
Amazon DocumentDB

开发人员指南

亚马逊云科技



Amazon DocumentDB: 开发人员指南

Copyright © 2022 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其它商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅[中国的 Amazon Web Services 服务入门](#)。

Table of Contents

什么是 Amazon DocumentDB	1
概览	1
集群	2
实例	2
区域和可用区	3
区域	4
可用区	4
定价	5
监控	5
接口	5
Amazon Web Services Management Console	5
Amazon CLI	6
mongo Shell	6
MongoDB 驱动程序	6
接下来做什么？	6
工作方式	6
Amazon DocumentDB D	7
TLS Support	9
Amazon DocumentDB D	9
Amazon DocumentDB D	10
Amazon DocumentDB D	10
读取首选选项	11
TTL 删除	13
应计费的资源	13
什么是文档数据库？	15
使用案例	16
了解文档	16
使用文档	20
入门指南	28
先决条件	28
第 1 步：创建 Amazon Cloud9 环境	29
第 2 步：创建安全组	31
第 3 步：创建 Amazon DocumentDB 集群	34
第 4 步：安装 mongo shell	36
第 5 步：Connect 到您的 Amazon DocumentDB 集群	37
第 6 步：插入和查询数据	38
步骤 7：了解	39
使用 Amazon CloudFormation 的快速入门	40
先决条件	40
所需的 IAM 权限	40
Amazon EC2 密钥对	41
启动 Amazon DocumentDB Amazon CloudFormation 堆栈	42
访问 Amazon DocumentDB 集群	45
终止保护和删除保护	45
MongoDB 4.0 兼容性	46
Amazon DocumentDB 4.0 中的新增功能	46
开始使用 Amazon DocumentDB 4.0	47
升级或迁移到 Amazon DocumentDB 4.0	47
功能差异	47
Amazon DocumentDB 3.6 和 4.0 之间的功能差异	47
Amazon DocumentDB 4.0 和 MongoDB 4.0 之间的功能差异	47
事务	49
要求	49
最佳实践	49

限制	49
监控和诊断	50
事务隔离级别	50
使用案例	51
多对账单交易	51
多项收款交易	52
回调 API 的交易 API 示例	53
核心 API 的交易 API 示例	53
支持的 命令	75
不支持的功能	75
会话	76
因果一致性	76
可重试写入	77
事务错误	77
最佳实践	78
基本操作指导方针	78
调整实例大小	79
使用索引	79
建立索引	79
索引选择性	80
索引对数据写入的影响	80
识别缺失的索引	80
识别未使用的索引	80
安全最佳实践	80
成本优化	81
使用指标确定性能问题	81
查看性能指标	81
设置 CloudWatch 警报	81
评估性能指标	81
优化查询	82
TTL 和时间序列工作负载	83
迁移	83
使用集群参数组	83
聚合管道查询	83
batchInsert 和 batchUpdate	83
与 MongoDB 之间的功能差异	84
Amazon DocumentDB B 的功能优势	84
隐式事务	84
更新的功能差异	85
数组索引	85
多键索引	86
字符串中的 Null 字符	86
基于角色的访问控制	86
\$regex 索引	86
嵌套文档的投影	87
与 MongoDB 之间的功能差异	87
管理数据库和集合	87
cursorMaxTimeMS	87
explain()	88
字段名称限制	88
索引构建	88
在路径中使用空键查找	88
MongoDB API、操作和数据类型	89
mongodump和mongorestore实用工具	89
结果排序	89
可重试写入	89
稀疏索引	89

存储压缩	90
在 \$all 表达式中使用 \$elemMatch	90
\$ne,\$nin,\$nor,\$not,\$exists , 以及\$elemMatch索引	90
\$lookup	90
支持的 MongoDB API、操作和数据类型	93
数据库命令	93
管理命令	93
聚合	94
身份验证	95
诊断命令	95
查询和写入操作	95
角色管理命令	96
会话命令	96
用户管理	97
查询和投影运算符	97
数组运算符	97
按位运算符	97
注释运算符	98
比较运算符	98
元素运算符	98
评估查询运算符	98
逻辑运算符	99
投影运算符	99
更新运算符	99
数组运算符	99
按位运算符	100
字段操作符	100
更新修改器	100
地理空间	101
几何体说明符	101
查询选择器	101
游标方法	101
聚合管道运算符	102
累加器表达式	103
算术运算符	103
数组运算符	104
布尔运算符	105
比较运算符	105
条件表达式运算符	105
数据类型运算符	105
日期运算符	105
文字运算符	106
合并运算符	106
自然运算符	106
集合运算符	107
阶段操作符	107
字符串运算符	108
系统变量	108
文本搜索运算符	109
类型转换运算符	109
变量运算符	109
数据类型	109
索引和索引属性	110
索引	110
索引属性	111
迁Amazon DocumentDB	112
在版本之间迁移	112

第 1 步：启用变更流	113
第 2 步：修改更改流保留期限	113
第 3 步：迁移 Migration	113
第 4 步：创建 Amazon DMS 复制实例	114
第 5 步：创建 Amazon DMS 源终端节点	115
第 6 步：创建 Amazon DMS 目标终端节点	117
步骤 7：创建和运行迁移任务	118
步骤 8：将应用程序终端节点更改为 Amazon DocumentDB 群集 4.0	119
迁移工具	119
Amazon Database Migration Service	120
命令行实用程序	120
发现	120
规划：Amazon DocumentDB 集群要求	122
迁移方法	124
离线	124
在线	125
混合	126
迁移源	127
迁移连接	128
测试	130
迁移计划测试注意事项	130
性能测试	131
故障转移测试	132
其他资源	132
安全性	133
数据保护	133
加密 静态数据	134
加密传输中的数据	136
密钥管理	142
Identity and Access Management	142
身份验证	142
访问管理概述	143
使用策略管理访问	146
使用基于身份的策略 (IAM 策略)	146
Amazon DocumentDB API 权限参考	149
Amazon DocumentDB 用户	154
主用户和 serviceadmin 用户	154
创建其他用户	154
自动轮换密码	156
基于角色的访问控制	156
RBAC 概念	156
RBAC 内置角色入门	158
RBAC 用户定义的角色入门	160
以用户身份连接到 Amazon DocumentDB	163
常见命令	164
功能差异	167
限制	167
基于角色的访问控制限制数据库访问	168
日志记录和监控	172
更新证书 (cn-northwest-1)	173
更新您的应用程序和 Amazon DocumentDB 集群	173
常见问题	175
更新您的应用程序和 Amazon DocumentDB 集群	173
常见问题	175
合规性验证	183
故障恢复能力	184

基础设施安全性	184
安全最佳实践	185
审核 事件	185
支持的事件	185
启用审核	187
禁用审核	191
访问审计事件	193
备份和还原	194
备份和还原：概念	194
了解备份存储使用量	195
转储、还原、导入和导出数据	196
mongodump	196
mongorestore	197
mongoexport	197
mongoimport	197
教程	198
集群快照注意事项	199
备份存储	200
备份时段	200
备份保留期	200
比较自动快照和手动快照	201
创建手动集群快照	201
使用 Amazon Web Services Management Console 创建集群快照	202
使用 Amazon CLI 创建集群快照	203
复制集群快照	203
复制共享快照	204
跨复制快照Amazon Web Services 区域	204
限制	204
处理加密	204
参数组注意事项	205
复制集群快照	205
共享集群快照	209
共享加密的快照	210
共享快照	212
从集群快照还原	213
使用 Amazon Web Services Management Console 从集群快照进行还原	213
使用 Amazon CLI 从集群快照进行还原	214
还原到某个时间点	217
使用 Amazon Web Services Management Console还原到时间点	217
使用 Amazon CLI 还原到时间点	219
删除集群快照	221
使用 Amazon Web Services Management Console删除集群快照	221
使用 Amazon CLI删除集群快照	221
管理 Amazon DocumentDB	222
操作任务概述	222
给 Amazon DocumentDB 集群添加副本	222
描述集群和实例	223
创建集群快照	224
从快照还原	225
从集群中删除实例	226
删除集群	226
全局群集	227
什么是全局群集？	227
全局集群有何用处？	227
全球集群当前的局限性是什么？	227
快速入门指南	227
管理全局集群	236

连接全局集群	240
监控全局集群	241
灾难恢复	241
管理集群	243
了解集群	243
集群设置	244
确定集群的状态	245
集群生命周期	246
扩展Amazon DocumentDB	271
克隆 Amazon DocumentDB 集群卷	272
了解容错能力	282
管理实例	282
管理实例类	283
确定实例的状态	288
实例生命周期	288
管理子网组	303
创建子网组	303
描述子网组	307
修改子网组	308
删除子网组	310
高可用性和复制	311
读取扩展	312
高可用性	312
添加 副本	313
故障转移	313
复制滞后	316
管理事件	316
查看 事件类别	317
查看 Amazon DocumentDB 事件	318
选择区域和可用区	320
区域可用性	321
管理集群参数组	322
描述集群参数组	322
创建集群参数组	327
修改群集参数组	328
修改集群以使用自定义集群参数组	331
复制集群参数组	332
重置集群参数组	334
删除集群参数组	335
集群参数参考	337
了解端节点	343
查找集群的终端节点	344
查找实例的终端节点	345
连接到 终端节点	348
了解 Amazon DocumentDB ARN	349
构建 ARN	349
查找 ARN	350
为资源添加标签	352
资源标签概述	352
标签约束	353
添加或更新标签	353
列出标签	354
删除标签	355
维护 Amazon DocumentDB	356
确定待处理的 维护操作	356
应用更新	357
用户启动的更新	360

