubung
Nama database tempat klien terhubung
Aplikasi yang terhubung ke database

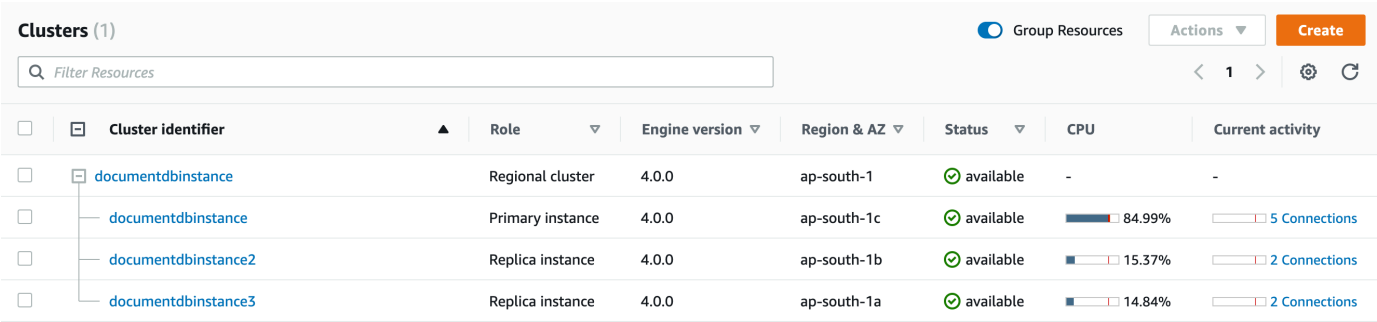
Untuk mempelajari cara menganalisis kueri menggunakan tab Kueri teratas, lihat. Ikhtisar tab Kueri teratas

Membuka dasbor Wawasan Kinerja

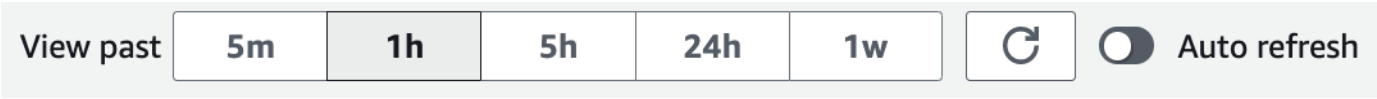
Untuk melihat dasbor Performance Insights di AWS Management Console, gunakan langkah-langkah berikut:

1. Buka konsol Performance Insights di https://console.aws.amazon.com/docdb/.
2. Pilih instans DB. Dasbor Performance Insights ditampilkan untuk instans Amazon DocumentDB tersebut.

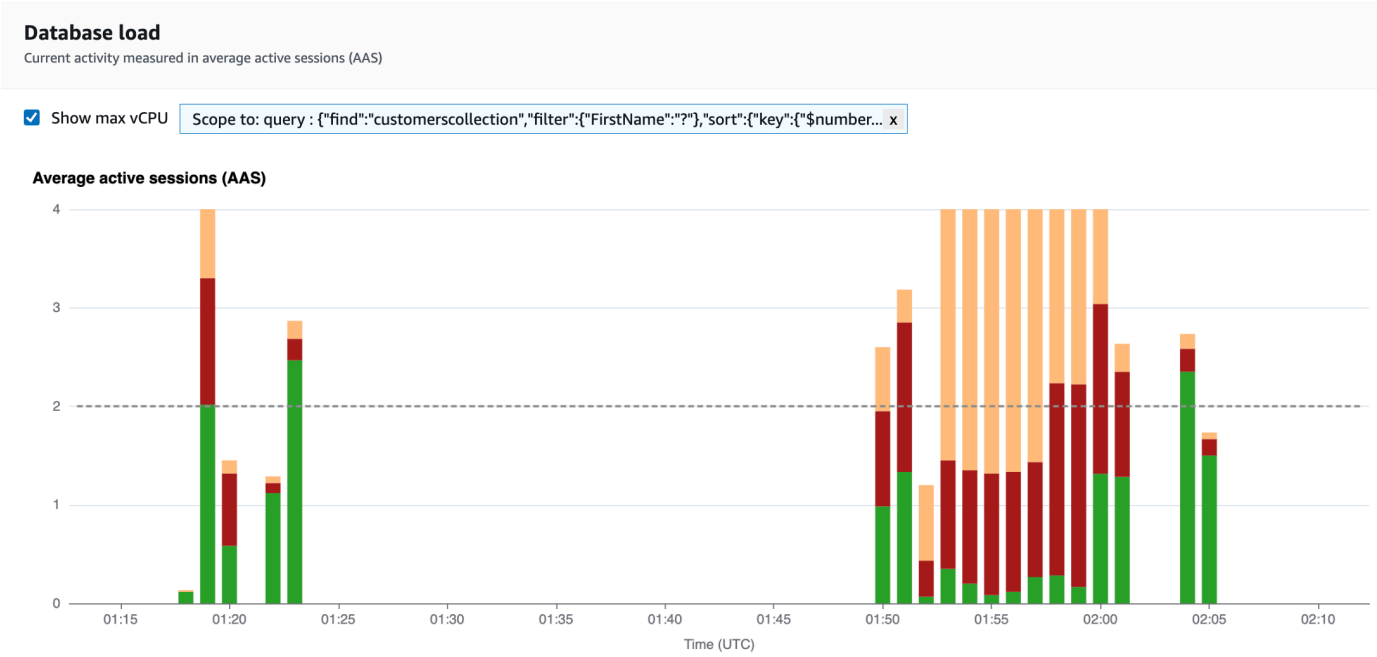
   Untuk instans Amazon DocumentDB dengan Performance Insights diaktifkan, Anda juga dapat mencapai dasbor dengan memilih item Sessions dalam daftar instance. Di bawah Aktivitas saat ini, item Sesi menunjukkan muatan database dalam sesi aktif rata-rata selama lima menit terakhir. Bilah secara grafis menunjukkan muatan. Ketika bar kosong, instance adalah idle. Saat muatan meningkat, bilah akan terisi dengan warna biru. Ketika beban melewati jumlah CPU virtual (vCPU) pada kelas instance, bilah berubah menjadi merah, menunjukkan potensi hambatan.



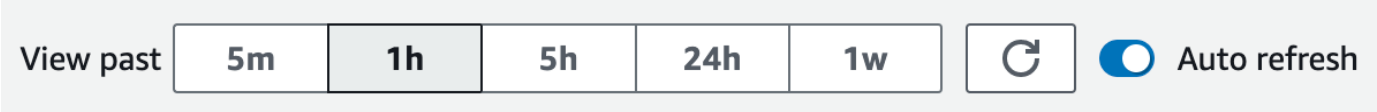
3. (Opsional) Pilih interval waktu yang berbeda dengan memilih tombol di kanan atas. Misalnya, untuk mengubah interval menjadi 1 jam, pilih 1 jam.



   Pada tangkapan layar berikut, interval beban DB adalah 1 jam.



4. Untuk menyegarkan data Anda secara otomatis, aktifkan Penyegaran otomatis.



Dasbor Wawasan Kinerja secara otomatis menyegarkan kembali data baru. Kecepatan penyegaran tergantung pada jumlah data yang ditampilkan:

- 5 menit menyegarkan setiap 5 detik.
- 1 jam menyegarkan setiap menit.
- 5 jam menyegarkan setiap menit.
- 24 jam menyegarkan setiap 5 menit.
- 1 minggu menyegarkan setiap jam.

Menganalisis beban basis data dengan status tunggu

Jika bagan beban Database (beban DB) menunjukkan kemacetan, Anda dapat mengetahui dari mana beban itu berasal. Untuk melakukannya, lihat tabel item beban teratas di bawah bagan beban Database. Pilih item tertentu, seperti kueri atau aplikasi, untuk menelusuri item itu dan melihat detailnya.

Beban DB yang dikelompokkan berdasarkan menunggu dan kueri teratas biasanya memberikan wawasan paling banyak tentang masalah kinerja. Muatan DB yang dikelompokkan berdasarkan

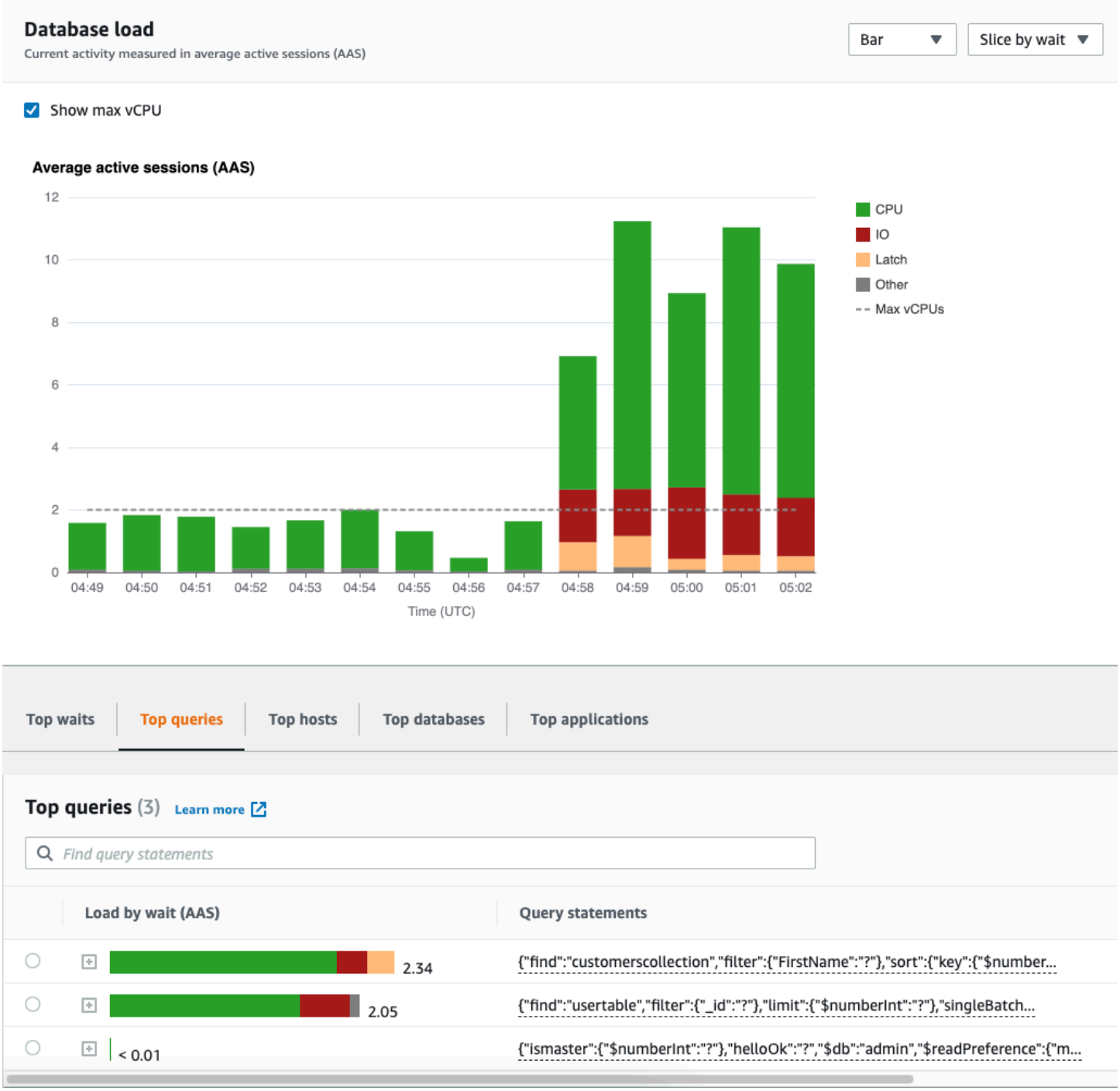
tunggu menunjukkan jika ada sumber daya atau kemacetan konkurensi dalam database. Dalam hal ini, tab Kueri teratas pada tabel item pemuatan atas menunjukkan kueri mana yang mendorong pemuatan itu.

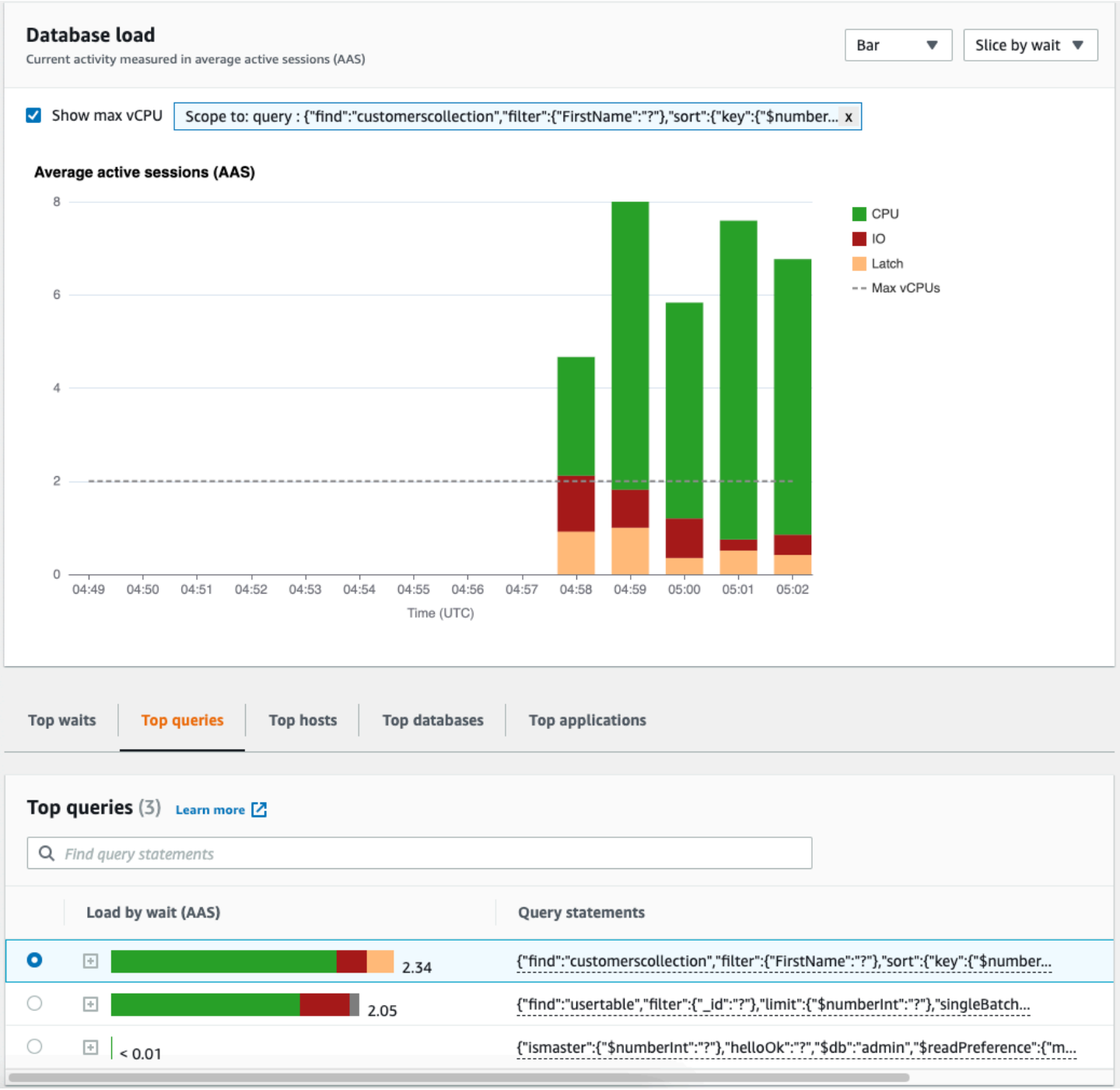
Alur kerja umum Anda untuk mendiagnosis masalah kinerja adalah sebagai berikut:

1. Tinjau bagan beban Database dan lihat apakah ada insiden beban basis data yang melebihi baris CPU Maks.
2. Jika ada, lihat bagan beban Database dan identifikasi status atau status tunggu mana yang paling bertanggung jawab.
3. Identifikasi kueri intisari yang menyebabkan pemuatan dengan melihat kueri mana yang tab Kueri teratas di tabel item pemuatan atas yang paling berkontribusi pada status tunggu tersebut. Anda dapat mengidentifikasi ini dengan kolom Load by Wait (AAS).
4. Pilih salah satu kueri intisari ini di tab Kueri teratas untuk memperluasnya dan melihat kueri anak yang terdiri darinya.

Anda juga dapat melihat host atau aplikasi mana yang berkontribusi paling banyak dengan memilih Host teratas atau Aplikasi teratas, masing-masing. Nama aplikasi ditentukan dalam string koneksi ke instance Amazon DocumentDB. `Unknown`menunjukkan bahwa bidang aplikasi tidak ditentukan.

Misalnya, di dasbor berikut, CPU menunggu sebagian besar beban DB. Memilih kueri teratas di bawah Kueri teratas akan mencakup bagan pemuatan Database untuk fokus pada beban terbanyak yang disumbangkan oleh kueri pilih.


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
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)
CPU
IO
Latch
Other
Max vCPUs
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...



Ikhtisar tab Kueri teratas

Secara default, tab Kueri teratas menunjukkan kueri yang paling berkontribusi pada pemuatan DB. Anda dapat menganalisis teks kueri untuk membantu menyetel kueri Anda.

Topik

- Intisari kueri

- Muat dengan menunggu (AAS)
- Melihat informasi kueri terperinci
- Mengakses teks kueri pernyataan
- Melihat dan mengunduh teks permintaan pernyataan

Intisari kueri

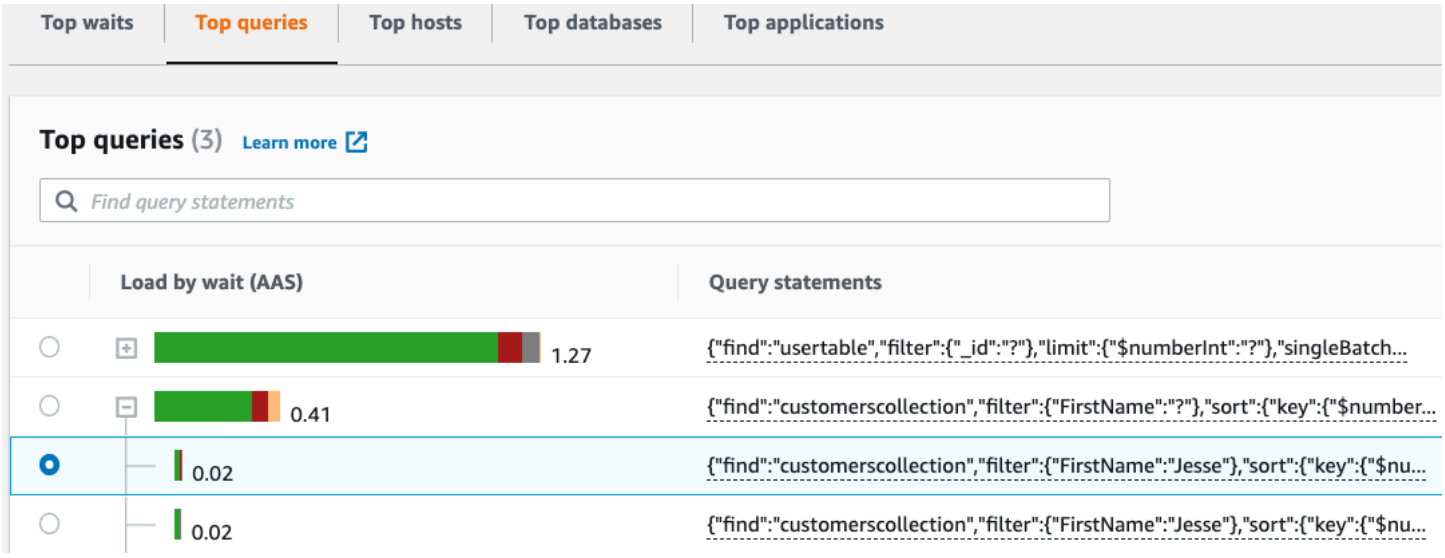
Intisari kueri adalah gabungan dari beberapa kueri aktual yang secara struktural serupa tetapi mungkin memiliki nilai literal yang berbeda. Digest menggantikan nilai yang terkodekan dengan tanda tanya. Misalnya, intisari kueri mungkin terlihat seperti ini:

```
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":
{"$numberInt":"?"}},"limit":{"$numberInt":"?"}}
```

Intisari ini mungkin mencakup kueri anak berikut:

```
{"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"}}
```

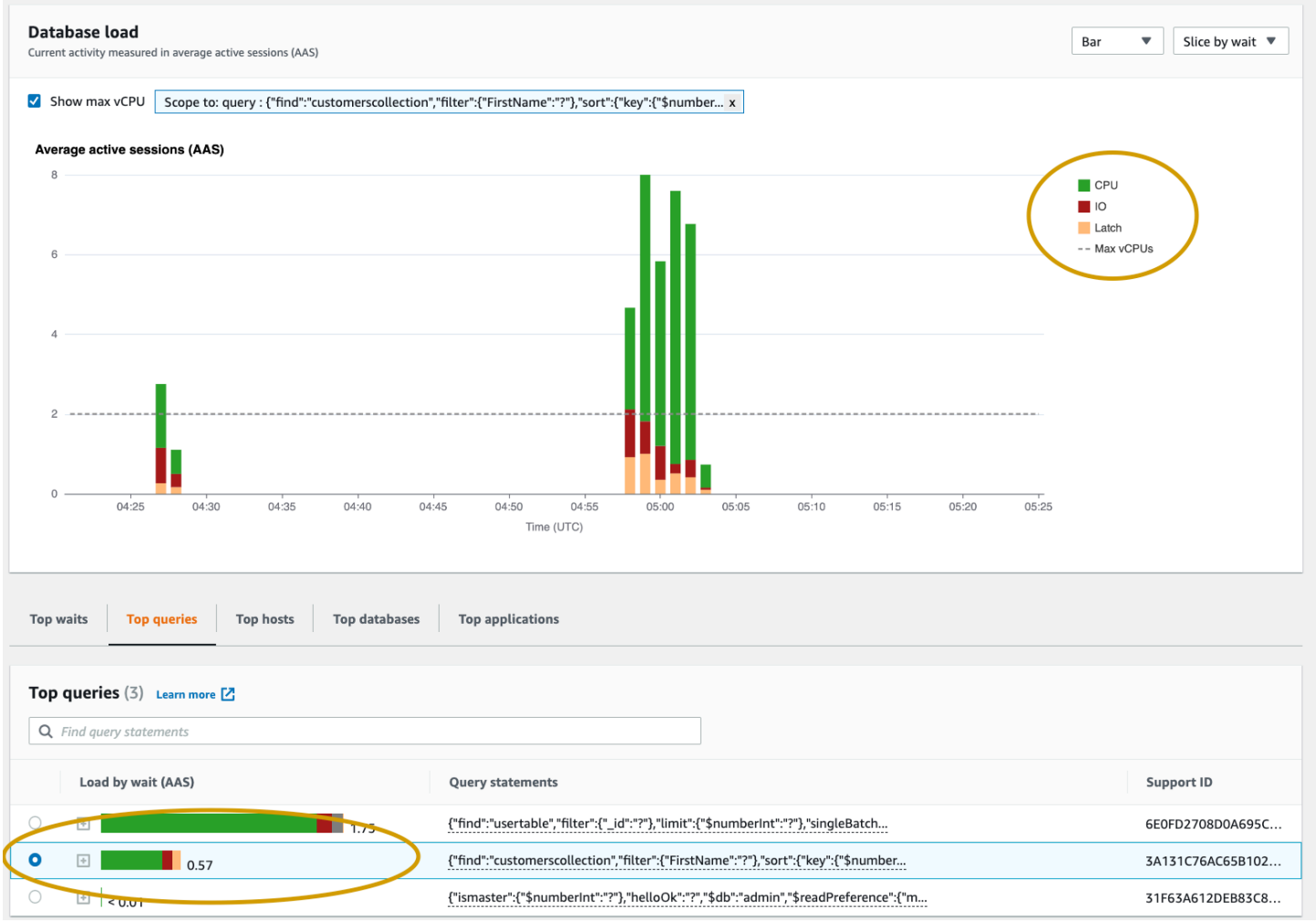
Untuk melihat pernyataan kueri literal dalam intisari, pilih kueri, lalu pilih simbol plus (+). Di tangkapan layar berikut, kueri yang dipilih adalah sebuah intisari.



> **Note**
>
> Intisari kueri mengelompokkan pernyataan kueri serupa, tetapi tidak menyunting informasi sensitif.

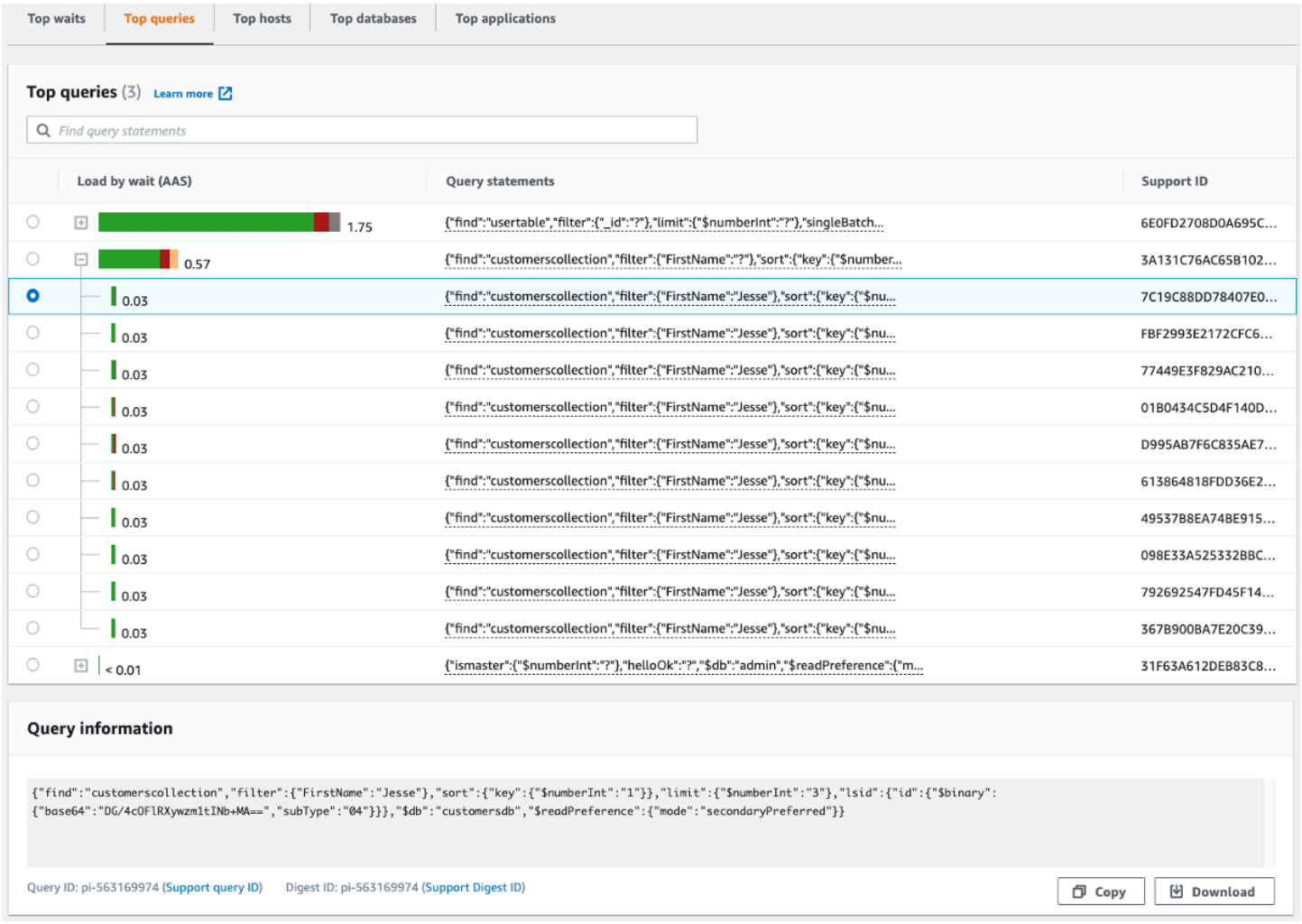
Muat dengan menunggu (AAS)

Di Kueri teratas, kolom Load by waits (AAS) menggambarkan persentase beban database yang terkait dengan setiap item pemuatan teratas. Kolom ini mencerminkan beban untuk item tersebut dengan pengelompokan apa pun yang saat ini dipilih dalam bagan beban DB. Misalnya, Anda dapat mengelompokkan bagan beban DB berdasarkan status tunggu. Dalam kasus ini, bilah Muatan DB berdasarkan Tunggu diberi ukuran, segmentasi, dan kode warna untuk menunjukkan seberapa banyak status tunggu tertentu yang dikontribusikan oleh kueri. Bilah ini juga menunjukkan status tunggu mana yang memengaruhi kueri yang dipilih.



Melihat informasi kueri terperinci

Dalam tabel kueri Top, Anda dapat membuka pernyataan intisari untuk melihat informasinya. Informasi tersebut muncul di panel bawah.



Jenis pengidentifikasi (ID) berikut dikaitkan dengan pernyataan kueri:

1. Support query ID - Nilai hash dari ID kueri. Nilai ini hanya untuk mereferensikan ID kueri saat Anda bekerja dengan AWS Support. AWS Support tidak memiliki akses ke ID kueri dan teks kueri Anda yang sebenarnya.
2. Support digest ID — Nilai hash dari ID intisari. Nilai ini hanya untuk mereferensikan ID intisari saat Anda bekerja dengan AWSSupport. AWS Support tidak memiliki akses ke ID intisari dan teks kueri Anda yang sebenarnya.

Mengakses teks kueri pernyataan

Secara default, setiap baris dalam tabel Kueri Top menunjukkan 500 byte teks kueri untuk setiap pernyataan kueri. Jika pernyataan intisari melebihi 500 byte, Anda dapat melihat teks lainnya dengan membuka pernyataan di dasbor Performance Insights. Dalam hal ini, panjang maksimum untuk kueri yang ditampilkan adalah 1 KB. Jika Anda melihat pernyataan kueri lengkap, Anda juga dapat memilih Unduh.

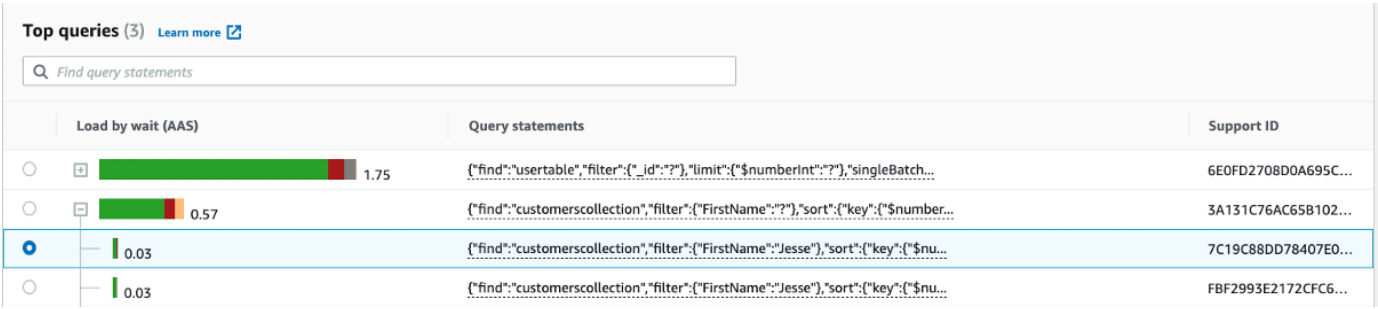
Melihat dan mengunduh teks permintaan pernyataan

Di dasbor Performance Insights, Anda dapat melihat atau mengunduh teks kueri.

Untuk melihat teks kueri lainnya di dasbor Performance Insights

1. Buka konsol Amazon DocumentDB di: https://console.aws.amazon.com/docdb/
2. Di panel navigasi, pilih Wawasan Kinerja.
3. Pilih instans DB. Dasbor Wawasan Kinerja ditampilkan untuk instans DB tersebut.

   Pernyataan kueri dengan teks yang lebih besar dari 500 byte akan terlihat seperti gambar berikut:



4. Periksa bagian informasi kueri untuk melihat lebih banyak teks kueri.

Dasbor Performance Insights dapat menampilkan hingga 1 KB untuk setiap pernyataan kueri lengkap.

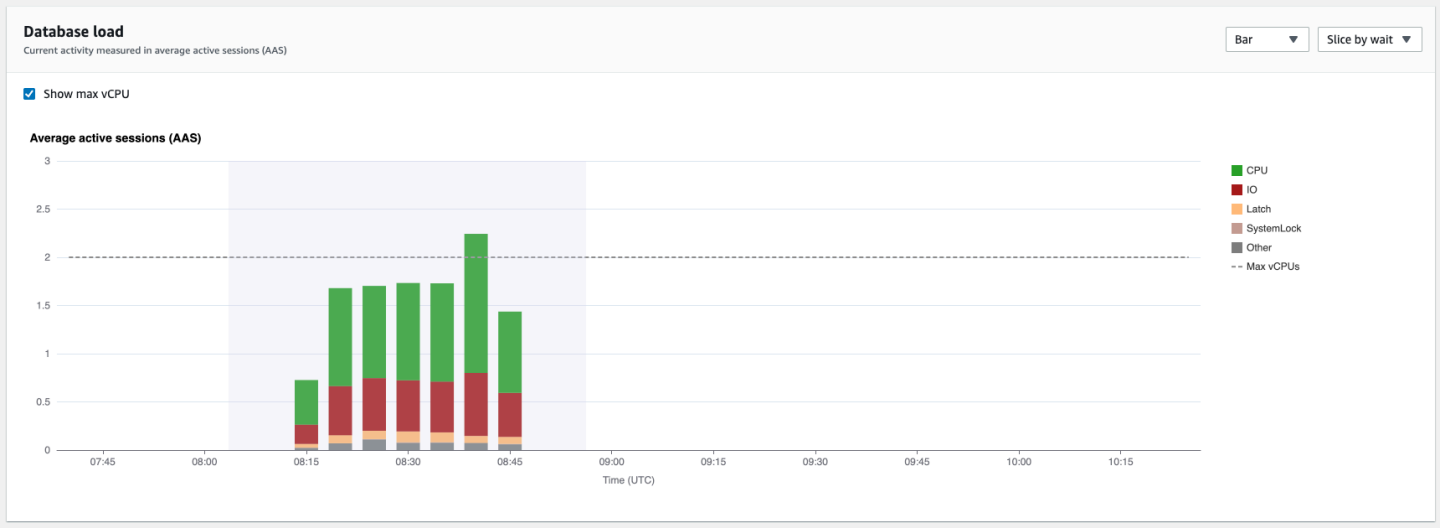
> **Note**
>
> Untuk menyalin atau mengunduh pernyataan kueri, nonaktifkan pemblokir pop-up apa pun.

Memperbesar bagan beban database

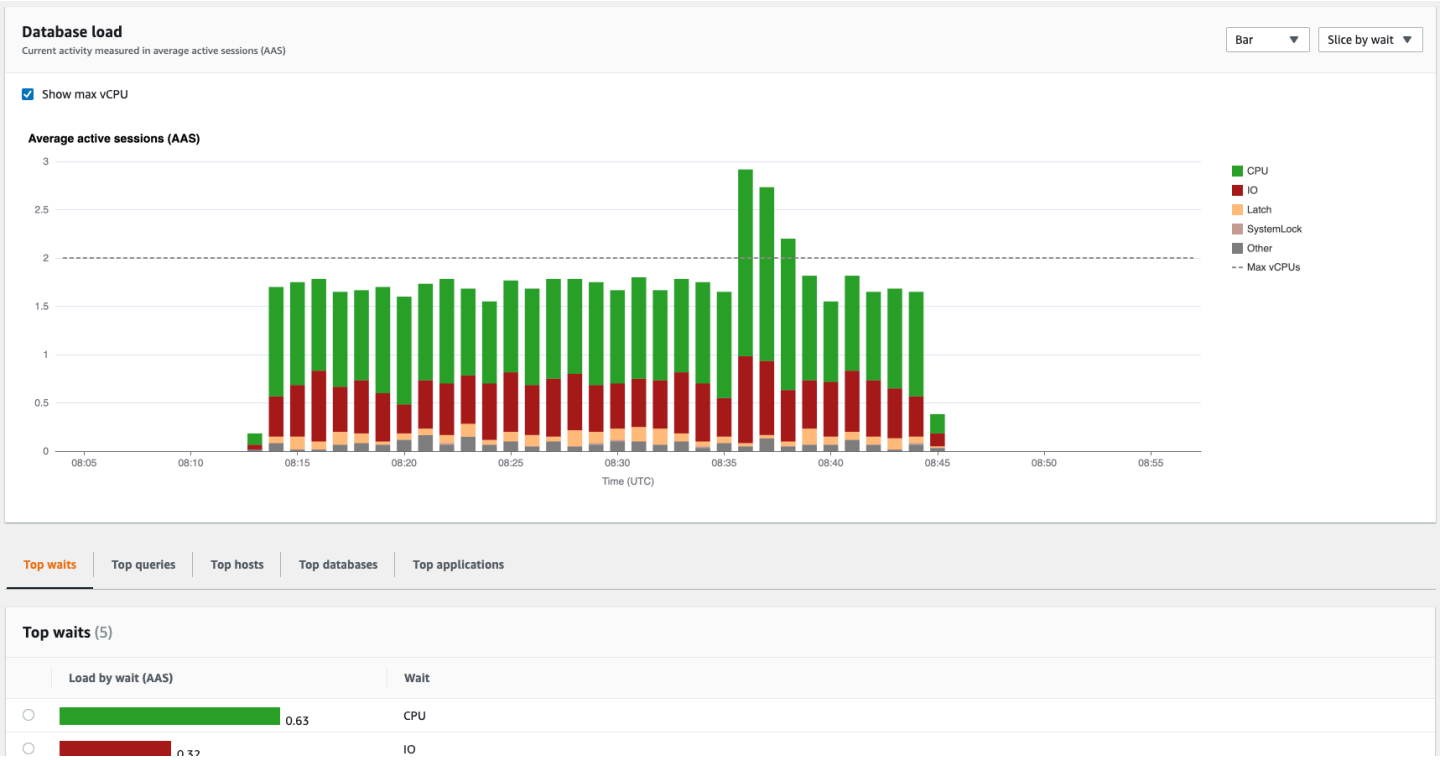
Anda dapat menggunakan fitur lain di antarmuka pengguna Wawasan Kinerja untuk membantu menganalisis data kinerja.

Perbesar dengan Klik dan Seret

Dalam antarmuka Wawasan Kinerja, Anda dapat memilih sebagian kecil bagan muatan dan memperbesar detailnya.



Untuk memperbesar sebagian bagan muatan, pilih waktu mulai dan seret ke akhir periode waktu yang Anda inginkan. Saat melakukannya, area yang dipilih akan disorot. Saat Anda melepaskan mouse, bagan beban memperbesar area yang dipilih, dan tabel Item teratas dihitung ulang.



Mengambil metrik dengan API Wawasan Kinerja

Saat Wawasan Kinerja diaktifkan, API menyediakan visibilitas ke dalam kinerja instans. Amazon CloudWatch Logs menyediakan sumber otoritatif untuk metrik pemantauan terjual untuk layanan. AWS

Wawasan Kinerja menawarkan tampilan spesifik domain dari muatan database yang diukur sebagai sesi aktif rata-rata (AAS). Metrik ini muncul sebagai kumpulan data seri waktu dua dimensi bagi konsumen API. Dimensi waktu data menyediakan data muatan DB untuk setiap titik waktu dalam rentang waktu yang diminta. Setiap titik waktu menguraikan keseluruhan muatan dalam hubungannya dengan dimensi yang diminta, seperti `Query`, `Wait-state`, `Application`, atau `Host`, diukur pada titik waktu tersebut.

Amazon DocumentDB Performance Insights memantau instans Amazon DocumentDB DB Anda sehingga Anda dapat menganalisis dan memecahkan masalah kinerja database. Salah satu cara untuk menampilkan data Wawasan Kinerja ada di AWS Management Console. Wawasan Kinerja juga menyediakan API publik sehingga Anda dapat meminta data Anda sendiri. Anda dapat menggunakan API untuk melakukan hal berikut:

- Membongkar data ke dalam basis data
- Menambahkan data Wawasan Kinerja ke dasbor pemantauan yang ada

- Bangun alat pemantauan

Untuk menggunakan API Performance Insights, aktifkan Performance Insights di salah satu instans Amazon DocumentDB Anda. Untuk informasi tentang mengaktifkan Wawasan Kinerja, lihat Mengaktifkan dan menonaktifkan Wawasan Kinerja. Untuk informasi selengkapnya tentang API Performance Insights, lihat Referensi API Performance Insights.

API Wawasan Kinerja menyediakan operasi berikut.

Tindakan Performance Insights	Perintah AWS CLI	Deskripsi
`DescribeDimensionKeys`	`aws pi describe-dimension-keys`	Mengambil kunci dimensi N teratas untuk metrik dalam jangka waktu tertentu.
`GetDimensionKeyDetails`	`aws pi get-dimension-key-details`	Mengambil atribut dari grup dimensi tertentu untuk instans DB atau sumber data. Misalnya, jika Anda menentukan ID kueri, dan jika detail dimensi tersedia, `GetDimensionKeyDetails` mengambil teks lengkap dari dimensi yang `db.query.statement` terkait dengan ID ini. Operasi ini berguna karena `GetResourceMetrics` dan `DescribeDimensionKeys` tidak mendukung pengambilan teks pernyataan kueri besar.
`GetResourceMetadata`	`aws pi get-resource-metadata`	Ambil metadata untuk fitur yang berbeda. Misalnya, metadata mungkin menunjukk

Tindakan Performance Insights	Perintah AWS CLI	Deskripsi
		an bahwa fitur diaktifkan atau dimatikan pada instans DB tertentu.
GetResourceMetrics	aws pi get-resource-metrics	Mengambil metrik Wawasan Kinerja untuk sekumpulan sumber data, selama jangka waktu tertentu. Anda dapat menyediakan grup dimensi dan dimensi tertentu, serta memberikan kriteria agregasi dan pemfilteran untuk setiap grup.
ListAvailableResourceDimensions	aws pi list-available-resource-dimensions	Ambil dimensi yang dapat ditanyakan untuk setiap jenis metrik yang ditentukan pada instance tertentu.
ListAvailableResourceMetrics	aws pi list-available-resource-metrics	Ambil semua metrik yang tersedia dari jenis metrik tertentu yang dapat ditanyakan untuk instans DB tertentu.

Topik

- AWS CLI untuk Wawasan Kinerja
- Mengambil metrik seri waktu
- Contoh AWS CLI untuk Wawasan Kinerja

AWS CLI untuk Wawasan Kinerja

Anda dapat melihat data Wawasan Kinerja menggunakan AWS CLI. Anda dapat menampilkan bantuan untuk perintah AWS CLI untuk Wawasan Kinerja dengan memasukkan yang berikut ini di baris perintah.

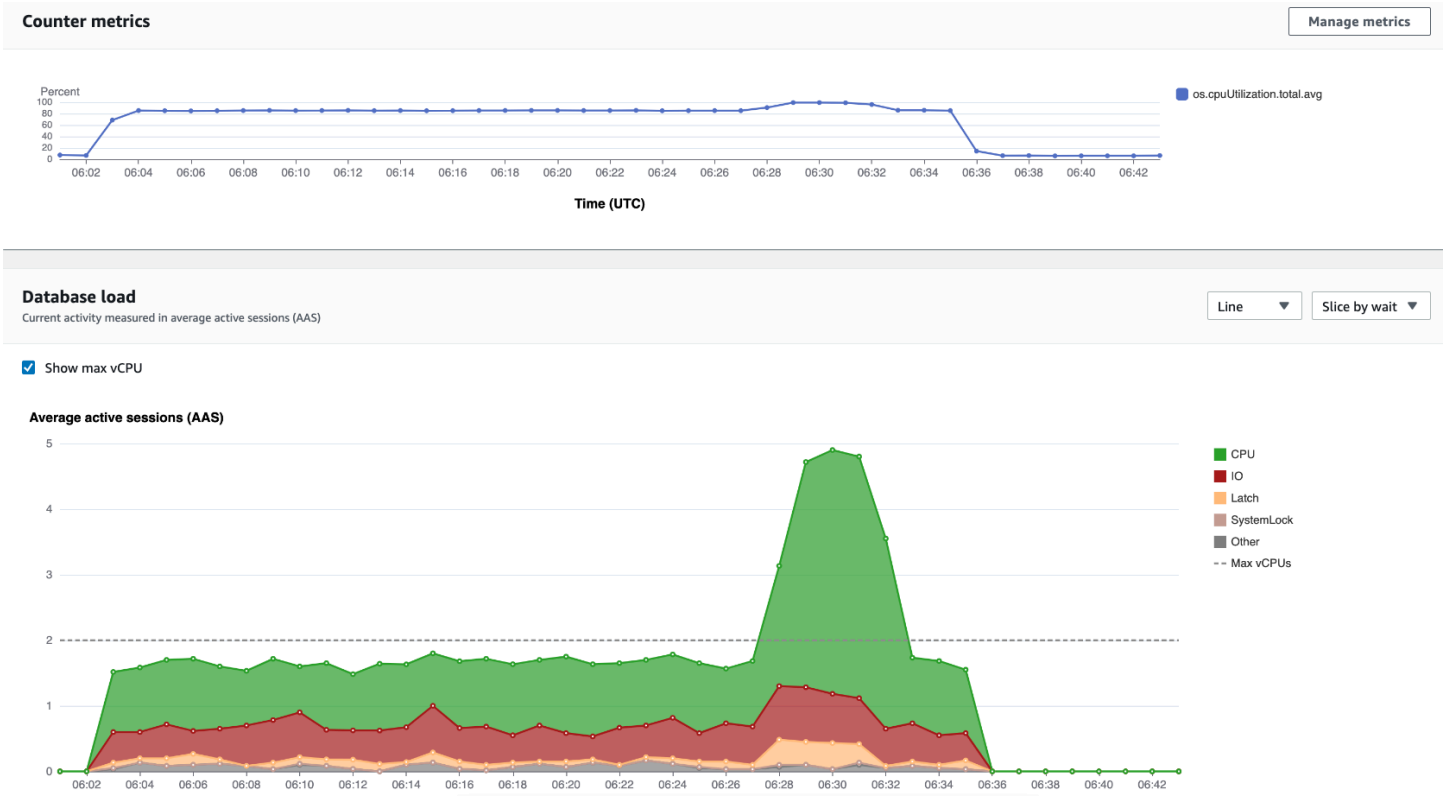
```
aws pi help
```

Jika Anda tidak memasang AWS CLI, lihat Memasang Antarmuka Baris Perintah AWS dalam Panduan Pengguna AWS CLI untuk informasi tentang pemasangan.

Mengambil metrik seri waktu

Operasi `GetResourceMetrics` mengambil satu atau lebih metrik seri waktu dari data Wawasan Kinerja. `GetResourceMetrics` memerlukan metrik dan periode waktu, dan mengembalikan respons dengan daftar poin data.

Misalnya, AWS Management Console menggunakan `GetResourceMetrics` untuk mengisi bagan Metrik Penghitung, dan dalam bagan Muatan Database seperti yang terlihat di gambar berikut.



Semua metrik yang dikembalikan oleh `GetResourceMetrics` adalah metrik seri waktu standar dengan pengecualian `db.load`. Metrik ini ditampilkan dalam bagan Muatan Database. Metrik

`db.load` berbeda dari metrik seri waktu lainnya karena Anda dapat membaginya menjadi subkomponen yang disebut dimensi. Di gambar sebelumnya, `db.load` dipecah dan dikelompokkan berdasarkan status tunggu yang membentuk `db.load`.

> **Note**
>
> `GetResourceMetrics` juga dapat mengembalikan metrik `db.sampleload`, tetapi metrik `db.load` sesuai dalam sebagian besar kasus.

Untuk informasi tentang metrik penghitung yang dikembalikan oleh `GetResourceMetrics`, lihat Performance Insights untuk metrik penghitung.

Perhitungan berikut didukung untuk metrik:

- Rata-rata – Nilai rata-rata untuk metrik selama periode waktu tertentu. Tambahkan `.avg` ke nama metrik.
- Minimum – Nilai minimum untuk metrik selama periode waktu tertentu. Tambahkan `.min` ke nama metrik.
- Maksimum – Nilai maksimum untuk metrik selama periode waktu tertentu. Tambahkan `.max` ke nama metrik.
- Jumlah – Jumlah nilai metrik selama periode waktu tertentu. Tambahkan `.sum` ke nama metrik.
- Jumlah sampel – Frekuensi pengumpulan metrik selama periode waktu tertentu. Tambahkan `.sample_count` ke nama metrik.

Misalnya, anggap bahwa metrik dikumpulkan selama 300 detik (5 menit), dan metrik dikumpulkan satu kali setiap menit. Nilai untuk setiap menit adalah 1, 2, 3, 4, dan 5. Dalam kasus ini, kalkulasi berikut dikembalikan:

- Rata-rata – 3
- Minimum – 1
- Maksimum – 5
- Jumlah – 15
- Jumlah sampel – 5

Untuk informasi tentang menggunakan `get-resource-metrics` AWS CLI perintah, lihat [get-resource-metrics](#).

Untuk opsi `--metric-queries`, tentukan satu atau lebih kueri yang Anda inginkan untuk mendapatkan hasil. Setiap kueri terdiri dari `Metric` wajib dan `GroupBy` opsional dan parameter `Filter`. Berikut ini adalah contoh spesifikasi opsi `--metric-queries`.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

Contoh AWS CLI untuk Wawasan Kinerja

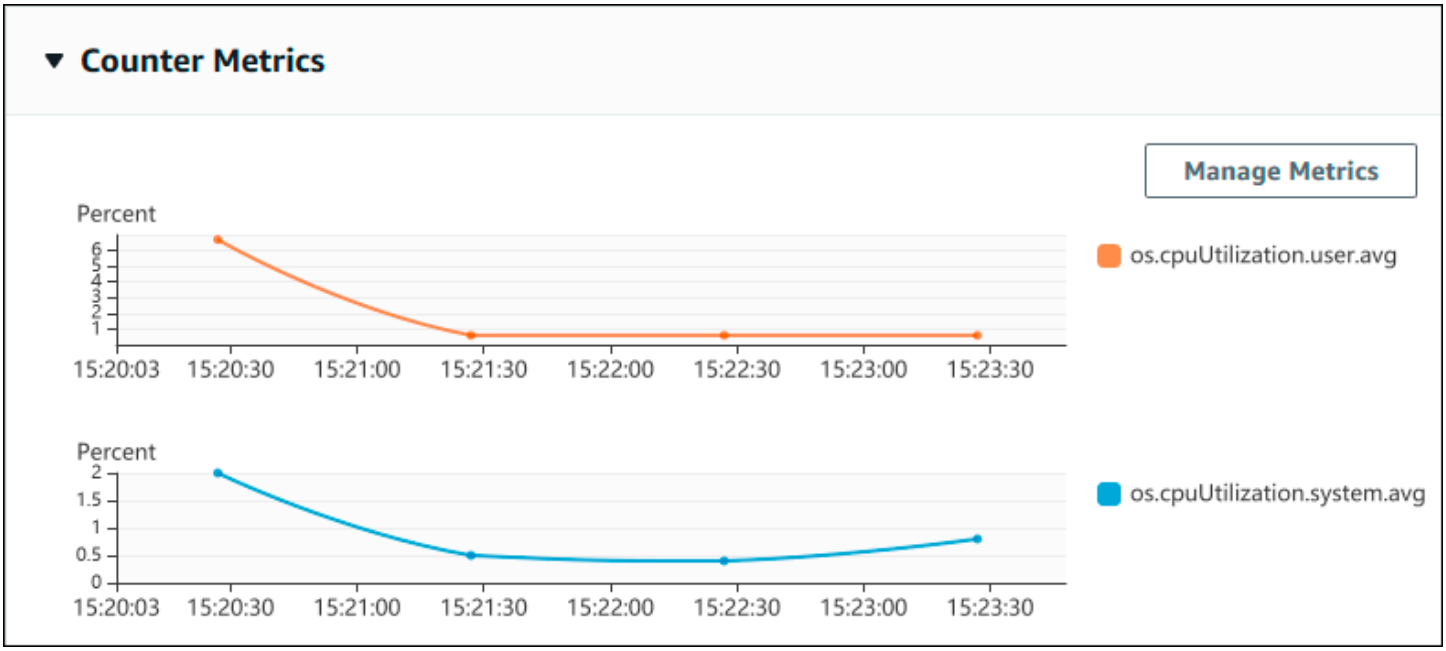
Contoh berikut menunjukkan cara menggunakan AWS CLI untuk Wawasan Kinerja.

Topik

- Mengambil metrik penghitung
- Mengambil rata-rata beban DB untuk status tunggu teratas
- Mengambil rata-rata beban DB untuk Query teratas
- Mengambil rata-rata beban DB yang disaring oleh Query

Mengambil metrik penghitung

Tangkapan layar berikut menunjukkan dua bagan metrik penghitung dalam AWS Management Console.



Contoh berikut menunjukkan cara mengumpulkan data yang sama yang digunakan oleh AWS Management Console untuk menghasilkan dua bagan metrik penghitung.

Untuk Linux, macOS, atau 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"}]'
```

Untuk 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"}]'
```

Anda juga dapat membuat perintah menjadi lebih mudah dibaca dengan menentukan file untuk opsi `--metrics-query`. Contoh berikut menggunakan file yang disebut query.json untuk opsi tersebut. File tersebut berisi konten berikut.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Jalankan perintah berikut untuk menggunakan file.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Contoh sebelumnya menentukan nilai-nilai berikut untuk opsi:

- `--service-type`- DOCDB untuk Amazon DocumentDB
- `--identifier` – ID sumber daya untuk instans DB
- `--start-time` dan `--end-time` – Nilai ISO 8601 `DateTime` untuk periode kueri, dengan berbagai format yang didukung

Ini akan meminta selama rentang waktu satu jam:

- `--period-in-seconds` – `60` untuk kueri per menit
- `--metric-queries` – Rangkaian dua kueri, masing-masing hanya untuk satu metrik.

  Nama metrik menggunakan titik untuk mengklasifikasikan metrik dalam kategori yang berguna, dengan elemen terakhir sebagai fungsi. Dalam contoh, fungsinya adalah `avg` untuk setiap kueri. Seperti halnya Amazon CloudWatch, fungsi yang didukung adalah `min`,`max`,`total`, dan`avg`.

Responsnya terlihat seperti berikut ini.

```
{
    "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
```

Respons memiliki `Identifier`, `AlignedStartTime`, dan `AlignedEndTime`. B nilai `--period-in-seconds` adalah `60`, waktu mulai dan selesai telah disesuaikan dengan menit. Jika `--period-in-seconds` adalah `3600`, waktu mulai dan selesai akan disesuaikan dengan jam.

`MetricList` dalam respons memiliki sejumlah entri, masing-masing dengan entri `Key` dan `DataPoints`. Masing-masing `DataPoint` memiliki `Timestamp` dan `Value`. Masing-masing daftar `Datapoints` memiliki 60 poin data karena kueri tersebut adalah untuk data per menit selama satu jam, dengan `Timestamp1/Minute1`, `Timestamp2/Minute2`, dan seterusnya, hingga `Timestamp60/Minute60`.

Karena kueri tersebut adalah untuk dua metrik penghitung yang berbeda, ada dua elemen dalam respons `MetricList`.

Mengambil rata-rata beban DB untuk status tunggu teratas

Contoh berikut adalah kueri yang sama yang digunakan oleh AWS Management Console untuk menghasilkan grafik garis area bertumpuk. Contoh ini mengambil `db.load.avg` untuk jam terakhir dengan beban dibagi menurut tujuh status tunggu teratas. Perintah ini sama dengan perintah dalam Mengambil metrik penghitung. Namun, file query.json berisi konten berikut.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 7 }
    }
]
```

Jalankan perintah berikut.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Contoh menentukan metrik `db.load.avg` dan a `GroupBy` dari tujuh status tunggu teratas. Untuk detail tentang nilai yang valid untuk contoh ini, lihat DimensionGroup dalam Referensi API Wawasan Kinerja.

Responsnya terlihat seperti berikut ini.

```
{
    "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
```

Dalam tanggapan ini, ada tiga entri dalam. `MetricList` Ada satu entri untuk total`db.load.avg`, dan tiga entri masing-masing untuk `db.load.avg` dibagi menurut salah satu dari tiga negara tunggu teratas. Karena ada dimensi pengelompokan (tidak seperti contoh pertama), harus ada satu kunci untuk setiap pengelompokan metrik. Tidak boleh hanya ada satu kunci untuk setiap metrik, seperti dalam kasus penggunaan metrik penghitung dasar.

Mengambil rata-rata beban DB untuk Query teratas

Contoh berikut dikelompokkan `db.wait_state` oleh 10 pernyataan kueri teratas. Ada dua kelompok berbeda untuk pernyataan kueri:

- `db.query`— Pernyataan kueri lengkap, seperti `{"find":"customers","filter": {"FirstName":"Jesse"},"sort":{"key":{"$numberInt":"1"}}}`
- `db.query_tokenized`— Pernyataan kueri tokenized, seperti `{"find":"customers","filter":{"FirstName":"?"},"sort":{"key": {"$numberInt":"?"}},"limit":{"$numberInt":"?"}}`

Saat menganalisis kinerja database, akan berguna untuk mempertimbangkan pernyataan kueri yang hanya berbeda dengan parameternya sebagai satu item logika. Jadi, Anda dapat menggunakan `db.query_tokenized` saat meminta. Namun, terutama ketika Anda tertarik`explain()`, terkadang lebih berguna untuk memeriksa pernyataan kueri lengkap dengan parameter. Ada hubungan orang tua-anak antara kueri tokenized dan full, dengan beberapa kueri lengkap (turunan) dikelompokkan di bawah kueri tokenized yang sama (induk).

Perintah dalam contoh ini terlihat seperti perintah di Mengambil rata-rata beban DB untuk status tunggu teratas. Namun, file query.json berisi konten berikut.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.query_tokenized", "Limit": 10 }
    }
]
```

Contoh berikut menggunakan `db.query_tokenized`.

Untuk Linux, macOS, atau 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
```

Untuk 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
```

Contoh ini menanyakan lebih dari 1 jam, dengan satu menit period-in-seconds.

Contoh menentukan metrik `db.load.avg` dan a `GroupBy` dari tujuh status tunggu teratas. Untuk detail tentang nilai yang valid untuk contoh ini, lihat DimensionGroup dalam Referensi API Wawasan Kinerja.

Responsnya terlihat seperti berikut ini.

```
{
    "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
```

Respons ini memiliki 11 entri dalam `MetricList` (total 1, 10 kueri tokenized teratas), dengan setiap entri memiliki 24 per jam. `DataPoints`

Untuk kueri tokenized, ada tiga entri di setiap daftar dimensi:

- `db.query_tokenized.statement`— Pernyataan kueri tokenized.
- `db.query_tokenized.db_id` — ID sintetis yang dihasilkan Performance Insights untuk Anda. Contoh ini mengembalikan ID sintetis `pi-1064184600`.
- `db.query_tokenized.id` – ID kueri di dalam Wawasan Kinerja.

  Di AWS Management Console, ID ini disebut sebagai Support ID. Dinamakan seperti ini karena ID adalah data yang dapat diperiksa oleh AWS Support untuk membantu memecahkan masalah dalam basis data Anda. AWS menangani keamanan dan privasi data Anda dengan sangat serius, dan hampir semua data disimpan terenkripsi dengan kunci utama pelanggan (CMK) AWS KMS. Oleh karena itu, tidak ada orang di dalam AWS yang dapat melihat data ini. Di contoh sebelumnya, baik `tokenized.statement` maupun `tokenized.db_id` disimpan dengan enkripsi. Jika Anda mengalami masalah dalam basis data, AWS Support dapat membantu Anda dengan merujuk Support ID.

Ketika melakukan kueri, mungkin lebih mudah untuk menentukan `Group` dalam `GroupBy`. Namun, untuk kontrol lebih detail atas data yang dikembalikan, tentukan daftar dimensi. Misalnya, jika yang dibutuhkan hanya `db.query_tokenized.statement`, atribut `Dimensions` dapat ditambahkan ke file query.json.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.query_tokenized",
            "Dimensions":["db.query_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Mengambil rata-rata beban DB yang disaring oleh Query

Kueri API terkait dalam contoh ini terlihat seperti perintah di Mengambil rata-rata beban DB untuk Query teratas. Namun, file query.json berisi konten berikut.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 5  },
```

```
        "Filter": { "db.query_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
]
```

Dalam tanggapan ini, semua nilai difilter sesuai dengan kontribusi kueri token AKIAIOSFODNN7EXAMPLE yang ditentukan dalam file query.json. Kunci juga mungkin mengikuti urutan yang berbeda dari kueri tanpa filter, karena ini adalah lima status tunggu teratas yang memengaruhi kueri yang difilter.

CloudWatch Metrik Amazon untuk Performance Insights

Performance Insights secara otomatis menerbitkan metrik ke Amazon. CloudWatch Data yang sama dapat ditanyakan dari Performance Insights, tetapi memiliki metrik CloudWatch memudahkan untuk menambahkan alarm. CloudWatch Ini juga memudahkan untuk menambahkan metrik ke Dasbor CloudWatch yang ada.

Metrik	Deskripsi
DBLoad	Jumlah sesi aktif untuk Amazon DocumentDB. Biasanya, Anda menginginkan data untuk jumlah rata-rata sesi aktif. Dalam Wawasan Kinerja, data ini diminta sebagai `db.load.avg` .
DBLoadCPU	Jumlah sesi aktif di mana tipe status tunggu adalah CPU. Di Performance Insights, data ini ditanyakan sebagai`db.load.avg` , difilter berdasarkan jenis status tunggu. `CPU`
LoadNonCPU DB	Jumlah sesi aktif di mana tipe status tunggu bukan CPU.

> **Note**
>
> Metrik ini dipublikasikan CloudWatch hanya jika ada beban pada instans DB.

Anda dapat memeriksa metrik ini menggunakan CloudWatch konsol, theAWS CLI, atau CloudWatch API.

Misalnya, Anda bisa mendapatkan statistik untuk `DBLoad` metrik dengan menjalankan get-metric-statisticsperintah.

```
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
```

Contoh ini menghasilkan output yang terlihat seperti berikut ini.

```
{
    "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"
}
```

Anda dapat menggunakan fungsi matematika `DB_PERF_INSIGHTS` metrik di CloudWatch konsol untuk menanyakan metrik penghitung Performance Insights Amazon DocumentDB. `DB_PERF_INSIGHTS`Fungsi ini juga mencakup `DBLoad` metrik pada interval sub-menit. Anda dapat mengatur CloudWatch alarm pada metrik ini. Untuk detail selengkapnya tentang cara membuat alarm, lihat Membuat alarm di metrik penghitung Performance Insights dari database. AWS

Untuk informasi selengkapnya CloudWatch, lihat Apa itu Amazon CloudWatch? di Panduan CloudWatch Pengguna Amazon.

Performance Insights untuk metrik penghitung

Metrik penghitung adalah metrik sistem operasi di dasbor Performance Insights. Untuk membantu mengidentifikasi dan menganalisis masalah kinerja, Anda dapat mengkorelasikan metrik penghitung dengan beban DB.

Penghitung sistem operasi Wawasan Kinerja

Penghitung sistem operasi berikut tersedia dengan DocumentDB Performance Insights.

Penghitung	Jenis	Metrik
aktif	memori	os.memory.active
buffer	memori	os.memory.buffers
dalam cache	memori	os.memory.cached
kotor	memori	os.memory.dirty

Penghitung	Jenis	Metrik
kosong	memori	os.memory.free
tidak aktif	memori	os.memory.inactive
dipetakan	memori	os.memory.mapped
pageTables	memori	os.memory.pageTables
lempengan	memori	os.memory.slab
total	memori	os.memory.total
tulis balik	memori	os.memory.writeback
diam	cpuUtilization	os.cpuUtilization.idle
sistem	cpuUtilization	os.cpuUtilization.system
total	cpuUtilization	os.cpuUtilization.total
pengguna	cpuUtilization	os.cpuUtilization.user
tunggu	cpuUtilization	os.cpuUtilization.wait
satu	loadAverageMinute	os. loadAverageMinute.satu
lima belas	loadAverageMinute	os. loadAverageMinute.lima belas
lima	loadAverageMinute	os. loadAverageMinute.lima
dalam cache	ganti	os.swap.cached
kosong	swap	os.swap.free
masuk	swap	os.swap.in
keluar	swap	os.swap.out
total	swap	os.swap.total

Penghitung	Jenis	Metrik
rx	jaringan	os.network.rx
tx	jaringan	os.network.tx
numVCPUs	umum	os.general.numVCPUs

Integrasi nol-ETL dengan Amazon Service OpenSearch

Topik

- OpenSearch Layanan Amazon sebagai tujuan
- Batasan

OpenSearch Layanan Amazon sebagai tujuan

OpenSearch Integrasi layanan dengan Amazon DocumentDB memungkinkan Anda melakukan streaming beban penuh dan mengubah peristiwa data ke domain. OpenSearch Infrastruktur konsumsi di-host sebagai saluran pipa OpenSearch konsumsi dan menyediakan mekanisme latensi rendah skala tinggi untuk terus mengalirkan data dari koleksi Amazon DocumentDB.

Selama pemuatan penuh, integrasi nol-ETL pertama-tama mengekstrak data beban penuh historis untuk OpenSearch menggunakan pipa konsumsi. Setelah data beban penuh tertelan, saluran pipa OpenSearch konsumsi akan mulai membaca data dari aliran perubahan Amazon DocumentDB dan akhirnya mengejar ketinggalan untuk mempertahankan konsistensi data mendekati waktu nyata antara Amazon DocumentDB dan. OpenSearch OpenSearch menyimpan dokumen dalam indeks. Data yang masuk dari koleksi Amazon DocumentDB dapat dikirim ke salah satu indeks atau dapat dipartisi ke dalam indeks yang berbeda. Saluran pipa konsumsi akan menyinkronkan semua acara pembuatan, pembaruan, dan penghapusan dalam koleksi Amazon DocumentDB sebagai pembuatan, pembaruan, dan penghapusan OpenSearch dokumen yang sesuai untuk menjaga kedua sistem data tetap sinkron. Saluran pipa konsumsi dapat dikonfigurasi untuk membaca data dari satu koleksi dan menulis ke satu indeks atau membaca data dari satu koleksi dan merutekan secara kondisional ke beberapa indeks.

Saluran pipa konsumsi dapat dikonfigurasi untuk mengalirkan data dari Amazon DocumentDB ke Amazon Service menggunakan: OpenSearch

- Beban penuh saja
- Streaming perubahan peristiwa aliran dari Amazon DocumentDB tanpa beban penuh
- Beban penuh diikuti oleh aliran perubahan dari Amazon DocumentDB

Untuk mengatur saluran konsumsi Anda, lakukan langkah-langkah berikut:

Langkah 1: Buat domain OpenSearch Layanan Amazon atau koleksi OpenSearch tanpa server

Diperlukan koleksi OpenSearch Layanan Amazon dengan izin yang sesuai untuk membaca data. Lihat Memulai OpenSearch Layanan Amazon atau Memulai Amazon OpenSearch Tanpa Server di Panduan Pengembang OpenSearch Layanan Amazon untuk membuat koleksi. Lihat Amazon OpenSearch Ingestion di Panduan Pengembang OpenSearch Layanan Amazon untuk membuat peran AIM dengan izin yang benar untuk mengakses data tulis ke koleksi atau domain.

Langkah 2: Aktifkan aliran perubahan di cluster Amazon DocumentDB

Pastikan aliran perubahan diaktifkan pada koleksi yang diperlukan di klaster Amazon DocumentDB. Lihat Menggunakan Aliran Perubahan dengan Amazon DocumentDB untuk informasi lebih lanjut.

Langkah 3: Siapkan peran pipeline dengan izin untuk menulis ke bucket Amazon S3 dan domain atau koleksi tujuan

Setelah koleksi Amazon DocumentDB dibuat dan mengubah aliran diaktifkan, siapkan peran pipeline yang ingin digunakan dalam konfigurasi pipeline, dan tambahkan izin berikut dalam peran:

```
{
    "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/*"
            ]
        }
    ]
}
```

Agar OpenSearch pipeline dapat menulis data ke OpenSearch domain, domain harus memiliki kebijakan akses tingkat domain yang memungkinkan peran pipeline sts_role_arn untuk mengaksesnya. Contoh kebijakan akses domain berikut memungkinkan peran pipeline bernama`pipeline-role`, yang Anda buat pada langkah sebelumnya, untuk menulis data ke domain bernama`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}/*"
    }
  ]
}
```

Langkah 4: Tambahkan izin yang diperlukan pada peran pipeline untuk membuat X-ENI

```
{
    "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" }
            }
        }
    ]
}
```

Langkah 5: Buat pipa

Konfigurasikan saluran OpenSearch konsumsi yang menentukan Amazon DocumentDB sebagai sumbernya. Konfigurasi pipeline sampel ini mengasumsikan penggunaan mekanisme pengambilan aliran perubahan. Lihat Menggunakan pipeline OpenSearch Ingestion dengan Amazon DocumentDB di Panduan Pengembang Layanan OpenSearch Amazon untuk informasi selengkapnya.

Batasan

Batasan berikut berlaku untuk integrasi Amazon OpenSearch DocumentDB:

- Hanya satu koleksi Amazon DocumentDB sebagai sumber per pipeline yang didukung.
- Konsumsi data lintas wilayah tidak didukung. Cluster OpenSearch dan domain Amazon DocumentDB Anda harus berada di wilayah yang sama. AWS
- Penyerapan data lintas akun tidak didukung. Cluster Amazon DocumentDB OpenSearch dan pipeline konsumsi Anda harus berada di akun yang sama. AWS

- Cluster elastis Amazon DocumentDB tidak didukung. Hanya cluster berbasis instans Amazon DocumentDB yang didukung.
- Pastikan klaster Amazon DocumentDB mengaktifkan autentikasi menggunakan rahasia. AWS AWS rahasia adalah satu-satunya mekanisme otentikasi yang didukung.
- Konfigurasi pipeline yang ada tidak dapat diperbarui untuk menyerap data dari database yang berbeda dan/atau koleksi yang berbeda. Untuk memperbarui database dan/atau nama koleksi pipeline, Anda harus membuat pipeline baru.

Melaksanakan Pengembangan dengan Amazon DocumentDB

Bagian ini mencakup pengembangan menggunakan Amazon DocumentDB (dengan kompatibilitas MongoDB).

Topik

- Menghubungkan Secara Terprogram ke Amazon DocumentDB
- Menggunakan Aliran Perubahan dengan Amazon DocumentDB
- MenggunakanAWS Lambdadengan Change Streams
- Menggunakan validasi skema JSON
- Menghubungkan ke Amazon DocumentDB sebagai Set Replika
- Menghubungkan ke Klaster Amazon DocumentDB dari Luar Amazon VPC
- Menghubungkan ke cluster Amazon DocumentDB dari Studio 3T
- Connect ke Amazon DocumentDB menggunakan DataGrip
- Connect menggunakan Amazon EC2
- Connect menggunakan driver Amazon DocumentDB JDBC
- Connect menggunakan driver ODBC Amazon DocumentDB

Menghubungkan Secara Terprogram ke Amazon DocumentDB

Bagian ini berisi contoh kode yang menunjukkan cara menghubungkan ke Amazon DocumentDB (dengan kompatibilitas MongoDB) menggunakan beberapa bahasa yang berbeda. Contoh dipisahkan menjadi dua bagian berdasarkan apakah Anda terhubung ke klaster yang memiliki Keamanan Lapisan Pengangkutan (TLS) diaktifkan atau dinonaktifkan. Secara default, TLS diaktifkan di klaster Amazon DocumentDB. Namun, Anda dapat menonaktifkan TLS jika Anda mau. Untuk informasi selengkapnya, lihat Mengenkripsi Data dalam Transit.

Jika Anda mencoba untuk terhubung ke Amazon DocumentDB Anda dari luar VPC di mana klaster Anda berada, silakan lihat Menghubungkan ke Klaster Amazon DocumentDB dari Luar Amazon VPC.

Sebelum Anda terhubung ke klaster Anda, Anda harus tahu apakah TLS diaktifkan di klaster. Bagian berikutnya menunjukkan kepada Anda bagaimana untuk menentukan nilai parameter `tls` klaster

Anda menggunakan AWS Management Console atau AWS CLI. Setelah itu, Anda dapat melanjutkan dengan menemukan dan menerapkan contoh kode yang sesuai.

Topik

- Menentukan Nilai Parameter tls Anda
- Menghubungkan dengan TLS yang Diaktifkan
- Menghubungkan dengan TLS yang Dinonaktifkan

Menentukan Nilai Parameter `tls` Anda

Menentukan apakah klaster Anda mengaktifkan TLS adalah proses dua langkah yang dapat Anda lakukan menggunakan atau. AWS Management Console AWS CLI

1. Tentukan grup parameter mana yang mengatur cluster Anda.

   Using the AWS Management Console

   1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
   2. Pada panel navigasi sebelah kiri, pilih Klaster.
   3. Dalam daftar klaster, pilih nama klaster Anda.
   4. Halaman yang dihasilkan menunjukkan detail klaster yang Anda pilih. Gulir ke bawah ke Detail klaster. Di bagian bawah bagian tersebut, temukan nama grup parameter di bawah Grup parameter klaster.

   Using the AWS CLI

   AWS CLI Kode berikut menentukan parameter mana yang mengatur cluster Anda. Pastikan Anda mengganti `sample-cluster` dengan nama klaster Anda.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
   ```

   Keluaran dari operasi ini terlihat seperti berikut ini:

   ```
   [
   ```

```
        [
            "sample-cluster",
            "sample-parameter-group"
        ]
]
```

2. Tentukan nilai **`tls`** parameter dalam grup parameter cluster Anda.

   Using the AWS Management Console

   1. Di panel navigasi, pilih Grup parameter.
   2. Di jendela grup parameter Cluster, pilih grup parameter cluster Anda.
   3. Halaman yang dihasilkan menunjukkan parameter grup parameter klaster Anda. Anda dapat melihat nilai dari parameter `tls` di sini. Untuk informasi tentang memodifikasi parameter ini, lihat Memodifikasi grup parameter cluster Amazon DocumentDB.

   Using the AWS CLI

   Anda dapat menggunakan `describe-db-cluster-parameters` AWS CLI perintah untuk melihat detail parameter dalam grup parameter cluster Anda.

   - **`--describe-db-cluster-parameters`** — Untuk membuat daftar semua parameter di dalam grup parameter dan nilainya.
     - **`--db-cluster-parameter-group name`** — Diperlukan. Nama grup parameter klaster Anda.

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Keluaran dari operasi ini terlihat seperti berikut ini:

   ```
   {
           "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 mendukung FIPS 140-3 endpoint dimulai dengan Amazon DocumentDB 5.0 (versi engine 3.0.3727) cluster di wilayah ini: ca-central-1, us-west-2, us-east-1, us-east-1, us-east-2, us-east-2, -1. us-gov-east us-gov-west

Setelah menentukan nilai dari parameter `tls` Anda, lanjutkan menghubungkan ke klaster Anda dengan menggunakan salah satu contoh kode di bagian berikut.

- Menghubungkan dengan TLS yang Diaktifkan
- Menghubungkan dengan TLS yang Dinonaktifkan

Menghubungkan dengan TLS yang Diaktifkan

Untuk melihat contoh kode guna menghubungkan secara terprogram ke klaster Amazon DocumentDB dengan TLS yang diaktifkan, pilih tab yang sesuai untuk bahasa yang ingin Anda gunakan.

Untuk mengenkripsi data dalam perjalanan, unduh kunci publik untuk Amazon DocumentDB bernama `global-bundle.pem` menggunakan operasi berikut.

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Jika aplikasi Anda ada di Micro Windows dan memerlukan file PKCS7, Anda dapat mengunduh paket sertifikat PKCS7. Paket ini berisi sertifikat perantara dan root di https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b.

Python

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Python ketika TLS diaktifkan.

```
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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Node.js ketika TLS diaktifkan.

```
var MongoClient = require('mongodb').MongoClient

//Create a MongoDB client, open a connection to DocDB; as a replica set,
//  and specify the read preference as secondary preferred

var client = MongoClient.connect(
'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
tls=true&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  tlsCAFile: `global-bundle.pem` //Specify the DocDB; cert
},
function(err, client) {
    if(err)
        throw err;

    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'DocDB;'}, function(err, result){
        //Print the result to the screen
        console.log(result);

        //Close the connection
        client.close()
      });
   });
});
```

PHP

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan PHP ketika TLS diaktifkan.

```
<?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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Go ketika TLS diaktifkan.

> **Note**
>
> Pada versi 1.2.1, Driver MongoDB Go hanya akan menggunakan sertifikat server CA pertama yang ditemukan di `sslcertificateauthorityfile`. Contoh kode di bawah ini mengatasi batasan ini dengan menambahkan semua sertifikat server secara manual yang ditemukan di `sslcertificateauthorityfile` ke konfigurasi TLS kustom yang digunakan selama pembuatan klien.

```
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

Saat menyambungkan ke klaster Amazon DocumentDB yang mendukung TLS dari aplikasi Java, program Anda harus menggunakan file otoritas sertifikat (CA) AWS yang disediakan untuk memvalidasi koneksi. Untuk menggunakan sertifikat CA Amazon RDS, lakukan hal berikut:

1. Unduh file CA Amazon RDS dari https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem .
2. Buat penyimpanan kepercayaan dengan sertifikat CA yang terdapat dalam file dengan melakukan perintah berikut. Pastikan untuk mengubah *<truststorePassword>* ke sesuatu yang lain. Jika Anda mengakses penyimpanan kepercayaan yang berisi sertifikat CA lama (`rds-ca-2015-root.pem`) dan sertifikat CA baru (`rds-ca-2019-root.pem`), Anda dapat mengimpor paket sertifikat ke dalam penyimpanan kepercayaan.

   Berikut adalah contoh skrip shell yang mengimpor paket sertifikat ke trust store di sistem operasi Linux. Dalam contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri. Terutama, di mana pun direktori contoh "*mydir*" berada di skrip, gantilah dengan direktori yang Anda buat untuk tugas ini.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
 {split_after=1}{print > "rds-ca-" n ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
```

```
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Berikut adalah contoh skrip shell yang mengimpor paket sertifikat ke trust store di sistem operasi macOS.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
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. Gunakan `keystore` di program Anda dengan mengatur properti sistem berikut di aplikasi Anda sebelum membuat koneksi ke klaster Amazon DocumentDB.

```
javax.net.ssl.trustStore: <truststore>
javax.net.ssl.trustStorePassword: <truststorePassword>
```

4. Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Java ketika TLS diaktifkan.

```
package com.example.documentdb;

import com.mongodb.client.*;
import org.bson.Document;

public final class Test {
    private Test() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
ssl=true&replicaSet=rs0&readpreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        String truststore = "<truststore>";
        String truststorePassword = "<truststorePassword>";

        System.setProperty("javax.net.ssl.trustStore", truststore);
        System.setProperty("javax.net.ssl.trustStorePassword",
 truststorePassword);

        MongoClient mongoClient = MongoClients.create(connectionString);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection =
 testDB.getCollection("sample-collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
```

```
            }
        } finally {
            cursor.close();
        }

    }
}
```

C# / .NET

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan C# / .NET ketika TLS diaktifkan.

```
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);
        }
    }
}
```

mongo shell

Kode berikut menunjukkan cara menghubungkan ke dan membuat kueri Amazon DocumentDB menggunakan shell mongo ketika TLS diaktifkan.

1. Terhubung ke Amazon DocumentDB dengan shell mongo. Jika Anda menggunakan versi shell mongo lebih awal dari 4.2, gunakan kode berikut untuk menghubungkan.

   ```
   mongo --ssl --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --
   sslCAFile global-bundle.pem --username <sample-user> --password <password>
   ```

Jika Anda menggunakan versi yang sama dengan atau lebih besar dari 4.2, gunakan kode berikut untuk menghubungkan. Penulisan yang dapat dicoba ulang tidak didukung di AWS DocumentDB. Pengecualian: Jika Anda menggunakan mongo shell, jangan sertakan `retryWrites=false` perintah dalam string kode apa pun. Secara default, penulisan yang dapat dicoba ulang dinonaktifkan. Termasuk `retryWrites=false` dapat menyebabkan kegagalan dalam perintah baca normal.

```
mongo --tls --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --
tlsCAFile global-bundle.pem --username <sample-user> --password <password>
```

2. Masukkan dokumen tunggal.

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. Temukan dokumen yang sebelumnya dimasukkan.

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB dengan R menggunakan mongolite (https://jeroen.github.io/mongolite/) ketika TLS diaktifkan.

```
#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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB dengan Ruby ketika TLS diaktifkan.

```
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
```

Menghubungkan dengan TLS yang Dinonaktifkan

Untuk melihat contoh kode guna menghubungkan secara terprogram ke klaster Amazon DocumentDB dengan TLS yang dinonaktifkan, pilih tab untuk bahasa yang ingin Anda gunakan.

Python

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Python ketika TLS dinonaktifkan.

```
## 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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Node.js ketika TLS dinonaktifkan.

```
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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan PHP ketika TLS dinonaktifkan.

```
<?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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Go ketika TLS dinonaktifkan.

```
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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan Java ketika TLS dinonaktifkan.

```
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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB menggunakan C# / .NET ketika TLS dinonaktifkan.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace CSharpSample
{
   class Program
    {
       static void Main(string[] args)
        {
           string template = "mongodb://{0}:{1}@{2}/sampledatabase?
replicaSet=rs0&readpreference={3}";
           string username = "<sample-user>";
           string password = "<password>";
           string clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
           string readPreference = "secondaryPreferred";
           string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);

           var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
           var client = new MongoClient(settings);

           var database = client.GetDatabase("sampledatabase");
           var collection =
 database.GetCollection<BsonDocument>("samplecollection");
           var docToInsert = new BsonDocument { { "pi", 3.14159 } };
            collection.InsertOne(docToInsert);
       }
    }
}
```

mongo shell

Kode berikut menunjukkan cara menghubungkan ke dan membuat kueri Amazon DocumentDB menggunakan shell mongo ketika TLS dinonaktifkan.

1. Terhubung ke Amazon DocumentDB dengan shell mongo.

```
mongo --host mycluster.node.us-east-1.docdb.amazonaws.com:27017 --
username <sample-user> --password <password>
```

2. Masukkan dokumen tunggal.

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. Temukan dokumen yang sebelumnya dimasukkan.

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB dengan R menggunakan mongolite (https://jeroen.github.io/mongolite/) ketika TLS dinonaktifkan.

```
#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

Kode berikut menunjukkan cara menghubungkan ke Amazon DocumentDB dengan Ruby ketika TLS dinonaktifkan.

```
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
```

Menggunakan Aliran Perubahan dengan Amazon DocumentDB

Fitur aliran perubahan di Amazon DocumentDB (dengan kompatibilitas MongoDB) menyediakan urutan waktu peristiwa perubahan yang terjadi dalam koleksi klaster Anda. Anda dapat membaca peristiwa dari aliran perubahan untuk menerapkan banyak kasus penggunaan yang berbeda, termasuk yang berikut:

- Notifikasi perubahan
- Pencarian teks lengkap dengan Amazon OpenSearch Layanan (OpenSearch Layanan)
- Analitik dengan Amazon Redshift

Aplikasi dapat menggunakan aliran perubahan untuk berlangganan perubahan data pada koleksi individu. Peristiwa aliran perubahan diurutkan saat terjadi di klaster dan disimpan selama 3 jam (secara default) setelah peristiwa dicatat. Periode retensi dapat diperpanjang hingga 7 hari menggunakan parameter `change_stream_log_retention_duration`. Untuk mengubah periode retensi aliran perubahan, lihat Memodifikasi Durasi Retensi Log Aliran Perubahan.

Topik

- Operasi yang Didukung
- Penagihan
- Keterbatasan:
- Mengaktifkan Aliran Perubahan
- Contoh: Menggunakan Aliran Perubahan dengan Python
- Pencarian Dokumen Lengkap
- Melanjutkan Aliran Perubahan
- Melanjutkan Aliran Perubahan dengan startAtOperationTime
- Transaksi di aliran perubahan
- Memodifikasi Durasi Retensi Log Aliran Perubahan

Operasi yang Didukung

Amazon DocumentDB mendukung operasi berikut untuk aliran perubahan:

- Semua peristiwa perubahan didukung di API `db.collection.watch()`, `db.watch()` dan `client.watch()` MongoDB.
- Pencarian dokumen lengkap untuk pembaruan.
- Tahap agregasi: `$match`, `$project`, `$redact`, serta `$addFields` dan `$replaceRoot`.
- Melanjutkan aliran perubahan dari token resume
- Melanjutkan aliran perubahan dari stempel waktu menggunakan `startAtOperation` (berlaku untuk Amazon DocumentDB v4.0+)

Penagihan

Fitur aliran perubahan Amazon DocumentDB dinonaktifkan secara default dan tidak dikenakan biaya tambahan apa pun hingga fitur tersebut diaktifkan. Menggunakan aliran perubahan dalam

klaster menimbulkan tambahan IO baca dan tulis serta tambahan biaya penyimpanan. Anda dapat menggunakan operasi API `modifyChangeStreams` untuk mengaktifkan fitur ini untuk klaster Anda. Untuk informasi selengkapnya tentang penentuan harga, lihat Penentuan harga Amazon DocumentDB.

Keterbatasan:

Aliran perubahan memiliki batasan berikut di Amazon DocumentDB:

- Aliran perubahan hanya dapat dibuka dari koneksi ke instans primer dari klaster Amazon DocumentDB. Membaca dari aliran perubahan pada instans replika saat ini tidak didukung. Ketika memanggil operasi API `watch()`, Anda harus menentukan preferensi baca **`primary`** untuk memastikan bahwa semua pembacaan diarahkan ke instans primer (lihat bagian Contoh).
- Peristiwa yang ditulis ke aliran perubahan untuk koleksi tersedia hingga 7 hari (defaultnya adalah 3 jam). Data aliran perubahan dihapus setelah jendela durasi retensi log, meskipun tidak ada perubahan baru yang terjadi.
- Operasi tulis yang berjalan lama pada koleksi seperti `updateMany` atau `deleteMany` dapat menghentikan sementara penulisan peristiwa aliran perubahan hingga operasi tulis yang berjalan lama selesai.
- Amazon DocumentDB tidak mendukung log operasi MongoDB (`oplog`).
- Dengan Amazon DocumentDB, Anda harus secara eksplisit mengaktifkan aliran perubahan pada koleksi tertentu.
- Jika ukuran total peristiwa aliran perubahan (termasuk data perubahan dan dokumen lengkap, jika diminta) lebih besar dari `16 MB`, klien akan mengalami kegagalan baca pada aliran perubahan.
- Driver Ruby saat ini tidak didukung ketika menggunakan `db.watch()` dan `client.watch()` dengan Amazon DocumentDB v3.6.

Mengaktifkan Aliran Perubahan

Anda dapat mengaktifkan aliran perubahan Amazon DocumentDB untuk semua koleksi dalam basis data tertentu, atau hanya untuk koleksi yang dipilih. Berikut ini adalah contoh cara mengaktifkan aliran perubahan untuk kasus penggunaan yang berbeda menggunakan shell mongo. String kosong diperlakukan sebagai wildcard ketika menentukan nama basis data dan koleksi.