管理 维护时段	360
了解服务相关角色	361
服务相关角色权限	362
创建服务相关角色	363
修改服务相关角色	363
删除服务相关角色	363
Amazon DocumentDB 服务相关角色支持的区域	364
监控 Amazon DocumentDB	365
监控集群状态	366
集群状态值	366
使用Amazon Web Services Management Console监控集群状态	366
使用Amazon CLI监控集群状态	367
监控 实例的状态	368
实例状态值	368
使用Amazon Web Services Management Console监控实例的状态	369
使用Amazon CLI监控实例的状态	370
事件订阅	370
订阅事件	371
管理订阅	372
类别和消息	374
使用 CloudWatch 监控 Amazon DocumentDB	376
Amazon DocumentDB 指标	376
查看 CloudWatch 数据	382
Amazon DocumentDB 维度	386
监控 Opcount	386
监控数据库连接	387
使用 CloudTrail 记录 Amazon DocumentDB API 调用	387
CloudTrail 中的 Amazon DocumentDB 信息	387
分析 操作	388
支持的 操作	388
限制	388
启用 分析器	389
禁用 分析器	391
禁用分析器日志导出	392
访问 分析器日志	394
常见查询	394
使用 Performance Insights 进行监控	394
Performance Insights 概念	395
启用和禁用 Performance Insights	397
为 Performance Insights 配置访问策略	399
使用性能详情控制面板分析指标	402
使用性能详情 API 检索指标	416
亚马逊 CloudWatch Performance Insights 的指标	425
Performance Insights 的计数器指标	426
使用 Amazon DocumentDB 进行开发	428
以编程方式连接	428
确定 tls 值	428
启用了 TLS 的情况下的连接	430
禁用了 TLS 的情况下的连接	439
使用变更流	445
支持的 操作	445
计费	445
限制	446
启用变更流	446
示例	447
完整文档查找	449
恢复变更流	449

使用恢复变更流startAtOperationTime	451
变更流中的交易	452
修改更改流日志保留持续时间	452
作为副本集进行连接	454
使用群集连接	456
多个连接池	456
摘要	457
从 Amazon VPC 外部连接	457
使用 Robo 3T 进行 Connect	458
先决条件	458
Connect Robo 3T 联系	458
使用 Studio 3T 进行 Connect	461
先决条件	458
Connect 工作室 3T	461
使用Amazon EC2 Connect	466
先决条件	466
第 1 步：创建 Amazon EC2 实例	467
第 2 步：创建安全组	471
第 3 步：创建 AAmazon DocumentDB 集群	473
第 4 步：Connect 您的 Amazon EC2 实例	475
第 5 步：安装 mongo shell	476
第 6 步：管理Amazon DocumentDB T	477
步骤 7：Connect 您的 Amazon DocumentDB 集群	477
第 8 步：插入和查询数据	38
第 9 步：探索	480
使用 JDBC 驱动程序 Connect	480
入门	480
从 Tableau 桌面进行 Connect	481
从 DBVisualizer Connect	487
自动生成模式	489
SQL Support 和限制	495
故障排除	495
配额和限制	496
支持的 实例类型	496
支持的区域	497
区域配额	497
聚合限制	499
集群限制	499
实例限制	500
命名约束	502
TTL 约束	503
查询	504
查询文档	504
检索所有文档	504
匹配字段值	504
嵌入文档	505
嵌入式文档中的字段值	505
匹配数组	505
匹配数组中的值	505
使用运算符	505
地理空间数	506
概览	1
索引和存储地理空间数据	506
查询地理空间数据	507
限制	509
查询计划	509
查询计划	509

查询计划缓存	510
EXplain 结果	510
扫描和筛选阶段	511
索引交叉	512
索引	512
多指数交口/联盟	512
复合索引	513
排序阶段	513
阶段	513
故障排除	515
连接问题	515
无法 Connect 到 Amazon DocumentDB 终端节点	515
测试与 Amazon DocumentDB 实例的连接	517
连接到无效终端节点	517
索引创建	518
索引构建失败	518
后台索引构建延迟问题和失败	518
性能和资源利用率	519
查找并终止长时间运行或受阻的查询	519
查看查询计划和优化查询	520
列出实例上正在运行的所有操作	521
知道查询何时进行	523
确定为什么系统突然运行缓慢	524
确定 CPU 使用率过高的原因	525
如何确定实例上的打开游标？	526
如何确定当前的 Amazon DocumentDB 引擎版本？	526
如何识别未使用的索引？	526
如何识别缺失的索引？	527
有用查询的摘要	528
资源管理 API 参考	529
操作	529
AddSourceIdentifierToSubscription	531
AddTagsToResource	533
ApplyPendingMaintenanceAction	535
CopyDBClusterParameterGroup	537
CopyDBClusterSnapshot	539
CreateDBCluster	542
CreateDBClusterParameterGroup	548
CreateDBClusterSnapshot	550
CreateDBInstance	552
CreateDBSubnetGroup	557
CreateEventSubscription	559
CreateGlobalCluster	562
DeleteDBCluster	565
DeleteDBClusterParameterGroup	567
DeleteDBClusterSnapshot	568
DeleteDBInstance	570
DeleteDBSubnetGroup	572
DeleteEventSubscription	574
DeleteGlobalCluster	576
DescribeCertificates	578
DescribeDBClusterParameterGroups	580
DescribeDBClusterParameters	582
DescribeDBClusters	584
DescribeDBClusterSnapshotAttributes	586
DescribeDBClusterSnapshots	588
DescribeDBEngineVersions	591

DescribeDBInstances	594
DescribeDBSubnetGroups	596
DescribeEngineDefaultClusterParameters	598
DescribeEventCategories	600
DescribeEvents	601
DescribeEventSubscriptions	604
DescribeGlobalClusters	606
DescribeOrderableDBInstanceOptions	608
DescribePendingMaintenanceActions	610
FailoverDBCluster	612
ListTagsForResource	614
ModifyDBCluster	616
ModifyDBClusterParameterGroup	621
ModifyDBClusterSnapshotAttribute	623
ModifyDBInstance	625
ModifyDBSubnetGroup	629
ModifyEventSubscription	631
ModifyGlobalCluster	633
RebootDBInstance	635
RemoveFromGlobalCluster	637
RemoveSourceIdentifierFromSubscription	639
RemoveTagsForResource	641
ResetDBClusterParameterGroup	643
RestoreDBClusterFromSnapshot	645
RestoreDBClusterToPointInTime	650
StartDBCluster	655
StopDBCluster	657
数据类型	658
AvailabilityZone	659
Certificate	660
CloudwatchLogsExportConfiguration	662
DBCluster	663
DBClusterMember	668
DBClusterParameterGroup	669
DBClusterRole	670
DBClusterSnapshot	671
DBClusterSnapshotAttribute	674
DBClusterSnapshotAttributesResult	675
DBEngineVersion	676
DBInstance	678
DBInstanceStatusInfo	682
DBSubnetGroup	683
Endpoint	685
EngineDefaults	686
Event	687
EventCategoriesMap	689
EventSubscription	690
Filter	692
GlobalCluster	693
GlobalClusterMember	695
OrderableDBInstanceOption	696
Parameter	698
PendingCloudwatchLogsExports	700
PendingMaintenanceAction	701
PendingModifiedValues	703
ResourcePendingMaintenanceActions	706
Subnet	707

Tag	708
UpgradeTarget	709
VpcSecurityGroupMembership	710
常见错误	710
常见参数	712
发行说明	714
2022 年 1 月 21 日	714
新功能	714
2021 年 10 月 25 日	714
新功能	714
错误修复和其他更改	715
2021 年 6 月 24 日	715
新功能	715
2021 年 5 月 4 日	715
新功能	715
错误修复和其他更改	716
2021 年 1 月 15 日	716
新功能	716
2020 年 11 月 9 日	716
新功能	716
错误修复和其他更改	717
2020 年 10 月 30 日	718
新功能	718
错误修复和其他更改	718
2020 年 9 月 22 日	718
新功能	718
错误修复和其他更改	718
2020 年 7 月 10 日	719
新功能	719
错误修复和其他更改	719
2020 年 6 月 30 日	719
新功能	719
错误修复和其他更改	719
文档历史记录	720
.....	dccxxv

什么是 Amazon DocumentDB (与 MongoDB 兼容)

Amazon DocumentDB (与 MongoDB 兼容) 是一项快速、可靠且完全托管的数据库服务。Amazon DocumentDB 允许您在云中轻松设置、操作和扩展与 MongoDB 兼容的数据库。借助 Amazon DocumentDB，您可以运行与 MongoDB 相同的应用程序代码并使用与 MongoDB 相同的驱动程序和工具。

在使用 Amazon DocumentDB 之前，您应查看中所述的概念和功能[工作方式 \(p. 6\)](#)。之后，完成 [入门指南 \(p. 28\)](#) 中的步骤。

主题

- [Amazon DocumentDB 概述 \(p. 1\)](#)
- [集群 \(p. 2\)](#)
- [实例 \(p. 2\)](#)
- [区域和可用区 \(p. 3\)](#)
- [Amazon DocumentDB 定价 \(p. 5\)](#)
- [监控 \(p. 5\)](#)
- [接口 \(p. 5\)](#)
- [接下来做什么？ \(p. 6\)](#)
- [Amazon DocumentDB 工作方式 \(p. 6\)](#)
- [什么是文档数据库？ \(p. 15\)](#)

Amazon DocumentDB 概述

以下是 Amazon DocumentDB 的一些高级功能：

- 随着您的数据库存储需求的增长，Amazon DocumentDB 会自动增加您的存储卷大小。您的存储卷以 10GB 的增量增加，最大可增加到 64 TiB。您无需为了满足未来增长需求而为集群预置任何多余存储空间。
- 借助 Amazon DocumentDB，您可以通过创建最多 15 个副本实例来增加读取吞吐量以支持大量应用程序请求。Amazon DocumentDB 副本共享相同的底层存储，降低了成本并避免了在副本节点执行写入的需求。此功能将释放更多的处理能力来提供读取请求并降低副本延迟时间通常可降低到几毫秒。无论存储卷大小如何，都可以在几分钟内添加副本。Amazon DocumentDB 还提供一个读取器终端节点，应用程序可以直接连接，而不必跟踪副本的添加和删除。
- Amazon DocumentDB 允许您为每个实例增加或减少计算和内存资源。计算扩展操作通常可在几分钟之内完成。
- Amazon DocumentDB 在 Amazon Virtual Private Cloud (Amazon VPC) 中运行，因此您可以将数据库隔离在您自己的虚拟网络中。您还可以配置防火墙设置以控制网络访问集群。
- Amazon DocumentDB 持续监控您的集群运行状况。发生实例故障时，Amazon DocumentDB 会自动重新启动实例和关联的进程。Amazon DocumentDB 不需要对数据库重做日志进行崩溃恢复重播，这大大减少了重启时间。Amazon DocumentDB 还将数据库缓存与数据库进程隔离出来，从而使缓存能够在实例重启后继续存在。
- 当实例故障时，Amazon DocumentDB 会自动将故障转移到最多 15 个 Amazon DocumentDB 副本中的一个，而这些副本是您在其他可用区中创建的。如果未预配置副本，当发生故障时，Amazon DocumentDB 尝试为您自动创建一个新的 Amazon DocumentDB 实例。

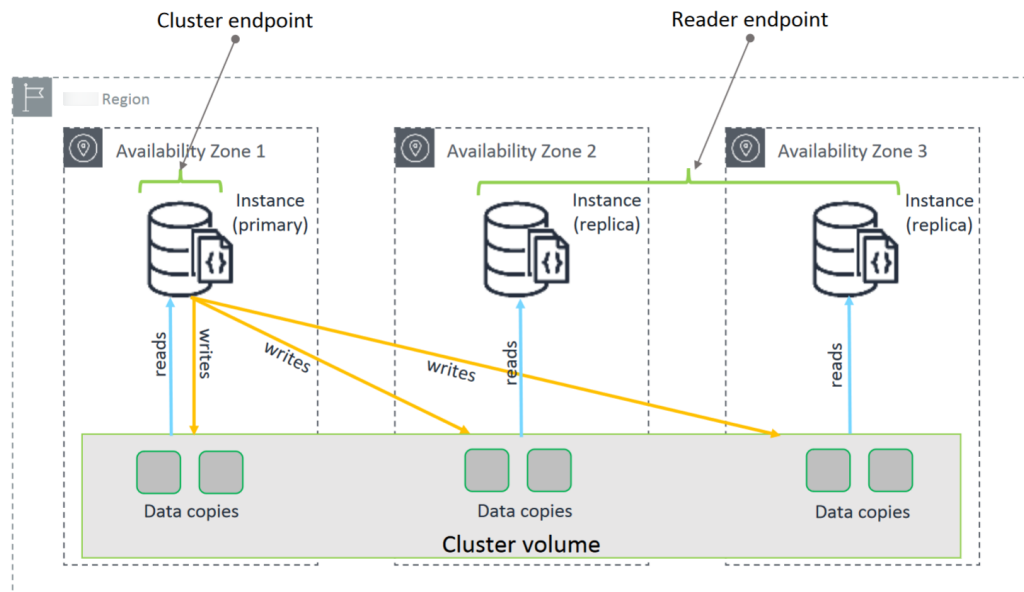
- Amazon DocumentDB 中的备份功能启用 point-in-time 为您的集群进行恢复。此功能允许您将集群恢复到保留期内（最多是近 5 分钟内）任何一秒钟的状态。可将自动备份保留期配置为最长 35 天。自动备份存储在 Amazon Simple Storage Service (Amazon S3) 中，该服务旨在实现 99.9999999999% 的耐久性。Amazon DocumentDB 备份是自动、增进式且连续的，不会对集群的性能产生任何影响。
- 借助 Amazon DocumentDB，可以使用通过创建和控制的密钥加密您的数据库 Amazon Key Management Service (Amazon KMS)。在通过 Amazon DocumentDB 加密运行的数据库集群上，静态存储于底层存储的数据都将加密。在同一个集群中的自动备份、快照和副本也会被加密。

如果你是新手 Amazon 服务，请使用以下资源了解更多信息：

- Amazon 提供针对计算、数据库、存储、分析和其他功能的服务。有关所有内容的概述 Amazon 服务，请参阅 [借助 Amazon Web Services 进行云计算](#)。
- Amazon 提供了丰富的数据库服务。有关最适合您的环境的服务指南，请参阅 [上的数据库 Amazon](#)。

集群

一个聚集包含 0 到 16 个实例和一个管理这些实例的数据的集群存储卷。所有写入操作都通过主实例完成。所有实例（主实例和副本实例）都支持读取。集群的数据存储在集群卷中，副本存储在三个不同的可用区中。



实例

Amazon DocumentDB 实例是在云中运行的独立数据库环境。一个实例可以包含多个由用户创建的数据库。您可以使用 Amazon Web Services Management Console 或 Amazon CLI 创建和修改实例。

实例的计算和内存容量由实例类。您可以选择最能满足您需求的实例。如果一段时间后您的需求出现了变化，可以选择其他实例类。有关实例类的规格，请参阅 [实例类规格 \(p. 286\)](#)。

Amazon DocumentDB 实例仅在亚马逊 VPC 环境中运行。借助 Amazon VPC，您可以控制虚拟联网环境：您可以选择自己的 IP 地址范围、创建子网以及配置路由和访问控制列表 (ACL)。

在创建 Amazon DocumentDB 实例之前，您必须创建一个包含实例的集群。

并非每个区域都支持所有实例类。下表显示了每个区域支持的实例类。

不同区域支持的实例类

区域	R6G	R5	R4	T4G	T3
美国东部（俄亥俄）	支持	支持	支持	支持	支持
美国东部（弗吉尼亚北部）	支持	支持	支持	支持	支持
美国西部（俄勒冈）	支持	支持	支持	支持	支持
南美洲（圣保罗）	支持	支持		支持	支持
亚太地区（孟买）	支持	支持		支持	支持
亚太地区（首尔）	支持	支持		支持	支持
亚太地区（悉尼）	支持	支持		支持	支持
亚太地区（新加坡）	支持	支持		支持	支持
亚太地区（东京）	支持	支持		支持	支持
加拿大（中部）	支持	支持		支持	支持
欧洲（法兰克福）	支持	支持		支持	支持
欧洲（爱尔兰）	支持	支持	支持	支持	支持
欧洲（伦敦）	支持	支持		支持	支持
欧洲（米兰）	支持	支持		支持	支持
欧洲（巴黎）	支持	支持		支持	支持
中国（北京）区域	支持	支持		支持	支持
中国（宁夏）	支持	支持		支持	支持
Amazon GovCloud (US)	支持	支持			支持

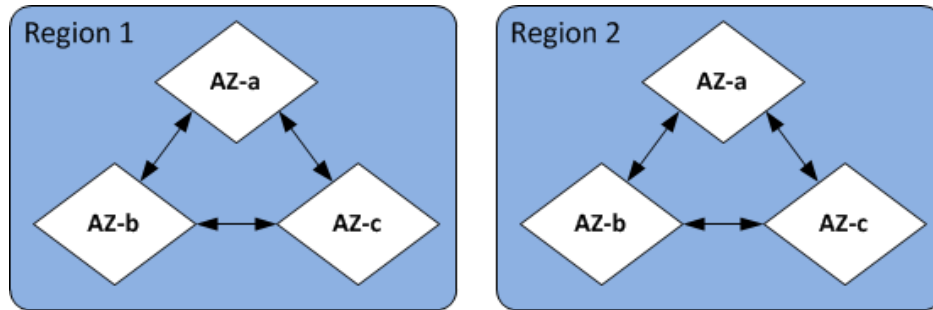
区域和可用区

区域和可用区定义集群和实例的物理位置。

区域

Amazon 云计算资源安置在世界不同地区 (例如北美、欧洲或亚洲) 高度可用的数据中心设施内。每个数据中心位置称为一个地区。

从设计而言，每个 Amazon 区域都与其他 Amazon 区域完全隔离。每个区域内有多个可用区。在不同的可用区内启动节点，可以实现可能的最大容错。下图显示了如何操作的高级视图。Amazon 区域和可用区可用。



可用区

每个 Amazon 区域包含多个称为的不同位置可用区。每个可用区都被设计成不受其他可用区故障的影响，并提供低价、低延迟的网络连接，以连接到同一地区的其他可用区。通过在多个可用区中启动给定集群的实例，您可以防止应用程序出现可用区故障，尽管这种情况很少发生。

Amazon DocumentDB 架构将存储和计算分开。对于存储层，Amazon DocumentDB 在三个之间复制 6 个数据副本 Amazon 可用区。例如，如果要在仅支持两个可用区的区域中启动 Amazon DocumentDB 集群，则系统会在三个可用区中以六种方式复制您的数据存储，但仅在两个可用区中具有计算实例。

下表列出了可在给定的中使用的可用区数量。Amazon Web Services 区域为您的集群配置计算实例。

区域名称	区域	可用区 (计算)
美国东部 (俄亥俄)	us-east-2	3
美国东部 (弗吉尼亚北部)	us-east-1	6
美国西部 (俄勒冈)	us-west-2	4
南美洲 (圣保罗)	sa-east-1	3
亚太地区 (孟买)	ap-south-1	3
亚太地区 (首尔)	ap-northeast-2	4
亚太地区 (新加坡)	ap-southeast-1	3
亚太地区 (悉尼)	ap-southeast-2	3
亚太地区 (东京)	ap-northeast-1	3
加拿大 (中部)	ca-central-1	3
中国 (北京) 区域	cn-north-1	3
中国 (宁夏)	cn-northwest-1	3
欧洲 (法兰克福)	eu-central-1	3