```
//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});
```

Aliran perubahan akan diaktifkan untuk koleksi jika apa pun dari berikut ini benar:

- Baik basis data dan koleksi diaktifkan secara eksplisit.
- Basis data yang berisi koleksi diaktifkan.
- Semua basis data diaktifkan.

Menjatuhkan koleksi dari basis data tidak menonaktifkan aliran perubahan untuk koleksi tersebut jika basis data induk juga mengaktifkan aliran perubahan, atau jika semua basis data di klaster diaktifkan. Jika koleksi baru dibuat dengan nama yang sama dengan koleksi yang dihapus, aliran perubahan akan diaktifkan untuk koleksi tersebut.

Anda dapat membuat daftar semua aliran perubahan yang diaktifkan klaster Anda dengan menggunakan tahap alur agregasi `$listChangeStreams`. Semua tahapan agregasi yang didukung oleh Amazon DocumentDB dapat digunakan dalam alur untuk pemrosesan tambahan. Jika koleksi yang sebelumnya diaktifkan telah dinonaktifkan, koleksi tersebut tidak akan muncul di output `$listChangeStreams`.

```
//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:{}}));
```

Contoh: Menggunakan Aliran Perubahan dengan Python

Berikut ini adalah contoh penggunaan aliran perubahan Amazon DocumentDB dengan Python di tingkat koleksi.

```
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': '015daf99d400000001010000000100009025'},
'clusterTime': Timestamp(1571789268, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 2}}}
"""
```

Berikut ini adalah contoh penggunaan aliran perubahan Amazon DocumentDB dengan Python di tingkat basis data.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']
#Create a stream object
stream = db.watch()
coll = db.get_collection('foo')
#Write a new document to the collection foo to generate a change event
coll.insert_one({'x': 1})

#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""
#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
Expected Output:
None
"""

coll = db.get_collection('foo1')

#Write a new document to another collection to generate a change event
coll.insert_one({'x': 1})
print(stream.try_next())
```

```
"""
Expected Output: Since the change stream cursor was the database level you can see
 change events from different collections in the same database
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo1', 'db': 'bar'},
'operationType': 'insert'}
"""
```

Pencarian Dokumen Lengkap

Peristiwa perubahan pembaruan tidak menyertakan dokumen lengkap; itu mencakup hanya perubahan yang telah dibuat. Jika kasus penggunaan Anda memerlukan dokumen lengkap yang terpengaruh oleh pembaruan, Anda dapat mengaktifkan pencarian dokumen lengkap saat membuka aliran.

Dokumen `fullDocument` untuk peristiwa aliran perubahan pembaruan mewakili versi terbaru dari dokumen yang diperbarui pada saat pencarian dokumen. Jika terjadi perubahan antara operasi pembaruan dan pencarian `fullDocument`, dokumen `fullDocument` mungkin tidak mewakili status dokumen pada waktu pembaruan.

```
#Create a stream object with update lookup enabled
stream = coll.watch(full_document='updateLookup')

#Generate a new change event by updating a document
result = coll.update_one({'x': 2}, {'$set': {'x': 3}})

stream.try_next()

#Output:
{'_id': {'_data': '015daf9b7c000000010100000100009025'},
'clusterTime': Timestamp(1571789692, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'fullDocument': {'_id': ObjectId('5daf9502ea258751778163d7'), 'x': 3},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 3}}}
```

Melanjutkan Aliran Perubahan

Anda dapat melanjutkan aliran perubahan nanti dengan menggunakan token resume, yang sama dengan bidang `_id` dari dokumen peristiwa perubahan yang terakhir diambil.

```
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}}
"""
```

Melanjutkan Aliran Perubahan dengan **`startAtOperationTime`**

Anda dapat melanjutkan aliran perubahan nanti dari stempel waktu tertentu dengan menggunakan `startAtOperationTime`.

> **Note**
>
> Kemampuan untuk menggunakan `startAtOperationTime` tersedia di Amazon DocumentDB 4.0+. Saat menggunakan `startAtOperationTime`, kursor aliran perubahan hanya akan mengembalikan perubahan yang terjadi pada atau setelah Stempel Waktu yang ditentukan. Perintah `startAtOperationTime` dan `resumeAfter` saling eksklusif dan karenanya tidak dapat digunakan bersama.

```
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}}
"""
```

Transaksi di aliran perubahan

Peristiwa aliran perubahan tidak akan berisi peristiwa dari transaksi yang tidak dikomit dan/atau dibatalkan. Misalnya, jika Anda memulai transaksi dengan satu operasi `INSERT` dan satu operasi `UPDATE` dan. Jika operasi `INSERT` Anda berhasil, tetapi operasi `UPDATE` gagal, transaksi akan digulung kembali. Karena transaksi ini digulung kembali, aliran perubahan Anda tidak akan berisi peristiwa apa pun untuk transaksi ini.

Memodifikasi Durasi Retensi Log Aliran Perubahan

Anda dapat mengubah durasi retensi log aliran perubahan menjadi antara 1 jam dan 7 hari menggunakan AWS Management Console atau AWS CLI.

Using the AWS Management Console

Untuk memodifikasi durasi retensi log aliran perubahan

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Grup Parameter .

> **Tip**
>
> Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

3. Di panel Grup parameter, pilih grup parameter klaster yang terkait dengan klaster Anda. Untuk mengidentifikasi grup parameter klaster yang terkait dengan klaster Anda, lihat Menentukan grup parameter cluster Amazon DocumentDB.
4. Halaman yang dihasilkan menunjukkan parameter dan detailnya yang sesuai untuk grup parameter klaster Anda. Pilih parameter `change_stream_log_retention_duration`.
5. Di kanan atas halaman, pilih Edit untuk mengubah nilai parameter. The`change_stream_log_retention_duration`parameter dapat dimodifikasi menjadi antara 1 jam dan 7 hari.
6. Buat perubahan Anda, lalu pilih Modifikasi parameter klaster untuk menyimpan perubahan. Untuk membuang perubahan Anda, pilih Batal.

Using the AWS CLI

Untuk mengubah parameter `change_stream_log_retention_duration` grup parameter klaster Anda, gunakan operasi `modify-db-cluster-parameter-group` dengan parameter berikut:

- **`--db-cluster-parameter-group-name`** — Diperlukan. Nama grup parameter klaster yang Anda modifikasi. Untuk mengidentifikasi grup parameter klaster yang terkait dengan klaster Anda, lihat Menentukan grup parameter cluster Amazon DocumentDB.
- **`--parameters`** — Diperlukan. Parameter yang Anda modifikasi. Setiap entri parameter harus menyertakan hal berikut:
  - **`ParameterName`** — Nama parameter yang Anda modifikasi. Dalam hal ini, itu adalah `change_stream_log_retention_duration`
  - **`ParameterValue`** — Nilai baru untuk parameter ini.
  - **`ApplyMethod`** — Bagaimana Anda ingin perubahan pada parameter ini diterapkan. Nilai yang diizinkan adalah `immediate` dan `pending-reboot`.

> **Note**
>
> Parameter dengan `ApplyType` dari `static` harus memiliki `ApplyMethod` dari `pending-reboot`.

1. Untuk mengubah nilai parameter `change_stream_log_retention_duration`, jalankan perintah berikut dan ganti `parameter-value` dengan nilai yang ingin Anda modifikasi parameternya.

   Untuk Linux, macOS, atau 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"
   ```

   Untuk 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"
   ```

   Output dari operasi ini terlihat seperti berikut (format JSON).

   ```
   {
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
   ```

2. Tunggu paling tidak 5 menit.
3. Daftar nilai parameter dari `sample-parameter-group` untuk memastikan bahwa perubahan Anda telah dibuat.

   Untuk Linux, macOS, atau Unix:

   ```
   aws docdb describe-db-cluster-parameters \
   ```

```
    --db-cluster-parameter-group-name sample-parameter-group
```

Untuk Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "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**
>
> Mengubah retensi log aliran tidak akan menghapus log yang lebih lama dari yang dikonfigurasi`change_stream_log_retention_duration`nilai sampai ukuran log lebih besar dari (>) 51.200MB.

MenggunakanAWS Lambdadengan Change Streams

Amazon DocumentDB terintegrasi denganAWS Lambdasehingga Anda dapat menggunakan fungsi Lambda untuk memproses catatan dalam aliran perubahan. Pemetaan sumber peristiwa Lambda adalah sumber daya yang dapat digunakan untuk menjalankan fungsi Lambda untuk memproses peristiwa Amazon DocumentDB yang tidak langsung memanggil Lambda. Dengan aliran perubahan Amazon DocumentDB sebagai sumber peristiwa, Anda dapat membuat aplikasi berbasis peristiwa yang merespons perubahan dalam data Anda. Misalnya, Anda dapat menggunakan fungsi Lambda untuk memproses dokumen baru, melacak pembaruan ke dokumen yang ada, atau mencatat dokumen yang dihapus.

Anda dapat mengonfigurasi pemetaan sumber peristiwa untuk mengirim catatan dari aliran perubahan Amazon DocumentDB ke fungsi Lambda. Acara dapat dikirim satu per satu atau batch untuk meningkatkan efisiensi dan akan diproses secara berurutan. Anda dapat mengonfigurasi perilaku batching pemetaan sumber peristiwa berdasarkan durasi jendela waktu tertentu (0 - 300 detik) atau jumlah catatan batch (batas maksimal 10.000 catatan). Anda dapat membuat beberapa pemetaan sumber peristiwa untuk memproses data yang sama dengan beberapa fungsi Lambda, atau untuk memproses item yang berbeda dari beberapa aliran dengan satu fungsi.

Jika fungsi Anda mengembalikan kesalahan, Lambda mencoba ulang batch hingga berhasil diproses. Jika peristiwa dalam aliran perubahan telah kedaluwarsa, Lambda akan menonaktifkan pemetaan sumber peristiwa. Dalam hal ini, Anda dapat membuat pemetaan sumber peristiwa baru dan mengonfigurasinya dengan posisi awal pilihan Anda. Pemetaan sumber peristiwa Lambda memproses peristiwa setidaknya sekali karena sifat pollernya yang terdistribusi. Akibatnya, fungsi Lambda Anda mungkin menerima kejadian duplikat dalam situasi yang jarang terjadi. Ikuti praktik terbaik untuk bekerja denganAWS Lambdafungsi dan membangun fungsi idempoten untuk menghindari masalah yang terkait dengan peristiwa duplikat. Untuk informasi lebih lanjut lihatMenggunakanAWS Lambda consoledengan Amazon DocumentDBdiAWS LambdaPanduan Pengembang.

Sebagai praktik terbaik kinerja, fungsi Lambda harus berumur pendek. Untuk menghindari penundaan pemrosesan yang tidak perlu, itu juga tidak boleh menjalankan logika yang kompleks. Untuk aliran kecepatan tinggi khususnya, lebih baik memicu alur kerja fungsi langkah pasca-pemrosesan asinkron daripada Lambdas yang berjalan lama secara sinkron. Untuk informasi selengkapnya tentang AWS Lambda, lihat Panduan Developer AWS Lambda.

Keterbatasan:

Berikut ini adalah batasan yang perlu dipertimbangkan saat bekerja dengan Amazon DocumentDB danAWS Lambda:

- AWS Lambdasaat ini hanya didukung di Amazon DocumentDB 4.0 dan 5.0.
- AWS Lambdasaat ini tidak didukung pada cluster elastis atau cluster global.
- AWS Lambdaukuran muatan tidak boleh melebihi 6MB. Untuk informasi selengkapnya tentang ukuran batch Lambda, lihat “Perilaku pengelompokan” diPemetaan sumber acara Lambdabagian diAWS LambdaPanduan Pengembang.

Menggunakan validasi skema JSON

Menggunakan operator kueri `$jsonSchema` evaluasi, Anda dapat memvalidasi dokumen yang dimasukkan ke dalam koleksi Anda.

Topik

- Membuat dan menggunakan validasi skema JSON
- Kata kunci yang didukung
- bypassDocumentValidation
- Batasan

Membuat dan menggunakan validasi skema JSON

Membuat koleksi dengan validasi skema

Anda dapat membuat koleksi dengan aturan `createCollection` operasi dan validasi. Aturan validasi ini diterapkan selama penyisipan atau pembaruan dokumen Amazon DocumentDB. Contoh kode berikut menunjukkan aturan validasi untuk koleksi karyawan:

```
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"
} )
```

Memasukkan dokumen yang valid

Contoh berikut menyisipkan dokumen yang sesuai dengan aturan validasi skema di atas:

```
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})
```

Memasukkan dokumen yang tidak valid

Contoh berikut menyisipkan dokumen yang tidak sesuai dengan aturan validasi skema di atas. Dalam contoh ini, nilai EmployeeId bukan string:

```
db.employees.insert({
    "name" : { "firstName" : "Carol" , "lastName" : "Smith"},
    "employeeId": 720 ,
    "salary": 1000.0
})
```

Contoh ini menunjukkan sintaks yang salah dalam dokumen.

Memodifikasi koleksi

`collMod`Perintah ini digunakan untuk menambah atau memodifikasi aturan validasi koleksi yang ada. Contoh berikut menambahkan bidang gaji ke daftar bidang wajib:

```
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
      }
   }
} )
```

Mengatasi dokumen yang ditambahkan sebelum aturan validasi diubah

Untuk menangani dokumen yang ditambahkan ke koleksi Anda sebelum aturan validasi diubah, gunakan `validationLevel` pengubah berikut:

- ketat: Menerapkan aturan validasi pada semua sisipan dan pembaruan.
- moderat: Menerapkan aturan validasi ke dokumen valid yang ada. Selama pembaruan, dokumen tidak valid yang ada tidak diperiksa.

Dalam contoh berikut, setelah memperbarui aturan validasi pada koleksi bernama “karyawan”, bidang gaji diperlukan. Memperbarui dokumen berikut akan gagal:

```
db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB mengembalikan output berikut:

```
{
"n" : 0,
    "nModified" : 0,
    "writeErrors" : [
        {
"index" : 0,
            "code" : 121,
            "errmsg" : "Document failed validation"
        }
```

```
    ],
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Memperbarui tingkat validasi `moderate` akan memungkinkan dokumen di atas diperbarui dengan sukses:

```
db.runCommand({
    "collMod" : "employees",
    validationLevel : "moderate"
})

db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB mengembalikan output berikut:

```
{
"n" : 1,
    "nModified" : 1,
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Mengambil dokumen dengan $JSONSchema

`$jsonSchema`Operator dapat digunakan sebagai filter untuk menanyakan dokumen yang cocok dengan skema JSON. Ini adalah operator tingkat atas yang dapat hadir dalam dokumen filter sebagai bidang tingkat atas atau digunakan dengan operator kueri seperti`$and`,`$or`, dan`$nor`. Contoh berikut menunjukkan penggunaan $JSONSchema sebagai filter individual dan dengan operator filter lainnya:

Dokumen dimasukkan ke dalam koleksi “karyawan”:

```
{ "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 }
```

Koleksi disaring dengan `$jsonSchema` operator saja:

```
db.employees.find({
       $jsonSchema: { required: ["age"] } })
```

Amazon DocumentDB mengembalikan output berikut:

```
{ "_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 }
```

Koleksi disaring dengan `$jsonSchema` operator dan operator lain:

```
db.employees.find({
       $or: [{ $jsonSchema: { required: ["age", "name"]}},
            { salary: { $lte:1000}}]});
```

Amazon DocumentDB mengembalikan output berikut:

```
{ "_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 }
```

Koleksi disaring dengan `$jsonSchema` operator dan dengan `$match` dalam filter agregat:

```
db.employees.aggregate(
    [{ $match: {
        $jsonSchema: {
            required: ["name", "employeeId"],
            properties: {"salary" :{"bsonType": "double"}}
```

```
        }
       }
    }]
)
```

Amazon DocumentDB mengembalikan output berikut:

```
{
"_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
}
```

Melihat aturan validasi yang ada

Untuk melihat aturan validasi yang ada pada koleksi, gunakan:

```
db.runCommand({
    listCollections: 1,
    filter: { name: 'employees' }
})
```

Amazon DocumentDB mengembalikan output berikut:

```
{
    "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 juga mempertahankan aturan validasi dalam tahap agregasi. `$out`

Kata kunci yang didukung

Bidang berikut didukung dalam `collMod` perintah `create` dan:

- **`Validator`**— `$jsonSchem` Mendukung operator.
- **`ValidationLevel`**— Mendukung`off`,`strict`, dan `moderate` nilai-nilai.
- **`ValidationAction`**— Mendukung `error` nilai.

Operator $JSONSchema mendukung kata kunci berikut:

- 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 `bypassDocumentValidation` mendukung perintah dan metode berikut:

- `insert`
- `update`
- `findAndModify`
- `$out`tahap dalam `aggregate` perintah dan `db.collection.aggregate()` metode

Amazon DocumentDB tidak mendukung perintah berikut untuk: `bypassDocumentValidation`

- `$merge`dalam `aggregate` perintah dan dalam `db.collection.aggregate()` metode
- `mapReduce`perintah dan `db.collection.mapReduce()` metode
- Perintah `applyOps`

Batasan

Batasan berikut berlaku untuk `$jsonSchema` validasi:

- Amazon DocumentDB mengembalikan kesalahan “Dokumen gagal validasi” ketika operasi gagal dalam aturan validasi.
- Cluster elastis Amazon DocumentDB tidak mendukung. `$jsonSchema`

Menghubungkan ke Amazon DocumentDB sebagai Set Replika

Saat Anda mengembangkan pada Amazon DocumentDB (dengan kompatibilitas MongoDB), Anda sebaiknya membuat sambungan ke klaster Anda sebagai set replika dan mendistribusikan pembacaan ke instans replika menggunakan kemampuan preferensi baca bawaan driver Anda. Bagian ini membahas lebih dalam apa artinya dan menjelaskan bagaimana Anda dapat terhubung ke klaster Amazon DocumentDB Anda sebagai set replika menggunakan SDK for Python sebagai contoh.

Amazon DocumentDB memiliki tiga titik akhir yang dapat Anda gunakan untuk terhubung ke klaster Anda:

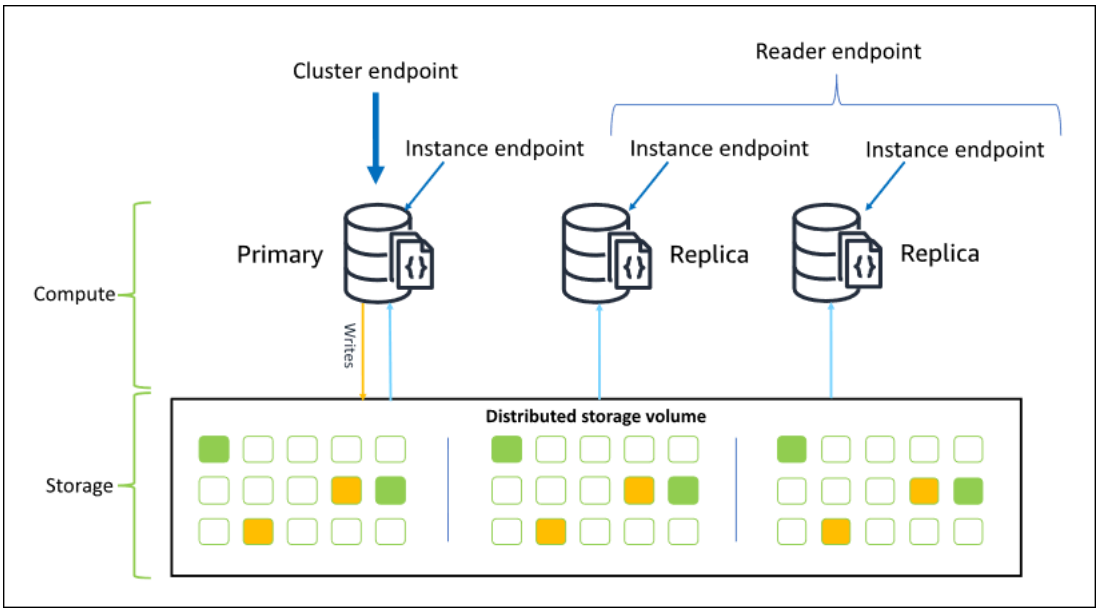
- Titik akhir klaster

- Reader endpoint
- Endpoint instans

Dalam kebanyakan kasus saat Anda menghubungkan ke Amazon DocumentDB, kami merekomendasikan Anda menggunakan titik akhir klaster. Ini adalah CNAME yang menunjuk ke instans primer di klaster Anda, seperti yang ditunjukkan pada diagram berikut.

Saat menggunakan terowongan SSH, kami rekomendasikan Anda menyambung ke klaster Anda menggunakan titik akhir klaster dan jangan mencoba untuk menyambung dalam mode set replika (yaitu, menentukan `replicaSet=rs0` dalam string koneksi Anda) karena akan menghasilkan kesalahan.

> **Note**
>
> Untuk informasi selengkapnya tentang titik akhir Amazon DocumentDB, lihat Titik Akhir Amazon DocumentDB.



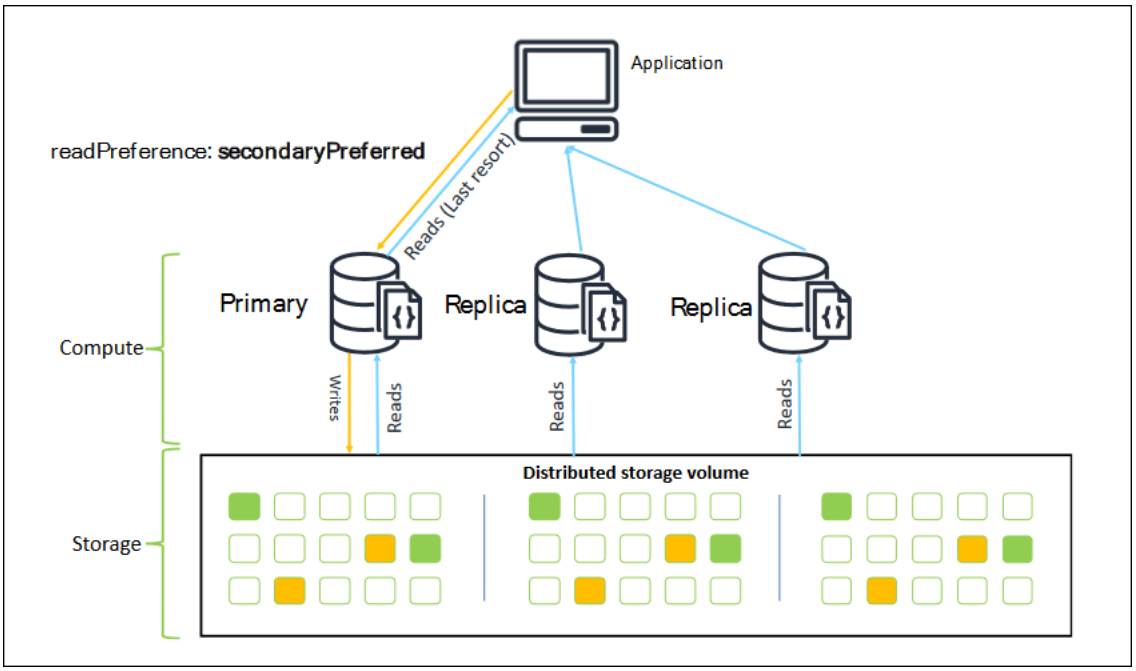
Dengan menggunakan titik akhir klaster, Anda dapat terhubung ke klaster Anda dalam mode set replika. Anda kemudian dapat menggunakan kemampuan driver preferensi baca bawaan. Dalam contoh berikut, menentukan `/?replicaSet=rs0` menandakan untuk SDK yang ingin Anda hubungkan sebagai set replika. Jika Anda menghilangkan `/?replicaSet=rs0'`, klien merutekan semua permintaan ke titik akhir klaster, yaitu instans primer Anda.

```
## 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')
```

Keuntungan menghubungkan sebagai set replika adalah memungkinkan SDK Anda menemukan topografi klaster secara otomatis, termasuk saat instans ditambahkan atau dihapus dari klaster. Anda kemudian dapat menggunakan klaster Anda secara lebih efisien dengan merutekan permintaan baca ke instans replika Anda.

Ketika Anda terhubung sebagai set replika, Anda dapat menentukan `readPreference` untuk koneksi tersebut. Jika Anda menentukan preferensi baca `secondaryPreferred`, klien merutekan kueri baca ke replika Anda dan menulis kueri ke instans primer Anda (seperti dalam diagram berikut). Ini adalah penggunaan yang lebih baik dari sumber daya klaster Anda. Untuk informasi selengkapnya, lihat Opsi Preferensi Baca.

```
## 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')
```



Pembacaan dari replika Amazon DocumentDB pada akhirnya adalah konsisten. Mereka mengembalikan data dalam urutan yang sama seperti yang tertulis di primer, dan sering kali ada jeda

replikasi kurang dari 50 ms. Anda dapat memantau jeda replika untuk klaster Anda menggunakan metrik Amazon CloudWatch `DBInstanceReplicaLag` dan `DBClusterReplicaLagMaximum`. Untuk informasi selengkapnya, lihat Memantau Amazon DocumentDB dengan CloudWatch.

Tidak seperti arsitektur basis data monolitik tradisional, Amazon DocumentDB memisahkan penyimpanan dan komputasi. Dengan arsitektur modern ini, kami mendorong Anda untuk membaca skala pada instans replika. Membaca pada instans replika tidak memblokir penulisan yang direplikasi dari instans primer. Anda dapat menambahkan hingga 15 instans replika baca dalam klaster dan menskalakan keluar hingga jutaan pembacaan per detik.

Manfaat utama menghubungkan sebagai set replika dan mendistribusikan pembacaan replika adalah bahwa hal itu meningkatkan sumber daya secara keseluruhan dalam klaster Anda yang tersedia untuk melakukan pekerjaan bagi aplikasi Anda. Sebaiknya hubungkan sebagai set replika sebagai praktik terbaik. Lebih lanjut, kami merekomendasikannya paling sering dalam skenario berikut:

- Anda menggunakan CPU hampir 100 persen pada primer Anda.
- Rasio hit buffer cache mendekati nol.
- Anda mencapai batas koneksi atau kursor untuk masing-masing instans.

Menskalakan ke atas ukuran instans klaster adalah sebuah opsi, dan dalam beberapa kasus, itu bisa menjadi cara terbaik untuk menskalakan klaster. Tetapi Anda juga harus mempertimbangkan cara menggunakan replika yang sudah Anda miliki di klaster dengan lebih baik. Hal ini memungkinkan Anda meningkatkan skala tanpa peningkatan biaya dari penggunaan tipe instans yang lebih besar. Kami juga menyarankan Anda untuk memantau dan mewaspadai batas-batas ini (yaitu `CPUUtilization`, `DatabaseConnections`, dan `BufferCacheHitRatio`) menggunakan alarm CloudWatch sehingga Anda tahu ketika sumber daya sedang banyak digunakan.

Untuk informasi lain, lihat topik berikut:

- Praktik Terbaik untuk Amazon DocumentDB
- Kuota dan Batas Amazon DocumentDB

Menggunakan Koneksi Klaster

Pertimbangkan skenario menggunakan semua koneksi di klaster Anda. Sebagai contoh, sebuah instans `r5.2xlarge` memiliki batas 4.500 koneksi (dan 450 kursor terbuka). Jika Anda membuat tiga instans klaster Amazon DocumentDB dan terhubung hanya ke instans primer menggunakan titik akhir

klaster, batas klaster Anda untuk koneksi terbuka dan kursor masing-masing adalah 4.500 dan 450. Anda mungkin mencapai batas ini jika Anda membangun aplikasi yang menggunakan banyak pekerja yang berputar dalam kontainer. Kontainer membuka sejumlah koneksi sekaligus dan memenuhi klaster.

Sebagai gantinya, Anda dapat terhubung ke klaster Amazon DocumentDB sebagai set replika dan mendistribusikan pembacaan Anda ke instans replika. Anda kemudian dapat secara efektif melipatgandakan jumlah koneksi dan kursor yang tersedia di klaster menjadi 13.500 dan 1.350 masing-masing. Menambahkan lebih banyak instans ke klaster hanya meningkatkan jumlah koneksi dan kursor untuk beban kerja baca. Jika Anda perlu meningkatkan jumlah koneksi untuk penulisan ke klaster Anda, sebaiknya tingkatkan ukuran instans.

> **Note**
>
> Jumlah koneksi untuk instans `large`, `xlarge`, dan `2xlarge` meningkat dengan ukuran instans hingga 4.500. Jumlah maksimum koneksi per instans untuk instans `4xlarge` atau lebih besar adalah 4.500. Untuk informasi selengkapnya tentang batasan menurut tipe instans, lihat Batas Instans.

Biasanya kami tidak menyarankan Anda untuk terhubung ke klaster Anda menggunakan preferensi baca `secondary`. Hal ini karena jika tidak ada instans replika di klaster Anda, pembacaan akan gagal. Misalnya, anggaplah bahwa Anda memiliki dua instans klaster Amazon DocumentDB dengan satu satu primer dan satu replika. Jika replika mengalami masalah, permintaan baca dari kolam koneksi yang ditetapkan sebagai `secondary` gagal. Keuntungan dari `secondaryPreferred` adalah bahwa jika klien tidak dapat menemukan instans replika yang cocok untuk terhubung ke, hal itu akan mengembalikan ke primer untuk pembacaan.

Beberapa Kolam Koneksi

Dalam beberapa skenario, pembacaan dalam aplikasi harus memiliki konsistensi baca-setelah-tulis, yang hanya dapat disajikan dari instans utama di Amazon DocumentDB. Dalam skenario ini, Anda dapat membuat dua kolam koneksi klien: satu untuk menulis dan satu untuk membaca yang memerlukan konsistensi baca-setelah-tulis. Untuk melakukan itu, kode Anda akan terlihat seperti berikut ini.

```
## 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')
```

Opsi lainnya adalah membuat kolam koneksi tunggal dan menimpa preferensi baca untuk koleksi tertentu.

```
##Specify the collection and set the read preference level for that collection
col = db.review.with_options(read_preference=ReadPreference.SECONDARY_PREFERRED)
```

Ringkasan

Untuk menggunakan sumber daya di klaster Anda dengan lebih baik, sebaiknya hubungkan ke klaster Anda menggunakan mode set replika. Jika cocok untuk aplikasi Anda, Anda dapat membaca skala aplikasi Anda dengan mendistribusikan pembacaan Anda ke instans replika.

Menghubungkan ke Klaster Amazon DocumentDB dari Luar Amazon VPC

Klaster Amazon DocumentDB (dengan kompatibilitas MongoDB) di-deploy dalam Amazon Virtual Private Cloud (Amazon VPC). Mereka dapat diakses langsung oleh instans Amazon EC2 atau layanan AWS lainnya yang di-deploy di Amazon VPC yang sama. Selain itu, Amazon DocumentDB dapat diakses oleh instans EC2 atau layanan AWS lainnya di VPC berbeda di Wilayah AWS yang sama atau Wilayah lain melalui peering VPC.

Namun, anggaplah kasus penggunaan Anda mengharuskan Anda (atau aplikasi Anda) mengakses sumber daya Amazon DocumentDB Anda dari luar VPC klaster. Dalam hal ini, Anda dapat menggunakan tunneling SSH (juga dikenal sebagai penerusan port) untuk mengakses sumber daya Amazon DocumentDB Anda.

Hal ini berada di luar cakupan topik ini untuk membahas tunneling SSH secara mendalam. Untuk informasi lebih lanjut tentang tunneling SSH, lihat hal berikut:

- Terowongan SSH
- Contoh Penerusan Port SSH, khususnya bagian Penerusan Lokal

Untuk membuat terowongan SSH, Anda memerlukan instans Amazon EC2 yang berjalan di Amazon VPC yang sama dengan klaster Amazon DocumentDB Anda. Anda dapat menggunakan instans EC2 yang ada di VPC yang sama sebagai klaster Anda atau membuatnya. Untuk informasi lebih lanjut, lihat topik yang sesuai untuk sistem operasi Anda:

- Memulai dengan Instans Linux Amazon EC2
- Memulai dengan Instans Windows Amazon EC2

Biasanya Anda terhubung ke instans EC2 dengan menggunakan perintah berikut.

```
ssh -i "ec2Access.pem" ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com
```

Jika demikian, Anda dapat mengatur terowongan SSH ke klaster Amazon DocumentDB `sample-cluster.node.us-east-1.docdb.amazonaws.com` dengan menjalankan perintah berikut di komputer lokal Anda. Bendera `-L` digunakan untuk meneruskan port lokal. Saat menggunakan terowongan SSH, kami rekomendasikan Anda menyambung ke klaster Anda menggunakan titik akhir klaster dan jangan mencoba untuk menyambung dalam mode set replika (yaitu, menentukan `replicaSet=rs0` dalam string koneksi Anda) karena akan menghasilkan kesalahan.

```
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
```

Setelah terowongan SSH dibuat, perintah apa pun yang Anda keluarkan ke `localhost:27017` diteruskan ke klaster Amazon DocumentDB `sample-cluster` yang berjalan di Amazon VPC. Jika Keamanan Lapisan Pengangkutan (TLS) diaktifkan pada klaster Amazon DocumentDB Anda, Anda perlu mengunduh kunci publik untuk Amazon DocumentDB dari https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem . Operasi berikut mengunduh file ini:

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

> **Note**
>
> TLS diaktifkan secara default untuk klaster Amazon DocumentDB baru. Namun, Anda dapat menonaktifkannya. Untuk informasi selengkapnya, lihat Mengelola Pengaturan TLS Klaster Amazon DocumentDB.

Untuk terhubung ke klaster Amazon DocumentDB Anda dari luar Amazon VPC, gunakan perintah berikut.

```
mongo --sslAllowInvalidHostnames --ssl --sslCAFile global-bundle.pem --username
 <yourUsername> --password <yourPassword>
```

Menghubungkan ke cluster Amazon DocumentDB dari Studio 3T

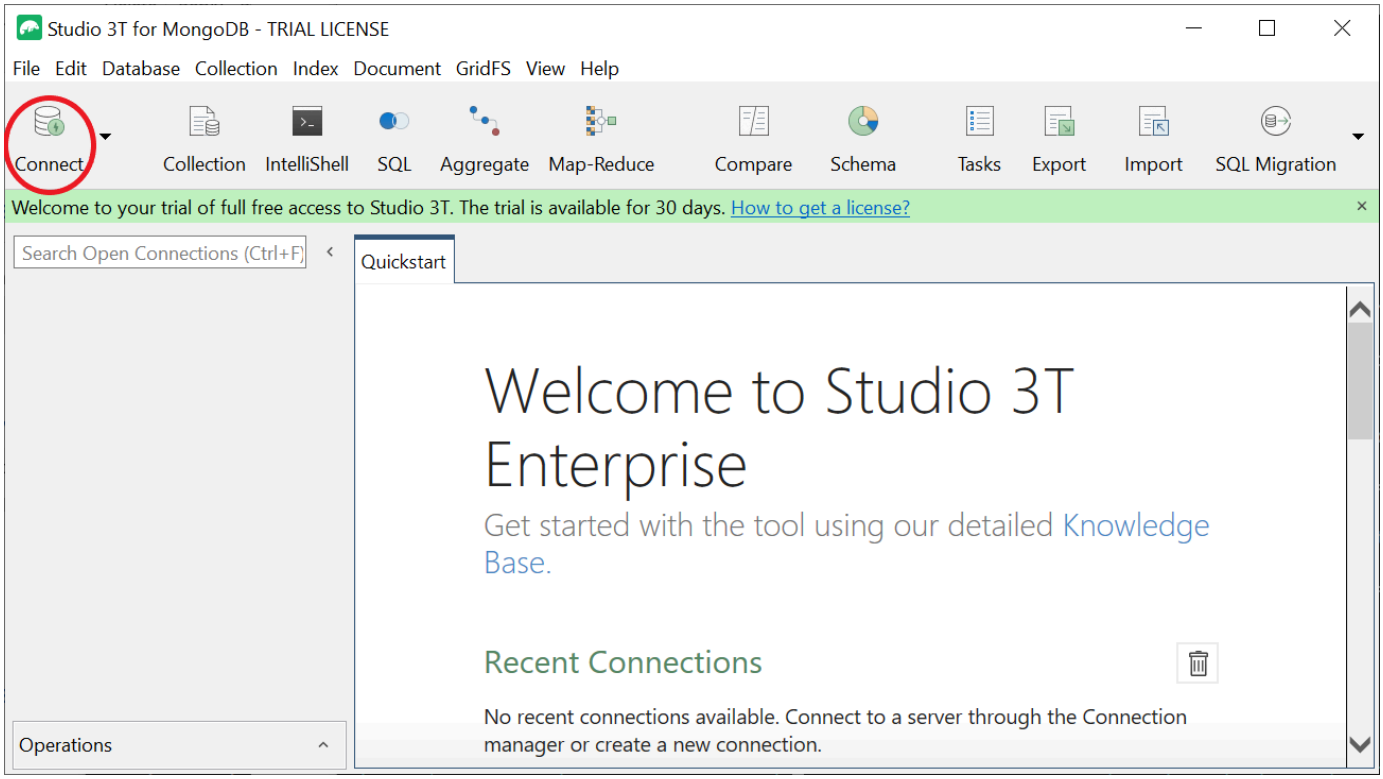
Studio 3T merupakan GUI dan IDE yang populer untuk developer dan teknisi data yang bekerja dengan MongoDB. Ini menawarkan beberapa kemampuan yang kuat Tampilan Tree, Table dan JSON data Anda, impor/ekspor mudah di CSV, JSON, SQL dan BSON/MongoDump, opsi kueri fleksibel, drag-and-drop UI visual, shell mongo bawaan dengan pelengkapan otomatis, editor pipeline agregasi, dan dukungan kueri SQL.

Prasyarat

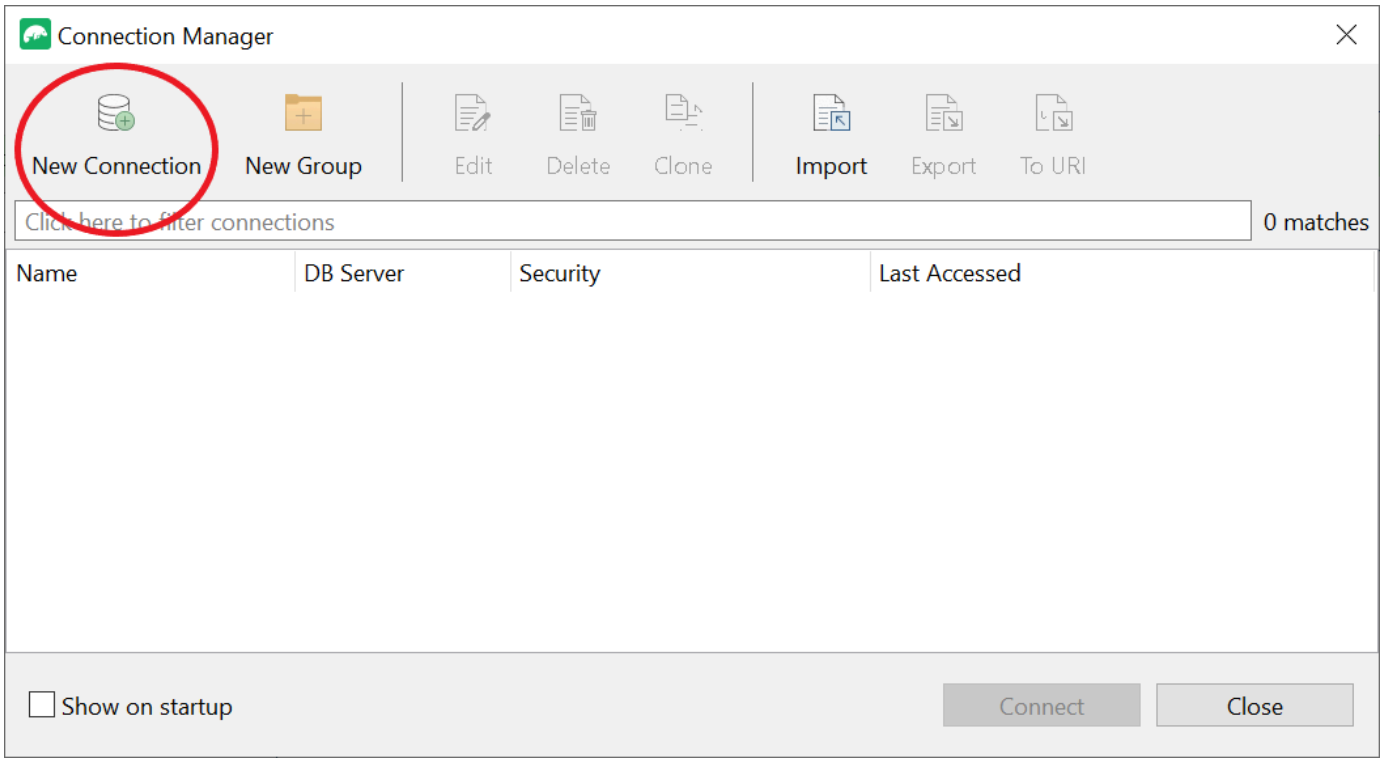
- Jika Anda belum memiliki cluster Amazon DocumentDB menggunakan Amazon EC2 sebagai host bastion/jump, ikuti petunjuk tentang cara Connect with Amazon EC2.
- Jika Anda tidak memiliki Studio 3T, unduh dan instal.

Connect dengan Studio 3T

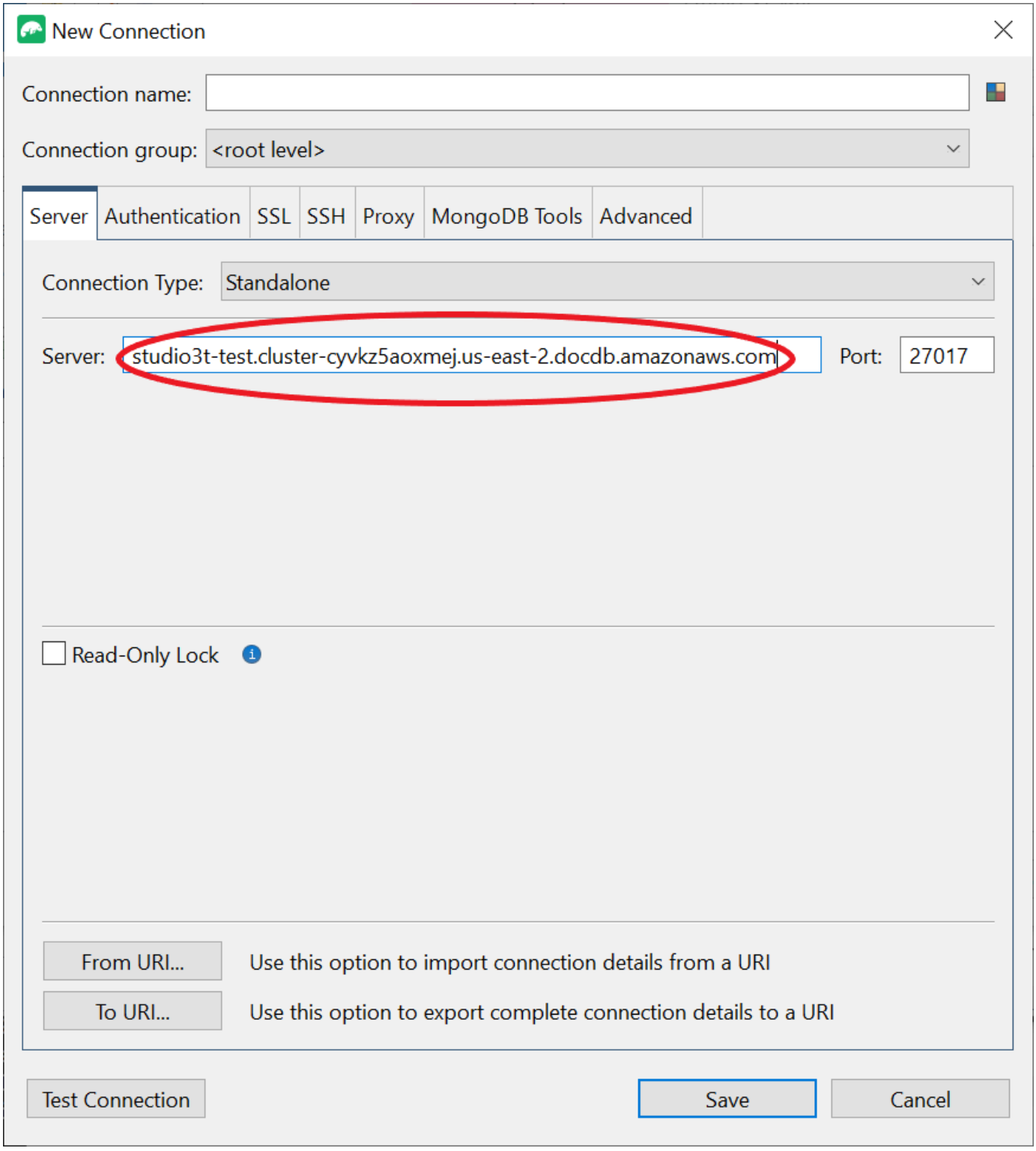
1. Pilih Connect di pojok kiri atas toolbar.



2. Pilih Koneksi Baru di pojok kiri atas toolbar.

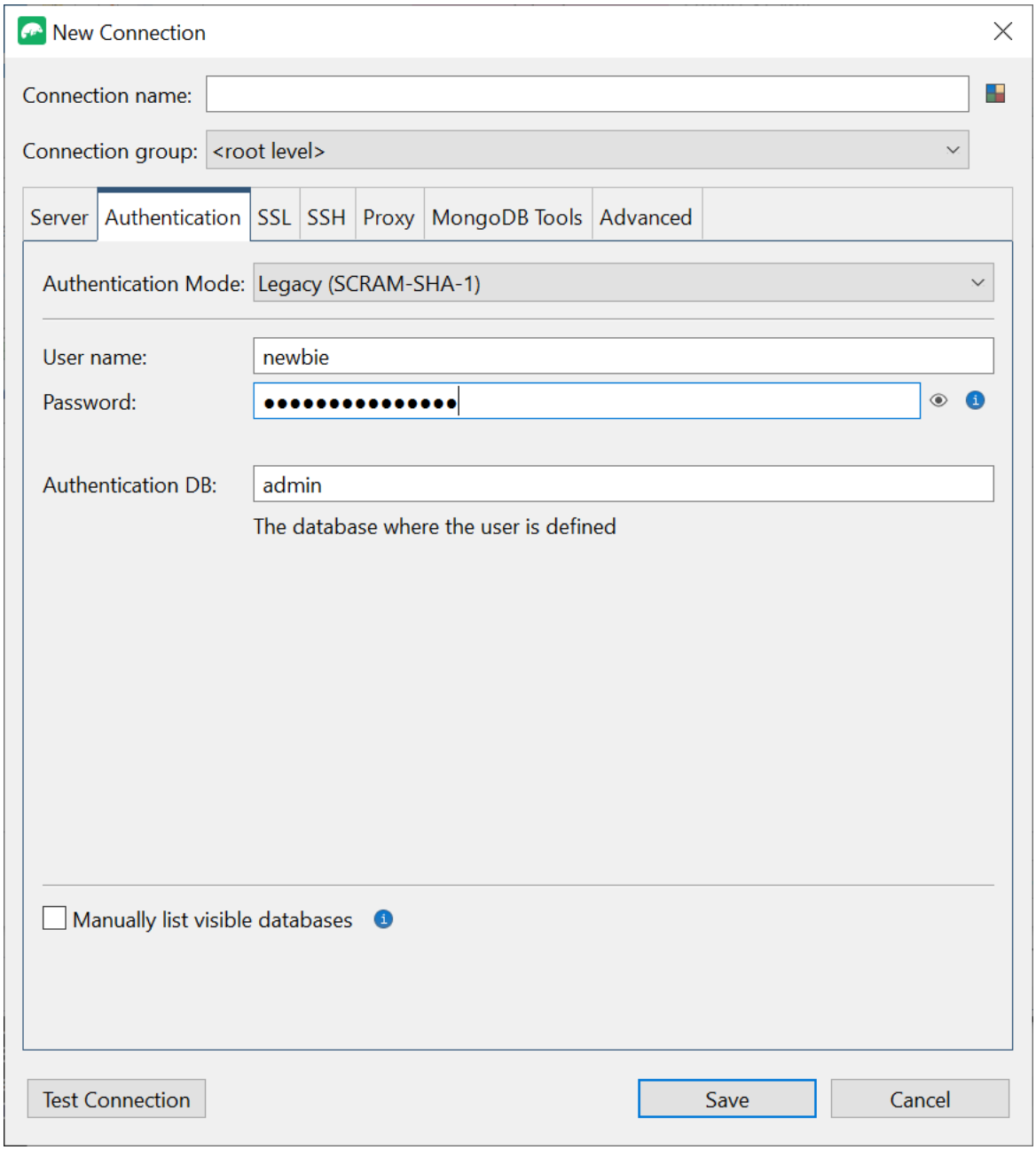


3. Pada tab Server, di bidang Server, masukkan informasi titik akhir klaster.

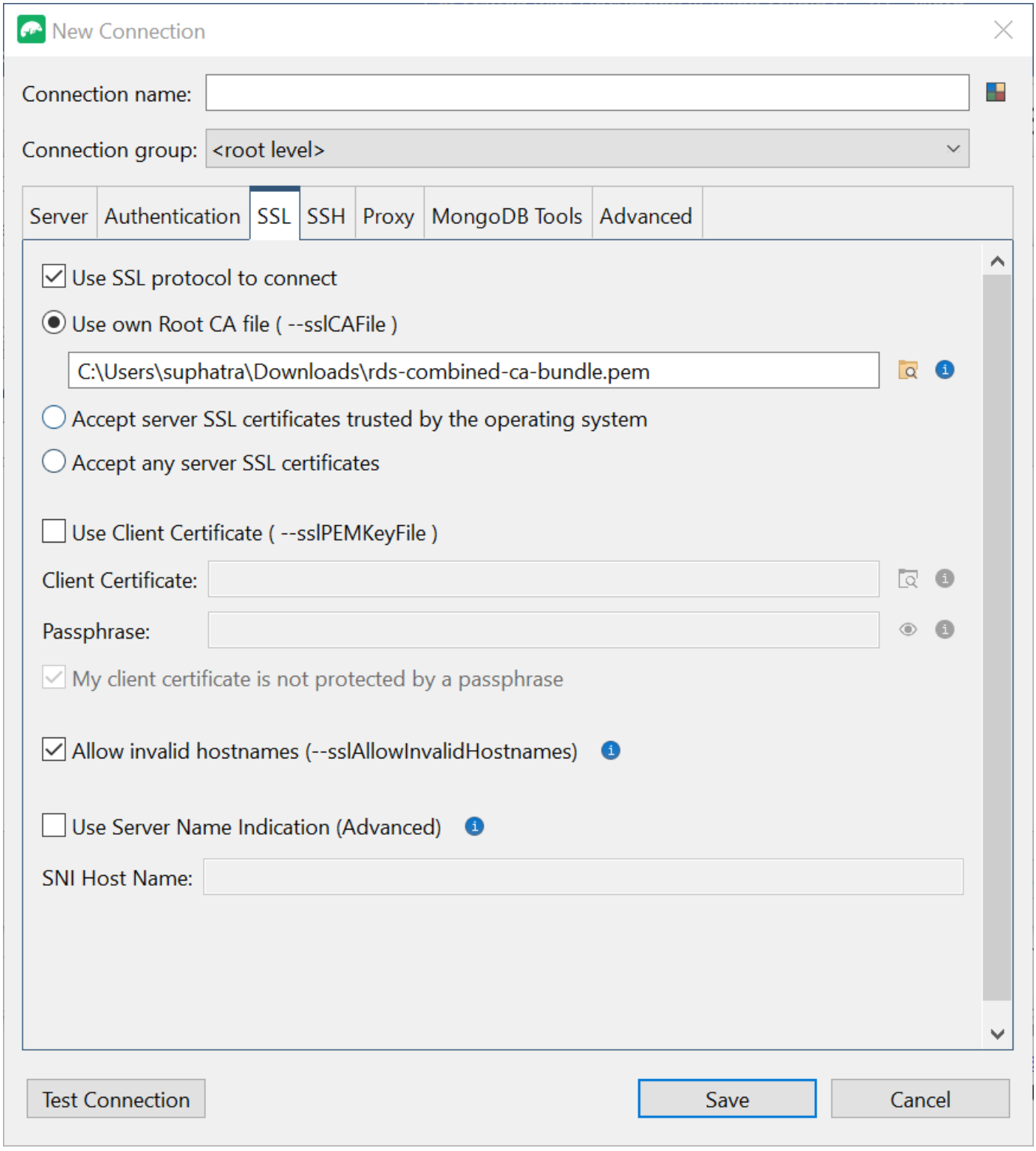


> **Note**
>
> Tidak dapat menemukan titik akhir klaster Anda? Ikuti saja langkah-langkahnya di sini.

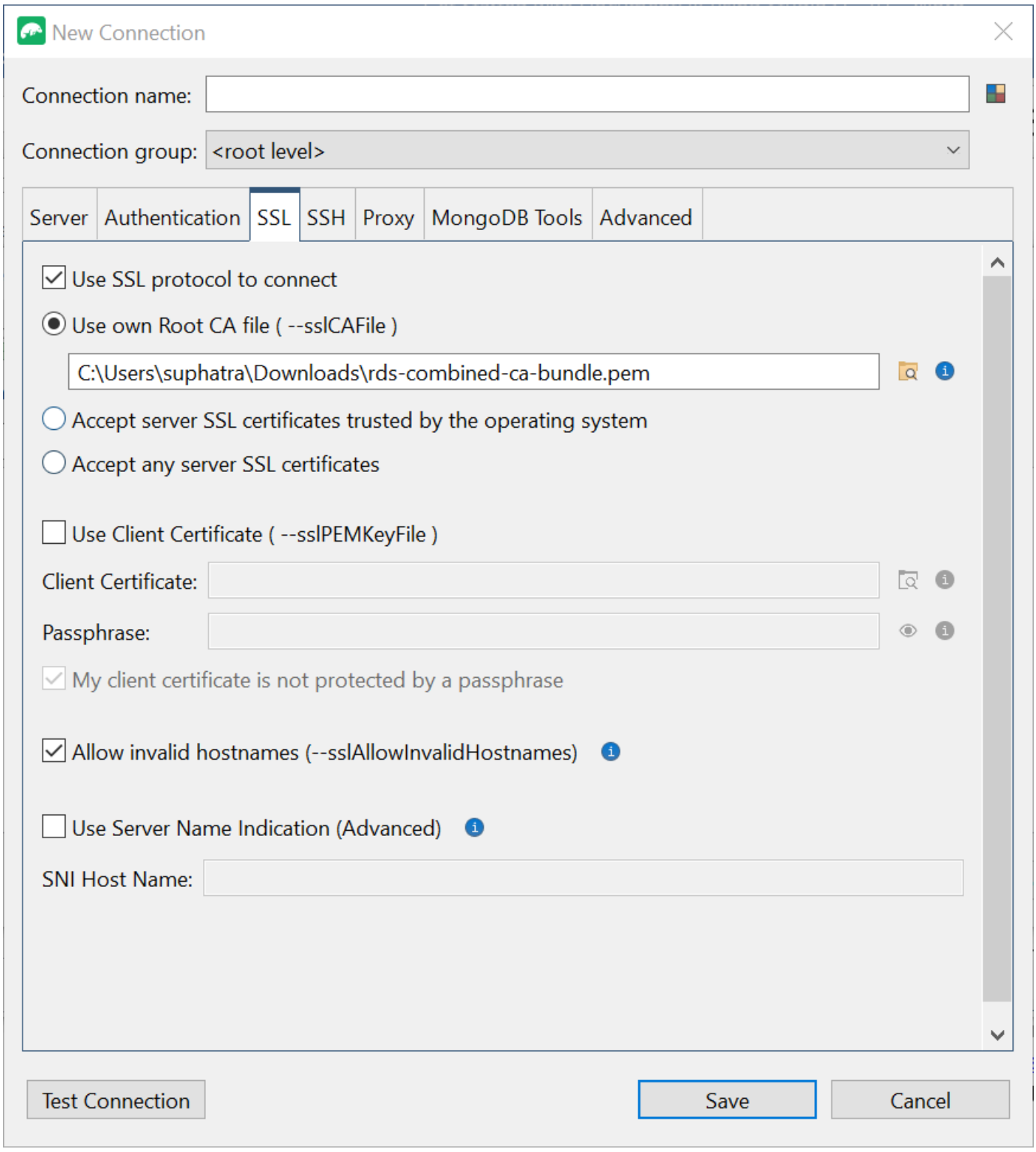
4. Pilih tab Autentikasi dan pilih Warisan di menu gulir ke bawah untuk Mode Autentikasi.



5. Masukkan nama pengguna dan kredensial Anda di bidang Nama pengguna dan Kata Sandi.
6. Pilih tab SSL dan centang kotak Gunakan protokol SSL untuk menghubungkan.



7. Pilih Gunakan file CA Root sendiri. Kemudian tambahkan sertifikat Amazon DocumentDB (Anda dapat melewatkan langkah ini jika SSL dinonaktifkan pada klaster DocumentDB Anda). Centang kotak untuk mengizinkan nama host tidak valid.



Note

Tidak memiliki sertifikat? Anda dapat mengunduhnya dengan perintah berikut ini:

```
wget https://truststore.pki.rds.amazonaws.com/global/global-
bundle.pem
```

8. Jika Anda menghubungkan dari mesin klien di luar Amazon VPC, Anda perlu membuat terowongan SSH. Anda akan melakukannya di tab SSH.

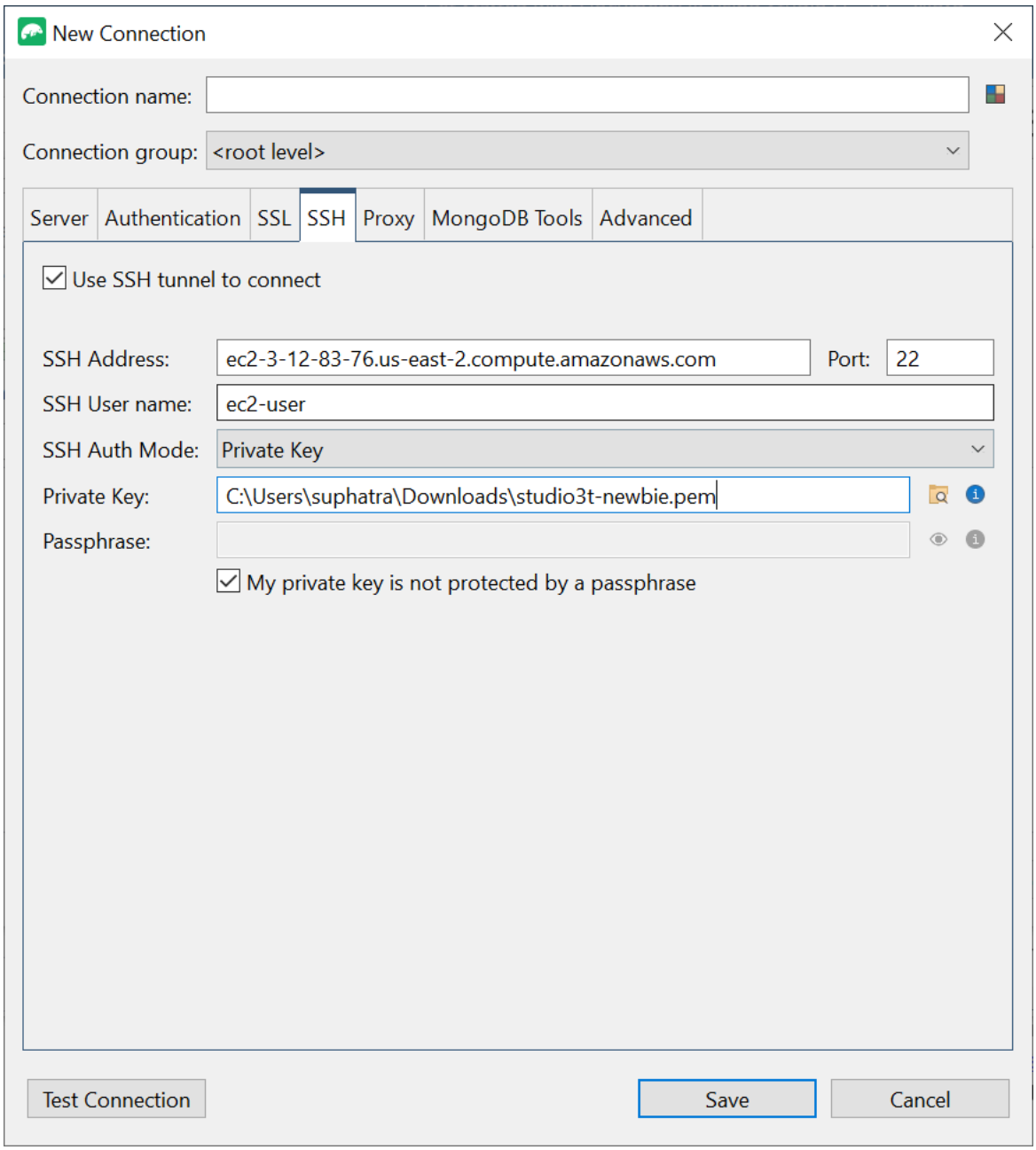
   a. Centang kotak untuk Gunakan terowongan SSH dan masukkan alamat SSH di bidang Alamat SSH. Ini adalah DNS Publik instans Anda (IPv4). Anda bisa mendapatkan URL ini dari Konsol Manajemen Amazon EC2.

   b. Masukkan nama pengguna Anda. Ini adalah nama pengguna instans Amazon EC2 Anda

   c. Untuk Mode Auten SSH, pilih Kunci Privat. Di bidang Kunci Privat, pilih ikon finder file untuk menemukan dan memilih kunci Privat dari instans Amazon EC2 Anda. Ini adalah file .pem (pasangan kunci) yang Anda simpan saat membuat instans Anda di Konsol Amazon EC2.

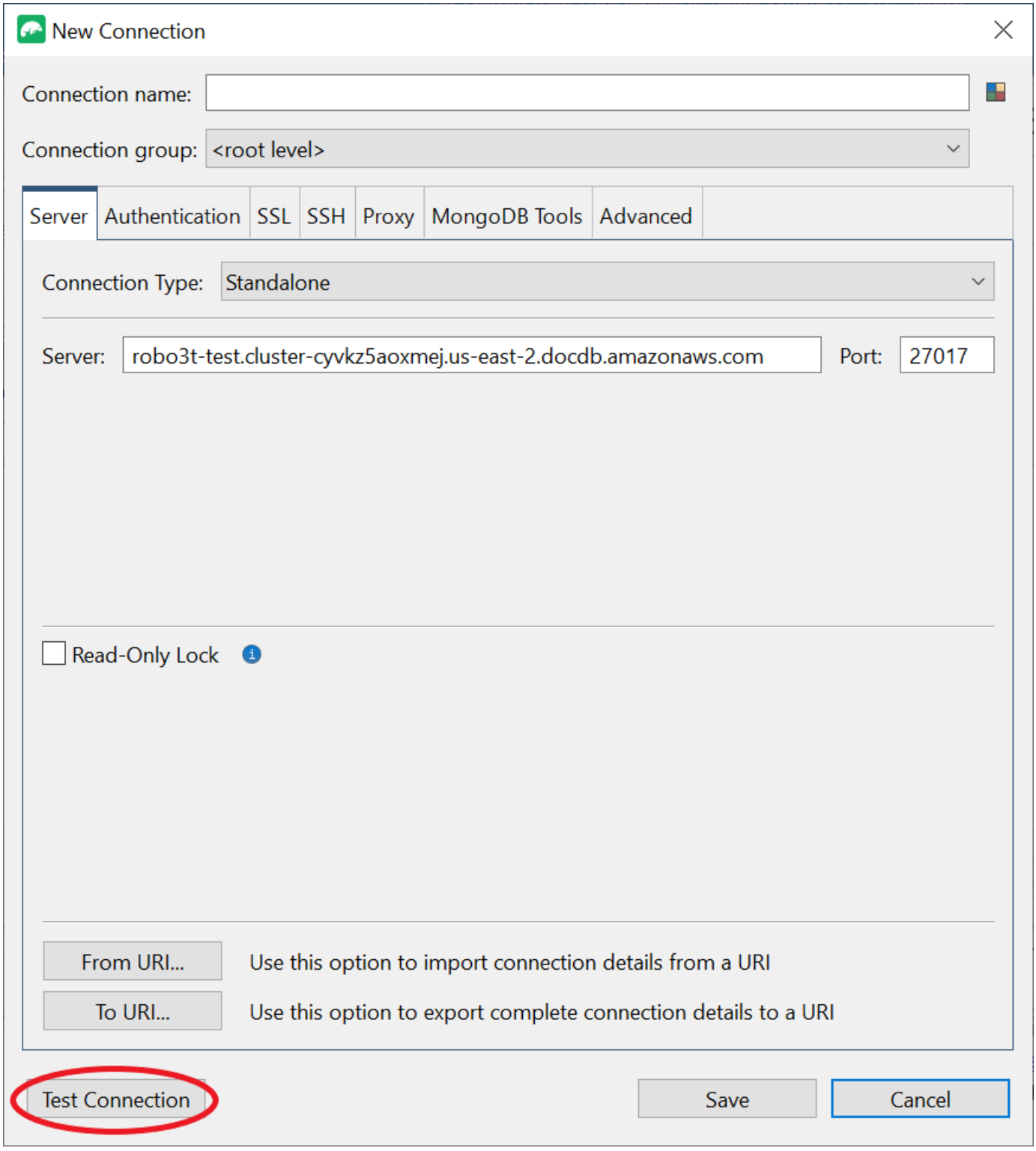
   d. Jika Anda menggunakan mesin klien Linux/macOS, Anda mungkin harus mengubah izin kunci privat Anda menggunakan perintah berikut ini:

      ```
      chmod 400 /fullPathToYourPemFile/<yourKey>.pem
      ```

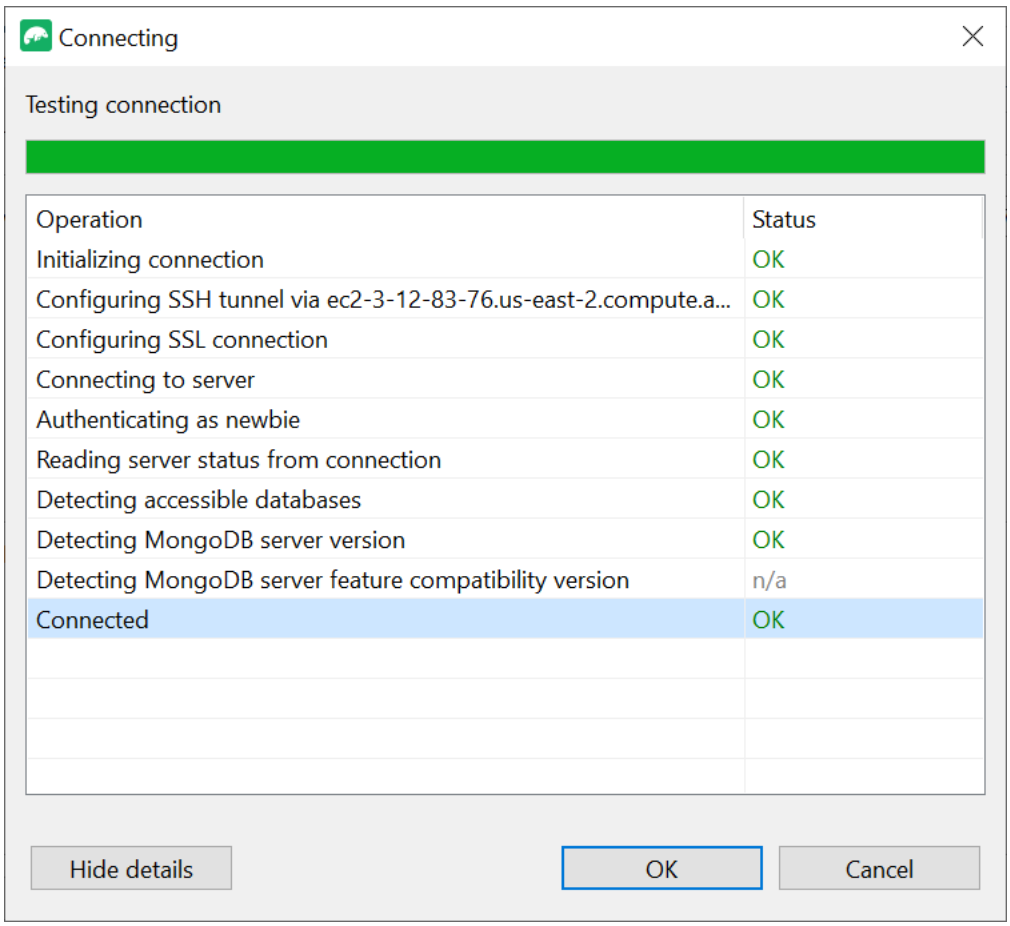


> **Note**
>
> Instans Amazon EC2 ini harus dalam Amazon VPC dan grup keamanan yang sama dengan klaster DocumentDB Anda. Anda bisa mendapatkan alamat SSH, nama pengguna, dan kunci privat dari Konsol Manajemen Amazon EC2.

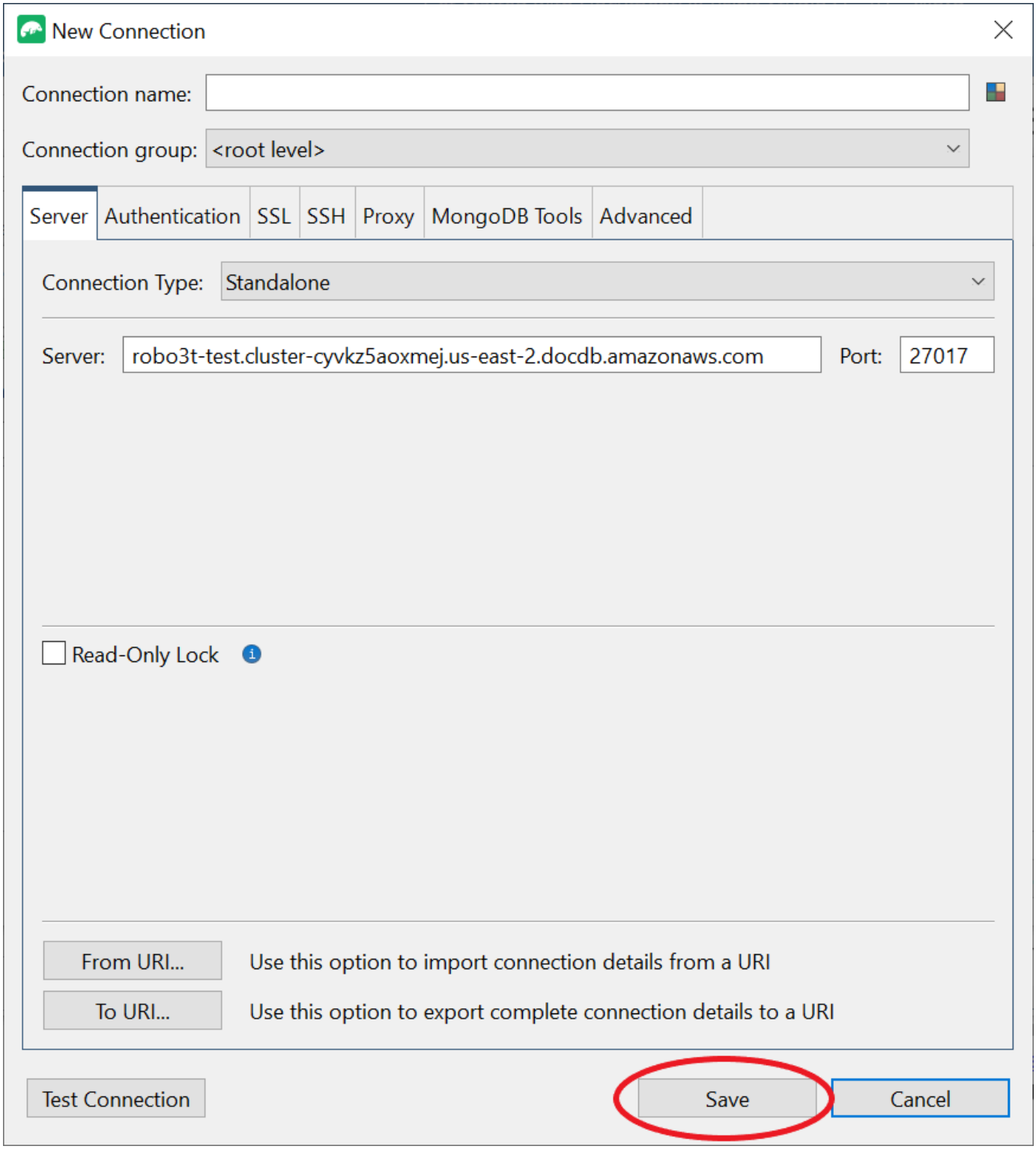
9. Sekarang uji konfigurasi Anda dengan memilih tombol Uji koneksi.



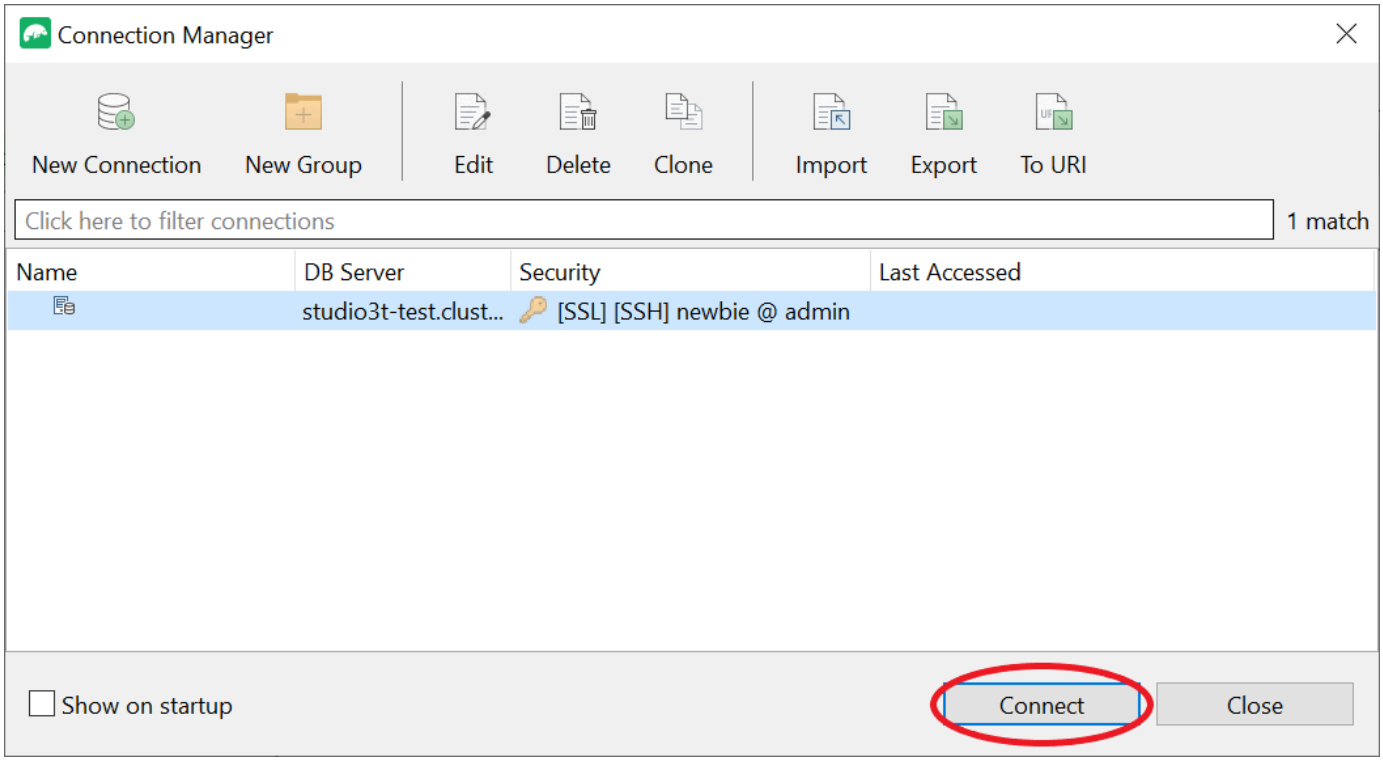
10. Jendela diagnostik harus memuat bar hijau untuk menunjukkan bahwa pengujian berhasil. Sekarang pilih OKE untuk menutup jendela diagnostik.



11. Pilih Simpan untuk menyimpan koneksi Anda untuk penggunaan masa depan.



12. Sekarang pilih klaster Anda dan pilih Connect.



Selamat! Anda sekarang berhasil terhubung ke klaster Amazon DocumentDB Anda melalui Studio 3T.

Connect ke Amazon DocumentDB menggunakan DataGrip

DataGripadalah lingkungan pengembangan terintegrasi (IDE) yang kuat yang mendukung berbagai sistem database, termasuk Amazon DocumentDB. Bagian ini memandu Anda melalui langkah-langkah untuk terhubung ke cluster Amazon DocumentDB Anda DataGrip menggunakan, memungkinkan Anda untuk dengan mudah mengelola dan menanyakan data Anda menggunakan antarmuka grafis.

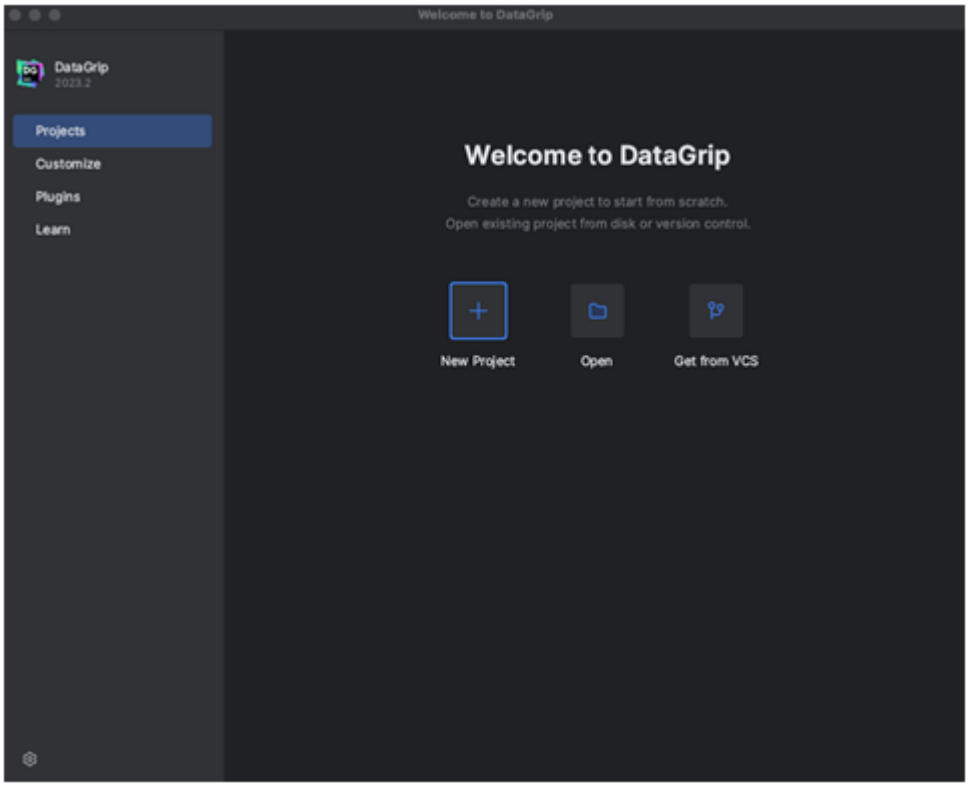
Prasyarat

- DataGrip IDE diinstal pada mesin Anda. Anda dapat mengunduhnya dari JetBrains.
- Instans Amazon EC2 yang berjalan di VPC yang sama dengan cluster Amazon DocumentDB Anda. Anda akan menggunakan instance ini untuk membuat terowongan aman dari mesin lokal Anda ke Amazon DocumentDBCluster. Ikuti instruksi tentang cara melakukannyaConnect menggunakan Amazon EC2.

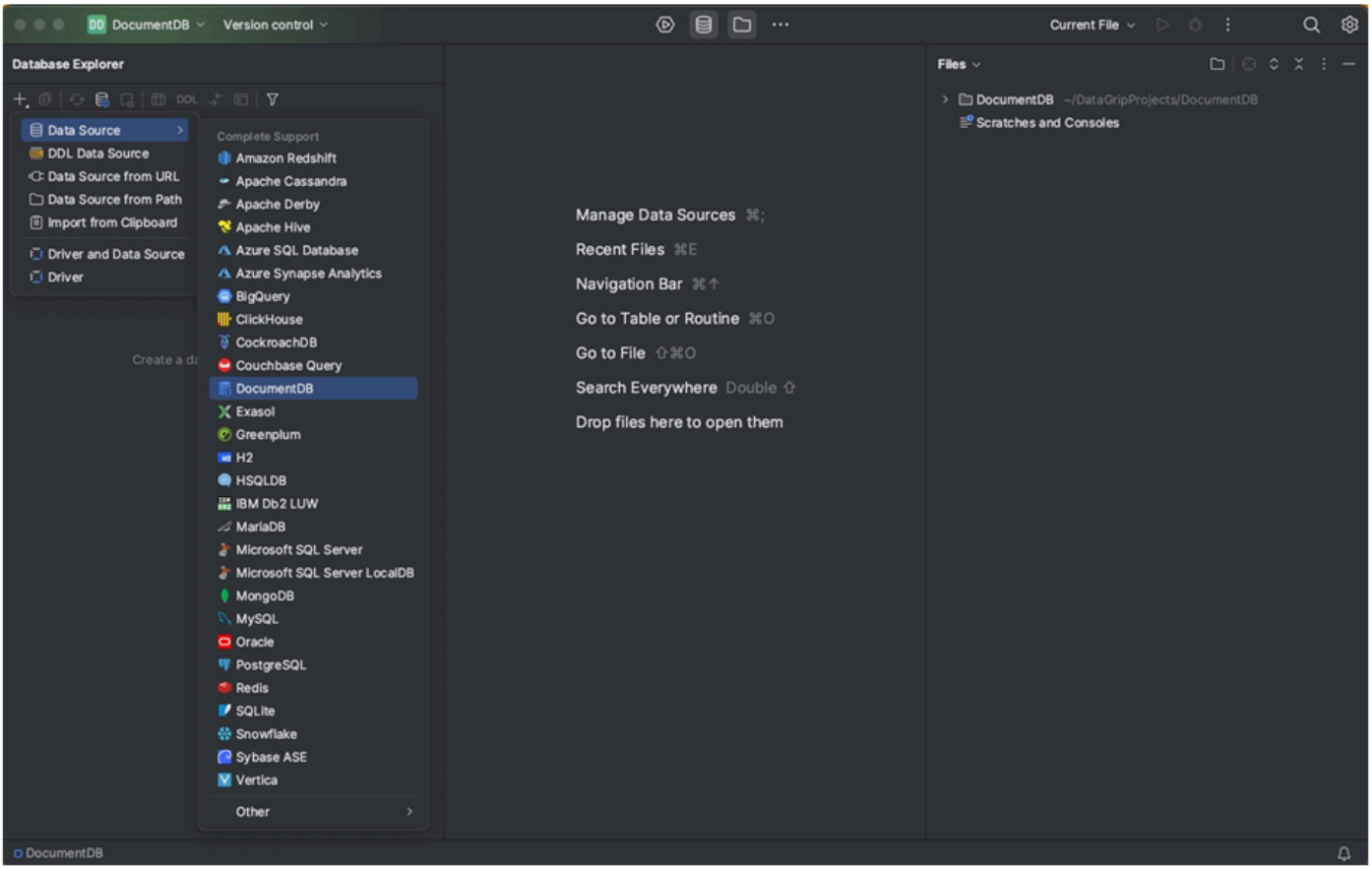
- Alternatif untuk instans Amazon EC2, koneksi VPN, atau jika Anda sudah mengakses AWS infrastruktur Anda menggunakan VPN aman. Jika Anda lebih suka opsi ini, ikuti petunjuk untuk mengakses Amazon DocumentDB dengan aman menggunakan. AWS Client VPN

Connect menggunakan DataGrip

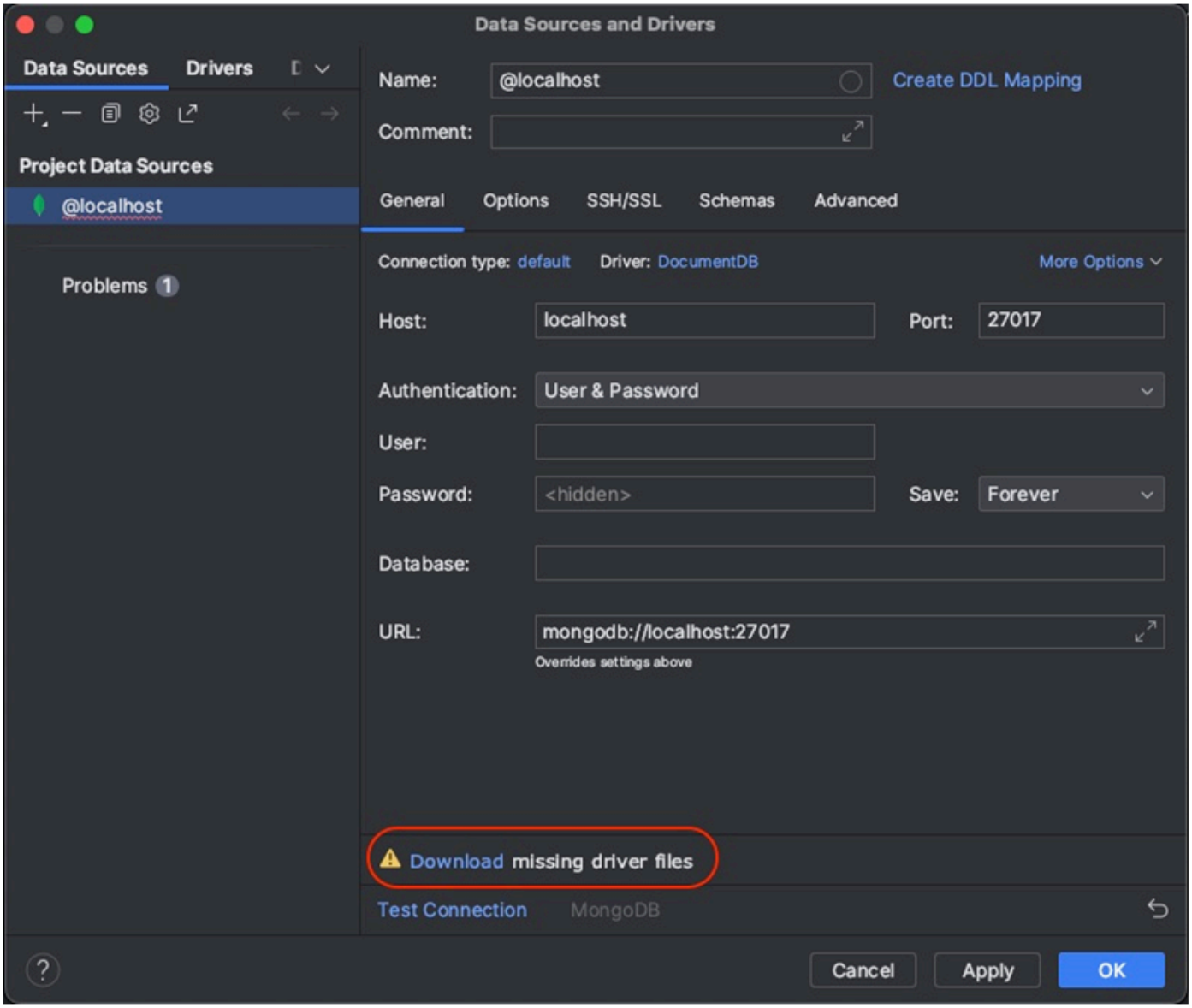
1. Luncurkan DataGrip di komputer Anda dan buat Proyek Baru.



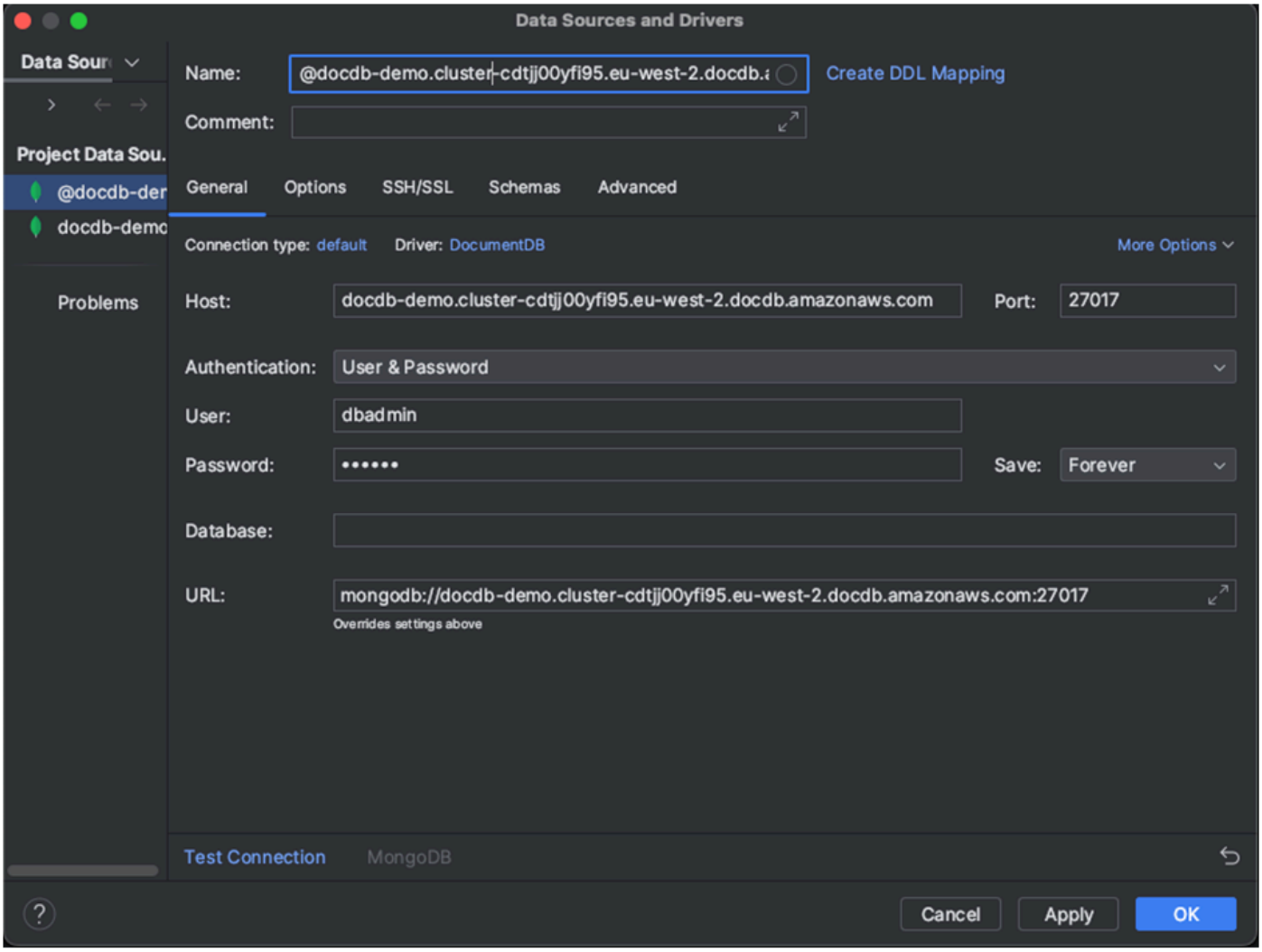
2. Tambahkan sumber data baru menggunakan salah satu cara berikut:
   a. Dari menu utama, arahkan ke File — New — Data Source dan pilih DocumentDB
   b. Di Database Explorer, klik ikon baru (+) di toolbar. Arahkan ke Sumber Data dan pilih DocumentDB.



3. Pada halaman Sumber Data di tab Umum, periksa apakah ada tautan Unduh file driver yang hilang di bagian bawah area pengaturan koneksi. Klik tautan ini untuk mengunduh driver yang diperlukan untuk berinteraksi dengan database. Untuk tautan unduhan langsung, lihat driver JetBrains JDBC.



4. Di tab Umum, tentukan detail koneksi:
   a. Di bidang Host, tentukan titik akhir cluster Amazon DocumentDB.
   b. Port sudah diatur ke 27017. Ubah jika cluster Anda diterapkan pada port yang berbeda.
   c. Untuk Otentikasi, pilih User & Password.
   d. Masukkan nama pengguna dan informasi kata sandi Anda.
   e. Bidang Database adalah opsional. Anda dapat menentukan database yang ingin Anda sambungkan.
   f. Bidang URL otomatis selesai saat Anda menambahkan detail di atas.



5. Di tab SSH/SSL, aktifkan Use SSH tunnel, lalu klik ikon untuk membuka dialog SSH Configuration. Masukkan informasi berikut:

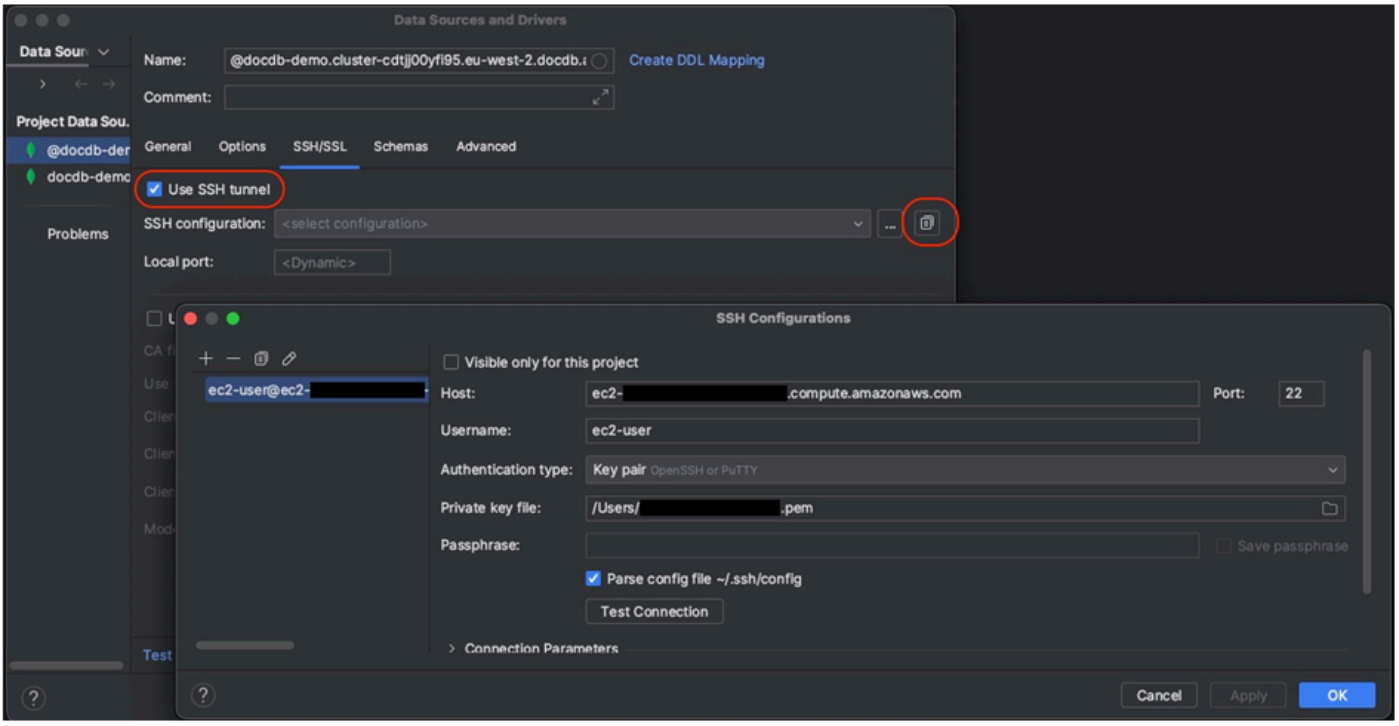
   a. di bidang Host, masukkan nama host instans Amazon EC2 Anda.

   b. Masukkan nama pengguna dan kata sandi untuk instans Amazon EC2 Anda.

   c. Untuk jenis Otentikasi, pilih Pasangan kunci.

   d. Masukkan file kunci pribadi Anda.

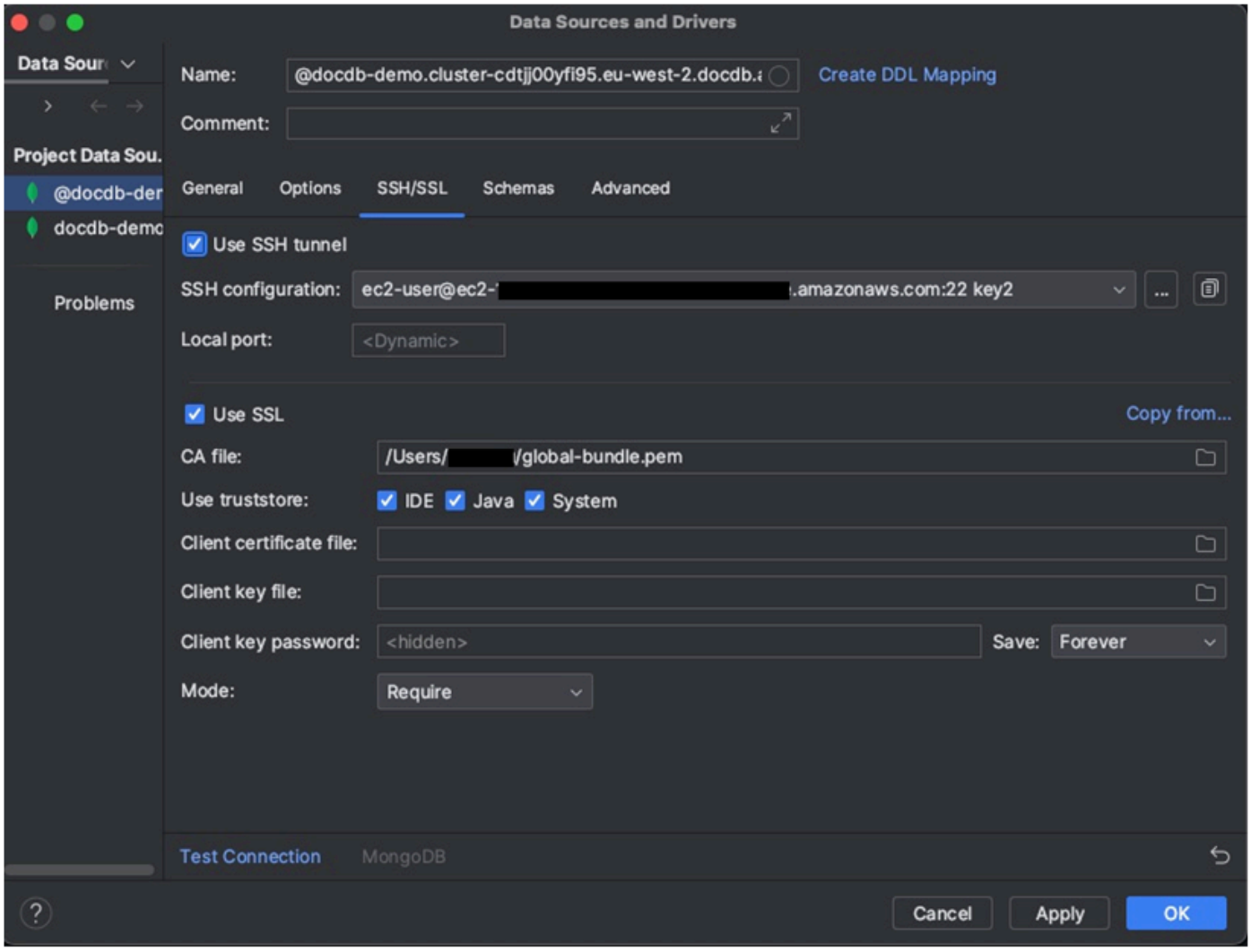
> **Note**
>
> Jika Anda menggunakan opsi VPN, tidak perlu mengkonfigurasi terowongan SSH.



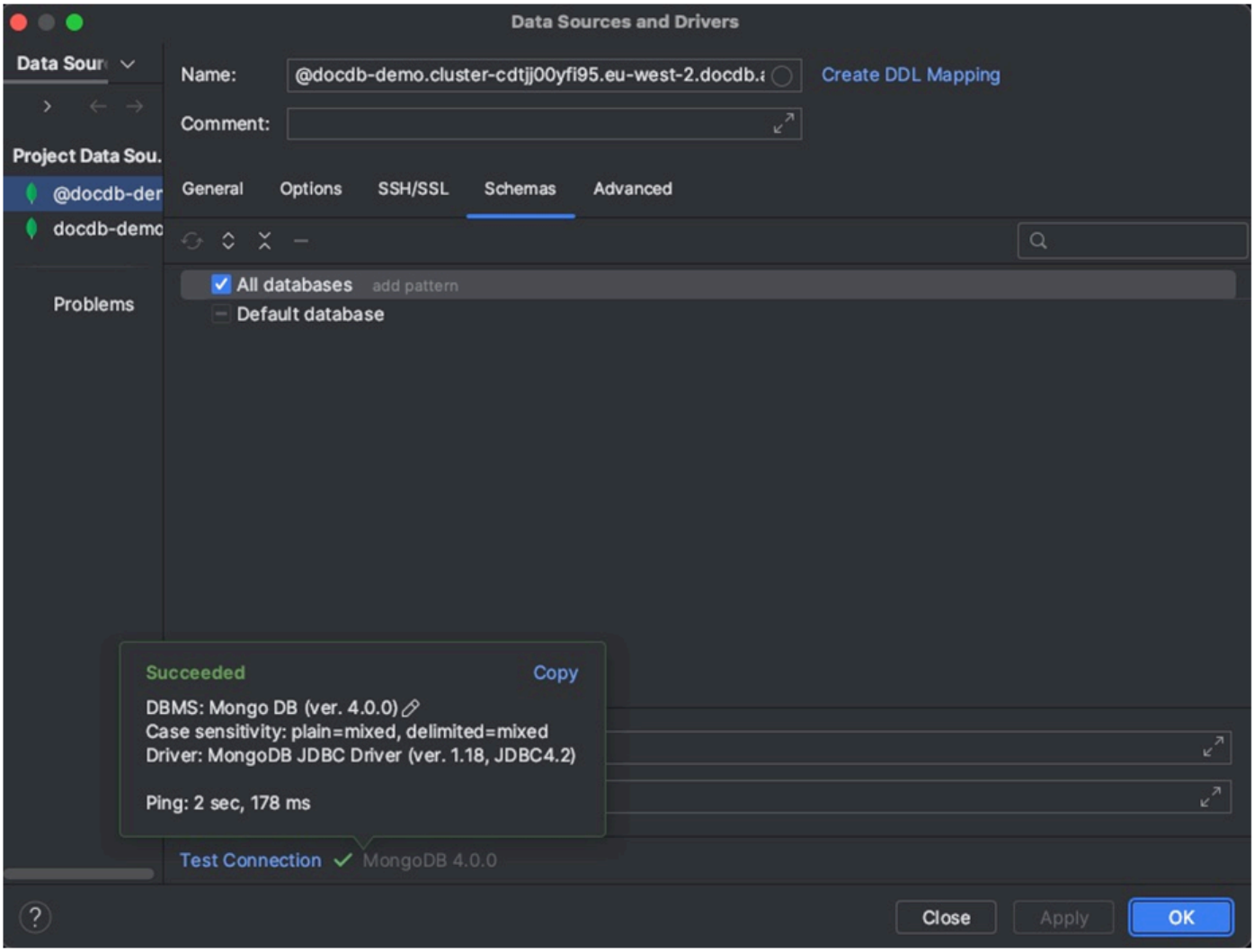
6. Di tab SSH/SSL, aktifkan Gunakan SSL. Di bidang file CA, masukkan lokasi ke `global-bundle.pem` file di komputer Anda. Untuk Mode, tinggalkan opsi Memerlukan.

> **Note**
>
> Anda dapat mengunduh sertifikat dari lokasi ini atau dengan perintah ini: wget https://aws.amazon.com/https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem

> **Note**
>
> Jika Anda terhubung ke Amazon DocumentDB cluster elastis, Anda tidak perlu menentukan file CA. Biarkan opsi Gunakan SSL dicentang dan semua opsi lainnya pada nilai defaultnya.



7. Di tab Skema, pilih Semua database atau masukkan filter “*: *” di bidang Pola skema. Klik tautan Test Connection untuk menguji koneksi.



8. Setelah koneksi berhasil diuji, klik OK untuk menyimpan konfigurasi sumber data.

DataGrip fitur

DataGrip menyediakan berbagai fitur untuk membantu Anda bekerja dengan Amazon DocumentDB secara efisien:

- SQL Editor - Tulis dan jalankan kueri seperti SQL pada koleksi DocumentDB Anda menggunakan editor SQL di. DataGrip
- Visual Query Builder - Gunakan pembangun kueri visual untuk membuat kueri secara grafis tanpa menulis kode SQL.
- Manajemen Skema - Kelola skema database Anda dengan mudah, termasuk membuat, mengubah, dan menjatuhkan koleksi.

- Visualisasi Data — Melihat dan menganalisis data Anda menggunakan berbagai alat visualisasi yang tersedia di. DataGrip
- Ekspor dan Impor Data - Transfer data antara Amazon DocumentDB dan database lain DataGrip menggunakan fitur ekspor dan impor.

Lihat DataGrip dokumentasi resmi untuk fitur dan kiat yang lebih canggih tentang bekerja dengan Amazon DocumentDB dan sistem basis data lainnya.

Connect menggunakan Amazon EC2

Bagian ini menjelaskan cara mengatur konektivitas antara cluster Amazon DocumentDB dan Amazon EC2 dan mengakses cluster Amazon DocumentDB dari instans Amazon EC2.

Ada dua opsi untuk mengkonfigurasi koneksi EC2:

- Sambungkan instans EC2 Anda secara otomatis ke database Amazon DocumentDB — Gunakan fitur koneksi otomatis di konsol EC2 untuk secara otomatis mengonfigurasi koneksi antara instans EC2 Anda dan database Amazon DocumentDB baru atau yang sudah ada. Koneksi ini memungkinkan lalu lintas untuk melakukan perjalanan antara instans EC2 dan database Amazon DocumentDB. Opsi ini biasanya digunakan untuk menguji dan membuat grup keamanan baru.
- Hubungkan instans EC2 Anda secara manual ke database Amazon DocumentDB Anda — Konfigurasikan koneksi antara instans EC2 Anda ke database Amazon DocumentDB Anda dengan mengonfigurasi dan menetapkan grup keamanan secara manual untuk mereproduksi konfigurasi yang dibuat oleh fitur koneksi otomatis. Opsi ini biasanya digunakan untuk mengubah pengaturan yang lebih maju dan menggunakan grup keamanan yang ada.

Prasyarat

Terlepas dari opsinya, dan sebelum Anda membuat cluster Amazon DocumentDB pertama Anda, Anda harus melakukan hal berikut:

Buatlah akun Amazon Web Services (AWS)

Sebelum Anda dapat mulai menggunakan Amazon DocumentDB, Anda harus memiliki akun Amazon Web Services (AWS). AWS Akun ini gratis. Anda hanya membayar untuk layanan dan sumber daya yang Anda gunakan.

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Secara optinal, atur izin yang diperlukan AWS Identity and Access Management (IAM).

Akses untuk mengelola sumber daya Amazon DocumentDB seperti cluster, instance, dan grup parameter cluster memerlukan kredensil yang dapat digunakan untuk AWS mengautentikasi permintaan Anda. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon DocumentDB.

1. Di bilah pencarian AWS Management Console, ketik IAM dan pilih IAM di menu tarik-turun yang muncul.
2. Setelah Anda berada di konsol IAM, pilih Pengguna dari panel navigasi.
3. Pilih nama pengguna Anda.
4. Klik tombol Tambahkan izin.
5. Pilih Lampirkan kebijakan yang ada secara langsung.
6. Ketik `AmazonDocDBFullAccess` di bilah pencarian dan pilih itu setelah muncul di hasil pencarian.
7. Klik tombol biru di bagian bawah yang bertuliskan Berikutnya: Tinjau.
8. Klik tombol biru di bagian bawah yang bertuliskan Tambahkan izin.

Buat Amazon Virtual Private Cloud (Amazon VPC)

Tergantung di mana Wilayah AWS Anda berada, Anda mungkin atau mungkin tidak memiliki VPC default yang sudah dibuat. Jika Anda tidak memiliki VPC default, selesaikan langkah 1 dari

Memulai Amazon VPC di Panduan Pengguna Amazon VPC. Ini akan memakan waktu kurang dari lima menit.

Hubungkan Amazon EC2 secara otomatis

Topik

- Secara otomatis menghubungkan instans EC2 ke database Amazon DocumentDB baru
- Secara otomatis menghubungkan instans EC2 ke database Amazon DocumentDB yang ada
- Ikhtisar konektivitas otomatis dengan instans EC2
- Melihat sumber daya komputasi terhubung

Sebelum menyiapkan sambungan antara instans EC2 dan database Amazon DocumentDB baru, pastikan Anda memenuhi persyaratan yang dijelaskan dalam. Ikhtisar konektivitas otomatis dengan instans EC2 Jika Anda membuat perubahan pada grup keamanan setelah mengonfigurasi konektivitas, perubahan tersebut dapat memengaruhi koneksi antara instans EC2 dan database Amazon DocumentDB.

> **Note**
>
> Anda hanya dapat mengatur koneksi antara instans EC2 dan database Amazon DocumentDB secara otomatis dengan menggunakan file. AWS Management Console Anda tidak dapat mengatur koneksi secara otomatis dengan Amazon DocumentDB API AWS CLI atau Amazon.

Secara otomatis menghubungkan instans EC2 ke database Amazon DocumentDB baru

Proses berikut mengasumsikan Anda telah menyelesaikan langkah-langkah dalam Prasyarat topik.

Langkah-langkah

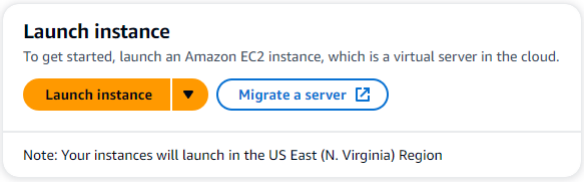
- Langkah 1: Buat instans Amazon EC2
- Langkah 2: Buat cluster Amazon DocumentDB
- Langkah 3: Connect ke instans Amazon EC2 Anda
- Langkah 4: Instal shell mongo

- Langkah 5: Kelola Amazon DocumentDB TLS
- Langkah 6: Connect ke cluster Amazon DocumentDB Anda
- Langkah 7: Masukkan dan kueri data
- Langkah 8: Jelajahi

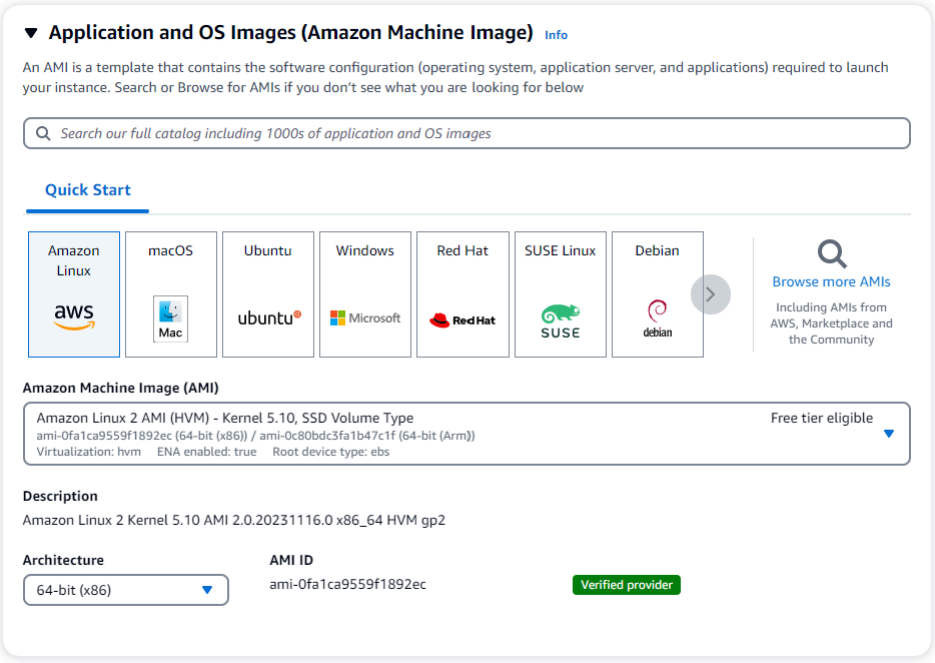
Langkah 1: Buat instans Amazon EC2

Pada langkah ini, Anda akan membuat instans Amazon EC2 di Wilayah yang sama dan Amazon VPC yang nantinya akan Anda gunakan untuk menyediakan klaster Amazon DocumentDB Anda.

1. Pada konsol Amazon EC2, pilih Luncurkan instans.

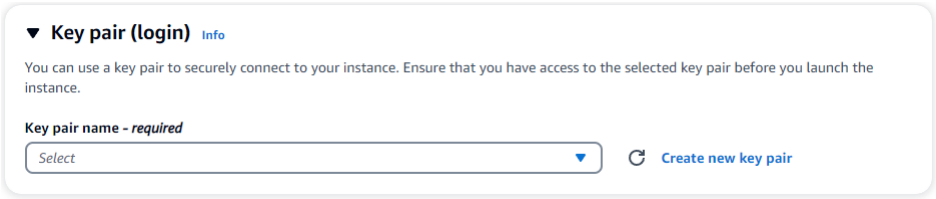


2. Masukkan nama atau pengenal di bidang Nama yang terletak di bagian Nama dan tag.
3. Di daftar drop-down Amazon Machine Image (AMI), cari Amazon Linux 2 AMI dan pilih.



4. Cari dan pilih t3.micro dalam daftar drop-down tipe Instance.

5. Di bagian Key pair (login), masukkan identifier dari key-pair yang ada, atau pilih Create new key pair.



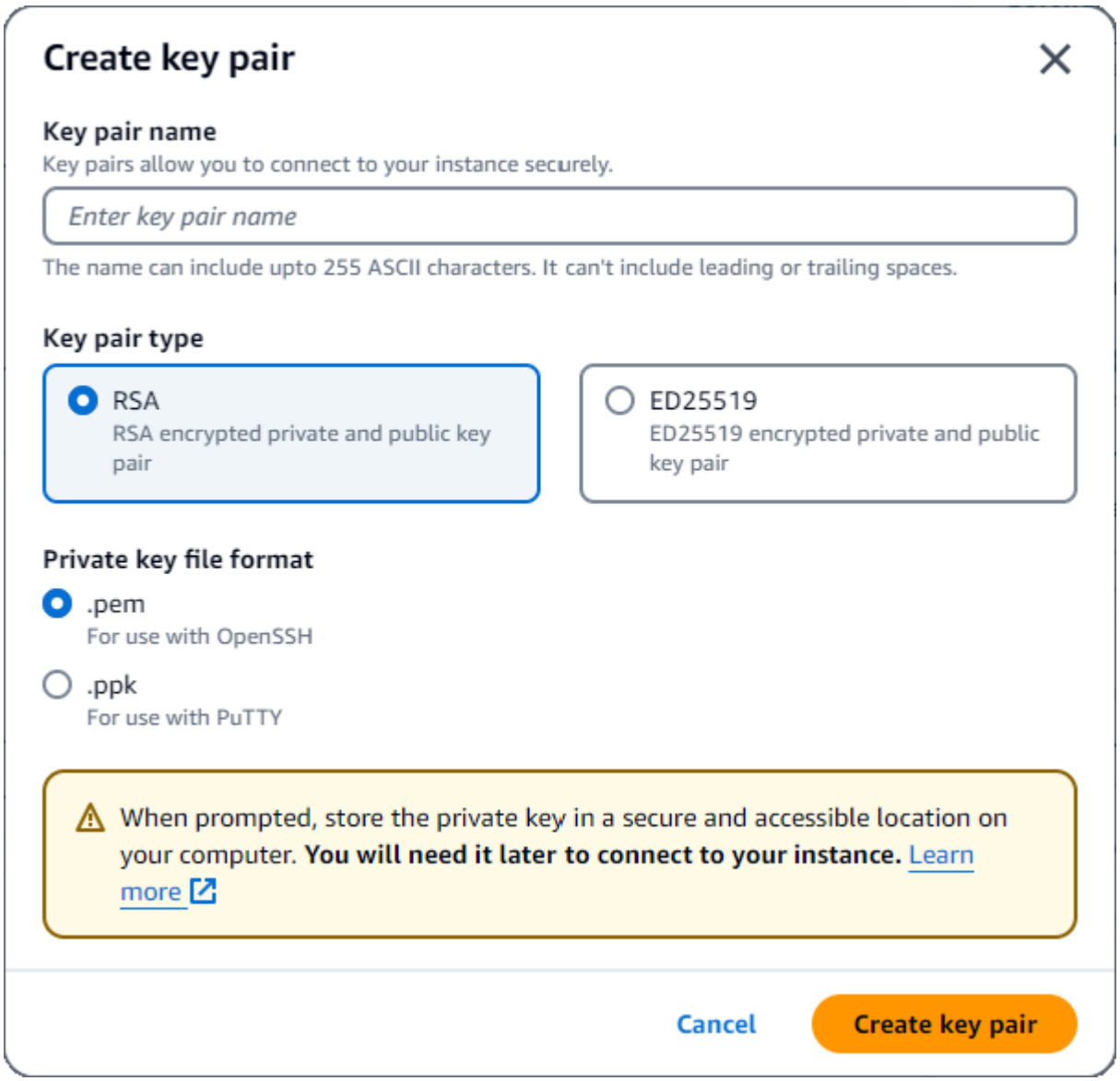
   Anda harus menyediakan pasangan kunci Amazon EC2.

   Jika Anda memiliki pasangan kunci Amazon EC2:

   a. Pilih pasangan kunci, pilih pasangan kunci Anda dari daftar.
   b. Anda harus sudah memiliki file kunci pribadi (file.pem atau .ppk) yang tersedia untuk masuk ke instans Amazon EC2 Anda.

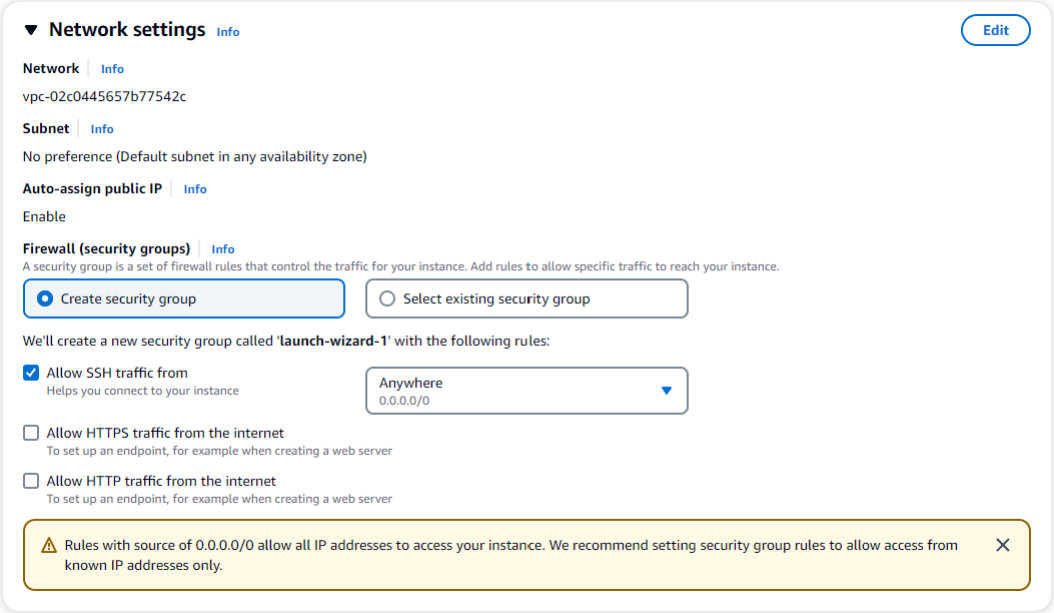
   Jika Anda tidak memiliki pasangan kunci Amazon EC2:

   a. Pilih Create new key pair, kotak dialog Create key pair muncul.
   b. Masukkan nama di bidang Nama pasangan kunci.
   c. Pilih jenis pasangan kunci dan format file kunci pribadi.
   d. Pilih Buat pasangan kunci.



> **Note**
>
> Untuk tujuan keamanan, kami sangat merekomendasikan penggunaan pasangan kunci untuk konektivitas SSH dan internet ke instans EC2 Anda.

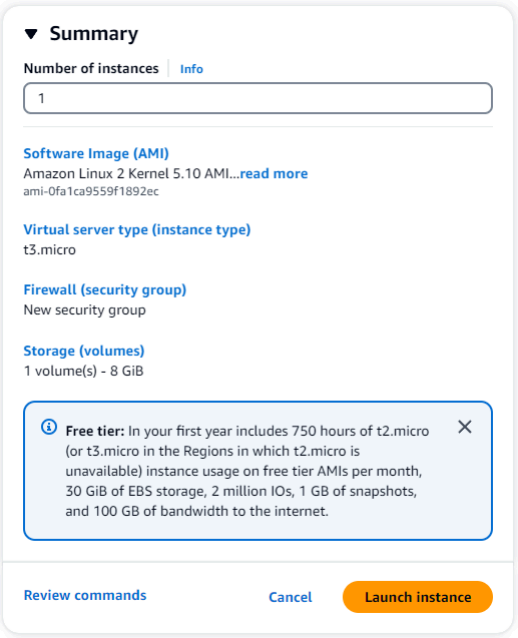
6. Opsional: Di bagian Network seetings, di bawah Firewall (grup keamanan), pilih Buat grup keamanan atau Pilih grup keamanan yang ada.



Jika Anda memilih untuk memilih grup keamanan yang ada, pilih salah satu dari daftar drop-down grup keamanan umum.

Jika Anda memilih untuk membuat grup keamanan baru, periksa semua aturan izin lalu lintas yang berlaku untuk konektivitas EC2 Anda.

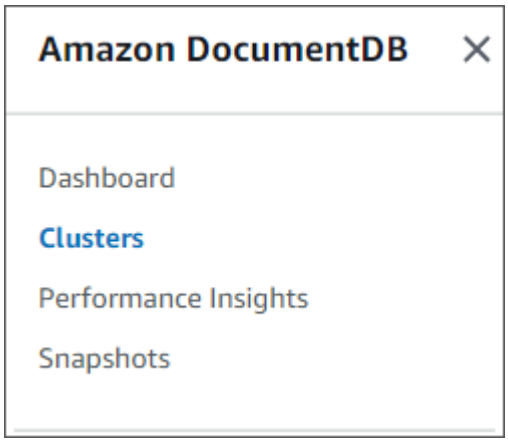
7. Di bagian Ringkasan, tinjau konfigurasi EC2 Anda dan pilih Launch instance jika benar. Edit grup keamanan.



Langkah 2: Buat cluster Amazon DocumentDB

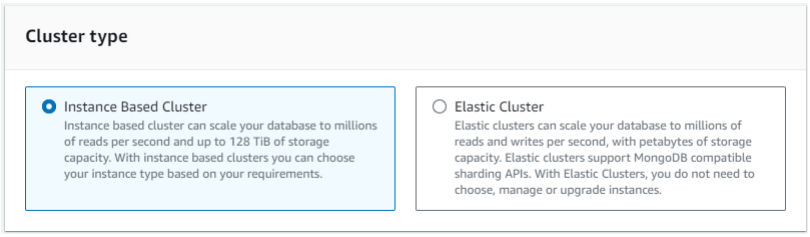
Saat instans Amazon EC2 sedang disediakan, Anda akan membuat klaster Amazon DocumentDB Anda.

1. Navigasikan ke konsol Amazon DocumentDB dan pilih Klaster dari panel navigasi.

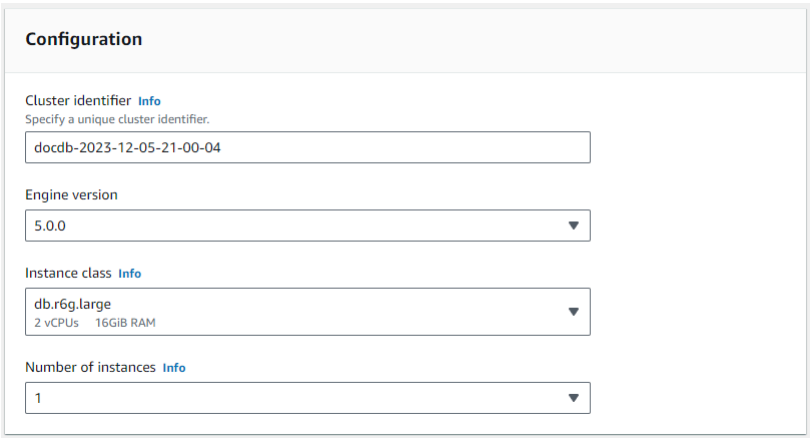


2. Pilih Buat.

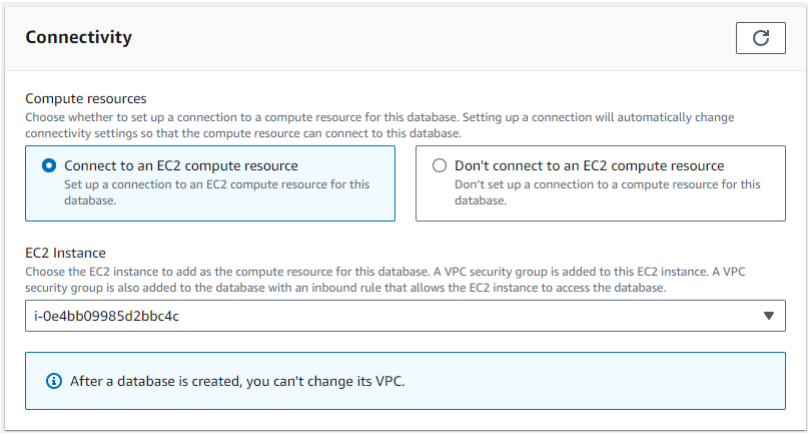
3. Biarkan pengaturan tipe Cluster pada defaultnya dari Instance Based Cluster.



4. Untuk Jumlah instans, pilih 1. Ini akan meminimalkan biaya. Biarkan pengaturan lain di defaultnya.

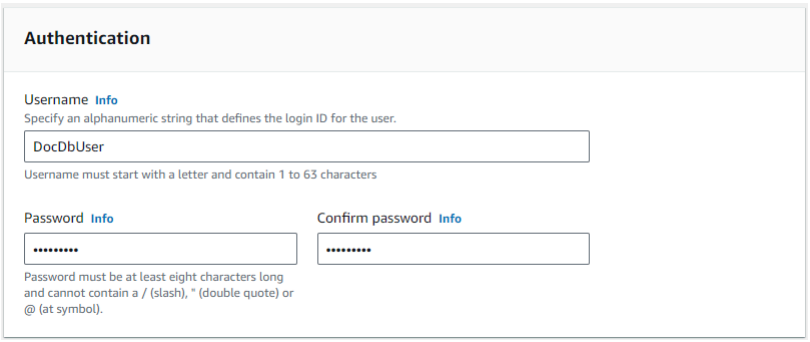


5. Untuk Konektivitas, pilih Connect ke sumber daya komputasi EC2. Ini adalah contoh EC2 yang Anda buat di Langkah 1.

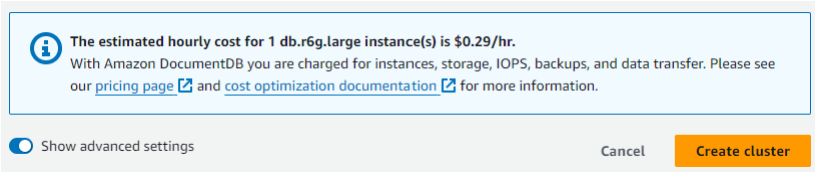


> **Note**
>
> Menyambung ke sumber daya komputasi EC2 secara otomatis akan membuat grup keamanan untuk koneksi sumber daya komputasi EC2 ke klaster Amazon DocumentDB Anda. Setelah selesai membuat klaster dan ingin melihat grup keamanan yang baru dibuat, navigasikan ke daftar klaster dan pilih pengenal klaster Anda. Di tab Konektivitas & keamanan, buka Grup Keamanan dan temukan grup Anda di bawah Nama grup keamanan (ID). Ini akan berisi nama cluster Anda dan terlihat mirip dengan ini:`docdb-ec2-docdb-2023-12-11-21-33-41:i-0e4bb09985d2bbc4c (sg-0238e0b0bf0f73877)`.

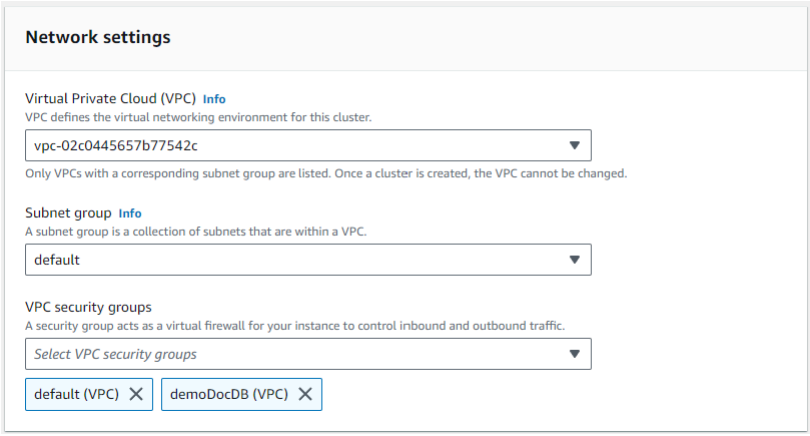
6. Untuk Otentikasi, masukkan kredensyal masuk. Penting: Anda akan memerlukan kredensyal masuk untuk mengautentikasi klaster Anda di langkah selanjutnya.



7. Hidupkan Tampilkan pengaturan lanjutan.



8. Di bagian Pengaturan jaringan, untuk Grup keamanan Amazon VPC, pilih demoDocDB.

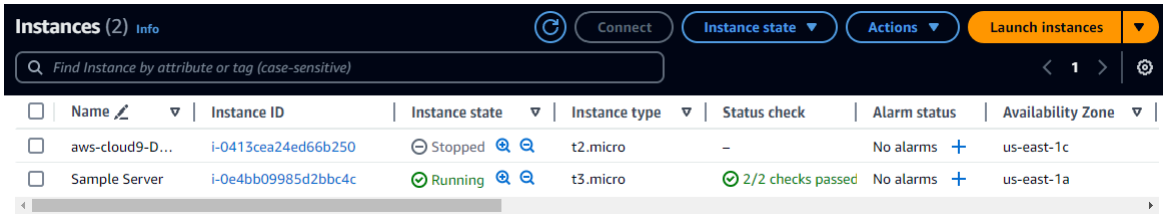


9. Pilih Buat klaster.

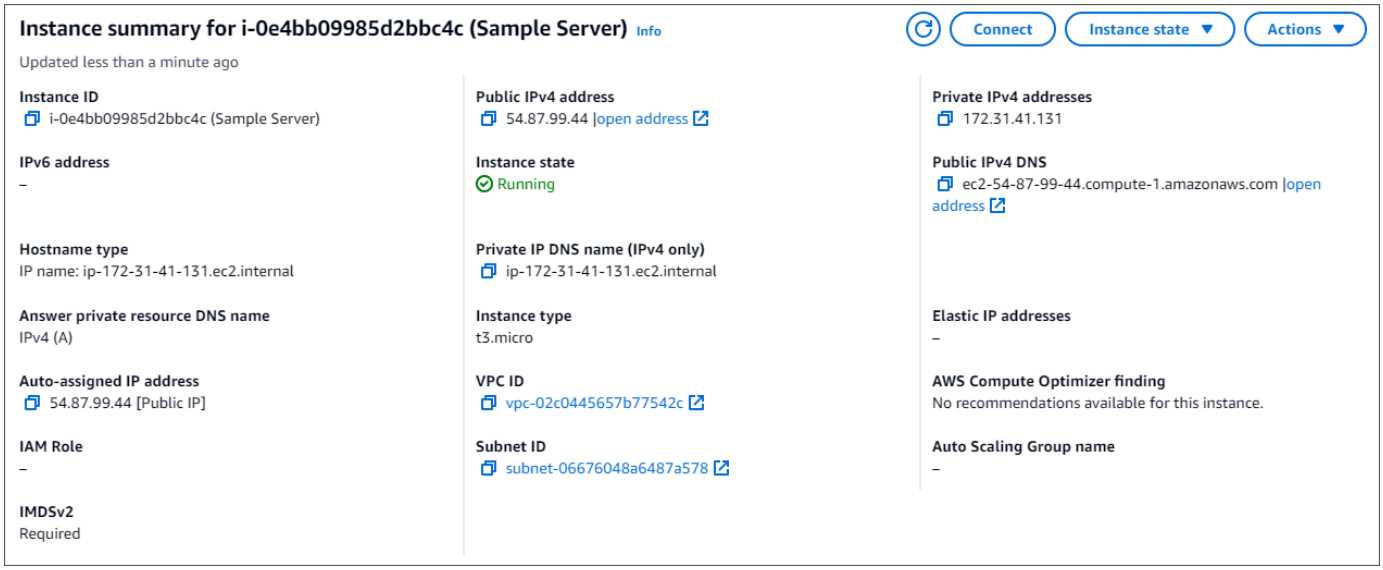
Langkah 3: Connect ke instans Amazon EC2 Anda

Untuk menginstal shell mongo, Anda harus terlebih dahulu terhubung ke instans Amazon EC2 Anda. Instalasi shell mongo memungkinkan Anda untuk terhubung ke dan kueri klaster Amazon DocumentDB Anda. Selesaikan langkah-langkah berikut:

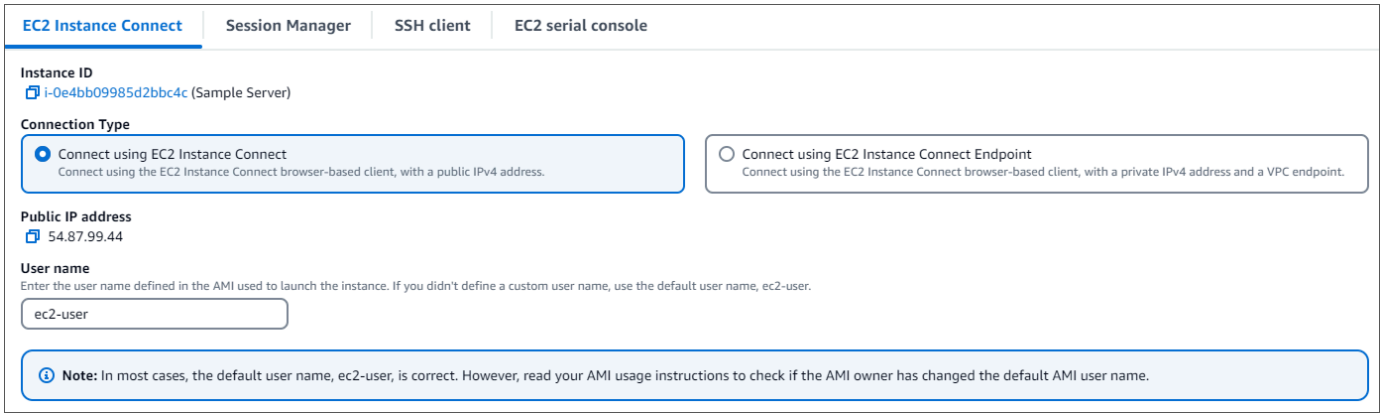
1. Pada konsol Amazon EC2, navigasikan ke instans Anda dan lihat apakah instans yang baru saja Anda buat berjalan. Jika ya, pilih instance dengan mengklik ID instance.



2. Pilih Hubungkan.



3. Ada empat opsi tab untuk metode koneksi Anda: Amazon EC2 Instance Connect, Session Manager, SSH client, atau konsol serial EC2. Anda harus memilih satu dan mengikuti instruksinya. Setelah selesai, pilih Connect.



> **Note**
>
> Jika alamat IP Anda berubah setelah Anda memulai panduan ini, atau Anda akan kembali ke lingkungan Anda di lain waktu, Anda harus memperbarui aturan masuk grup keamanan demoEC2 Anda untuk mengaktifkan lalu lintas masuk dari alamat API baru Anda.

Langkah 4: Instal shell mongo

Anda sekarang dapat menginstal shell mongo, yang merupakan utilitas baris perintah yang Anda gunakan untuk menghubungkan dan mengajukan kueri pada klaster Amazon DocumentDB Anda. Ikuti petunjuk di bawah ini untuk menginstal shell mongo untuk sistem operasi Anda.

On Amazon Linux

Untuk menginstal shell mongo di Amazon Linux

1. Buat file repositori. Di baris perintah instans EC2 Anda, jalankan perintah berikut:

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-5.0.repo
```

2. Setelah selesai, instal shell mongo dengan menjalankan perintah berikut:

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

On Ubuntu 18.04

Untuk menginstal shell mongo di Ubuntu 18.04

1. Impor kunci publik yang akan digunakan oleh sistem manajemen paket.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. Buat file daftar `/etc/apt/sources.list.d/mongodb-org-3.6.list` untuk MongoDB menggunakan perintah yang sesuai untuk versi Ubuntu Anda.

   Ubuntu 18.04

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/
mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
org-3.6.list
```

> **Note**
>
> Perintah di atas akan menginstal shell mongo 3.6 untuk Bionic dan Xenial.

3. Muat ulang basis data paket lokal menggunakan perintah berikut:

```
sudo apt-get update
```

4. Instal shell MongoDB.

```
sudo apt-get install -y mongodb-org-shell
```

Untuk informasi tentang menginstal versi sebelumnya dari MongoDB pada sistem Ubuntu Anda, lihat Instal MongoDB Community Edition di Ubuntu.

On other operating systems

Untuk menginstal shell mongo pada sistem operasi lain, lihat Instal MongoDB Community Edition dalam dokumentasi MongoDB.

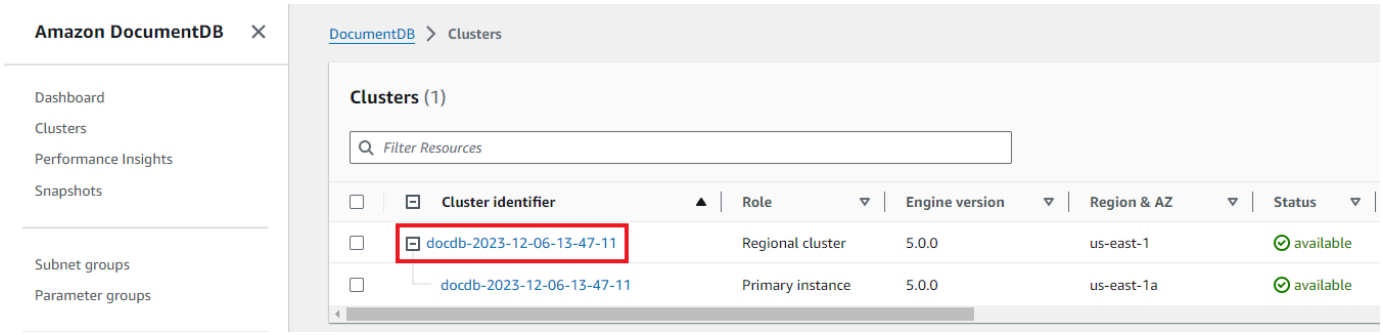
Langkah 5: Kelola Amazon DocumentDB TLS

Unduh sertifikat CA untuk Amazon DocumentDB dengan kode berikut: `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

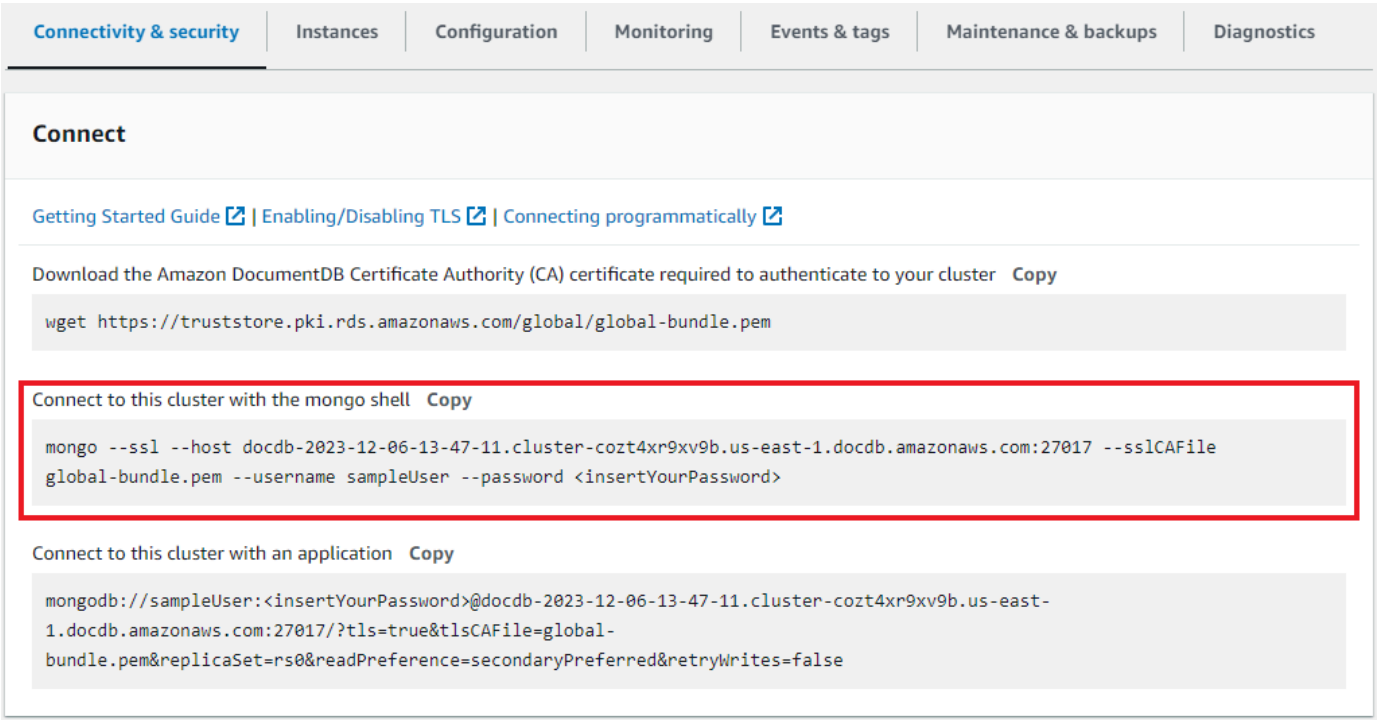
> **Note**
>
> Keamanan Lapisan Pengangkutan (TLS) diaktifkan secara default untuk setiap klaster Amazon DocumentDB baru. Untuk informasi selengkapnya, lihat Mengelola setelan TLS cluster Amazon DocumentDB.

Langkah 6: Connect ke cluster Amazon DocumentDB Anda

1. Di konsol Amazon DocumentDB, di bawah Clusters, temukan klaster Anda. Pilih cluster yang Anda buat dengan mengklik pengenal Cluster.



2. Di tab Konektivitas dan keamanan, cari Connect to this cluster dengan shell mongo di kotak Connect:



Salin string koneksi yang disediakan dan tempel ke terminal Anda.

Buat perubahan berikut untuk itu:

a. Pastikan Anda memiliki nama pengguna yang benar dalam string.

b. Hilangkan `<insertYourPassword>` sehingga Anda diminta kata sandi oleh shell mongo saat Anda terhubung.

String koneksi Anda akan terlihat serupa dengan berikut ini:

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem
--username demoUser --password
```

3. Tekan enter di terminal Anda. Anda sekarang diminta untuk kata sandi Anda. Masukkan kata sandi Anda.
4. Saat Anda memasukkan kata sandi dan dapat melihat perintah `rs0:PRIMARY>`, Anda berhasil terhubung ke klaster Amazon DocumentDB Anda.

Mengalami masalah saat menghubungkan? Lihat Pemecahan Masalah Amazon DocumentDB.

Langkah 7: Masukkan dan kueri data

Sekarang setelah Anda terhubung ke klaster, Anda dapat menjalankan beberapa kueri untuk mengenali penggunaan basis data dokumen.

1. Untuk memasukkan satu dokumen, masukkan hal berikut ini:

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

2. Anda mendapatkan output berikut ini:

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

3. Anda dapat membaca dokumen yang Anda tulis dengan perintah `findOne()` (karena hanya mengembalikan satu dokumen). Masukkan yang berikut ini:

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

4. Anda mendapatkan output berikut ini:

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

5. Untuk mengajukan beberapa kueri lagi, pertimbangkan kasus penggunaan profil game. Pertama, masukkan beberapa entri ke dalam koleksi berjudul `profiles`. Masukkan yang berikut ini:

```
db.profiles.insertMany([
           { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
```

```
            { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,
"score":9},
            { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
"score":87},
            { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
            ])
```

6. Anda mendapatkan output berikut ini:

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

7. Gunakan perintah `find()` untuk mengembalikan semua dokumen dalam koleksi profil. Masukkan yang berikut ini:

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

8. Anda akan mendapatkan output yang akan cocok dengan data yang Anda ketik di Langkah 5.
9. Gunakan kueri untuk satu dokumen dengan menggunakan filter. Masukkan yang berikut ini:

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

10. Anda harus mendapatkan kembali output ini:

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

11. Sekarang mari kita coba untuk menemukan profil dan memodifikasinya dengan menggunakan perintah `findAndModify`. Kami akan memberi pengguna Matt sepuluh poin tambahan dengan kode berikut:

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

12. Anda mendapatkan output berikut (perhatikan bahwa skornya belum meningkat):

```
    {
    "_id" : 1,
```

```
    "name" : "Matt",
    "status" : "active",
    "level" : 12,
    "score" : 202
    }
```

13. Anda dapat memverifikasi bahwa skornya telah berubah dengan kueri berikut:

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

14. Anda mendapatkan output berikut ini:

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

Langkah 8: Jelajahi

Selamat! Anda telah berhasil menyelesaikan Panduan Quick Start untuk Amazon DocumentDB.

Apa selanjutnya? Pelajari cara memanfaatkan sepenuhnya basis data yang kuat ini dengan beberapa fitur populernya:

- Mengelola Amazon DocumentDB
- Penskalaan
- Mencadangkan dan memulihkan

> **Note**
>
> Untuk menghemat biaya, Anda dapat menghentikan klaster Amazon DocumentDB untuk mengurangi biaya atau menghapus klaster. Secara default, setelah 30 menit tidak aktif, AWS Cloud9 lingkungan Anda akan menghentikan instans Amazon EC2 yang mendasarinya.

Secara otomatis menghubungkan instans EC2 ke database Amazon DocumentDB yang ada

Prosedur berikut mengasumsikan Anda memiliki klaster Amazon DocumentDB yang sudah ada dan instans Amazon EC2 yang sudah ada.

Akses kluster Amazon DocumentDB Anda dan atur koneksi Amazon EC2

1. Akses cluster Amazon DocumentDB Anda.

   a. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.

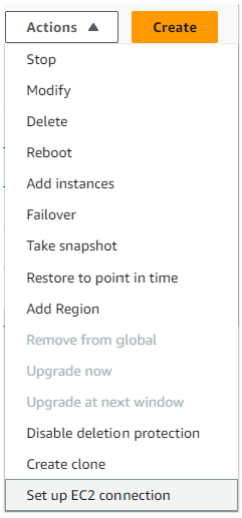
   b. Di panel navigasi, pilih Klaster.

   > **Tip**
   >
   > Jika Anda tidak melihat panel navigasi di sisi kiri layar, pilih ikon menu (≡) di pojok kiri atas halaman.

   c. Tentukan cluster yang Anda inginkan dengan memilih tombol di sebelah kiri nama cluster.

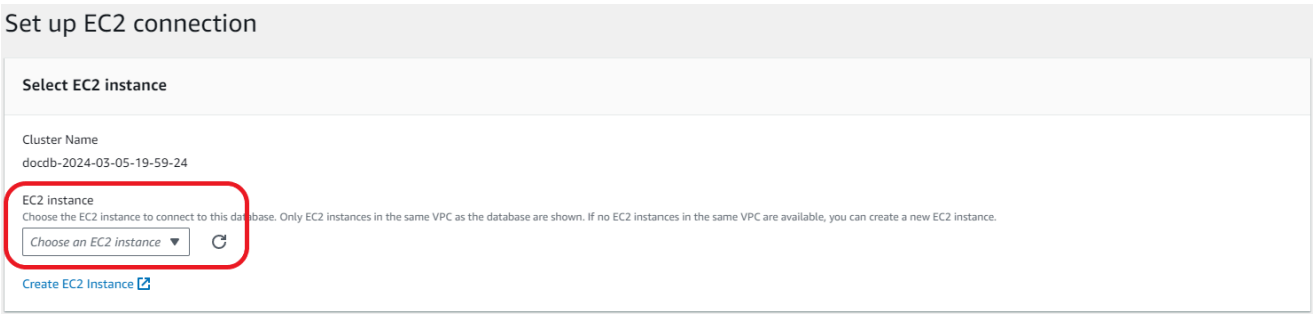
2. Siapkan koneksi Amazon EC2.

   a. Pilih Tindakan, lalu pilih Siapkan koneksi EC2.



   Dialog Mengatur koneksi EC2 muncul.

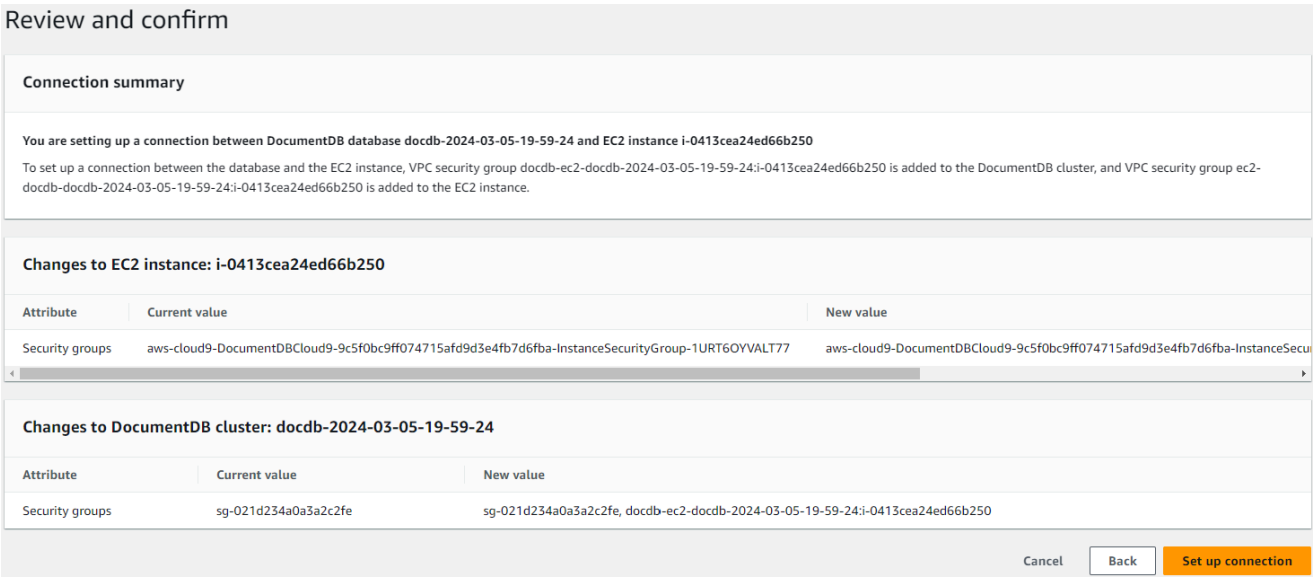
   b. Di bidang instans EC2, pilih instans EC2 yang ingin Anda sambungkan ke cluster Anda.



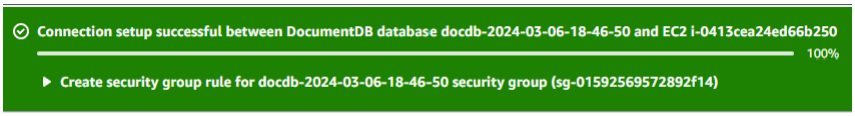
c. Pilih Lanjutkan.

Dialog Tinjau dan konfirmasi muncul.

d. Pastikan perubahannya benar. Kemudian pilih Siapkan koneksi.



Jika berhasil, verifikasi berikut akan muncul:



Ikhtisar konektivitas otomatis dengan instans EC2

Saat Anda mengatur koneksi antara instans EC2 dan database Amazon DocumentDB, Amazon DocumentDB secara otomatis mengonfigurasi grup keamanan VPC untuk instans EC2 Anda dan untuk database Amazon DocumentDB Anda.

Berikut ini adalah persyaratan untuk menghubungkan instans EC2 dengan database Amazon DocumentDB:

- Instans EC2 harus ada di VPC yang sama dengan database Amazon DocumentDB.

  Jika tidak ada instans EC2 di VPC yang sama, maka konsol menyediakan tautan untuk membuatnya.

- Pengguna yang menyiapkan konektivitas harus memiliki izin untuk melakukan operasi-operasi Amazon EC2 berikut:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Jika instans basis data dan instans EC2 berada di Zona Ketersediaan yang berbeda, akun Anda mungkin dikenakan biaya lintas Zona Ketersediaan.

Saat Anda menyiapkan sambungan ke instans EC2, Amazon DocumentDB bertindak sesuai dengan konfigurasi grup keamanan saat ini yang terkait dengan database Amazon DocumentDB dan instans EC2, seperti yang dijelaskan dalam tabel berikut:

Konfigurasi grup keamanan Amazon DocumentDB saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan Amazon DocumentDB
Ada satu atau lebih grup keamanan yang terkait dengan database Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan	Ada satu atau lebih grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `DocumentDB-ec2-n` (di mana n adalah angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan	Amazon DocumentDB tidak mengambil tindakan. Koneksi sudah dikonfigurasi secara otomatis antara instans EC2 dan database Amazon DocumentDB. Karena koneksi sudah ada antara instans EC2 dan database Amazon DocumentDB, grup keamanan tidak dimodifikasi.

Konfigurasi grup keamanan Amazon DocumentDB saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan Amazon DocumentDB
grup keamanan VPC instans EC2 sebagai sumbernya.	grup keamanan VPC dari database Amazon DocumentDB sebagai sumbernya.	

Konfigurasi grup keamanan Amazon DocumentDB saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan Amazon DocumentDB
Salah satu syarat berikut berlaku: • Tidak ada grup keamanan yang terkait dengan database Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` • Ada satu atau lebih grup keamanan yang terkait dengan Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` Namun, Amazon DocumentDB tidak dapat menggunakan salah satu grup keamanan ini untuk koneksi dengan instans EC2. Amazon DocumentDB tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC dari instans EC2 sebagai sumbernya. Amazon DocumentDB juga tidak dapat menggunakan grup keamanan yang telah dimodifikasi. Contoh-contoh perubahan meliputi penambahan aturan atau	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-DocumentDB-n` . • Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-DocumentDB-n`. Namun, Amazon DocumentDB tidak dapat menggunakan salah satu grup keamanan ini untuk koneksi dengan database Amazon DocumentDB. Amazon DocumentDB tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC dari database Amazon DocumentDB sebagai sumbernya. Amazon DocumentDB juga tidak dapat menggunakan grup keamanan yang telah dimodifikasi.	Tindakan Amazon DocumentDB: buat grup keamanan baru

Konfigurasi grup keamanan Amazon DocumentDB saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan Amazon DocumentDB
pengubahan port aturan yang ada.		
Ada satu atau lebih grup keamanan yang terkait dengan database Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-DocumentDB-n` . Namun, Amazon DocumentDB tidak dapat menggunakan salah satu grup keamanan ini untuk koneksi dengan database Amazon DocumentDB. Amazon DocumentDB tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC dari database Amazon DocumentDB sebagai sumbernya. Amazon DocumentDB juga tidak dapat menggunakan grup keamanan yang telah dimodifikasi.	Tindakan Amazon DocumentDB: buat grup keamanan baru

Konfigurasi grup keamanan Amazon DocumentDB saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan Amazon DocumentDB
Ada satu atau lebih grup keamanan yang terkait dengan database Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya.	Ada grup keamanan EC2 yang valid untuk koneksi, tetapi tidak terkait dengan instans EC2. Grup keamanan ini memiliki nama yang cocok dengan pola `DocumentDB-ec2-n`. Grup tersebut belum diubah. Ini hanya memiliki satu aturan keluar dengan grup keamanan VPC dari database Amazon DocumentDB sebagai sumbernya.	Tindakan Amazon DocumentDB: asosiasikan grup keamanan EC2

Konfigurasi grup keamanan Amazon DocumentDB saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan Amazon DocumentDB
Salah satu syarat berikut berlaku: • Tidak ada grup keamanan yang terkait dengan database Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` • Ada satu atau lebih grup keamanan yang terkait dengan database Amazon DocumentDB dengan nama yang cocok dengan polanya. `DocumentDB-ec2-n` Namun, Amazon DocumentDB tidak dapat menggunakan salah satu grup keamanan ini untuk koneksi dengan instans EC2. Amazon DocumentDB tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC dari instans EC2 sebagai sumbernya. Amazon DocumentDB juga tidak dapat menggunakan grup keamanan yang telah dimodifikasi.	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `DocumentDB-ec2-n` . Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC dari database Amazon DocumentDB sebagai sumbernya.	Tindakan Amazon DocumentDB: buat grup keamanan baru

Tindakan Amazon DocumentDB: buat grup keamanan baru

Amazon DocumentDB mengambil tindakan berikut:

- Membuat grup keamanan baru yang cocok dengan pola `DocumentDB-ec2-n`. Grup keamanan ini memiliki aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Grup keamanan ini dikaitkan dengan database Amazon DocumentDB dan memungkinkan instans EC2 untuk mengakses database Amazon DocumentDB.
- Membuat grup keamanan baru yang cocok dengan pola `ec2-DocumentDB-n`. Grup keamanan ini memiliki aturan keluar dengan grup keamanan VPC dari database Amazon DocumentDB sebagai sumbernya. Grup keamanan ini dikaitkan dengan instans EC2 dan memungkinkan instans EC2 mengirim lalu lintas ke database Amazon DocumentDB.

Tindakan Amazon DocumentDB: asosiasikan grup keamanan EC2

Amazon DocumentDB mengaitkan grup keamanan EC2 yang valid dan sudah ada dengan instans EC2. Grup keamanan ini memungkinkan instans EC2 untuk mengirim lalu lintas ke database Amazon DocumentDB.

Melihat sumber daya komputasi terhubung

Anda dapat menggunakan AWS Management Console untuk melihat sumber daya komputasi yang terhubung ke database Amazon DocumentDB. Sumber daya yang ditampilkan meliputi koneksi sumber daya komputasi yang disiapkan secara otomatis. Anda dapat menyiapkan secara otomatis konektivitas dengan sumber daya komputasi dengan cara berikut:

- Anda dapat memilih sumber daya komputasi saat membuat basis data. Untuk informasi selengkapnya, lihat Membuat cluster Amazon DocumentDB dan Membuat klaster DB multi-AZ.
- Anda dapat menyiapkan konektivitas antara basis data yang ada dan sumber daya komputasi. Untuk informasi selengkapnya, lihat Hubungkan Amazon EC2 secara otomatis.

Sumber daya komputasi yang tercantum tidak menyertakan sumber daya yang dihubungkan secara manual dengan basis data. Misalnya, Anda dapat mengizinkan sumber daya komputasi untuk mengakses basis data secara manual dengan menambahkan aturan ke grup keamanan VPC yang terkait dengan basis data.

Agar sumber daya komputasi tercantum, syarat-syarat berikut harus dipenuhi:

- Nama grup keamanan yang terkait dengan sumber daya komputasi cocok dengan pola `ec2-DocumentDB-n` (di mana n adalah angka).
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan rentang port yang disetel ke port yang digunakan database Amazon DocumentDB.
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan sumber yang disetel ke grup keamanan yang terkait dengan database Amazon DocumentDB.
- Nama grup keamanan yang terkait dengan database Amazon DocumentDB cocok dengan `DocumentDB-ec2-n` pola (di mana n adalah angka).
- Grup keamanan yang terkait dengan database Amazon DocumentDB memiliki aturan masuk dengan rentang port yang disetel ke port yang digunakan database Amazon DocumentDB.
- Grup keamanan yang terkait dengan database Amazon DocumentDB memiliki aturan masuk dengan sumber yang disetel ke grup keamanan yang terkait dengan sumber daya komputasi.

Untuk melihat sumber daya komputasi yang terhubung ke database Amazon DocumentDB

1. Masuk ke AWS Management Console, dan buka konsol Amazon DocumentDB di https://console.aws.amazon.com/docdb.
2. Di panel navigasi, pilih Databases, lalu pilih nama database Amazon DocumentDB.
3. Pada tab Konektivitas & keamanan, lihat sumber daya komputasi di bagian Sumber daya komputasi yang terhubung.

Hubungkan Amazon EC2 secara manual

Topik

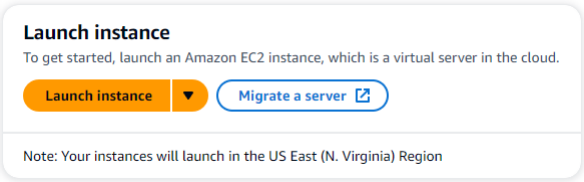
- Langkah 1: Buat instans Amazon EC2
- Langkah 2: Buat grup keamanan
- Langkah 3: Buat klaster Amazon DocumentDB
- Langkah 4: Hubungkan ke instans Amazon EC2 Anda
- Langkah 5: Instal shell mongo
- Langkah 6: Kelola Amazon DocumentDB TLS
- Langkah 7: Hubungkan ke klaster Amazon DocumentDB Anda
- Langkah 8: Masukkan dan kueri data
- Langkah 9: Jelajahi

Langkah-langkah berikut mengasumsikan Anda telah menyelesaikan langkah-langkah dalam Prasyarat topik.

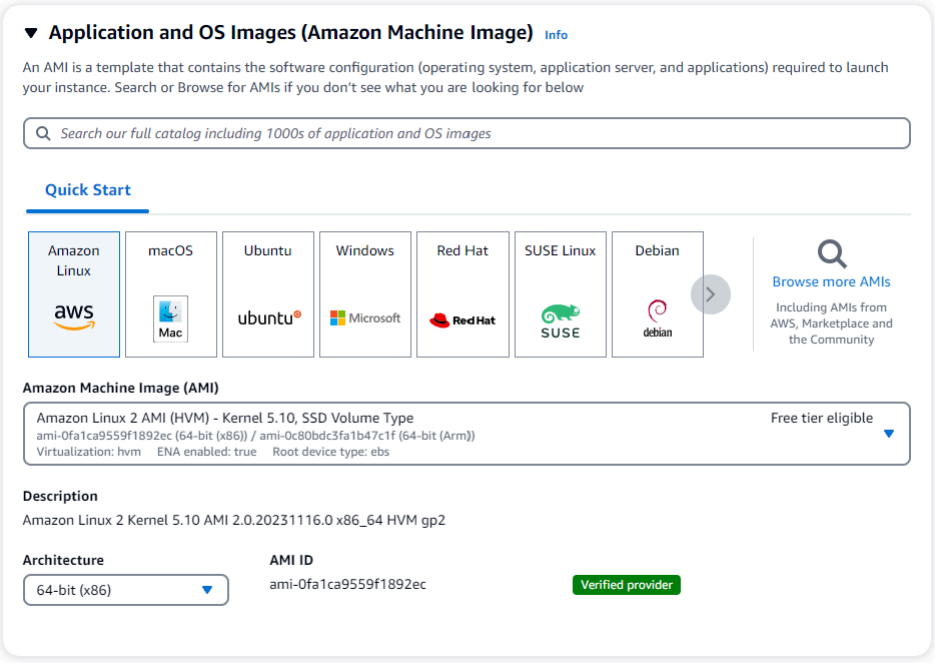
Langkah 1: Buat instans Amazon EC2

Pada langkah ini, Anda akan membuat instans Amazon EC2 di Wilayah yang sama dan Amazon VPC yang nantinya akan Anda gunakan untuk menyediakan klaster Amazon DocumentDB Anda.

1. Pada konsol Amazon EC2, pilih Luncurkan instans.

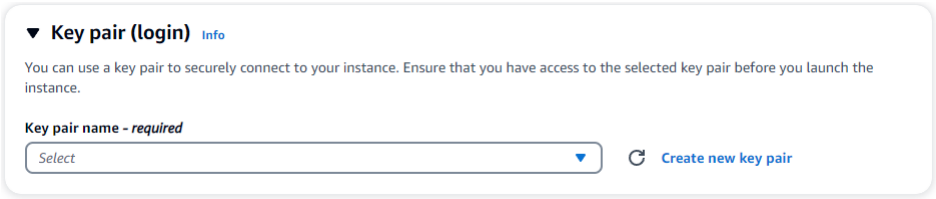


2. Masukkan nama atau pengenal di bidang Nama yang terletak di bagian Nama dan tag.
3. Di daftar drop-down Amazon Machine Image (AMI), cari Amazon Linux 2 AMI dan pilih.



4. Cari dan pilih t3.micro dalam daftar drop-down tipe Instance.

5. Di bagian Key pair (login), masukkan identifier dari key-pair yang ada, atau pilih Create new key pair.



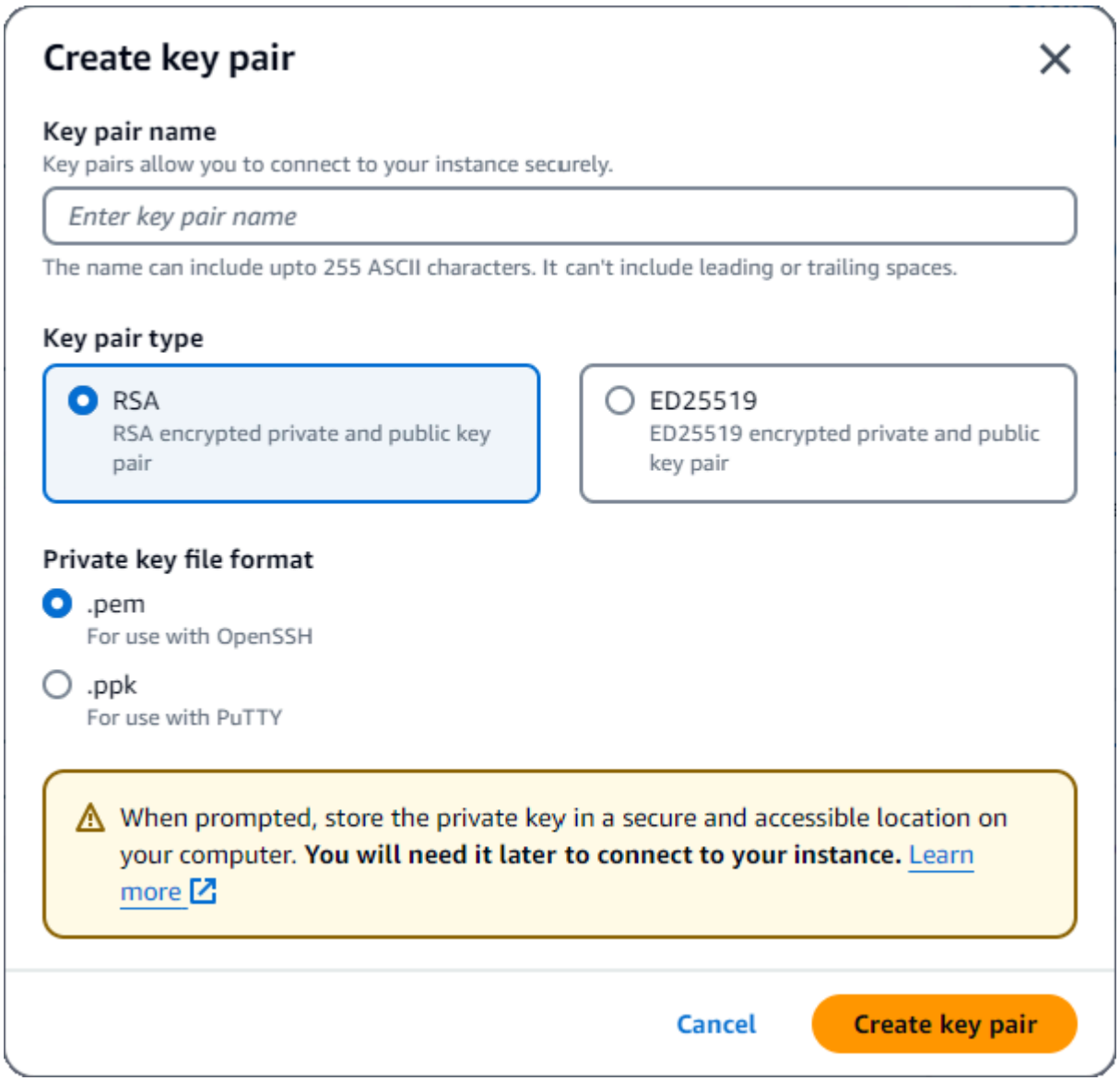
   Anda harus menyediakan pasangan kunci Amazon EC2.

   Jika Anda memiliki pasangan kunci Amazon EC2:

   a. Pilih pasangan kunci, pilih pasangan kunci Anda dari daftar.
   b. Anda harus sudah memiliki file kunci pribadi (file.pem atau .ppk) yang tersedia untuk masuk ke instans Amazon EC2 Anda.

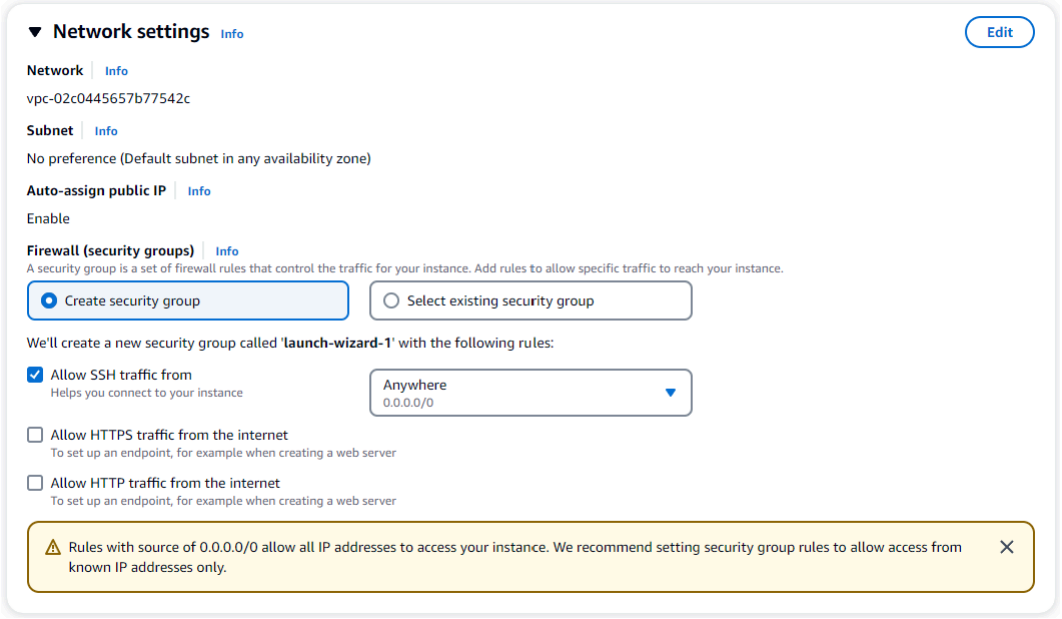
   Jika Anda tidak memiliki pasangan kunci Amazon EC2:

   a. Pilih Create new key pair, kotak dialog Create key pair muncul.
   b. Masukkan nama di bidang Nama pasangan kunci.
   c. Pilih jenis pasangan kunci dan format file kunci pribadi.
   d. Pilih Buat pasangan kunci.