区域名称	区域	可用区 (计算)
欧洲 (爱尔兰)	eu-west-1	3
欧洲 (伦敦)	eu-west-2	3
欧洲 (米兰)	eu-south-1	3
欧洲 (巴黎)	eu-west-3	3
Amazon GovCloud (US)	us-gov-west-1	3

Amazon DocumentDB 定价

Amazon DocumentDB 集群根据以下组件进行计费。Amazon DocumentDB 目前没有免费套餐，因此创建集群将产生费用。

- 实例小时数 (每小时) — 根据实例的实例类 (例如, db.r5.xlarge)。定价以每小时为单位列出，但账单向下计算至秒，并以十进制形式显示时间。Amazon DocumentDB 使用量以一秒的增量进行计费，时长最少 10 分钟。有关更多信息，请参阅 [管理实例类 \(p. 283\)](#)。
- I/O 请求 (每月每 100 万个请求) — 您在计费周期中发出的存储 I/O 请求总数。
- Backup 存储 (每月每 GiB) — Backup 存储是指与自动数据库备份和拍摄的有效数据库快照相关联的存储。延长备份保留期或增加快照创建数量，将增加数据库所消耗的备份存储。备份存储以 GB 月使用量为单位计量，每秒计量方式不适用。有关更多信息，请参阅 [在 Amazon DocumentDB 中备份和还原 \(p. 194\)](#)。
- 数据传输 (每 GB) — 从 Internet 或其他方式传入和传出您的实例的数据Amazon地区。

有关详细信息，请参阅[Amazon DocumentDB \(MongoDB 兼容\) 定价](#)。

监控

您可以使用多种方法对实例性能和运行状况进行跟踪。您可以使用免费的亚马逊 CloudWatch 服务监控实例的性能和运行状况。您可以查找 Amazon DocumentDB 控制台上的性能图表。您可以订阅 Amazon DocumentDB 事件，以便在实例、快照、参数组或者安全组发生更改时收取通知。

有关更多信息，请参阅下列内容：

- 使用 [CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)
- 使用记录 [Amazon DocumentDB API 调用Amazon CloudTrail \(p. 387\)](#)

接口

有多种方式可供您与 Amazon DocumentDB 进行交互，包括Amazon Web Services Management Console和 Amazon CLI。

Amazon Web Services Management Console

Amazon Web Services Management Console是一个基于 Web 的简单用户界面。您可以从控制台中管理集群和实例，而无需进行任何编程。要访问 Amazon DocumentDB 控制台，请登录到Amazon Web Services

Management Console然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>.

Amazon CLI

您可以使用Amazon Command Line Interface(Amazon CLI)管理 Amazon DocumentDB 集群和实例。只需极少配置即可从您喜爱的终端程序开始使用 Amazon DocumentDB 控制台提供的所有功能。

- 要安装 Amazon CLI，请参阅[安装 Amazon 命令行界面](#)。
- 要开始使用Amazon CLI有关 Amazon DocumentDB，请参阅[AmazonAmazon DocumentDB 的命令行界面参考](#)。

mongo Shell

要连接到您的集群以创建、读取、更新和删除您的数据库中的文档，可以使用mongo使用 Amazon DocumentDB 的外壳。要下载并安装mongo实例 4.0 壳，请参阅[第 4 步：安装 mongo shell \(p. 36\)](#)。

MongoDB 驱动程序

要针对 Amazon DocumentDB 集群开发和编写应用程序，您还可以将 MongoDB 驱动程序与 Amazon DocumentDB 结合使用。

接下来做什么？

上一部分为您介绍 Amazon DocumentDB 提供的基本基础设施组件。您下一步该做什么？根据您的情况，请参阅以下主题之一了解其用法。

- 通过使用创建集群和实例开始使用 Amazon DocumentDBAmazon CloudFormation [Amazon DocumentDB 快速入门使用Amazon CloudFormation \(p. 40\)](#)。
- 按照我们的中的说明，通过创建集群和实例开始使用 Amazon DocumentDB[入门指南 \(p. 28\)](#)。
- 使用中的指南将您的 MongoDB 实现迁移到 Amazon DocumentDB 实施[迁Amazon DocumentDB \(p. 112\)](#)

Amazon DocumentDB 工作方式

Amazon DocumentDB (兼容 MongoDB) 是一项与 MongoDB 兼容的完全托管数据库服务。借助 Amazon DocumentDB，您可以运行与 MongoDB 相同的应用程序代码，并使用与 MongoDB 相同的驱动程序和工具。Amazon DocumentDB 与 MongoDB 3.6 和 4.0 兼容。

主题

- [Amazon DocumentDB D \(p. 7\)](#)
- [TLS Support \(p. 9\)](#)
- [Amazon DocumentDB D \(p. 9\)](#)
- [Amazon DocumentDB D \(p. 10\)](#)
- [Amazon DocumentDB D \(p. 10\)](#)
- [读取首选项选项 \(p. 11\)](#)
- [TTL 删除 \(p. 13\)](#)
- [应计费的资源 \(p. 13\)](#)

使用 Amazon DocumentDB 时，请首先创建一个聚集。集群由零个或多个数据库实例以及管理这些实例的数据的集群卷组成。Amazon DocumentDB 集群卷是一个跨多个可用区的虚拟数据库存储卷。每个可用区都有一个集群数据副本。

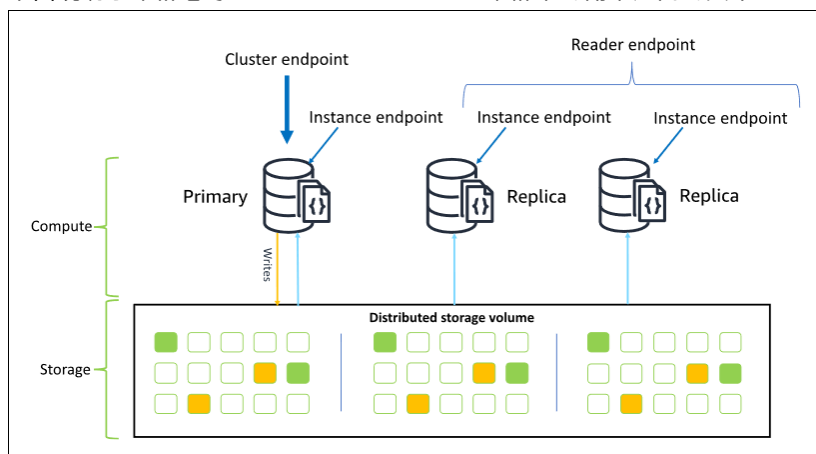
Amazon DocumentDB 集群由两部分组成：

- 集群卷— 使用云原生存储服务在三个可用区中复制数据六次，从而提供高度持久且可用的存储。Amazon DocumentDB 集群只有一个集群卷，最多可存储 64 TiB 的数据。
- 实例— 提供数据库处理能力，以及向集群存储卷写入数据和从中读取数据的能力。一个 Amazon DocumentDB 集群可以有 0-16 个实例。

实例具有以下两种角色之一：

- 主实例— 支持读取和写入操作，并执行针对集群卷的所有数据修改。Amazon DocumentDB 集群均有一个主实例。
- 副本实例— 仅支持读取操作。除主实例之外，一个 Amazon DocumentDB 集群最多可拥有 15 个副本。拥有多个副本使您可以分配读取工作负载。此外，通过将副本置于单独的可用区中，您还可以提高集群可用性。

下图说明了集群卷与 Amazon DocumentDB 集群中的副本之间的关系：



集群实例无需为相同的实例类，可以根据需要预置和终止它们。此体系结构允许您独立于集群的存储扩展集群的计算容量。

当您的应用程序将数据写入主实例时，主实例会执行对集群卷的持久写入。然后，它将该写入的状态（而不是数据）复制到每个活动副本。Amazon DocumentDB 副本不参与处理写入，因此 Amazon DocumentDB 副本对于读取扩展是有利的。从 Amazon DocumentDB 副本中读取最终是一致的，且副本滞后时间最短——通常少于主实例写入数据后的 100 毫秒。保证按照将数据写入主实例的顺序读取副本中的数据。副本滞后取决于数据更改的速率，高写入活动的时间段可能会增加副本滞后。有关更多信息，请参阅[Amazon DocumentDB 指标 \(p. 376\)](#)中的 ReplicationLag 指标。

Amazon DocumentDB D

Amazon DocumentDB 提供多种连接选项，以便为各种使用案例提供服务。要连接到 Amazon DocumentDB 集群中的实例，您需要指定实例的终端节点。终端节点是主机地址和端口号，用冒号分隔。

我们建议您使用集群终端节点以副本集模式（请参阅[作为副本集连接到 Amazon DocumentDB \(p. 454\)](#)）连接到集群，除非您有用于连接读取器终端节点或实例终端节点的特定使用案例。要将请求路由到您的副本，请选择一项驱动程序读取首选项设置，以便在满足应用程序的读取一致性要求的同时，最大程度地提高读取扩展能力。secondaryPreferred 读取首选项将启用副本读取，并释放主实例以执行更多工作。

Amazon DocumentDB 集群可以使用以下终端节点。

集群端点

集群终端节点 连接到集群的当前主实例。集群终端节点可用于读写操作。Amazon DocumentDB 集群只有一个集群终端节点。

集群终端节点为集群的读取/写入连接提供故障转移支持。如果集群的当前主实例失败并且您的集群至少有一个活动的只读副本，则集群终端节点会自动将连接请求重定向到新的主实例。在连接到 Amazon DocumentDB 集群时，我们建议您使用集群终端节点以副本集模式（请参阅[作为副本集连接到 Amazon DocumentDB \(p. 454\)](#)）。

以下是 Amazon DocumentDB 集群终端节点的示例：

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

以下是使用此集群终端节点的示例连接字符串：

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

有关查找集群终端节点的信息，请参阅[查找集群的终端节点 \(p. 344\)](#)。

读取器终端节点

读取器终端节点 跨集群中的所有可用副本对只读连接进行负载平衡。尝试通过与读取器终端节点的连接执行写入操作会导致错误。Amazon DocumentDB 集群只有一个读取器终端节点。

如果集群只包含一个（主）实例，则读取器终端节点将连接到该主实例。将副本实例添加到 Amazon DocumentDB 集群时，读取器终端节点会在新副本处于活动状态后打开与新副本的只读连接。

以下是 Amazon DocumentDB 集群的示例读取器终端节点：

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

以下是使用读取器终端节点的示例连接字符串：

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

读取器终端节点对只读连接而不是读取请求进行负载平衡。如果某些读取器终端节点连接的使用频率高于其他连接，则读取请求可能无法在集群实例之间以相同方式取得平衡。建议以副本集形式连接到集群终端节点并利用 secondaryPreferred 读取选项来分发请求。

有关查找集群终端节点的信息，请参阅[查找集群的终端节点 \(p. 344\)](#)。

实例终端节点

实例终端节点 连接到集群中的特定实例。当前主实例的实例终端节点可用于读取和写入操作。但是，尝试对只读副本的实例终端节点执行写入操作会导致错误。Amazon DocumentDB 集群的每个活动实例有一个实例终端节点。

对于可能不适合使用集群终端节点或读取器终端节点的场景，实例终端节点提供对特定实例的连接的直接控制。示例使用案例是配置周期性只读分析工作负载。您可以预置一个 larger-than-normal 副本实例使用其实

例终端节点直接连接到新的较大实例，运行分析查询，然后终止实例。使用实例终端节点可防止分析流量影响其他集群实例。

下面是 Amazon DocumentDB 集群中单个实例的示例实例终端节点：

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

下面是使用此实例终端节点的示例连接字符串：

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

Note

由于故障转移事件，实例的主要角色或副本角色可能会发生变化。您的应用程序永远不应该假设特定的实例终端节点是主实例。我们不建议连接到生产应用程序的实例终端节点。相反，我们建议您使用集群终端节点以副本集模式（请参阅 [作为副本集连接到 Amazon DocumentDB \(p. 454\)](#)）连接到您的集群。要想对实例故障转移优先级进行更高级的控制，请参阅 [了解 Amazon DocumentDB B 集群容错能力 \(p. 282\)](#)。

有关查找集群终端节点的信息，请参阅 [查找实例的终端节点 \(p. 345\)](#)。

副本集模式

您可以通过指定副本集名称以副本集模式连接到您的 Amazon DocumentDB 集群终端节点。rs0. 在副本集模式下连接可以指定“Read Concern (读取问题)”、“Write Concern (写入问题)”和“Read Preference (读取首选)”选项。有关更多信息，请参阅 [读取一致性 \(p. 11\)](#)。

下面是以副本集模式连接的示例连接字符串：

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

在副本集模式下连接时，您的 Amazon DocumentDB 集群将作为副本集显示给您的驱动程序和客户端。在 Amazon DocumentDB 集群中添加和删除的实例会自动反映在副本集配置中。

Amazon DocumentDB 集群均包含一个默认名称为的副本集rs0. 该副本集名称无法修改。

在副本集模式下连接到集群终端节点是常规用途的推荐方法。

Note

Amazon DocumentDB 集群中的所有实例都在同一 TCP 端口上侦听连接。

TLS Support

有关使用传输层安全性 (TLS) 连接到 Amazon DocumentDB 的更多详细信息，请参阅 [加密传输中的数据 \(p. 136\)](#)。

Amazon DocumentDB D

Amazon DocumentDB 数据存储在集群卷，该卷是使用固态硬盘 (SSD) 的单个虚拟卷。集群卷由六个数据副本组成，可在单个可用区之间自动复制Amazon Web Services 区域. 此复制有助于确保您的数据具有高持久性，减少数据丢失的可能性。它还有助于确保在故障转移期间您的集群具有更高可用性，因为您的数据副本已存在于其他可用区中。这些副本可以继续向您的 Amazon DocumentDB 集群中的实例提供数据请求服务。

数据存储的计费方式

随着数据量的增加，Amazon DocumentDB 会自动增加集群卷的大小。Amazon DocumentDB 集群卷的最大大小可增大到 64 TiB；但是，您只需为在 Amazon DocumentDB 集群卷中使用的空间付费。从 Amazon DocumentDB 4.0 开始，当例如，例如，通过删除集合或索引时，整个分配的空间将减少相当数量。因此，您可以通过删除不再需要的集合、索引和数据库来减少存储费用。使用 Amazon DocumentDB 3.6，在删除数据时（例如，删除集合或索引），整个分配的空间保持不变。在将来数据量增加时，可以自动重用可用空间。

Note

在 Amazon DocumentDB 3.6 中，存储成本基于存储“高水位”（在任何时间点为 Amazon DocumentDB 集群分配的最大容量），即在任何时间点为 Amazon DocumentDB 集群分配的最大容量。您可以避免采用创建大量临时信息的 ETL 做法以控制成本，或者避免在删除不需要的旧数据之前加载大量新数据的 ETL 做法。如果从 Amazon DocumentDB 集群中删除数据导致分配大量未使用的空间，则重置高水位需要使用之类的工具执行逻辑数据转储和还原以转储和还原到新集群 `mongodump` 要么 `mongorestore`。创建和还原快照不会减少分配的存储，因为基础存储的物理布局在还原的快照中保持不变。

Note

使用 `mongodump` 和 `mongorestore` 等实用程序会导致发生 I/O 费用，具体取决于所读取和写入存储卷的数据量。

有关 Amazon DocumentDB 数据存储和 I/O 定价的信息，请参阅 [Amazon DocumentDB \(与 MongoDB 兼容\) 定价和定价常见问题](#)。

Amazon DocumentDB D

在 Amazon DocumentDB 集群中，每个副本实例都公开一个独立的终端节点。这些副本终端节点提供对集群卷中数据的只读访问。它们使您能够在多个复制实例上扩展数据的读取工作负载。它们还有助于提高数据读取的性能，并提高 Amazon DocumentDB 集群中数据的可用性。Amazon DocumentDB 副本也是故障转移目标，如果您的 Amazon DocumentDB 集群的主实例失败，这些副本会快速得到提升。

Amazon DocumentDB D

Amazon DocumentDB 的设计具有可靠、持久和容错的特点。（为了提高可用性，您应该配置您的 Amazon DocumentDB 集群，使其在不同的可用区中具有多个副本实例。）Amazon DocumentDB 包括多种自动化功能，使其成为了可靠的数据库解决方案。

存储自动修复

Amazon DocumentDB 将多个数据副本保留在三个可用区中，从而大大降低了因存储故障而丢失数据的可能性。Amazon DocumentDB 会自动检测集群卷中的故障。如果集群卷的某个区段发生故障，Amazon DocumentDB 会立即修复该区段。它使用集群卷包含的其他卷中的数据以帮助确保已修复区段中的数据是最新的。因此，Amazon DocumentDB 将避免数据丢失，并减少了执行 Amazon DocumentDB 的需求 point-in-time 恢复以从实例故障中恢复。

自动恢复缓存预热

Amazon DocumentDB 在与数据库不同的进程中管理其页面缓存，以便页面缓存可以独立于数据库而存在。万一发生数据库故障，页面缓存仍保留在内存中。这可确保在数据库重新启动时使用最新状态对缓冲池进行预热。

崩溃恢复

Amazon DocumentDB 设计为几乎可以立即从崩溃中恢复，并继续为您的应用程序数据提供服务。Amazon DocumentDB 在 parallel 线程上异步执行崩溃恢复，以便在发生崩溃后数据库几乎可立即打开并可供使用。

资源监管

Amazon DocumentDB 保护在服务中运行关键流程（例如运行状况检查）所需的资源。为此，当实例遇到高内存压力时，Amazon DocumentDB 将限制请求。因此，某些操作可能会排队等待内存压力消退。如果内存压力持续存在，排队的操作可能会超时。您可以使用以下命令监视服务是否因内存不足而限制操作 CloudWatch 指标：LowMemThrottleQueueDepth、LowMemThrottleMaxQueueDepth、LowMemNumOperationsThrottled、LowMemThrottleQueueDepth。

有关更多信息，请参阅使用监控 Amazon DocumentDB 数据库 CloudWatch。如果您发现由于以下原因导致实例持续承受内存压力 LowMem CloudWatch 指标，我们建议您纵向扩展实例，以便为您的工作负载提供更多内存。

读取首选选项

Amazon DocumentDB 使用云原生共享存储服务，该服务在三个可用区中复制数据六次，以提供高级别的持久性。Amazon DocumentDB 不依赖将数据复制到多个实例来实现持久性。无论集群是包含单个实例还是 15 个实例，集群的数据都具有持久性。

写入持久性

Amazon DocumentDB 使用一种独特的、分布式、容错、自我修复的存储系统。此系统跨三个 (V=6) 数据副本 (V=6) 数据 Amazon 可用区，以提供高可用性和持久性。在写入数据时，Amazon DocumentDB 确保在确认写入客户端之前，所有写入都已在大多数节点上持久记录。如果您运行的是三节点 MongoDB 副本集，则使用写关注点 {w:3, j:true} 与 Amazon DocumentDB 进行比较时，将产生尽可能好的配置。