> **Note**
>
> Untuk tujuan keamanan, kami sangat merekomendasikan penggunaan pasangan kunci untuk konektivitas SSH dan internet ke instans EC2 Anda.

6. Di bagian Network seetings, di bawah Firewall (grup keamanan), pilih Buat grup keamanan atau Pilih grup keamanan yang ada.



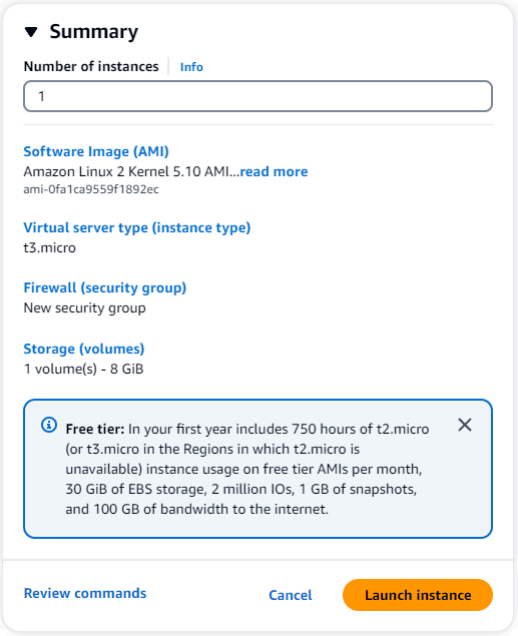
Jika Anda memilih untuk memilih grup keamanan yang ada, pilih salah satu dari daftar drop-down grup keamanan umum.

Jika Anda memilih untuk membuat grup keamanan baru, lakukan hal berikut:

a. Periksa semua aturan izin lalu lintas yang berlaku untuk konektivitas EC2 Anda.

b. Di bidang IP, pilih IP Saya atau pilih Kustom untuk memilih dari daftar blok CIDR, daftar awalan, atau grup securioty. Kami tidak merekomendasikan Anywhere sebagai pilihan, kecuali instans EC2 Anda berada di jaringan yang terisolasi, karena memungkinkan akses alamat IP apa pun ke instans EC2 Anda.

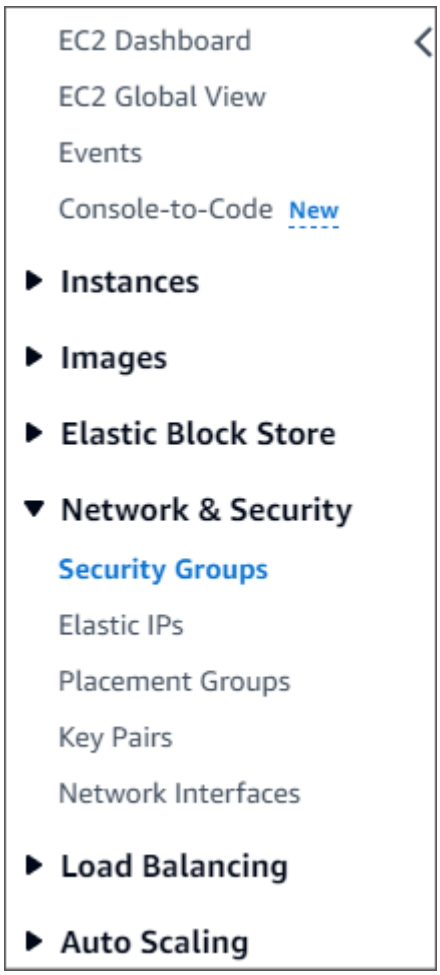
7. Di bagian Ringkasan, tinjau konfigurasi EC2 Anda dan pilih Launch instance jika benar. Edit grup keamanan.



Langkah 2: Buat grup keamanan

Anda sekarang akan membuat grup keamanan baru di Amazon VPC default Anda. Grup keamanan demoDocDB memungkinkan Anda untuk terhubung ke klaster Amazon DocumentDB Anda pada port 27017 (port default untuk Amazon DocumentDB) dari instans Amazon EC2 Anda.

1. Di Konsol Manajemen Amazon EC2, di bawah Jaringan dan Keamanan, pilih Grup keamanan.



2. Pilih Buat grup keamanan.

3. Di bagian Detail dasar:

   a. Untuk Nama grup keamanan, masukkan `demoDocDB`.

   b. Untuk Deskripsi, masukkan deskripsi.

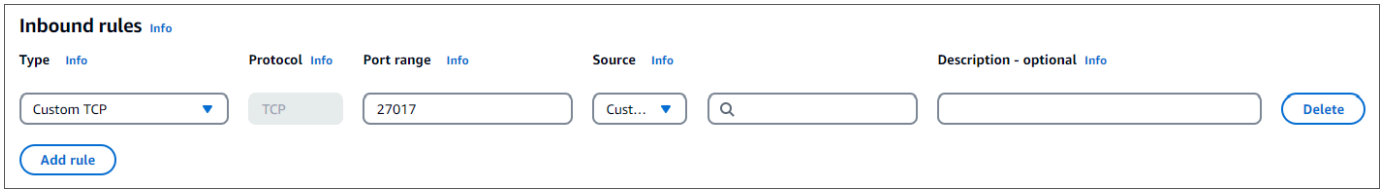
   c. Untuk VPC, terima penggunaan VPC default Anda.

4. Di bagian Aturan masuk, pilih Tambahkan aturan.

   a. Untuk Jenis, pilih Aturan TCP Kustom.

   b. Untuk Rentang port, masukkan `27017`.

   c. Untuk Tujuan, pilih Kustom. Di bidang di sebelahnya, cari grup keamanan yang baru saja Anda buat bernama `demoEC2`. Anda mungkin perlu untuk me-refresh peramban Anda bagi konsol Amazon EC2 untuk mengisi otomatis nama sumber `demoEC2`.

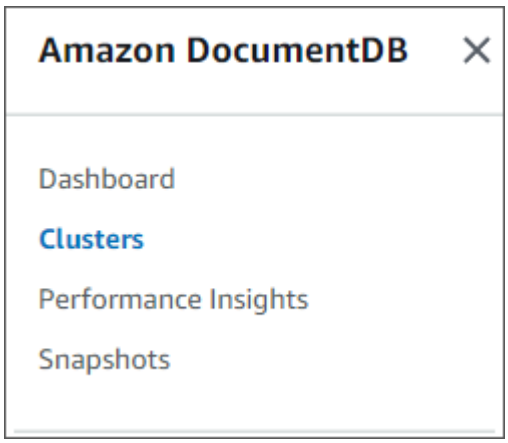


5. Terima semua default lainnya dan pilih Buat grup keamanan.

Langkah 3: Buat klaster Amazon DocumentDB

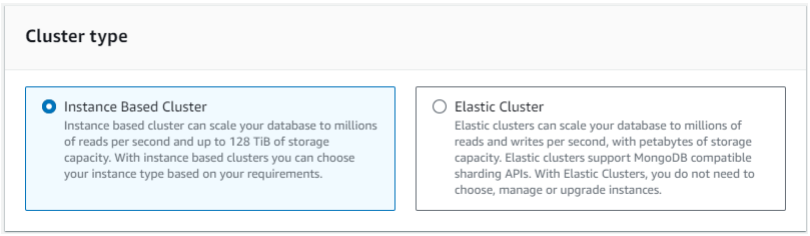
Saat instans Amazon EC2 sedang disediakan, Anda akan membuat klaster Amazon DocumentDB Anda.

1. Navigasikan ke konsol Amazon DocumentDB dan pilih Klaster dari panel navigasi.

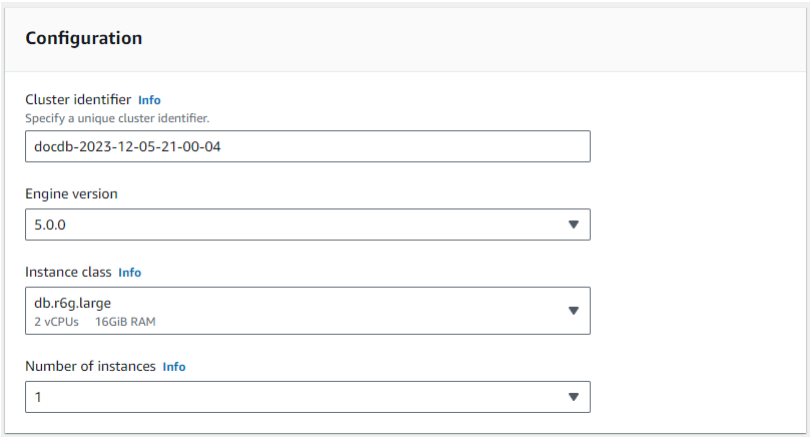


2. Pilih Buat.

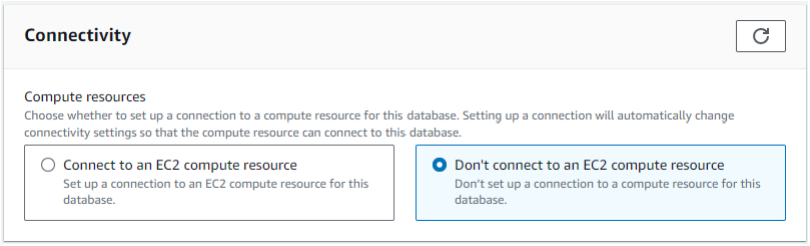
3. Biarkan pengaturan tipe Cluster pada defaultnya dari Instance Based Cluster.



4. Untuk Jumlah instans, pilih 1. Ini akan meminimalkan biaya. Biarkan pengaturan lain di defaultnya.

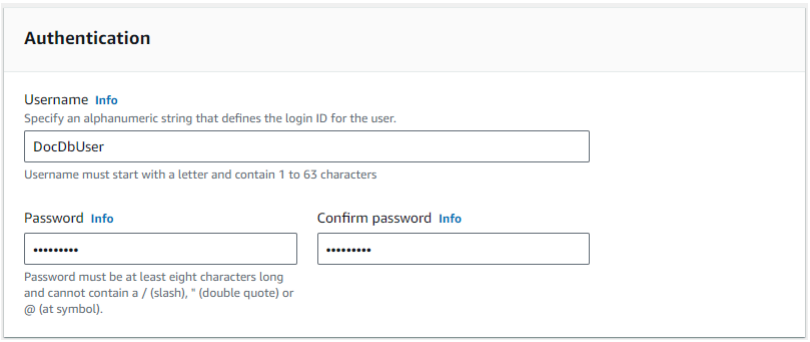


5. Untuk Konektivitas, biarkan pengaturan default Jangan sambungkan ke sumber daya komputasi EC2.

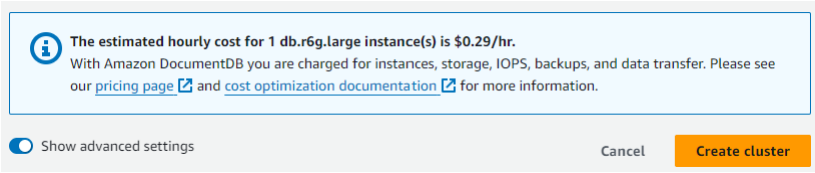


> **Note**
>
> Menyambung ke sumber daya komputasi EC2 secara otomatis membuat grup keamanan untuk koneksi sumber daya komputasi EC2 ke klaster Anda. Karena Anda membuat grup keamanan ini secara manual pada langkah sebelumnya, Anda harus memilih Jangan sambungkan ke sumber daya komputasi EC2 agar tidak membuat kumpulan grup keamanan kedua.

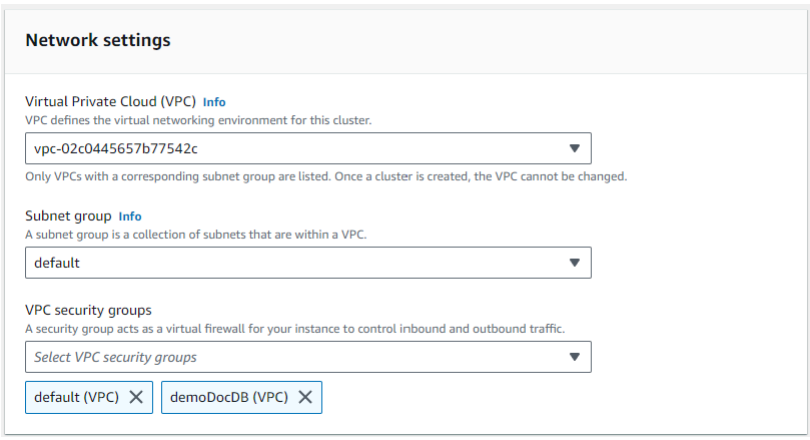
6. Untuk Otentikasi, masukkan kredensyal masuk. Penting: Anda akan memerlukan kredensyal masuk untuk mengautentikasi klaster Anda di langkah selanjutnya.



7. Hidupkan Tampilkan pengaturan lanjutan.



8. Di bagian Pengaturan jaringan, untuk Grup keamanan Amazon VPC, pilih demoDocDB.

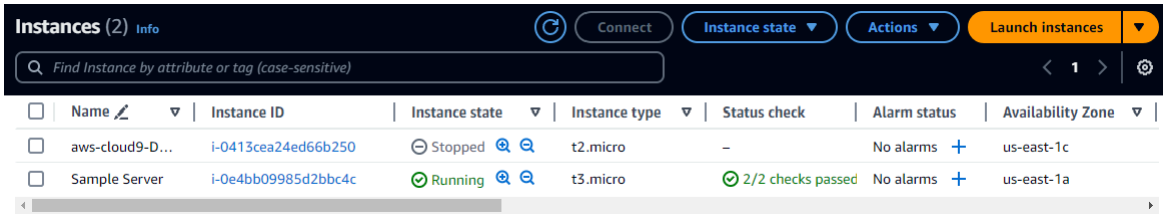


9. Pilih Buat klaster.

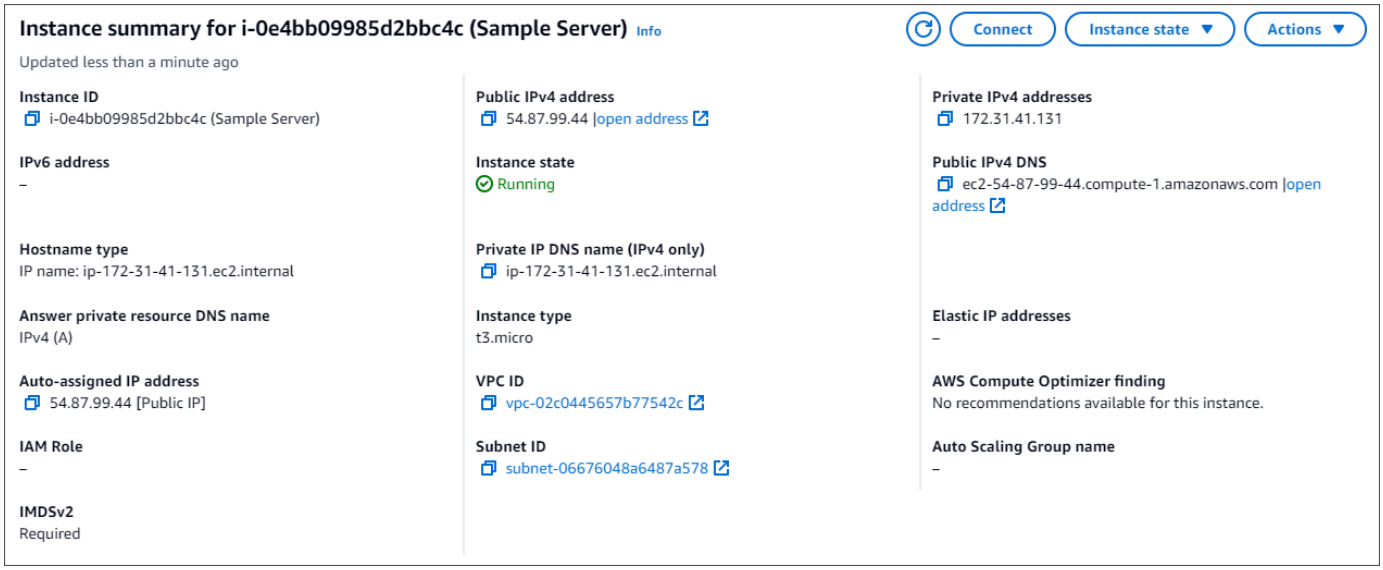
Langkah 4: Hubungkan ke instans Amazon EC2 Anda

Untuk menginstal shell mongo, Anda harus terlebih dahulu terhubung ke instans Amazon EC2 Anda. Instalasi shell mongo memungkinkan Anda untuk terhubung ke dan kueri klaster Amazon DocumentDB Anda. Selesaikan langkah-langkah berikut:

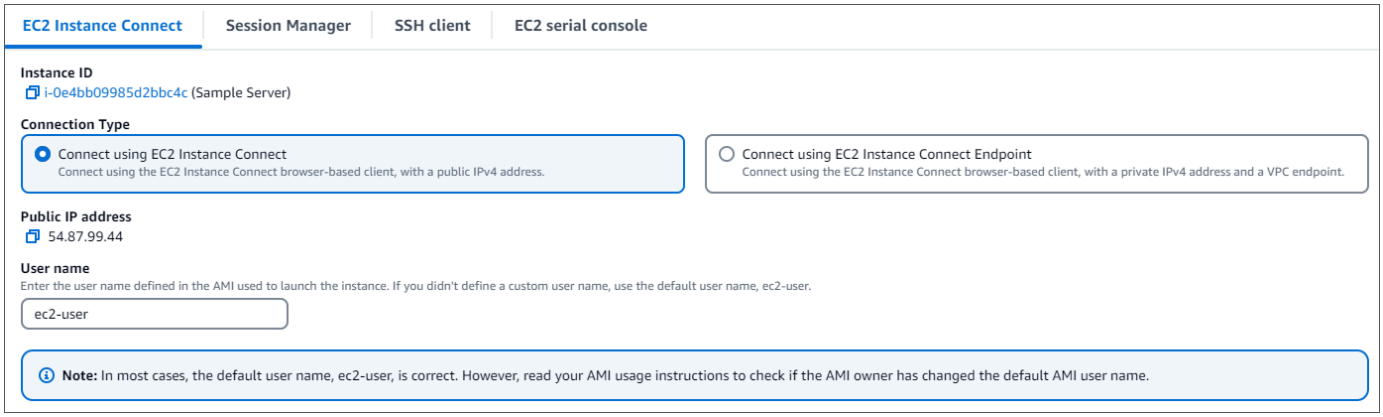
1. Pada konsol Amazon EC2, navigasikan ke instans Anda dan lihat apakah instans yang baru saja Anda buat berjalan. Jika ya, pilih instance dengan mengklik ID instance.



2. Pilih Hubungkan.



3. Ada empat opsi tab untuk metode koneksi Anda: Amazon EC2 Instance Connect, Session Manager, SSH client, atau konsol serial EC2. Anda harus memilih satu dan mengikuti instruksinya. Setelah selesai, pilih Connect.



> **Note**
>
> Jika alamat IP Anda berubah setelah Anda memulai panduan ini, atau Anda akan kembali ke lingkungan Anda di lain waktu, Anda harus memperbarui aturan masuk grup keamanan demoEC2 Anda untuk mengaktifkan lalu lintas masuk dari alamat API baru Anda.

Langkah 5: Instal shell mongo

Anda sekarang dapat menginstal shell mongo, yang merupakan utilitas baris perintah yang Anda gunakan untuk menghubungkan dan mengajukan kueri pada klaster Amazon DocumentDB Anda. Ikuti petunjuk di bawah ini untuk menginstal shell mongo untuk sistem operasi Anda.

On Amazon Linux

Untuk menginstal shell mongo di Amazon Linux

1. Buat file repositori. Di baris perintah instans EC2 Anda, jalankan perintah berikut:

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-5.0.repo
```

2. Setelah selesai, instal shell mongo dengan menjalankan perintah berikut:

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

On Ubuntu 18.04

Untuk menginstal shell mongo di Ubuntu 18.04

1. Impor kunci publik yang akan digunakan oleh sistem manajemen paket.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. Buat file daftar `/etc/apt/sources.list.d/mongodb-org-3.6.list` untuk MongoDB menggunakan perintah yang sesuai untuk versi Ubuntu Anda.

   Ubuntu 18.04

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/
mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
org-3.6.list
```

> **Note**
>
> Perintah di atas akan menginstal shell mongo 3.6 untuk Bionic dan Xenial.

3. Muat ulang basis data paket lokal menggunakan perintah berikut:

```
sudo apt-get update
```

4. Instal shell MongoDB.

```
sudo apt-get install -y mongodb-org-shell
```

Untuk informasi tentang menginstal versi sebelumnya dari MongoDB pada sistem Ubuntu Anda, lihat Instal MongoDB Community Edition di Ubuntu.

On other operating systems

Untuk menginstal shell mongo pada sistem operasi lain, lihat Instal MongoDB Community Edition dalam dokumentasi MongoDB.

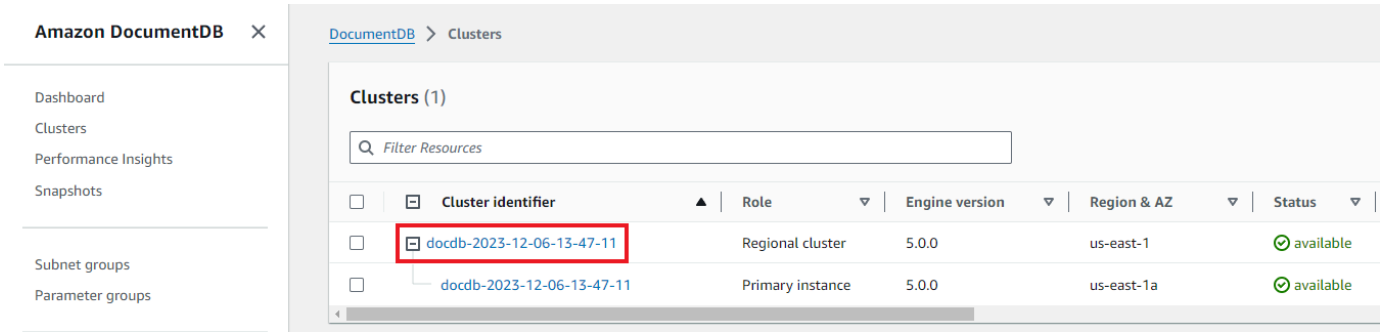
Langkah 6: Kelola Amazon DocumentDB TLS

Unduh sertifikat CA untuk Amazon DocumentDB dengan kode berikut: `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

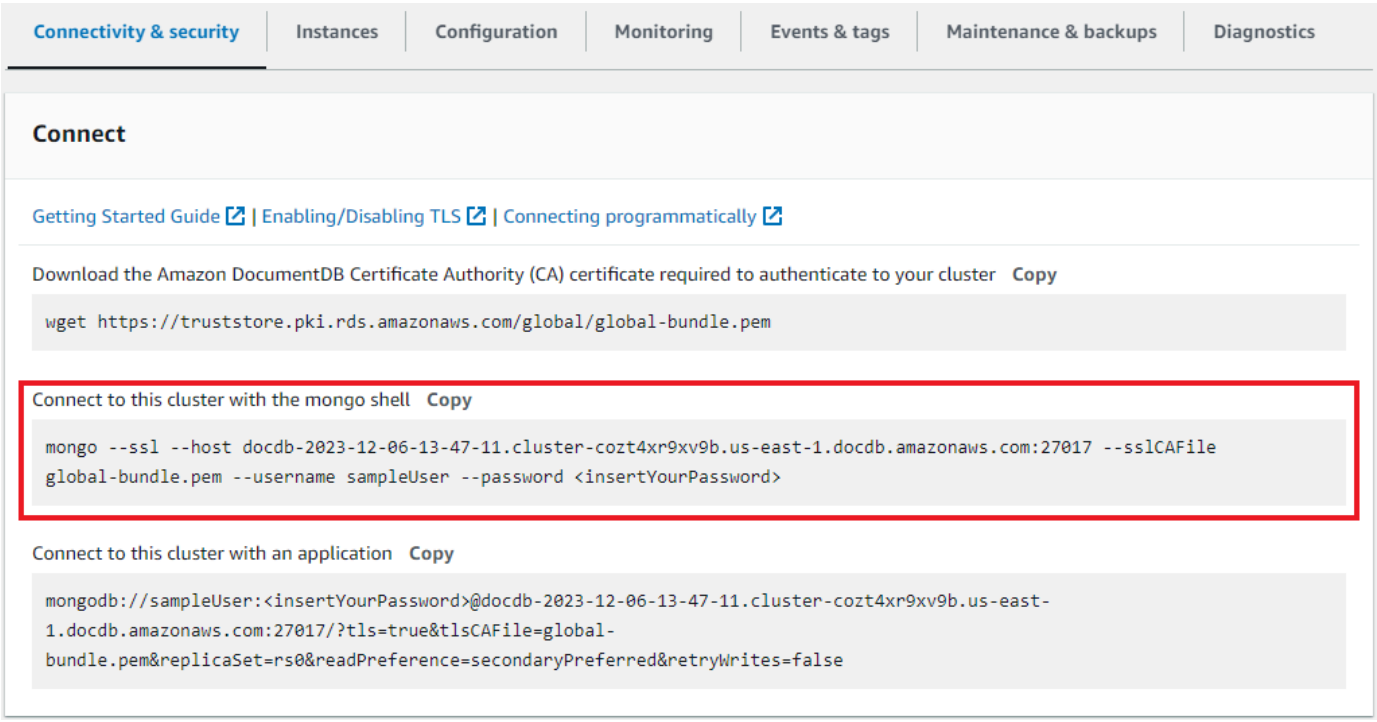
> **Note**
>
> Keamanan Lapisan Pengangkutan (TLS) diaktifkan secara default untuk setiap klaster Amazon DocumentDB baru. Untuk informasi selengkapnya, lihat Mengelola setelan TLS cluster Amazon DocumentDB.

Langkah 7: Hubungkan ke klaster Amazon DocumentDB Anda

1. Di konsol Amazon DocumentDB, di bawah Clusters, temukan klaster Anda. Pilih cluster yang Anda buat dengan mengklik pengenal Cluster.



2. Di tab Konektivitas dan keamanan, cari Connect to this cluster dengan shell mongo di kotak Connect:



Salin string koneksi yang disediakan dan tempel ke terminal Anda.

Buat perubahan berikut untuk itu:

a. Pastikan Anda memiliki nama pengguna yang benar dalam string.

b. Hilangkan `<insertYourPassword>` sehingga Anda diminta kata sandi oleh shell mongo saat Anda terhubung.

String koneksi Anda akan terlihat serupa dengan berikut ini:

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem
--username demoUser --password
```

3. Tekan enter di terminal Anda. Anda sekarang diminta untuk kata sandi Anda. Masukkan kata sandi Anda.
4. Saat Anda memasukkan kata sandi dan dapat melihat perintah `rs0:PRIMARY>`, Anda berhasil terhubung ke klaster Amazon DocumentDB Anda.

Mengalami masalah saat menghubungkan? Lihat Pemecahan Masalah Amazon DocumentDB.

Langkah 8: Masukkan dan kueri data

Sekarang setelah Anda terhubung ke klaster, Anda dapat menjalankan beberapa kueri untuk mengenali penggunaan basis data dokumen.

1. Untuk memasukkan satu dokumen, masukkan hal berikut ini:

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

2. Anda mendapatkan output berikut ini:

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

3. Anda dapat membaca dokumen yang Anda tulis dengan perintah `findOne()` (karena hanya mengembalikan satu dokumen). Masukkan yang berikut ini:

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

4. Anda mendapatkan output berikut ini:

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

5. Untuk mengajukan beberapa kueri lagi, pertimbangkan kasus penggunaan profil game. Pertama, masukkan beberapa entri ke dalam koleksi berjudul `profiles`. Masukkan yang berikut ini:

```
db.profiles.insertMany([
            { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
```

```
           { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,
"score":9},
           { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
"score":87},
           { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
           ])
```

6. Anda mendapatkan output berikut ini:

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

7. Gunakan perintah `find()` untuk mengembalikan semua dokumen dalam koleksi profil. Masukkan yang berikut ini:

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

8. Anda akan mendapatkan output yang akan cocok dengan data yang Anda ketik di Langkah 5.
9. Gunakan kueri untuk satu dokumen dengan menggunakan filter. Masukkan yang berikut ini:

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

10. Anda harus mendapatkan kembali output ini:

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

11. Sekarang mari kita coba untuk menemukan profil dan memodifikasinya dengan menggunakan perintah `findAndModify`. Kami akan memberi pengguna Matt sepuluh poin tambahan dengan kode berikut:

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

12. Anda mendapatkan output berikut (perhatikan bahwa skornya belum meningkat):

```
{
"_id" : 1,
```

```
    "name" : "Matt",
    "status" : "active",
    "level" : 12,
    "score" : 202
    }
```

13. Anda dapat memverifikasi bahwa skornya telah berubah dengan kueri berikut:

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

14. Anda mendapatkan output berikut ini:

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

Langkah 9: Jelajahi

Selamat! Anda telah berhasil menyelesaikan Panduan Quick Start untuk Amazon DocumentDB.

Apa selanjutnya? Pelajari cara memanfaatkan sepenuhnya basis data yang kuat ini dengan beberapa fitur populernya:

- Mengelola Amazon DocumentDB
- Penskalaan
- Mencadangkan dan memulihkan

> **Note**
>
> Untuk menghemat biaya, Anda dapat menghentikan klaster Amazon DocumentDB untuk mengurangi biaya atau menghapus klaster. Secara default, setelah 30 menit tidak aktif, AWS Cloud9 lingkungan Anda akan menghentikan instans Amazon EC2 yang mendasarinya.

Connect menggunakan driver Amazon DocumentDB JDBC

Driver JDBC untuk Amazon DocumentDB menyediakan antarmuka relasional SQL untuk pengembang dan memungkinkan konektivitas dari alat BI seperti Tableau dan. DbVisualizer

Untuk informasi lebih rinci, lihat dokumentasi Driver Amazon DocumentDB JDBC di. GitHub

Topik

- Memulai
- Connect ke Amazon DocumentDB dari Tableau Desktop
- Connect ke Amazon DocumentDB dari DbVisualizer
- Pembuatan skema otomatis JDBC
- Dukungan dan keterbatasan SQL
- Pemecahan Masalah

Memulai

Langkah 1. Buat Cluster Amazon DocumentDB

Jika Anda tidak memiliki kluster Amazon DocumentDB yang dibuat, maka buat cluster menggunakan petunjuk di bagian Memulai Panduan Pengembang Amazon DocumentDB.

> **Note**
>
> DocumentDB adalah layanan Virtual Private Cloud (VPC) saja. Jika Anda terhubung dari mesin lokal, di luar VPC cluster, Anda harus membuat koneksi SSH ke instans Amazon EC2. Dalam hal ini, luncurkan cluster Anda menggunakan instruksi di Connect with EC2. Lihat Menggunakan Terowongan SSH untuk Menyambung ke Amazon DocumentDB untuk informasi selengkapnya tentang tunneling SSH dan kapan Anda mungkin membutuhkannya.

Langkah 2. Instalasi JRE atau JDK

Tergantung pada aplikasi BI Anda, Anda mungkin perlu memastikan instalasi JRE atau JDK 64-bit versi 8 atau yang lebih baru diinstal pada komputer Anda. Anda dapat mengunduh Java SE Runtime Environment 8 di sini.

Langkah 3. Unduh Driver DocumentDB JDBC

Unduh driver DocumentDB JDBC dari sini. Driver dikemas sebagai file JAR tunggal (misalnya documentdb-jdbc-1.0.0-all.jar).

Langkah 4. Menggunakan Terowongan SSH untuk Connect ke Amazon DocumentDB

Klaster Amazon DocumentDB (dengan kompatibilitas MongoDB) di-deploy dalam Amazon Virtual Private Cloud (Amazon VPC). Mereka dapat diakses langsung oleh instans Amazon EC2 atau AWS layanan lain yang digunakan di VPC Amazon yang sama. Selain itu, Amazon DocumentDB dapat diakses oleh instans EC2a atau layanan AWS lain di VPC yang berbeda di Wilayah yang sama atau AWS Wilayah lain melalui peering VPC.

Anda dapat menggunakan SSH tunneling (juga dikenal sebagai port forwarding) untuk mengakses sumber daya Amazon DocumentDB Anda, dari luar VPC cluster. Ini akan menjadi kasus bagi sebagian besar pengguna yang tidak menjalankan aplikasi mereka pada VM di VPC yang sama dengan cluster DocumentDB.

Untuk membuat terowongan SSH, Anda memerlukan instans Amazon EC2 yang berjalan di Amazon VPC yang sama dengan klaster Amazon DocumentDB Anda. Anda dapat menggunakan instans EC2 yang ada di VPC yang sama sebagai klaster Anda atau membuatnya. Anda dapat mengatur terowongan SSH ke cluster Amazon `sample-cluster.node.us-east-1.docdb.amazonaws.com` DocumentDB dengan menjalankan perintah berikut di komputer lokal Anda.

```
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
```

Bendera -L digunakan untuk meneruskan port lokal. Ini adalah prasyarat untuk menghubungkan ke alat BI apa pun yang berjalan pada klien di luar VPC Anda. Setelah Anda menjalankan langkah di atas, Anda dapat melanjutkan ke langkah berikutnya untuk alat BI pilihan Anda.

Untuk informasi lebih lanjut tentang SSH tunneling, silakan lihat dokumentasi tentang Menggunakan terowongan SSH untuk terhubung ke Amazon DocumentDB.

Connect ke Amazon DocumentDB dari Tableau Desktop

Topik

- Menambahkan Driver Amazon DocumentDB JDBC
- Menghubungkan ke Amazon DocumentDB menggunakan Tableau - SSH Tunnel

Menambahkan Driver Amazon DocumentDB JDBC

Untuk terhubung ke Amazon DocumentDB dari Tableau Desktop Anda harus mengunduh dan menginstal driver DocumentDB JDBC dan konektor DocumentDB Tableau.

1. Unduh file JAR driver JDBC DocumentDB dan salin ke salah satu direktori ini sesuai dengan sistem operasi Anda:
   - Jendela - `C:\Program Files\Tableau\Drivers`
   - macOS - `~/Library/Tableau/Drivers`
2. Unduh konektor DocumentDB Tableau (file TACO) dan salin ke direktori My Tableau Repository/Connectors Anda.
   - Jendela - `C:\Users\[user]\Documents\My Tableau Repository\Connectors`
   - macOS - `/Users/[user]/Documents/My Tableau Repository/Connectors`

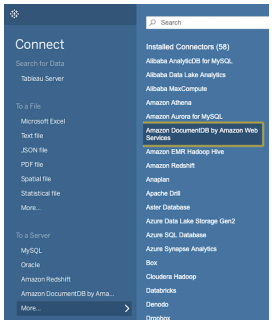
Untuk informasi tambahan, lihat dokumentasi Tableau.

> **Note**
>
> Jika Anda menggunakan sertifikat CA yang lebih baru, pastikan untuk meningkatkan driver JDBC Anda ke v1.4.5 (tersedia di repositori ini AWS GitHub .).

Menghubungkan ke Amazon DocumentDB menggunakan Tableau - SSH Tunnel

Untuk terhubung ke Tableau dari mesin klien di luar VPC cluster DocumentDB Anda, Anda harus menyiapkan terowongan SSH sebelum mengikuti langkah-langkah di bawah ini:

1. Luncurkan aplikasi Tableau Desktop.
2. Arahkan ke Connect > To A Server > More.
3. Pilih Amazon DocumentDB oleh Amazon Web Services di bawah Konektor Terpasang.

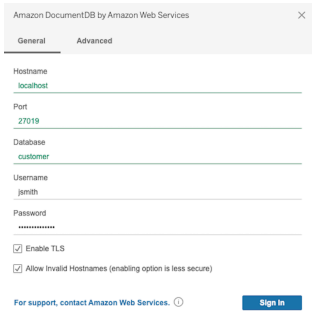


Menghubungkan ke Amazon DocumentDB Menggunakan Tableau - Terowongan SSH Eksternal

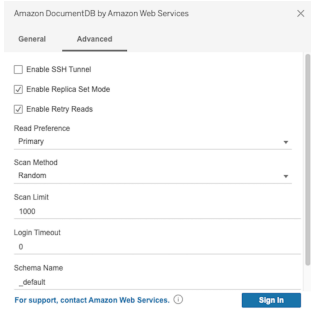
1. Masukkan parameter koneksi yang diperlukan Nama host, Port, Database, Nama Pengguna dan Kata Sandi. Parameter koneksi dalam contoh di bawah ini setara dengan string koneksi JDBC:

   ```
   jdbc:documentdb://localhost:27019/test?
   tls=true&tlsAllowInvalidHostnames=true&scanMethod=random&scanLimit=1000&logir
   ```

   parameter nama pengguna dan kata sandi diteruskan secara terpisah dalam koleksi properti. Untuk informasi selengkapnya tentang parameter string koneksi, lihat dokumentasi github Amazon DocumentDB JDBC Driver.



2. (Opsional) Opsi yang lebih canggih dapat ditemukan di Advanced tab.

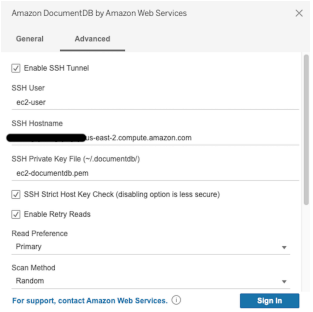


3. Pilih Masuk.

Menghubungkan ke Amazon DocumentDB Menggunakan Tableau - Terowongan SSH Internal

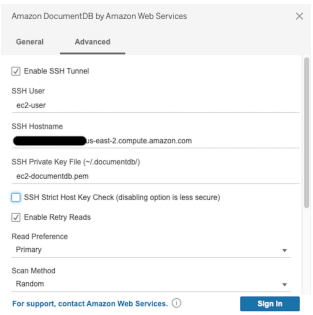
> **Note**
>
> Jika Anda memilih untuk tidak mengatur terowongan SSH menggunakan terminal, Anda dapat menggunakan GUI Tableau untuk menentukan detail instans EC2 Anda yang secara inheren akan digunakan driver JDBC untuk membuat terowongan SSH.

1. Pada tab Advanced, pilih opsi Enable SSH Tunnel untuk meninjau properti lebih lanjut.



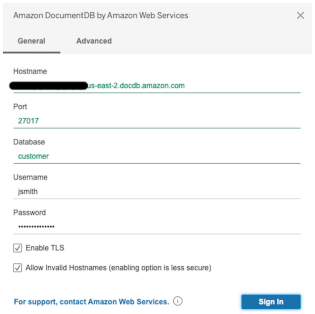
2. Masukkan SSH User, SSH Hostname, dan SSH Private Key File.
3. (Opsional) Anda dapat menonaktifkan opsi Pemeriksaan Kunci Host Ketat SSH yang melewati pemeriksaan kunci host terhadap file host yang dikenal.

> **Note**
>
> Menonaktifkan opsi ini kurang aman karena dapat menyebabkan serangan. man-in-the-middle



4. Masukkan parameter yang diperlukan; Nama host, Port, Database, Nama Pengguna dan Kata Sandi.

> **Note**
>
> Pastikan Anda menggunakan endpoint cluster DocumentDB dan bukan localhost saat menggunakan opsi terowongan SSH internal.



5. Pilih Masuk.

Connect ke Amazon DocumentDB dari DbVisualizer

Topik

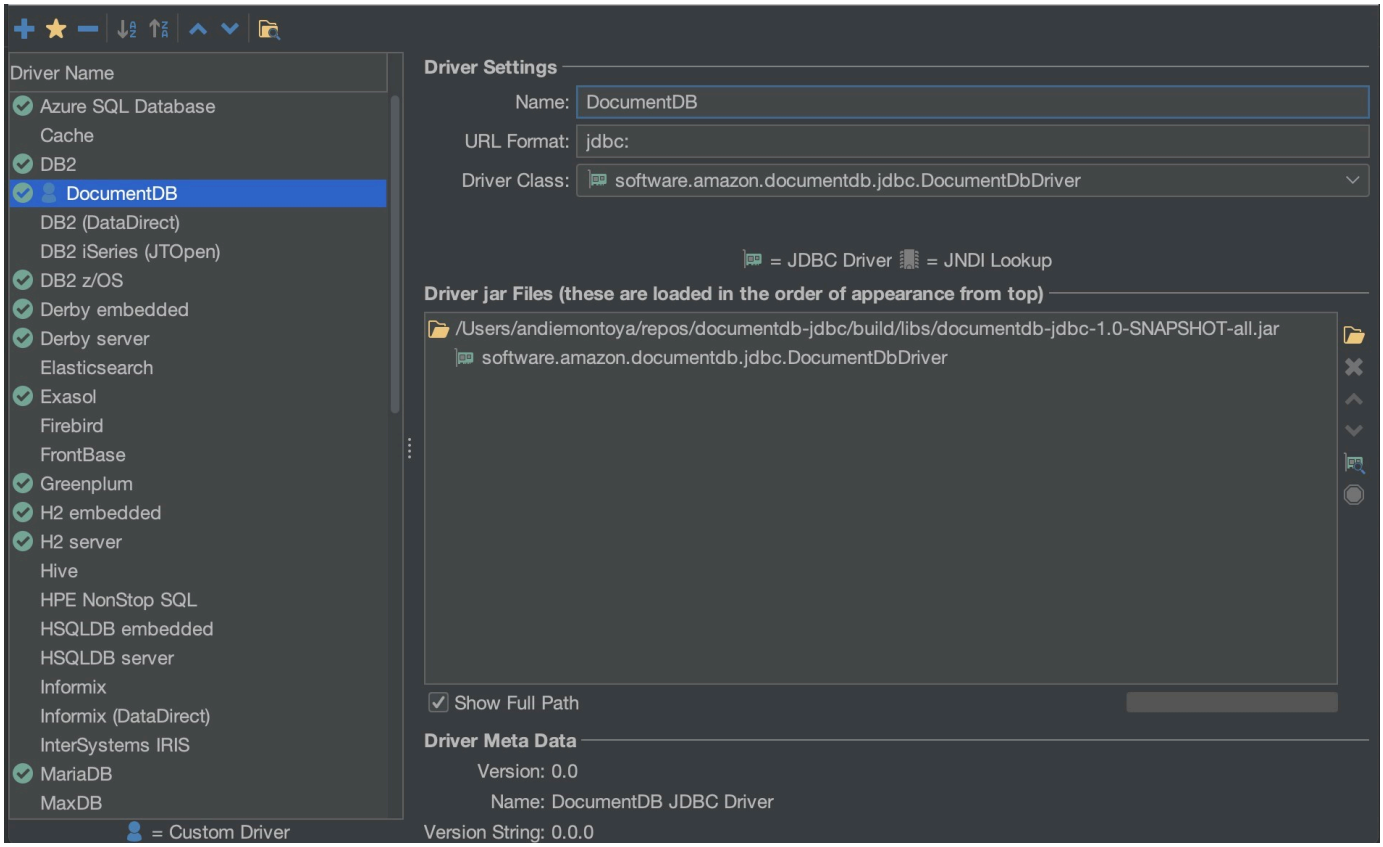
- Menambahkan driver Amazon DocumentDB JDBC
- Menghubungkan ke Amazon DocumentDB menggunakan DbVisualizer

Menambahkan driver Amazon DocumentDB JDBC

Untuk terhubung ke Amazon DocumentDB DbVisualizer dari Anda harus terlebih dahulu mengimpor Driver Amazon DocumentDB JDBC

1. Mulai DbVisualizer aplikasi dan arahkan ke jalur menu: Tools > Driver Manager...
2. Pilih + (atau di menu, pilih Driver > Create Driver).
3. Setel Nama ke`DocumentDB`.
4. Setel Format URL ke `jdbc:documentdb://<host>[:port]/<database>[?option=value[&option=value[...]]]`
5. Pilih tombol folder dan kemudian pilih file JAR driver Amazon DocumentDB JDBC dan pilih tombol Buka.

6. Verifikasi bahwa bidang Kelas Driver diatur `kesoftware.amazon.documentdb.jdbc.DocumentDbDriver`. Pengaturan Driver Manager Anda untuk DocumentDB akan terlihat seperti contoh berikut.



7. Tutup dialog. Driver Amazon DocumentDB JDBC akan disiapkan dan siap digunakan.

Menghubungkan ke Amazon DocumentDB menggunakan DbVisualizer

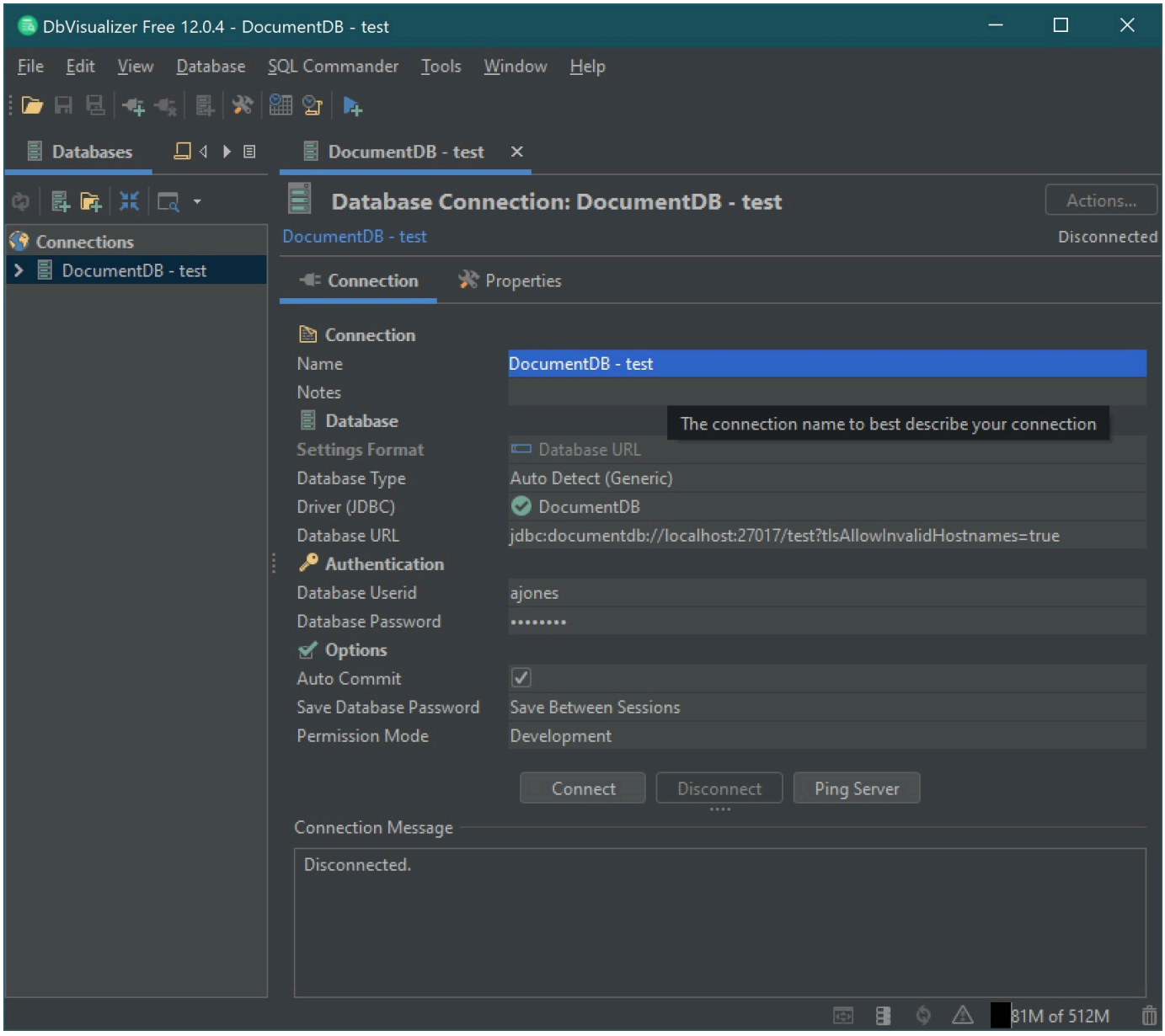
Connect ke Amazon DocumentDB Menggunakan DbVisualizer

1. Jika Anda terhubung dari luar VPC cluster Amazon DocumentDB, pastikan Anda telah menyiapkan terowongan SSH.
2. Pilih Database > Buat Koneksi Database dari menu tingkat atas.
3. Masukkan nama deskriptif untuk bidang Nama.
4. Setel Driver (JDBC) ke driver DocumentDB yang Anda buat di bagian sebelumnya.
5. Setel URL Database ke string koneksi JDBC Anda.

   Misalnya: `jdbc:documentdb://localhost:27017/database?tlsAllowInvalidHostnames=true`

6. Setel Userid Database ke ID pengguna Amazon DocumentDB Anda.
7. Atur Kata Sandi Database ke kata sandi yang sesuai untuk ID pengguna.

   Dialog Koneksi Database Anda akan terlihat seperti dialog berikut:



8. Pilih Hubungkan.

Pembuatan skema otomatis JDBC

Amazon DocumentDB adalah database dokumen dan karena itu tidak memiliki konsep tabel dan skema. Namun, alat BI seperti Tableau akan mengharapkan database yang terhubung untuk

menyajikan skema. Secara khusus, ketika koneksi driver JDBC perlu mendapatkan skema untuk koleksi dalam database, itu akan polling untuk semua koleksi dalam database. Pengemudi akan menentukan apakah versi skema yang di-cache untuk koleksi itu sudah ada. Jika versi cache tidak ada, itu akan mengambil sampel koleksi untuk dokumen dan membuat skema berdasarkan perilaku berikut.

Topik

- Keterbatasan pembuatan skema
- Opsi metode pemindaian
- Jenis data Amazon DocumentDB
- Memetakan bidang dokumen skalar
- Penanganan tipe data objek dan array

Keterbatasan pembuatan skema

Driver DocumentDB JDBC memberlakukan batas pada panjang pengidentifikasi pada 128 karakter. Generator skema dapat memotong panjang pengidentifikasi yang dihasilkan (nama tabel dan nama kolom) untuk memastikan mereka sesuai dengan batas itu.

Opsi metode pemindaian

Perilaku pengambilan sampel dapat dimodifikasi menggunakan string koneksi atau opsi sumber data.

- ScanMethod= <option>
  - acak - (default) - Dokumen sampel dikembalikan dalam urutan acak.
  - IDForward - Contoh dokumen dikembalikan dalam urutan id.
  - IDReverse - Contoh dokumen dikembalikan dalam urutan terbalik id.
  - semua - Sampel semua dokumen dalam koleksi.
- ScanLimit= <n>- Jumlah dokumen untuk sampel. Nilai harus berupa bilangan bulat positif. Nilai default adalah 1000. Jika ScanMethod diatur ke semua, opsi ini diabaikan.

Jenis data Amazon DocumentDB

Server DocumentDB mendukung sejumlah tipe data MongoDB. Di bawah ini adalah tipe data yang didukung, dan tipe data JDBC yang terkait.

Tipe Data MongoDB	Didukung di DocumentDB	Tipe Data JDBC
Data Biner	Ya	VARBINARY
Boolean	Ya	BOOLEAN
Ganda	Ya	DOUBLE
Bilangan bulat 32-bit	Ya	INTEGER
Integer 64-bit	Ya	BIGINT
Tali	Ya	VARCHAR
ObjectId	Ya	VARCHAR
Tanggal	Ya	TIMESTAMP
Null	Ya	VARCHAR
Ekspresi reguler	Ya	VARCHAR
Stempel Waktu	Ya	VARCHAR
MinKey	Ya	VARCHAR
MaxKey	Ya	VARCHAR
Objek	Ya	meja virtual
Array	Ya	meja virtual
Decimal128	Tidak	DECIMAL
JavaScript	Tidak	VARCHAR
JavaScript (dengan ruang lingkup)	Tidak	VARCHAR
Tidak terdefinisi	Tidak	VARCHAR
Simbol	Tidak	VARCHAR

Tipe Data MongoDB	Didukung di DocumentDB	Tipe Data JDBC
DBPointer (4.0+)	Tidak	VARCHAR

Memetakan bidang dokumen skalar

Saat memindai sampel dokumen dari koleksi, driver JDBC akan membuat satu atau lebih skema untuk mewakili sampel dalam koleksi. Secara umum, bidang skalar dalam dokumen memetakan ke kolom dalam skema tabel. Misalnya, dalam kumpulan bernama tim, dan satu dokumen`{ "_id" : "112233", "name" : "Alastair", "age": 25 }`, ini akan dipetakan ke skema:

Nama Tabel	Nama kolom	Tipe data	Kunci
team	id tim	VARCHAR	PK
team	name	VARCHAR	
team	usia	INTEGER	

Promosi konflik tipe data

Saat memindai dokumen sampel, ada kemungkinan bahwa tipe data untuk bidang tidak konsisten dari dokumen ke dokumen. Dalam hal ini, driver JDBC akan mempromosikan tipe data JDBC ke tipe data umum yang akan sesuai dengan semua tipe data dari dokumen sampel.

Sebagai Contoh:

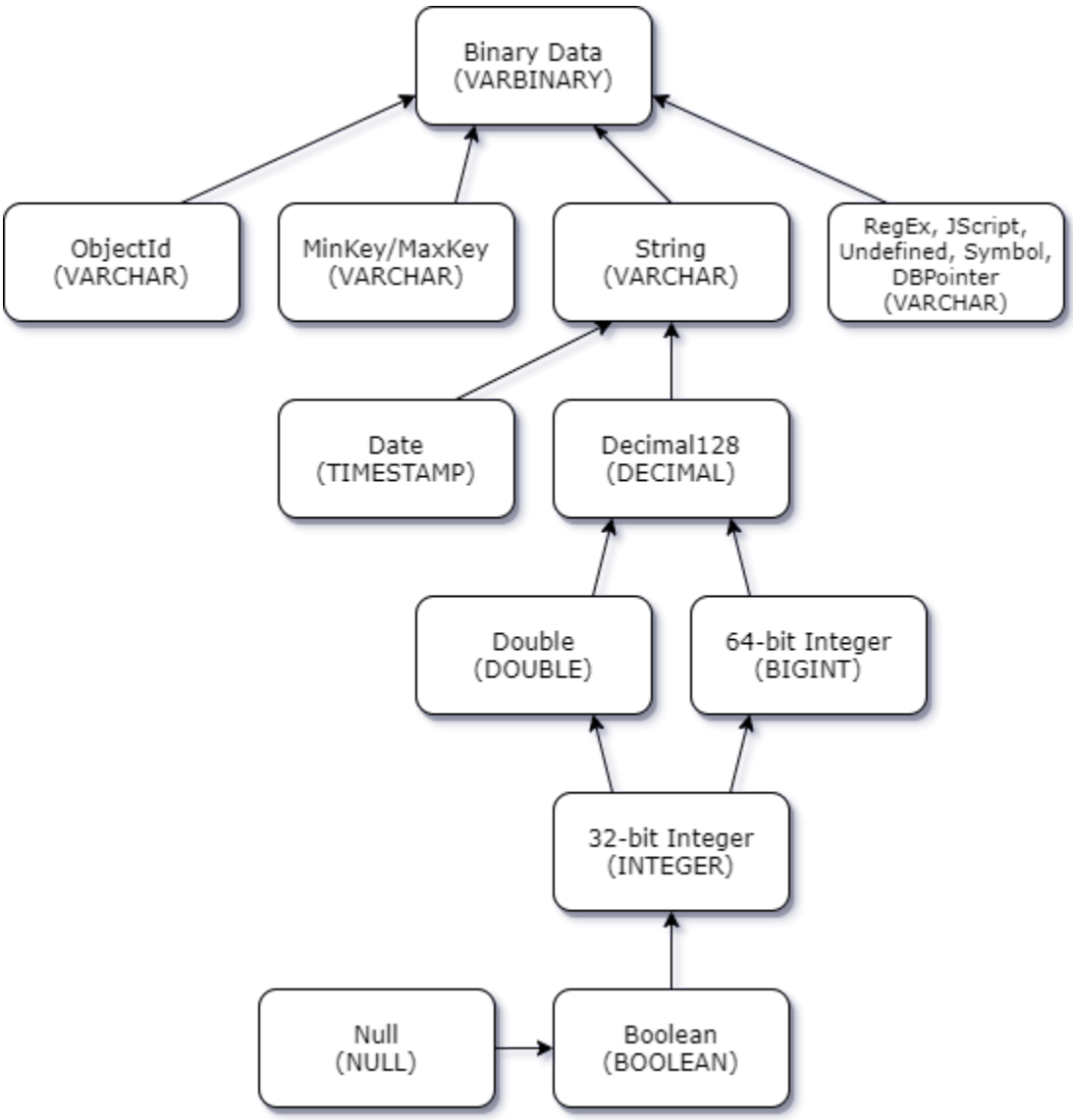
```
{
"_id" : "112233",
"name" : "Alastair", "age" : 25
}

{
"_id" : "112244",
"name" : "Benjamin",
"age" : "32"
}
```

Bidang usia adalah tipe 32-bit integer dalam dokumen pertama tetapi string dalam dokumen kedua. Di sini driver JDBC akan mempromosikan tipe data JDBC ke VARCHAR untuk menangani salah satu tipe data saat ditemui.

Nama Tabel	Nama kolom	Tipe data	Kunci
team	id tim	VARCHAR	PK
team	name	VARCHAR	
team	usia	VARCHAR	

Promosi konflik skalar-skalar

Diagram berikut menunjukkan cara konflik tipe data skalar-skalar diselesaikan.



Promosi konflik tipe skalar-kompleks

Seperti konflik tipe skalar-skalar, bidang yang sama dalam dokumen yang berbeda dapat memiliki tipe data yang bertentangan antara kompleks (array dan objek) dan skalar (integer, boolean, dll.). Semua konflik ini diselesaikan (dipromosikan) ke VARCHAR untuk bidang tersebut. Dalam hal ini, array dan data objek dikembalikan sebagai representasi JSON.

Array Tertanam - Contoh Konflik Bidang String:

```
{
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
```

```
      "Vogue",
      "People",
      "USA Today"
   ]
}
{
   "_id":"112244",
   "name":"Joan Starr",
   "subscriptions":1
}
```

Contoh di atas memetakan skema untuk tabel customer2:

Nama Tabel	Nama kolom	Tipe data	Kunci
pelanggan2	id pelanggan2	VARCHAR	PK
pelanggan2	name	VARCHAR	
pelanggan2	langganan	VARCHAR	

dan tabel virtual customer1_subscriptions:

Nama Tabel	Nama kolom	Tipe data	Kunci
customer1_subscriptions	id pelanggan1	VARCHAR	PK/FK
customer1_subscriptions	berlangganan_index_lvl0	BIGINT	PK
customer1_subscriptions	nilai	VARCHAR	
customer_address	kota	VARCHAR	
customer_address	region	VARCHAR	
customer_address	negeri	VARCHAR	

Nama Tabel	Nama kolom	Tipe data	Kunci
customer_address	code	VARCHAR	

Penanganan tipe data objek dan array

Sejauh ini, kami hanya menjelaskan bagaimana tipe data skalar dipetakan. Jenis data Object dan Array (saat ini) dipetakan ke tabel virtual. Driver JDBC akan membuat tabel virtual untuk mewakili bidang objek atau array dalam dokumen. Nama tabel virtual yang dipetakan akan menggabungkan nama koleksi asli diikuti dengan nama bidang yang dipisahkan oleh karakter garis bawah (“_”).

Kunci utama tabel dasar (“_id”) mengambil nama baru di tabel virtual baru dan disediakan sebagai kunci asing ke tabel dasar terkait.

Untuk bidang tipe array tertanam, kolom indeks dihasilkan untuk mewakili indeks ke dalam array di setiap tingkat array.

Contoh bidang objek tertanam

Untuk bidang objek dalam dokumen, pemetaan ke tabel virtual dibuat oleh driver JDBC.

```
{
   "Collection: customer",
   "_id":"112233",
   "name":"George Jackson",
   "address":{
      "address1":"123 Avenue Way",
      "address2":"Apt. 5",
      "city":"Hollywood",
      "region":"California",
      "country":"USA",
      "code":"90210"
   }
}
```

Contoh di atas memetakan skema untuk tabel pelanggan:

Nama Tabel	Nama kolom	Tipe data	Kunci
pelanggan	id pelanggan	VARCHAR	PK

Nama Tabel	Nama kolom	Tipe data	Kunci
pelanggan	name	VARCHAR	

dan tabel virtual customer_address:

Nama Tabel	Nama kolom	Tipe data	Kunci
customer_address	id pelanggan	VARCHAR	PK/FK
customer_address	alamat1	VARCHAR	
customer_address	alamat2	VARCHAR	
customer_address	kota	VARCHAR	
customer_address	region	VARCHAR	
customer_address	negeri	VARCHAR	
customer_address	code	VARCHAR	

Contoh bidang array tertanam

Untuk bidang array dalam dokumen, pemetaan ke tabel virtual juga dibuat oleh driver JDBC.

```
{
   "Collection: customer1",
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
      "Vogue",
      "People",
      "USA Today"
   ]
}
```

Contoh di atas memetakan skema untuk tabel pelanggan1:

Nama Tabel	Nama kolom	Tipe data	Kunci
pelanggan1	id pelanggan1	VARCHAR	PK
pelanggan1	name	VARCHAR	

dan tabel virtual customer1_subscriptions:

Nama Tabel	Nama kolom	Tipe data	Kunci
customer1_subscriptions	id pelanggan1	VARCHAR	PK/FK
customer1_subscriptions	berlangganan_index_lvl0	BIGINT	PK
customer1_subscriptions	nilai	VARCHAR	
customer_address	kota	VARCHAR	
customer_address	region	VARCHAR	
customer_address	negeri	VARCHAR	
customer_address	code	VARCHAR	

Dukungan dan keterbatasan SQL

Driver Amazon DocumentDB JDBC adalah driver read-only yang mendukung subset SQL-92 dan beberapa ekstensi umum. Lihat dokumentasi batasan SQL dan dokumentasi batasan JDBC untuk informasi lebih lanjut.

Pemecahan Masalah

Jika Anda mengalami masalah saat menggunakan driver Amazon DocumentDB JDBC, lihat Panduan Pemecahan Masalah.

Connect menggunakan driver ODBC Amazon DocumentDB

Driver ODBC untuk Amazon DocumentDB menyediakan antarmuka relasional SQL untuk pengembang dan memungkinkan konektivitas dari alat BI seperti Power BI Desktop dan Microsoft Excel.

Untuk informasi lebih rinci, lihat dokumentasi Driver ODBC Amazon DocumentDB di GitHub.

Topik

- Mulai
- Menyiapkan Driver ODBC Amazon DocumentDB di Windows
- Connect ke Amazon DocumentDB dari Microsoft Excel
- Connect ke Amazon DocumentDB dari Microsoft Power BI Desktop
- Generasi skema otomatis
- Support dan batasan SQL
- Pemecahan Masalah

Mulai

Langkah 1. Membuat Cluster Amazon DocumentDB

Jika Anda belum memiliki klaster Amazon DocumentDB, ada sejumlah cara untuk memulai.

> **Note**
>
> Amazon DocumentDB adalah layanan khusus Virtual Private Cloud (VPC). Jika Anda terhubung dari mesin lokal di luar VPC klaster, Anda harus membuat koneksi SSH ke instans Amazon EC2. Dalam hal ini, luncurkan klaster Anda menggunakan petunjuk di Connect dengan EC2. Lihat Menggunakan Tunnel SSH untuk Connect ke Amazon DocumentDB untuk informasi selengkapnya tentang tunneling SSH dan kapan Anda mungkin membutuhkannya.

Langkah 2. Instalasi JRE atau JDK

Tergantung pada aplikasi BI Anda, Anda mungkin perlu memastikan 64-bit JRE atau JDK versi instalasi 8 atau yang lebih baru diinstal pada komputer Anda. Anda dapat mengunduh Java SE Runtime Environment 8 di sini.

Langkah 3. Unduh Driver ODBC Amazon DocumentDB

Unduh driver Amazon DocumentDB ODBC di sini. Pilih installer yang tepat (misalnya, documentdb-odbc-1.0.0.msi). Ikuti panduan penginstalan.

Langkah 4. Menggunakan SSH Tunnel untuk Terhubung ke Amazon DocumentDB

Klaster Amazon DocumentDB di-deploy dalam Amazon Virtual Private Cloud (Amazon VPC) Mereka dapat diakses langsung oleh instans Amazon EC2 atau layanan AWS lainnya yang di-deploy di Amazon VPC yang sama. Selain itu, Amazon DocumentDB dapat diakses oleh instans Amazon EC2 atauAWS layanan lainnya di VPC berbeda di yang sama atauAWS wilayah lain melalui peering VPC.

Namun, anggaplah kasus penggunaan Anda mengharuskan Anda (atau aplikasi Anda) mengakses sumber daya Amazon DocumentDB Anda dari luar VPC klaster. Ini akan menjadi kasus bagi sebagian besar pengguna yang tidak menjalankan aplikasi mereka pada VM di VPC yang sama dengan klaster Amazon DocumentDB. Saat terhubung dari luar VPC, Anda dapat menggunakan tunneling SSH (juga dikenal sebagai penerusan port) untuk mengakses sumber daya Amazon DocumentDB Anda.

Untuk membuat terowongan SSH, Anda memerlukan instans Amazon EC2 yang berjalan di Amazon VPC yang sama dengan klaster Amazon DocumentDB Anda. Anda dapat menggunakan instans EC2 yang ada di VPC yang sama sebagai klaster Anda atau membuatnya. Anda dapat mengatur terowongan SSH ke klaster Amazon DocumentDB`sample-cluster.node.us-east-1.docdb.amazonaws.com` dengan menjalankan perintah berikut di komputer lokal Anda:

```
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
```

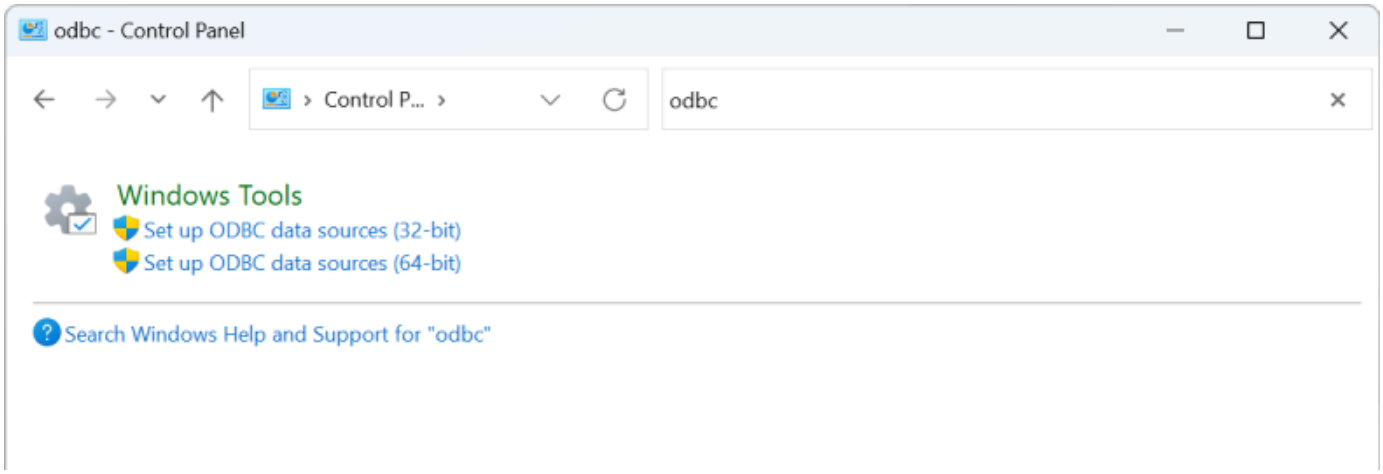
Bendera `-L` digunakan untuk meneruskan port lokal. Ini adalah prasyarat untuk menghubungkan ke alat BI apa pun yang berjalan pada klien di luar VPC Anda. Setelah Anda menjalankan langkah di atas, Anda dapat melanjutkan ke langkah berikutnya untuk alat BI pilihan Anda.

Untuk informasi lebih lanjut tentang tunneling SSH, lihat dokumentasi tentang Menggunakan Tunnel SSH untuk Terhubung ke Amazon DocumentDB.

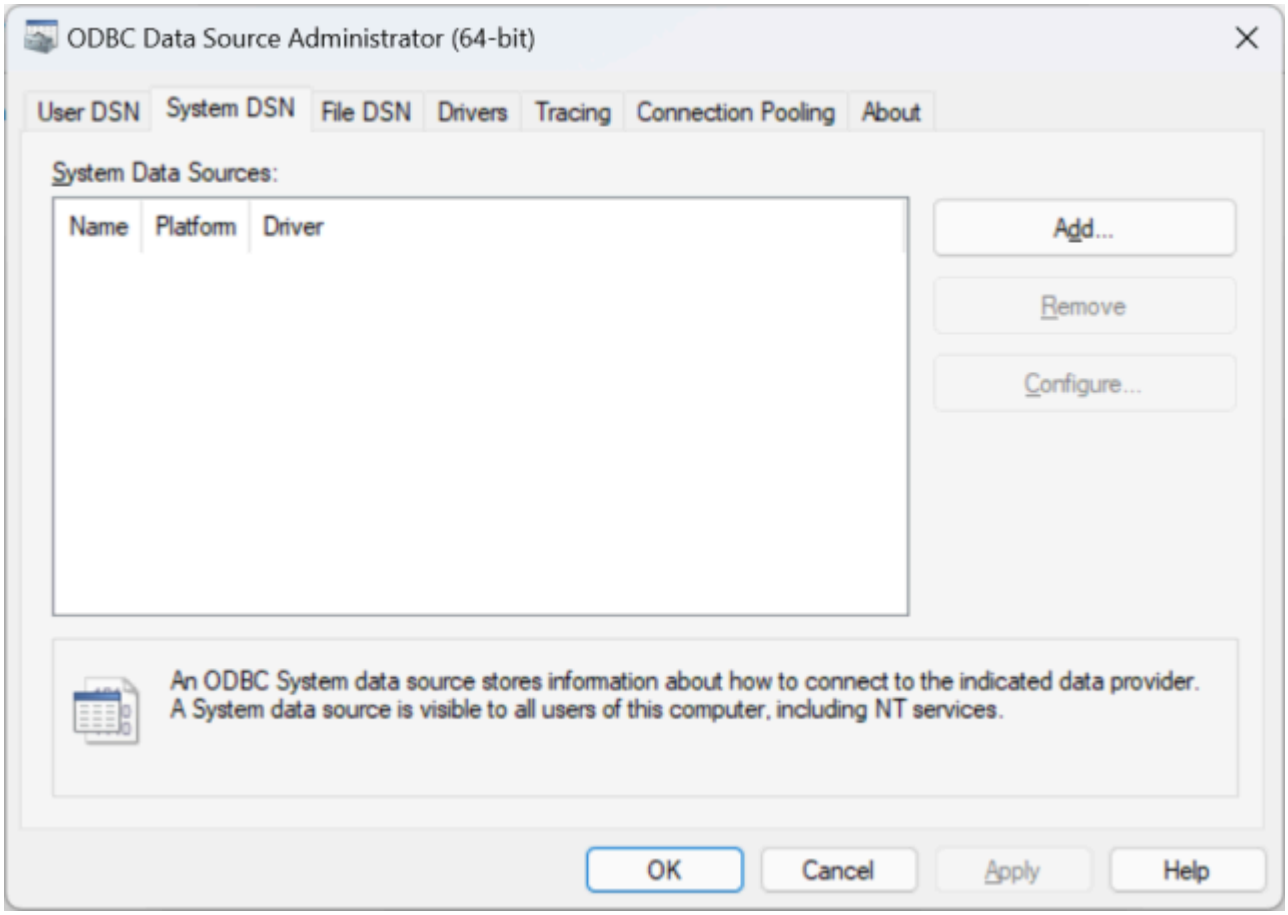
Menyiapkan Driver ODBC Amazon DocumentDB di Windows

Gunakan prosedur berikut untuk menyiapkan driver Amazon DocumentDB ODBC di windows:

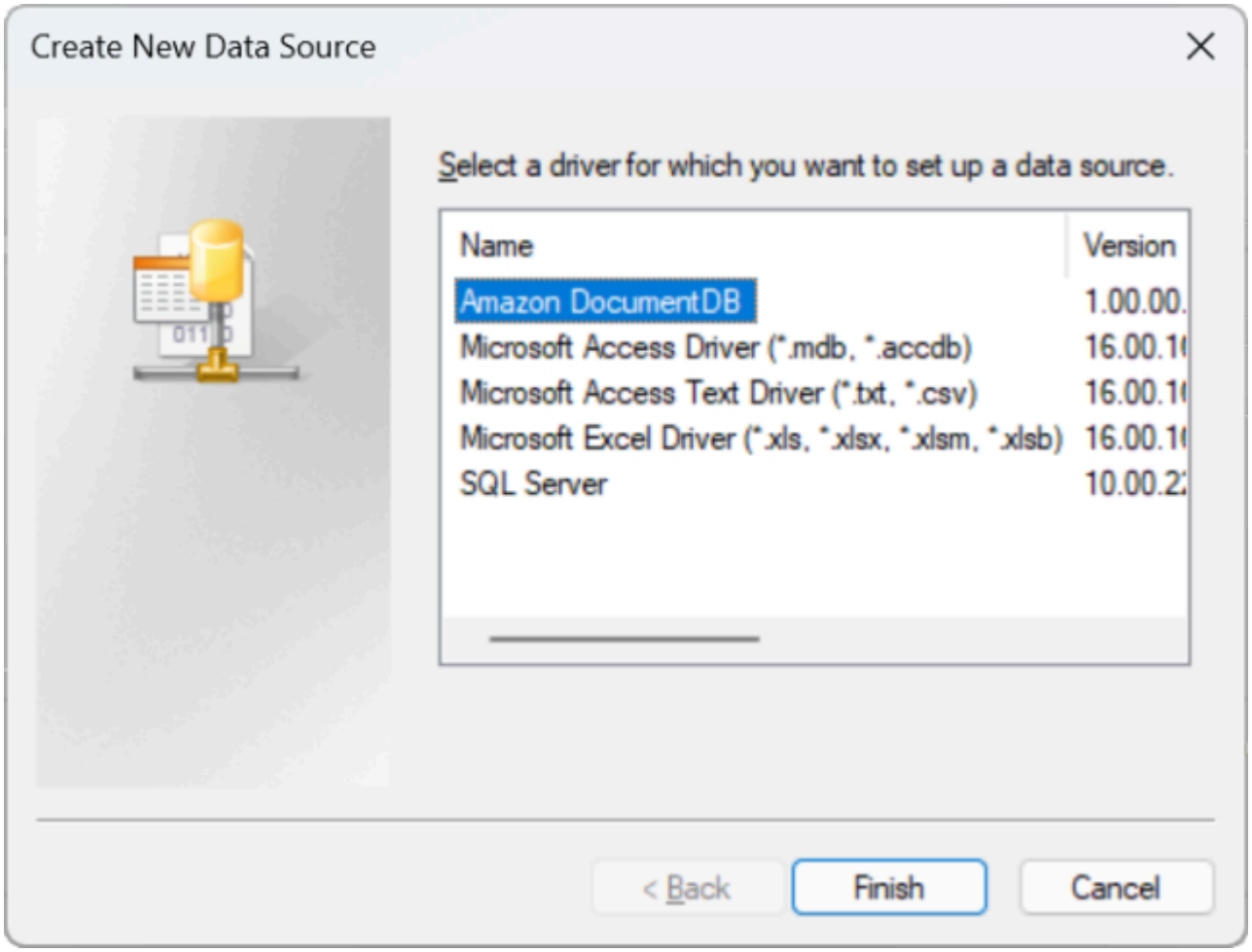
1. Buka Control Panel di Windows dan cari ODBC (atau di menu, pilih Windows Tools > Sumber Data ODBC (32-bit) atau Sumber Data ODBC (64-bit)):



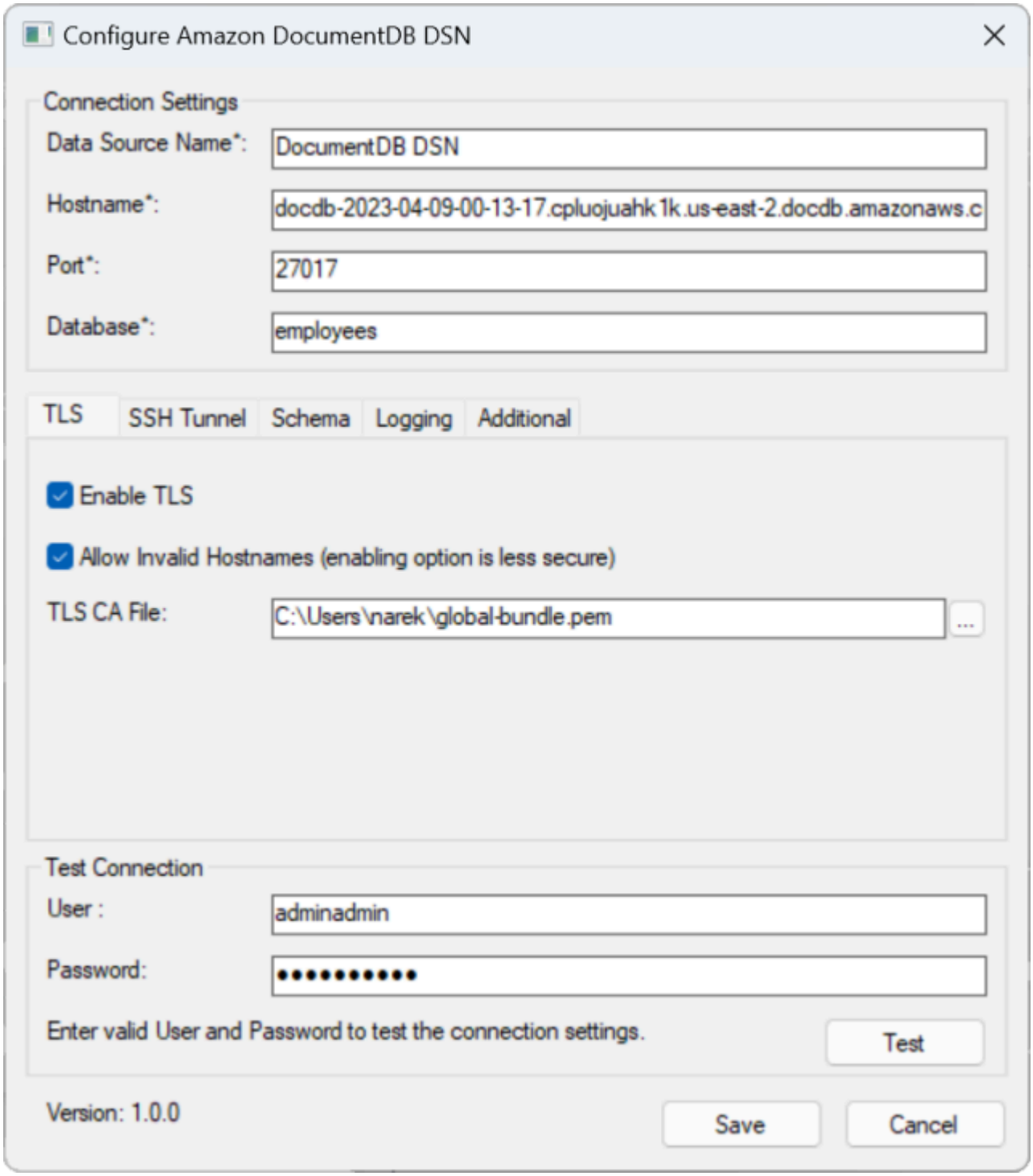
2. Pilih Administrator Sumber Data Driver ODBC yang sesuai: pilih versi 32-bit jika diinstal, jika tidak, pilih versi 64-bit.
3. Pilih tab Sytem DSN dan kemudian klik Add... untuk menambahkan DSN baru:



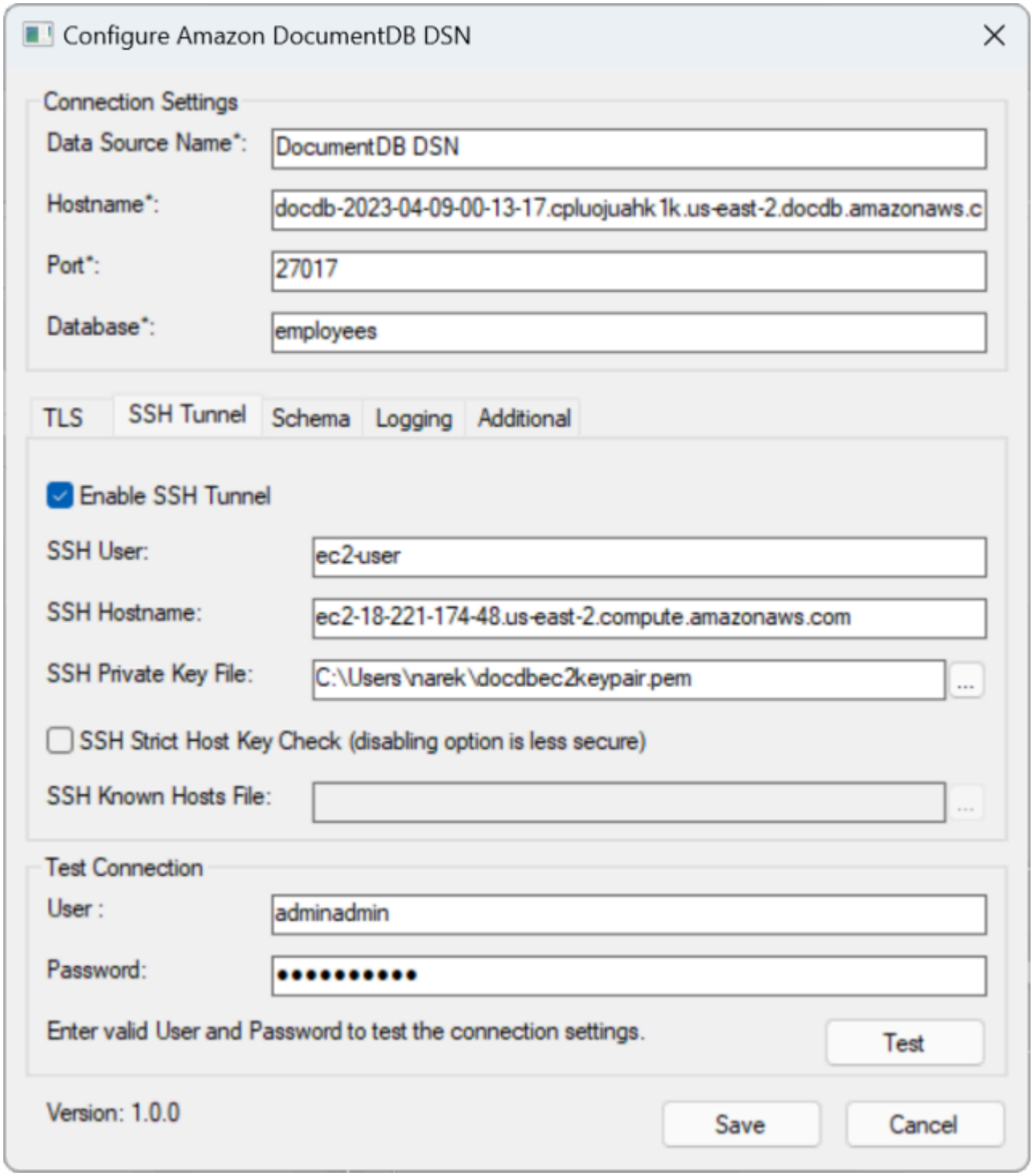
4. Pilih Amazon DocumentDB dari daftar driver sumber data:



5. Dalam dialog Konfigurasi Amazon DocumentDB DSN, lengkapi bidang Pengaturan Konfigurasi, tab TLS, dan Uji Koneksi, lalu klik Simpan:



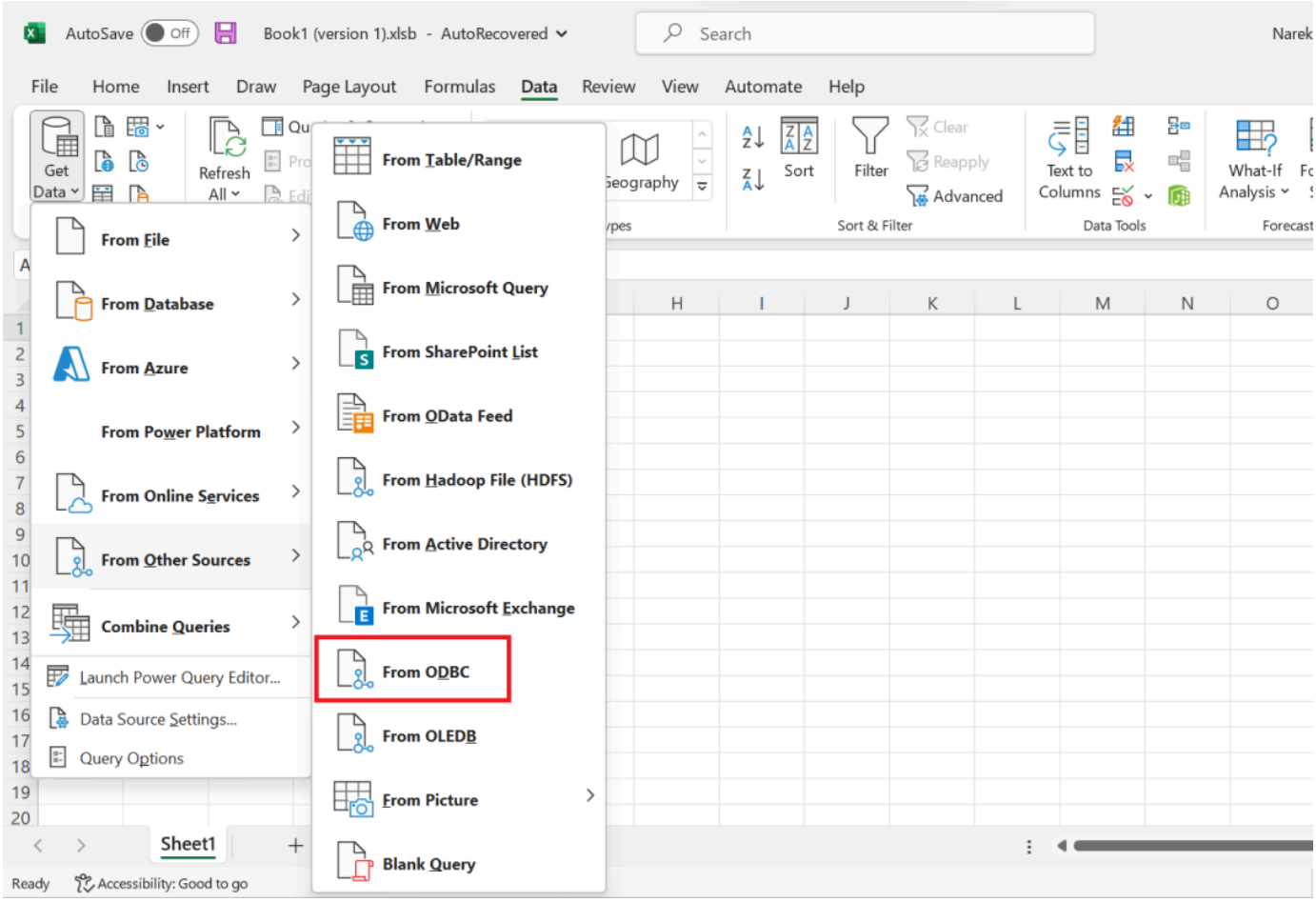
6. Pastikan Anda melengkapi formulir Windows secara akurat, karena detail koneksi akan berbeda tergantung pada metode tunneling SSH yang Anda pilih ke instans EC2. Lihat metode tunneling SSH di sini. Lihat Connection String Syntax dan Options untuk informasi lebih lanjut tentang masing-masing properti.



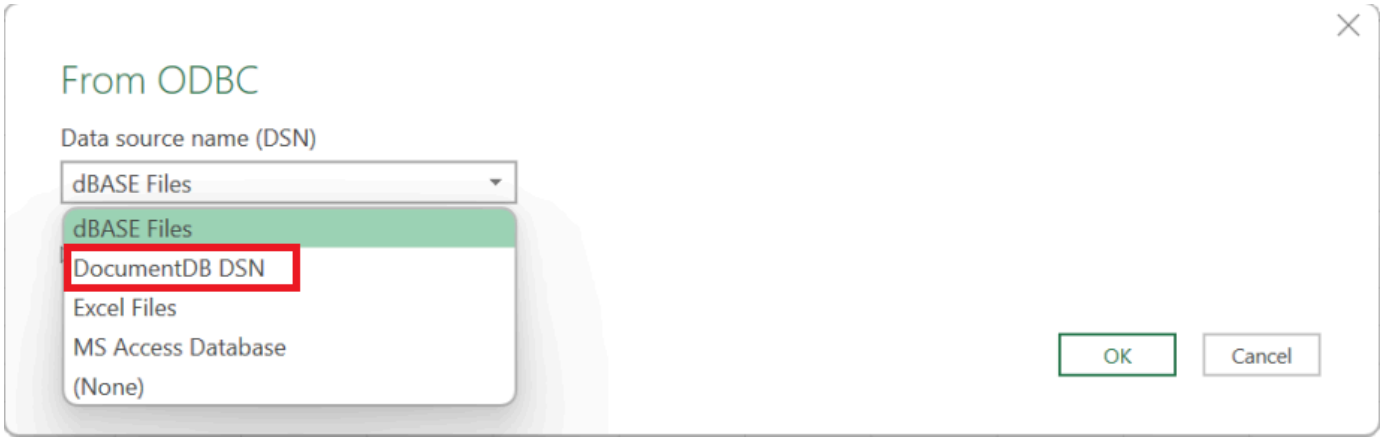
Untuk informasi selengkapnya tentang mengonfigurasi Driver ODBC Amazon DocumentDB di Windows, klik di sini.

Connect ke Amazon DocumentDB dari Microsoft Excel

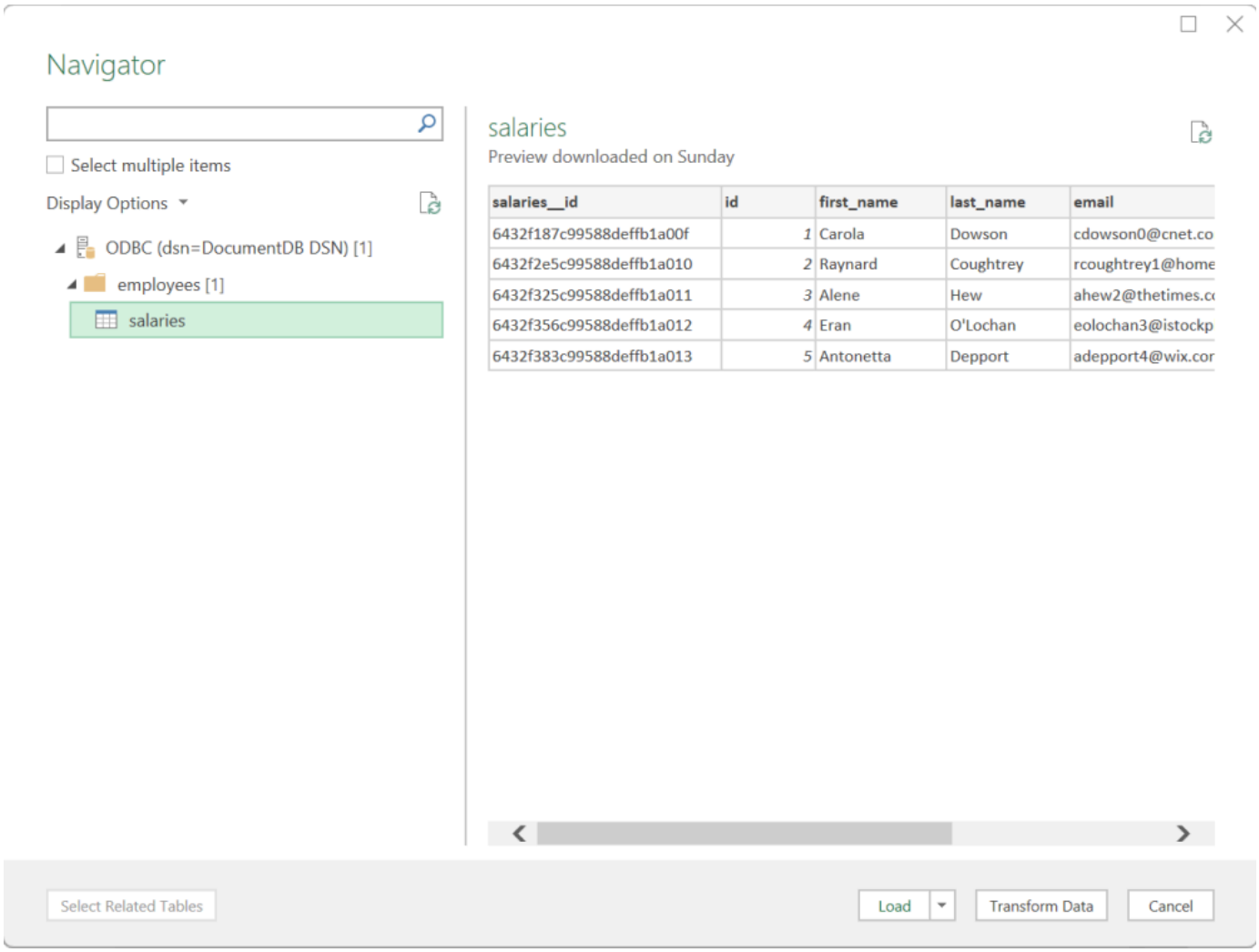
1. Pastikan bahwa driver Amazon DocumentDB telah diinstal dan dikonfigurasi dengan benar. Untuk informasi tambahan, lihat Menyiapkan driver ODBC di Windows.
2. Luncurkan Microsoft Excel.
3. Arahkan ke Data > Dapatkan Data > Dari Sumber Lain.
4. Pilih Dari ODBC:



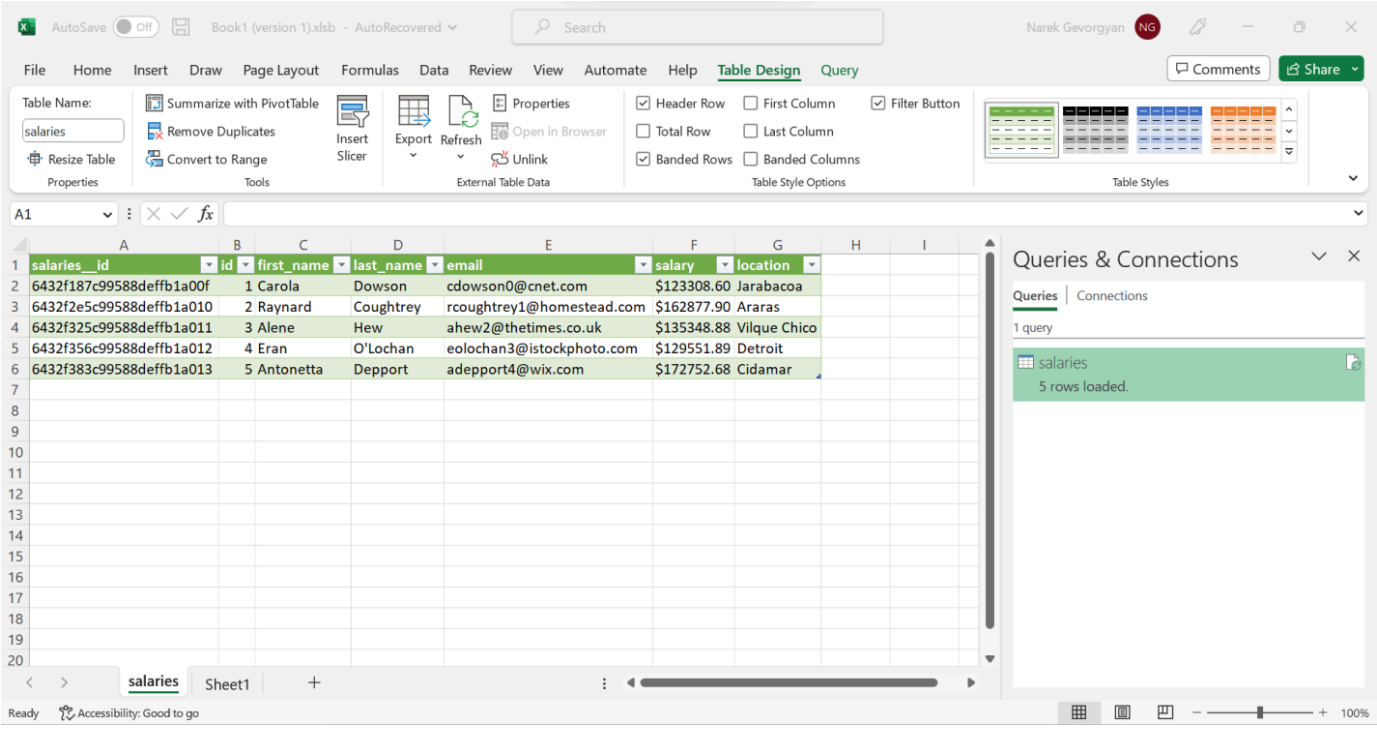
5. Pilih sumber data dari menu tarik-turun Nama sumber data (DSN) yang terkait dengan Amazon DocumentDB:



6. Pilih koleksi dari mana Anda ingin memuat data ke Excel:



7. Muat data ke Excel:



Connect ke Amazon DocumentDB dari Microsoft Power BI Desktop

Topik

- Prasyarat
- Menambahkan konektor kustom Microsoft Power BI Desktop
- Menghubungkan menggunakan konektor kustom Amazon DocumentDB
- Mengkonfigurasi Microsoft Power BI Gateway

Prasyarat

Sebelum memulai, pastikan driver ODBC Amazon DocumentDB diinstal dengan benar.

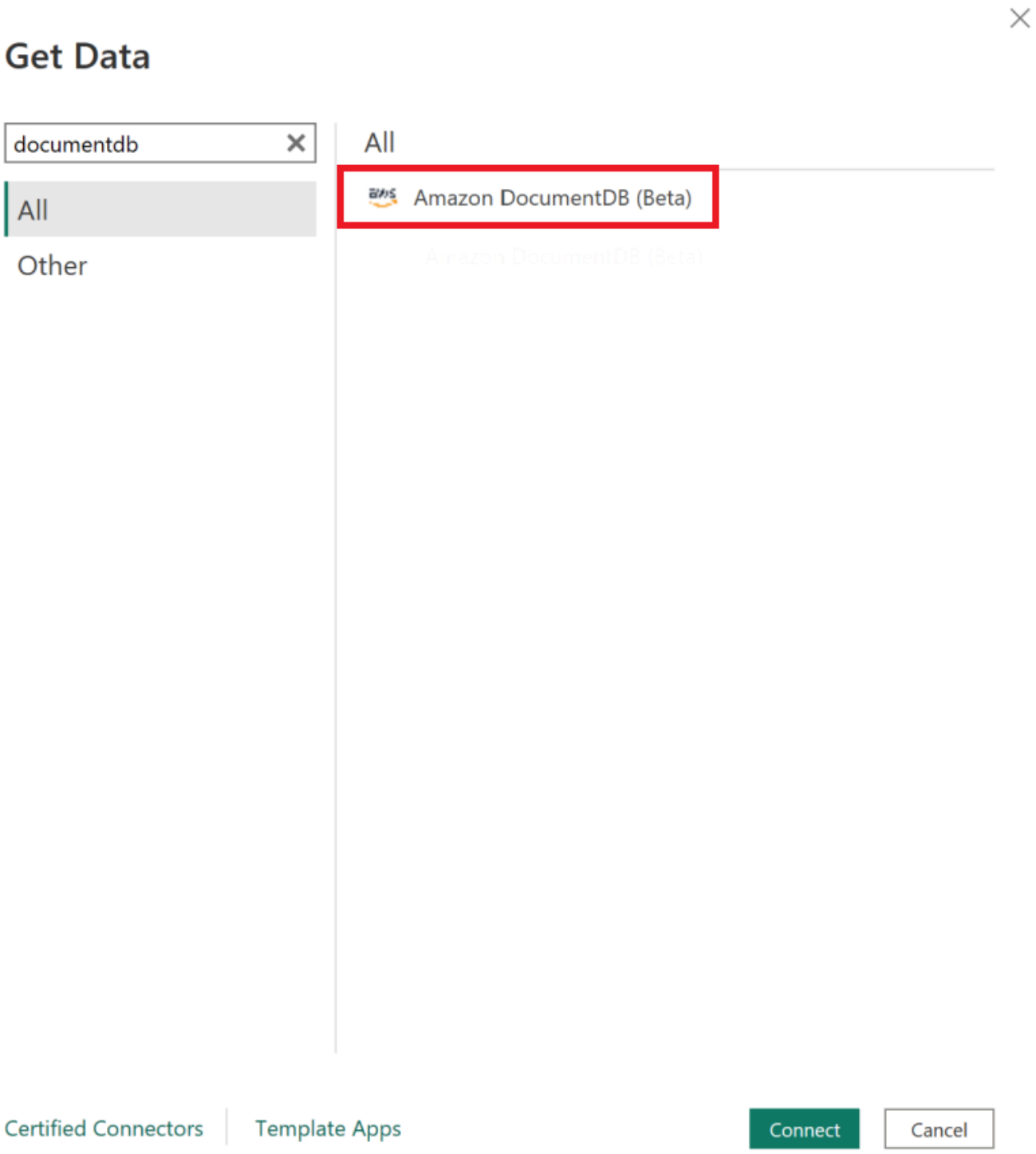
Menambahkan konektor kustom Microsoft Power BI Desktop

Salin`AmazonDocumentDBConnector.mez` file ke`<User>\Documents\Power BI Desktop \Custom Connectors\` folder (atau`<User>\OneDrive\Documents\Power BI Desktop \Custom Connectors` jika menggunakan OneDrive). Ini akan memungkinkan Power BI untuk mengakses konektor khusus. Anda bisa mendapatkan konektor ke Power BI Desktop di sini. Restart Power BI Desktop untuk memastikan konektor dimuat.

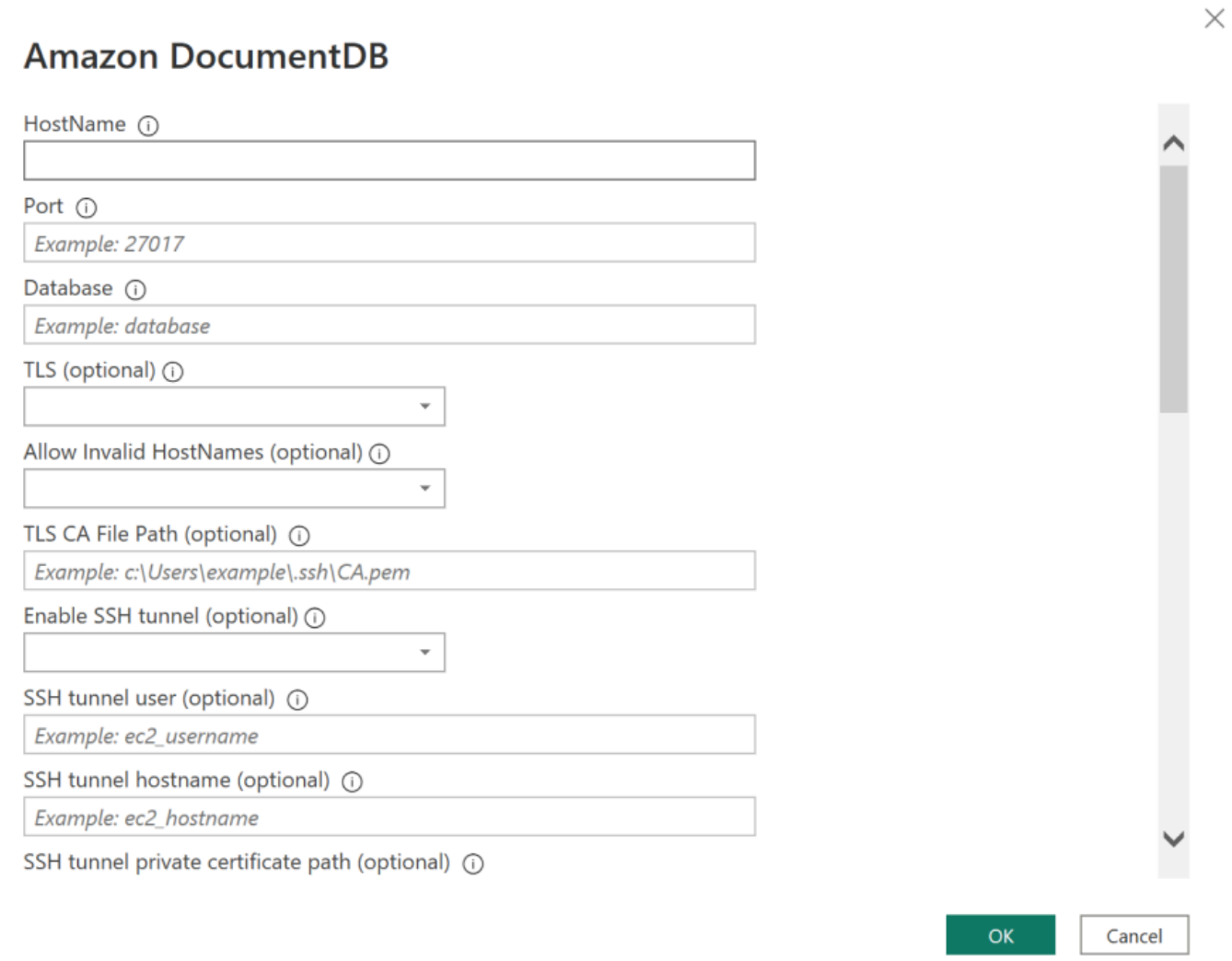
> **Note**
>
> Konektor khusus hanya mendukung nama pengguna dan kata sandi Amazon DocumentDB untuk otentikasi.

Menghubungkan menggunakan konektor kustom Amazon DocumentDB

1. Pilih Amazon DocumentDB (Beta) dari Get Data dan klik Connect. Jika Anda mendapat peringatan untuk menggunakan layanan pihak ketiga, klik Lanjutkan.



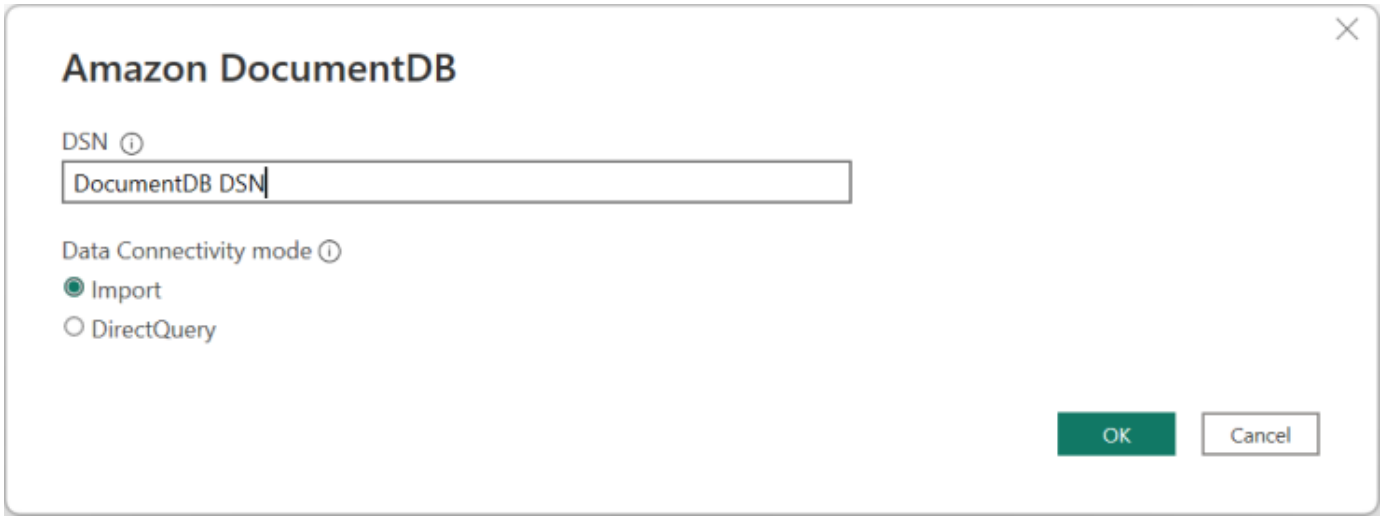
2. Masukkan semua informasi yang diperlukan untuk terhubung ke klaster Amazon DocumentDB Anda, lalu klik OK:



> **Note**
>
> Bergantung pada konfigurasi Nama Sumber Data (DSN) driver ODBC Anda, layar detail koneksi SSH mungkin tidak ditampilkan jika Anda telah memberikan informasi yang diperlukan dalam pengaturan DSN.

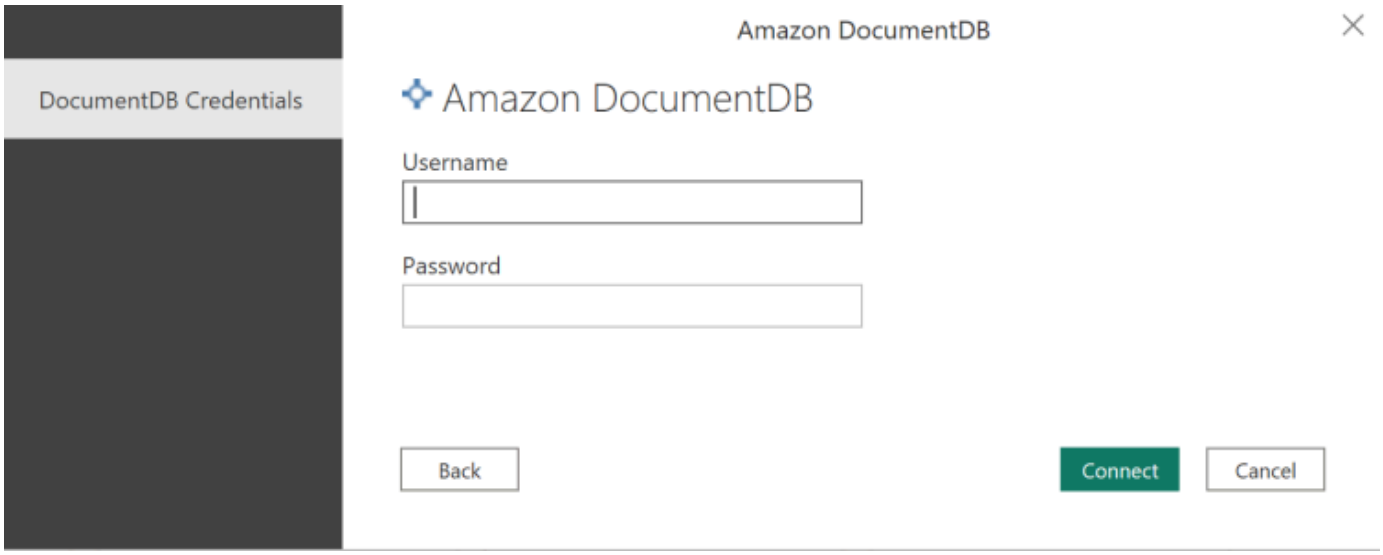
3. Pilih mode konektrivitas data:

   - Impor - memuat semua data dan menyimpan informasi pada disk. Data harus disegarkan dan dimuat ulang untuk menampilkan pembaruan data.
   - Direct Query - tidak memuat data, tetapi tidak hidup query pada data. Ini berarti bahwa data tidak perlu di-refresh dan dimuat ulang untuk menampilkan pembaruan data.

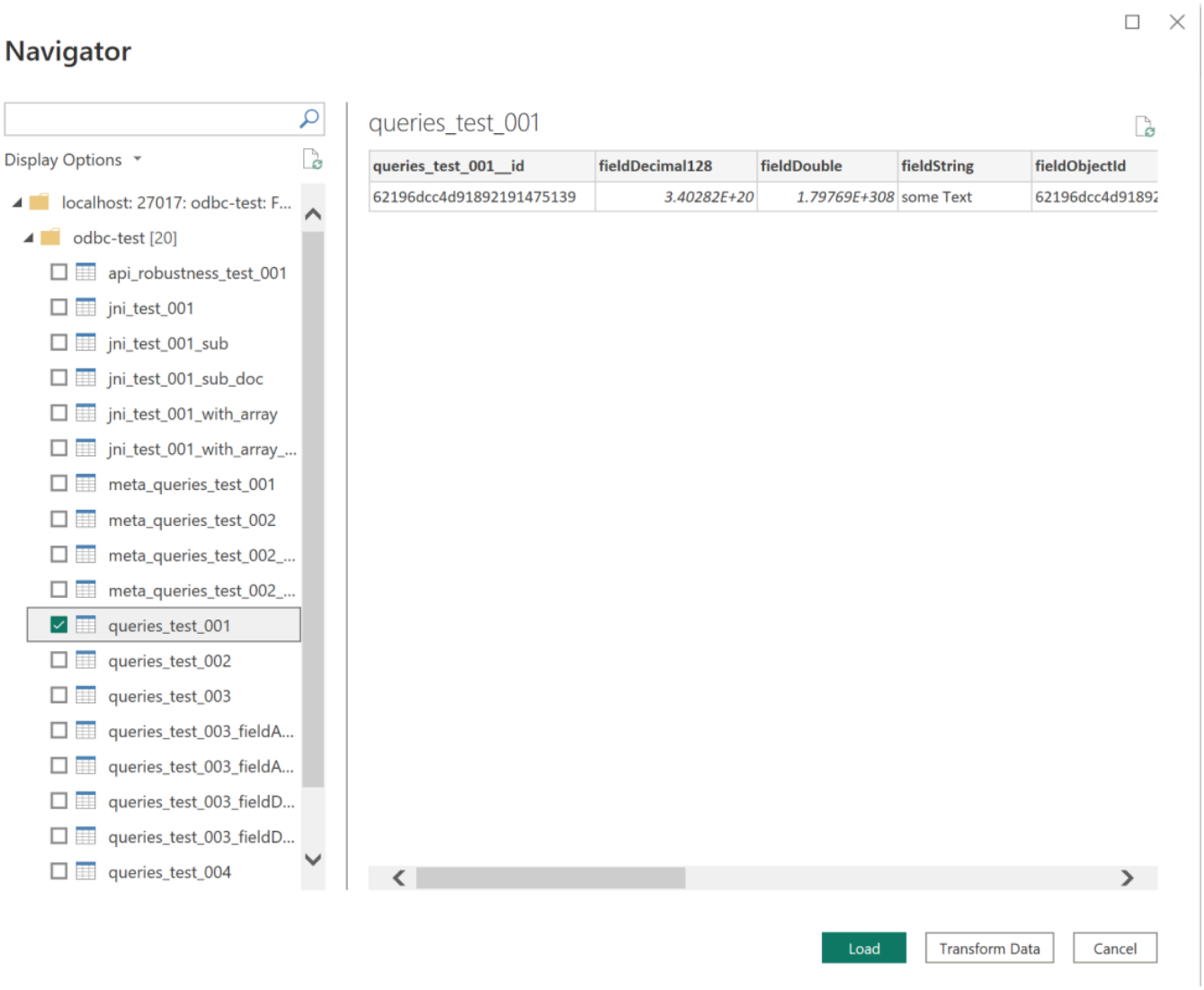


> **Note**
>
> Jika Anda menggunakan dataset yang sangat besar, mengimpor semua data mungkin membutuhkan waktu yang lebih lama.

4. Jika ini adalah pertama kalinya menghubungkan ke sumber data ini, pilih jenis otentikasi dan masukkan kredenial Anda saat diminta. Kemudian klik Connect:



5. Dalam Navigator dialog, pilih tabel database yang Anda inginkan, lalu klik memuat untuk memuat data atau Transform Data untuk terus mengubah data.



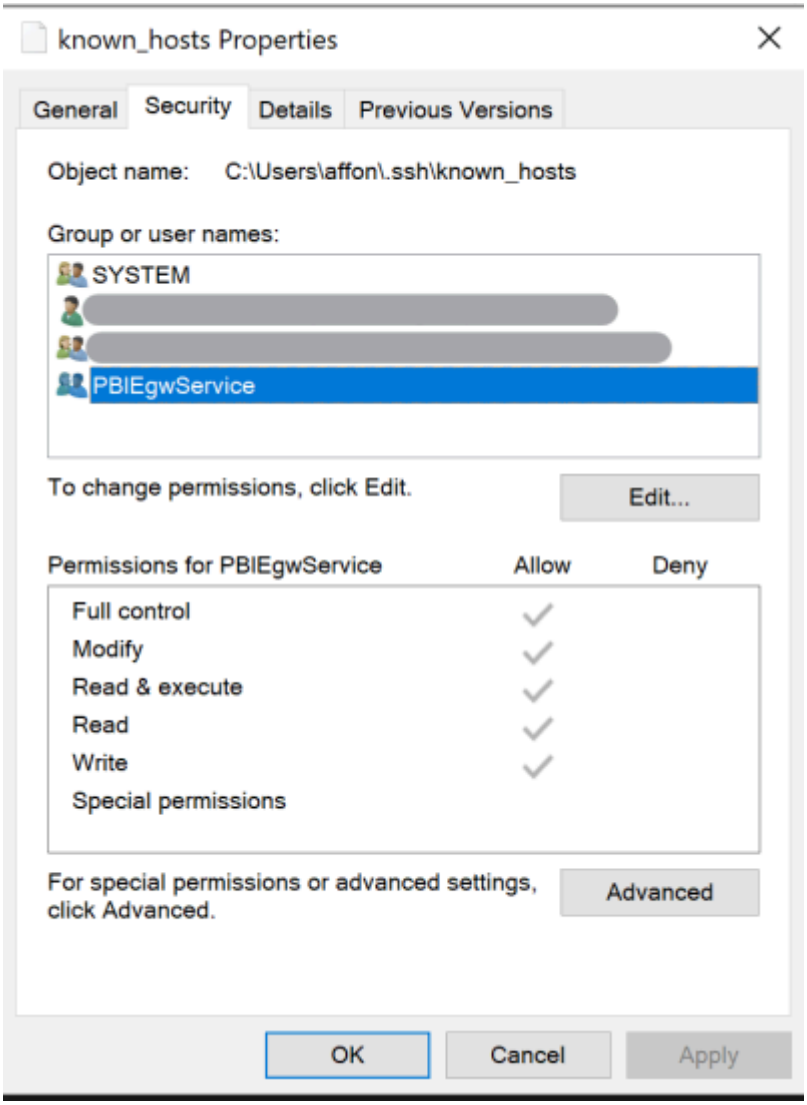
> **Note**
>
> Pengaturan sumber data Anda disimpan setelah Anda terhubung. Untuk memodifikasinya, pilih Transform Data > Pengaturan Sumber Data.

Mengkonfigurasi Microsoft Power BI Gateway

Prasyarat:

- Enure bahwa konektor kustom akan bekerja dengan Power BI Gateway.
- Pastikan bahwa ODBC DSN dibuat di sumber data ODBC di tab Sistem pada mesin tempat Power BI Gateway diinstal.

Jika Anda menggunakan fitur terowongan SSH internal, file`known_hosts` harus ditempatkan di mana akun layanan Power BI memiliki akses ke sana.



> **Note**
>
> Ini juga berlaku untuk file apa pun yang mungkin Anda perlukan untuk dapat membuat sambungan ke klaster Amazon DocumentDB Anda, seperti file sertifikat otoritas sertifikat (CA) (file pem).

Generasi skema otomatis

Driver ODBC menggunakan driver Amazon DocumentDB JDBC melalui JNI (Java Native Interface) - membuat fitur pembuatan skema otomatis untuk bekerja sama di driver JDBC. Untuk informasi

lebih lanjut tentang generasi skema otomatis, lihat JDBC generasi skema otomatis. Selain itu, untuk mempelajari lebih lanjut tentang arsitektur driver ODBC, klik di sini.

Support dan batasan SQL

Driver Amazon DocumentDB ODBC adalah driver hanya-baca yang mendukung subset SQL-92 dan beberapa ekstensi umum. Lihat dokumentasi dukungan dan batasan ODBC untuk informasi lebih lanjut.

Pemecahan Masalah

Jika Anda mengalami masalah saat menggunakan driver ODBC Amazon DocumentDB, lihat Panduan Pemecahan Masalah.

Kuota dan Batas Amazon DocumentDB

Topik ini menjelaskan kuota sumber daya, batas, dan batasan penamaan untuk Amazon DocumentDB (dengan kompatibilitas MongoDB).

Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS) dan Amazon Neptune.

Topik

- Tipe Instans Yang Didukung
- Wilayah yang Didukung
- Kuota Wilayah
- Batas Agregasi
- Batas Klaster
- Batas Instans
- Kendala Penamaan
- Kendala TTL
- Batas cluster elastis
- Batas pecahan cluster elastis
- CPU cluster elastis, memori, koneksi, dan batas kursor per pecahan

Tipe Instans Yang Didukung

Amazon DocumentDB mendukung instans sesuai permintaan dan jenis instans berikut:

- Memori teroptimalkan:
  - Jenis contoh R6G:`db.r6g.large`,,,`db.r6g.2xlarge`, `db.r6g.4xlargedb.r6g.8xlarge`,`db.r6g.12xlarge`. `db.r6g.16xlarge`
  - Tipe instans R5: `db.r5.large`,`db.r5.2xlarge`,`db.r5.4xlarge`,`db.r5.8xlarge`,`db.r5.12xlarge`,`db.r5.16xla` `db.r5.24xlarge`.
  - Tipe instans R4: `db.r4.large`,`db.r4.2xlarge`,`db.r4.4xlarge`,`db.r4.8xlarge`,`db.r4.16xlarge`.

- Performa Yang Dapat Diledakkan:
  - Jenis instans T4G:`db.t4g.medium`.
  - Tipe instans T3: `db.t3.medium`.

Untuk informasi lebih lanjut tentang jenis instans yang didukung dan spesifikasi mereka, lihat Spesifikasi Kelas Instans.

Wilayah yang Didukung

Amazon DocumentDB tersedia di wilayah berikut: AWS

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
AS Timur (Ohio)	`us-east-2`	3
AS Timur (Virginia Utara)	`us-east-1`	6
AS Barat (Oregon)	`us-west-2`	4
Amerika Selatan (Sao Paulo)	`sa-east-1`	3
Asia Pasifik (Hong Kong)	`ap-east-1`	3
Asia Pasifik (Hyderabad)	`ap-south-2`	3
Asia Pasifik (Mumbai)	`ap-south-1`	3
Asia Pasifik (Seoul)	`ap-northeast-2`	4
Asia Pasifik (Singapura)	`ap-southeast-1`	3
Asia Pasifik (Sydney)	`ap-southeast-2`	3
Asia Pasifik (Tokyo)	`ap-northeast-1`	3
Kanada (Pusat)	`ca-central-1`	3
Wilayah Tiongkok (Beijing)	`cn-north-1`	3
Tiongkok (Ningxia)	`cn-northwest-1`	3

Nama Wilayah	Wilayah	Availability Zone (penghitungan)
Eropa (Frankfurt)	`eu-central-1`	3
Eropa (Irlandia)	`eu-west-1`	3
Eropa (London)	`eu-west-2`	3
Eropa (Milan)	`eu-south-1`	3
Eropa (Paris)	`eu-west-3`	3
Timur Tengah (UEA)	`me-central-1`	3
AWS GovCloud (AS-Barat)	`us-gov-west-1`	3
AWS GovCloud (AS-Timur)	`us-gov-east-1`	3

Kuota Wilayah

Untuk fitur pengelolaan tertentu, Amazon DocumentDB menggunakan teknologi operasional yang dibagi dengan Amazon Relational Database Service (Amazon RDS). Tabel berikut berisi batas regional yang dibagi antara Amazon DocumentDB dan Amazon RDS.

> **Note**
>
> Teknologi bersama Amazon RDS yang dijelaskan di atas hanya berlaku untuk cluster berbasis instans Amazon DocumentDB. Cluster elastis Amazon DocumentDB tidak berbagi teknologi dengan Amazon RDS.

Batasan berikut berlaku untuk klaster berbasis instans Amazon DocumentDB dan per akun per wilayah. AWS

Sumber daya	AWS batas default
Klaster	40
Grup parameter klaster	50
Berlangganan acara	20
Instans	40
Snapshot klaster manual	100
Replika baca per klaster	15
Grup Subnet	50
Subnet per grup subnet	20
Tag per sumber daya	50
Grup keamanan VPC per instans	5

Batasan berikut berlaku untuk cluster elastis Amazon DocumentDB dan per akun per wilayah. AWS

Sumber daya	AWS batas default
Cluster elastis	20
Cluster elastis vCPU	1024
Cuplikan cluster elastis manual	20

Anda dapat menggunakan Service Quotas untuk meminta kenaikan kuota, jika kuota dapat disesuaikan. Beberapa permintaan diselesaikan secara otomatis, sementara yang lain dikirimkan ke AWS Support. Anda dapat melacak status permintaan peningkatan kuota yang dikirimkan AWS Support. Permintaan untuk meningkatkan service quotas tidak menerima prioritas dukungan. Jika Anda memiliki permintaan mendesak, silakan hubungi AWS Support. Untuk informasi lebih lanjut tentang Service Quotas, lihat Apa Itu Service Quotas?

Untuk meminta peningkatan kuota untuk Amazon DocumentDB:

1. Buka konsol Service Quotas di https://console.aws.amazon.com/servicequotas dan, jika perlu, masuk.
2. Di panel navigasi, pilih Layanan AWS .
3. Pilih Amazon DocumentDB (dengan kompatibilitas MongoDB) atau Amazon DocumentDB Elastic Cluster dari daftar, atau ketik di kolom pencarian.
4. Jika kuota dapat disesuaikan, Anda dapat memilih tombol radio atau namanya, lalu pilih Minta peningkatan kuota dari kanan atas halaman.
5. Untuk Ubah nilai kuota, masukkan nilai baru. Nilai baru lebih besar dari nilai saat ini.
6. Pilih Permintaan. Setelah permintaan diselesaikan, Nilai kuota yang diterapkanuntuk kuota tersebut diatur ke nilai baru.
7. Untuk melihat permintaan yang tertunda atau baru diselesaikan, pilih Dasbor dari panel navigasi. Untuk permintaan yang tertunda, pilih status permintaan untuk membuka tanda terima permintaan. Status awal dari permintaan adalah `Pending`. Setelah status berubah`Quota requested`, Anda akan melihat nomor kasus dengan AWS Support. Pilih nomor kasus untuk membuka tiket untuk permintaan Anda.

Batas Agregasi

Tabel berikut menjelaskan batas agregasi di Amazon DocumentDB.

Sumber daya	Kuota
Jumlah maksimum tahapan yang didukung	500

Batas Klaster

Tabel berikut menjelaskan batas klaster berbasis instans Amazon DocumentDB.

Sumber daya	Kuota
Ukuran klaster (jumlah dari semua koleksi dan indeks)	128 TiB
Ukuran koleksi (jumlah dari semua koleksi tidak dapat melebihi batas klaster) — tidak termasuk ukuran indeks	32 TB
Koleksi per klaster	100.000
Basis Data per klaster	100.000
Ukuran basis data (jumlah dari semua basis data tidak dapat melebihi batas klaster)	128 TiB
Kedalaman nest dokumen	200 tingkat
Ukuran dokumen	16 MB
Ukuran kunci indeks	2.048 byte
Indeks per koleksi	64
Kunci dalam indeks gabungan	32
Jumlah maksimum penulisan dalam perintah batch tunggal	100.000
Jumlah pengguna per klaster	1000

Batas Instans

Tabel berikut menjelaskan batas Amazon DocumentDB per instans.

Tipe Instans	Memori Instans (GiB)	Koneksi (semua)	Batas Kursor	Buka Transaksi	Koneksi (aktif)
T3.medium	4	500	30	50	102
T4g.Sedang	4	500	30	50	102
R4.large	15.25	1700	450	N/A	1100
R4.xlarge	30,5	3400	450	N/A	2700
R4.2xlarge	61	6800	450	N/A	4500
R4.4xlarge	122	13600	725	N/A	4500
R4.8xlarge	288	27200	1450	N/A	4500
R4.16xlarge	488	30000	2900	N/A	4500
R5.large	16	1700	450	200	1100
R5.xlarge	32	3500	450	400	2700
R5.2xlarge	64	7100	450	800	4500
R5.4xlarge	128	14200	760	1600	4500
R5.8xlarge	256	28400	1520	3200	4500
R5.12xlarge	383	30000	2280	4800	4500
R5.16xlarge	512	30000	3040	6400	4500
R5.24xlarge	768	30000	4560	9600	4500
R6g.besar	16	1700	450	200	1100

Tipe Instans	Memori Instans (GiB)	Koneksi (semua)	Batas Kursor	Buka Transaksi	Koneksi (aktif)
R6g.xLarge	32	3500	450	400	2700
R6G.2xLarge	64	7100	450	800	4500
R6G.4xLarge	128	14200	760	1600	4500
R6G.8xLarge	256	28400	1520	3200	4500
R6G.12xLarge	383	30000	2280	4800	4500
R6G.16xLarge	512	30000	3040	6400	4500

Anda dapat memantau dan alarm pada batas per instans menggunakan CloudWatch metrik berikut. Untuk selengkapnya tentang metrik Amazon CloudWatch DocumentDB, lihat. Memantau Amazon DocumentDB dengan CloudWatch

Kuota	CloudWatch Metrik
Memori Instans	FreeableMemory
Koneksi	DatabaseConnectionsMax
Cursors	DatabaseCursorsMax
Transaksi	TransactionsOpenMax

Kendala Penamaan

Tabel berikut menjelaskan kendala penamaan di Amazon DocumentDB.

Sumber daya	Batas default
Pengidentifikasi klaster	• Panjangnya adalah [1–63] huruf, angka, atau tanda hubung. • Karakter pertama harus berupa huruf. • Tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan. • Harus unik untuk semua cluster (di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB) per akun, per Wilayah. AWS
Nama koleksi: `<col>`	Panjangnya adalah [1–57] karakter.
Nama database: `<db>`	Panjangnya adalah [1–63] karakter.
Nama koleksi yang memenuhi syarat: `<db>.<col>`	Panjangnya adalah [3-120] karakter.
Nama indeks yang memenuhi syarat: `<db>.<col>.$<index>`	Panjangnya adalah [6-127] karakter.
Nama Indeks: `<col>$<index>`	Panjangnya adalah [3-63] karakter.
Pengidentifikasi instans	• Panjangnya adalah [1–63] huruf, angka, atau tanda hubung • Karakter pertama harus berupa huruf

Sumber daya	Batas default
	• Tidak boleh diakhiri dengan tanda hubung atau berisi dua tanda hubung yang berurutan • Harus unik untuk semua instans (di Amazon RDS, Amazon Neptune, dan Amazon DocumentDB) per akun, per Wilayah. AWS

Kata sandi master	• Panjangnya adalah [8-100] karakter ASCII yang dapat dicetak.
	• Dapat menggunakan karakter ASCII apa pun yang dapat dicetak kecuali untuk yang berikut ini:
	  • `/` (garis miring ke depan)
	  • `"` (tanda kutip ganda)
	  • `@` (simbol at)

Nama pengguna utama	• Panjangnya adalah [1-63] karakter alfanumerik. • Karakter pertamanya harus berupa huruf. • Tidak dapat berupa kata yang dicadangkan oleh mesin basis data.

Sumber daya	Batas default
Nama grup parameter	• Panjangnya adalah [1–255] karakter alfanumerik. • Karakter pertama harus berupa huruf. • Tidak boleh diakhiri dengan tanda hubung atau berisi dua tanda hubung berurutan.

Kendala TTL

Penghapusan dari indeks TTL tidak dijamin dalam jangka waktu tertentu dan merupakan upaya terbaik. Faktor-faktor seperti pemanfaatan sumber daya instans, ukuran dokumen, dan throughput keseluruhan dapat memengaruhi waktu penghapusan TTL.

Batas cluster elastis

Tabel berikut menjelaskan batas maksimum dalam cluster elastis Amazon DocumentDB.

Sumber daya	Kuota
Cluster elastis per wilayah	20
vCPU dijumlahkan di semua cluster elastis per wilayah	1024
Cuplikan cluster manual per wilayah	20
Pecahan per cluster	32
Penyimpanan per cluster (ketika data didistrib usikan secara merata oleh shard-key)	4 PiB
Koneksi ke cluster	Nilai yang lebih rendah dari 300.000 atau jumlah pecahan x batas koneksi yang terkait dengan vCPU per shard

Sumber daya	Kuota
UnSharded ukuran koleksi	32TB
Ukuran koleksi sharded (ketika data didistribusikan secara merata oleh shard-key)	1PB
Basis Data per klaster	10.000
UnSharded koleksi per cluster	100.000
Koleksi sharded per cluster	1000
Pengguna per cluster	100
Menulis dalam satu perintah batch	100.000
Indeks per koleksi	64
Kedalaman nest dokumen	100 tingkat
Ukuran dokumen	16MB
Ukuran kunci indeks	2048 byte
Kunci dalam indeks gabungan	32

Batas pecahan cluster elastis

Tabel berikut menjelaskan batas pecahan maksimum di cluster elastis Amazon DocumentDB.

Sumber daya	Kuota
vCPU per contoh pecahan	64
Contoh per pecahan	16
Strorage per shard	128 TiB
Penyimpanan per koleksi per pecahan	32TB

CPU cluster elastis, memori, koneksi, dan batas kursor per pecahan

Tabel berikut menjelaskan batas maksimum CPU, memori, koneksi, dan kursor dalam pecahan cluster elastis Amazon DocumentDB.

vCPU per pecahan	Memori instance (GiB)	Batas koneksi	Batas kursor
2	16	1700	450
4	32	3500	450
8	64	7100	450
16	128	14200	760
32	256	28400	1520
48	383	30000	2280
64	512	30000	3040

Melakukan kueri

Bagian ini menjelaskan semua aspek dari melakukan kueri dengan Amazon DocumentDB.

Topik

- Menanyakan dokumen
- Rencana kueri
- Jelaskan hasil
- Menanyakan data geospasial dengan Amazon DocumentDB
- Indeks sebagian
- Melakukan pencarian teks dengan Amazon DocumentDB

Menanyakan dokumen

Terkadang, Anda mungkin perlu mencari inventaris toko online Anda sehingga pelanggan dapat melihat dan membeli apa yang Anda jual. Melakukan kueri sebuah koleksi relatif mudah, apakah Anda ingin semua dokumen dalam koleksi atau hanya dokumen yang memenuhi kriteria tertentu.

Untuk melakukan kueri dokumen, gunakan operasi `find()`. Perintah `find()` memiliki parameter dokumen tunggal yang mendefinisikan kriteria yang akan digunakan dalam memilih dokumen yang akan dikembalikan. Output dari `find()` adalah dokumen yang diformat sebagai satu baris teks tanpa jeda baris. Untuk memformat dokumen output agar lebih mudah dibaca, gunakan `find().pretty()`. Semua contoh dalam topik ini menggunakan `.pretty()` untuk memformat output.

Contoh kode berikut menggunakan empat dokumen yang Anda sisipkan ke dalam koleksi `example` dalam dua latihan sebelumnya — `insertOne()` dan `insertMany()` yang terletak di bagian Menambahkan Dokumen pada Bekerja dengan Dokumen.

Topik

- Mengambil semua dokumen dalam koleksi
- Mengambil dokumen yang cocok dengan nilai bidang
- Mengambil dokumen yang cocok dengan dokumen yang disematkan
- Mengambil dokumen yang cocok dengan nilai bidang dalam dokumen yang disematkan

- Mengambil dokumen yang cocok dengan array
- Mengambil dokumen yang cocok dengan nilai dalam array
- Mengambil dokumen menggunakan operator

Mengambil semua dokumen dalam koleksi

Untuk mengambil semua dokumen dalam koleksi Anda, gunakan operasi `find()` dengan dokumen kueri kosong.

Kueri berikut mengembalikan semua dokumen di dalam koleksi `example`.

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

Mengambil dokumen yang cocok dengan nilai bidang

Untuk mengambil semua dokumen yang cocok dengan bidang dan nilai, gunakan operasi `find()` dengan dokumen kueri yang mengidentifikasi bidang dan nilai yang akan dicocokkan.

Menggunakan dokumen sebelumnya, kueri ini mengembalikan semua dokumen di mana bidang "Item" sama dengan "Pena".

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

Mengambil dokumen yang cocok dengan dokumen yang disematkan

Untuk menemukan semua dokumen yang cocok dengan dokumen yang ditanamkan, gunakan operasi `find()` dengan dokumen kueri yang menentukan nama dokumen yang ditanamkan dan semua bidang dan nilai untuk dokumen yang ditanamkan itu.

Saat mencocokkan dokumen yang ditanamkan, dokumen yang ditanamkan tersebut harus memiliki nama yang sama seperti dalam kueri. Selain itu, bidang dan nilai dalam dokumen yang ditanamkan harus cocok dengan kueri.

Kueri berikut hanya mengembalikan dokumen "Cat Poster". Ini karena "Pena" memiliki nilai yang berbeda untuk "`OnHand`" dan "`MinOnHand`", dan "Cat Semprot" memiliki satu bidang (`OrderQnty`) lebih banyak daripada dokumen kueri.

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

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

Mengambil dokumen yang cocok dengan nilai bidang dalam dokumen yang disematkan

Untuk menemukan semua dokumen yang cocok dengan dokumen yang ditanamkan, gunakan operasi `find()` dengan dokumen kueri yang menentukan nama dokumen yang ditanamkan dan semua bidang dan nilai untuk dokumen yang ditanamkan itu.

Mengingat dokumen sebelumnya, kueri berikut menggunakan "notasi titik" untuk menentukan dokumen yang ditanamkan dan bidang yang diminati. Setiap dokumen yang cocok dengan ini dikembalikan, terlepas dari apa bidang lain mungkin hadir dalam dokumen tertanam. Kueri mengembalikan "Cat Poster" dan "Cat Semprot" karena keduanya cocok dengan bidang dan nilai yang ditentukan.

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

Mengambil dokumen yang cocok dengan array

Untuk menemukan semua dokumen yang cocok dengan array, gunakan operasi `find()` dengan nama array yang Anda minati dan semua nilai dalam array itu. Kueri mengembalikan semua dokumen yang memiliki array dengan nama itu di mana nilai arraynya identik dengan dan dalam urutan yang sama seperti dalam kueri.

Kueri berikut hanya mengembalikan "Pena" karena "Cat Poster" memiliki warna tambahan (Putih), dan "Cat Semprot" memiliki warna dalam urutan yang berbeda.

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

Mengambil dokumen yang cocok dengan nilai dalam array

Untuk menemukan semua dokumen yang memiliki nilai array tertentu, gunakan operasi `find()` dengan nama array dan nilai yang Anda minati.

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

Operasi sebelumnya mengembalikan ketiga dokumen karena masing-masing memiliki array bernama `Colors` dan nilai "`Red`" di suatu tempat dalam array. Jika Anda menentukan nilai "`White`", kueri hanya akan mengembalikan "Cat Poster."

Mengambil dokumen menggunakan operator

Kueri berikut mengembalikan semua dokumen dengan nilai "`Inventory.OnHand`" kurang dari 50.

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

Untuk daftar operator kueri yang didukung, lihat Kueri dan Proyeksi Operator.

Rencana kueri

Bagaimana Saya Dapat Melihat **`executionStats`** untuk Rencana Kueri?

Saat menentukan mengapa kueri dijalankan lebih lambat dari yang diharapkan, akan berguna untuk memahami apa `executionStats` untuk rencana kueri. `executionStats` memberikan jumlah dokumen yang dikembalikan dari tahap tertentu (`nReturned`), jumlah waktu eksekusi yang dihabiskan pada setiap tahap (`executionTimeMillisEstimate`), dan jumlah waktu yang diperlukan untuk menghasilkan rencana kueri (`planningTimeMillis`). Anda dapat menentukan tahapan kueri yang paling memakan waktu untuk membantu memfokuskan upaya pengoptimalan Anda dari keluaran `executionStats`, seperti yang ditunjukkan dalam contoh kueri di bawah ini. Parameter `executionStats` saat ini tidak mendukung perintah `update` dan `delete`.

> **Note**
>
> Amazon DocumentDB mengemulasi MongoDB 3.6 API pada mesin basis data yang dibuat khusus yang menggunakan sistem penyimpanan terdistribusi, toleran terhadap kesalahan, dan dapat pulih sendiri. Akibatnya, rencana kueri dan keluaran dari `explain()` mungkin berbeda antara Amazon DocumentDB dan MongoDB. Pelanggan yang ingin kontrol atas rencana kueri mereka dapat menggunakan operator `$hint` untuk memaksa pemilihan indeks yang diutamakan.

Jalankan kueri yang ingin Anda tingkatkan di bawah perintah `explain()` sebagai berikut.

```
db.runCommand({explain: {query document}}).
explain("executionStats").executionStats;
```

Berikut ini adalah contoh operasi.

```
db.fish.find({}).limit(2).explain("executionStats");
```

Output dari operasi ini terlihat seperti berikut.

```
{
    "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
}
```

Jika Anda tertarik hanya melihat `executionStats` dari kueri di atas, Anda dapat menggunakan perintah berikut. Untuk koleksi kecil, prosesor kueri Amazon DocumentDB dapat memilih untuk tidak menggunakan indeks jika peningkatan performa dapat diabaikan.

```
db.fish.find({}).limit(2).explain("executionStats").executionStats;
```

Cache rencana kueri

Untuk mengoptimalkan performa dan mengurangi durasi perencanaan, Amazon DocumentDB melakukan cache rencana kueri secara internal. Ini memungkinkan kueri dengan bentuk yang sama untuk dieksekusi secara langsung menggunakan rencana cache.

Namun, cache ini terkadang menyebabkan penundaan acak untuk kueri yang sama; misalnya, kueri yang biasanya membutuhkan waktu satu detik untuk dijalankan terkadang membutuhkan waktu sepuluh detik. Ini karena seiring waktu, instans pembaca melakukan cache berbagai bentuk kueri, sehingga menghabiskan memori. Jika Anda mengalami kelambatan acak ini, tidak ada tindakan yang perlu Anda lakukan untuk melepaskan memori--sistem akan mengelola penggunaan memori untuk Anda dan begitu memori mencapai ambang batas tertentu, memori akan dilepaskan secara otomatis.

Jelaskan hasil

Jika Anda ingin mengembalikan informasi tentang rencana kueri, Amazon DocumentDB mendukung mode verbositas `queryPlanner`. Hasil `explain` mengembalikan rencana kueri terpilih yang dipilih oleh pengoptimal dalam format yang mirip dengan berikut ini:

```
{
   "queryPlanner" : {
      "plannerVersion" : <int>,
      "namespace" : <string>,
```

```
      "winningPlan" : {
         "stage" : <STAGE1>,
         ...
         "inputStage" : {
            "stage" : <STAGE2>,
            ...
            "inputStage" : {
               ...
            }
         }
      }
   }
}
```

Bagian berikut akan menentukan hasil `explain` umum.

Topik

- Tahap pemindaian dan filter
- Persimpangan indeks
- Serikat indeks
- Beberapa persilangan indeks/serikat
- Indeks majemuk
- Sortir tahap
- Panggung grup

Tahap pemindaian dan filter

Pengoptimal dapat memilih salah satu pemindaian berikut:

COLLSCAN

Tahap ini adalah pemindaian koleksi sekuensial.

```
{
    "stage" : "COLLSCAN"
}
```

IXSCAN

Tahap ini memindai kunci indeks. Pengoptimal dapat mengambil dokumen dalam tahap ini dan ini dapat mengakibatkan tahap FETCH ditambahkan kemudian.

```
db.foo.find({"a": 1})
{
    "stage" : "IXSCAN",
    "direction" : "forward",
    "indexName" : <idx_name>
}
```

FETCH

Jika pengoptimal mengambil dokumen dalam tahap selain IXSCAN, hasilnya akan menyertakan tahap FETCH. Misalnya, kueri IXSCAN di atas dapat menghasilkan kombinasi tahapan FETCH dan IXSCAN:

```
db.foo.find({"a": 1})
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXSCAN",
        "indexName" : <idx_name>
    }
}
```

IXONLYSCAN hanya memindai kunci indeks. Buat indeks gabungan tidak akan menghindari FETCH.

Persimpangan indeks

IXAND

Amazon DocumentDB dapat menyertakan tahap IXAND dengan array inputStages IXSCAN jika dapat memanfaatkan persimpangan indeks. Misalnya, kita mungkin melihat output seperti:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXAND",
```

```
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Serikat indeks

IXOR

Mirip dengan persimpangan indeks, Amazon DocumentDB dapat menyertakan tahap `IXOR` dengan array `inputStages` untuk operator `$or`.

```
db.foo.find({"$or": [{"a": {"$gt": 2}}, {"b": {"$lt": 2}}]})
```

Untuk kueri di atas, output yang dijelaskan mungkin terlihat seperti ini:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Beberapa persilangan indeks/serikat

Amazon DocumentDB dapat menggabungkan beberapa tahap persimpangan atau persatuan indeks bersama-sama kemudian mengambil hasilnya. Sebagai contoh:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                ...
            },
            {
                "stage" : "IXAND",
                "inputStages" : [
                    {
                        "stage" : "IXSCAN",
                        ...
                    },
                    {
                        "stage" : "IXSCAN",
                        ...
                    }
                ]
            }
        ]
    }
}
```

Penggunaan tahap indeks persimpangan atau persatuan tidak dipengaruhi oleh jenis indeks (jarang, majemuk, dll).

Indeks majemuk

Penggunaan indeks gabungan Amazon DocumentDB tidak dibatasi pada subset awal bidang indeks; itu dapat menggunakan indeks dengan bagian akhiran tetapi mungkin tidak terlalu efisien.

Misalnya, indeks gabungan `{ a: 1, b: -1 }` dapat mendukung ketiga kueri di bawah ini:

`db.orders.find( { a: 1 } } )`

```
db.orders.find( { b: 1 } } )

db.orders.find( { a: 1, b: 1 } } )
```

Sortir tahap

Jika ada indeks pada kunci sortir yang diminta, Amazon DocumentDB dapat menggunakan indeks untuk mendapatkan pesanan. Dalam hal ini, hasilnya tidak akan mencakup tahap `SORT`, melainkan tahap `IXSCAN`. Jika pengoptimal menyukai pengurutan biasa, itu akan mencakup tahap seperti ini:

```
{
    "stage" : "SORT",
    "sortPattern" : {
        "a" : 1,
        "b" : -1
    }
}
```

Panggung grup

Amazon DocumentDB mendukung dua strategi grup yang berbeda:

- `SORT_AGGREGATE`: Pada agregat pengurutan disk.
- `HASH_AGGREGATE`: Dalam agregat hash memori.

Menanyakan data geospasial dengan Amazon DocumentDB

Bagian ini mencakup bagaimana Anda dapat menanyakan data Geospasial dengan Amazon DocumentDB. Setelah Anda membaca bagian ini, Anda akan dapat menjawab bagaimana menyimpan, menanyakan, dan mengindeks data Geospasial di Amazon DocumentDB.

Topik

- Gambaran Umum
- Mengindeks dan menyimpan data geospasial
- Meminta data geospasial
- Batasan

Gambaran Umum

Kasus penggunaan umum untuk Geospasial melibatkan analisis kedekatan dari data Anda. Misalnya, “menemukan semua bandara dalam jarak 50 mil dari Seattle”, atau “temukan restoran terdekat dari lokasi tertentu”. Amazon DocumentDB menggunakan spesifikasi GeoJSON untuk mewakili data geospasial. GeoJSON adalah spesifikasi open-source untuk format JSON bentuk dalam ruang koordinat. Koordinat GeoJSON menangkap bujur dan lintang, mewakili posisi pada bola mirip bumi.

Mengindeks dan menyimpan data geospasial

Amazon DocumentDB menggunakan tipe GeoJSON 'Point' untuk menyimpan data geospasial. Setiap dokumen GeoJSON (atau subdokumen) umumnya terdiri dari dua bidang:

- type - bentuk yang diwakili, yang menginformasikan Amazon DocumentDB bagaimana menafsirkan bidang “koordinat”. Saat ini, Amazon DocumentDB hanya mendukung poin
- koordinat — pasangan lintang dan bujur direpresentasikan sebagai objek dalam array — [bujur, lintang]

Amazon DocumentDB juga menggunakan indeks 2dsphere untuk mengindeks data Geospasial. Amazon DocumentDB mendukung poin pengindeksan. Amazon DocumentDB mendukung query proximity dengan pengindeksan 2dsphere.

Mari pertimbangkan skenario di mana Anda sedang membangun aplikasi untuk layanan pengiriman makanan. Anda ingin menyimpan berbagai lintang dan pasangan bujur restoran di Amazon DocumentDB. Untuk melakukannya, pertama-tama kami sarankan Anda membuat indeks pada bidang Geospasial yang memegang pasangan lintang dan bujur.

```
use restaurantsdb
db.usarestaurants.createIndex({location:"2dsphere"})
```

Output dari perintah ini akan terlihat seperti ini:

```
{
 "createdCollectionAutomatically" : true,
 "numIndexesBefore" : 1,
 "numIndexesAfter" : 2,
 "ok" : 1
}
```

Setelah Anda membuat indeks, Anda dapat mulai memasukkan data ke dalam koleksi Amazon DocumentDB Anda.

```
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
      ]
   }
```

```
});
```

Meminta data geospasial

Amazon DocumentDB mendukung kueri kedekatan, inklusi, dan persimpangan data Geospasial. Contoh yang baik dari kueri kedekatan adalah menemukan semua titik (semua bandara) yang kurang dari jarak tertentu dan lebih dari jarak dari titik lain (kota). Contoh yang baik dari kueri inklusi adalah menemukan semua titik (semua bandara) yang terletak di area/poligon tertentu (negara bagian New York). Contoh yang baik dari kueri persimpangan adalah menemukan poligon (status) yang berpotongan dengan titik (kota). Anda dapat menggunakan operator Geospasial berikut untuk mendapatkan wawasan dari data Anda.

- **`$nearSphere`**- `$nearSphere` adalah operator find yang mendukung pencarian titik dari terdekat ke terjauh dari titik GeoJSON.
- **`$geoNear`**- `$geoNear` adalah operator agregasi yang mendukung penghitungan jarak dalam meter dari titik GeoJSON.
- **`$minDistance`**- `$minDistance` adalah operator find yang digunakan bersama dengan `$nearSphere` atau `$geoNear` untuk memfilter dokumen yang setidaknya pada jarak minimum yang ditentukan dari titik pusat.
- **`$maxDistance`**- `$maxDistance` adalah operator find yang digunakan bersama dengan `$nearSphere` atau `$geoNear` untuk memfilter dokumen yang paling banyak pada jarak maksimum yang ditentukan dari titik pusat.
- **`$geoWithin`**- `$geoWithin` adalah operator find yang mendukung pencarian dokumen dengan data geospasial yang ada seluruhnya dalam bentuk tertentu seperti poligon.
- **`$geoIntersects`**- `$geoIntersects` adalah operator find yang mendukung pencarian dokumen yang data geospasialnya berpotongan dengan objek GeoJSON tertentu.

> **Note**
>
> `$geoNear`dan `$nearSphere` memerlukan indeks 2dsphere pada bidang GeoJSON yang Anda gunakan dalam kueri kedekatan Anda.

Contoh 1

Dalam contoh ini, Anda akan belajar bagaimana menemukan semua restoran (titik) yang diurutkan berdasarkan jarak terdekat dari alamat (titik).

Untuk melakukan kueri seperti itu, Anda dapat menggunakan `$geoNear` untuk menghitung jarak set titik dari titik lain. Anda juga dapat menambahkan `distanceMultiplier` untuk mengukur jarak dalam kilometer.

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001
      }
   }
])
```

Perintah di atas akan mengembalikan restoran yang diurutkan berdasarkan jarak (paling dekat dengan terjauh) dari titik yang ditentukan. Output dari perintah ini akan terlihat seperti ini

```
{ "_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 }
```

Untuk membatasi jumlah hasil dalam kueri, gunakan num opsi `limit` or.

`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**
>
> `$geoNear`tahap mendukung `limit` dan `num` pilihan untuk menentukan jumlah maksimum dokumen untuk kembali. `$geoNear`mengembalikan maksimal 100 dokumen secara default

jika num opsi `limit` atau tidak ditentukan. Ini diganti dengan nilai `$limit` tahap jika ada dan nilainya kurang dari 100.

Contoh 2

Dalam contoh ini, Anda akan belajar cara menemukan semua restoran (poin) dalam jarak 2 kilometer dari alamat (titik) tertentu. Untuk melakukan kueri seperti itu, Anda dapat menggunakan `$nearSphere` dalam minimum `$minDistance` dan maksimum `$maxDistance` dari GeoJSON Point

```
db.usarestaurants.find({
   "location":{
      "$nearSphere":{
         "$geometry":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "$minDistance":1,
         "$maxDistance":2000
      }
   }
},
{
   "name":1
})
```

Perintah di atas akan mengembalikan restoran pada jarak maksimum 2 kilometer dari titik yang ditentukan. Output dari perintah ini akan terlihat seperti ini

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "name" : "Noodle House" }
```

Batasan

Amazon DocumentDB tidak mendukung kueri atau pengindeksan LineString Poligon,,,,, dan. MultiPoint MultiPolygon MultiLineString GeometryCollection

Indeks sebagian

Indeks sebagian mengindeks dokumen dalam koleksi yang memenuhi kriteria filter tertentu. Fitur indeks sebagian didukung di kluster berbasis instans Amazon DocumentDB 5.0.

Topik

- Buat indeks sebagian
- Operator yang didukung
- Kueri menggunakan indeks paral
- Fungsionalitas indeks sebagian
- Keterbatasan indeks sebagian

Buat indeks sebagian

Untuk membuat indeks sebagian, gunakan `createIndex()` metode dengan `partialFilterExpression` opsi. Misalnya, operasi berikut membuat indeks gabungan unik dalam koleksi pesanan yang mengindeks dokumen yang memiliki `OrderID` dan memiliki `isDelivered` bidang sebagai true:

```
db.orders.createIndex(
  {"category": 1, "CustomerId": 1, "OrderId": 1},
  {"unique": true, "partialFilterExpression":
    {"$and": [
      {"OrderId": {"$exists": true}},
      {"isDelivered": {"$eq": false}}
    ]}
  }
)
```

Operator yang didukung

- $ eq
- $ ada
- $ dan (hanya di tingkat atas)
- $gt/$gte/$lt/$lte (pemindaian indeks hanya digunakan ketika filter, diprediksikan dalam kueri, cocok dengan ekspresi filter sebagian persis) (Lihat Batasan)

Kueri menggunakan indeks paral

Pola kueri berikut dimungkinkan menggunakan indeks sebagian:

- Predikat kueri sama persis dengan ekspresi filter indeks sebagian:

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}}
  ]}).explain()
```

- Hasil yang diharapkan filter kueri adalah subset logis dari filter paral:

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}},
    {"OrderAmount": {"$eq": "5"}}
  ]}).explain()
```

- Sub-predikat kueri dapat digunakan bersama dengan indeks lain:

```
db.orders.createIndex({"anotherIndex":1})
db.orders.find({ "$or": [
      {"$and": [
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]},
      {"anotherIndex": {"$eq": 5}}
    ]
  }).explain()
```

> **Note**
>
> Perencana kueri dapat memilih untuk menggunakan pemindaian koleksi daripada pemindaian indeks jika efisien untuk melakukannya. Ini biasanya terlihat untuk koleksi atau kueri yang sangat kecil yang akan mengembalikan sebagian besar koleksi.

Fungsionalitas indeks sebagian

Daftar indeks sebagian

Buat daftar indeks sebagian dengan partialFilterExpression menggunakan `getIndex` operasi. Misalnya, `getIndex` operasi yang dikeluarkan dalam mencantumkan indeks sebagian dengan bidang kunci, nama, dan partialFilterExpressions:

```
db.orders.getIndexes()
```

Contoh ini mengembalikan output sebagai berikut:

```
[
  {
    "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}}
      ]
    }
  }
]
```

Beberapa ekspresi filter paral pada kunci yang sama: urutan

Indeks paral yang berbeda dapat dibuat untuk kombinasi bidang yang sama (key:order). Indeks ini harus memiliki nama yang berbeda.

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"firstPartialIndex",
    partialFilterExpression:{"OrderId":{"$exists": true}}
  }
)
```

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"secondPartialIndex",
    partialFilterExpression:{"OrderId":{"$gt": 1000}}
  }
)
```

Jalankan `getIndexes` operasi untuk mencantumkan semua indeks dalam koleksi:

```
db.orders.getIndexes()
```

Contoh-contoh ini mengembalikan output berikut:

```
[
  {
    "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**
>
> Nama indeks harus berbeda dan harus dihapus dengan nama saja.

Indeks dengan properti parsi dan TTL

Anda juga dapat membuat indeks yang memiliki properti parsi dan TTL dengan menentukan keduanya `partialFilterExpression` dan `expireAfterSeconds` opsi selama pembuatan indeks. Ini memungkinkan Anda untuk memberikan kontrol lebih besar atas dokumen mana yang sekarang dihapus dari koleksi.

Misalnya, Anda mungkin memiliki indeks TTL yang mengidentifikasi dokumen yang akan dihapus setelah jangka waktu tertentu. Anda sekarang dapat memberikan ketentuan tambahan tentang kapan harus menghapus dokumen menggunakan opsi indeks sebagian:

```
db.orders.createIndex(
    { "OrderTimestamp": 1 },
    {
        expireAfterSeconds: 3600 ,
        partialFilterExpression: { "isDelivered": { $eq: true } }
    }
)
```

Contoh ini mengembalikan output sebagai berikut:

```
{
        "createdCollectionAutomatically" : false,
```

```
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1,
        "operationTime" : Timestamp(1234567890, 1)
}
```

Jalankan `getIndexes` operasi untuk mencantumkan indeks yang ada dalam koleksi:

```
db.orders.getIndexes()
[
    {
        "v" : 4,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.orders"
    }
```

Contoh ini mengembalikan output sebagai berikut:

```
[
    {
        "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
```

```
    }
]
```

Keterbatasan indeks sebagian

Batasan berikut berlaku untuk fitur indeks sebagian:

- Kueri ketidaksetaraan di Amazon DocumentDB hanya akan menggunakan indeks sebagian ketika predikat filter kueri sama persis dengan dan memiliki tipe data `partialFilterExpression` yang sama.

  > **Note**
  >
  > Bahkan `$hint` tidak dapat digunakan untuk memaksa IXSCAN untuk kasus di atas.

  Dalam contoh berikut, hanya `partialFilterExpression` diterapkan pada `field1` tetapi tidak`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` dengan operator array tidak didukung. Operasi berikut akan menghasilkan kesalahan:

  ```
  db.orders.createIndex(
    {"CustomerId":1},
    {'partialFilterExpression': {'OrderId': {'$eq': [1000, 1001, 1002]}}}
  )
  ```

- Operator berikut tidak didukung di partialFilterExpression lapangan:
  - `$all`(operator array)
  - $mod(operator array)

- `$or`
- `$xor`
- `$not`
- `$nor`

- Tipe data dari ekspresi filter dan filter harus sama.

Melakukan pencarian teks dengan Amazon DocumentDB

Fitur pencarian teks lengkap asli Amazon DocumentDB memungkinkan Anda melakukan pencarian teks pada kumpulan data tekstual besar menggunakan indeks teks tujuan khusus. Bagian ini menjelaskan fungsionalitas fitur indeks teks dan memberikan langkah-langkah tentang cara membuat dan menggunakan indeks teks di Amazon DocumentDB. Batasan pencarian teks juga terdaftar.

Topik

- Fungsionalitas yang didukung
- Menggunakan indeks teks Amazon DocumentDB
- Perbedaan dengan MongoDB
- Praktik dan pedoman terbaik
- Batasan

Fungsionalitas yang didukung

Pencarian teks Amazon DocumentDB mendukung fungsionalitas yang kompatibel dengan MongoDB API berikut:

- Buat indeks teks pada satu bidang.
- Buat indeks teks majemuk yang menyertakan lebih dari satu bidang teks.
- Lakukan pencarian kata tunggal atau multi-kata.
- Kontrol hasil pencarian menggunakan bobot.
- Urutkan hasil pencarian berdasarkan skor.
- Gunakan indeks teks dalam pipa agregasi.
- Cari frase yang tepat.

Menggunakan indeks teks Amazon DocumentDB

Untuk membuat indeks teks pada bidang yang berisi data string, tentukan string “teks” seperti yang ditunjukkan di bawah ini:

Indeks bidang tunggal:

```
db.test.createIndex({"comments": "text"})
```

Indeks ini mendukung kueri pencarian teks di bidang string “komentar” dalam koleksi yang ditentukan.

Buat indeks teks majemuk pada lebih dari satu bidang string:

```
db.test.createIndex({"comments": "text", "title":"text"})
```

Indeks ini mendukung kueri pencarian teks di bidang string “komentar” dan “judul” dalam koleksi yang ditentukan. Anda dapat menentukan hingga 30 bidang saat membuat indeks teks majemuk. Setelah dibuat, kueri pencarian teks Anda akan menanyakan semua bidang yang diindeks.

> **Note**
>
> Hanya satu indeks teks yang diizinkan pada setiap koleksi.

Daftar indeks teks pada koleksi Amazon DocumentDB

Anda dapat menggunakan `getIndexes()` pada koleksi Anda untuk mengidentifikasi dan mendeskripsikan indeks, termasuk indeks teks, seperti yang ditunjukkan pada contoh di bawah ini:

```
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
   }
]
```

Setelah Anda membuat indeks, mulailah memasukkan data ke dalam koleksi Amazon DocumentDB Anda.

```
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"}])
```

Menjalankan kueri pencarian teks

Jalankan kueri pencarian teks satu kata

Anda perlu menggunakan `$text` dan `$search` operator untuk melakukan pencarian teks. Contoh berikut mengembalikan semua dokumen di mana bidang teks diindeks berisi string “apel” atau “apel” dalam format lain seperti “apel”:

```
db.test.find({$text: {$search: "apple"}})
```

Output:

Output dari perintah ini terlihat seperti ini:

```
{ "_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" }
```

Jalankan pencarian teks multi-kata

Anda juga dapat melakukan pencarian teks multi-kata pada data Amazon DocumentDB Anda. Perintah di bawah ini mengembalikan dokumen dengan bidang teks yang diindeks yang berisi “apple” atau “pie”:

```
db.test.find({$text: {$search: "apple pie"}})
```

Output:

Output dari perintah ini terlihat seperti ini:

```
{ "_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" }
```

Jalankan pencarian teks frase multi-kata

Untuk pencarian frase multi-kata, gunakan contoh ini:

```
db.test.find({$text: {$search: "\"apple pie\""}})
```

Output:

Perintah di atas mengembalikan dokumen dengan bidang teks diindeks yang berisi frase yang tepat “apple pie”. Output dari perintah ini terlihat seperti ini:

```
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Jalankan pencarian teks dengan filter

Anda juga dapat menggabungkan pencarian teks dengan operator kueri lain untuk memfilter hasil berdasarkan kriteria tambahan:

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "interest"}}]})
```

Output:

Perintah di atas mengembalikan dokumen dengan bidang teks yang diindeks yang berisi segala bentuk “minat” dan “star_rating” sama dengan 5. Output dari perintah ini terlihat seperti ini:

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
{ "_id" : 6, "star_rating" : 5, "comments" : "interested in couch for sale, year
 2022" }
```

Batasi jumlah dokumen yang dikembalikan dalam pencarian teks

Anda dapat memilih untuk membatasi jumlah dokumen yang dikembalikan dengan menggunakan`limit`:

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "couch"}}]}).limit(1)
```

Output:

Perintah di atas mengembalikan satu hasil yang memenuhi filter:

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
```

Urutkan hasil berdasarkan skor teks

Contoh berikut mengurutkan hasil pencarian teks berdasarkan skor teks:

```
db.test.find({$text: {$search: "apple"}}, {score: {$meta: "textScore"}}).sort({score:
 {$meta: "textScore"}})
```

Output:

Perintah di atas mengembalikan dokumen dengan bidang yang diindeks teks yang berisi “apel”, atau “apel” dalam format lain seperti “apel”, dan mengurutkan hasilnya berdasarkan seberapa relevan dokumen tersebut terkait dengan istilah pencarian. Output dari perintah ini terlihat seperti ini:

```
{ "_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 }
```

`$text`dan `$search` juga didukung untuk`aggregate`,`count`,`findAndModify`,`update`, dan `delete` perintah.

Operator agregasi

Pipa agregasi menggunakan **`$match`**

```
db.test.aggregate(
   [{ $match: { $text: { $search: "apple pie" } } }]
)
```

Output:

Perintah di atas mengembalikan hasil sebagai berikut:

```
{ "_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" }
```

Kombinasi dari operator agregasi lainnya

```
db.test.aggregate(
   [
      { $match: { $text: { $search: "apple pie" } } },
      { $sort: { score: { $meta: "textScore" } } },
      { $project: { score: { $meta: "textScore" } } }
   ]
)
```

Output:

Perintah di atas mengembalikan hasil sebagai berikut:

```
{ "_id" : 4, "score" : 0.6079270860936958 }
```

```
{ "_id" : 1, "score" : 0.3039635430468479 }
{ "_id" : 2, "score" : 0.3039635430468479 }
{ "_id" : 3, "score" : 0.3039635430468479 }
```

Tentukan beberapa bidang saat membuat indeks teks

Anda dapat menetapkan bobot hingga tiga bidang dalam indeks teks majemuk Anda. Bobot default yang ditetapkan ke bidang dalam indeks teks adalah satu (1). Berat adalah parameter opsional dan harus dalam kisaran 1 hingga 100000.

```
db.test.createIndex(
   {
     "firstname": "text",
     "lastname": "text",
     ...
   },
   {
     weights: {
       "firstname": 5,
       "lastname":10,
       ...
     },
     name: "name_text_index"
   }
 )
```

Perbedaan dengan MongoDB

Fitur indeks teks Amazon DocumentDB menggunakan indeks terbalik dengan algoritma frekuensi istilah. Indeks teks jarang secara default. Karena perbedaan dalam logika parsing, pembatas tokenisasi, dan lainnya, set hasil yang sama seperti MongoDB mungkin tidak dikembalikan untuk kumpulan data atau bentuk kueri yang sama.

Perbedaan tambahan berikut antara indeks teks Amazon DocumentDB dan MongoDB ada:

- Indeks majemuk menggunakan indeks non-teks tidak didukung.
- Indeks teks Amazon DocumentDB tidak peka huruf besar/kecil dan tidak sensitif terhadap diakritik.
- Hanya bahasa Inggris yang didukung dengan indeks teks.
- Pengindeksan teks bidang array (atau multi-kunci) tidak didukung. Misalnya, membuat indeks teks pada “a “dengan dokumen {“a”: [“apple”, “pie”]} akan gagal.

- Pengindeksan teks wildcard tidak didukung.
- Indeks teks unik tidak didukung.
- Tidak termasuk istilah tidak didukung.

Praktik dan pedoman terbaik

- Untuk kinerja optimal pada kueri penelusuran teks yang melibatkan pengurutan berdasarkan skor teks, sebaiknya Anda membuat indeks teks sebelum memuat data.
- Indeks teks memerlukan penyimpanan tambahan untuk salinan internal yang dioptimalkan dari data yang diindeks. Ini memiliki implikasi biaya tambahan.

Batasan

Pencarian teks memiliki batasan berikut di Amazon DocumentDB:

- Pencarian teks hanya didukung di klaster berbasis instans Amazon DocumentDB 5.0.

Pemecahan Masalah Amazon DocumentDB

Bagian berikut ini menyediakan informasi tentang cara untuk memecahkan masalah yang mungkin Anda alami ketika menggunakan Amazon DocumentDB (dengan kompatibilitas MongoDB).

Topik

- Masalah koneksi
- Pembuatan indeks
- Kinerja dan pemanfaatan sumber daya

Masalah koneksi

Mengalami masalah saat menghubungkan? Berikut ini adalah beberapa skenario umum dan cara untuk mengatasinya.

Topik

- Tidak dapat terhubung ke titik akhir Amazon DocumentDB
- Menguji koneksi ke instans Amazon DocumentDB
- Menghubungkan ke titik akhir yang tidak valid
- Konfigurasi driver yang memengaruhi jumlah koneksi

Tidak dapat terhubung ke titik akhir Amazon DocumentDB

Ketika Anda mencoba untuk menyambungkan ke Amazon DocumentDB, berikut ini adalah salah satu pesan kesalahan yang paling umum yang mungkin Anda terima.

```
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
```

Arti pesan kesalahan ini biasanya adalah bahwa klien Anda (mongo shell dalam contoh ini) tidak dapat mengakses titik akhir Amazon DocumentDB. Hal ini dapat terjadi karena beberapa alasan:

Topik

- Menghubungkan dari titik akhir publik
- Koneksi lintas wilayah
- Menghubungkan dari berbagai VPC Amazon
- Grup keamanan memblokir koneksi masuk
- Driver Java Mongo membaca masalah preferensi

Menghubungkan dari titik akhir publik

Anda mencoba terhubung ke cluster Amazon DocumentDB langsung dari laptop atau mesin pengembangan lokal Anda.

Mencoba untuk menghubungkan ke klaster Amazon DocumentDB langsung dari titik akhir publik, seperti laptop Anda atau mesin pengembangan lokal, akan gagal. Amazon DocumentDB adalah virtual private cloud (VPC)-satu-satunya dan saat ini tidak mendukung titik akhir publik. Dengan demikian, Anda tidak dapat menghubugkan secara langsung ke klaster Amazon DocumentDB Anda dari laptop Anda atau lingkungan pengembangan lokal di luar VPC Anda.

Untuk menghubungkan ke klaster Amazon DocumentDB dari luar Amazon VPC, Anda dapat menggunakan terowongan SSH. Untuk informasi selengkapnya, lihat Menghubungkan ke Klaster Amazon DocumentDB dari Luar Amazon VPC. Selain itu, jika lingkungan pengembangan Anda adalah dalam Amazon VPC yang berbeda, Anda juga dapat menggunakan Peering VPC dan menghubungkan ke klaster Amazon DocumentDB Anda dari Amazon VPC lainnya di wilayah yang sama atau wilayah yang berbeda.

Koneksi lintas wilayah

Anda mencoba terhubung ke cluster Amazon DocumentDB di wilayah lain.

Jika Anda mencoba untuk menghubungkan ke klaster Amazon DocumentDB dari instans Amazon EC2 di Wilayah selain Wilayah klaster ini—sebagai contoh, mencoba untuk menghubungkan ke

klaster di Wilayah US East (N. Virginia) (us-east-1) dari Wilayah US West (Oregon) (us-west-2)—koneksi akan gagal.

Untuk memverifikasi Wilayah klaster Amazon DocumentDB Anda, jalankan perintah berikut ini. Wilayah ini berada di titik akhir.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].Endpoint'
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
[
    "sample-cluster.node.us-east-1.docdb.amazonaws.com"
]
```

Untuk memverifikasi Wilayah instans EC2 Anda, jalankan perintah berikut ini.

```
 aws ec2 describe-instances \
     --query 'Reservations[*].Instances[*].Placement.AvailabilityZone'
```

Keluaran dari operasi ini terlihat seperti berikut ini.

```
[
    [
        "us-east-1a"
    ]
]
```

Menghubungkan dari berbagai VPC Amazon

Anda mencoba terhubung ke cluster Amazon DocumentDB dari VPC yang berbeda dari VPC Amazon yang digunakan cluster Anda.

Jika cluster Amazon DocumentDB dan instans Amazon EC2 Anda sama, tetapi tidak di VPC Amazon Wilayah AWS yang sama, Anda tidak dapat terhubung langsung ke cluster Amazon DocumentDB Anda kecuali VPC Peering diaktifkan di antara dua VPC Amazon.

Untuk memverifikasi Amazon VPC dari instans Amazon DocumentDB Anda, jalankan perintah berikut ini.

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

Untuk memverifikasi Amazon VPC dari instans Amazon EC2 Anda, jalankan perintah berikut ini.