对 Amazon DocumentDB 集群的写入必须由集群的写入实例处理。尝试写入读取器会导致错误。来自 Amazon DocumentDB 主实例的已确认写入具有持久性，无法回滚。Amazon DocumentDB 默认具有高持久性，不支持非持久性写入选项。您无法修改持久性级别（即写关注）。Amazon DocumentDB 忽略了 w=anything，实际上是 w:3 和 j:true。您无法减少该配置。

在 Amazon DocumentDB 架构中，存储与计算是分离的，因此具有单个实例的集群拥有很高的持久性。持久性在存储层处理。因此，具有单个实例和具有三个实例的 Amazon DocumentDB 集群实现了相同级别的持久性。您可以根据特定用例配置集群，同时仍为数据提供高持久性。

Amazon DocumentDB 集群的写入操作在单个文档中是原子操作。

Amazon DocumentDB 不支持 wtimeout 选项，并且如果指定了值，将不会返回错误。对主 Amazon DocumentDB 实例的写入操作保证不会无限期阻塞。

读取隔离

从 Amazon DocumentDB 实例读取只返回查询开始之前已持久存在的数据。读取从不返回在查询开始执行后修改的数据，即，任何情况下都不会进行脏读取。

读取一致性

从 Amazon DocumentDB 集群读取的数据具有持久性，将不会回滚。您可以通过指定请求或连接的读取首选选项来修改 Amazon DocumentDB 读取的读取一致性。Amazon DocumentDB 不支持非持久性读取选项。

从 Amazon DocumentDB 集群的主实例读取在正常操作条件下具有强一致性，并且具有 read-after-write 一致性。如果在写入与后续读取之间发生故障转移事件，系统可能短暂返回不具有强一致性的读取。从只读副本的所有读取最终是一致的，以相同的顺序返回数据，并且通常具有小于 100 毫秒的副本滞后。

Amazon DocumentDB Re

仅当以副本集模式从集模式下的集群终端节点读取数据时，Amazon DocumentDB 才支持设置读取首选选项。设置读取首选选项会影响 MongoDB 客户端或驱动程序将读取请求路由到 Amazon DocumentDB 集群中的实例。

的方式。您可以为特定查询设置读取首选选项，或者作为 MongoDB 驱动程序中的常规选项。（有关如何设置读取首选选项的说明，请查阅客户端或驱动程序文档。）

如果您的客户端或驱动程序未在副本集模式下连接到 Amazon DocumentDB 集群终端节点，则指定读取首选选项的结果未定义。

Amazon DocumentDB 不支持设置标签集作为读取首选选项。

支持的读取首选选项

- **primary**—指定primary读取首选选项有助于确保将所有读取请求路由到集群的主实例。如果主实例不可用，则读取操作将失败。一个primary读取偏好产量 read-after-write 一致性，适用于优先级较高的用例 read-after-write 一致性超过高可用性和读取扩展。

以下示例指定 primary 读取首选选项：

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

- **primaryPreferred**—指定primaryPreferred读取首选选项会在正常操作条件下，将读取请求路由到主实例。如果发生主实例故障转移，则客户端会将请求路由到副本。一个primaryPreferred读取偏好产量 read-after-write 在正常操作期间保持一致性，并在故障转移事件期间产生最终一致性读取。一个primaryPreferred读取首选选项适用于优先级较高的用例 read-after-write 一致性超过读取扩展，但仍需要高可用性。

以下示例指定 primaryPreferred 读取首选选项：

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

- **secondary**—指定secondary读取首选选项可确保只将读取请求路由到副本，从不路由到主实例。如果集群中没有副本实例，则读取请求将失败。一个secondary读取首选选项会产生最终一致性读取，适用于主实例写入吞吐量优先于高可用性和 read-after-write 一致性。

以下示例指定 secondary 读取首选选项：

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

- **secondaryPreferred**—指定secondaryPreferred读取首选选项可确保只将读取路由到只读副本，不会路由到主实例。如果集群中没有活动的副本实例，则读取请求将路由到主实例。当读取由只读副本提供服务时，secondaryPreferred 读取首选选项会产生最终一致性读取。它产生了 read-after-write 当读取由主实例提供服务时保持一致（禁止故障转移事件）。一个secondaryPreferred读取首选选项适用于读取扩展和高可用性优先于读取扩展和高可用性的使用案例 read-after-write 一致性。

以下示例指定 secondaryPreferred 读取首选选项：

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

- **nearest**—指定nearest读取首选选项将仅基于客户端与 Amazon DocumentDB 集群中所有实例之间测量的延迟来路由读取请求。当读取由只读副本提供服务时，nearest 读取首选选项会产生最终一致性读取。它产生了 read-after-write 当读取由主实例提供服务时保持一致（禁止故障转移事件）。一个nearest读取首选选项适用于实现尽可能低的读取延迟和高可用性优先于实现尽可能低的读取延迟和高可用性的使用案例 read-after-write 一致性和读取扩展。

以下示例指定 `nearest` 读取首选项：

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

高可用性

Amazon DocumentDB 通过将副本用作主实例的故障转移目标来支持高度可用的集群配置。如果主实例失败，则将一个 Amazon DocumentDB 副本提升为新的主实例，并会出现短暂中断，在此期间，对主实例发出的读写请求将失败，并会出现异常。

如果您的 Amazon DocumentDB 集群不包含任何副本，则会在失败期间重新创建主实例。但是，提升 Amazon DocumentDB 副本要比重新创建主实例快得多。因此，我们建议您创建一个或多个 Amazon DocumentDB 副本作为故障转移目标。

旨在用作故障转移目标的副本应与主实例具有相同的实例类。它们应在不同于主实例的可用区中进行配置。您可以控制哪些副本作为首选故障转移目标。有关配置 Amazon DocumentDB 以实现高可用性的最佳实践，请参阅 [了解 Amazon DocumentDB B 集群容错能力 \(p. 282\)](#)。

扩展读取

Amazon DocumentDB 副本是读取扩展的理想选择。它们完全专用于集群卷上的读取操作，即副本不处理写入。数据复制发生在集群卷中，而不是在实例之间。因此，每个副本的资源都专用于处理查询，而不是复制和写入数据。

如果您的应用程序需要更多读取容量，则可以快速向集群添加副本（通常在不到十分钟的时间内完成）。如果您的读取容量要求减少，则可以删除不需要的副本。使用 Amazon DocumentDB 副本，您只需为所需的读取容量付费。

Amazon DocumentDB 通过使用“读取首选项”选项支持客户端读取扩展。有关更多信息，请参阅 [Amazon DocumentDB Re \(p. 11\)](#)。

TTL 删除

在特定时间范围内无法保证从通过后台进程实现的 TTL 索引区域中进行删除，只能尽力而为。实例大小、实例资源利用率、文档大小和总体吞吐量等因素会影响 TTL 删除的时间。

TTL 监视器每次删除您的文档都会产生 IO 成本，这将增加您的账单费用。如果吞吐量和 TTL 删除率提高，您应预计到账单费用会因 IO 使用量的增加而增多。

在现有集合上创建 TTL 索引时，必须先删除所有过期的文档，然后才能创建索引。当前的 TTL 实现经过优化，可以删除集合中的一小部分文档，如果从一开始就在集合上启用 TTL，则这种情况很常见；如果需要一次性删除大量文档，则可能会导致比必要的 IOPS 更高。

如果您不想创建 TTL 索引来删除文档，则可以根据时间将文档归入各个集合中，并在不再需要文档时将其删除。例如：您可以每周创建一个集合，然后将其丢弃，而不会产生 IO 成本。这样做比使用 TL 索引更具成本效益。

应计费的资源

识别应计费的 Amazon DocumentDB 资源

作为一项完全托管的数据库服务，Amazon DocumentDB 会对实例、存储、I/O、备份和数据传输收费。有关更多信息，请参阅 [Amazon DocumentDB \(与 MongoDB 兼容\) 定价](#)。

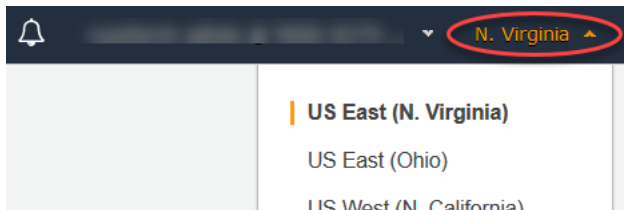
要查找您账户中应计费的资源并删除这些资源，您可以使用 Amazon Web Services Management Console 或 Amazon CLI。

使用 Amazon Web Services Management Console

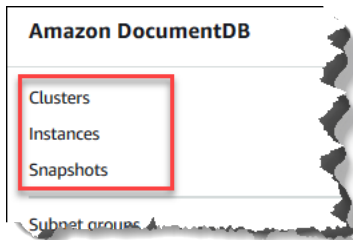
使用Amazon Web Services Management Console，您可以查找为给定预置的 Amazon DocumentDB 集群、实例和快照Amazon Web Services 区域。

查找集群、实例和快照

1. 登录到Amazon Web Services Management Console，然后打开亚马逊 DocumentDB 控制台<https://console.aws.amazon.com/docdb>.
2. 要在非默认区域中查找应计费资源，请在屏幕右上角处，选择Amazon Web Services 区域要搜索的资源。



3. 在导航窗格中，选择您感兴趣的应计费资源类型：集群、实例，或者快照。



4. 右侧窗格中列出了该区域的所有已预置的集群、实例或快照。您将需要为集群、实例和快照付费。

使用 Amazon CLI

使用Amazon CLI，您可以查找为给定预置的 Amazon DocumentDB 集群、实例和快照Amazon Web Services 区域。

查找集群和实例

以下代码将列出指定区域的所有集群和实例。如果要在默认区域中搜索集群和实例，可以省略 `--region` 参数。

Example

对于 Linux、macOS 或 Unix：

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

对于 Windows：

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

此操作的输出将类似于下文。

```
"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",
```

查找快照

以下代码将列出指定区域的所有快照。如果要在默认区域中搜索快照，可以省略 `--region` 参数。

对于 Linux、macOS 或 Unix：

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

对于 Windows：

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

此操作的输出将类似于下文。

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

您只需要删除 `manual` 快照。`Automated` 快照会在您删除集群时删除。

删除不需要的应计费资源

要删除集群，必须先删除集群中的所有实例。

- 要删除实例，请参阅[删除 Amazon DocumentDB 实例 \(p. 300\)](#)。

Important

即使您删除集群中的实例，仍需要为与该集群关联的存储和备份用量付费。要停止所有计费，您还必须删除集群和手动快照。

- 要删除集群，请参阅[删除 Amazon DocumentDB 集群 \(p. 267\)](#)。
- 要删除手动快照，请参阅[删除集群快照 \(p. 221\)](#)。

什么是文档数据库？

一些开发人员不根据规范化行和列来考虑其数据模型。通常，在应用程序层中，数据表示为 JSON 文档，因为开发人员将其数据模型视为文档更为直观。

文档数据库的受欢迎程度不断提高，因为开发人员可以使用他们在其应用程序代码中使用的相同文档模型格式来保存数据库中的数据。文档数据库为灵活和敏捷的开发提供了强大直观的 API。

主题

- [文档数据库使用案例 \(p. 16\)](#)
- [了解文档 \(p. 16\)](#)
- [使用文档 \(p. 20\)](#)

文档数据库使用案例

无论您是否需要文档数据库或其他类型的数据库来管理数据，您的使用案例都将启动。对于需要一个灵活架构以实现快速迭代开发的工作负载来说，文档数据库很有用。以下是一些使用案例的示例，文档数据库可以为这些案例提供显著的优势：

主题

- [用户资料 \(p. 16\)](#)
- [实时大数据 \(p. 16\)](#)
- [内容管理 \(p. 16\)](#)

用户资料

由于文档数据库具有灵活架构，他们可以存储具有不同属性和数据值的文档。文档数据库是在线资料的实际解决方案，其中不同的用户提供不同类型的信息。使用文档数据库，通过仅存储特定于每个用户的属性，您可以高效地存储每个用户的资料。

假设用户选择添加或删除其资料中的信息。在这种情况下，其文档可轻易被一个更新的版本所取代，更新的版本包含任何最近添加的属性和数据，或者省略任何新省略的属性和数据。文档数据库轻松管理此级别的个人性和流动性。

实时大数据

历史上，从业务数据中提取信息的能力受到影响，因为业务数据库和分析数据库是在不同的环境中维护的，分别是运营和业务/报告。在竞争激烈的商业环境中，能够实时提取运营信息至关重要。通过使用文档数据库，企业可以存储和管理任何来源的运营数据，并将数据并发到选择的 BI 引擎进行分析。无需具备两种环境。

内容管理

要有效地管理内容，您必须能够从各种来源收集和汇总内容，然后将其交付给客户。由于其灵活的架构，文档数据库非常适合用于收集和存储任何类型的数据。您可以使用它们来创建和引入新类型的内容，包括用户生成的内容，如图像、评论和视频。

了解文档

文档数据库用于将半结构化数据存储为文档，而不是像关系数据库那样在多个表之间对数据进行规范化，每个表都有唯一的固定结构。存储在文档数据库中的文档使用嵌套键值对来提供文档的结构或架构。不过，不同类型的文档可以存储在同一文档数据库中，从而满足了处理不同格式的类似数据的要求。例如，由于每个文档都是自描述的，主题[文档数据库中的示例文档 \(p. 18\)](#)中所述的在线存储的 JSON 编码文档可以存储在同一个文档数据库中。

主题

- [SQL 与非关系术语 \(p. 17\)](#)

- [简单文档 \(p. 17\)](#)
- [嵌入文档 \(p. 17\)](#)
- [文档数据库中的示例文档 \(p. 18\)](#)
- [了解文档数据库中的规范化 \(p. 19\)](#)

SQL 与非关系术语

下表对文档数据库 (MongoDB) 使用的术语与 SQL 数据库使用的术语进行了比较。

SQL	MongoDB
表	集合
行	文档
列	字段
主键	ObjectId
索引	索引
查看	查看
嵌套表或对象	嵌入文档
数组	数组

简单文档

文档数据库中的所有文档都是自描述的。虽然此文档使用类似 JSON 格式的文档，但您可以使用其他编码方式。

简单文档具有一个或多个字段，这些字段在文档中都处于同一级别。在以下示例中，字段 SSN、LName、FName、DOB、Street、City、State-Province、PostalCode 和 Country 在文档中属于同级。

```
{
  "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"
}
```

在简单文档中组织信息时，每个字段将被单独管理。要检索个人的地址，必须作为单个数据项检索 Street、City、State-Province、PostalCode 和 Country。

嵌入文档

复杂文档通过在文档中创建嵌入文档来组织数据。嵌入文档帮助管理分组中作为单个数据项的数据，这些数据项在特定情况下效率更高。使用上述示例，您可以在主文档中嵌入 Address 文档。这样做会生成以下文档结构：

```
{
  "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"
  }
}
```

您现在可以作为单个字段访问文档中的数据 ("SSN":), 作为嵌入式文档 ("Address":), 或者作为嵌入式文档的成员 ("Address":{"Street":})。

文档数据库中的示例文档

如上所述, 因为数据库中的每个文档是自描述的, 文档数据库中的文档结构可能彼此不同。以下两个文档, 一个是图书文档, 另一个是为期刊文档, 在结构上有所不同。不过, 它们可以位于同一个文档数据库中。

下面是一个示例图书文档:

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

以下是带有两篇文章的示例期刊文档:

```
{
  "_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"
  }
}
```

比较这两个文档的结构。借助关系数据库，您需要单独的“periodical”和“books”表或未使用字段的单个表作为 null 值，例如“发布”、“问题”、“文章”和“MI”。由于文档数据库是半结构化的，每个文档都定义了自己的结构，因此这两个文档可以在同一个文档数据库中共存，而没有 null 字段。文档数据库善于处理稀疏数据。

针对文档数据库进行开发，可以实现快速、迭代的开发。这是因为您可以动态地更改文档的数据结构，而不必更改整个集合的架构。文档数据库非常适合敏捷开发和动态变化的环境。

了解文档数据库中的规范化

文档数据库未规范化；在一个文档中发现的数据可以在另一个文档中重复。此外，文档之间可能存在一些数据差异。例如，假设您在网上商店购物，所有购物的详细信息都存储在一个文档中。文档应类似于以下 JSON 文档：

```
{
  "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"
}
```

所有这些信息都作为文档存储在交易集合中。后来，您意识到忘记购买一件物品。因此，您再次登录同一家商店，进行另一次购买，这也是作为另一个文档存储在交易集合中。

```
{
  "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"
}
```

请注意这两个文档之间的冗余您的姓名和购物者 ID（此外，如果您使用相同的信用卡，您的信用卡信息）。但这没关系，因为存储成本低廉，每个文档都完整记录了单个交易，只需简单的键值查询就可以快速检索，不需要连接。

两个文档之间也存在明显差异，您的信用卡信息。这只是一个明显差异，因为可能您每次购买都使用不同的信用卡。每个文档对其记录的交易都是准确的。

使用文档

作为文档数据库，Amazon DocumentDB 可以轻松存储、查询和索引 JSON 数据。除了没有对所有文档强制实施单一架构外，Amazon DocumentDB 中的集合类似于关系数据库中的表。集合允许您将类似的文档分组在一起，同时将它们保存在同一数据库中，而不需要它们在结构上相同。

使用前面部分的示例文档，您可能会有针对 `reading_material` 和 `office_supplies` 的集合。您的软件有责任强制执行文档属于哪个集合。

以下示例使用 MongoDB API 来展示如何添加、查询、更新和删除文档。

主题

- [添加文档 \(p. 20\)](#)
- [查询文档 \(p. 22\)](#)
- [更新文档 \(p. 23\)](#)
- [删除文档 \(p. 27\)](#)

添加文档

在 Amazon DocumentDB 中，首次向集合添加文档时会创建数据库。在此示例中，您将在 `test` 数据库中创建名为 `example` 的集合，该集合是连接到集群时的默认数据库。由于插入第一个文档时隐式创建集合，因此集合名称没有错误检查。因此，集合名称中的拼写错误（例如 `eexample` 而不是 `example`）将创建文档并将其添加到 `eexample` 集合，而不是预期集合中。错误检查必须由您的应用程序进行处理。

以下示例使用 MongoDB API 来添加文档。

主题

- [添加单个文档 \(p. 20\)](#)
- [添加多个文档 \(p. 21\)](#)

添加单个文档

要将单个文档添加至集合，请使用带有您想添加至集合的文档的 `insertOne( {} )` 操作。

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

此操作的输出将类似于下文 (JSON 格式)。

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

添加多个文档

要将多个文档添加至集合，请使用带有您想添加至集合的文档列表的 `insertMany( [{}, ..., {}] )` 操作。虽然此特定列表中的文档具有不同的架构，但它们都可以添加至同一个集合。

```
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,  
        "OrderQty": 36  
      }  
    }  
  ]  
)
```

此操作的输出将类似于下文 (JSON 格式)。

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

```
        ObjectId("5bedb07941ca8d9198f5934e")
    ]
}
```

查询文档

有时，您可能需要查看在线商店的库存，这样客户就能看到并购买您销售的物品。查询集合相对容易，无论您想要集合中的所有文档，还是仅需要那些满足特定标准的文档。

要查询文档，请使用 `find()` 操作。`find()` 命令具有单个文档参数，该参数定义了在选择要返回的文档时要使用的标准。`find()` 的输出是一个文档，其格式为一行文本，不含换行符。要格式化输出文档，从而更加轻松地读取，请使用 `find().pretty()`。本主题中的所有示例都使用 `.pretty()` 设置输出的格式。