```
aws ec2 describe-instances \
   --query 'Reservations[*].Instances[*].VpcId'
```

Grup keamanan memblokir koneksi masuk

Anda mencoba terhubung ke klaster Amazon DocumentDB, dan grup keamanan klaster tidak mengizinkan koneksi masuk pada port cluster (port default: 27017).

Anggap bahwa klaster Amazon DocumentDB Anda dan instans Amazon EC2 keduanya di Wilayah dan Amazon VPC yang sama dan menggunakan grup keamanan Amazon VPC yang sama. Jika Anda tidak dapat menghubungkan ke klaster Amazon DocumentDB Anda, kemungkinan penyebabnya adalah bahwa grup keamanan Anda (yaitu, firewall) untuk klaster Anda tidak mengizinkan koneksi inbound pada port yang Anda pilih untuk klaster Amazon DocumentDB Anda (port default adalah 27017).

Untuk memverifikasi port untuk klaster Amazon DocumentDB Anda , jalankan perintah berikut ini.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port]'
```

Untuk mendapatkan grup keamanan Amazon DocumentDB Anda untuk klaster Anda, jalankan perintah berikut ini.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[VpcSecurityGroups[*],VpcSecurityGroupId]'
```

Untuk memeriksa aturan inbound untuk grup keamanan Anda, lihat topik berikut ini dalam dokumentasi Amazon EC2:

- Otorisasi Lalu Lintas Masuk untuk Instans Linux Anda

- Mengotorisasi Lalu Lintas Masuk untuk Instans Windows Anda

Driver Java Mongo membaca masalah preferensi

Preferensi baca klien tidak dihormati dan beberapa klien tidak dapat menulis ke Amazon DocumentDB setelah failover kecuali mereka reboot.

Masalah ini, pertama kali ditemukan di Java Mongo Driver 3.7.x, terjadi ketika klien membuat koneksi ke Amazon DocumentDB menggunakan `MongoClientSettings` dan, khususnya, saat merantai metode. `applyToClusterSettings` Pengaturan MongoClient Cluster dapat didefinisikan menggunakan beberapa metode yang berbeda, sepertihosts(),requiredReplicaSetName(), danmode().

Ketika klien hanya menentukan satu host dalam `hosts()` metode, mode diatur ke `ClusterConnectionMode.SINGLE` bukan `ClusterConnectionMode.MULTIPLE` Ini menyebabkan klien mengabaikan preferensi baca dan hanya terhubung ke server yang dikonfigurasi. `hosts()` Jadi, bahkan jika pengaturan klien diinisialisasi seperti di bawah ini, semua pembacaan akan tetap pergi ke primer, bukan yang sekunder.

```
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);
```

Kasus failover

Menggunakan pengaturan koneksi klien di atas, jika ada failover dan pembaruan catatan DNS yang tertunda untuk titik akhir penulis cluster, klien masih akan mencoba mengeluarkan penulisan ke penulis lama (sekarang pembaca setelah failover). Ini menghasilkan kesalahan sisi server (bukan

master) yang tidak ditangani dengan tepat oleh driver Java (ini masih dalam penyelidikan). Dengan demikian, klien dapat dibiarkan dalam keadaan buruk sampai server aplikasi di-boot ulang, misalnya.

Ada dua solusi untuk ini:

- Klien yang terhubung ke Amazon DocumentDB melalui string koneksi tidak akan mengalami masalah ini, `ClusterConnectionMode` karena akan disetel `MULTIPLE` ke saat menyetel preferensi baca.

  ```
  MongoClientURI mongoClientURI = new MongoClientURI("mongodb://usr:pass:cluster-
  endpoint:27317/test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred");
  MongoClient mongoClient = MongoClients.create(mongoClientURI.getURI());
  ```

  Atau menggunakan `MongoClientSettings` pembangun dengan `applyConnectionString` metode ini.

  ```
  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);
  ```

- Secara eksplisit diatur `ClusterConnectionMode` ke. `MULTIPLE` Ini hanya diperlukan saat menggunakan `applyToClusterSettings` dan`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);
```

Menguji koneksi ke instans Amazon DocumentDB

Anda dapat menguji koneksi ke klaster menggunakan alat Linux atau Windows umum.

Dari terminal Linux atau Unix, Anda dapat menguji koneksi dengan memasukkan hal berikut ini (ganti `cluster-endpoint` dengan titik akhir, dan ganti `port` dengan port instans Anda):

```
nc -zv cluster-endpoint port
```

Berikut ini adalah contoh operasi sampel dan nilai balik:

```
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!
```

Menghubungkan ke titik akhir yang tidak valid

Ketika menghubungkan ke klaster Amazon DocumentDB dan Anda menggunakan titik akhir klaster yang tidak valid, muncul kesalahan yang mirip dengan berikut ini.

```
mongo --ssl \
   --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 \
   --sslCAFile global-bundle.pem \
   --username <user-name> \
   --password <password>
```

Keluaran terlihat seperti ini:

```
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
```

Untuk mendapatkan titik akhir yang valid untuk klaster, jalankan perintah berikut ini:

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[Endpoint,Port]'
```

Untuk mendapatkan titik akhir yang valid untuk instans, jalankan perintah berikut ini:

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].[Endpoint.Address,Endpoint.Port]'
```

Untuk informasi selengkapnya, lihat Memahami Titik Akhir Amazon DocumentDB.

Konfigurasi driver yang memengaruhi jumlah koneksi

Saat menggunakan driver klien untuk terhubung ke cluster Amazon DocumentDB, penting untuk mempertimbangkan `maxPoolSize` parameter konfigurasi. `maxPoolSize`Pengaturan menentukan jumlah maksimum koneksi yang akan dipertahankan oleh driver klien di kumpulan koneksinya.

Pembuatan indeks

Topik berikut ini membahas apa yang harus dilakukan jika indeks atau pembuatan indeks latar belakang gagal.

Topik

- Pembuatan indeks gagal
- Indeks latar belakang membangun masalah latensi dan gagal

Pembuatan indeks gagal

Amazon DocumentDB menggunakan penyimpanan lokal pada instans sebagai bagian dari proses pembuatan indeks. Anda dapat memantau penggunaan disk ini menggunakan CloudWatch metrik

FreeLocalStorage (`CloudWatch -> Metrics -> DocDB -> Instance Metrics`). Ketika pembuatan indeks mengonsumsi semua disk lokal dan gagal, Anda akan menerima kesalahan. Ketika memigrasi data ke Amazon DocumentDB, kami mendorong Anda untuk membuat indeks terlebih dahulu dan kemudian memasukkan data. Untuk informasi selengkapnya tentang strategi migrasi dan membuat indeks, lihat Migrasi ke Amazon DocumentDB dalam dokumentasi Amazon DocumentDB dan blog: Bermigrasi dari MongoDB ke Amazon DocumentDB menggunakan metode offline.

Ketika membuat indeks pada klaster yang ada, jika pembuatan indeks membutuhkan waktu lebih lama dari yang diharapkan atau gagal, kami sarankan Anda menskalakan ke atas instans untuk membuat indeks kemudian, setelah indeks dibuat, menskalakan kembali ke bawah. Amazon DocumentDB memungkinkan Anda menskalakan ukuran instans dengan cepat dalam hitungan menit menggunakan AWS Management Console atau file. AWS CLI Untuk informasi selengkapnya, lihat Mengelola Kelas Instans. Dengan harga instans per detik, Anda hanya membayar untuk sumber daya yang Anda gunakan hingga hitungan detik.

Indeks latar belakang membangun masalah latensi dan gagal

Indeks latar belakang yang dibangun di Amazon DocumentDB tidak mulai hingga semua kueri pada instans primer yang dimulai sebelum pembuatan indeks dijalankan secara lengkap saat dimulai. Jika terdapat kueri yang berjalan lama, pembuatan indeks latar belakang akan memblokir hingga kueri selesai dan dengan demikian dapat membutuhkan waktu lebih lama dari yang diharapkan untuk menyelesaikan. Hal ini berlaku bahkan jika koleksi kosong.

Pembuatan indeks latar depan tidak menunjukkan perilaku pemblokiran yang sama. Sebaliknya, pembuatan indeks latar depan mengambil kunci eksklusif pada koleksi hingga pembuatan indeks selesai. Dengan demikian, untuk membuat indeks pada koleksi kosong dan untuk menghindari pemblokiran pada setiap kueri yang berjalan lama, kami sarankan menggunakan pembuatan indeks latar depan.

> **Note**
>
> Amazon DocumentDB hanya mengizinkan satu pembuatan indeks latar belakang terjadi pada koleksi pada waktu tertentu. Jika operasi DDL (Data Definition Language) seperti `createIndex()` atau `dropIndex()` terjadi pada koleksi yang sama selama pembuatan indeks latar belakang, pembuatan indeks latar belakang akan gagal.

Kinerja dan pemanfaatan sumber daya

Bagian ini menyediakan pertanyaan dan solusi untuk masalah diagnostik umum dalam deployment Amazon DocumentDB. Contoh disediakan menggunakan mongo shell dan dicakupkan ke instans individu. Untuk menemukan titik akhir instans, lihat Memahami Titik Akhir Amazon DocumentDB.

Topik

- Bagaimana cara menentukan jumlah operasi penyisipan, pembaruan, dan penghapusan yang dilakukan pada koleksi saya melalui Mongo API?
- Bagaimana cara menganalisis kinerja cache?
- Bagaimana cara menemukan dan menghentikan kueri yang berjalan lama atau diblokir?
- Bagaimana saya bisa melihat rencana kueri dan mengoptimalkan kueri?
- Bagaimana saya bisa melihat rencana kueri dalam cluster elastis?
- Bagaimana cara saya mencantumkan semua operasi yang berjalan pada sebuah instance?
- Bagaimana saya tahu kapan kueri membuat kemajuan?
- Bagaimana cara menentukan mengapa sistem tiba-tiba berjalan lambat?
- Bagaimana cara menentukan penyebab pemanfaatan CPU yang tinggi pada satu atau lebih instance cluster?
- Bagaimana cara menentukan kursor terbuka pada sebuah instance?
- Bagaimana cara menentukan versi mesin Amazon DocumentDB saat ini?
- Bagaimana cara menganalisis penggunaan indeks dan mengidentifikasi indeks yang tidak digunakan?
- Bagaimana cara mengidentifikasi indeks yang hilang?
- Ringkasan pertanyaan yang berguna

Bagaimana cara menentukan jumlah operasi penyisipan, pembaruan, dan penghapusan yang dilakukan pada koleksi saya melalui Mongo API?

Untuk melihat jumlah operasi penyisipan, pembaruan, dan penghapusan yang dilakukan pada koleksi tertentu, jalankan perintah berikut pada koleksi tersebut:

```
db.collection.stats()
```

Output dari perintah ini menjelaskan hal berikut di bawah `opCounters` bidangnya:

- numDocsIns- Jumlah dokumen yang dimasukkan ke dalam koleksi ini. Ini termasuk dokumen yang dimasukkan menggunakan `insertMany` perintah `insert` dan, serta dokumen yang dimasukkan oleh upsert.
- numDocsUpd- Jumlah pembaruan dokumen dalam koleksi ini. Ini termasuk dokumen yang diperbarui menggunakan `findAndModify` perintah `update` dan.
- numDocsDel- Jumlah dokumen yang dihapus dari koleksi ini. Ini termasuk dokumen yang dihapus menggunakan`deleteOne`,`deleteMany`,`remove`, dan `findAndModify` perintah.
- LastReset - Waktu penghitung ini telah diatur ulang terakhir. Statistik yang disediakan oleh perintah ini diatur ulang saat memulai/menghentikan cluster atau menskalakan naik/turun instance.

Contoh output dari menjalankan `db.collection.stats()` ditunjukkan di bawah ini.

```
{
    "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)
}
```

Perintah statistik ini harus digunakan saat melihat penghitung khusus koleksi untuk menyisipkan, memperbarui, dan menghapus operasi melalui Mongo API. Cara lain untuk melihat penghitung operasi khusus koleksi adalah dengan mengaktifkan audit DHTML. Jumlah operasi penyisipan, perbarui, dan hapus pada semua koleksi selama interval waktu satu menit dapat dilihat diMemantau Amazon DocumentDB dengan CloudWatch.

Bagaimana cara menganalisis kinerja cache?

Menganalisis kinerja cache dapat memberikan wawasan tentang efisiensi pengambilan data dan kinerja sistem, dan didasarkan pada berapa banyak data yang dibaca dari disk versus cache. Kami menyediakan statistik cache tentang jumlah klik cache (data yang dibaca dari cache) dan cache yang salah (data yang tidak ditemukan di cache dan dibaca dari disk) untuk memberikan wawasan tentang kinerja cache. Statistik cache untuk koleksi tertentu dapat ditemukan dengan menjalankan perintah berikut pada koleksi itu:

```
db.collection.stats()
```

Nilai-nilai di `cacheStats` bidang dalam output dari perintah ini menyediakan statistik cache untuk koleksi serta statistik cache total untuk indeks yang dibuat pada koleksi. Statistik ini tercantum di bawah ini:

- **`collBlksHit`**- Jumlah blok yang dibaca dari cache selama operasi pada koleksi ini.
- **`collBlksRead`**- Jumlah blok yang dibaca dari disk (cache meleset) selama operasi pada koleksi ini.
- **`collHitRatio`**- Rasio hit cache untuk koleksi ini (`100 * [collBlksHit / (collBlksHit + collBlksRead)]`).
- **`idxBlksHit`**- Jumlah blok yang dibaca dari cache untuk indeks apa pun yang dibuat pada koleksi ini.
- **`idxBlksRead`**- Jumlah blok yang dibaca dari disk (cache meleset) untuk indeks apa pun yang dibuat pada koleksi ini.

- **idxHitRatio**- Rasio hit cache untuk indeks yang dibuat pada koleksi ini (`100 * [idxBlksHit / (idxBlksHit + idxBlksRead)]`).
- **lastReset**- Waktu statistik ini telah diatur ulang terakhir. Statistik yang disediakan oleh `db.collection.stats()` diatur ulang saat memulai/menghentikan cluster atau menskalakan naik/turun instance.

Rincian `idxBlksRead` bidang `idxBlksHit` dan untuk setiap indeks juga dapat ditemukan menggunakan `indexStats` perintah. Statistik cache indeks tertentu dapat ditemukan dengan menjalankan perintah berikut:

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Untuk setiap indeks, statistik cache berikut dapat ditemukan di bawah `cacheStats` bidang:

- **blksHit**- Jumlah blok yang dibaca dari cache untuk indeks ini.
- **blksRead**- Jumlah blok yang dibaca dari disk untuk indeks ini.
- **blksHitRatio**- Rasio hit cache dibulatkan ke empat tempat desimal, dihitung dengan. `100 * [blksHit / (blksHit + blksRead)]`

Bagaimana cara menemukan dan menghentikan kueri yang berjalan lama atau diblokir?

Kueri pengguna dapat berjalan lambat karena rencana kueri suboptimal atau dapat diblokir akibat kontensi sumber daya.

Untuk menemukan kueri yang berjalan lama yang melambat akibat rencana kueri suboptimal, atau kueri yang diblokir akibat kontensi sumber daya, gunakan perintah `currentOp`. Anda dapat menyaring perintah untuk membantu mempersempit daftar kueri yang relevan untuk mengakhiri. Anda harus memiliki `opid` yang berkaitan dengan kueri yang berjalan lama untuk dapat mengakhiri kueri.

Kueri berikut ini menggunakan perintah `currentOp` untuk mendaftar semua kueri yang diblokir maupun yang berjalan selama lebih dari 10 detik.

```
db.adminCommand({
    aggregate: 1,
    pipeline: [
```

```
        {$currentOp: {}},
        {$match:
            {$or: [
                {secs_running: {$gt: 10}},
                {WaitState: {$exists: true}}]}},
        {$project: {_id:0, opid: 1, secs_running: 1}}],
    cursor: {}
});
```

Selanjutnya, Anda dapat mempersempit kueri untuk menemukan `opid` dari kueri yang berjalan selama lebih dari 10 detik dan mengakhirinya.

Untuk menemukan dan mengakhiri kueri yang berjalan selama lebih dari 10 detik

1. Temukan `opid` kueri.

```
db.adminCommand({
    aggregate: 1,
    pipeline: [
        {$currentOp: {}},
        {$match:
            {$or:
                [{secs_running: {$gt: 10}},
                 {WaitState: {$exists: true}}]}}],
    cursor: {}
});
```

   Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "opid" : 24646,
                "secs_running" : 12
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
```

```
}
```

2. Mengakhiri kueri menggunakan operasi `killOp`.

```
db.adminCommand({killOp: 1, op: 24646});
```

Bagaimana saya bisa melihat rencana kueri dan mengoptimalkan kueri?

Jika kueri berjalan lambat, itu bisa karena eksekusi kueri memerlukan pemindaian penuh koleksi untuk memilih dokumen yang relevan. Kadang-kadang membuat indeks yang sesuai memungkinkan kueri untuk berjalan lebih cepat. Untuk mendeteksi skenario ini dan memutuskan bidang untuk membuat indeks, gunakan perintah `explain`.

> **Note**
>
> Amazon DocumentDB mengemulasi MongoDB 3.6 API pada mesin basis data yang dibuat khusus yang memanfaatkan sistem penyimpanan yang didistribusikan, toleran terhadap kesalahan, dan penyembuhkan diri sendiri. Akibatnya, rencana kueri dan keluaran dari `explain()` mungkin berbeda antara Amazon DocumentDB dan MongoDB. Pelanggan yang ingin kontrol atas rencana kueri mereka dapat menggunakan operator `$hint` untuk memaksa pemilihan indeks yang diutamakan.

Jalankan kueri yang ingin Anda tingkatkan di bawah perintah `explain` sebagai berikut.

```
db.runCommand({explain: {<query document>}})
```

Berikut ini adalah operasi contoh.

```
db.runCommand({explain:{
    aggregate: "sample-document",
    pipeline: [{$match: {x: {$eq: 1}}}],
    cursor: {batchSize: 1}}
});
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "queryPlanner" : {
```

```
        "plannerVersion" : 1,
        "namespace" : "db.test",
        "winningPlan" : {
            "stage" : "COLLSCAN"
        }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
}
```

Keluaran sebelumnya menunjukkan bahwa tahap `$match` membutuhkan pemindaian seluruh koleksi dan memeriksa apakah bidang `"x"` di setiap dokumen adalah sama dengan 1. Jika ada banyak dokumen dalam koleksi, pemindaian koleksi (dan karena itu performa kueri keseluruhan) sangat lambat. Dengan demikian adanya `"COLLSCAN"` dalam keluaran dari perintah `explain` menunjukkan bahwa performa kueri dapat ditingkatkan dengan membuat indeks yang sesuai.

Dalam contoh ini, kueri memeriksa apakah bidang `"x"` sama dengan 1 di semua dokumen. Jadi embuat indeks di bidang `"x"` memungkinkan kueri untuk menghindari pemindaian koleksi lengkap dan menggunakan indeks untuk mengembalikan dokumen yang relevan dengan lebih cepat.

Setelah membuat indeks pada bidang `"x"`, keluaran `explain` adalah sebagai berikut.

```
{
    "queryPlanner" : {
         "plannerVersion" : 1,
         "namespace" : "db.test",
         "winningPlan" : {
             "stage" : "IXSCAN",
             "indexName" : "x_1",
             "direction" : "forward"
         }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
```

```
}
```

Membuat indeks pada bidang `"x"` mengaktifkan tahap `$match` untuk menggunakan pemindaian indeks untuk mengurangi jumlah dokumen diana predikat `"x = 1"` harus dievaluasi.

Untuk koleksi kecil, prosesor kueri Amazon DocumentDB dapat memilih untuk tidak menggunakan indeks jika keuntungan performa diabaikan.

Bagaimana saya bisa melihat rencana kueri dalam cluster elastis?

Untuk memeriksa rencana kueri dalam cluster elastis, gunakan `explain` perintah. Berikut ini adalah contoh `explain` operasi pada kueri find yang menargetkan koleksi sharded:

```
db.runCommand(
   {
     explain: { find: "cities", filter: {"name": "Seoul"}}
   }
)
```

> **Note**
>
> Amazon DocumentDB mengemulasi MongoDB pada mesin database yang dibuat khusus. Akibatnya, rencana kueri dan keluaran dari `explain()` mungkin berbeda antara Amazon DocumentDB dan MongoDB. Anda dapat mengontrol rencana kueri dengan menggunakan `$hint` operator untuk menerapkan pemilihan indeks pilihan.

Output dari operasi ini mungkin terlihat seperti berikut (format JSON):

```
{
  "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)
}
```

Output sebelumnya menunjukkan rencana kueri untuk `find` kueri pada cluster tiga-shard. Setiap pecahan memiliki beberapa partisi data yang dapat memiliki tahapan input yang berbeda. Dalam contoh ini, “COLLSCAN “(pemindaian koleksi) dijalankan pada semua partisi sebelum hasilnya digabungkan pada tahap” PARTITION_MERGE “dalam setiap pecahan. Hasil di seluruh pecahan kemudian digabungkan bersama pada tahap” SHARD_MERGE “sebelum dikirim kembali ke klien.

Bagaimana cara saya mencantumkan semua operasi yang berjalan pada sebuah instance?

Sebagai pengguna atau pengguna utama, Anda sering ingin mencantumkan semua operasi saat ini yang berjalan pada sebuah instance untuk tujuan diagnostik dan pemecahan masalah. (Untuk informasi tentang mengelola pengguna, lihat Mengelola Pengguna Amazon DocumentDB.)

Dengan mongo shell, Anda dapat menggunakan kueri berikut ini untuk mendaftar semua operasi yang berjalan pada instans Amazon DocumentDB.

```
db.adminCommand({currentOp: 1, $all: 1});
```

Kueri mengembalikan daftar lengkap dari semua kueri pengguna dan tugas sistem internal yang saat ini beroperasi pada instans.

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "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
}
```

Berikuti ini adalah nilai valid untuk bidang `"desc"`:

- **`INTERNAL`** — Tugas sistem internal seperti pembersihan kursor atau tugas pembersihan pengguna usang.
- **`TTLMonitor`** — Urutan monitor Waktu untuk Tayang (TTL). Statusnya yang sedang berjalan tercermin di bidang `"active"`.
- **`GARBAGE_COLLECTION`** — Urutan pengumpul sampah internal.
- **`CONN`** — Kueri pengguna.
- **`CURSOR`** — Operasi adalah kursor diam yang menunggu pengguna untuk memanggil perintah "getMore" untuk mendapatkan hasil batch berikutnya. Dalam keadaan ini, kursor mengonsumsi memori, tetapi tidak mengonsumsi komputasi apa pun.

Keluaran sebelumnya juga mendaftar semua kueri pengguna yang berjalan di sistem. Setiap kueri pengguna berjalan dalam konteks basis data dan koleksi, dan penyatuan keduanya disebut namespace. Namespace dari setiap kueri pengguna tersedia di bidang `"ns"`.

Kadang-kadang Anda perlu mendaftar semua kueri pengguna yang berjalan di namespace tertentu. Sehingga keluaran sebelumnya harus difilter pada bidang `"ns"`. Berikut ini adalah contoh kueri untuk mencapai keluaran untuk filter. Kueri mendaftar semua kueri pengguna yang saat ini berjalan di basis data `"db"` dan koleksi `"test"` (yaitu, namespace `"db.test"`).

```
db.adminCommand({aggregate: 1,
    pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
               {$match: {ns: {$eq: "db.test"}}}],
    cursor: {}
});
```

Sebagai pengguna utama sistem, Anda dapat melihat kueri semua pengguna dan juga semua tugas sistem internal. Semua pengguna lain hanya dapat melihat kueri mereka masing-masing.

Jika jumlah total kueri dan tugas sistem internal melebihi ukuran kursor batch default , mongo shell secara otomatis menghasilkan sebuah objek iterator `'it'` untuk melihat sisa hasil. Tetap jalankan perintah `'it'` hingga semua hasil telah dikeluarkan.

Bagaimana saya tahu kapan kueri membuat kemajuan?

Kueri pengguna dapat berjalan lambat akibat rencana kueri suboptimal, atau mereka dapat diblokir akibat kontensi sumber daya. Debugging kueri semacam itu adalah proses multi-langkah yang mungkin perlu menjalankan langkah yang sama beberapa kali.

Langkah pertama debugging adalah untuk mendaftar semua kueri yang berjalan lama atau diblokir. Kueri berikut iniendaftar semua kueri pengguna yang telah berjalan selama lebih dari 10 detik atau yang menunggu sumber daya.

```
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: {}
                });
```

Ulangi kueri sebelumnya secara berkala untuk menentukan apakah daftar kueri berubah dan untuk mengidentifikasi kueri yang berjalan lama atau diblokir.

Jika dokumen keluaran untuk kueri yang menjadi perhatian memiliki bidang `WaitState`, ini menunjukkan bahwa kontensi sumber daya adalah mengapa kueri berjalan lambat atau diblokir. Kontensi sumber daya dapat akibat I/O, tugas sistem internal, maupun kueri pengguna lainnya.

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
```

```
                "opid" : 201,
                "command" : {
                    "aggregate" : ...
                },
                "secs_running" : 208,
                "WaitState" : "IO"
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

I/O dapat menjadi hambatan jika banyak kueri di seluruh koleksi yang berbeda berjalan bersamaan pada instans yang sama, atau jika ukuran instans terlalu kecil untuk set data tempat kueri berjalan. Jika kueri adalah kueri hanya baca, Anda dapat memitigasi situasi sebelumnya dengan memisahkan kueri untuk setiap koleksi di seluruh replika terpisah. Untuk pembaruan bersamaan di koleksi yang berbeda, atau ketika ukuran instans terlalu kecil untuk set data, Anda dapat memitigasi dengan menskalakan ke atas instans.

Jika kontensi sumber daya adalah akibat kueri pengguna lain, bidang `"blockedOn"` di dokumen keluaran akan memiliki `"opid"` dari kueri yang memengaruhi kueri ini. Dengan menggunakan `"opid"` ikuti rantai bidang `"WaitState"` dan `"blockedOn"` dari semua kueri untuk menemukan kueri pada kepala rantai.

Jika tugas kepala rantai adalah tugas internal, satu-satunya mitigasi dalam kasus ini akan mengakhiri kueri dan nantinya akan menjalankan kembali.

Berikut ini adalah keluaran contoh di mana kueri penemuan diblokir pada kunci koleksi yang dimiliki oleh tugas lain.

```
{
    "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
}
```

Jika "`WaitState`" memiliki nilai "`Latch`", "`SystemLock`", "`BufferLock`", "`BackgroundActivity`", atau "`Other`", sumber kontensi sumber daya adalah tugas sistem internal. Jika situasi terus berlangsung selama waktu yang lama, hanya mitigasi yang akan mengakhiri kueri dan nantinya akan menjalankan kembali.

Bagaimana cara menentukan mengapa sistem tiba-tiba berjalan lambat?

Berikut ini adalah beberapa alasan umum mengapa sistem melambat:

- Kontensi sumber daya yang berlebihan di antara kueri yang bersamaan
- Jumlah kueri bersamaan yang aktif meningkat seiring waktu
- Tugas sistem internal seperti `"GARBAGE_COLLECTION"`

Untuk memantau penggunaan sistem dari waktu ke waktu, jalankan kueri `"currentOp"` berikut ini secara berkala dan keluarkan hasil ke penyimpanan eksternal. Kueri menghitung jumlah kueri dan operasi di setiap namespace dalam sistem. Anda kemudian dapat menganalisis hasil penggunaan sistem untuk memahami beban pada sistem dan mengambil tindakan yang tepat.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Kueri ini mengembalikan agregat dari semua kueri yang berjalan di setiap namespace, semua tugas sistem internal, dan jumlah unik keadaan menunggu (jika ada) per namespace.

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{
    "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
}
```

Dalam keluaran sebelumnya, dua kueri pengguna dalam `"db.test"` namespace diblokir pada kunci koleksi: satu kueri di `"admin.$cmd"` namespace, dan satu tugas `"TTLMonitor"` internal.

Jika keluaran menunjukkan banyak kueri dengan memblokir keadaan tunggu, lihat Bagaimana cara menemukan dan menghentikan kueri yang berjalan lama atau diblokir?

Bagaimana cara menentukan penyebab pemanfaatan CPU yang tinggi pada satu atau lebih instance cluster?

Bagian berikut ini dapat membantu Anda mengidentifikasi penyebab pemanfaatan CPU instans yang tinggi. Hasil Anda dapat bervariasi bergantung pada beban kerja.

- Untuk menentukan mengapa instans tiba-tiba berjalan lambat, lihat Bagaimana cara menentukan mengapa sistem tiba-tiba berjalan lambat?
- Untuk mengidentifikasi dan mengakhiri kueri yang berjalan lamaa pada instans tertentu, lihat Bagaimana cara menemukan dan menghentikan kueri yang berjalan lama atau diblokir?
- Untuk memahami apakah kueri sedang berjalan, lihat Bagaimana saya tahu kapan kueri membuat kemajuan?
- Untuk menentukan mengapa kueri membutuhkan waktu yang lama untuk berjalan, lihat Bagaimana saya bisa melihat rencana kueri dan mengoptimalkan kueri?
- Untuk melacak kueri yang berjalan lama dari waktu ke waktu, lihat Membuat Profil Operasi Amazon DocumentDB.

Bergantung pada alasan untuk pemanfaatan CPU instans Anda yang tinggi, dengan melakukan satu atau lebih hal berikut ini dapat membantu.

- Jika instans primer menunjukkan pemanfaatan CPU yang tinggi, tetapi instans replika tidak menunjukkannya, pertimbangkan untuk mendistribusikan lalu lintas baca di replika melalui pengaturan preferensi baca klien (sebagai contoh, `secondaryPreferred`). Untuk informasi selengkapnya, lihat Menghubungkan ke Amazon DocumentDB sebagai Set Replika.

  Dengan menggunakan replika untuk pembacaan dapat membuat penggunaan klaster sumber daya yang lebih baik dengan mengizinkan instans primer untuk memproses lebih banyak lalu lintas tulis. Pembacaan dari replika pada akhirnya adalah konsisten.
- Jika pemanfaatan CPU yang tinggi adalah hasil dari beban kerja tulis Anda, mengubah ukuran instans klaster untuk jenis instans yang lebih besar meningkatkan jumlah inti CPU yang tersedia untuk layanan beban kerja. Lihat informasi yang lebih lengkap di Instans dan Spesifikasi Kelas Instans.
- Jika semua instans klaster menunjukkan pemanfaatan CPU yang tinggi, dan beban kerja menggunakan replika untuk pembacaan, akan menambahkan lebih banyak replika klaster akan meningkatkan sumber daya yang tersedia untuk lalu lintas baca. Untuk informasi selengkapnya, lihat Menambahkan instance Amazon DocumentDB ke cluster.

Bagaimana cara menentukan kursor terbuka pada sebuah instance?

Ketika terhubung ke instans Amazon DocumentDB, Anda dapat menggunakan perintah `db.runCommand("listCursors")` untuk mendaftar kursor terbuka pada instans tersebut. Terdapat batas hingga 4.560 kursor aktif terbuka pada waktu tertentu pada instans Amazon DocumentDB tertentu, bergantung pada tipe instans. Umumnya disarankan untuk menutup kursor yang tidak lagi digunakan karena kursor memanfaatkan sumber daya pada instans dan memiliki batas atas. Lihat Kuota dan Batas Amazon DocumentDB untuk batas spesifik.

```
db.runCommand("listCursors")
```

Bagaimana cara menentukan versi mesin Amazon DocumentDB saat ini?

Untuk menentukan versi mesin Amazon DocumentDB Anda saat ini, jalankan perintah berikut ini.

```
db.runCommand({getEngineVersion: 1})
```

Keluaran dari operasi ini terlihat seperti berikut ini (format JSON).

```
{ "engineVersion" : "2.x.x", "ok" : 1 }
```

> **Note**
>
> Versi mesin untuk Amazon DocumentDB 3.6 adalah 1.x.x dan versi mesin untuk Amazon DocumentDB 4.0 adalah 2.x.x.

Bagaimana cara menganalisis penggunaan indeks dan mengidentifikasi indeks yang tidak digunakan?

Untuk mengidentifikasi indeks untuk koleksi tertentu, jalankan perintah berikut ini:

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

Untuk menganalisis berapa banyak indeks yang digunakan selama operasi yang dilakukan pada koleksi, `indexStats` perintah `collStats` dan dapat digunakan. Untuk melihat jumlah total pemindaian yang dilakukan menggunakan indeks (pemindaian indeks) dibandingkan dengan jumlah pemindaian yang dilakukan tanpa indeks (pemindaian koleksi), jalankan perintah berikut:

```
db.collection.stats()
```

Output untuk perintah ini mencakup nilai-nilai berikut:

- **`idxScans`**- Jumlah pemindaian yang dilakukan pada koleksi ini menggunakan indeks.
- **`collScans`**- Jumlah pemindaian yang dilakukan pada koleksi ini tanpa menggunakan indeks. Pemindaian ini akan melibatkan pemeriksaan dokumen dalam koleksi satu per satu.
- **`lastReset`**- Waktu penghitung ini telah diatur ulang terakhir. Statistik yang disediakan oleh perintah ini diatur ulang saat memulai/menghentikan cluster atau menskalakan naik/turun instance.

Rincian berapa banyak setiap indeks digunakan dapat ditemukan dalam output dari perintah berikut. Ini adalah praktik terbaik untuk secara teratur mengidentifikasi dan menghapus indeks yang tidak digunakan untuk meningkatkan kinerja dan mengurangi biaya, karena menghilangkan komputasi, penyimpanan, dan I/O yang tidak perlu yang digunakan untuk mempertahankan indeks.

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Output dari perintah ini memberikan nilai-nilai berikut untuk setiap indeks yang dibuat pada koleksi:

- **`ops`**- Jumlah operasi yang menggunakan indeks. Jika beban kerja Anda telah berjalan selama waktu yang cukup lama dan Anda yakin bahwa beban kerja Anda dalam keadaan stabil, nilai `ops` sebesar nol akan menunjukkan bahwa indeks tidak digunakan sama sekali.
- **`numDocsRead`**- Jumlah dokumen yang dibaca selama operasi menggunakan indeks ini.
- **`since`**- Waktu sejak Amazon DocumentDB mulai mengumpulkan statistik tentang penggunaan indeks, yang biasanya merupakan nilai sejak restart database terakhir atau tindakan pemeliharaan.
- **`size`**- Ukuran indeks ini dalam byte.

Contoh berikut adalah contoh output dari menjalankan perintah di atas:

```
{
    "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" : ...
    }
}
```

Untuk menentukan ukuran indeks keseluruhan untuk koleksi, jalankan perintah berikut ini:

```
db.collection.stats()
```

Untuk menghapus indeks yang tidak digunakan, jalankan perintah berikut ini:

```
db.collection.dropIndex("indexName")
```

Bagaimana cara mengidentifikasi indeks yang hilang?

Anda dapat menggunakan profiler Amazon DocumentDB untuk log kueri yang lambat. Kueri yang muncul berulang kali di log kueri lambat mungkin menunjukkan bahwa indeks tambahan diperlukan untuk meningkatkan performa kueri tersebut.

Anda dapat mengidentifikasi peluang untuk indeks yang bermanfaat dengan mencari kueri yang berjalan lama yang memiliki satu atau lebih tahapan yang melakukan setidaknya satu tahap `COLLSCAN`, yang berarti bahwa kueri tahap tersebut harus membaca setiap dokumen dalam koleksi untuk menyediakan respons terhadap kueri.

Contoh berikut ini menunjukkan kueri pada koleksi perjalanan taksi yang berjalan pada koleksi besar.

```
db.rides.count({"fare.totalAmount":{$gt:10.0}}))
```

Untuk menjalankan contoh ini, kueri harus melakukan pemindaian koleksi (yaitu membaca setiap dokumen tunggal dalam koleksi) karena tidak ada indeks pada bidang `fare.totalAmount`. Keluaran dari profiler Amazon DocumentDB untuk kueri ini terlihat seperti berikut ini:

```
{
    ...
```

```
    "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"
}
```

Untuk mempercepat kueri dalam contoh ini, Anda ingin membuat indeks pada `fare.totalAmount`, sebagaimana ditunjukkan di bawah ini.

```
db.rides.createIndex( {"fare.totalAmount": 1}, {background: true} )
```

> **Note**
>
> Indeks yang dibuat di latar depan (artinya jika opsi `{background:true}` tidak disediakan saat membuat indeks) ambil kunci tulis eksklusif, yang mencegah aplikasi agar tidak menulis data ke koleksi hingga pembuatan indeks selesai. Waspadalah dampak potensial ini saat membuat indeks pada klaster produksi. Saat membuat indeks, kami menyarankan pengaturan `{background:true}`.

Secara umum, Anda ingin membuat indeks pada bidang yang memiliki kardinalitas tinggi (sebagai contoh, sejumlah besar nilai unik). Membuat indeks di bidang dengan kardinalitas rendah dapat menghasilkan indeks besar yang tidak digunakan. Pengoptimal kueri Amazon DocumentDB menganggap ukuran keseluruhan koleksi dan selektivitas indeks saat membuat rencana kueri. Ada kalanya Anda akan melihat prosesor kueri memilih `COLLSCAN` bahkan ketika ada indeks. Hal ini terjadi ketika kueri prosesor memperkirakan bahwa pemanfaatan indeks tidak akan menghasilkan keuntungan performa pada saat memindai seluruh koleksi. Jika Anda ingin memaksa prosesor kueri untuk memanfaatkan indeks tertentu, Anda dapat menggunakan operator `hint()` sebagaimana ditunjukkan di bawah ini.

```
db.collection.find().hint("indexName")
```

Ringkasan pertanyaan yang berguna

Kueri berikut ini dapat berguna untuk memantau peforma dan pemanfaatan sumber daya di Amazon DocumentDB.

- Gunakan perintah berikut untuk melihat statistik tentang koleksi tertentu, termasuk penghitung operasi, statistik cache, statistik akses, dan statistik ukuran:

```
db.collection.stats()
```

- Gunakan perintah berikut untuk melihat statistik tentang setiap indeks yang dibuat pada koleksi termasuk ukuran indeks, statistik cache khusus indeks, dan statistik penggunaan indeks:

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

- Gunakan kueri berikut ini untuk mendaftar semua aktivitas.

```
db.adminCommand({currentOp: 1, $all: 1});
```

- Kode berikut ini mencantumkan semua kueri yang berjalan lama atau diblokir.

```
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: {}
                });
```

- Kode berikut ini mengakhiri kueri.

```
db.adminCommand({killOp: 1, op: <opid of running or blocked query>});
```

- Gunakan kode berikut ini untuk mendapatkan tampilan agregat dari keadaan sistem.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Referensi API Manajemen Klaster, Instans, dan Sumber Daya Amazon DocumentDB

Bagian ini menjelaskan operasi manajemen klaster, instans, dan sumber daya untuk Amazon DocumentDB (dengan kompatibilitas MongoDB) yang dapat diakses melalui HTTP, AWS Command Line Interface (AWS CLI), atau AWS SDK. Anda dapat menggunakan API ini untuk membuat, menghapus, serta memodifikasi klaster dan instans.

> **Important**
>
> API ini hanya digunakan untuk mengelola klaster, instans, dan sumber daya terkait. Untuk informasi tentang cara menghubungkan ke klaster Amazon DocumentDB yang sedang berjalan, lihat Panduan Memulai.

Topik

- Tindakan
- Tipe Data
- Kesalahan Umum
- Parameter Umum

Tindakan

Tindakan berikut didukung oleh 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
- ListTagsForResource
- ModifyDBCluster

- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster

Tindakan berikut didukung oleh Amazon DocumentDB Elastic Clusters:

- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster
- GetClusterSnapshot
- ListClusters
- ListClusterSnapshots
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster

- TagResource
- UntagResource
- UpdateCluster

Amazon DocumentDB (with MongoDB compatibility)

Tindakan berikut didukung olehAmazon DocumentDB (with MongoDB compatibility):

- AddSourceIdentifierToSubscription
- AddTagsToResource
- ApplyPendingMaintenanceAction
- CopyDBClusterParameterGroup
- CopyDBClusterSnapshot
- CreateDBCluster
- CreateDBClusterParameterGroup
- CreateDBClusterSnapshot
- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster
- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters
- DescribeDBClusterSnapshotAttributes

- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- ListTagsForResource
- ModifyDBCluster
- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster

AddSourceIdentifierToSubscription

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menambahkan pengidentifikasi sumber ke langganan notifikasi peristiwa yang ada.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SourceIdentifier

Pengidentifikasi sumber peristiwa yang akan ditambahkan:

- Jika jenis sumber adalah sebuah instans, `DBInstanceIdentifier` harus disediakan.
- Jika jenis sumber adalah sebuah grup keamanan, `DBSecurityGroupName` harus disediakan.
- Jika jenis sumber adalah sebuah grup parameter, `DBParameterGroupName` harus disediakan.
- Jika jenis sumber adalah sebuah snapshot, `DBSnapshotIdentifier` harus disediakan.

Tipe: String

Diperlukan: Ya

SubscriptionName

Nama langganan notifikasi peristiwa Amazon DocumentDB yang ingin Anda tambahkan pengidentifikasi sumbernya.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EventSubscription

Informasi mendetail tentang peristiwa yang telah Anda langgani.

Tipe: Objek EventSubscription

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

SourceNotFound

Sumber yang diminta tidak dapat ditemukan.

Kode Status HTTP: 404

SubscriptionNotFound

Nama langganan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

AddTagsToResource

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menambahkan tag metadata ke sumber daya Amazon DocumentDB. Anda dapat menggunakan tag ini dengan pelaporan alokasi biaya untuk melacak biaya yang terkait dengan sumber daya Amazon DocumentDB atau `Condition` dalam pernyataan AWS Identity and Access Management dalam kebijakan (IAM) untuk Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

ResourceName

Sumber daya Amazon DocumentDB tempat tag ditambahkan. Nilai ini adalah Nama Sumber Daya Amazon.

Tipe: String

Diperlukan: Ya

Tags.Tag.N

Tag yang akan ditetapkan ke sumber daya Amazon DocumentDB.

Tipe: Array objek Tag

Wajib: Ya

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` tidak mengacu ke sebuah snapshot yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ApplyPendingMaintenanceAction

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menerapkan tindakan pemeliharaan tertunda ke sumber daya (misalnya, ke instans Amazon DocumentDB).

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

ApplyAction

Tindakan pemeliharaan tertunda untuk diterapkan ke sumber daya ini.

Nilai valid: `system-update`, `db-upgrade`

Tipe: String

Diperlukan: Ya

OptInType

Nilai yang menentukan jenis permintaan opt-in atau membatalkan permintaan opt-in. Permintaan opt-in dari jenis `immediate` tidak dapat dibatalkan.

Nilai valid:

- `immediate` - Terapkan tindakan pemeliharaan segera.
- `next-maintenance` - Terapkan tindakan pemeliharaan selama jendela pemeliharaan berikutnya untuk sumber daya.
- `undo-opt-in` - Batalkan permintaan opt-in `next-maintenance` apa pun yang ada.

Tipe: String

Diperlukan: Ya

ResourceIdentifier

Amazon Resource Name (ARN) dari sumber daya tempat tindakan pemeliharaan tertunda diterapkan.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

ResourcePendingMaintenanceActions

Merupakan output dari ApplyPendingMaintenanceAction.

Tipe: Objek ResourcePendingMaintenanceActions

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

ResourceNotFoundFault

ID sumber daya yang ditentukan tidak ditemukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CopyDBClusterParameterGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Salinan grup parameter klaster tertentu.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SourceDBClusterParameterGroupIdentifier

Pengidentifikasi atau Amazon Resource Name (ARN) untuk grup parameter klaster sumber.

Batasan:

- Harus menentukan grup parameter klaster yang valid.
- Jika kelompok parameter cluster sumber Wilayah AWS sama dengan salinan, tentukan pengidentifikasi grup parameter yang valid; misalnya,`my-db-cluster-param-group`, atau ARN yang valid.
- Jika kelompok parameter sumber berbeda Wilayah AWS dari salinan, tentukan grup parameter cluster yang valid ARN; misalnya,. `arn:aws:rds:us-east-1:123456789012:sample-cluster:sample-parameter-group`

Tipe: String

Diperlukan: Ya

TargetDBClusterParameterGroupDescription

Deskripsi untuk grup parameter klaster yang disalin.

Tipe: String

Diperlukan: Ya

TargetDBClusterParameterGroupIdentifier

Pengidentifikasi untuk grup parameter klaster yang disalin.

Batasan:

- Tidak dapat berupa null, kosong, atau blank.
- Harus berisi 1 sampai 255 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.

- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-cluster-param-group1`

Tipe: String

Diperlukan: Ya

Tags.Tag.N

Tag yang akan ditetapkan ke grup parameter.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterParameterGroup

Informasi mendetail tentang grup parameter klaster.

Tipe: Objek DBClusterParameterGroup

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupAlreadyExists

Grup parameter dengan nama yang sama sudah ada.

Kode Status HTTP: 400

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

DBParameterGroupQuotaExceeded

Permintaan ini akan menyebabkan Anda melebihi jumlah grup parameter yang diizinkan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CopyDBClusterSnapshot

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Salinan sebuah snapshot dari sebuah klaster.

Untuk menyalin snapshot klaster dari snapshot klaster manual bersama, `SourceDBClusterSnapshotIdentifier` harus menjadi Amazon Resource Name (ARN) dari snapshot klaster bersama. Anda hanya dapat menyalin snapshot klaster DB bersama, baik dienkripsi atau tidak, di Wilayah AWS yang sama.

Untuk membatalkan operasi penyalinan setelah ia berlangsung, hapus snapshot klaster target yang diidentifikasi oleh `TargetDBClusterSnapshotIdentifier` saat snapshot klaster tersebut berada dalam status menyalin.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SourceDBClusterSnapshotIdentifier

Pengidentifikasi snapshot klaster yang akan disalin. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus menentukan snapshot sistem yang valid dalam status tersedia.
- Jika snapshot sumber Wilayah AWS sama dengan salinan, tentukan pengenal snapshot yang valid.
- Jika snapshot sumber berbeda Wilayah AWS dari salinan, tentukan ARN snapshot cluster yang valid.

Contoh: `my-cluster-snapshot1`

Tipe: String

Diperlukan: Ya

TargetDBClusterSnapshotIdentifier

Pengidentifikasi snapshot klaster baru yang akan dibuat dari snapshot klaster sumber. Parameter ini tidak peka huruf besar kecil.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-cluster-snapshot2`

Tipe: String

Diperlukan: Ya

CopyTags

Atur ke `true` untuk menyalin semua tag dari snapshot klaster sumber ke snapshot klaster target, dan sebaliknya `false`. Defaultnya adalah `false`.

Tipe: Boolean

Wajib: Tidak

KmsKeyId

ID AWS KMS kunci untuk snapshot cluster terenkripsi. ID AWS KMS kunci adalah Nama Sumber Daya Amazon (ARN), pengenal AWS KMS kunci, atau alias kunci untuk AWS KMS kunci enkripsi. AWS KMS

Jika Anda menyalin snapshot cluster terenkripsi dari Anda Akun AWS, Anda dapat menentukan nilai `KmsKeyId` untuk mengenkripsi salinan dengan kunci enkripsi baru. AWS KMS Jika Anda tidak menentukan nilai untuk`KmsKeyId`, maka salinan snapshot cluster dienkripsi dengan AWS KMS kunci yang sama dengan snapshot cluster sumber.

Jika Anda menyalin snapshot cluster terenkripsi yang dibagikan dari yang lain Akun AWS, maka Anda harus menentukan nilai untuk. `KmsKeyId`

Untuk menyalin snapshot kluster terenkripsi ke yang lain Wilayah AWS, atur `KmsKeyId` ke ID AWS KMS kunci yang ingin Anda gunakan untuk mengenkripsi salinan snapshot cluster di Wilayah tujuan. AWS KMS kunci enkripsi khusus untuk tempat Wilayah AWS mereka dibuat, dan Anda tidak dapat menggunakan kunci enkripsi dari satu Wilayah AWS sama lain Wilayah AWS.

Jika Anda menyalin snapshot klaster yang tidak terenkripsi dan menentukan nilai untuk parameter `KmsKeyId`, kesalahan akan dikembalikan.

Tipe: String

Wajib: Tidak

PreSignedUrl

URL yang berisi permintaan ditandatangani Signature Version 4 untuk tindakan `CopyDBClusterSnapshot` API Wilayah AWS yang berisi snapshot cluster sumber untuk disalin. Anda harus menggunakan parameter `PreSignedUrl` ketika menyalin snapshot klaster dari Wilayah AWS yang lain.

Jika Anda menggunakan alat AWS SDK atau AWS CLI, Anda dapat menentukan `SourceRegion` (atau `--source-region` untuk AWS CLI) alih-alih menentukan secara manual`PreSignedUrl`. Menentukan `SourceRegion` secara otomatis menghasilkan URL pre-signed yang merupakan permintaan yang valid untuk operasi yang dapat dijalankan di Wilayah AWS sumber.

URL yang telah ditetapkan sebelumnya harus berupa permintaan valid untuk tindakan `CopyDBClusterSnapshot` API yang dapat dijalankan di sumber Wilayah AWS yang berisi snapshot cluster yang akan disalin. Permintaan URL yang ditandatangani sebelumnya harus berisi nilai parameter berikut:

- `SourceRegion` - ID dari wilayah yang berisi snapshot yang akan disalin.
- `SourceDBClusterSnapshotIdentifier` - Pengidentifikasi untuk snapshot klaster terenkripsi yang akan disalin. Pengidentifikasi ini harus dalam format Amazon Resource Name (ARN) untuk Wilayah AWS sumber. Misalnya, jika Anda menyalin snapshot klaster terenkripsi dari Wilayah AWS us-east-1, maka `SourceDBClusterSnapshotIdentifier` Anda akan terlihat seperti berikut: `arn:aws:rds:us-east-1:12345678012:sample-cluster:sample-cluster-snapshot`.
- `TargetDBClusterSnapshotIdentifier` - Pengidentifikasi untuk snapshot klaster baru yang akan dibuat. Parameter ini tidak peka huruf besar kecil.

Tipe: String

Wajib: Tidak

Tags.Tag.N

Tag yang akan ditetapkan ke snapshot klaster.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterSnapshot

Informasi mendetail tentang snapshot klaster.

Tipe: Objek DBClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterSnapshotAlreadyExistsFault

Anda sudah memiliki snapshot klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

KMSKeyNotAccessibleFault

Terjadi kesalahan saat mengakses AWS KMS kunci.

Kode Status HTTP: 400

SnapshotQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah snapshot yang diizinkan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateDBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat klaster Amazon DocumentDB baru.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster. Parameter ini disimpan sebagai string huruf kecil.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-cluster`

Tipe: String

Diperlukan: Ya

Engine

Nama mesin basis data yang akan digunakan untuk klaster ini.

Nilai yang valid: `docdb`

Tipe: String

Diperlukan: Ya

AvailabilityZones. AvailabilityZone.N

Daftar Availability Zone Amazon EC2 tempat instans dalam klaster dapat dibuat.

Tipe: Array string

Wajib: Tidak

BackupRetentionPeriod

Jumlah hari penyimpanan cadangan otomatis. Anda harus menentukan nilai minimum 1.

Default: 1

Batas:

- Harus berupa nilai dari 1 hingga 35.

Tipe: Integer

Wajib: Tidak

DBClusterParameterGroupName

Nama grup parameter klaster yang akan dikaitkan dengan klaster ini.

Tipe: String

Wajib: Tidak

DBSubnetGroupName

Grup subnet yang akan dikaitkan dengan klaster ini.

Batas: Harus cocok dengan nama `DBSubnetGroup` yang ada. Tidak harus default.

Contoh: `mySubnetgroup`

Tipe: String

Wajib: Tidak

DeletionProtection

Menentukan apakah klaster ini dapat dihapus. Jika `DeletionProtection` diaktifkan, klaster tidak dapat dihapus kecuali jika diubah dan `DeletionProtection` dinonaktifkan. `DeletionProtection` melindungi klaster agar tidak terhapus secara tidak sengaja.

Tipe: Boolean

Wajib: Tidak

EnableCloudwatchLogsExports.member.n

Daftar jenis log yang perlu diaktifkan untuk mengekspor ke Amazon CloudWatch Logs. Anda dapat mengaktifkan log audit atau log profiler. Untuk informasi selengkapnya, lihat Peristiwa Pengauditan Amazon DocumentDB dan Operasi Pembuatan Profil Amazon DocumentDB.

Tipe: Array string

Wajib: Tidak

EngineVersion

Nomor versi mesin basis data yang akan digunakan. `--engine-version` akan menjadi default ke versi mesin utama terbaru. Untuk beban kerja produksi, kami sarankan nyatakan parameter ini secara eksplisit dengan versi mesin utama yang dimaksud.

Tipe: String

Wajib: Tidak

GlobalClusterIdentifier

Pengidentifikasi klaster dari klaster global baru.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Wajib: Tidak

KmsKeyId

Pengidentifikasi AWS KMS kunci untuk cluster terenkripsi.

Pengidentifikasi AWS KMS kunci adalah Amazon Resource Name (ARN) untuk kunci enkripsi AWS KMS . Jika Anda membuat cluster menggunakan yang sama Akun AWS yang memiliki kunci AWS KMS enkripsi yang digunakan untuk mengenkripsi cluster baru, Anda dapat menggunakan alias AWS KMS kunci alih-alih ARN untuk kunci enkripsi. AWS KMS

Jika kunci enkripsi tidak ditentukan dalam `KmsKeyId`:

- Jika parameter `StorageEncrypted` adalah `true`, Amazon DocumentDB menggunakan kunci enkripsi default Anda.

AWS KMS membuat kunci enkripsi default untuk Anda Akun AWS. Anda Akun AWS memiliki kunci enkripsi default yang berbeda untuk masing-masing Wilayah AWS.

Tipe: String

Wajib: Tidak

MasterUsername

Nama pengguna utama untuk klaster.

Batas:

- Harus dari 1 hingga 63 huruf atau angka.
- Karakter pertama harus berupa huruf.
- Tidak dapat menjadi kata yang direservasi untuk mesin basis data yang dipilih.

Tipe: String

Wajib: Tidak

MasterUserPassword

Kata sandi untuk pengguna basis data utama. Kata sandi ini dapat berisi karakter ASCII apa pun yang dapat dicetak kecuali garis miring (/), tanda kutip ganda ("), atau simbol "at" (@).

Kendala: Harus berisi dari 8 hingga 100 karakter.

Tipe: String

Wajib: Tidak

Port

Nomor port tempat instans dalam klaster menerima koneksi.

Tipe: Integer

Wajib: Tidak

PreferredBackupWindow

Rentang waktu harian selama pencadangan otomatis dibuat jika pencadangan otomatis diaktifkan menggunakan parameter `BackupRetentionPeriod`.

Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing. Wilayah AWS

Batasan:

- Harus dalam format `hh24:mi-hh24:mi`.

- Harus dalam Waktu Universal Terkoordinasi (UTC).
- Tidak boleh bertentangan dengan jendela pemeliharaan yang diinginkan.
- Harus setidaknya 30 menit.

Tipe: String

Wajib: Tidak

PreferredMaintenanceWindow

Rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, dalam Waktu Universal Terkoordinasi (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing Wilayah AWS, terjadi pada hari acak dalam seminggu.

Hari yang valid: Sen, Sel, Rab, Kam, Jum, Sab, Min

Kendala: Minimum 30 menit jendela.

Tipe: String

Wajib: Tidak

PreSignedUrl

Saat ini tidak didukung.

Tipe: String

Wajib: Tidak

StorageEncrypted

Menentukan apakah klaster dienkripsi.

Tipe: Boolean

Wajib: Tidak

StorageType

Jenis penyimpanan untuk dikaitkan dengan cluster DB.

Untuk informasi tentang jenis penyimpanan untuk klaster Amazon DocumentDB, lihat Konfigurasi penyimpanan klaster di Panduan Pengembang Amazon DocumentDB.

Nilai yang valid untuk jenis penyimpanan - `standard | iopt1`

Nilai default adalah `standard`

> **Note**
>
> Saat Anda membuat cluster DB DocumentDB dengan jenis penyimpanan disetel `iopt1` ke, jenis penyimpanan dikembalikan dalam respons. Jenis penyimpanan tidak dikembalikan saat Anda mengaturnya`standard`.

Tipe: String

Wajib: Tidak

Tags.Tag.N

Tag yang akan ditetapkan ke klaster.

Tipe: Array objek Tag

Wajib: Tidak

VpcSecurityGroupIds. VpcSecurityGroupId.N

Sebuah daftar grup keamanan VPC EC2 untuk dikaitkan dengan kluster ini.

Tipe: Array string

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterAlreadyExistsFault

Anda sudah memiliki klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName` tidak mengacu ke grup parameter klaster yang ada.

Kode Status HTTP: 404

DBClusterQuotaExceededFault

Klaster tidak dapat dibuat karena Anda telah mencapai kuota maksimum klaster yang diizinkan.

Kode Status HTTP: 403

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

DBSubnetGroupDoesNotCoverEnoughAZs

Subnet dalam grup subnet harus mencakup setidaknya dua Availability Zone kecuali hanya ada satu Availability Zone.

Kode Status HTTP: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

GlobalClusterNotFoundFault

`GlobalClusterIdentifier` tidak mengacu ke sebuah klaster global yang ada.

Kode Status HTTP: 404

InsufficientStorageClusterCapacity

Tidak ada cukup penyimpanan yang tersedia untuk tindakan saat ini. Anda mungkin dapat mengatasi kesalahan ini dengan memperbarui grup subnet Anda untuk menggunakan Availability Zone berbeda yang memiliki lebih banyak penyimpanan yang tersedia.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

InvalidDBSubnetGroupStateFault

Grup subnet tidak dapat dihapus karena sedang digunakan.

Kode Status HTTP: 400

InvalidGlobalClusterStateFault

Operasi yang diminta tidak dapat dilakukan saat klaster dalam keadaan ini.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

InvalidVPCNetworkStateFault

Grup subnet tidak mencakup semua Availability Zone setelah dibuat karena perubahan yang dibuat.

Kode Status HTTP: 400

KMSKeyNotAccessibleFault

Terjadi kesalahan saat mengakses AWS KMS kunci.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateDBClusterParameterGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat grup parameter klaster baru.

Parameter dalam grup parameter klaster berlaku untuk semua instans dalam sebuah klaster.

Grup parameter klaster awalnya dibuat dengan parameter default untuk mesin basis data yang digunakan oleh instans dalam klaster. Di Amazon DocumentDB, Anda tidak dapat membuat modifikasi langsung ke grup parameter klaster `default.docdb3.6`. Jika klaster Amazon DocumentDB Anda menggunakan grup parameter klaster default dan Anda ingin mengubah nilai di dalamnya, Anda harus terlebih dahulu membuat grup parameter baru atau menyalin grup parameter yang ada, memodifikasinya, lalu menerapkan grup parameter yang dimodifikasi ke klaster Anda. Agar grup parameter klaster baru dan pengaturan terkait diterapkan, Anda harus me-reboot instans di klaster tanpa failover. Untuk informasi selengkapnya, lihat Memodifikasi Grup Parameter Klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterParameterGroupName

Nama grup parameter klaster.

Batasan:

- Tidak boleh sesuai dengan nama `DBClusterParameterGroup` yang ada.

> **Note**
>
> Nilai ini disimpan sebagai string huruf kecil.

Tipe: String

Diperlukan: Ya

DBParameterGroupFamily

Nama famili grup parameter klaster.

Tipe: String

Diperlukan: Ya

Description

Deskripsi untuk grup parameter klaster.

Tipe: String

Diperlukan: Ya

Tags.Tag.N

Tag yang akan ditetapkan ke grup parameter klaster.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterParameterGroup

Informasi mendetail tentang grup parameter klaster.

Tipe: Objek DBClusterParameterGroup

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupAlreadyExists

Grup parameter dengan nama yang sama sudah ada.

Kode Status HTTP: 400

DBParameterGroupQuotaExceeded

Permintaan ini akan menyebabkan Anda melebihi jumlah grup parameter yang diizinkan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateDBClusterSnapshot

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat snapshot dari sebuah klaster.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster untuk membuat snapshotnya. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus cocok dengan pengidentifikasi `DBCluster` yang ada.

Contoh: `my-cluster`

Tipe: String

Diperlukan: Ya

DBClusterSnapshotIdentifier

Pengidentifikasi snapshot klaster. Parameter ini disimpan sebagai string huruf kecil.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-cluster-snapshot1`

Tipe: String

Diperlukan: Ya

Tags.Tag.N

Tag yang akan ditetapkan ke snapshot klaster.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterSnapshot

Informasi mendetail tentang snapshot klaster.

Tipe: Objek DBClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBClusterSnapshotAlreadyExistsFault

Anda sudah memiliki snapshot klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

SnapshotQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah snapshot yang diizinkan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateDBInstance

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat instans baru.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster tempat instans akan berada.

Tipe: String

Diperlukan: Ya

DBInstanceClass

Kapasitas komputasi dan memori instans; misalnya, `db.r5.large`.

Tipe: String

Diperlukan: Ya

DBInstanceIdentifier

Pengidentifikasi instans. Parameter ini disimpan sebagai string huruf kecil.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `mydbinstance`

Tipe: String

Diperlukan: Ya

Engine

Nama mesin basis data yang akan digunakan untuk instans ini.

Nilai yang valid: `docdb`

Tipe: String

Diperlukan: Ya

AutoMinorVersionUpgrade

Parameter ini tidak berlaku untuk Amazon DocumentDB. Amazon DocumentDB tidak melakukan peningkatan versi minor terlepas dari nilai yang ditetapkan.

Default: `false`

Jenis: Boolean

Wajib: Tidak

AvailabilityZone

Availability Zone Amazon EC2 tempat instans dibuat.

Default: Availability Zone acak yang dipilih sistem di titik akhir. Wilayah AWS

Contoh: `us-east-1d`

Tipe: String

Wajib: Tidak

CACertificateIdentifier

Pengidentifikasi sertifikat CA yang akan digunakan untuk sertifikat server instans DB.

Untuk informasi selengkapnya, lihat Memperbarui Sertifikat Amazon DocumentDB TLS Anda dan Mengenkripsi Data dalam Transit di Panduan Pengembang Amazon DocumentDB.

Tipe: String

Wajib: Tidak

CopyTagsToSnapshot

Nilai yang menunjukkan apakah akan menyalin tanda dari instans DB untuk snapshot instans DB. Secara default, tanda tidak disalin.

Tipe: Boolean

Wajib: Tidak

EnablePerformanceInsights

Nilai yang menunjukkan apakah akan mengaktifkan Performance Insights untuk Instans DB. Untuk informasi selengkapnya, lihat Menggunakan Amazon Performance Insights.

Tipe: Boolean

Wajib: Tidak

PerformanceInsightsKMSKeyId

Pengenal AWS KMS kunci untuk enkripsi data Performance Insights.

Pengidentifikasi AWS KMS kunci adalah kunci ARN, ID kunci, alias ARN, atau nama alias untuk kunci KMS.

Jika Anda tidak menentukan nilai untuk PerformanceInsights KMSKeyId, maka Amazon DocumentDB menggunakan kunci KMS default Anda. Ada kunci KMS default untuk akun Amazon Web Services Anda. Akun Amazon Web Services Anda memiliki kunci KMS default yang berbeda untuk setiap wilayah Amazon Web Services.

Tipe: String

Wajib: Tidak

PreferredMaintenanceWindow

Rentang waktu setiap minggu untuk melakukan pemeliharaan sistem, dalam Waktu Universal Terkoordinasi (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing Wilayah AWS, terjadi pada hari acak dalam seminggu.

Hari yang valid: Sen, Sel, Rab, Kam, Jum, Sab, Min

Kendala: Minimum 30 menit jendela.

Tipe: String

Wajib: Tidak

PromotionTier

Nilai yang menentukan urutan di mana replika Amazon DocumentDB dipromosikan ke instans primer setelah kegagalan instans primer yang ada.

Default: 1

Nilai yang valid: 0-15

Tipe: Integer

Wajib: Tidak

Tags.Tag.N

Tag yang akan ditetapkan ke instans. Anda dapat menetapkan hingga 10 tag ke sebuah instans.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBInstance

Informasi mendetail tentang sebuah instans.

Tipe: Objek DBInstance

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AuthorizationNotFound

Grup keamanan CIDR IP atau Amazon EC2 yang ditentukan tidak diotorisasi untuk grup keamanan yang ditentukan.

Amazon DocumentDB juga mungkin tidak diizinkan untuk melakukan tindakan yang diperlukan atas nama Anda menggunakan IAM.

Kode Status HTTP: 404

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBInstanceAlreadyExists

Anda sudah memiliki instans dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

DBSecurityGroupNotFound

`DBSecurityGroupName` tidak mengacu ke grup keamanan yang ada.

Kode Status HTTP: 404

DBSubnetGroupDoesNotCoverEnoughAZs

Subnet dalam grup subnet harus mencakup setidaknya dua Availability Zone kecuali hanya ada satu Availability Zone.

Kode Status HTTP: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

InstanceQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah instans yang diizinkan.

Kode Status HTTP: 400

InsufficientDBInstanceCapacity

Kelas instans yang ditentukan tidak tersedia di Availability Zone yang ditentukan.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

InvalidVPCNetworkStateFault

Grup subnet tidak mencakup semua Availability Zone setelah dibuat karena perubahan yang dibuat.

Kode Status HTTP: 400

KMSKeyNotAccessibleFault

Terjadi kesalahan saat mengakses AWS KMS kunci.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

StorageTypeNotSupported

Penyimpanan `StorageType` yang ditentukan tidak dapat dikaitkan dengan instans DB.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateDBSubnetGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat grup subnet baru. Grup subnet harus berisi setidaknya satu subnet di setidaknya dua Availability Zone di Wilayah AWS.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBSubnetGroupDescription

Deskripsi untuk grup subnet.

Tipe: String

Diperlukan: Ya

DBSubnetGroupName

Nama grup subnet. Nilai ini disimpan sebagai string huruf kecil.

Kendala: Harus berisi tidak lebih dari 255 huruf, angka, titik, garis bawah, spasi, atau tanda hubung. Tidak harus default.

Contoh: `mySubnetgroup`

Tipe: String

Diperlukan: Ya

SubnetIds. SubnetIdentifier.N

ID subnet Amazon EC2 untuk grup subnet.

Tipe: Array string

Wajib: Ya

Tags.Tag.N

Tag yang akan ditetapkan ke grup subnet.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBSubnetGroup

Informasi mendetail tentang grup subnet.

Tipe: Objek DBSubnetGroup

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBSubnetGroupAlreadyExists

`DBSubnetGroupName` telah digunakan oleh grup subnet yang ada.

Kode Status HTTP: 400

DBSubnetGroupDoesNotCoverEnoughAZs

Subnet dalam grup subnet harus mencakup setidaknya dua Availability Zone kecuali hanya ada satu Availability Zone.

Kode Status HTTP: 400

DBSubnetGroupQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah grup subnet yang diizinkan.

Kode Status HTTP: 400

DBSubnetQuotaExceededFault

Permintaan akan menyebabkan Anda melebihi jumlah subnet yang diizinkan dalam grup subnet.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateEventSubscription

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat langganan notifikasi peristiwa Amazon DocumentDB. Tindakan ini memerlukan topik Amazon Resource Name (ARN) yang dibuat dengan menggunakan konsol Amazon DocumentDB, konsol Amazon SNS, atau API Amazon SNS. Untuk mendapatkan ARN dengan Amazon SNS, Anda harus membuat topik di Amazon SNS dan berlangganan topik tersebut. ARN ditampilkan di konsol Amazon SNS.

Anda dapat menentukan jenis sumber (`SourceType`) yang Anda ingin untuk diberi tahu. Anda juga dapat memberikan daftar sumber Amazon DocumentDB (`SourceIds`) yang memicu peristiwa, dan Anda dapat memberikan daftar kategori peristiwa (`EventCategories`) untuk peristiwa yang Anda ingin untuk diberi tahu. Misalnya, Anda dapat menentukan `SourceType = db-instance`, `SourceIds = mydbinstance1, mydbinstance2` dan `EventCategories = Availability, Backup`.

Jika Anda menentukan `SourceType` dan `SourceIds`, (seperti `SourceType = db-instance` dan `SourceIdentifier = myDBInstance1`), Anda akan diberi tahu tentang semua peristiwa `db-instance` untuk sumber yang ditentukan. Jika Anda menentukan `SourceType`, tetapi tidak menentukan `SourceIdentifier`, Anda menerima pemberitahuan peristiwa untuk jenis sumber tersebut untuk semua sumber Amazon DocumentDB Anda. Jika Anda tidak menentukan `SourceType` atau `SourceIdentifier`, Anda menerima pemberitahuan tentang kejadian yang dihasilkan dari semua sumber Amazon DocumentDB milik akun pelanggan Anda.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SnsTopicArn

Amazon Resource Name (ARN) dari topik SNS yang dibuat untuk notifikasi peristiwa. Amazon SNS membuat ARN saat Anda membuat topik dan berlangganan topik tersebut.

Tipe: String

Diperlukan: Ya

SubscriptionName

Nama dari langganan.

Kendala: Nama harus kurang dari 255 karakter.

Tipe: String

Diperlukan: Ya

Enabled

Nilai Boolean; atur ke `true` untuk mengaktifkan langganan, atur ke `false` untuk membuat langganan tetapi tidak mengaktifkannya.

Tipe: Boolean

Wajib: Tidak

EventCategories. EventCategory.N

Daftar kategori peristiwa untuk `SourceType` yang ingin Anda langgani.

Tipe: Array string

Wajib: Tidak

SourceIds. SourceId.N

Daftar pengidentifikasi sumber acara yang acaranya dikembalikan. Jika tidak ditentukan, maka semua sumber disertakan dalam respons. Pengidentifikasi harus dimulai dengan huruf dan hanya boleh berisi huruf ASCII, angka, dan tanda hubung; tidak boleh diakhiri dengan tanda hubung atau berisi dua tanda hubung berturut-turut.

Batasan:

- Jika `SourceIds` disediakan, `SourceType` juga harus disediakan.
- Jika jenis sumber adalah sebuah instans, `DBInstanceIdentifier` harus disediakan.
- Jika jenis sumber adalah sebuah grup keamanan, `DBSecurityGroupName` harus disediakan.
- Jika jenis sumber adalah sebuah grup parameter, `DBParameterGroupName` harus disediakan.
- Jika jenis sumber adalah sebuah snapshot, `DBSnapshotIdentifier` harus disediakan.

Tipe: Array string

Wajib: Tidak

SourceType

Jenis sumber yang menghasilkan peristiwa. Misalnya, jika Anda ingin diberi tahu tentang peristiwa yang dihasilkan oleh sebuah instans, Anda akan mengatur parameter ini ke `db-instance`. Jika nilai ini tidak ditentukan, semua peristiwa dikembalikan.

Nilai yang valid:`db-instance`,`db-cluster`,`db-parameter-group`,`db-security-group`, `db-cluster-snapshot`

Tipe: String

Wajib: Tidak

Tags.Tag.N

Tag yang akan ditetapkan ke langganan acara.

Tipe: Array objek Tag

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EventSubscription

Informasi mendetail tentang peristiwa yang telah Anda langgani.

Tipe: Objek EventSubscription

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

EventSubscriptionQuotaExceeded

Anda telah mencapai jumlah maksimum langganan peristiwa.

Kode Status HTTP: 400

SNSInvalidTopic

Amazon SNS telah merespons bahwa ada masalah dengan topik yang ditentukan.

Kode Status HTTP: 400

SNSNoAuthorization

Anda tidak memiliki izin untuk memublikasikan ke Amazon Resource Name (ARN) topik SNS.

Kode Status HTTP: 400

SNSTopicArnNotFound

Amazon Resource Name (ARN) topik SNS tidak ada.

Kode Status HTTP: 404

SourceNotFound

Sumber yang diminta tidak dapat ditemukan.

Kode Status HTTP: 404

SubscriptionAlreadyExist

Nama langganan yang diberikan sudah ada.

Kode Status HTTP: 400

SubscriptionCategoryNotFound

Kategori yang diberikan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateGlobalCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat klaster global Amazon DocumentDB yang dapat menjangkau beberapa Wilayah AWS. Klaster global berisi satu klaster primer dengan kemampuan baca-tulis, dan hingga memberikan klaster sekunder hanya-baca. Klaster global menggunakan replikasi cepat berbasis penyimpanan di seluruh wilayah dengan latensi kurang dari satu detik, menggunakan infrastruktur khusus tanpa memengaruhi performa beban kerja Anda.

Anda dapat membuat sebuah klaster global yang awalnya kosong, dan kemudian menambahkan primer dan sekunder ke dalamnya. Atau Anda dapat menentukan klaster yang ada selama operasi pembuatan, dan klaster ini menjadi yang primer dari klaster global.

> **Note**
>
> Tindakan ini hanya berlaku untuk klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

GlobalClusterIdentifier

Pengidentifikasi klaster dari klaster global baru.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Wajib: Ya

DatabaseName

Nama untuk basis data Anda hingga 64 karakter alfanumerik. Jika Anda tidak memberikan nama, Amazon DocumentDB tidak akan membuat basis data di klaster global yang sedang Anda buat.

Tipe: String

Wajib: Tidak

DeletionProtection

Pengaturan perlindungan penghapusan untuk klaster global baru. Klaster global tidak dapat dihapus saat perlindungan penghapusan diaktifkan.

Tipe: Boolean

Wajib: Tidak

Engine

Nama mesin basis data yang akan digunakan untuk klaster ini.

Tipe: String

Wajib: Tidak

EngineVersion

Versi mesin dari klaster global.

Tipe: String

Wajib: Tidak

SourceDBClusterIdentifier

Amazon Resource Name (ARN) untuk digunakan sebagai klaster primer dari klaster global. Parameter ini bersifat opsional.

Tipe: String

Wajib: Tidak

StorageEncrypted

Pengaturan enkripsi penyimpanan untuk klaster global baru.

Tipe: Boolean

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

GlobalCluster

Sebuah tipe data yang mewakili sebuah klaster global Amazon DocumentDB.

Tipe: Objek GlobalCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

GlobalClusterAlreadyExistsFault

Parameter `GlobalClusterIdentifier` sudah ada. Pilih pengidentifikasi klaster global baru (nama unik) untuk membuat klaster global baru.

Kode Status HTTP: 400

GlobalClusterQuotaExceededFault

Jumlah klaster global untuk akun ini sudah mencapai batas maksimum yang diizinkan.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteDBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus klaster yang telah disediakan sebelumnya. Ketika Anda menghapus sebuah klaster, semua backup otomatis untuk kluster tersebut dihapus dan tidak dapat dipulihkan. Snapshot klaster DB manual dari klaster yang ditentukan tidak dihapus.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster untuk klaster yang akan dihapus. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus cocok dengan `DBClusterIdentifier` yang ada.

Tipe: String

Diperlukan: Ya

FinalDBSnapshotIdentifier

Pengidentifikasi snapshot klaster dari snapshot klaster baru yang dibuat saat `SkipFinalSnapshot` diatur ke `false`.

> **Note**
>
> Menentukan parameter ini dan juga mengatur parameter `SkipFinalShapshot` ke `true` menghasilkan kesalahan.

Batasan:

- Harus dari 1 sampai 255 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Tipe: String

Wajib: Tidak

SkipFinalSnapshot

Menentukan apakah snapshot klaster akhir dibuat sebelum klaster dihapus. Jika `true` ditentukan, tidak ada snapshot klaster yang dibuat. Jika `false` ditentukan, snapshot klaster dibuat sebelum klaster DB dihapus.

> **Note**
>
> Jika `SkipFinalSnapshot` adalah `false`, Anda harus menentukan parameter `FinalDBSnapshotIdentifier`.

Default: `false`

Jenis: Boolean

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBClusterSnapshotAlreadyExistsFault

Anda sudah memiliki snapshot klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

SnapshotQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah snapshot yang diizinkan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteDBClusterParameterGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus grup parameter klaster tertentu. Grup parameter klaster yang akan dihapus tidak dapat dikaitkan dengan klaster mana pun.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterParameterGroupName

Nama grup parameter klaster.

Batasan:

- Harus nama grup parameter klaster yang ada.
- Anda tidak dapat menghapus grup parameter klaster default.
- Tidak dapat dikaitkan dengan klaster mana pun.

Tipe: String

Diperlukan: Ya

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

InvalidDBParameterGroupState

Grup parameter sedang digunakan, atau dalam keadaan tidak valid. Jika Anda mencoba menghapus grup parameter, Anda tidak dapat menghapusnya ketika grup parameter berada dalam keadaan ini.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteDBClusterSnapshot

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus snapshot klaster. Jika snapshot sedang disalin, operasi penyalinan dihentikan.

> **Note**
>
> Snapshot klaster harus dalam status `available` untuk dihapus.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterSnapshotIdentifier

Pengidentifikasi snapshot klaster yang akan dihapus.

Kendala: Harus berupa nama snapshot klaster yang ada dalam status `available`.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterSnapshot

Informasi mendetail tentang snapshot klaster.

Tipe: Objek DBClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteDBInstance

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus instans yang disediakan sebelumnya.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBInstanceIdentifier

Pengidentifikasi instans untuk instans yang akan dihapus. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus cocok dengan nama instans yang ada.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBInstance

Informasi mendetail tentang sebuah instans.

Tipe: Objek DBInstance

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

DBSnapshotAlreadyExists

`DBSnapshotIdentifier` telah digunakan oleh snapshot yang ada.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

SnapshotQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah snapshot yang diizinkan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteDBSubnetGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus grup subnet.

> **Note**
>
> Grup subnet basis data yang ditentukan tidak boleh dikaitkan dengan instans DB apa pun.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBSubnetGroupName

Nama grup subnet basis data yang akan dihapus.

> **Note**
>
> Anda tidak dapat menghapus grup subnet default.

Batasan:

Harus cocok dengan nama `DBSubnetGroup` yang ada. Tidak harus default.

Contoh: `mySubnetgroup`

Tipe: String

Diperlukan: Ya

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

InvalidDBSubnetGroupStateFault

Grup subnet tidak dapat dihapus karena sedang digunakan.

Kode Status HTTP: 400

InvalidDBSubnetStateFault

Subnet tidak dalam status tersedia.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteEventSubscription

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus langganan notifikasi peristiwa Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SubscriptionName

Nama langganan notifikasi peristiwa Amazon DocumentDB yang ingin Anda hapus.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EventSubscription

Informasi mendetail tentang peristiwa yang telah Anda langgani.

Tipe: Objek EventSubscription

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InvalidEventSubscriptionState

Orang lain mungkin memodifikasi langganan. Tunggu beberapa detik, dan coba lagi.

Kode Status HTTP: 400

SubscriptionNotFound

Nama langganan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteGlobalCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus klaster global. Klaster primer dan sekunder harus sudah terlepas atau dihapus sebelum mencoba menghapus klaster global.

> **Note**
>
> Tindakan ini hanya berlaku untuk klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

GlobalClusterIdentifier

Pengidentifikasi klaster dari klaster global yang sedang dihapus.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

GlobalCluster

Sebuah tipe data yang mewakili sebuah klaster global Amazon DocumentDB.

Tipe: Objek GlobalCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

GlobalClusterNotFoundFault

`GlobalClusterIdentifier` tidak mengacu ke sebuah klaster global yang ada.

Kode Status HTTP: 404

InvalidGlobalClusterStateFault

Operasi yang diminta tidak dapat dilakukan saat klaster dalam keadaan ini.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeCertificates

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar sertifikat otoritas sertifikat (CA) yang disediakan oleh Amazon DocumentDB untuk Akun AWS ini.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

CertificateIdentifier

Pengidentifikasi sertifikat yang disediakan pengguna. Jika parameter ini ditentukan, informasi untuk hanya sertifikat tertentu dikembalikan. Jika parameter ini dihilangkan, daftar hingga `MaxRecords` sertifikat dikembalikan. Parameter ini tidak peka huruf besar kecil.

Kendala

- Harus cocok dengan `CertificateIdentifier` yang ada.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan `DescribeCertificates` sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination (pemberian nomor halaman) yang disebut penanda disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Batasan:

- Minimum: 20
- Maksimum: 100

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

Sertifikat.Certificate.N

Daftar sertifikat untuk Akun AWS ini.

Tipe: Array objek Certificate

Marker

Token pagination (pemberian nomor halaman) opsional disediakan jika jumlah catatan yang diambil lebih besar dari `MaxRecords`. Jika parameter ini ditentukan, penanda menentukan catatan berikutnya dalam daftar. Termasuk nilai `Marker` dalam panggilan berikutnya untuk hasil `DescribeCertificates` di halaman sertifikat berikutnya.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

CertificateNotFound

`CertificateIdentifier` tidak mengacu ke sertifikat yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBClusterParameterGroups

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar deskripsi `DBClusterParameterGroup`. Jika parameter `DBClusterParameterGroupName` ditentukan, daftar berisi hanya deskripsi grup parameter klaster yang ditentukan.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterParameterGroupName

Nama grup parameter klaster tertentu yang detailnya dikembalikan.

Batasan:

- Jika disediakan, harus cocok dengan nama `DBClusterParameterGroup` yang ada.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

DB ClusterParameterGroups .DB .N ClusterParameterGroup

Daftar grup parameter klaster.

Tipe: Array objek DBClusterParameterGroup

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBClusterParameters

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar parameter mendetail untuk grup parameter klaster tertentu.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterParameterGroupName

Nama grup parameter klaster tertentu yang detail parameternya dikembalikan.

Batasan:

- Jika disediakan, harus cocok dengan nama `DBClusterParameterGroup` yang ada.

Tipe: String

Diperlukan: Ya

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Source

Nilai yang menunjukkan untuk mengembalikan hanya parameter untuk sumber tertentu. Sumber parameter dapat berupa `engine`, `service`, atau `customer`.

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Parameter.Parameter.N

Menyediakan daftar parameter untuk grup parameter klaster.

Tipe: Array objek Parameter

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBClusters

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan informasi tentang klaster Amazon DocumentDB yang disediakan. Operasi API ini mendukung pagination (pemberian nomor halaman). Untuk fitur manajemen tertentu seperti manajemen siklus hidup klaster dan instans, Amazon DocumentDB memanfaatkan teknologi operasional yang dibagikan dengan Amazon RDS dan Amazon Neptune. Gunakan parameter filter `filterName=engine,Values=docdb` untuk mengembalikan hanya klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster yang disediakan pengguna. Jika parameter ini ditentukan, informasi hanya dari klaster tertentu yang dikembalikan. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Jika disediakan, harus cocok dengan `DBClusterIdentifier` yang ada.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Filter yang menentukan satu atau lebih klaster untuk dideskripsikan.

Filter yang didukung:

- `db-cluster-id` - Menerima pengidentifikasi klaster dan Amazon Resource Name (ARN) klaster. Daftar hasil hanya mencakup informasi tentang klaster yang diidentifikasi oleh ARN ini.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

dbclusters.dbcluster.n

Sebuah daftar klaster.

Tipe: Array objek DBCluster

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBClusterSnapshotAttributes

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar nama dan nilai atribut snapshot klaster untuk snapshot klaster DB manual.

Saat Anda berbagi snapshot dengan yang lain Akun AWS, `DescribeDBClusterSnapshotAttributes` mengembalikan `restore` atribut dan daftar ID untuk Akun AWS yang diizinkan untuk menyalin atau memulihkan snapshot cluster manual. Jika `all` disertakan dalam daftar nilai untuk atribut `restore`, maka snapshot klaster manual bersifat publik dan dapat disalin atau dipulihkan oleh semua Akun AWS.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterSnapshotIdentifier

Pengidentifikasi untuk snapshot klaster untuk mendeskripsikan atributnya.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterSnapshotAttributesResult

Informasi mendetail tentang atribut yang terkait dengan snapshot klaster.

Tipe: Objek DBClusterSnapshotAttributesResult

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBClusterSnapshots

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan informasi tentang snapshot klaster. Operasi API ini mendukung pagination (pemberian nomor halaman).

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

ID dari klaster untuk mengambil daftar snapshot klaster. Parameter ini tidak dapat digunakan dengan parameter `DBClusterSnapshotIdentifier`. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Jika disediakan, harus cocok dengan pengidentifikasi `DBCluster` yang ada.

Tipe: String

Wajib: Tidak

DBClusterSnapshotIdentifier

Pengidentifikasi snapshot klaster tertentu untuk menggambarkan. Parameter ini tidak dapat digunakan dengan parameter `DBClusterIdentifier`. Nilai ini disimpan sebagai string huruf kecil.

Batasan:

- Jika disediakan, harus cocok dengan pengidentifikasi `DBClusterSnapshot` yang ada.
- Jika pengidentifikasi ini adalah untuk snapshot otomatis, parameter `SnapshotType` juga harus ditentukan.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

IncludePublic

Setel `true` untuk menyertakan snapshot cluster manual yang bersifat publik dan dapat disalin atau dipulihkan oleh siapa pun Akun AWS, dan sebaliknya. `false` Defaultnya adalah `false`.

Tipe: Boolean

Wajib: Tidak

IncludeShared

Setel `true` untuk menyertakan snapshot cluster manual bersama dari Akun AWS yang lain yang Akun AWS telah diberi izin untuk menyalin atau memulihkan, dan sebaliknya`false`. Defaultnya adalah `false`.

Tipe: Boolean

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

SnapshotType

Jenis snapshot klaster untuk dikembalikan. Anda dapat menentukan salah satu nilai berikut:

- `automated` - Mengembalikan semua snapshot klaster yang telah dibuat secara otomatis oleh Amazon DocumentDB untuk Akun AWS Anda.
- `manual` - Mengembalikan semua snapshot klaster yang telah Anda buat secara manual untuk Akun AWS Anda.
- `shared` - Mengembalikan semua snapshot klaster manual yang telah dibagikan ke Akun AWS Anda.
- `public` - Mengembalikan semua snapshot klaster yang telah ditandai sebagai publik.

Jika Anda tidak menentukan nilai `SnapshotType`, maka baik snapshot klaster otomatis dan manual dikembalikan. Anda dapat menyertakan snapshot klaster bersama dengan hasil ini dengan mengatur parameter `IncludeShared` ke `true`. Anda dapat menyertakan snapshot klaster publik dengan hasil ini dengan mengatur parameter `IncludePublic` ke `true`.

Parameter `IncludeShared` dan `IncludePublic` tidak berlaku untuk nilai `SnapshotType` dari `manual` atau `automated`. Parameter `IncludePublic` tidak berlaku bila `SnapshotType` diatur ke `shared`. Parameter `IncludeShared` tidak berlaku bila `SnapshotType` diatur ke `public`.

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

DB ClusterSnapshots .DB .N ClusterSnapshot

Menyediakan daftar snapshot klaster.

Tipe: Array objek DBClusterSnapshot

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBEngineVersions

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar dari mesin yang tersedia.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBParameterGroupFamily

Nama famili grup parameter tertentu yang detailnya dikembalikan.

Batasan:

- Jika disediakan, harus cocok dengan `DBParameterGroupFamily` yang ada.

Tipe: String

Wajib: Tidak

DefaultOnly

Menunjukkan bahwa hanya versi default dari mesin tertentu atau mesin dan kombinasi versi utama dikembalikan.

Tipe: Boolean

Wajib: Tidak

Engine

Mesin basis data untuk mengembalikan.

Tipe: String

Wajib: Tidak

EngineVersion

Versi mesin basis data untuk mengembalikan.

Contoh: `3.6.0`

Tipe: String

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

ListSupportedCharacterSets

Jika parameter ini ditentukan dan mesin yang diminta mendukung parameter `CharacterSetName` untuk `CreateDBInstance`, responsnya mencakup daftar set karakter yang didukung untuk setiap versi mesin.

Tipe: Boolean

Wajib: Tidak

ListSupportedTimezones

Jika parameter ini ditentukan dan mesin yang diminta mendukung parameter `TimeZone` untuk `CreateDBInstance`, responsnya mencakup daftar zona waktu yang didukung untuk setiap versi mesin.

Tipe: Boolean

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

DB EngineVersions .DB .N EngineVersion

Informasi mendetail tentang satu atau lebih versi mesin.

Tipe: Array objek DBEngineVersion

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBInstances

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan informasi tentang instans Amazon DocumentDB yang disediakan. API ini mendukung pagination (pemberian nomor halaman).

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBInstanceIdentifier

Pengidentifikasi instans yang disediakan pengguna. Jika parameter ini ditentukan, informasi hanya dari instans tertentu yang dikembalikan. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Jika disediakan, harus cocok dengan pengidentifikasi `DBInstance` yang ada.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Filter yang menentukan satu atau lebih instans untuk dideskripsikan.

Filter yang didukung:

- `db-cluster-id` - Menerima pengidentifikasi klaster dan Amazon Resource Name (ARN) klaster. Daftar hasil mencakup hanya informasi tentang instans yang terkait dengan klaster yang diidentifikasi oleh ARN ini.
- `db-instance-id` - Menerima pengidentifikasi instans dan ARN instans. Daftar hasil mencakup hanya informasi tentang instans yang diidentifikasi oleh ARN ini.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

dbInstances.dbInstance.n

Informasi mendetail tentang satu atau lebih instans.

Tipe: Array objek DBInstance

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeDBSubnetGroups

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar deskripsi `DBSubnetGroup`. Jika `DBSubnetGroupName` ditentukan, daftar akan berisi hanya deskripsi dari `DBSubnetGroup` yang ditentukan.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBSubnetGroupName

Nama grup subnet yang akan dikembalikan detailnya.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

DB SubnetGroups .DB .N SubnetGroup

Informasi mendetail tentang satu atau lebih grup subnet.

Tipe: Array objek DBSubnetGroup

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeEngineDefaultClusterParameters

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan mesin default dan informasi parameter sistem untuk mesin basis data klaster.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBParameterGroupFamily

Nama famili grup parameter klaster untuk mengembalikan informasi parameter mesin.

Tipe: String

Diperlukan: Ya

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EngineDefaults

Berisi hasil pemanggilan operasi `DescribeEngineDefaultClusterParameters` yang berhasil.

Tipe: Objek EngineDefaults

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeEventCategories

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menampilkan daftar kategori untuk semua jenis sumber peristiwa, atau, jika ditentukan, untuk jenis sumber tertentu.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

SourceType

Jenis sumber yang menghasilkan acara.

Nilai valid: `db-instance`, `db-parameter-group`, `db-security-group`

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EventCategoriesMapList. EventCategoriesMap.N

Daftar peta kategori peristiwa.

Tipe: Array objek EventCategoriesMap

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeEvents

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan peristiwa yang terkait dengan instans, grup keamanan, snapshot, dan grup parameter DB selama 14 hari terakhir. Anda dapat memperoleh peristiwa khusus untuk instans DB, grup keamanan, snapshot, atau grup parameter tertentu dengan memberikan nama sebagai parameter. Secara default, peristiwa jam terakhir dikembalikan.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Duration

Jumlah menit untuk mengambil peristiwa.

Default: 60

Tipe: Integer

Wajib: Tidak

EndTime

Akhir interval waktu untuk mengambil peristiwa, ditentukan dalam format ISO 8601.

Contoh: 2009-07-08T18:00Z

Tipe: Timestamp

Wajib: Tidak

EventCategories. EventCategory.N

Daftar kategori peristiwa yang memicu notifikasi untuk langganan notifikasi peristiwa.

Tipe: Array string

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

SourceIdentifier

Pengidentifikasi sumber peristiwa yang peristiwanya dikembalikan. Jika tidak ditentukan, maka semua sumber disertakan dalam respons.

Batasan:

- Jika `SourceIdentifier` disediakan, `SourceType` juga harus disediakan.
- Jika jenis sumber adalah `DBInstance`, sebuah `DBInstanceIdentifier` harus disediakan.
- Jika jenis sumber adalah `DBSecurityGroup`, sebuah `DBSecurityGroupName` harus disediakan.
- Jika jenis sumber adalah `DBParameterGroup`, sebuah `DBParameterGroupName` harus disediakan.
- Jika jenis sumber adalah `DBSnapshot`, sebuah `DBSnapshotIdentifier` harus disediakan.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Tipe: String

Wajib: Tidak

SourceType

Sumber peristiwa untuk mengambil peristiwa. Jika tidak ada nilai yang ditentukan, semua peristiwa dikembalikan.

Jenis: String

Nilai yang Valid: `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Wajib: Tidak

StartTime

Awal interval waktu untuk mengambil peristiwa, ditentukan dalam format ISO 8601.

Contoh: 2009-07-08T18:00Z

Tipe: Timestamp

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

Peristiwa.Event.N

Informasi mendetail tentang satu atau lebih peristiwa.

Tipe: Array objek Event

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeEventSubscriptions

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mencantumkan semua deskripsi langganan untuk akun pelanggan. Deskripsi untuk berlangganan termasuk `SubscriptionName`, `SNSTopicARN`, `CustomerID`, `SourceType`, `SourceID`, `CreationTime`, dan `Status`.

Jika Anda menentukan `SubscriptionName`, cantumkan deskripsi untuk langganan tersebut.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

SubscriptionName

Nama langganan notifikasi peristiwa Amazon DocumentDB yang ingin Anda deskripsikan.

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

EventSubscriptionsList. EventSubscription.N

Daftar langganan peristiwa.

Tipe: Array objek EventSubscription

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

SubscriptionNotFound

Nama langganan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeGlobalClusters

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan informasi tentang klaster global Amazon DocumentDB. API ini mendukung pagination (pemberian nomor halaman).

> **Note**
>
> Tindakan ini hanya berlaku untuk klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Filter.Filter.N

Filter yang menentukan satu atau lebih klaster DB global untuk dideskripsikan.

Filter yang didukung: `db-cluster-id` menerima pengidentifikasi klaster dan Amazon Resource Name (ARN) klaster. Daftar hasil akan hanya menyertakan informasi tentang klaster yang diidentifikasi oleh ARN ini.

Tipe: Array objek Filter

Wajib: Tidak

GlobalClusterIdentifier

Pengidentifikasi klaster yang disediakan pengguna. Jika parameter ini ditentukan, informasi hanya dari klaster tertentu yang dikembalikan. Parameter ini tidak peka huruf besar kecil.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan `DescribeGlobalClusters` sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination (pemberian nomor halaman) yang disebut penanda disertakan dalam respons sehingga Anda dapat mengambil hasil yang tersisa.

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

GlobalClusters. GlobalClusterMember.N

Tipe: Array objek GlobalCluster

Marker

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

GlobalClusterNotFoundFault

`GlobalClusterIdentifier` tidak mengacu ke sebuah klaster global yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeOrderableDBInstanceOptions

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar opsi instans yang dapat dipesan untuk mesin yang ditentukan.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Engine

Nama dari mesin untuk mengambil opsi instans.

Tipe: String

Diperlukan: Ya

DBInstanceClass

Nilai filter kelas instans. Tentukan parameter ini untuk hanya menampilkan penawaran yang tersedia yang cocok dengan kelas instans yang ditentukan.

Tipe: String

Wajib: Tidak

EngineVersion

Nilai filter versi mesin. Tentukan parameter ini untuk hanya menampilkan penawaran yang tersedia yang cocok dengan versi mesin yang ditentukan.

Tipe: String

Wajib: Tidak

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

LicenseModel

Nilai filter model lisensi. Tentukan parameter ini untuk hanya menampilkan penawaran yang tersedia yang cocok dengan model lisensi yang ditentukan.

Tipe: String

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

Vpc

Nilai filter virtual private cloud (VPC). Tentukan parameter ini untuk hanya menampilkan penawaran VPC atau non-VPC yang tersedia.

Tipe: Boolean

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

DipesananB .OrderableDB InstanceOptions .N InstanceOption

Opsi yang tersedia untuk instans tertentu yang dapat dipesan.

Tipe: Array objek OrderableDBInstanceOption

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribePendingMaintenanceActions

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan daftar sumber daya (misalnya, instans) yang memiliki setidaknya satu tindakan pemeliharaan tertunda.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Filter.Filter.N

Filter yang menentukan satu atau lebih sumber daya untuk mengembalikan tindakan pemeliharaan yang tertunda.

Filter yang didukung:

- `db-cluster-id` - Menerima pengidentifikasi klaster dan Amazon Resource Name (ARN) klaster. Daftar yang mencakup hanya tindakan pemeliharaan yang tertunda untuk klaster yang diidentifikasi oleh ARN ini.
- `db-instance-id` - Menerima pengidentifikasi instans dan ARN instans. Daftar hasil mencakup hanya tindakan pemeliharaan yang tertunda untuk instans DB yang diidentifikasi oleh ARN ini.

Tipe: Array objek Filter

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

MaxRecords

Jumlah maksimum catatan yang akan dikembalikan dalam respons. Jika lebih banyak catatan ada daripada nilai `MaxRecords` yang ditentukan, token pagination / pemberian nomor halaman (penanda) disertakan dalam respons sehingga hasil yang tersisa dapat diambil.

Default: 100

Kendala: Minimum 20, maksimum 100.

Tipe: Integer

Wajib: Tidak

ResourceIdentifier

ARN dari sumber daya untuk mengembalikan tindakan pemeliharaan yang tertunda.

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen-elemen berikut dikembalikan oleh layanan.

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Jenis: String

PendingMaintenanceActions. ResourcePendingMaintenanceActions.N

Tindakan pemeliharaan yang akan diterapkan.

Tipe: Array objek ResourcePendingMaintenanceActions

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

ResourceNotFoundFault

ID sumber daya yang ditentukan tidak ditemukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

FailoverDBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memaksa failover untuk sebuah klaster.

Failover untuk klaster mempromosikan salah satu replika Amazon DocumentDB (instans hanya-baca) di klaster menjadi instans primer (penulis klaster).

Jika instans primer gagal, Amazon DocumentDB secara otomatis gagal ke replika Amazon DocumentDB, jika ada. Anda dapat memaksa failover saat ingin mensimulasikan kegagalan instans primer untuk pengujian.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster untuk memaksa failover. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus cocok dengan pengidentifikasi `DBCluster` yang ada.

Tipe: String

Wajib: Tidak

TargetDBInstanceIdentifier

Nama instans yang akan dipromosikan ke instans primer.

Anda harus menentukan pengidentifikasi instans untuk replika Amazon DocumentDB di klaster. Sebagai contoh, `mydbcluster-replica1`.

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3

- AWS SDK untuk Python
- AWS SDK for Ruby V3

ListTagsForResource

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mencantumkan semua tag pada sumber daya Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

ResourceName

Sumber daya Amazon DocumentDB dengan tag untuk dicantumkan. Nilai ini adalah sebuah Amazon Resource Name (ARN).

Tipe: String

Diperlukan: Ya

Filter.Filter.N

Parameter ini saat ini tidak didukung.

Tipe: Array objek Filter

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

TagList.Tag.N

Daftar dari satu atau beberapa tag.

Tipe: Array objek Tag

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` tidak mengacu ke sebuah snapshot yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyDBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memodifikasi pengaturan untuk klaster Amazon DocumentDB. Anda dapat mengubah satu atau lebih parameter konfigurasi basis data dengan menentukan parameter ini dan nilai-nilai baru dalam permintaan.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster untuk klaster yang sedang dimodifikasi. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus cocok dengan pengidentifikasi `DBCluster` yang ada.

Tipe: String

Diperlukan: Ya

AllowMajorVersionUpgrade

Nilai yang menunjukkan apakah pembaruan versi utama diizinkan.

Kendala: Anda harus mengizinkan upgrade versi mayor saat menentukan nilai untuk `EngineVersion` parameter yang merupakan versi mayor yang berbeda dari versi cluster DB saat ini.

Tipe: Boolean

Wajib: Tidak

ApplyImmediately

Nilai yang menentukan apakah perubahan dalam permintaan ini dan perubahan yang tertunda diterapkan secara asinkron sesegera mungkin, terlepas dari pengaturan `PreferredMaintenanceWindow` untuk klaster. Jika parameter ini diatur ke `false`, perubahan ke klaster diterapkan selama jendela pemeliharaan berikutnya.

Parameter `ApplyImmediately` memengaruhi hanya nilai `NewDBClusterIdentifier` dan `MasterUserPassword`. Jika Anda mengatur nilai parameter ini ke `false`, perubahan nilai `NewDBClusterIdentifier` dan `MasterUserPassword` diterapkan selama jendela pemeliharaan berikutnya. Semua perubahan lainnya akan diterapkan dengan segera, terlepas dari nilai parameter `ApplyImmediately`.

Default: `false`

Jenis: Boolean

Wajib: Tidak

BackupRetentionPeriod

Jumlah hari penyimpanan cadangan otomatis. Anda harus menentukan nilai minimum 1.

Default: 1

Batas:

- Harus berupa nilai dari 1 hingga 35.

Tipe: Integer

Wajib: Tidak

CloudwatchLogsExportConfiguration

Pengaturan konfigurasi untuk jenis log yang akan diaktifkan untuk diekspor ke Amazon CloudWatch Logs untuk instance atau klaster tertentu. `DisableLogTypes`Array `EnableLogTypes` dan menentukan log mana yang diekspor (atau tidak diekspor) ke Log. CloudWatch

Tipe: Objek CloudwatchLogsExportConfiguration

Wajib: Tidak

DBClusterParameterGroupName

Nama grup parameter klaster yang digunakan untuk klaster.

Tipe: String

Wajib: Tidak

DeletionProtection

Menentukan apakah klaster ini dapat dihapus. Jika `DeletionProtection` diaktifkan, klaster tidak dapat dihapus kecuali jika diubah dan `DeletionProtection` dinonaktifkan. `DeletionProtection` melindungi klaster agar tidak terhapus secara tidak sengaja.

Tipe: Boolean

Wajib: Tidak

EngineVersion

Nomor versi mesin basis data yang ingin Anda tingkatkan. Mengubah parameter ini menghasilkan pemadaman. Perubahan diterapkan selama jendela pemeliharaan berikutnya kecuali `ApplyImmediately` diaktifkan.

Untuk membuat daftar semua versi engine yang tersedia untuk Amazon DocumentDB gunakan perintah berikut:

```
aws docdb describe-db-engine-versions --engine docdb --query
"DBEngineVersions[].EngineVersion"
```

Tipe: String

Wajib: Tidak

MasterUserPassword

Kata sandi untuk pengguna basis data utama. Kata sandi ini dapat berisi karakter ASCII apa pun yang dapat dicetak kecuali garis miring (/), tanda kutip ganda ("), atau simbol "at" (@).

Kendala: Harus berisi dari 8 hingga 100 karakter.

Tipe: String

Wajib: Tidak

NewDBClusterIdentifier

Pengidentifikasi klaster baru untuk klaster ketika mengubah nama klaster. Nilai ini disimpan sebagai string huruf kecil.

Kendala:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.

- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-cluster2`

Tipe: String

Wajib: Tidak

Port

Nomor port tempat klaster menerima koneksi.

Kendala: Harus berupa nilai dari `1150` hingga `65535`.

Default: Port yang sama dengan klaster asli.

Tipe: Integer

Wajib: Tidak

PreferredBackupWindow

Kisaran waktu harian selama backup otomatis dibuat jika backup otomatis diaktifkan, menggunakan parameter `BackupRetentionPeriod`.

Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing. Wilayah AWS

Batasan:

- Harus dalam format `hh24:mi-hh24:mi`.
- Harus dalam Waktu Universal Terkoordinasi (UTC).
- Tidak boleh bertentangan dengan jendela pemeliharaan yang diinginkan.
- Harus setidaknya 30 menit.

Tipe: String

Wajib: Tidak

PreferredMaintenanceWindow

Rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, dalam Waktu Universal Terkoordinasi (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Defaultnya adalah jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing Wilayah AWS, terjadi pada hari acak dalam seminggu.

Hari yang valid: Sen, Sel, Rab, Kam, Jum, Sab, Min

Kendala: Minimum 30 menit jendela.

Tipe: String

Wajib: Tidak

StorageType

Jenis penyimpanan untuk mengasosiasikan dengan cluster DB.

Untuk informasi tentang jenis penyimpanan untuk klaster Amazon DocumentDB, lihat Konfigurasi penyimpanan klaster di Panduan Pengembang Amazon DocumentDB.

Nilai yang valid untuk jenis penyimpanan - `standard | iopt1`

Nilai defaultnya adalah `standard`

Tipe: String

Wajib: Tidak

VpcSecurityGroupIds. VpcSecurityGroupId.N

Daftar grup keamanan virtual private cloud (VPC) tempat klaster akan berada.

Tipe: Array string

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterAlreadyExistsFault

Anda sudah memiliki klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName` tidak mengacu ke grup parameter klaster yang ada.

Kode Status HTTP: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

InvalidDBSecurityGroupState

Status grup keamanan tidak mengizinkan penghapusan.

Kode Status HTTP: 400

InvalidDBSubnetGroupStateFault

Grup subnet tidak dapat dihapus karena sedang digunakan.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

InvalidVPCNetworkStateFault

Grup subnet tidak mencakup semua Availability Zone setelah dibuat karena perubahan yang dibuat.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyDBClusterParameterGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memodifikasi parameter dari grup parameter klaster. Untuk mengubah lebih dari satu parameter, kirimkan daftar berikut ini: `ParameterName`, `ParameterValue`, dan `ApplyMethod`. Maksimum 20 parameter dapat dimodifikasi dalam satu permintaan.

> **Note**
>
> Perubahan pada parameter dinamis diterapkan segera. Perubahan pada parameter statis memerlukan boot ulang atau jendela pemeliharaan sebelum perubahan dapat diterapkan.

> **Important**
>
> Setelah Anda membuat grup parameter klaster, Anda harus menunggu setidaknya 5 menit sebelum membuat klaster pertama Anda yang menggunakan grup parameter klaster tersebut sebagai grup parameter default. Ini memungkinkan Amazon DocumentDB untuk menyelesaikan sepenuhnya tindakan pembuatan sebelum grup parameter digunakan sebagai default untuk klaster baru. Langkah ini sangat penting untuk parameter yang sangat penting saat membuat basis data default untuk sebuah klaster, seperti kumpulan karakter untuk basis data default yang ditentukan oleh parameter `character_set_database`.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterParameterGroupName

Nama grup parameter klaster yang akan dimodifikasi.

Tipe: String

Diperlukan: Ya

Parameter.Parameter.N

Daftar parameter dalam grup parameter klaster untuk dimodifikasi.

Tipe: Array objek Parameter

Wajib: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterParameterGroupName

Nama sebuah grup parameter klaster.

Batasan:

- Harus dari 1 hingga 255 huruf atau angka.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

> **Note**
>
> Nilai ini disimpan sebagai string huruf kecil.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

InvalidDBParameterGroupState

Grup parameter sedang digunakan, atau dalam keadaan tidak valid. Jika Anda mencoba menghapus grup parameter, Anda tidak dapat menghapusnya ketika grup parameter berada dalam keadaan ini.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyDBClusterSnapshotAttribute

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menambahkan atribut dan nilai-nilai ke, atau menghapus atribut dan nilai-nilai dari, snapshot klaster manual.

Untuk berbagi snapshot cluster manual dengan yang lain Akun AWS, tentukan `restore` sebagai`AttributeName`, dan gunakan `ValuesToAdd` parameter untuk menambahkan daftar ID Akun AWS yang diotorisasi untuk memulihkan snapshot cluster manual. Gunakan nilai `all` untuk membuat snapshot klaster manual menjadi publik, yang berarti bahwa itu dapat disalin atau dipulihkan oleh semua Akun AWS. Jangan menambahkan nilai `all` untuk setiap snapshot klaster manual yang berisi informasi privat yang Anda tidak ingin tersedia untuk semua Akun AWS. Jika snapshot cluster manual dienkripsi, itu dapat dibagikan, tetapi hanya dengan menentukan daftar Akun AWS ID resmi untuk parameter tersebut. `ValuesToAdd` Anda tidak dapat menggunakan `all` sebagai nilai untuk parameter tersebut dalam kasus ini.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

AttributeName

Nama dari atribut snapshot klaster yang akan diubah.

Untuk mengelola otorisasi bagi orang lain Akun AWS untuk menyalin atau memulihkan snapshot cluster manual, tetapkan nilai ini ke. `restore`

Tipe: String

Diperlukan: Ya

DBClusterSnapshotIdentifier

Pengidentifikasi untuk snapshot klaster untuk memodifikasi atributnya.

Tipe: String

Diperlukan: Ya

ValuesToAdd. AttributeValue.N

Daftar atribut snapshot klaster untuk ditambahkan ke atribut yang ditentukan oleh `AttributeName`.

Untuk mengotorisasi Akun AWS orang lain untuk menyalin atau memulihkan snapshot cluster manual, setel daftar ini untuk menyertakan satu atau beberapa Akun AWS ID. Untuk membuat snapshot cluster manual dapat dipulihkan oleh siapa pun Akun AWS, setel ke. `all` Jangan menambahkan nilai `all` untuk setiap snapshot klaster manual yang berisi informasi privat yang Anda tidak ingin menjadi tersedia untuk semua Akun AWS.

Tipe: Array string

Wajib: Tidak

ValuesToRemove. AttributeValue.N

Daftar atribut snapshot klaster untuk dihapus dari atribut yang ditentukan oleh `AttributeName`.

Untuk menghapus otorisasi bagi orang lain Akun AWS untuk menyalin atau memulihkan snapshot cluster manual, setel daftar ini untuk menyertakan satu atau beberapa Akun AWS pengidentifikasi. Untuk menghapus otorisasi bagi siapa pun Akun AWS untuk menyalin atau memulihkan snapshot cluster, atur ke. `all` Jika Anda menentukan`all`, ID Akun AWS akunnya secara eksplisit ditambahkan ke `restore` atribut masih dapat menyalin atau memulihkan snapshot cluster manual.

Tipe: Array string

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterSnapshotAttributesResult

Informasi mendetail tentang atribut yang terkait dengan snapshot klaster.

Tipe: Objek DBClusterSnapshotAttributesResult

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

SharedSnapshotQuotaExceeded

Anda telah melampaui jumlah maksimum akun yang dapat Anda bagikan snapshot DB manual.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyDBInstance

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memodifikasi pengaturan untuk sebuah instans. Anda dapat mengubah satu atau lebih parameter konfigurasi basis data dengan menentukan parameter ini dan nilai-nilai baru dalam permintaan.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBInstanceIdentifier

Pengidentifikasi instans. Nilai ini disimpan sebagai string huruf kecil.

Batasan:

- Harus cocok dengan pengidentifikasi `DBInstance` yang ada.

Tipe: String

Diperlukan: Ya

ApplyImmediately

Menentukan apakah modifikasi dalam permintaan ini dan modifikasi yang tertunda diterapkan secara asinkron sesegera mungkin, terlepas dari pengaturan `PreferredMaintenanceWindow` untuk instans.

Jika parameter ini diatur ke `false`, perubahan ke instans diterapkan selama jendela pemeliharaan berikutnya. Beberapa perubahan parameter dapat menyebabkan pemadaman dan diterapkan pada boot ulang berikutnya.

Default: `false`

Jenis: Boolean

Wajib: Tidak

AutoMinorVersionUpgrade

Parameter ini tidak berlaku untuk Amazon DocumentDB. Amazon DocumentDB tidak melakukan peningkatan versi minor terlepas dari nilai yang ditetapkan.

Tipe: Boolean

Wajib: Tidak

CACertificateIdentifier

Menunjukkan sertifikat yang perlu dikaitkan dengan instans.

Tipe: String

Wajib: Tidak

CertificateRotationRestart

Menentukan apakah instans DB dimulai ulang ketika Anda memutar sertifikat SSL/TLS Anda.

Secara default, instans DB dimulai ulang saat Anda memutar sertifikat SSL/TLS Anda. Sertifikat tidak diperbarui sampai instans DB dimulai ulang.

> **Important**
>
> Atur parameter ini hanya jika Anda tidak menggunakan SSL/TLS untuk terhubung ke instans DB.

Jika Anda menggunakan SSL/TLS untuk menyambung ke instans DB, lihat Memperbarui Sertifikat Amazon DocumentDB TLS Anda dan Mengenkripsi Data dalam Transit di Panduan Pengembang Amazon DocumentDB.

Tipe: Boolean

Wajib: Tidak

CopyTagsToSnapshot

Nilai yang menunjukkan apakah akan menyalin semua tag dari instans DB ke snapshot dari instans DB. Secara default, tanda tidak disalin.

Tipe: Boolean

Wajib: Tidak

DBInstanceClass

Kapasitas komputasi dan memori baru dari instans; misalnya, `db.r5.large`. Tidak semua kelas instans tersedia di semua Wilayah AWS.

Jika Anda memodifikasi kelas instans, pemadaman terjadi selama perubahan. Perubahan diterapkan selama jendela pemeliharaan berikutnya, kecuali `ApplyImmediately` ditentukan sebagai `true` untuk permintaan ini.

Default: Menggunakan pengaturan yang ada.

Tipe: String

Wajib: Tidak

EnablePerformanceInsights

Nilai yang menunjukkan apakah akan mengaktifkan Performance Insights untuk Instans DB. Untuk informasi selengkapnya, lihat Menggunakan Amazon Performance Insights.

Tipe: Boolean

Wajib: Tidak

NewDBInstanceIdentifier

Pengidentifikasi instans baru untuk instans saat mengganti nama instans. Ketika Anda mengubah pengidentifikasi instans, boot ulang instans terjadi segera jika Anda mengatur `Apply Immediately` ke `true`. Ini terjadi selama jendela pemeliharaan berikutnya jika Anda mengatur `Apply Immediately` ke `false`. Nilai ini disimpan sebagai string huruf kecil.

Kendala:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `mydbinstance`

Tipe: String

Wajib: Tidak

PerformanceInsightsKMSKeyId

Pengenal AWS KMS kunci untuk enkripsi data Performance Insights.

Pengidentifikasi AWS KMS kunci adalah kunci ARN, ID kunci, alias ARN, atau nama alias untuk kunci KMS.

Jika Anda tidak menentukan nilai untuk PerformanceInsights KMSKeyId, maka Amazon DocumentDB menggunakan kunci KMS default Anda. Ada kunci KMS default untuk akun Amazon Web Services Anda. Akun Amazon Web Services Anda memiliki kunci KMS default yang berbeda untuk setiap wilayah Amazon Web Services.

Tipe: String

Wajib: Tidak

PreferredMaintenanceWindow

Rentang waktu mingguan (dalam UTC) untuk melakukan pemeliharaan sistem dapat terjadi, yang dapat mengakibatkan pemadaman. Mengubah parameter ini tidak mengakibatkan pemadaman kecuali dalam situasi berikut, dan perubahan diterapkan secara asinkron sesegera mungkin. Jika ada tindakan tertunda yang menyebabkan boot ulang, dan jendela pemeliharaan diubah untuk menyertakan waktu saat ini, mengubah parameter ini akan menyebabkan boot ulang instans. Jika Anda memindahkan jendela ke waktu saat ini, harus ada minimal 30 menit antara waktu saat ini dan waktu selesai jendela untuk memastikan bahwa setiap perubahan yang tertunda sudah diterapkan.

Default: Menggunakan pengaturan yang ada.

Format: `ddd:hh24:mi-ddd:hh24:mi`

Hari yang valid: Sen, Sel, Rab, Kam, Jum, Sab, Min

Kendala: Harus setidaknya 30 menit.

Tipe: String

Wajib: Tidak

PromotionTier

Nilai yang menentukan urutan di mana replika Amazon DocumentDB dipromosikan ke instans primer setelah kegagalan instans primer yang ada.

Default: 1

Nilai yang valid: 0-15

Tipe: Integer

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBInstance

Informasi mendetail tentang sebuah instans.

Tipe: Objek DBInstance

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AuthorizationNotFound

Grup keamanan CIDR IP atau Amazon EC2 yang ditentukan tidak diotorisasi untuk grup keamanan yang ditentukan.

Amazon DocumentDB juga mungkin tidak diizinkan untuk melakukan tindakan yang diperlukan atas nama Anda menggunakan IAM.

Kode Status HTTP: 404

CertificateNotFound

`CertificateIdentifier` tidak mengacu ke sertifikat yang ada.

Kode Status HTTP: 404

DBInstanceAlreadyExists

Anda sudah memiliki instans dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

DBSecurityGroupNotFound

`DBSecurityGroupName` tidak mengacu ke grup keamanan yang ada.

Kode Status HTTP: 404

DBUpgradeDependencyFailure

Peningkatan gagal karena sumber daya yang tempat bergantung tidak dapat diubah.

Kode Status HTTP: 400

InsufficientDBInstanceCapacity

Kelas instans yang ditentukan tidak tersedia di Availability Zone yang ditentukan.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

InvalidDBSecurityGroupState

Status grup keamanan tidak mengizinkan penghapusan.

Kode Status HTTP: 400

InvalidVPCNetworkStateFault

Grup subnet tidak mencakup semua Availability Zone setelah dibuat karena perubahan yang dibuat.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

StorageTypeNotSupported

Penyimpanan `StorageType` yang ditentukan tidak dapat dikaitkan dengan instans DB.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyDBSubnetGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memodifikasi grup subnet yang ada. Grup subnet harus berisi setidaknya satu subnet di setidaknya dua Availability Zone di Wilayah AWS.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBSubnetGroupName

Nama grup subnet. Nilai ini disimpan sebagai string huruf kecil. Anda tidak dapat memodifikasi grup subnet default.

Kendala: Harus cocok dengan nama `DBSubnetGroup` yang ada. Tidak harus default.

Contoh: `mySubnetgroup`

Tipe: String

Diperlukan: Ya

SubnetIds. SubnetIdentifier.N

ID subnet Amazon EC2 untuk grup subnet.

Tipe: Array string

Wajib: Ya

DBSubnetGroupDescription

Deskripsi untuk grup subnet.

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBSubnetGroup

Informasi mendetail tentang grup subnet.

Tipe: Objek DBSubnetGroup

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBSubnetGroupDoesNotCoverEnoughAZs

Subnet dalam grup subnet harus mencakup setidaknya dua Availability Zone kecuali hanya ada satu Availability Zone.

Kode Status HTTP: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

DBSubnetQuotaExceededFault

Permintaan akan menyebabkan Anda melebihi jumlah subnet yang diizinkan dalam grup subnet.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

SubnetAlreadyInUse

Subnet sudah digunakan di Availability Zone.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyEventSubscription

Layanan: Amazon DocumentDB (with MongoDB compatibility)

memodifikasi langganan notifikasi peristiwa Amazon DocumentDB yang ada.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SubscriptionName

Nama langganan notifikasi peristiwa Amazon DocumentDB.

Tipe: String

Diperlukan: Ya

Enabled

Nilai Boolean; atur ke `true` untuk mengaktifkan langganan.

Tipe: Boolean

Wajib: Tidak

EventCategories. EventCategory.N

Daftar kategori peristiwa untuk `SourceType` yang ingin Anda langgani.

Tipe: Array string

Wajib: Tidak

SnsTopicArn

Amazon Resource Name (ARN) dari topik SNS yang dibuat untuk notifikasi acara. ARN dibuat oleh Amazon SNS saat Anda membuat topik dan berlangganan topik tersebut.

Tipe: String

Wajib: Tidak

SourceType

Jenis sumber yang menghasilkan acara. Misalnya, jika Anda ingin diberi tahu tentang peristiwa yang dihasilkan oleh sebuah instans, atur parameter ini ke `db-instance`. Jika nilai ini tidak ditentukan, semua peristiwa dikembalikan.

Nilai valid: `db-instance`, `db-parameter-group`, `db-security-group`

Tipe: String

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EventSubscription

Informasi mendetail tentang peristiwa yang telah Anda langgani.

Tipe: Objek EventSubscription

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

EventSubscriptionQuotaExceeded

Anda telah mencapai jumlah maksimum langganan peristiwa.

Kode Status HTTP: 400

SNSInvalidTopic

Amazon SNS telah merespons bahwa ada masalah dengan topik yang ditentukan.

Kode Status HTTP: 400

SNSNoAuthorization

Anda tidak memiliki izin untuk memublikasikan ke Amazon Resource Name (ARN) topik SNS.

Kode Status HTTP: 400

SNSTopicArnNotFound

Amazon Resource Name (ARN) topik SNS tidak ada.

Kode Status HTTP: 404

SubscriptionCategoryNotFound

Kategori yang diberikan tidak ada.

Kode Status HTTP: 404

SubscriptionNotFound

Nama langganan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ModifyGlobalCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Modifikasi pengaturan untuk klaster global Amazon DocumentDB. Anda dapat mengubah satu atau lebih parameter konfigurasi (misalnya: perlindungan penghapusan), atau pengidentifikasi klaster global dengan menentukan parameter ini dan nilai-nilai baru dalam permintaan.

> **Note**
>
> Tindakan ini hanya berlaku untuk klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

GlobalClusterIdentifier

Pengidentifikasi untuk kluster global yang sedang dimodifikasi. Parameter ini tidak peka huruf besar kecil.

Batasan:

- Harus cocok dengan pengidentifikasi klaster global yang ada.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Wajib: Ya

DeletionProtection

Menunjukkan jika klaster global memiliki perlindungan penghapusan yang diaktifkan. Klaster global tidak dapat dihapus saat perlindungan penghapusan diaktifkan.

Tipe: Boolean

Wajib: Tidak

NewGlobalClusterIdentifier

Pengidentifikasi baru untuk klaster global saat Anda mengubah klaster global. Nilai ini disimpan sebagai string huruf kecil.

- Harus berisi 1 sampai 63 huruf, angka, atau tanda hubung

  Karakter pertama harus berupa huruf

  Tidak dapat diakhiri dengan sebuah tanda hubung atau mengandung dua tanda hubung berturut-turut

Contoh: `my-cluster2`

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Diperlukan: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

GlobalCluster

Sebuah tipe data yang mewakili sebuah klaster global Amazon DocumentDB.

Tipe: Objek GlobalCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

GlobalClusterNotFoundFault

`GlobalClusterIdentifier` tidak mengacu ke sebuah klaster global yang ada.

Kode Status HTTP: 404

InvalidGlobalClusterStateFault

Operasi yang diminta tidak dapat dilakukan saat klaster dalam keadaan ini.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

RebootDBInstance

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Anda mungkin perlu me-restart instans Anda, biasanya untuk alasan pemeliharaan. Misalnya, jika Anda melakukan perubahan tertentu, atau jika Anda mengubah grup parameter klaster yang terkait dengan instans, Anda harus mem-boot ulang instans agar perubahan diterapkan.

Menyalakan ulang instans akan memulai ulang layanan mesin basis data. Menyalakan ulang instans akan menyebabkan matinya sementara, selama status instans diatur menyalakan ulang.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBInstanceIdentifier

Pengidentifikasi instans. Parameter ini disimpan sebagai string huruf kecil.

Batas:

- Harus cocok dengan pengidentifikasi `DBInstance` yang ada.

Tipe: String

Diperlukan: Ya

ForceFailover

Saat `true`, boot ulang dilakukan melalui failover multi-AZ.

Kendala: Anda tidak dapat menentukan `true` jika instans tidak dikonfigurasi untuk Multi-AZ.

Tipe: Boolean

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBInstance

Informasi mendetail tentang sebuah instans.

Tipe: Objek DBInstance

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

RemoveFromGlobalCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Melepaskan klaster sekunder Amazon DocumentDB dari klaster global. Klaster menjadi klaster mandiri dengan kemampuan baca-tulis alih-alih menjadi hanya-baca dan menerima data dari klaster primer di wilayah yang berbeda.

> **Note**
>
> Tindakan ini hanya berlaku untuk klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DbClusterIdentifier

Amazon Resource Name (ARN) mengidentifikasi klaster yang terlepas dari klaster global Amazon DocumentDB.

Tipe: String

Diperlukan: Ya

GlobalClusterIdentifier

Pengidentifikasi klaster untuk dilepaskan dari klaster global Amazon DocumentDB.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

GlobalCluster

Sebuah tipe data yang mewakili sebuah klaster global Amazon DocumentDB.

Tipe: Objek GlobalCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

GlobalClusterNotFoundFault

`GlobalClusterIdentifier` tidak mengacu ke sebuah klaster global yang ada.

Kode Status HTTP: 404

InvalidGlobalClusterStateFault

Operasi yang diminta tidak dapat dilakukan saat klaster dalam keadaan ini.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3

- AWS SDK untuk Python
- AWS SDK for Ruby V3

RemoveSourceIdentifierFromSubscription

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus pengidentifikasi sumber dari langganan notifikasi peristiwa Amazon DocumentDB yang ada.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

SourceIdentifier

Pengidentifikasi sumber yang akan dihapus dari langganan, seperti pengidentifikasi instans untuk instans, atau nama grup keamanan.

Tipe: String

Diperlukan: Ya

SubscriptionName

Nama langganan notifikasi peristiwa Amazon DocumentDB yang ingin Anda hapus pengidentifikasi sumbernya.

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

EventSubscription

Informasi mendetail tentang peristiwa yang telah Anda langgani.

Tipe: Objek EventSubscription

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

SourceNotFound

Sumber yang diminta tidak dapat ditemukan.

Kode Status HTTP: 404

SubscriptionNotFound

Nama langganan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

RemoveTagsFromResource

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghapus tag metadata dari sumber daya Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

ResourceName

Sumber daya Amazon DocumentDB tempat tag dihapus. Nilai ini adalah sebuah Amazon Resource Name (ARN).

Tipe: String

Diperlukan: Ya

TagKeys.member.n

Kunci tag (nama) dari tag yang akan dihapus.

Tipe: Array string

Wajib: Ya

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBInstanceNotFound

`DBInstanceIdentifier` tidak mengacu ke sebuah instans yang ada.

Kode Status HTTP: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` tidak mengacu ke sebuah snapshot yang ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ResetDBClusterParameterGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memodifikasi parameter grup parameter klaster ke nilai default. Untuk mereset parameter tertentu, kirimkan daftar berikut ini: `ParameterName` dan `ApplyMethod`. Untuk mereset seluruh grup parameter cluster, tentukan parameter `DBClusterParameterGroupName` dan `ResetAllParameters`.

Ketika Anda mereset seluruh grup, parameter dinamis diperbarui segera dan parameter statis diatur ke `pending-reboot` untuk berlaku pada boot ulang instans DB berikutnya.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterParameterGroupName

Nama grup parameter klaster yang akan direset.

Tipe: String

Diperlukan: Ya

Parameter.Parameter.N

Daftar nama parameter dalam grup parameter klaster untuk direset ke nilai default. Anda tidak dapat menggunakan parameter ini jika parameter `ResetAllParameters` diatur ke `true`.

Tipe: Array objek Parameter

Wajib: Tidak

ResetAllParameters

Nilai yang diatur ke `true` untuk me-reset semua parameter dalam grup parameter klaster ke nilai defaultnya, dan `false` sebaliknya. Anda tidak dapat menggunakan parameter ini jika ada daftar nama parameter yang ditentukan untuk parameter `Parameters`.

Tipe: Boolean

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBClusterParameterGroupName

Nama sebuah grup parameter klaster.

Batasan:

- Harus dari 1 hingga 255 huruf atau angka.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

> **Note**
>
> Nilai ini disimpan sebagai string huruf kecil.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBParameterGroupNotFound

`DBParameterGroupName` tidak mengacu ke grup parameter yang ada.

Kode Status HTTP: 404

InvalidDBParameterGroupState

Grup parameter sedang digunakan, atau dalam keadaan tidak valid. Jika Anda mencoba menghapus grup parameter, Anda tidak dapat menghapusnya ketika grup parameter berada dalam keadaan ini.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

RestoreDBClusterFromSnapshot

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Membuat klaster baru dari snapshot atau snapshot klaster.

Jika snapshot ditentukan, klaster target dibuat dari snapshot DB sumber dengan konfigurasi default dan grup keamanan default.

Jika snapshot klaster ditentukan, klaster target dibuat dari titik pemulihan klaster sumber dengan konfigurasi yang sama seperti klaster DB sumber asli, kecuali klaster baru dibuat dengan grup keamanan default.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Nama klaster yang akan dibuat dari snapshot atau snapshot klaster. Parameter ini tidak peka huruf besar kecil.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-snapshot-id`

Tipe: String

Diperlukan: Ya

Engine

Mesin basis data yang akan digunakan untuk klaster baru.

Default: Sama seperti sumber.

Kendala: Harus kompatibel dengan mesin sumber.

Tipe: String

Diperlukan: Ya

SnapshotIdentifier

Pengidentifikasi untuk snapshot atau snapshot klaster yang akan dipulihkan.

Anda dapat menggunakan nama atau Amazon Resource Name (ARN) untuk menentukan snapshot klaster. Namun, Anda dapat menggunakan hanya ARN untuk menentukan snapshot.

Batas:

- Harus cocok dengan pengidentifikasi snapshot yang ada.

Tipe: String

Diperlukan: Ya

AvailabilityZones. AvailabilityZone.N

Menyediakan daftar Availability Zone Amazon EC2 tempat instans dalam klaster DB yang dipulihkan dapat dibuat.

Tipe: Array string

Wajib: Tidak

DBClusterParameterGroupName

Nama grup parameter klaster yang akan dikaitkan dengan klaster DB ini.

Tipe: String. Diperlukan: Tidak.

Jika argumen ini dihilangkan, grup parameter cluster DB default digunakan. Jika disediakan, harus cocok dengan nama grup parameter cluster DB default yang ada. String harus terdiri dari 1 hingga 255 huruf, angka atau tanda hubung. Karakter pertamanya harus berupa huruf, dan tidak dapat diakhiri dengan tanda hubung atau berisi dua tanda hubung berturut-turut.

Tipe: String

Wajib: Tidak

DBSubnetGroupName

Nama grup subnet yang akan digunakan untuk klaster baru.

Kendala: Jika disediakan, harus cocok dengan nama `DBSubnetGroup` yang ada.

Contoh: `mySubnetgroup`

Tipe: String

Wajib: Tidak

DeletionProtection

Menentukan apakah klaster ini dapat dihapus. Jika `DeletionProtection` diaktifkan, klaster tidak dapat dihapus kecuali jika diubah dan `DeletionProtection` dinonaktifkan. `DeletionProtection` melindungi klaster agar tidak terhapus secara tidak sengaja.

Tipe: Boolean

Wajib: Tidak

EnableCloudwatchLogsExports.member.n

Daftar jenis log yang harus diaktifkan untuk mengekspor ke Amazon CloudWatch Logs.

Tipe: Array string

Wajib: Tidak

EngineVersion

Versi mesin basis data yang akan digunakan untuk klaster baru.

Tipe: String

Wajib: Tidak

KmsKeyId

Pengidentifikasi AWS KMS kunci yang akan digunakan saat memulihkan cluster terenkripsi dari snapshot DB atau snapshot cluster.

Pengidentifikasi AWS KMS kunci adalah Amazon Resource Name (ARN) untuk kunci enkripsi AWS KMS . Jika Anda memulihkan cluster dengan yang sama Akun AWS yang memiliki kunci AWS KMS enkripsi yang digunakan untuk mengenkripsi cluster baru, maka Anda dapat menggunakan alias AWS KMS kunci alih-alih ARN untuk kunci enkripsi. AWS KMS

Jika Anda tidak menentukan nilai untuk parameter `KmsKeyId`, maka hal berikut akan terjadi:

- Jika snapshot atau snapshot cluster `SnapshotIdentifier` dienkripsi, maka cluster yang dipulihkan dienkripsi menggunakan AWS KMS kunci yang digunakan untuk mengenkripsi snapshot atau snapshot cluster.

- Jika snapshot atau snapshot klaster di `SnapshotIdentifier` tidak dienkripsi, maka klaster DB yang dipulihkan tidak dienkripsi.

Tipe: String

Wajib: Tidak

Port

Nomor port tempat klaster baru menerima koneksi.

Kendala: Harus berupa nilai dari `1150` hingga `65535`.

Default: Port yang sama dengan klaster asli.

Tipe: Integer

Wajib: Tidak

StorageType

Jenis penyimpanan untuk dikaitkan dengan cluster DB.

Untuk informasi tentang jenis penyimpanan untuk klaster Amazon DocumentDB, lihat Konfigurasi penyimpanan klaster di Panduan Pengembang Amazon DocumentDB.

Nilai yang valid untuk jenis penyimpanan - `standard | iopt1`

Nilai default adalah `standard`

Tipe: String

Wajib: Tidak

Tags.Tag.N

Tag yang akan ditetapkan ke klaster yang dipulihkan.

Tipe: Array objek Tag

Wajib: Tidak

VpcSecurityGroupIds. VpcSecurityGroupId.N

Daftar grup keamanan virtual private cloud (VPC) tempat klaster baru akan berada.

Tipe: Array string

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterAlreadyExistsFault

Anda sudah memiliki klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBClusterQuotaExceededFault

Klaster tidak dapat dibuat karena Anda telah mencapai kuota maksimum klaster yang diizinkan.

Kode Status HTTP: 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` tidak mengacu ke sebuah snapshot yang ada.

Kode Status HTTP: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

InsufficientDBClusterCapacityFault

Klaster tidak memiliki kapasitas yang cukup untuk operasi saat ini.

Kode Status HTTP: 403

InsufficientStorageClusterCapacity

Tidak ada cukup penyimpanan yang tersedia untuk tindakan saat ini. Anda mungkin dapat mengatasi kesalahan ini dengan memperbarui grup subnet Anda untuk menggunakan Availability Zone berbeda yang memiliki lebih banyak penyimpanan yang tersedia.

Kode Status HTTP: 400

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

InvalidDBSnapshotState

Status snapshot tidak mengizinkan penghapusan.

Kode Status HTTP: 400

InvalidRestoreFault

Anda tidak dapat memulihkan dari backup virtual private cloud (VPC) ke instans DB non-VPC.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

InvalidVPCNetworkStateFault

Grup subnet tidak mencakup semua Availability Zone setelah dibuat karena perubahan yang dibuat.

Kode Status HTTP: 400

KMSKeyNotAccessibleFault

Terjadi kesalahan saat mengakses AWS KMS kunci.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

RestoreDBClusterToPointInTime

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Memulihkan klaster ke titik waktu yang arbitrer. Pengguna dapat memulihkan ke titik waktu mana pun sebelum `LatestRestorableTime` hingga `BackupRetentionPeriod` hari. Klaster target dibuat dari klaster sumber dengan konfigurasi yang sama seperti klaster asli, kecuali klaster baru dibuat dengan grup keamanan default.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Nama klaster baru yang akan dibuat.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Tipe: String

Diperlukan: Ya

SourceDBClusterIdentifier

Pengidentifikasi klaster sumber yang akan dipulihkan.

Batasan:

- Harus cocok dengan pengidentifikasi `DBCluster` yang ada.

Tipe: String

Diperlukan: Ya

DBSubnetGroupName

Nama grup subnet yang akan digunakan untuk klaster baru.

Kendala: Jika disediakan, harus cocok dengan nama `DBSubnetGroup` yang ada.

Contoh: `mySubnetgroup`

Tipe: String

Wajib: Tidak

DeletionProtection

Menentukan apakah klaster ini dapat dihapus. Jika `DeletionProtection` diaktifkan, klaster tidak dapat dihapus kecuali jika diubah dan `DeletionProtection` dinonaktifkan. `DeletionProtection` melindungi klaster agar tidak terhapus secara tidak sengaja.

Tipe: Boolean

Wajib: Tidak

EnableCloudwatchLogsExports.member.n

Daftar jenis log yang harus diaktifkan untuk mengekspor ke Amazon CloudWatch Logs.

Tipe: Array string

Wajib: Tidak

KmsKeyId

Pengidentifikasi AWS KMS kunci yang akan digunakan saat memulihkan cluster terenkripsi dari cluster terenkripsi.

Pengidentifikasi AWS KMS kunci adalah Amazon Resource Name (ARN) untuk kunci enkripsi AWS KMS . Jika Anda memulihkan cluster dengan yang sama Akun AWS yang memiliki kunci AWS KMS enkripsi yang digunakan untuk mengenkripsi cluster baru, maka Anda dapat menggunakan alias AWS KMS kunci alih-alih ARN untuk kunci enkripsi. AWS KMS

Anda dapat mengembalikan ke cluster baru dan mengenkripsi cluster baru dengan AWS KMS kunci yang berbeda dari AWS KMS kunci yang digunakan untuk mengenkripsi cluster sumber. Cluster DB baru dienkripsi dengan AWS KMS kunci yang diidentifikasi oleh parameter. `KmsKeyId`

Jika Anda tidak menentukan nilai untuk parameter `KmsKeyId`, maka hal berikut akan terjadi:

- Jika cluster dienkripsi, maka cluster yang dipulihkan dienkripsi menggunakan AWS KMS kunci yang digunakan untuk mengenkripsi cluster sumber.
- Jika klaster tidak dienkripsi, maka klaster yang dipulihkan tidak dienkripsi.

Jika `DBClusterIdentifier` mengacu pada sebuah klaster yang tidak dienkripsi, maka permintaan pemulihan ditolak.

Tipe: String

Wajib: Tidak

Port

Nomor port tempat klaster baru menerima koneksi.

Kendala: Harus berupa nilai dari `1150` hingga `65535`.

Default: Port default untuk mesin.

Tipe: Integer

Wajib: Tidak

RestoreToTime

Tanggal dan waktu untuk memulihkan klaster.

Nilai yang valid: Waktu dalam format Waktu Universal Terkoordinasi (UTC).

Batasan:

- Harus sebelum waktu pemulihan terbaru untuk instans.
- Harus ditentukan jika parameter `UseLatestRestorableTime` tidak disediakan.
- Tidak dapat ditentukan jika parameter `UseLatestRestorableTime` adalah `true`.
- Tidak dapat ditentukan jika parameter `RestoreType` adalah `copy-on-write`.

Contoh: `2015-03-07T23:45:00Z`

Tipe: Timestamp

Wajib: Tidak

RestoreType

Tipe pemulihan yang akan dilakukan. Anda dapat menentukan salah satu nilai berikut:

- `full-copy` - Klaster DB baru dipulihkan sebagai salinan lengkap dari klaster DB sumber.
- `copy-on-write` - Klaster DB baru dipulihkan sebagai klon dari klaster DB sumber.

Kendala: Anda tidak dapat menentukan `copy-on-write` jika versi mesin klaster DB sumber lebih awal dari 1,11.

Jika Anda tidak menentukan `RestoreType` nilai, maka klaster DB baru dipulihkan sebagai salinan lengkap klaster DB sumber.

Tipe: String

Wajib: Tidak

StorageType

Jenis penyimpanan untuk mengasosiasikan dengan cluster DB.

Untuk informasi tentang jenis penyimpanan untuk klaster Amazon DocumentDB, lihat Konfigurasi penyimpanan klaster di Panduan Pengembang Amazon DocumentDB.

Nilai yang valid untuk jenis penyimpanan - `standard | iopt1`

Nilai default adalah `standard`

Tipe: String

Wajib: Tidak

Tags.Tag.N

Tag yang akan ditetapkan ke klaster yang dipulihkan.

Tipe: Array objek Tag

Wajib: Tidak

UseLatestRestorableTime

Nilai yang diatur ke `true` untuk memulihkan klaster ke waktu backup terbaru yang dapat dipulihkan, dan `false` sebaliknya.

Default: `false`

Kendala: Tidak dapat ditentukan jika parameter `RestoreToTime` disediakan.

Tipe: Boolean

Wajib: Tidak

VpcSecurityGroupIds. VpcSecurityGroupId.N

Daftar grup keamanan VPC tempat klaster baru berada.

Tipe: Array string

Wajib: Tidak

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterAlreadyExistsFault

Anda sudah memiliki klaster dengan pengidentifikasi yang diberikan.

Kode Status HTTP: 400

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

DBClusterQuotaExceededFault

Klaster tidak dapat dibuat karena Anda telah mencapai kuota maksimum klaster yang diizinkan.

Kode Status HTTP: 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` tidak mengacu ke sebuah snapshot klaster yang ada.

Kode Status HTTP: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` tidak mengacu ke grup subnet yang ada.

Kode Status HTTP: 404

InsufficientDBClusterCapacityFault

Klaster tidak memiliki kapasitas yang cukup untuk operasi saat ini.

Kode Status HTTP: 403

InsufficientStorageClusterCapacity

Tidak ada cukup penyimpanan yang tersedia untuk tindakan saat ini. Anda mungkin dapat mengatasi kesalahan ini dengan memperbarui grup subnet Anda untuk menggunakan Availability Zone berbeda yang memiliki lebih banyak penyimpanan yang tersedia.

Kode Status HTTP: 400

InvalidDBClusterSnapshotStateFault

Nilai yang diberikan bukan status snapshot klaster yang valid.

Kode Status HTTP: 400

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBSnapshotState

Status snapshot tidak mengizinkan penghapusan.

Kode Status HTTP: 400

InvalidRestoreFault

Anda tidak dapat memulihkan dari backup virtual private cloud (VPC) ke instans DB non-VPC.

Kode Status HTTP: 400

InvalidSubnet

Subnet yang diminta tidak valid, atau beberapa subnet yang diminta tidak semuanya berada dalam virtual private cloud (VPC) yang umum.

Kode Status HTTP: 400

InvalidVPCNetworkStateFault

Grup subnet tidak mencakup semua Availability Zone setelah dibuat karena perubahan yang dibuat.

Kode Status HTTP: 400

KMSKeyNotAccessibleFault

Terjadi kesalahan saat mengakses AWS KMS kunci.

Kode Status HTTP: 400

StorageQuotaExceeded

Permintaan akan menyebabkan Anda melebihi jumlah penyimpanan yang diizinkan yang tersedia di semua instans.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

StartDBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mulai ulang klaster yang terhenti yang ditentukan oleh `DBClusterIdentifier`. Untuk informasi selengkapnya, lihat Penghentian dan Pemulaian Klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi dari klaster untuk dimulai ulang. Contoh: `docdb-2019-05-28-15-24-52`

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

StopDBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menghentikan klaster yang sedang berjalan yang ditentukan oleh `DBClusterIdentifier`. Klaster harus dalam status tersedia. Untuk informasi selengkapnya, lihat Penghentian dan Pemulaian Klaster Amazon DocumentDB.

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

DBClusterIdentifier

Pengidentifikasi klaster untuk dihentikan. Contoh: `docdb-2019-05-28-15-24-52`

Tipe: String

Diperlukan: Ya

Elemen Respons

Elemen berikut dikembalikan oleh layanan.

DBCluster

Informasi mendetail tentang klaster.

Tipe: Objek DBCluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

DBClusterNotFoundFault

`DBClusterIdentifier` tidak mengacu ke sebuah klaster yang ada.

Kode Status HTTP: 404

InvalidDBClusterStateFault

Klaster tidak dalam status valid.

Kode Status HTTP: 400

InvalidDBInstanceState

Instans yang ditentukan tidak dalam status tersedia.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

Cluster Elastis Amazon DocumentDB

Tindakan berikut didukung oleh Amazon DocumentDB Elastic Clusters:

- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster
- GetClusterSnapshot
- ListClusters

- ListClusterSnapshots
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster
- TagResource
- UntagResource
- UpdateCluster

CopyClusterSnapshot

Layanan: Amazon DocumentDB Elastic Clusters

Menyalin snapshot dari cluster elastis.

Minta Sintaks

```
POST /cluster-snapshot/snapshotArn/copy HTTP/1.1
Content-type: application/json

{
   "copyTags": boolean,
   "kmsKeyId": "string",
   "tags": {
      "string" : "string"
   },
   "targetSnapshotName": "string"
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

snapshotArn

Pengidentifikasi Amazon Resource Name (ARN) dari snapshot cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

targetSnapshotName

Pengidentifikasi snapshot cluster elastis baru untuk dibuat dari snapshot cluster sumber. Parameter ini tidak peka huruf besar kecil.

Batas:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.

- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `elastic-cluster-snapshot-5`

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 63.

Wajib: Ya

copyTags

Setel `true` untuk menyalin semua tag dari snapshot cluster sumber ke snapshot cluster elastis target. Defaultnya adalah `false`.

Tipe: Boolean

Wajib: Tidak

kmsKeyId

ID kunci AWS KMS untuk snapshot cluster elastis terenkripsi. ID kunci AWS KMS adalah Nama Sumber Daya Amazon (ARN) AWS , pengidentifikasi kunci KMS, atau alias kunci KMS AWS untuk kunci enkripsi KMS. AWS

Jika Anda menyalin snapshot cluster elastis terenkripsi dari AWS akun Anda, Anda dapat menentukan nilai `KmsKeyId` untuk mengenkripsi salinan dengan kunci enkripsi AWS S KMS baru. Jika Anda tidak menentukan nilai untuk`KmsKeyId`, maka salinan snapshot cluster elastis dienkripsi dengan kunci `AWS` KMS yang sama dengan snapshot cluster elastis sumber.

Untuk menyalin snapshot cluster elastis terenkripsi ke AWS wilayah lain, atur `KmsKeyId` ke ID kunci AWS KMS yang ingin Anda gunakan untuk mengenkripsi salinan snapshot cluster elastis di wilayah tujuan. AWS Kunci enkripsi KMS khusus untuk AWS wilayah tempat mereka dibuat, dan Anda tidak dapat menggunakan kunci enkripsi dari satu AWS wilayah di AWS wilayah lain.

Jika Anda menyalin snapshot cluster elastis yang tidak terenkripsi dan menentukan nilai untuk `KmsKeyId` parameter, kesalahan akan dikembalikan.

Tipe: String

Wajib: Tidak

tags

Tag yang akan ditetapkan ke snapshot cluster elastis.

Tipe: Peta string ke string

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Wajib: Tidak

Sintaksis Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

snapshot

Mengembalikan informasi tentang snapshot cluster elastis tertentu.

Tipe: Objek ClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ServiceQuotaExceededException

Kuota layanan untuk tindakan terlampaui.

Kode Status HTTP: 402

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateCluster

Layanan: Amazon DocumentDB Elastic Clusters

Membuat cluster elastis Amazon DocumentDB baru dan mengembalikan struktur klasternya.

Minta Sintaks

```
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" ]
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

adminUserName

Nama administrator cluster elastis Amazon DocumentDB.

Kendala:

- Harus dari 1 hingga 63 huruf atau angka.
- Karakter pertama harus berupa huruf.
- Tidak dapat berupa kata yang dicadangkan.

Tipe: String

Diperlukan: Ya

adminUserPassword

Kata sandi untuk administrator cluster elastis Amazon DocumentDB. Kata sandi dapat berisi karakter ASCII yang dapat dicetak.

Kendala:

- Harus berisi 8 hingga 100 karakter.
- Tidak dapat berisi garis miring (/), tanda kutip ganda (“), atau simbol “at” (@).

Tipe: String

Diperlukan: Ya

authType

Jenis otentikasi yang digunakan untuk menentukan di mana untuk mengambil password yang digunakan untuk mengakses cluster elastis. Jenis yang valid adalah `PLAIN_TEXT` atau`SECRET_ARN`.

Jenis: String

Nilai yang Valid: `PLAIN_TEXT | SECRET_ARN`

Wajib: Ya

clusterName

Nama cluster elastis baru. Parameter ini disimpan sebagai string huruf kecil.

Kendala:

- Harus berisi 1 hingga 63 huruf, angka, atau tanda hubung.
- Karakter pertama harus berupa huruf.
- Tidak dapat diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Contoh: `my-cluster`

Tipe: String

Diperlukan: Ya

shardCapacity

Jumlah vCPU yang ditetapkan untuk setiap pecahan cluster elastis. Maksimal adalah 64. Nilai yang diizinkan adalah 2, 4, 8, 16, 32, 64.

Jenis: Integer

Wajib: Ya

shardCount

Jumlah pecahan yang ditugaskan ke cluster elastis. Maksimal adalah 32.

Jenis: Integer

Wajib: Ya

backupRetentionPeriod

Jumlah hari di mana snapshot otomatis dipertahankan.

Tipe: Integer

Wajib: Tidak

clientToken

Token klien untuk cluster elastis.

Tipe: String

Wajib: Tidak

kmsKeyId

Pengidentifikasi kunci KMS yang digunakan untuk mengenkripsi cluster elastis baru.

Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN) untuk kunci enkripsi KMS. Jika Anda membuat cluster menggunakan akun Amazon yang sama yang memiliki kunci enkripsi KMS ini, Anda dapat menggunakan alias kunci KMS alih-alih ARN sebagai kunci enkripsi KMS.

Jika kunci enkripsi tidak ditentukan, Amazon DocumentDB menggunakan kunci enkripsi default yang dibuat KMS untuk akun Anda. Akun Anda memiliki kunci enkripsi default yang berbeda untuk setiap Wilayah Amazon.

Tipe: String

Wajib: Tidak

preferredBackupWindow

Rentang waktu harian di mana pencadangan otomatis dibuat jika pencadangan otomatis diaktifkan, sebagaimana ditentukan oleh. `backupRetentionPeriod`

Tipe: String

Wajib: Tidak

preferredMaintenanceWindow

Rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, dalam Waktu Universal Terkoordinasi (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Default: jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing Wilayah AWS, terjadi pada hari acak dalam seminggu.

Hari yang berlaku: Senin, Sel, Rabu, Kam, Jumat, Sabtu, Minggu

Kendala: Jendela minimum 30 menit.

Tipe: String

Wajib: Tidak

shardInstanceCount

Jumlah contoh replika yang berlaku untuk semua pecahan di cluster elastis. `shardInstanceCount`Nilai 1 berarti ada satu contoh penulis, dan setiap contoh tambahan adalah replika yang dapat digunakan untuk membaca dan untuk meningkatkan ketersediaan.

Tipe: Integer

Wajib: Tidak

subnetIds

ID subnet Amazon EC2 untuk cluster elastis baru.

Tipe: Array string

Wajib: Tidak

tags

Tag yang akan ditetapkan ke cluster elastis baru.

Tipe: Peta string ke string

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Wajib: Tidak

vpcSecurityGroupIds

Daftar grup keamanan VPC EC2 untuk diasosiasikan dengan cluster elastis baru.

Tipe: Array string

Wajib: Tidak

Sintaksis Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Cluster elastis baru yang telah dibuat.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ServiceQuotaExceededException

Kuota layanan untuk tindakan terlampaui.

Kode Status HTTP: 402

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateClusterSnapshot

Layanan: Amazon DocumentDB Elastic Clusters

Membuat snapshot dari cluster elastis.

Minta Sintaks

```
POST /cluster-snapshot HTTP/1.1
Content-type: application/json

{
   "clusterArn": "string",
   "snapshotName": "string",
   "tags": {
      "string" : "string"
   }
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

clusterArn

Pengidentifikasi ARN dari cluster elastis yang ingin Anda buat snapshot.

Tipe: String

Diperlukan: Ya

snapshotName

Nama snapshot cluster elastis baru.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 63.

Wajib: Ya

tags

Tag yang akan ditetapkan ke snapshot cluster elastis baru.

Tipe: Peta string ke string

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Wajib: Tidak

Sintaksis Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

snapshot

Mengembalikan informasi tentang snapshot cluster elastis baru.

Tipe: Objek ClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ServiceQuotaExceededException

Kuota layanan untuk tindakan terlampaui.

Kode Status HTTP: 402

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteCluster

Layanan: Amazon DocumentDB Elastic Clusters

Hapus cluster elastis.

Minta Sintaks

```
DELETE /cluster/clusterArn HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

clusterArn

Pengidentifikasi ARN dari cluster elastis yang akan dihapus.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Mengembalikan informasi tentang cluster elastis yang baru dihapus.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteClusterSnapshot

Layanan: Amazon DocumentDB Elastic Clusters

Hapus snapshot cluster elastis.

Minta Sintaks

```
DELETE /cluster-snapshot/snapshotArn HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

snapshotArn

Pengidentifikasi ARN dari snapshot cluster elastis yang akan dihapus.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
   }
```

```
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

snapshot

Mengembalikan informasi tentang snapshot cluster elastis yang baru dihapus.

Tipe: Objek ClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

GetCluster

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan informasi tentang cluster elastis tertentu.

Minta Sintaks

```
GET /cluster/clusterArn HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Mengembalikan informasi tentang cluster elastis tertentu.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

GetClusterSnapshot

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan informasi tentang snapshot cluster elastis tertentu

Minta Sintaks

```
GET /cluster-snapshot/snapshotArn HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

snapshotArn

Pengidentifikasi ARN dari snapshot cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
   }
```

```
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

snapshot

Mengembalikan informasi tentang snapshot cluster elastis tertentu.

Tipe: Objek ClusterSnapshot

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ListClusters

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan informasi tentang klaster elastis Amazon DocumentDB yang disediakan.

Minta Sintaks

```
GET /clusters?maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

maxResults

Jumlah maksimum hasil snapshot cluster elastis yang diterima dalam respons.

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 100.

nextToken

Token pagination yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya mencakup catatan di luar token ini, hingga nilai yang ditentukan oleh`max-results`.

Jika tidak ada lagi data dalam respons, tidak `nextToken` akan dikembalikan.

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
   "clusters": [
      {
         "clusterArn": "string",
         "clusterName": "string",
         "status": "string"
      }
```

```
    ],
    "nextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

clusters

Daftar cluster elastis Amazon DocumentDB.

Tipe: Array objek ClusterInList

nextToken

Token pagination yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya mencakup catatan di luar token ini, hingga nilai yang ditentukan oleh`max-results`.

Jika tidak ada lagi data dalam respons, tidak `nextToken` akan dikembalikan.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ListClusterSnapshots

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan informasi tentang snapshot untuk cluster elastis tertentu.

Minta Sintaks

```
GET /cluster-snapshots?
clusterArn=clusterArn&maxResults=maxResults&nextToken=nextToken&snapshotType=snapshotType
 HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

maxResults

Jumlah maksimum hasil snapshot cluster elastis yang diterima dalam respons.

Rentang Valid: Nilai minimum 20. Nilai maksimum 100.

nextToken

Token pagination yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya mencakup catatan di luar token ini, hingga nilai yang ditentukan oleh`max-results`.

Jika tidak ada lagi data dalam respons, tidak `nextToken` akan dikembalikan.

snapshotType

Jenis snapshot klaster untuk dikembalikan. Anda dapat menentukan salah satu nilai berikut:

- `automated`- Kembalikan semua snapshot cluster yang Amazon DocumentDB telah secara otomatis dibuat untuk akun Anda. AWS
- `manual`- Kembalikan semua snapshot cluster yang telah Anda buat secara manual untuk AWS akun Anda.

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "snapshots": [
      {
         "clusterArn": "string",
         "snapshotArn": "string",
         "snapshotCreationTime": "string",
         "snapshotName": "string",
         "status": "string"
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

nextToken

Token pagination yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya mencakup catatan di luar token ini, hingga nilai yang ditentukan oleh`max-results`.

Jika tidak ada lagi data dalam respons, tidak `nextToken` akan dikembalikan.

Jenis: String

snapshots

Daftar snapshot untuk cluster elastis tertentu.

Tipe: Array objek ClusterSnapshotInList

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ListTagsForResource

Layanan: Amazon DocumentDB Elastic Clusters

Daftar semua tag pada sumber daya cluster elastis

Minta Sintaks

```
GET /tags/resourceArn HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

resourceArn

Pengidentifikasi ARN dari sumber daya cluster elastis.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1011.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

tags

Daftar tag untuk sumber daya cluster elastis yang ditentukan.

Tipe: Peta string ke string

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

RestoreClusterFromSnapshot

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan cluster elastis dari snapshot.

Minta Sintaks

```
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" ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

snapshotArn

Pengidentifikasi ARN dari snapshot cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

clusterName

Nama cluster elastis.

Tipe: String

Diperlukan: Ya

kmsKeyId

Pengidentifikasi kunci KMS yang digunakan untuk mengenkripsi cluster elastis Amazon DocumentDB baru.

Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN) untuk kunci enkripsi KMS. Jika Anda membuat cluster menggunakan akun Amazon yang sama yang memiliki kunci enkripsi KMS ini, Anda dapat menggunakan alias kunci KMS alih-alih ARN sebagai kunci enkripsi KMS.

Jika kunci enkripsi tidak ditentukan di sini, Amazon DocumentDB menggunakan kunci enkripsi default yang dibuat KMS untuk akun Anda. Akun Anda memiliki kunci enkripsi default yang berbeda untuk setiap Wilayah Amazon.

Tipe: String

Wajib: Tidak

shardCapacity

Kapasitas setiap pecahan dalam cluster elastis baru yang dipulihkan.

Tipe: Integer

Wajib: Tidak

shardInstanceCount

Jumlah contoh replika yang berlaku untuk semua pecahan di cluster elastis. `shardInstanceCount`Nilai 1 berarti ada satu contoh penulis, dan setiap contoh tambahan adalah replika yang dapat digunakan untuk membaca dan untuk meningkatkan ketersediaan.

Tipe: Integer

Wajib: Tidak

subnetIds

ID subnet Amazon EC2 untuk cluster elastis.

Tipe: Array string

Wajib: Tidak

tags

Daftar nama tag yang akan ditetapkan ke cluster elastis yang dipulihkan, dalam bentuk array pasangan kunci-nilai di mana kuncinya adalah nama tag dan nilainya adalah nilai kunci.

Tipe: Peta string ke string

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Wajib: Tidak

vpcSecurityGroupIds

Daftar grup keamanan VPC EC2 untuk diasosiasikan dengan cluster elastis.

Tipe: Array string

Wajib: Tidak

Sintaksis Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Mengembalikan informasi tentang cluster elastis dipulihkan.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ServiceQuotaExceededException

Kuota layanan untuk tindakan terlampaui.

Kode Status HTTP: 402

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

StartCluster

Layanan: Amazon DocumentDB Elastic Clusters

Memulai ulang cluster elastis berhenti yang ditentukan oleh`clusterARN`.

Minta Sintaks

```
POST /cluster/clusterArn/start HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Mengembalikan informasi tentang cluster elastis tertentu.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

StopCluster

Layanan: Amazon DocumentDB Elastic Clusters

Menghentikan cluster elastis yang berjalan yang ditentukan oleh`clusterArn`. Cluster elastis harus dalam keadaan tersedia.

Minta Sintaks

```
POST /cluster/clusterArn/stop HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Mengembalikan informasi tentang cluster elastis tertentu.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

TagResource

Layanan: Amazon DocumentDB Elastic Clusters

Menambahkan tag metadata ke sumber daya cluster elastis

Minta Sintaks

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

resourceArn

Pengidentifikasi ARN dari sumber daya cluster elastis.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1011.

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

tags

Tag yang ditetapkan ke sumber daya cluster elastis.

Tipe: Peta string ke string

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

UntagResource

Layanan: Amazon DocumentDB Elastic Clusters

Menghapus tag metadata dari sumber daya cluster elastis

Minta Sintaks

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

resourceArn

Pengidentifikasi ARN dari sumber daya cluster elastis.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1011.

Wajib: Ya

tagKeys

Kunci tag yang akan dihapus dari sumber daya cluster elastis.

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 50 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python

- AWS SDK for Ruby V3

UpdateCluster

Layanan: Amazon DocumentDB Elastic Clusters

Memodifikasi cluster elastis. Ini termasuk memperbarui admin-username/password, memutakhirkan versi API, dan menyiapkan jendela cadangan dan jendela pemeliharaan

Minta Sintaks

```
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" ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

adminUserPassword

Kata sandi yang terkait dengan administrator cluster elastis. Kata sandi ini dapat berisi karakter ASCII apa pun yang dapat dicetak kecuali garis miring (/), tanda kutip ganda ("), atau simbol "at" (@).

Kendala: Harus berisi dari 8 hingga 100 karakter.

Tipe: String

Wajib: Tidak

authType

Jenis otentikasi yang digunakan untuk menentukan di mana untuk mengambil password yang digunakan untuk mengakses cluster elastis. Jenis yang valid adalah `PLAIN_TEXT` atau`SECRET_ARN`.

Jenis: String

Nilai yang Valid: `PLAIN_TEXT | SECRET_ARN`

Wajib: Tidak

backupRetentionPeriod

Jumlah hari di mana snapshot otomatis dipertahankan.

Tipe: Integer

Wajib: Tidak

clientToken

Token klien untuk cluster elastis.

Tipe: String

Wajib: Tidak

preferredBackupWindow

Rentang waktu harian di mana pencadangan otomatis dibuat jika pencadangan otomatis diaktifkan, sebagaimana ditentukan oleh. `backupRetentionPeriod`

Tipe: String

Wajib: Tidak

preferredMaintenanceWindow

Rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, dalam Waktu Universal Terkoordinasi (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Default: jendela 30 menit yang dipilih secara acak dari blok waktu 8 jam untuk masing-masing Wilayah AWS, terjadi pada hari acak dalam seminggu.

Hari yang berlaku: Senin, Sel, Rabu, Kam, Jumat, Sabtu, Minggu

Kendala: Jendela minimum 30 menit.

Tipe: String

Wajib: Tidak

shardCapacity

Jumlah vCPU yang ditetapkan untuk setiap pecahan cluster elastis. Maksimal adalah 64. Nilai yang diizinkan adalah 2, 4, 8, 16, 32, 64.

Tipe: Integer

Wajib: Tidak

shardCount

Jumlah pecahan yang ditugaskan ke cluster elastis. Maksimal adalah 32.

Tipe: Integer

Wajib: Tidak

shardInstanceCount

Jumlah contoh replika yang berlaku untuk semua pecahan di cluster elastis. `shardInstanceCount`Nilai 1 berarti ada satu contoh penulis, dan setiap contoh tambahan adalah replika yang dapat digunakan untuk membaca dan untuk meningkatkan ketersediaan.

Tipe: Integer

Wajib: Tidak

subnetIds

ID subnet Amazon EC2 untuk cluster elastis.

Tipe: Array string

Wajib: Tidak

vpcSecurityGroupIds

Daftar grup keamanan VPC EC2 untuk diasosiasikan dengan cluster elastis.

Tipe: Array string

Wajib: Tidak

Sintaksis Respons

```
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" ]
    }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

cluster

Mengembalikan informasi tentang cluster elastis diperbarui.

Tipe: Objek Cluster

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Pengecualian yang terjadi ketika tidak ada izin yang cukup untuk melakukan tindakan.

Kode Status HTTP: 403

ConflictException

Terjadi konflik akses.

Kode Status HTTP: 409

InternalServerException

Ada kesalahan server internal.

Kode Status HTTP: 500

ResourceNotFoundException

Sumber daya yang ditentukan tidak dapat ditemukan.

Kode Status HTTP: 404

ThrottlingException

ThrottlingException akan dilemparkan ketika permintaan ditolak karena pembatasan permintaan.

Kode Status HTTP: 429

ValidationException

Struktur yang mendefinisikan pengecualian validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

Tipe Data

Tipe data berikut didukung oleh Amazon DocumentDB (with MongoDB compatibility):

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- DBCluster

- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

Tipe data berikut didukung oleh Amazon DocumentDB Elastic Clusters:

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- Shard
- ValidationExceptionField

Amazon DocumentDB (with MongoDB compatibility)

Tipe data berikut didukung olehAmazon DocumentDB (with MongoDB compatibility):

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter

- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

AvailabilityZone

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi tentang Availability Zone.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Name

Nama Availability Zone.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Certificate

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Sertifikat otoritas sertifikat (CA) untuk Akun AWS.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

CertificateArn

Amazon Resource Name (ARN) untuk sertifikat.

Contoh: `arn:aws:rds:us-east-1::cert:rds-ca-2019`

Tipe: String

Wajib: Tidak

CertificateIdentifier

Kunci unik yang mengidentifikasi sertifikat.

Contoh: `rds-ca-2019`

Tipe: String

Wajib: Tidak

CertificateType

Jenis dari sertifikat.

Contoh: `CA`

Tipe: String

Wajib: Tidak

Thumbprint

Cap jempol dari sertifikat.

Tipe: String

Wajib: Tidak

ValidFrom

Tanggal-waktu mulai yang mana sertifikat valid.

Contoh: `2019-07-31T17:57:09Z`

Tipe: Timestamp

Wajib: Tidak

ValidTill

Tanggal-waktu setelah sertifikat tidak valid lagi.

Contoh: `2024-07-31T17:57:09Z`

Tipe: Timestamp

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CertificateDetails

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Mengembalikan rincian sertifikat server instans DB.

Untuk informasi selengkapnya, lihat Memperbarui Sertifikat Amazon DocumentDB TLS Anda dan Mengenkripsi Data dalam Transit di Panduan Pengembang Amazon DocumentDB.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

CAIdentifier

Pengidentifikasi CA dari sertifikat CA yang digunakan untuk sertifikat server instans DB.

Tipe: String

Wajib: Tidak

ValidTill

Tanggal kedaluwarsa sertifikat server instans DB.

Tipe: Timestamp

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CloudwatchLogsExportConfiguration

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Pengaturan konfigurasi untuk jenis log yang akan diaktifkan untuk diekspor ke Amazon CloudWatch Logs untuk instance atau klaster tertentu.

`DisableLogTypes`Array `EnableLogTypes` dan menentukan log mana yang diekspor (atau tidak diekspor) ke Log. CloudWatch Nilai-nilai dalam array ini tergantung pada mesin yang sedang digunakan.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DisableLogTypes.member.N

Daftar jenis log yang akan dinonaktifkan.

Tipe: Array string

Wajib: Tidak

EnableLogTypes.member.N

Daftar jenis log yang akan diaktifkan.

Tipe: Array string

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang klaster.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AssociatedRoles.DBClusterRole.N

Menyediakan daftar peran AWS Identity and Access Management (IAM) yang terkait dengan cluster. (IAM) peran yang terkait dengan izin pemberian klaster untuk klaster untuk mengakses AWS layanan lain atas nama Anda.

Tipe: Array objek DBClusterRole

Wajib: Tidak

AvailabilityZones.AvailabilityZone.N

Menyediakan daftar Availability Zone Amazon EC2 tempat instans dalam klaster dapat dibuat.

Tipe: Array string

Wajib: Tidak

BackupRetentionPeriod

Menentukan jumlah hari yang mana snapshot otomatis dipertahankan.

Tipe: Integer

Wajib: Tidak

CloneGroupId

Mengidentifikasi grup klon yang terkait dengan klaster DB.

Tipe: String

Wajib: Tidak

ClusterCreateTime

Menentukan waktu saat klaster dibuat, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: Timestamp

Wajib: Tidak

DBClusterArn

Amazon Resource Name (ARN) untuk klaster.

Tipe: String

Wajib: Tidak

DBClusterIdentifier

Berisi pengidentifikasi klaster yang disediakan pengguna. Pengidentifikasi ini adalah kunci unik yang mengidentifikasi sebuah klaster.

Tipe: String

Wajib: Tidak

DBClusterMembers.DBClusterMember.N

Menyediakan daftar instans yang membentuk klaster.

Tipe: Array objek DBClusterMember

Wajib: Tidak

DBClusterParameterGroup

Menentukan nama grup parameter klaster untuk klaster.

Tipe: String

Wajib: Tidak

DbClusterResourceId

Pengidentifikasi Wilayah AWS-unik dan tidak dapat diubah untuk cluster. Pengenal ini ditemukan di entri AWS CloudTrail log setiap kali AWS KMS kunci untuk cluster diakses.

Tipe: String

Wajib: Tidak

DBSubnetGroup

Menentukan informasi tentang grup subnet yang terkait dengan klaster, termasuk nama, deskripsi, dan subnet dalam grup subnet.

Tipe: String

Wajib: Tidak

DeletionProtection

Menentukan apakah klaster ini dapat dihapus. Jika `DeletionProtection` diaktifkan, klaster tidak dapat dihapus kecuali jika diubah dan `DeletionProtection` dinonaktifkan. `DeletionProtection` melindungi klaster agar tidak terhapus secara tidak sengaja.

Tipe: Boolean

Wajib: Tidak

EarliestRestorableTime

Waktu paling awal dimana database dapat dipulihkan dengan point-in-time restore.

Tipe: Timestamp

Wajib: Tidak

EnabledCloudwatchLogsExports.member.N

Daftar jenis log yang dikonfigurasi oleh cluster ini untuk diekspor ke Amazon CloudWatch Logs.

Tipe: Array string

Wajib: Tidak

Endpoint

Menentukan titik akhir koneksi untuk instans primer klaster.

Tipe: String

Wajib: Tidak

Engine

Menyediakan nama mesin basis data yang akan digunakan untuk klaster ini.

Tipe: String

Wajib: Tidak

EngineVersion

Menunjukkan versi mesin basis data.

Tipe: String

Wajib: Tidak

HostedZoneId

Menentukan ID yang ditetapkan Amazon Route 53 saat Anda membuat zona yang di-hosting.

Tipe: String

Wajib: Tidak

KmsKeyId

Jika `StorageEncrypted` yatrue, pengidentifikasi AWS KMS kunci untuk cluster terenkripsi.

Tipe: String

Wajib: Tidak

LatestRestorableTime

Menentukan waktu terbaru yang database dapat dipulihkan dengan point-in-time restore.

Tipe: Timestamp

Wajib: Tidak

MasterUsername

Berisi nama pengguna utama untuk klaster.

Tipe: String

Wajib: Tidak

MultiAZ

Menentukan apakah klaster memiliki instans di beberapa Availability Zone.

Tipe: Boolean

Wajib: Tidak

PercentProgress

Menentukan kemajuan operasi sebagai persentase.

Tipe: String

Wajib: Tidak

Port

Menentukan port yang didengarkan oleh mesin basis data.

Tipe: Integer

Wajib: Tidak

PreferredBackupWindow

Menentukan rentang waktu harian selama pencadangan otomatis dibuat jika pencadangan otomatis diaktifkan, seperti yang ditentukan oleh `BackupRetentionPeriod`.

Tipe: String

Wajib: Tidak

PreferredMaintenanceWindow

Menentukan rentang waktu mingguan untuk melakukan pemeliharaan sistem, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: String

Wajib: Tidak

ReaderEndpoint

Reader endpoint untuk klaster. Reader endpoint untuk koneksi menyeimbangkan beban klaster di seluruh replika Amazon DocumentDB yang tersedia dalam klaster. Saat klien meminta koneksi baru ke reader endpoint, Amazon DocumentDB mendistribusikan permintaan koneksi di antara

replika Amazon DocumentDB di klaster. Fungsionalitas ini dapat membantu menyeimbangkan beban kerja baca Anda di beberapa replika Amazon DocumentDB di klaster Anda.

Jika terjadi failover, dan replika Amazon DocumentDB yang terhubung dengan Anda dipromosikan menjadi instans primer, koneksi Anda dijatuhkan. Untuk terus mengirimkan beban kerja baca Anda ke replika Amazon DocumentDB lainnya di klaster, Anda kemudian dapat menyambungkan kembali ke reader endpoint.

Tipe: String

Wajib: Tidak

ReadReplicaIdentifiers.ReadReplicaIdentifier.N

Berisi satu atau lebih pengidentifikasi klaster sekunder yang terkait dengan klaster ini.

Tipe: Array string

Wajib: Tidak

ReplicationSourceIdentifier

Berisi pengidentifikasi klaster sumber jika klaster ini adalah klaster sekunder.

Tipe: String

Wajib: Tidak

Status

Menentukan status saat ini dari klaster ini.

Tipe: String

Wajib: Tidak

StorageEncrypted

Menentukan apakah klaster dienkripsi.

Tipe: Boolean

Wajib: Tidak

StorageType

Jenis penyimpanan yang terkait dengan cluster Anda

Jenis penyimpanan yang terkait dengan cluster Anda

Untuk informasi tentang jenis penyimpanan untuk klaster Amazon DocumentDB, lihat Konfigurasi penyimpanan klaster di Panduan Pengembang Amazon DocumentDB.

Nilai yang valid untuk jenis penyimpanan - `standard | iopt1`

Nilai default adalah `standard`

Tipe: String

Wajib: Tidak

VpcSecurityGroups.VpcSecurityGroupMembership.N

Menyediakan daftar grup keamanan virtual private cloud (VPC) yang dimiliki klaster.

Tipe: Array objek VpcSecurityGroupMembership

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterMember

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Berisi informasi tentang sebuah instans yang merupakan bagian dari sebuah klaster.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBClusterParameterGroupStatus

Menentukan status grup parameter klaster untuk anggota klaster DB ini.

Tipe: String

Wajib: Tidak

DBInstanceIdentifier

Menentukan pengidentifikasi instans untuk anggota klaster ini.

Tipe: String

Wajib: Tidak

IsClusterWriter

Sebuah nilai yang `true` jika anggota klaster adalah instans primer untuk klaster dan `false` sebaliknya.

Tipe: Boolean

Wajib: Tidak

PromotionTier

Nilai yang menentukan urutan di mana replika Amazon DocumentDB dipromosikan ke instans primer setelah kegagalan instans primer yang ada.

Tipe: Integer

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterParameterGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang grup parameter klaster.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBClusterParameterGroupArn

Amazon Resource Name (ARN) untuk grup parameter klaster.

Tipe: String

Wajib: Tidak

DBClusterParameterGroupName

Menyediakan nama grup parameter klaster.

Tipe: String

Wajib: Tidak

DBParameterGroupFamily

Menyediakan nama famili grup parameter yang kompatibel dengan grup parameter klaster ini.

Tipe: String

Wajib: Tidak

Description

Menyediakan deskripsi yang ditentukan pelanggan untuk grup parameter klaster ini.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterRole

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menjelaskan peran AWS Identity and Access Management (IAM) yang terkait dengan cluster.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

RoleArn

Amazon Resource Name (ARN) dari IAMrole yang dikaitkan dengan klaster DB.

Tipe: String

Wajib: Tidak

Status

Menjelaskan status keterkaitan antara IAMrole dan klaster. Properti `Status` mengembalikan salah satu nilai berikut:

- `ACTIVE`- ARN iamRole dikaitkan dengan cluster dan dapat digunakan untuk mengakses AWS layanan lain atas nama Anda.
- `PENDING` - ARN IAMRole sedang dikaitkan dengan klaster.
- `INVALID`- ARN IamRole dikaitkan dengan cluster, tetapi cluster tidak dapat mengasumsikan iamRole untuk mengakses layanan lain atas nama Anda. AWS

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterSnapshot

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang snapshot klaster.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AvailabilityZones.AvailabilityZone.N

Menyediakan daftar Availability Zone Amazon EC2 tempat instans dalam snapshot klaster dapat dipulihkan.

Tipe: Array string

Wajib: Tidak

ClusterCreateTime

Menentukan waktu saat klaster dibuat, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: Timestamp

Wajib: Tidak

DBClusterIdentifier

Menentukan pengidentifikasi klaster dari klaster tempat snapshot klaster ini dibuat.

Tipe: String

Wajib: Tidak

DBClusterSnapshotArn

Amazon Resource Name (ARN) untuk snapshot klaster.

Tipe: String

Wajib: Tidak

DBClusterSnapshotIdentifier

Menentukan pengidentifikasi untuk snapshot klaster.

Tipe: String

Wajib: Tidak

Engine

Menentukan nama mesin basis data.

Tipe: String

Wajib: Tidak

EngineVersion

Menyediakan versi mesin basis data untuk snapshot klaster ini.

Tipe: String

Wajib: Tidak

KmsKeyId

Jika `StorageEncrypted` yatrue, pengenal AWS KMS kunci untuk snapshot cluster terenkripsi.

Tipe: String

Wajib: Tidak

MasterUsername

Menyediakan nama pengguna utama untuk snapshot klaster.

Tipe: String

Wajib: Tidak

PercentProgress

Menentukan persentase perkiraan data yang telah ditransfer.

Tipe: Integer

Wajib: Tidak

Port

Menentukan port yang didengarkan klaster pada saat snapshot.

Tipe: Integer

Wajib: Tidak

SnapshotCreateTime

Menyediakan waktu ketika snapshot diambil, dalam UTC.

Tipe: Timestamp

Wajib: Tidak

SnapshotType

Menyediakan jenis snapshot klaster.

Tipe: String

Wajib: Tidak

SourceDBClusterSnapshotArn

Jika snapshot klaster disalin dari snapshot klaster sumber, ARN untuk snapshot klaster sumber; jika tidak, nilai null.

Tipe: String

Wajib: Tidak

Status

Menentukan status snapshot klaster ini.

Tipe: String

Wajib: Tidak

StorageEncrypted

Menentukan apakah snapshot klaster dienkripsi.

Tipe: Boolean

Wajib: Tidak

StorageType

Jenis penyimpanan yang terkait dengan snapshot klaster Anda

Untuk informasi tentang jenis penyimpanan untuk klaster Amazon DocumentDB, lihat Konfigurasi penyimpanan klaster di Panduan Pengembang Amazon DocumentDB.

Nilai yang valid untuk jenis penyimpanan - `standard | iopt1`

Nilai default adalah `standard`

Tipe: String

Wajib: Tidak

VpcId

Menyediakan ID virtual private cloud (VPC) yang terkait dengan snapshot klaster.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterSnapshotAttribute

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Berisi nama dan nilai-nilai atribut snapshot klaster manual.

Atribut snapshot cluster manual digunakan untuk mengotorisasi lainnya Akun AWS untuk memulihkan snapshot cluster manual.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AttributeName

Nama dari atribut snapshot klaster manual.

Atribut bernama `restore` mengacu pada daftar Akun AWS yang memiliki izin untuk menyalin atau mengembalikan snapshot cluster manual.

Tipe: String

Wajib: Tidak

AttributeValues.AttributeValue.N

Nilai untuk atribut snapshot klaster manual.

Jika `AttributeName` bidang diatur `kerestore`, maka elemen ini mengembalikan daftar ID Akun AWS yang berwenang untuk menyalin atau mengembalikan snapshot cluster manual. Jika nilai `all` ada dalam daftar, maka snapshot cluster manual bersifat publik dan tersedia Akun AWS untuk disalin atau dipulihkan.

Tipe: Array string

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterSnapshotAttributesResult

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang atribut yang terkait dengan snapshot klaster.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBClusterSnapshotAttributes.DBClusterSnapshotAttribute.N

Daftar atribut dan nilai untuk snapshot klaster.

Tipe: Array objek DBClusterSnapshotAttribute

Wajib: Tidak

DBClusterSnapshotIdentifier

Pengidentifikasi dari snapshot klaster tempat atribut diterapkan.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBEngineVersion

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang versi mesin.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBEngineDescription

Deskripsi mesin basis data.

Tipe: String

Wajib: Tidak

DBEngineVersionDescription

Deskripsi versi mesin basis data.

Tipe: String

Wajib: Tidak

DBParameterGroupFamily

Nama famili grup parameter untuk mesin basis data.

Tipe: String

Wajib: Tidak

Engine

Nama mesin basis data.

Tipe: String

Wajib: Tidak

EngineVersion

Nomor versi mesin basis data.

Tipe: String

Wajib: Tidak

ExportableLogTypes.member.N

Jenis log yang dimiliki mesin database untuk diekspor ke Amazon CloudWatch Logs.

Tipe: Array string

Wajib: Tidak

SupportedCACertificateIdentifiers.member.N

Daftar pengidentifikasi sertifikat CA yang didukung.

Untuk informasi selengkapnya, lihat Memperbarui Sertifikat Amazon DocumentDB TLS Anda dan Mengenkripsi Data dalam Transit di Panduan Pengembang Amazon DocumentDB.

Tipe: Array string

Wajib: Tidak

SupportsCertificateRotationWithoutRestart

Menunjukkan apakah versi mesin mendukung memutar sertifikat server tanpa me-reboot instance DB.

Tipe: Boolean

Wajib: Tidak

SupportsLogExportsToCloudwatchLogs

Nilai yang menunjukkan apakah versi engine mendukung ekspor jenis log yang ditentukan oleh `ExportableLogTypes` ke CloudWatch Log.

Tipe: Boolean

Wajib: Tidak

ValidUpgradeTarget.UpgradeTarget.N

Daftar versi mesin yang dapat ditingkatkan untuk versi mesin basis data ini.

Tipe: Array objek UpgradeTarget

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBInstance

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang sebuah instans.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AutoMinorVersionUpgrade

Tidak berlaku. Parameter ini tidak berlaku untuk Amazon DocumentDB. Amazon DocumentDB tidak melakukan peningkatan versi minor terlepas dari nilai yang ditetapkan.

Tipe: Boolean

Wajib: Tidak

AvailabilityZone

Menentukan nama Availability Zone tempat instans berada.

Tipe: String

Wajib: Tidak

BackupRetentionPeriod

Menentukan jumlah hari yang mana snapshot otomatis dipertahankan.

Tipe: Integer

Wajib: Tidak

CACertificateIdentifier

Pengidentifikasi sertifikat CA untuk instans DB ini.

Tipe: String

Wajib: Tidak

CertificateDetails

Rincian sertifikat server instans DB.

Tipe: Objek CertificateDetails

Wajib: Tidak

CopyTagsToSnapshot

Nilai yang menunjukkan apakah akan menyalin tanda dari instans DB untuk snapshot instans DB. Secara default, tanda tidak disalin.

Tipe: Boolean

Wajib: Tidak

DBClusterIdentifier

Berisi nama klaster tempat instans menjadi anggota jika instans adalah anggota klaster.

Tipe: String

Wajib: Tidak

DBInstanceArn

Amazon Resource Name (ARN) untuk instans.

Tipe: String

Wajib: Tidak

DBInstanceClass

Berisi nama kelas komputasi dan kapasitas memori instans.

Tipe: String

Wajib: Tidak

DBInstanceIdentifier

Berisi pengidentifikasi basis data yang disediakan pengguna. Pengidentifikasi ini adalah kunci unik yang mengidentifikasi sebuah instans.

Tipe: String

Wajib: Tidak

DBInstanceStatus

Menentukan status saat ini dari basis data ini.

Tipe: String

Wajib: Tidak

DbiResourceId

Pengidentifikasi Wilayah AWS-unik dan tidak dapat diubah untuk instance. Pengenal ini ditemukan di entri AWS CloudTrail log setiap kali AWS KMS kunci untuk instance diakses.

Tipe: String

Wajib: Tidak

DBSubnetGroup

Menentukan informasi tentang grup subnet yang terkait dengan instans, termasuk nama, deskripsi, dan subnet dalam grup subnet.

Tipe: Objek DBSubnetGroup

Wajib: Tidak

EnabledCloudwatchLogsExports.member.N

Daftar jenis log yang instance ini dikonfigurasi untuk diekspor ke CloudWatch Log.

Tipe: Array string

Wajib: Tidak

Endpoint

Menentukan titik akhir koneksi.

Tipe: Objek Endpoint

Wajib: Tidak

Engine

Menyediakan nama mesin basis data yang akan digunakan untuk instans ini.

Tipe: String

Wajib: Tidak

EngineVersion

Menunjukkan versi mesin basis data.

Tipe: String

Wajib: Tidak

InstanceCreateTime

Menyediakan tanggal dan waktu saat instans dibuat.

Tipe: Timestamp

Wajib: Tidak

KmsKeyId

Jika `StorageEncrypted` yatrue, pengidentifikasi AWS KMS kunci untuk instance terenkripsi.

Tipe: String

Wajib: Tidak

LatestRestorableTime

Menentukan waktu terbaru yang database dapat dipulihkan dengan point-in-time restore.

Tipe: Timestamp

Wajib: Tidak

PendingModifiedValues

Menentukan bahwa perubahan pada instans sedang tertunda. Elemen ini disertakan hanya ketika perubahan tertunda. Perubahan spesifik diidentifikasi oleh sub elemen.

Tipe: Objek PendingModifiedValues

Wajib: Tidak

PreferredBackupWindow

Menentukan rentang waktu harian selama pencadangan otomatis dibuat jika pencadangan otomatis diaktifkan, seperti yang ditentukan oleh `BackupRetentionPeriod`.

Tipe: String

Wajib: Tidak

PreferredMaintenanceWindow

Menentukan rentang waktu mingguan untuk melakukan pemeliharaan sistem, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: String

Wajib: Tidak

PromotionTier

Nilai yang menentukan urutan di mana replika Amazon DocumentDB dipromosikan ke instans primer setelah kegagalan instans primer yang ada.

Tipe: Integer

Wajib: Tidak

PubliclyAccessible

Tidak didukung. Amazon DocumentDB saat ini tidak mendukung titik akhir publik. Nilai dari `PubliclyAccessible` selalu `false`.

Tipe: Boolean

Wajib: Tidak

StatusInfos.DBInstanceStatusInfo.N

Status replika baca. Jika instans bukan replika baca, ini kosong.

Tipe: Array objek DBInstanceStatusInfo

Wajib: Tidak

StorageEncrypted

Menentukan apakah instans dienkripsi atau tidak.

Tipe: Boolean

Wajib: Tidak

VpcSecurityGroups.VpcSecurityGroupMembership.N

Menyediakan daftar elemen grup keamanan VPC tempat instans berada.

Tipe: Array objek VpcSecurityGroupMembership

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBInstanceStatusInfo

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menyediakan daftar informasi status untuk sebuah instans.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Message

Detail kesalahan jika ada kesalahan untuk instans. Jika instans tidak dalam status kesalahan, nilai ini kosong.

Tipe: String

Wajib: Tidak

Normal

Nilai Boolean yaitu `true` jika instans beroperasi secara normal, atau `false` jika instans berada dalam status kesalahan.

Tipe: Boolean

Wajib: Tidak

Status

Status dari instans. Untuk `StatusType` dari replika baca, nilainya bisa `replicating`, error, `stopped`, atau `terminated`.

Tipe: String

Wajib: Tidak

StatusType

Nilai ini saat ini adalah "`read replication`."

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBSubnetGroup

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang grup subnet.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBSubnetGroupArn

Amazon Resource Name (ARN) untuk grup subnet DB.

Tipe: String

Wajib: Tidak

DBSubnetGroupDescription

Menyediakan deskripsi grup subnet.

Tipe: String

Wajib: Tidak

DBSubnetGroupName

Nama grup subnet.

Tipe: String

Wajib: Tidak

SubnetGroupStatus

Menyediakan status grup subnet.

Tipe: String

Wajib: Tidak

Subnets.Subnet.N

Informasi mendetail tentang satu atau lebih subnet dalam grup subnet.

Tipe: Array objek Subnet

Wajib: Tidak

VpcId

Menyediakan ID virtual private cloud (VPC) dari grup subnet.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Endpoint

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi jaringan untuk mengakses klaster atau instans. Program klien harus menentukan titik akhir yang valid untuk mengakses sumber daya Amazon DocumentDB ini.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Address

Menentukan alamat DNS dari instans.

Tipe: String

Wajib: Tidak

HostedZoneId

Menentukan ID yang ditetapkan Amazon Route 53 saat Anda membuat zona yang di-hosting.

Tipe: String

Wajib: Tidak

Port

Menentukan port yang didengarkan oleh mesin basis data.

Tipe: Integer

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

EngineDefaults

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Berisi hasil pemanggilan operasi `DescribeEngineDefaultClusterParameters` yang berhasil.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBParameterGroupFamily

Nama famili grup parameter klaster untuk mengembalikan informasi parameter mesin.

Tipe: String

Wajib: Tidak

Marker

Token pagination opsional yang disediakan oleh permintaan sebelumnya. Jika parameter ini ditentukan, respons hanya menyertakan catatan di luar penanda, hingga nilai yang ditentukan dengan `MaxRecords`.

Tipe: String

Wajib: Tidak

Parameters.Parameter.N

Parameter dari famili grup parameter klaster tertentu.

Tipe: Array objek Parameter

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Event

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang sebuah peristiwa.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Date

Menentukan tanggal dan waktu peristiwa.

Tipe: Timestamp

Wajib: Tidak

EventCategories.EventCategory.N

Menentukan kategori untuk peristiwa tersebut.

Tipe: Array string

Wajib: Tidak

Message

Menyediakan teks peristiwa ini.

Tipe: String

Wajib: Tidak

SourceArn

Amazon Resource Name (ARN) untuk peristiwa.

Tipe: String

Wajib: Tidak

SourceIdentifier

Menyediakan pengidentifikasi untuk sumber peristiwa.

Tipe: String

Wajib: Tidak

SourceType

Menentukan jenis sumber untuk peristiwa ini.

Jenis: String

Nilai yang Valid: `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EventCategoriesMap

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Jenis sumber peristiwa, disertai dengan satu atau beberapa nama kategori peristiwa.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

EventCategories.EventCategory.N

Kategori peristiwa untuk jenis sumber yang ditentukan.

Tipe: Array string

Wajib: Tidak

SourceType

Jenis sumber yang dimiliki oleh kategori yang dikembalikan.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EventSubscription

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang peristiwa yang telah Anda langgani.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

CustomerAwsId

Akun AWS pelanggan yang terkait dengan langganan notifikasi acara Amazon DocumentDB.

Tipe: String

Wajib: Tidak

CustSubscriptionId

ID langganan notifikasi peristiwa Amazon DocumentDB.

Tipe: String

Wajib: Tidak

Enabled

Nilai Boolean yang menunjukkan apakah langganan diaktifkan. Nilai dari `true` menunjukkan bahwa langganan diaktifkan.

Tipe: Boolean

Wajib: Tidak

EventCategoriesList.EventCategory.N

Daftar kategori acara untuk langganan notifikasi peristiwa Amazon DocumentDB.

Tipe: Array string

Wajib: Tidak

EventSubscriptionArn

Amazon Resource Name (ARN) untuk langganan peristiwa.

Tipe: String

Wajib: Tidak

SnsTopicArn

ARN topik langganan notifikasi peristiwa Amazon DocumentDB.

Tipe: String

Wajib: Tidak

SourceIdsList.SourceId.N

Daftar ID sumber untuk langganan notifikasi peristiwa Amazon DocumentDB.

Tipe: Array string

Wajib: Tidak

SourceType

Jenis sumber untuk langganan notifikasi peristiwa Amazon DocumentDB.

Tipe: String

Wajib: Tidak

Status

Status langganan notifikasi peristiwa Amazon DocumentDB.

Batasan:

Dapat menjadi salah satu dari yang berikut: `creating`, `modifying`, `deleting`, `active`, `no-permission`, `topic-not-exist`

Status `no-permission` menunjukkan bahwa Amazon DocumentDB tidak lagi memiliki izin untuk memposting ke topik SNS. Status `topic-not-exist` menunjukkan bahwa topik telah dihapus setelah langganan dibuat.

Tipe: String

Wajib: Tidak

SubscriptionCreationTime

Waktu saat langganan notifikasi peristiwa Amazon DocumentDB dibuat.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Filter

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Kumpulan nilai filter bernama, digunakan untuk menampilkan daftar hasil yang lebih spesifik. Anda dapat menggunakan filter untuk mencocokkan sekumpulan sumber daya menurut kriteria tertentu, seperti ID.

Wildcard tidak didukung dalam filter.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Name

Nama filter. Nama filter peka huruf besar/kecil.

Tipe: String

Diperlukan: Ya

Values.Value.N

Satu atau lebih nilai filter. Nilai filter peka huruf besar/kecil.

Tipe: Array string

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalCluster

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Sebuah tipe data yang mewakili sebuah klaster global Amazon DocumentDB.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DatabaseName

Nama basis data default dalam klaster global baru.

Tipe: String

Wajib: Tidak

DeletionProtection

Pengaturan perlindungan penghapusan untuk klaster global baru.

Tipe: Boolean

Wajib: Tidak

Engine

Mesin basis data Amazon DocumentDB yang digunakan oleh klaster global.

Tipe: String

Wajib: Tidak

EngineVersion

Menunjukkan versi mesin basis data.

Tipe: String

Wajib: Tidak

GlobalClusterArn

Amazon Resource Name (ARN) untuk klaster global.

Tipe: String

Wajib: Tidak

GlobalClusterIdentifier

Berisi pengidentifikasi klaster global yang disediakan pengguna. Pengidentifikasi ini adalah kunci unik yang mengidentifikasi sebuah klaster global.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `[A-Za-z][0-9A-Za-z-:._]*`

Wajib: Tidak

GlobalClusterMembers.GlobalClusterMember.N

Daftar ID klaster untuk klaster sekunder dalam klaster global. Saat ini terbatas pada satu item.

Tipe: Array objek GlobalClusterMember

Wajib: Tidak

GlobalClusterResourceId

Pengidentifikasi Wilayah AWS-unik dan tidak dapat diubah untuk cluster database global. Pengenal ini ditemukan di entri AWS CloudTrail log setiap kali kunci master AWS KMS pelanggan (CMK) untuk cluster diakses.

Tipe: String

Wajib: Tidak

Status

Menentukan status saat ini dari klaster global ini.

Tipe: String

Wajib: Tidak

StorageEncrypted

Pengaturan enkripsi penyimpanan untuk klaster global.

Tipe: Boolean

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalClusterMember

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Struktur data dengan informasi tentang setiap klaster primer dan sekunder yang terkait dengan klaster global Amazon DocumentDB.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

DBClusterArn

Amazon Resource Name (ARN) untuk setiap klaster Amazon DocumentDB.

Tipe: String

Wajib: Tidak

IsWriter

Menentukan apakah klaster Amazon DocumentDB adalah klaster primer (yaitu, memiliki kemampuan baca-tulis) untuk klaster global Amazon DocumentDB yang terkait dengannya.

Tipe: Boolean

Wajib: Tidak

Readers.member.N

Amazon Resource Name (ARN) untuk setiap klaster sekunder baca saja yang terkait dengan klaster global Aurora.

Tipe: Array string

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OrderableDBInstanceOption

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Opsi yang tersedia untuk instans.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AvailabilityZones.AvailabilityZone.N

Daftar Availability Zone untuk instans.

Tipe: Array objek AvailabilityZone

Wajib: Tidak

DBInstanceClass

Kelas instans untuk sebuah instans.

Tipe: String

Wajib: Tidak

Engine

Jenis mesin dari sebuah instans.

Tipe: String

Wajib: Tidak

EngineVersion

Versi mesin dari sebuah instans.

Tipe: String

Wajib: Tidak

LicenseModel

Model lisensi untuk sebuah instans.

Tipe: String

Wajib: Tidak

Vpc

Menunjukkan apakah sebuah instans berada dalam virtual private cloud (VPC).

Tipe: Boolean

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Parameter

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang parameter individu.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AllowedValues

Menentukan rentang nilai yang valid untuk parameter.

Tipe: String

Wajib: Tidak

ApplyMethod

Menunjukkan kapan harus menerapkan pembaruan parameter.

Jenis: String

Nilai yang Valid: `immediate | pending-reboot`

Wajib: Tidak

ApplyType

Menentukan jenis parameter spesifik mesin.

Tipe: String

Wajib: Tidak

DataType

Menentukan tipe data yang valid untuk parameter.

Tipe: String

Wajib: Tidak

Description

Menyediakan deskripsi parameter.

Tipe: String

Wajib: Tidak

IsModifiable

Menunjukkan apakah (`true`) atau tidak (`false`) parameter dapat dimodifikasi. Beberapa parameter memiliki implikasi keamanan atau operasional yang mencegahnya untuk diubah.

Tipe: Boolean

Wajib: Tidak

MinimumEngineVersion

Versi mesin paling awal yang parameternya dapat diterapkan.

Tipe: String

Wajib: Tidak

ParameterName

Menentukan nama dari parameter.

Tipe: String

Wajib: Tidak

ParameterValue

Menentukan nilai dari parameter.

Tipe: String

Wajib: Tidak

Source

Menunjukkan sumber nilai parameter.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingCloudwatchLogsExports

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Daftar jenis log yang konfigurasinya masih tertunda. Jenis log ini sedang dalam proses diaktifkan atau dinonaktifkan.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

LogTypesToDisable.member.N

Jenis log yang sedang dalam proses diaktifkan. Setelah diaktifkan, jenis log ini diekspor ke Amazon CloudWatch Logs.

Tipe: Array string

Wajib: Tidak

LogTypesToEnable.member.N

Jenis log yang sedang dalam proses dinonaktifkan. Setelah dinonaktifkan, jenis log ini tidak diekspor ke CloudWatch Log.

Tipe: Array string

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingMaintenanceAction

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Menyediakan informasi tentang tindakan pemeliharaan yang tertunda untuk sumber daya.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Action

Jenis tindakan pemeliharaan tertunda yang tersedia untuk sumber daya.

Tipe: String

Wajib: Tidak

AutoAppliedAfterDate

Tanggal jendela pemeliharaan saat tindakan diterapkan. Tindakan pemeliharaan diterapkan ke sumber daya selama jendela pemeliharaan pertama setelah tanggal ini. Jika tanggal ini ditentukan, semua permintaan opt-in `next-maintenance` diabaikan.

Tipe: Timestamp

Wajib: Tidak

CurrentApplyDate

Tanggal efektif ketika tindakan pemeliharaan tertunda diterapkan ke sumber daya.

Tipe: Timestamp

Wajib: Tidak

Description

Deskripsi yang memberikan detail lebih lanjut tentang tindakan pemeliharaan.

Tipe: String

Wajib: Tidak

ForcedApplyDate

Tanggal ketika tindakan pemeliharaan diterapkan secara otomatis. Tindakan pemeliharaan diterapkan ke sumber daya pada tanggal ini terlepas dari jendela pemeliharaan untuk sumber daya. Jika tanggal ini ditentukan, semua permintaan opt-in `immediate` diabaikan.

Tipe: Timestamp

Wajib: Tidak

OptInStatus

Menunjukkan jenis permintaan opt-in yang telah diterima untuk sumber daya.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingModifiedValues

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Satu atau lebih pengaturan yang dimodifikasi untuk sebuah instans. Pengaturan yang dimodifikasi ini telah diminta, tetapi belum diterapkan.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AllocatedStorage

Berisi ukuran `AllocatedStorage` baru untuk instans yang akan diterapkan atau sedang diterapkan.

Tipe: Integer

Wajib: Tidak

BackupRetentionPeriod

Menentukan jumlah hari yang tertunda untuk penyimpanan cadangan otomatis dipertahankan.

Tipe: Integer

Wajib: Tidak

CACertificateIdentifier

Menentukan pengidentifikasi sertifikat otoritas sertifikat (CA) untuk instans DB.

Tipe: String

Wajib: Tidak

DBInstanceClass

Berisi `DBInstanceClass` baru untuk instans yang akan diterapkan atau sedang diterapkan.

Tipe: String

Wajib: Tidak

DBInstanceIdentifier

Berisi `DBInstanceIdentifier` baru untuk instans yang akan diterapkan atau sedang diterapkan.

Tipe: String

Wajib: Tidak

DBSubnetGroupName

Grup subnet baru untuk instans.

Tipe: String

Wajib: Tidak

EngineVersion

Menunjukkan versi mesin basis data.

Tipe: String

Wajib: Tidak

Iops

Menentukan nilai Provisioned IOPS baru untuk instans yang akan diterapkan atau sedang diterapkan.

Tipe: Integer

Wajib: Tidak

LicenseModel

Model lisensi untuk instans.

Nilai valid: `license-included`, `bring-your-own-license`, `general-public-license`

Tipe: String

Wajib: Tidak

MasterUserPassword

Berisi perubahan yang tertunda atau sedang dalam proses dari kredensial utama untuk instans.

Tipe: String

Wajib: Tidak

MultiAZ

Menunjukkan bahwa instans Single-AZ akan berubah menjadi deployment Multi-AZ.

Tipe: Boolean

Wajib: Tidak

PendingCloudwatchLogsExports

Daftar jenis log yang konfigurasinya masih tertunda. Jenis log ini sedang dalam proses diaktifkan atau dinonaktifkan.

Tipe: Objek PendingCloudwatchLogsExports

Wajib: Tidak

Port

Menentukan port tertunda untuk instans.

Tipe: Integer

Wajib: Tidak

StorageType

Menentukan jenis penyimpanan yang akan dikaitkan dengan instans.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

ResourcePendingMaintenanceActions

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Merupakan output dari ApplyPendingMaintenanceAction.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

PendingMaintenanceActionDetails.PendingMaintenanceAction.N

Daftar yang menyediakan detail tentang tindakan pemeliharaan yang tertunda untuk sumber daya.

Tipe: Array objek PendingMaintenanceAction

Wajib: Tidak

ResourceIdentifier

Amazon Resource Name (ARN) dari sumber daya yang memiliki tindakan pemeliharaan tertunda.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Subnet

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Informasi mendetail tentang subnet.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

SubnetAvailabilityZone

Menentukan Availability Zone untuk subnet.

Tipe: Objek AvailabilityZone

Wajib: Tidak

SubnetIdentifier

Menentukan pengidentifikasi dari subnet.

Tipe: String

Wajib: Tidak

SubnetStatus

Menentukan status dari subnet.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

Tag

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Metadata yang ditetapkan ke sumber daya Amazon DocumentDB yang terdiri dari pasangan nilai kunci.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Key

Nama tag yang diperlukan. Nilai string dapat terdiri dari 1 hingga 128 karakter Unicode dan tidak boleh diawali dengan "`aws:`" atau "`rds:`". String tersebut dapat berisi rangkaian huruf Unicode, angka, spasi, '_', '.', '/', '=', '+', '-' (Java regex: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$").

Tipe: String

Wajib: Tidak

Value

Nilai opsional dari tag. Nilai string dapat terdiri dari 1 hingga 256 karakter Unicode dan tidak boleh diawali dengan "`aws:`" atau "`rds:`". String tersebut dapat berisi rangkaian huruf Unicode, angka, spasi, '_', '.', '/', '=', '+', '-' (Java regex: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$").

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpgradeTarget

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Versi mesin basis data yang dapat ditingkatkan instans.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

AutoUpgrade

Nilai yang menunjukkan apakah versi target diterapkan ke instans DB sumber apa pun yang memiliki `AutoMinorVersionUpgrade` yang diatur ke `true`.

Tipe: Boolean

Wajib: Tidak

Description

Versi mesin basis data yang dapat ditingkatkan instans.

Tipe: String

Wajib: Tidak

Engine

Nama mesin basis data target peningkatan.

Tipe: String

Wajib: Tidak

EngineVersion

Nomor versi mesin basis data target peningkatan.

Tipe: String

Wajib: Tidak

IsMajorVersionUpgrade

Nilai yang menunjukkan apakah mesin basis data ditingkatkan ke versi major.

Tipe: Boolean

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

VpcSecurityGroupMembership

Layanan: Amazon DocumentDB (with MongoDB compatibility)

Digunakan sebagai elemen respons untuk kueri tentang keanggotaan grup keamanan virtual private cloud (VPC).

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Status

Status grup keamanan VPC.

Tipe: String

Wajib: Tidak

VpcSecurityGroupId

Nama grup keamanan VPC.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Cluster Elastis Amazon DocumentDB

Tipe data berikut didukung oleh Amazon DocumentDB Elastic Clusters:

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- Shard
- ValidationExceptionField

Cluster

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan informasi tentang cluster elastis tertentu.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

adminUserName

Nama administrator cluster elastis.

Tipe: String

Diperlukan: Ya

authType

Jenis otentikasi untuk cluster elastis.

Jenis: String

Nilai yang Valid: `PLAIN_TEXT | SECRET_ARN`

Wajib: Ya

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Tipe: String

Diperlukan: Ya

clusterEndpoint

URL yang digunakan untuk terhubung ke cluster elastis.

Tipe: String

Diperlukan: Ya

clusterName

Nama cluster elastis.

Tipe: String

Diperlukan: Ya

createTime

Waktu ketika cluster elastis dibuat di Universal Coordinated Time (UTC).

Tipe: String

Diperlukan: Ya

kmsKeyId

Pengidentifikasi kunci KMS yang digunakan untuk mengenkripsi cluster elastis.

Tipe: String

Diperlukan: Ya

preferredMaintenanceWindow

Rentang waktu mingguan selama pemeliharaan sistem dapat terjadi, dalam Waktu Universal Terkoordinasi (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Tipe: String

Diperlukan: Ya

shardCapacity

Jumlah vCPU yang ditetapkan untuk setiap pecahan cluster elastis. Maksimal 64. Nilai yang diizinkan adalah 2, 4, 8, 16, 32, 64.

Jenis: Integer

Wajib: Ya

shardCount

Jumlah pecahan yang ditugaskan ke cluster elastis. Maksimal 32.

Jenis: Integer

Wajib: Ya

status

Status cluster elastis.

Jenis: String

Nilai yang Valid: `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`

Wajib: Ya

subnetIds

ID subnet Amazon EC2 untuk cluster elastis.

Tipe: Array string

Wajib: Ya

vpcSecurityGroupIds

Daftar grup keamanan VPC EC2 yang terkait dengan cluster elastis ini.

Tipe: Array string

Wajib: Ya

backupRetentionPeriod

Jumlah hari di mana snapshot otomatis dipertahankan.

Tipe: Integer

Wajib: Tidak

preferredBackupWindow

Rentang waktu harian di mana pencadangan otomatis dibuat jika pencadangan otomatis diaktifkan, sebagaimana ditentukan oleh. `backupRetentionPeriod`

Tipe: String

Wajib: Tidak

shardInstanceCount

Jumlah contoh replika yang diterapkan ke semua pecahan di cluster. `shardInstanceCount`Nilai 1 berarti ada satu contoh penulis, dan setiap contoh tambahan adalah replika yang dapat digunakan untuk membaca dan untuk meningkatkan ketersediaan.

Tipe: Integer

Wajib: Tidak

shards

Jumlah total pecahan di cluster.

Tipe: Array objek Shard

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterInList

Layanan: Amazon DocumentDB Elastic Clusters

Daftar cluster elastis Amazon DocumentDB.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Tipe: String

Diperlukan: Ya

clusterName

Nama cluster elastis.

Tipe: String

Diperlukan: Ya

status

Status cluster elastis.

Jenis: String

Nilai yang Valid: `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`

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterSnapshot

Layanan: Amazon DocumentDB Elastic Clusters

Mengembalikan informasi tentang snapshot cluster elastis tertentu.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

adminUserName

Nama administrator cluster elastis.

Tipe: String

Diperlukan: Ya

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Tipe: String

Diperlukan: Ya

clusterCreationTime

Waktu ketika cluster elastis dibuat di Universal Coordinated Time (UTC).

Tipe: String

Diperlukan: Ya

kmsKeyId

Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN) untuk kunci enkripsi KMS. Jika Anda membuat cluster menggunakan akun Amazon yang sama yang memiliki kunci enkripsi KMS ini, Anda dapat menggunakan alias kunci KMS alih-alih ARN sebagai kunci enkripsi KMS. Jika kunci enkripsi tidak ditentukan di sini, Amazon DocumentDB menggunakan kunci enkripsi default yang dibuat KMS untuk akun Anda. Akun Anda memiliki kunci enkripsi default yang berbeda untuk setiap Wilayah Amazon.

Tipe: String

Diperlukan: Ya

snapshotArn

Pengidentifikasi ARN dari snapshot cluster elastis.

Tipe: String

Diperlukan: Ya

snapshotCreationTime

Waktu ketika snapshot cluster elastis dibuat di Universal Coordinated Time (UTC).

Tipe: String

Diperlukan: Ya

snapshotName

Nama snapshot cluster elastis.

Tipe: String

Diperlukan: Ya

status

Status snapshot cluster elastis.

Jenis: String

Nilai yang Valid: `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`

Wajib: Ya

subnetIds

ID subnet Amazon EC2 untuk cluster elastis.

Tipe: Array string

Wajib: Ya

vpcSecurityGroupIds

Daftar grup keamanan VPC EC2 untuk diasosiasikan dengan cluster elastis.

Tipe: Array string

Wajib: Ya

snapshotType

Jenis snapshot klaster untuk dikembalikan. Anda dapat menentukan salah satu nilai berikut:

- `automated`- Kembalikan semua snapshot cluster yang Amazon DocumentDB telah secara otomatis dibuat untuk akun Anda. AWS
- `manual`- Kembalikan semua snapshot cluster yang telah Anda buat secara manual untuk AWS akun Anda.

Jenis: String

Nilai yang Valid: `MANUAL | AUTOMATED`

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterSnapshotInList

Layanan: Amazon DocumentDB Elastic Clusters

Daftar snapshot cluster elastis.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

clusterArn

Pengidentifikasi ARN dari cluster elastis.

Tipe: String

Diperlukan: Ya

snapshotArn

Pengidentifikasi ARN dari snapshot cluster elastis.

Tipe: String

Diperlukan: Ya

snapshotCreationTime

Waktu ketika snapshot cluster elastis dibuat di Universal Coordinated Time (UTC).

Tipe: String

Diperlukan: Ya

snapshotName

Nama snapshot cluster elastis.

Tipe: String

Diperlukan: Ya

status

Status snapshot cluster elastis.

Jenis: String

Nilai yang Valid: `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`

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Shard

Layanan: Amazon DocumentDB Elastic Clusters

Nama pecahan.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

createTime

Waktu ketika pecahan dibuat di Universal Coordinated Time (UTC).

Tipe: String

Diperlukan: Ya

shardId

ID dari pecahan.

Tipe: String

Diperlukan: Ya

status

Status pecahan saat ini.

Jenis: String

Nilai yang Valid: `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`

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ValidationExceptionField

Layanan: Amazon DocumentDB Elastic Clusters

Bidang tertentu di mana pengecualian validasi yang diberikan terjadi.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

message

Pesan kesalahan yang menjelaskan pengecualian validasi di bidang ini.

Tipe: String

Diperlukan: Ya

name

Nama bidang tempat pengecualian validasi terjadi.

Tipe: String

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Kesalahan Umum

Bagian ini berisi daftar kesalahan yang umum terjadi pada tindakan API dari semua layanan AWS. Untuk kesalahan khusus pada tindakan API untuk layanan ini, lihat topik untuk tindakan API tersebut.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

IncompleteSignature

Tanda tangan permintaan tidak sesuai dengan standar AWS.

Kode Status HTTP: 400

InternalFailure

Pemrosesan permintaan telah gagal karena kesalahan yang tidak diketahui, pengecualian atau kegagalan.

Kode Status HTTP: 500

InvalidAction

Tindakan atau operasi yang diminta tidak valid. Verifikasi bahwa tindakan diketik dengan benar.

Kode Status HTTP: 400

InvalidClientTokenId

Sertifikat X.509 atau access key ID AWS yang diberikan tidak ada dalam catatan kami.

Kode Status HTTP: 403

NotAuthorized

Anda tidak memiliki izin untuk melakukan tindakan ini.

Kode Status HTTP: 400

OptInRequired

Access key ID AWS membutuhkan berlangganan untuk layanan.

Kode Status HTTP: 403

RequestExpired

Permintaan menjangkau layanan lebih dari 15 menit setelah stempel tanggal pada permintaan atau lebih dari 15 menit setelah tanggal kedaluwarsa permintaan (seperti untuk URL pre-signed), atau stempel tanggal pada permintaan lebih dari 15 menit di masa mendatang.

Kode Status HTTP: 400

ServiceUnavailable

Permintaan telah gagal karena kegagalan sementara server.

Kode Status HTTP: 503

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

ValidationError

Input gagal untuk memenuhi batasan yang ditentukan oleh layanan AWS.

Kode Status HTTP: 400

Parameter Umum

Daftar berikut berisi parameter yang digunakan semua tindakan untuk menandatangani permintaan Tanda Tangan Versi 4 dengan string kueri. Setiap parameter khusus tindakan tercantum dalam topik untuk tindakan tersebut. Untuk informasi selengkapnya tentang Signature Versi 4, lihat Menandatangani permintaanAWS API di Panduan Pengguna IAM.

Action

Tindakan yang harus dilakukan.

Tipe: string

Wajib: Ya

Version

Versi API yang ditulis dalam permintaan, dinyatakan dalam format HH-BB-TTTT.

Tipe: string

Wajib: Ya

X-Amz-Algorithm

Algoritme hash yang Anda gunakan untuk membuat tanda tangan permintaan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Nilai yang Valid: `AWS4-HMAC-SHA256`

Diperlukan: Kondisional

X-Amz-Credential

Nilai lingkup kredensial, yang merupakan string yang menyertakan access key Anda, tanggal, wilayah yang Anda targetkan, layanan yang Anda minta, dan string penghentian ("aws4_request"). Nilai dinyatakan dalam format berikut: access_key/HHBBTTTT/wilayah/layanan/aws4_request.

Untuk informasi selengkapnya, lihat Membuat permintaanAWS API yang ditandatangani di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

X-Amz-Date

Tanggal yang digunakan untuk membuat tanda tangan. Format harus berupa format dasar ISO 8601 (YYYYMMDD'T'HMMSS'Z'). Misalnya, waktu tanggal berikut adalah nilai X-Amz-Date yang valid: `20120325T120000Z`.

Syarat: X-Amz-Date bersifat opsional untuk semua permintaan; nilai ini dapat digunakan untuk mengganti tanggal yang digunakan untuk menandatangani permintaan. Jika header Tanggal ditentukan dalam format dasar ISO 8601, X-Amz-Date tidak diperlukan. Ketika X-Amz-Date digunakan, ia selalu mengganti nilai header Tanggal. Untuk informasi selengkapnya, lihat Elemen tanda tangan permintaanAWS API di Panduan Pengguna IAM.

Tipe: string

Diperlukan: Kondisional

X-Amz-Security-Token

Token keamanan keamanan sementara yang diperoleh melalui panggilan keAWS Security Token Service (AWS STS). Untuk daftar layanan yang mendukung kredenial keamanan sementaraAWS STS, lihat Layanan AWSlayanan IAM dalam Panduan Pengguna IAM.

Syarat: Jika Anda menggunakan kredenial keamanan keamanan.AWS STS

Tipe: string

Diperlukan: Kondisional

X-Amz-Signature

Menentukan tanda tangan yang dikodekan oleh hex yang dihitung dari string to sign dan kunci penandatanganan turunan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

X-Amz-SignedHeaders

Menentukan semua header HTTP yang disertakan sebagai bagian dari permintaan kanonik. Untuk informasi selengkapnya tentang menentukan header yang ditandatangani, lihat Membuat permintaanAWS API yang ditandatangani di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

Catatan Rilis

Catatan rilis ini menjelaskan fitur Amazon DocumentDB, peningkatan, dan perbaikan bug berdasarkan tanggal rilis. Catatan rilis mencakup pembaruan untuk semua versi mesin Amazon DocumentDB saat terjadi.

Anda dapat menentukan versi patch mesin Amazon DocumentDB saat ini dengan menjalankan perintah berikut:

```
db.runCommand({getEngineVersion: 1})
```

Jika klaster Anda tidak menggunakan versi terbaru mesin, kemungkinan Anda memiliki pemeliharaan tersedia yang tertunda yang akan meningkatkan mesin Anda. Untuk informasi selengkapnya, lihat Memelihara Amazon DocumentDB di Panduan Pengembang.

Topik

- 29 Mei 2024
- April 3, 2024
- Februari 22, 2024
- Januari 30, 2024
- 10 Januari 2024
- 20 Desember 2023
- 13 Desember 2023
- November 29, 2023
- 21 November 2023
- 17 November 2023
- 6 November 2023
- 20 Oktober 2023
- 25 September 2023
- 20 September 2023
- 15 September 2023
- 11 September 2023

- 3 Agustus 2023
- 13 Juli 2023
- Juni 7, 2023
- 10 Mei 2023
- 4 April 2023
- 22 Maret 2023
- 1 Maret 2023
- 27 Februari 2023
- 2 Februari 2023
- 30 November 2022
- Agustus 9, 2022
- 25 Juli 2022
- Juni 27, 2022
- 29 April 2022
- 7 April 2022
- 16 Maret 2022
- 8 Februari 2022
- 24 Januari 2022
- Januari 21, 2022
- 25 Oktober 2021
- 24 Juni 2021
- 4 Mei 2021
- 15 Januari 2021
- 9 November 2020
- 30 Oktober 2020
- 22 September 2020
- 10 Juli 2020
- 30 Juni 2020

29 Mei 2024

> **Note**
>
> Patch mesin Amazon DocumentDB berikut sedang dalam proses dikirim ke semua wilayah Amazon DocumentDB dalam beberapa minggu ke depan. Ketika patch mesin ini tersedia di wilayah Anda, Anda akan menerima pemberitahuan patch layanan melalui Dasbor AWS Kesehatan (AHD) di AWS Management Console dan melalui email ke alamat email pengguna root AWS akun Anda.
> Patch mesin ini mencakup fitur-fitur baru berikut dan perbaikan bug. Harap dicatat bahwa daftar di bawah ini, bersama dengan dokumentasi pendukung yang relevan, dapat diperbarui untuk menyertakan pengumuman fitur tambahan setelah patch mesin tersedia di semua wilayah.

Fitur baru

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.6742)

- Ditambahkan dukungan untuk `regexMatch` dan `regexFind` operator.
- Menambahkan dukungan untuk memastikan presisi penuh dalam log audit saat menangani bilangan bulat besar. Log audit sekarang mempertahankan representasi numerik yang tepat untuk semua angka, mencegah hilangnya presisi.

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.10593)

- Menambahkan dukungan untuk memastikan presisi penuh dalam log audit saat menangani bilangan bulat besar. Log audit sekarang mempertahankan representasi numerik yang tepat untuk semua angka, mencegah hilangnya presisi.

April 3, 2024

Amazon DocumentDB sekarang tersedia di wilayah Timur Tengah (UEA). Untuk informasi lebih lanjut, lihat posting blog ini.

Fitur baru

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.5721)

- Ditambahkan dukungan untuk `bypassDocumentValidation` dan pesan kesalahan granular untuk`$jsonSchema`. Untuk informasi selengkapnya tentang `bypassDocumentValidation`, lihat bypassDocumentValidation.
- Menambahkan dukungan dari`$expr`.
- Menambahkan dukungan untuk Uncorrelated Joins in. `$lookup`
- Menambahkan dukungan untuk mempertahankan aturan validasi dalam tahap `$out` agregasi.

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.10392)

- Menambahkan dukungan untuk `bypassDocumentValidation` untuk`$jsonSchema`. Untuk informasi selengkapnya tentang `bypassDocumentValidation`, lihat bypassDocumentValidation.
- Menambahkan dukungan dari`$expr`.
- Menambahkan dukungan untuk Uncorrelated Joins in. `$lookup`
- Menambahkan dukungan untuk mempertahankan aturan validasi dalam tahap `$out` agregasi.

Perbaikan bug dan perubahan lainnya

- Memperbaiki kesalahan saat memanggil `db.coll.stats()` mongo shell versi 1.7 dan yang lebih baru.
- Memperbaiki masalah kebocoran memori untuk kueri aliran perubahan yang berisi `$regex` sebagai bagian dari pipeline agregasi yang sama.

Februari 22, 2024

Fitur baru

Cluster elastis Amazon DocumentDB

Cluster elastis Amazon DocumentDB sekarang mendukung fitur-fitur berikut:

- Replika instance pecahan sekunder yang dapat dibaca - untuk informasi lebih lanjut, lihat langkah 5b dari. Langkah 1: Buat cluster elastis

- Start/stop cluster - untuk informasi lebih lanjut, lihat. Menghentikan dan memulai cluster elastis Amazon DocumentDB
- Instans pecahan yang dapat dikonfigurasi - untuk informasi lebih lanjut, lihat langkah 5b dari. Langkah 1: Buat cluster elastis
- Pencadangan otomatis untuk snapshot - untuk informasi lebih lanjut, lihat. Mengelola cadangan otomatis snapshot cluster elastis
- Salin snapshot - untuk informasi lebih lanjut, lihatMenyalin snapshot cluster elastis.

Januari 30, 2024

Fitur baru

Cluster elastis Amazon DocumentDB

Cluster elastis Amazon DocumentDB sekarang tersedia di wilayah berikut:

- Asia Pasifik (Mumbai)
- Asia Pasifik (Seoul)
- Amerika Selatan (Sao Paulo)
- Eropa (London)

Untuk informasi selengkapnya, lihat Wilayah cluster elastis dan ketersediaan versi.

Cluster global Amazon DocumentDB

Cluster global sekarang tersedia di kedua AWS GovCloud (US) wilayah: AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat).

10 Januari 2024

Fitur baru

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.4574, 3.0.4780, 3.0.4960)

- Ditambahkan dukungan untuk indeks vektor HNSW. Untuk informasi selengkapnya, lihat Pencarian vektor untuk Amazon DocumentDB.

- Menambahkan dukungan untuk indeks sebagian. Untuk informasi selengkapnya, lihat Indeks sebagian.
- Menambahkan dukungan untuk runtime GC pada koleksi dalam `currentOp` perintah.
- Menambahkan dukungan indeks teks untuk pencarian teks asli di Amazon DocumentDB. Untuk informasi selengkapnya, lihat Melakukan pencarian teks dengan Amazon DocumentDB.
- Ditambahkan dukungan untuk kata kunci `$jsonSchema` skematype,`allOf`,`oneOf`,`anyOf`,,`not`,`maxItems`,`minItems`,`maxProperties`,`minProperties`,`patte` `multipleOfdependencies`, danuniqueItems.

  Untuk mengetahui informasi selengkapnya, lihat Menggunakan validasi skema JSON.
- Ditambahkan dukungan untuk operator aritmatika`$ceil`,,,`$floor`,`$ln`, `$log`$log10, `$sqrt` dan. `$exp`

  Untuk mengetahui informasi selengkapnya, lihat Operator aritmatika.
- Ditambahkan dukungan untuk operator `$switch` ekspresi kondisional.
- Menambahkan dukungan untuk build indeks `IVFFLAT` vektor paralel. Dokumentasi diperbarui dengan menghapus batasan build indeks `IVFFLAT` vektor paralel dari panduan pengembang.

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.10124, 2.0.10179, 2.0.10221)

- Menambahkan dukungan untuk runtime GC pada koleksi dalam `currentOp` perintah.
- Ditambahkan dukungan untuk kata kunci `$jsonSchema` skematype,`allOf`,`oneOf`,`anyOf`,,`not`,`maxItems`,`minItems`,`maxProperties`,`minProperties`,`patte` `multipleOfdependencies`, danuniqueItems.

  Untuk mengetahui informasi selengkapnya, lihat Menggunakan validasi skema JSON.
- Ditambahkan dukungan untuk operator aritmatika`$ceil`,,,`$floor`,`$ln`, `$log`$log10, `$sqrt` dan. `$exp`

  Untuk mengetahui informasi selengkapnya, lihat Operator aritmatika.
- Ditambahkan dukungan untuk operator `$switch` ekspresi kondisional.

Perbaikan bug dan perubahan lainnya

- Menambahkan kembali fungsionalitas case-insensitive untuk pemanggilan. `db.runCommand("dbstats")` Pelanggan Amazon DocumentDB 5.0 dan 4.0 pada versi patch mesin sebelum 3.0.4960 atau 2.0.10221 harus menerapkan tambalan mesin terbaru ini.
- Memperbaiki kesalahan saat memanggil `db.coll.stats()` mongo shell versi 1.7 dan yang lebih baru. Dokumentasi diperbarui dengan menghapus tip `db.coll.stats()` pemecahan masalah shell mongo dari panduan pengembang.

20 Desember 2023

Perubahan lainnya

Dukungan yang diaktifkan untuk peningkatan versi utama di tempat di Amazon DocumentDB 3.6 dan 4.0. Untuk informasi selengkapnya, lihat Peningkatan versi utama Amazon DocumentDB di tempat.

13 Desember 2023

Fitur baru

Menambahkan dukungan untuk konektivitas EC2 1-klik. Untuk informasi selengkapnya, lihat Connect menggunakan Amazon EC2.

November 29, 2023

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.3727)

Fitur baru

Ditambahkan dukungan untuk pencarian vektor. Untuk informasi lebih lanjut, lihat posting blog ini dan kunjungi Pencarian vektor untuk Amazon DocumentDB di Panduan Pengembang Amazon DocumentDB.

21 November 2023

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.3727)

Fitur baru

Menambahkan dukungan untuk penyimpanan yang dioptimalkan I/O. Untuk informasi selengkapnya, lihat Konfigurasi penyimpanan klaster Amazon DocumentDB di Panduan Pengembang Amazon DocumentDB.

Menambahkan integrasi untuk pembelajaran mesin tanpa kode dengan SageMaker Canvas. Untuk informasi selengkapnya, lihat Pembelajaran mesin tanpa kode dengan Amazon Canvas SageMaker di Panduan Pengembang Amazon DocumentDB.

17 November 2023

Fitur baru

Amazon DocumentDB sekarang tersedia di wilayah (AWS GovCloud AS-Timur). Untuk informasi lebih lanjut, lihat posting blog ini.

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.208570)

Nama variabel lokal yang ditentukan pengguna sekarang mendukung “_” (garis bawah) untuk operator proyeksi seperti `$let` dan. `$filter`

6 November 2023

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.3727) dan 4.0 (Versi Patch Mesin 2.0.9876)

Fitur baru

- Ditambahkan dukungan untuk kata kunci `$jsonSchema` skema`maxLength`,`minLength`,`maximum`,`minimum`,,`exclusiveMaximum`, `exclusiveMinimumitems`, dan`additionalItems`.

  Harap dicatat bahwa validasi skema JSON hanya didukung di cluster berbasis instance.
- Menambahkan dukungan untuk operator pipa `$convert` agregasi dan operator turunan singkatan`$toBool`,,,`$toInt`,`$toLong`, `$toDouble` `$toString`$toDecimal, `$toObjectId` dan. `$toDate`

- Menambahkan dukungan untuk operator ekspresi set`$setDifference`,`$anyElementTrue`, dan`$allElementTrue`.

Perbaikan bug dan perubahan lainnya

Memperbaiki masalah saat pembaruan aliran perubahan dari `-NaN` ke `NaN` tidak ditampilkan.

20 Oktober 2023

Perubahan lainnya

Amazon DocumentDB telah mengidentifikasi masalah dan sementara melarang upgrade versi utama (MVU) di semua wilayah. Kami telah mengidentifikasi akar penyebab masalah dan telah mengembangkan perbaikan yang saat ini sedang menjalani pengujian. Kami mengantisipasi perbaikan ini akan diterapkan di semua wilayah sebelum akhir Q4 2023. MVU akan tetap dinonaktifkan hingga perbaikan diterapkan ke semua wilayah. Silakan periksa halaman catatan rilis ini untuk pembaruan lebih lanjut tentang ketersediaan fitur MVU.

Sementara itu, Anda dapat menggunakan AWS DMS untuk melakukan upgrade versi mayor dengan memigrasikan database Amazon DocumentDB Anda dari klaster versi yang lebih rendah ke versi yang lebih tinggi. Ikuti langkah-langkah Memutakhirkan cluster Amazon DocumentDB Anda menggunakan AWS Database Migration Service untuk meningkatkan penggunaan AWS DMS. Anda juga dapat merujuk ke posting blog ini untuk info tambahan tentang praktik terbaik untuk diikuti saat meningkatkan penggunaan AWS DMS.

25 September 2023

Fitur baru

Amazon DocumentDB sekarang tersedia di wilayah Asia Pasifik (Hong Kong). Untuk informasi lebih lanjut, lihat posting blog ini.

20 September 2023

Fitur baru

Menambahkan dukungan untuk peningkatan versi utama di tempat di Amazon DocumentDB 3.6 dan 4.0. Untuk mengetahui informasi selengkapnya, lihat Peningkatan versi utama Amazon DocumentDB di tempat.

15 September 2023

Fitur baru

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.3140) dan 4.0 (Versi Patch Mesin 2.0.9686)

- Menambahkan dukungan untuk validator skema $JSONSchema hanya dalam cluster berbasis instance.

  Untuk mengetahui informasi selengkapnya, lihat Menggunakan validasi skema JSON.

11 September 2023

Fitur baru

Amazon DocumentDB sekarang tersedia di wilayah Asia Pasifik (Hyderabad). Untuk informasi lebih lanjut, lihat posting blog ini.

3 Agustus 2023

Fitur baru

Amazon DocumentDB Cluster elastis

- Amazon DocumentDB Elastic cluster sekarang mendukung operasi berikut:
  - `top`
  - `collStats`
  - `hint`

- `dataSize`

  Lihat API, Operasi, dan Tipe Data MongoDB Yang Didukung daftar lengkap perintah dan operasi yang didukung.

- Indeks Time to Live (TTL) sekarang didukung.
- Indeks `hints` sekarang didukung dengan ekspresi indeks.

13 Juli 2023

Fitur baru

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.1948)

- Menambahkan dukungan untuk kompresi dokumen.
- Menambahkan dukungan untuk build indeks paralel.
- Menambahkan dukungan untuk status build indeks.

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.9259)

- Menambahkan dukungan untuk build indeks paralel.

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.1948)

- Memperbaiki masalah otentikasi `createCollection` untuk klaster elastis Amazon DocumentDB saat pengguna tidak memiliki akses ke koleksi sistem.
- Memperbaiki masalah saat instance wilayah sekunder tidak dapat menggunakan nama instans wilayah primer yang sama.

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.9259)

- Berhenti menambahkan kueri pemantauan internal ke log audit.

Juni 7, 2023

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 5.0

- instans r5 dan t3.medium sekarang didukung di Amazon DocumentDB 5.0.
- `engineVersion`opsi default ada `5.0.0` di AWS SDK, AWS CLI, dan AWS CloudFormation.

10 Mei 2023

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.1361)

- Ditambahkan dukungan untuk `ignoreunknownindexoptions` dalam createIndex perintah.
- Berhenti menambahkan kueri pemantauan internal ke log audit.
- Nama variabel lokal yang ditentukan pengguna sekarang mendukung "_" (garis bawah) untuk operator proyeksi seperti `$let` dan. `$filter`

4 April 2023

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.8934)

- Memperbaiki masalah dengan audit DHTML saat diaktifkan selama beban kerja yang sedang berlangsung.
- Memperbaiki masalah dengan audit DHTML saat perintah agregat dengan petunjuk diteruskan nilai string.
- Memperbaiki masalah dengan `listCollections` perintah yang tidak berfungsi saat pengguna dengan peran readwriteanydatabase memiliki opsi AuthorizedCollections dan NameOnly disetel ke true.
- Memperbaiki masalah untuk mengurai string numerik dengan benar dalam nama bidang.

- Batalkan kursor yang berjalan lama saat berdampak pada pengumpulan sampah.
- Nama variabel lokal yang ditentukan pengguna sekarang mendukung "_" (garis bawah) untuk operator proyeksi seperti `$let` dan. `$filter`

22 Maret 2023

Fitur baru

Cluster elastis Amazon DocumentDB sekarang tersedia di wilayah Asia Pasifik (Singapura), Asia Pasifik (Sydney), dan Asia Pasifik (Tokyo). Untuk informasi selengkapnya, lihat Wilayah cluster elastis dan ketersediaan versi.

1 Maret 2023

Fitur baru

Amazon DocumentDB 5.0 (Versi Patch Mesin 3.0.775)

- Memperkenalkan Amazon DocumentDB 5.0
  - Kompatibilitas MongoDB 5.0 (dukungan untuk driver MongoDB 5.0 API)
  - Support untuk Client-side Field Level Encryption (FLE). Anda sekarang dapat mengenkripsi bidang di sisi klien sebelum menulis data ke cluster Amazon DocumentDB. Untuk informasi selengkapnya, lihat Enkripsi tingkat bidang sisi klien
  - Operator agregasi baru:`$dateAdd`, `$dateSubtract`
- Peningkatan batas penyimpanan menjadi 128 TiB untuk semua cluster Amazon DocumentDB berbasis instans dan cluster elastis berbasis shard.
- Amazon DocumentDB 5.0 sekarang mendukung pemindaian indeks dengan operator `$elemMatch` di tingkat bersarang pertama. Pemindaian indeks didukung ketika kueri hanya memiliki satu tingkat `$elemMatch` filter dan `$elemMatch` kueri bersarang tidak mendukung pemindaian indeks.

  Bentuk kueri yang mendukung pemindaian indeks:

  ```
  db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
  ```

  Bentuk kueri yang tidak mendukung pemindaian indeks:

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e":
 "abc"} }} } })
```

27 Februari 2023

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 4.0

Menambahkan dukungan untuk AWS Lambda. Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda dengan Mengubah Aliran.

2 Februari 2023

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.208432)

- Memperbaiki masalah dengan audit DHTML saat diaktifkan selama beban kerja yang sedang berlangsung.
- Memperbaiki masalah dengan audit DHTML saat perintah agregat dengan petunjuk diteruskan nilai string.
- Memperbaiki masalah dengan `listCollections` perintah yang tidak berfungsi saat pengguna dengan peran readwriteanydatabase memiliki opsi AuthorizedCollections dan NameOnly disetel ke true.
- Memperbaiki masalah untuk mengurai string numerik dengan benar dalam nama bidang.
- Batalkan kursor yang berjalan lama saat berdampak pada pengumpulan sampah.

30 November 2022

Fitur baru

Amazon DocumentDB Cluster elastis

Amazon DocumentDB elastis cluster adalah jenis baru dari Amazon DocumentDB cluster yang memungkinkan pengguna untuk memanfaatkan API sharding MongoDB untuk skala cluster mereka.

Cluster elastis menangani hampir semua jumlah pembacaan dan penulisan dengan kapasitas penyimpanan petabyte dengan mendistribusikan data dan menghitung di beberapa instance dan volume komputasi yang mendasarinya. Untuk mempelajari selengkapnya, lihat Menggunakan cluster elastis Amazon DocumentDB.

Agustus 9, 2022

Fitur baru

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.208152) dan 4.0

- Ditambahkan dukungan untuk tipe data Decimal128. Decimal128 adalah tipe data BSON yang didukung di semua wilayah di mana DocumentDB tersedia.

  Untuk informasi selengkapnya, lihat jenis data DynamoDB.
- Menambahkan dukungan untuk audit kueri DHTML dengan Amazon CloudWatch Logs. Sekarang Amazon DocumentDB dapat merekam peristiwa Data Manipulation Language (DHTML) dan peristiwa Data Definition Language (DDL) ke Amazon Logs. CloudWatch

  Untuk informasi lebih lanjut, lihat posting blog ini.

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.208152) dan 4.0

- Anda sekarang dapat mengubah passwoprd Anda sendiri dengan kata sandi sendiri dengan hak istimewa. `changeOwnPassword`

25 Juli 2022

Fitur baru

Amazon DocumentDB 4.0

Anda sekarang dapat membuat cluster lebih cepat dengan kemampuan untuk membuat klon yang menggunakan volume cluster DocumentDB yang sama dan memiliki data yang sama dengan cluster asli. Untuk detailnya, lihat Mengelola Cluster Amazon DocumentDB.

Juni 27, 2022

Fitur baru

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.7509)

Amazon DocumentDB secara dinamis mengubah ukuran database Anda berdasarkan pola penggunaan. Menambahkan lebih banyak data meningkatkan ruang hingga 64 Tebibytes (TiB) dan menghapus data mengurangi ruang yang dialokasikan.

29 April 2022

Fitur baru

Amazon DocumentDB sekarang tersedia di wilayah China (Beijing). Untuk informasi lebih lanjut, lihat posting blog ini.

7 April 2022

Fitur baru

Amazon DocumentDB 3.6 (Versi Patch Engine 1.0.207836 dan 1.0.208015) dan 4.0 (Versi Patch Engine 2.0.6142 dan 2.0.6948)

Amazon DocumentDB Performance Insights sekarang dalam pratinjau. Anda sekarang dapat menyimpan tujuh hari riwayat kinerja di jendela bergulir tanpa biaya tambahan. Untuk informasi selengkapnya, lihat Monitoring with Performance Insights.

16 Maret 2022

Fitur baru

Amazon DocumentDB sekarang tersedia di wilayah Eropa (Milan). Untuk informasi lebih lanjut, lihat posting blog ini.

8 Februari 2022

Fitur baru

Instans Amazon DocumentDB R6g dan T4G sekarang tersedia di Asia Pasifik, Amerika Selatan, dan Eropa. Untuk informasi lebih lanjut, lihat posting blog ini.

24 Januari 2022

Fitur baru

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207684) dan 4.0 (Versi Patch Mesin 2.0.5170)

- DocDB; sekarang menawarkan uji coba gratis. Untuk detailnya, lihat halaman uji coba gratis Amazon DocumentDB.
- Anda sekarang dapat menggunakan fitur yang disempurnakan dengan kueri Geospasial, termasuk API berikut:
  - `$geoWithin`
  - `$geoIntersects`
- Ditambahkan dukungan untuk operator MongoDB berikut:
  - `$mergeObjects`
  - `$reduce`

  Untuk informasi selengkapnya, lihat Menanyakan data Geospasial dengan Amazon DocumentDB.

Januari 21, 2022

Fitur baru

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.5706)

- Amazon DocumentDB Graviton2 (r6g.large, r6g.2xlarge, r6g.4xlarge, r6g.8xlarge, r6g.12xlarge, r6g.16xlarge dan t4g.medium) instans sekarang didukung

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207781) dan 4.0 (Versi Patch Mesin 2.0.5706)

- Menambahkan dukungan untuk API MongoDB berikut:
  - `$reduce`
  - `$mergeObjects`
  - `$geoWithin`
  - `$geoIntersects`

25 Oktober 2021

Fitur baru

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207780) dan 4.0 (Versi Patch Mesin 2.0.5704)

- Menambahkan dukungan untuk API MongoDB berikut
  - `$literal`
  - `$map`
  - `$$ROOT`
- Support untuk kemampuan GeoSpatial Query. Lihat posting blog ini untuk lebih jelasnya
- Support untuk kontrol akses dengan peran yang ditentukan pengguna. Lihat posting blog ini untuk lebih jelasnya
- Driver Amazon DocumentDB JDBC untuk mengaktifkan konektivitas dari alat BI seperti Tableau dan alat kueri seperti SQL Workbench

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207780) dan 4.0 (Versi Patch Mesin 2.0.5704)

- Perbaikan bug `$natural` untuk mengurutkan dengan benar ketika eksplisit `.sort()` hadir bersama dengan `$natural`
- Perbaikan bug untuk aliran perubahan agar berfungsi `$redact`
- Perbaikan bug `$ifNull` untuk bekerja dengan array kosong
- Perbaikan bug untuk konsumsi sumber daya/kerusakan server yang berlebihan saat pengguna yang saat ini masuk dihapus atau hak istimewa pengguna untuk aktivitas yang sedang berlangsung dicabut
- Perbaikan bug `listDatabase` dan pemeriksaan `listCollection` hak istimewa

- Bug Fix dedupe logika untuk elemen multi-kunci

24 Juni 2021

Fitur baru

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207117) dan 4.0 (Versi Patch Mesin 2.0.3371)

- instans r5.8xlarge dan r5.16xlarge sekarang didukung. Pelajari lebih lanjut di posting blog Amazon DocumentDB Sekarang Mendukung Instans r5.8xlarge dan r5.16xlarge.
- Cluster global sekarang didukung untuk memberikan pemulihan bencana dari pemadaman di seluruh wilayah dan memungkinkan pembacaan global latensi rendah dengan memungkinkan pembacaan dari cluster Amazon DocumentDB terdekat.

4 Mei 2021

Fitur baru

Lihat semua fitur baru di posting blog ini.

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207117) dan 4.0 (Versi Patch Mesin 2.0.3371)

- `renameCollection`
- `$zip`
- `$indexOfArray`
- `$reverseArray`
- `$natural`
- `$hint`dukungan untuk pembaruan
- Pemindaian indeks untuk `distinct`

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.207117) dan 4.0 (Versi Patch Mesin 2.0.3371)

- Mengurangi penggunaan memori untuk `$in` kueri

- Memperbaiki kebocoran memori dalam indeks multikey
- Memperbaiki rencana penjelasan dan keluaran profiler untuk `$out`
- Menambahkan batas waktu untuk operasi dari sistem pemantauan internal untuk meningkatkan keandalan
- Memperbaiki cacat yang memengaruhi predikat kueri yang diteruskan ke indeks multikey

15 Januari 2021

Fitur baru

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.722)

- Tidak ada

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Kemampuan untuk menggunakan indeks dengan tahap `$lookup` agregasi
- `find()`kueri dengan proyeksi dapat disajikan arah dari indeks (kueri tertutup)
- Kemampuan untuk digunakan `hint()` dengan `findAndModify`
- Optimalisasi kinerja untuk operator `$addToSet`
- Perbaikan untuk mengurangi ukuran indeks keseluruhan
- Operator agregasi baru:`$ifNull`,`$replaceRoot`,,`$setIsSubset`, `$setInstersection``$setUnion`, dan `$setEquals`
- Pengguna juga dapat mengakhiri kursor mereka sendiri tanpa memerlukan peran `KillCursor`

9 November 2020

Fitur baru

Lihat semua fitur baru di posting blog ini.

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.722)

- Kompatibilitas MongoDB 4.0
- Transaksi ACID

- Support untuk `cluster(client.watch()` atau `mongo.watch())` dan aliran `(db.watch())` perubahan tingkat database
- Kemampuan untuk memulai atau melanjutkan aliran perubahan menggunakan `startAtOperationTime`
- Perpanjang periode retensi aliran perubahan Anda menjadi 7 hari (sebelumnya 24 jam)
- AWS DMS target untuk Amazon DocumentDB 4.0
- CloudWatch metrik:`TransactionsOpen`,,`TransactionsOpenMax`, `TransactionsAbortedTransactionsStarted`, dan `TransactionsCommitted`
- Bidang baru untuk transaksi di`currentOp`,`ServerStatus`, dan`profiler`.
- Kemampuan untuk menggunakan indeks dengan tahap `$lookup` agregasi
- `find()`kueri dengan proyeksi dapat disajikan arah dari indeks (kueri tertutup)
- Kemampuan untuk digunakan `hint()` dengan `findAndModify`
- Optimalisasi kinerja untuk operator `$addToSet`
- Perbaikan untuk mengurangi ukuran indeks secara keseluruhan.
- Operator agregasi baru:`$ifNull`,`$replaceRoot`,,`$setIsSubset`, `$setInstersection``$setUnion`, dan `$setEquals`
- Dengan `ListDatabase` perintah `ListCollection` dan, Anda sekarang dapat secara opsional menggunakan `authorizedDatabases` parameter `authorizedCollections` dan untuk memungkinkan pengguna untuk membuat daftar koleksi dan database yang mereka memiliki izin untuk mengakses tanpa memerlukan `listDatabase` peran `listCollections` dan, masing-masing
- Pengguna juga dapat mengakhiri kursor mereka sendiri tanpa memerlukan peran `KillCursor`
- Membandingkan jenis numerik subdokumen sekarang konsisten dengan membandingkan jenis numerik dokumen tingkat pertama. Perilaku di Amazon DocumentDB 4.0 sekarang kompatibel dengan MongoDB.

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Tidak ada

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 4.0 (Versi Patch Mesin 2.0.722)

- `$setOnInsert`tidak lagi mengizinkan pembaruan saat menggunakan operator $ posisi. Perilaku di Amazon DocumentDB 4.0 sekarang kompatibel dengan MongoDB.
- Memperbaiki masalah dengan `$createCollection` dan mengatur `autoIndexId`
- Proyeksi untuk dokumen bersarang
- Pengaturan default yang diubah untuk memori kerja agar diskalakan dengan ukuran memori instans
- Perbaikan pengumpulan sampah
- Pencarian dengan kunci kosong di jalur, perbedaan perilaku dengan mongo
- Memperbaiki `dateToString` bug dalam perilaku zona waktu
- Tetap `$push` (agregasi) untuk menghormati urutan pengurutan
- Memperbaiki bug `$currentOp` dengan agregat
- Memperbaiki masalah dengan `readPreference` di sekunder
- Masalah tetap dengan memvalidasi `$createIndex` adalah database yang sama dengan perintah yang dikeluarkan
- Memperbaiki perilaku yang tidak konsisten untuk`minKey`, `maxKey` pencarian gagal
- Memperbaiki masalah dengan `$size` operator yang tidak bekerja dengan array komposit
- Memperbaiki masalah dengan negasi `$in` dengan regex
- Memperbaiki masalah dengan `$distinct` perintah yang dijalankan terhadap tampilan
- Memperbaiki masalah dengan agregasi dan menemukan perintah yang menyortir bidang yang hilang secara berbeda
- Tetap `$eq` ke ekspresi reguler yang tidak memeriksa jenis
- Memperbaiki `$currentDate` bug dalam perilaku posisi ordinal stempel waktu
- Granularitas milidetik tetap untuk `$currentDate`

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Tidak ada

30 Oktober 2020

Fitur baru

Lihat semua fitur baru di posting blog ini.

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Ditambahkan kemampuan untuk membuka kursor aliran perubahan di tingkat cluster (`client.watch()` atau `mongo.watch()`) dan database (`db.watch()`)
- Kemampuan untuk meningkatkan periode retensi aliran perubahan menjadi 7 hari (sebelumnya 24 jam)

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Berbagai peningkatan kinerja kasus umum
- Peningkatan keamanan yang ditargetkan
- Memperbaiki masalah dengan skip sort pada bidang kedua dari indeks gabungan
- Aktifkan indeks reguler untuk kesetaraan pada bidang tunggal indeks multi-kunci (bukan gabungan)
- Kondisi balapan otentikasi tetap
- Memperbaiki masalah yang menyebabkan kecelakaan pengumpulan sampah yang jarang terjadi
- Peningkatan keamanan RBAC
- Ditambahkan `databaseConnectionsMax` metrik
- Peningkatan kinerja untuk beban kerja tertentu pada instans `r5.24xlarge`

22 September 2020

Fitur baru

Lihat semua fitur baru di posting blog ini.

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- `$outtahap` agregasi
- Meningkatkan jumlah maksimum koneksi dan kursor per instance sebanyak 10x

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Tidak ada

10 Juli 2020

Fitur baru

Lihat semua fitur baru di posting blog ini.

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Salinan Snapshot Lintas Wilayah

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Tidak ada

30 Juni 2020

Fitur baru

Lihat semua fitur baru di posting blog ini.

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Contoh menengah T3

Perbaikan bug dan perubahan lainnya

Amazon DocumentDB 3.6 (Versi Patch Mesin 1.0.206295)

- Reklamasi memori idle untuk instance t3
- Peningkatan otentikasi

- Peningkatan kinerja otentikasi SASL
- Memperbaiki `currentOp` masalah saat melebihi operasi maksimum yang mungkin
- `killOps`Masalah tetap untuk pembaruan massal dan hapus
- Perbaikan `$sample` kinerja dengan `$match`
- Dukungan tetap untuk $$ kasus cond dalam tahap redact
- Memperbaiki berbagai penyebab crash root yang berulang
- Perbaikan penyapuan TTL untuk mengurangi iOS dan latensi
- Pemanfaatan memori yang dioptimalkan untuk `$unwind`
- Kondisi balapan statistik koleksi tetap dengan indeks drop
- Kondisi balapan tetap selama pembuatan indeks bersamaan
- Memperbaiki crash yang jarang terjadi `hash_search` dalam indeks

Riwayat Dokumen untuk Panduan Developer Amazon DocumentDB

- Versi API: 2014-10-31
- Pembaruan dokumentasi terbaru: 2 Juni 2023

Tabel berikut menguraikan dokumentasi untuk rilis Panduan Developer Amazon DocumentDB ini.

Perubahan	Deskripsi	Tanggal
AWS pembaruan kebijakan terkelola - perubahan kebijakan	Amazon DocumentDB memperbarui kebijakan akses penuh untuk cluster elastis.	Februari 21, 2024
AWS pembaruan kebijakan terkelola - perubahan kebijakan	Amazon DocumentDB memperbarui kebijakan read-only dan akses penuh untuk cluster elastis.	Juni 21, 2023
AWS pembaruan kebijakan terkelola - kebijakan baru	Amazon DocumentDB memperkenalkan kebijakan hanya-baca baru untuk cluster elastis.	8 Juni 2023
AWS pembaruan kebijakan terkelola - kebijakan baru	Amazon DocumentDB memperkenalkan kebijakan akses penuh baru untuk cluster elastis.	Juni 5, 2023
MongoDB 5.0 Kompatibilitas	Amazon DocumentDB sekarang kompatibel dengan versi 5.0 dari MongoDB.	1 Maret 2023
Pembaruan Kebijakan	Untuk mendukung fitur cluster elastis Amazon DocumentD B AmazonDoc, kebijakan	30 November 2022

	ConsoleFullAccess DB diperbarui, dan DB ElasticSe rviceRolePolicy - AmazonDoc diperkenalkan.	
Cluster Elastis	Menambahkan fitur Elastic Cluster baru yang mendukung partisi berbasis hash (sharding) data di seluruh sistem penyimpanan terdistribusi Amazon DocumentDB.	30 November 2022
Cluster Global	Menambahkan dokumentasi tentang cara menggunakan Klaster Global.	2 Juni 2021
Langganan Acara	Menambahkan dokumentasi langganan peristiwa.	26 Maret 2021
Versi 3.6 Upgrade	Perbaikan terdokumentasi ke versi 3.6 dalam kontrol akses berbasis peran, operator agregasi, dan performa.	15 Januari 2021
MongoDB 4.0 Kompatibilitas	Amazon DocumentDB kini kompatibel dengan MongoDB versi 4.0.	9 November 2020
Panduan Memulai	Panduan Memulai Baru untuk memulai menggunakan Amazon DocumentDB, AWS Cloud9 Amazon EC2, Robo3T atau Studio3T.	15 Agustus 2020

Zona Ketersediaan Tambahan didukung	Amazon DocumentDB menambahkan dukungan untuk Availability Zone tambahan di Asia Pacific (Seoul) (ap-northeast-2).	14 Juli 2020
Menambahkan dukungan untuk menyalin snapshot di seluruh Wilayah.	Amazon DocumentDB menambahkan dukungan untuk menyalin snapshot klaster di seluruh Wilayah AWS. Untuk informasi lebih lanjut, lihat Menyalin Snapshot di Seluruh Wilayah.	10 Juli 2020
Ditambahkan dukungan untuk kelas contoh T3.	Menambahkan dukungan untuk tipe instans T3 di semua Wilayah yang mendukung Amazon DocumentDB. Untuk informasi selengkapnya, lihat Kelas Instans yang Didukung berdasarkan Wilayah dan Spesifikasi Kelas Instans.	30 Juni 2020
Menambahkan dukungan untuk AWS GovCloud (US).	Amazon DocumentDB sekarang tersedia di Wilayah us-gov-west (-1 AWS GovCloud (US)).	29 Juni 2020
Menambahkan 16 CloudWatch metrik baru.	Amazon DocumentDB menambahkan dukungan untuk 16 metrik Amazon baru. CloudWatch Untuk informasi selengkapnya, lihat Memantau Amazon CloudWatch DocumentDB dengan.	23 Juni 2020

Ditambahkan dukungan untuk karakter null dan $ regex operator.	Amazon DocumentDB menambahkan dukungan untuk karakter null dalam string dan kemampuan untuk menggunakan sebuah indeks untuk $regex. Untuk melihat API MongoDB dan kemampuan alur agregasi yang didukung untuk Amazon DocumentDB, lihat Perbedaan Fungsional dengan MongoDB.	22 Juni 2020
Menambahkan dukungan untuk meningkatkan kemampuan pengindeksan multi-kunci.	Amazon DocumentDB menambahkan dukungan untuk kemampuan pengindeksan multi-kunci yang ditingkatkan yang mencakup pengindeksan array yang lebih besar dari 2.048 byte dan kemampuan untuk membuat indeks multi-kunci campuran dengan beberapa kunci dalam array yang sama. Untuk informasi selengkapnya, lihat Perbedaan Fungsional dengan MongoDB.	23 April 2020
Menambahkan dukungan untuk perlindungan penghapusan untuk tumpukan Amazon DocumentDB. AWS CloudFormation	Amazon DocumentDB menambahkan dukungan untuk mengaktifkan perlindungan penghapusan saat membuat tumpukan Amazon DocumentDB. AWS CloudFormation	20 April 2020

Menambahkan dukungan untuk kontrol akses berbasis peran.	Amazon DocumentDB menambahkan dukungan untuk kontrol akses berbasis peran menggunakan peran bawaan.	26 Maret 2020
Menambahkan dukungan untuk Availability Zone tambahan di Kanada (Tengah) (ca-central-1).	Amazon DocumentDB kini tersedia di Wilayah Canada (Central) (ca-central-1) dengan instans kelas R5 dan 3 Availability Zone.	26 Maret 2020
Menambahkan dukungan untuk dua API MongoDB tambahan.	Amazon DocumentDB menambahkan dukungan untuk API MongoDB `$dateFromString` dan `executionStats` .	23 Maret 2020
Menambahkan dukungan untuk lima API MongoDB tambahan.	Amazon DocumentDB menambahkan dukungan untuk API MongoDB `$objectToArray` , `$arrayToObject` , `$slice`, `$mod`, dan `$range`.	6 Februari 2020
Menambahkan dukungan untuk Kanada (Tengah).	Amazon DocumentDB kini tersedia di Wilayah Canada (Central) (ca-central-1) dengan instans kelas R5.	11 Desember 2019
Menambahkan dukungan untuk ChangeStreamLogSize.	Amazon DocumentDB menambahkan dukungan untuk `ChangeStreamLogSize` untuk metrik Cloudwatch.	22 November 2019

Menambahkan dukungan untuk wilayah Eropa (Paris)	Amazon DocumentDB kini tersedia di wilayah Europe (Paris) (eu-west-3) dengan instans kelas R5.	30 Oktober 2019
Menambahkan dukungan untuk wilayah Asia Pasifik (Mumbai)	Amazon DocumentDB kini tersedia di wilayah Asia Pacific (Mumbai) (ap-south-1) dengan instans kelas R5.	17 Oktober 2019
Menambahkan dukungan untuk tiga API MongoDB tambahan	Amazon DocumentDB menambahkan dukungan untuk API MongoDB `$addFields` , `$concatArrays` , dan `$lookup`.	16 Oktober 2019
Menambahkan dukungan untuk wilayah Asia Pasifik (Singapura)	Amazon DocumentDB kini tersedia di wilayah Asia Pacific (Singapore) (ap-southeast-1) dengan instans kelas R5.	14 Oktober 2019
Menambahkan dokumen baru untuk memperbarui sertifikat TLS	Menambahkan instruksi untuk memperbarui sertifikat CA guna menggunakan sertifikat CA baru untuk membuat koneksi TLS.	2 Oktober 2019
Menambahkan dukungan API untuk sertifikat	Amazon DocumentDB tipe data Sertifikat baru untuk instans. Untuk informasi selengkapnya, lihat DBInstance.	1 Oktober 2019
Support untuk pembuatan profil kueri	Amazon DocumentDB menambahkan kemampuan untuk membuat profil operasi yang didukung pada instans dan basis data klaster Anda.	19 Agustus 2019

Menambahkan AZ ketiga di Asia Pasifik (Tokyo)	Amazon DocumentDB menambahkan Availability Zone (AZ) ketiga untuk instans komputasi Anda di Asia Pacific (Tokyo).	9 Agustus 2019
Support untuk API Mongo tambahan	Menambahkan dukungan untuk kemampuan alur agregasi tambahan yang mencakup operator agregasi `$in`, `$isoWeek`, `$isoWeekYear` , `$isoDayOfWeek` , dan `$dateToString` serta tahap agregasi `$addToSet` . Amazon DocumentDB juga menambahkan dukungan untuk perintah `top()` guna diagnostik tingkat koleksi dan kemampuan untuk memodifikasi parameter `expireAfterSeconds` untuk pengindeksan TTL menggunakan perintah `collMod()` .	31 Juli 2019
Menambahkan dukungan untuk Eropa (London)	Amazon DocumentDB kini tersedia di Europe (London) (eu-west-2) dengan instans kelas R5.	18 Juli 2019
Contoh kode yang ditambahkan	Menambahkan contoh kode di R dan Ruby untuk menghubungkan secara terprogram ke Amazon DocumentDB.	17 Juli 2019

Menambahkan praktik terbaik	Menambahkan Praktik Terbaik untuk membantu Anda mengelola biaya Amazon DocumentDB Anda.	17 Juli 2019
Support untuk menghentikan dan memulai cluster	Amazon DocumentDB menambahkan dukungan untuk menghentikan dan memulai klaster guna mengelola biaya pengembangan dan lingkungan pengujian.	1 Juli 2019
Support untuk perlindungan penghapusan klaster	Untuk melindungi klaster Anda dari penghapusan yang tidak disengaja, Amazon DocumentDB menambahkan perlindungan penghapusan. Untuk informasi selengkapnya, lihat topik berikut: Membuat Klaster Amazon DocumentDB, Memodifikasi Klaster Amazon DocumentDB, Menghapus Klaster Amazon DocumentDB, dan `DeletionProtection` dalam topik API DBCluster.	1 Juli 2019
Pembaruan perbedaan fungsional	Menambahkan Transaksi Implisit ke Perbedaan Fungsional.	26 Juni 2019
Penambahan perbedaan fungsional	Menambahkan catatan mengenai penyimpanan dan kompresi indeks di Amazon DocumentDB.	13 Juni 2019

Wilayah tambahan didukung	Amazon DocumentDB kini tersedia di wilayah Asia Pacific (Sydney) (ap-southeast-2) dengan instans kelas R5.	5 Juni 2019
Kelas instans R5 didukung di wilayah tambahan	Menambahkan dukungan kelas instans R5 untuk 4 wilayah tambahan: US East (Ohio), US East (N. Virginia), US West (Oregon), dan EU (Ireland). Dengan perubahan ini, instans R5 didukung di semua wilayah yang mendukung Amazon DocumentDB.	17 Mei 2019
Wilayah tambahan yang didukung	Menambahkan dukungan untuk 2 wilayah tambahan, Asia Pacific (Tokyo) (ap-northeast-1) dan Asia Pacific (Seoul) (ap-northeast-2) dengan kelas instans R5. Untuk informasi selengkap nya, lihat Kelas Instans yang Didukung berdasarkan Wilayah dan Spesifikasi Kelas Instans.	8 Mei 2019
Menambahkan lebih banyak contoh kode koneksi	Menambahkan contoh kode di Java dan C# untuk menghubungkan ke Amazon DocumentDB.	24 April 2019

Dukungan Mongo API tambahan	Menambahkan dukungan untuk tujuh operator string agregasi (`$indexOfBytes`, `$indexOfCP`, `$strLenBytes`, `$strLenCP`, `$toLower`, `$toUpper`, dan `$split`), sembilan operator tanggal-waktu (`$dayOfYear`, `$dayOfMonth`, `$dayOfWeek`, `$year`, `$month`, `$hour`, `$minute`, `$second`, dan `$millisecond`), dan tahap alur agregasi `$sample`.	4 April 2019
Ditambahkan contoh kode koneksi	Menambahkan contoh kode dalam Python, Node.js, PHP, dan Go untuk menghubungkan ke Amazon DocumentDB.	21 Maret 2019
Support untuk Wilayah Frankfurt dan instans R5	Dukungan tambahan untuk Wilayah Europe (Frankfurt) (eu-central-1) dengan kelas instans R5. Untuk informasi selengkapnya, lihat Kelas Instans yang Didukung berdasarkan Wilayah dan Spesifikasi Kelas Instans.	13 Maret 2019

Dukungan operator pipa agregasi	Menambahkan dukungan untuk operator string agregasi baru (`$concat`, `$substr`, `$substrBytes` , `$substrCP` , `$strcasecmp`), sebuah operator agregasi array (`$size`), sebuah operator akumulator grup agregasi (`$push`), dan tahap agregasi (`$redact` dan `$indexStats`). Kami juga menambahkan dukungan untuk operator array posisional (`$[]` dan `$[<identifier>]`) dan `hint()`.	28 Februari 2019
Peningkatan mesin	Menambahkan dokumentasi untuk menentukan modifikasi klaster yang tertunda dan meningkatkan versi mesin klaster Anda.	15 Februari 2019
Acara audit	Menambahkan dukungan untuk mengaudit peristiwa database dengan Amazon CloudWatch Logs.	12 Februari 2019
Mulai Cepat	Menambahkan topik Mulai Cepat untuk membantu Anda memulai dengan mudah menggunakan Amazon AWS CloudFormation DocumentDB.	11 Januari 2019

Rilis Publik	Ini adalah rilis publik awal Amazon DocumentDB (dengan kompatibilitas MongoDB). Rilis ini mencakup Panduan Developer dan Referensi API Manajemen Sumber Daya terintegrasi.	9 Januari 2019

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.