使用你插入的四个文档 `example` 前两个练习中的收藏 — `insertOne()` 和 `insertMany()`。

主题

- [检索集合中的所有文档 \(p. 22\)](#)
- [检索与字段值匹配的文档 \(p. 22\)](#)
- [检索与嵌入文档匹配的文档 \(p. 22\)](#)
- [检索与嵌入文档中的字段值匹配的文档 \(p. 23\)](#)
- [检索与数组匹配的文档 \(p. 23\)](#)
- [检索与数组中的值匹配的文档 \(p. 23\)](#)
- [使用运算符检索文档 \(p. 23\)](#)

检索集合中的所有文档

要检索集合中的所有文档，请将 `find()` 操作和空查询文档结合使用。

以下查询返回 `example` 集合中的所有文档。

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

检索与字段值匹配的文档

要检索与字段和值匹配的所有文档，请将 `find()` 操作和查询文档（标识要匹配的字段和值）结合使用。

通过使用前述文档，此查询将返回其中“Item”字段等于“Pen”的所有文档。

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

检索与嵌入文档匹配的文档

要查找与嵌入文档匹配的所有文档，请将 `find()` 操作和查询文档（指定嵌入文档名称和嵌入文档的所有字段和值）结合使用。

在与嵌入文档匹配时，该文档的嵌入文档的名称必须与查询中的名称相同。此外，嵌入文档中的字段和值必须与查询匹配。

以下查询仅返回“Poster Paint”文档。这是因为“Pen”具有不同的“OnHand”和“MinOnHand”值，并且“Spray Paint”比查询文档多一个字段（`OrderQty`）。

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

检索与嵌入文档中的字段值匹配的文档

要查找与嵌入文档匹配的所有文档，请将 `find()` 操作和查询文档（指定嵌入文档名称和嵌入文档的所有字段和值）结合使用。

考虑到上述文档，以下查询使用“点表示法”来指定嵌入文档和感兴趣的字段。将返回所有与这些内容匹配的文档，而不管嵌入文档中可能存在哪些其他字段。此查询将返回“Poster Paint”和“Spray Paint”，因为它们与指定的字段和值匹配。

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

检索与数组匹配的文档

要查找所有与数组匹配的文档，请将 `find()` 操作和您感兴趣的数组名称以及数组中的所有值结合使用。此查询将返回所有包含带该名称的数组（其中数组值和顺序与查询中的完全相同）的文档。

以下查询仅返回“Pen”，因为“Poster Paint”具有其他颜色 (White)，并且“Spray Paint”具有顺序不同的颜色。

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

检索与数组中的值匹配的文档

要查找所有具有特定数组值的文档，请将 `find()` 操作与您感兴趣的数组名称和值结合使用。

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

上述操作将返回所有三个文档，因为它们都有一个名为 `Colors` 的数组，并且此数组中的某个位置具有“Red”值。如果您指定值“White”，则查询将仅返回“Poster Paint”。

使用运算符检索文档

以下查询返回“Inventory.OnHand”值小于 50 的所有文档。

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

有关支持的查询运算符的列表，请参阅 [查询和投影运算符 \(p. 97\)](#)。

更新文档

通常，您的文档不是静态的，而是作为应用程序工作流的一部分进行更新。以下示例展示了更新文档的一些方法。

要更新现有文档，请使用 `update()` 操作。`update()` 操作具有两个文档参数。第一个文档标识要更新的文档。第二个文档指定要进行的更新。

在更新现有字段时，无论该字段是简单字段、数组还是嵌入文档，都请指定字段名及其值。在操作结束时，旧文档中的字段似乎已替换为新的字段和值。

主题

- [更新现有字段的值 \(p. 24\)](#)
- [添加新字段 \(p. 25\)](#)
- [替换嵌入文档 \(p. 25\)](#)
- [将新字段插入嵌入文档 \(p. 26\)](#)

- [从文档中删除字段 \(p. 27\)](#)
- [从多个文档中删除字段 \(p. 27\)](#)

更新现有字段的值

在下面的更新操作中，使用您之前添加的以下四个文档。

```
{
  "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,
    "OrderQty": 36
  }
}
```

更新简单字段

要更新简单字段，请将 `update()` 和 `$set` 结合使用以指定字段名和新值。以下示例将 `Item` 从“Pen”更改为“Gel Pen”。

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

此操作的结果将类似于下文。

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

```
}  
}
```

更新数组

以下示例将现有颜色数组替换为新数组（其中包括 Orange）并从颜色列表中删除 White。新的颜色列表的顺序是在 update() 操作中指定的。

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

此操作的结果将类似于下文。

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

添加新字段

要通过添加一个或多个新字段修改文档，请使用带有查询文档的 update() 操作，该查询文档使用 \$set 运算符标识要插入的文档及新字段和值。

以下示例将值为 3.99 的字段 UnitPrice 添加到 Spray Paints 文档。请注意，值 3.99 是数字，而不是字符串。

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

此操作的结果将类似于下文（JSON 格式）。

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

替换嵌入文档

要通过替换嵌入文档来修改文档，请将 update() 操作和文档（使用 \$set 运算符标识嵌入文档及其新字段和值）结合使用。

给定以下文档。

```
db.example.insert({
```

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

替换嵌入文档

以下示例将当前 Date 文档替换为新文档，后者仅具有 Month 和 Day 字段；并且已消除 Year。

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

此操作的结果将类似于下文。

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

将新字段插入嵌入文档

将字段添加到嵌入文档

要通过向嵌入文档添加一个或多个新字段来修改文档，请将 update() 操作与文档（标识嵌入文档）结合使用，并使用“点表示法”通过 \$set 运算符指定嵌入文档以及要插入的新字段和值。

给定下面的文档，以下代码使用“点表示法”将 Year 和 DoW 字段插入嵌入式 Date 文档中，并将 words 插入父文档中。

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

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

此操作的结果将类似于下文。

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

```
    "Year": 1987,  
    "DoW": "Saturday"  
  },  
  "Words": 2482  
}
```

从文档中删除字段

要通过删除文档中的字段来修改文档，请将 `update()` 操作与查询文档（标识要从中删除字段的文档）和 `$unset` 运算符（指定要删除的字段）结合使用。

以下示例从前述文档中删除 `Words` 字段。

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

此操作的结果将类似于下文。

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

从多个文档中删除字段

要通过从多个文档中删除字段来修改文档，请使用具有 `$unset` 运算符和 `multi` 选项的 `update()` 操作设置为 `true`。

以下示例从示例集合中的所有文档中删除 `Inventory` 字段。如果文档没有 `Inventory` 字段，则不对该文档采取任何操作。如果省略了 `multi: true`，则仅只在符合标准的第一个文档上执行此操作。

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

删除文档

要从您的数据库删除文档，请使用 `remove()` 操作，指定要删除哪个文档。以下代码将从 `example` 集合中删除“Gel Pen”。

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

要删除数据库中的所有文档，请将 `remove()` 操作和空查询结合使用，如下所示。

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

开始使用 Amazon DocumentDB

可通过多种方式连接并开始使用 Amazon DocumentDB。我们创建本指南是因为我们发现这种方法是用户开始使用我们强大的文档数据库的最快捷、最简单和最简单的方法。本指南使用 [Amazon Cloud9](#)，一个基于 Web 的终端，用于直接从 Amazon Web Services Management Console。有资格使用 Amazon 免费套餐可以使用 Amazon DocumentDB 和 Amazon Cloud9 免费。如果您的 Amazon Cloud9 Amazon DocumentDB 集群使用免费套餐之外的资源，您需要以正常的方式付费 Amazon 这些资源的费率。本指南将在不到 15 分钟的时间内让你开始使用 Amazon DocumentDB。

主题

- [先决条件](#) (p. 28)
- [第 1 步：创建 Amazon Cloud9 环境](#) (p. 29)
- [第 2 步：创建安全组](#) (p. 31)
- [第 3 步：创建 Amazon DocumentDB 集群](#) (p. 34)
- [第 4 步：安装 mongo shell](#) (p. 36)
- [第 5 步：Connect 到您的 Amazon DocumentDB 集群](#) (p. 37)
- [第 6 步：插入和查询数据](#) (p. 38)
- [步骤 7：了解](#) (p. 39)

如果您希望通过创建与 Amazon EC2 实例的 SSH 连接从本地计算机连接到 Amazon DocumentDB，请参阅 [Connect EC2 说明](#)

先决条件

在创建第一个 Amazon DocumentDB 集群之前，您必须先执行以下操作：

创建 Amazon Web Services (Amazon) 账户

在开始使用 Amazon DocumentDB 之前，您必须先拥有 Amazon Web Services (Amazon) 账户。Amazon 账户是免费的。您只需为使用的服务和资源付费。

如果您还没有 Amazon Web Services 账户，请完成以下步骤创建一个。

注册 Amazon Web Services 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

设置所需 Amazon Identity and Access Management(IAM) 权限。

访问以管理 Amazon DocumentDB 资源（如集群、实例和集群参数组）时需要提供以下凭证：Amazon 您可以使用来验证您的请求。有关更多信息，请参阅 [Amazon DocumentDB 中的 Identity and Access Management](#) (p. 142)。

1. 在 Amazon Web Services Management Console，键入 IAM 然后选择 IAM 在显示的下拉菜单中。

2. 在 IAM 控制台中，选择用户从导航窗格中。
3. 选择您的用户名。
4. 单击按钮添加权限。
5. 选择 Attach existing policies directly (直接附加现有策略)。
6. 类型AmazonDocDBFullAccess在搜索栏中，然后在搜索结果中出现后将其选中。
7. 点击底部显示的蓝色按钮后续：审核。
8. 点击底部显示的蓝色按钮添加权限。

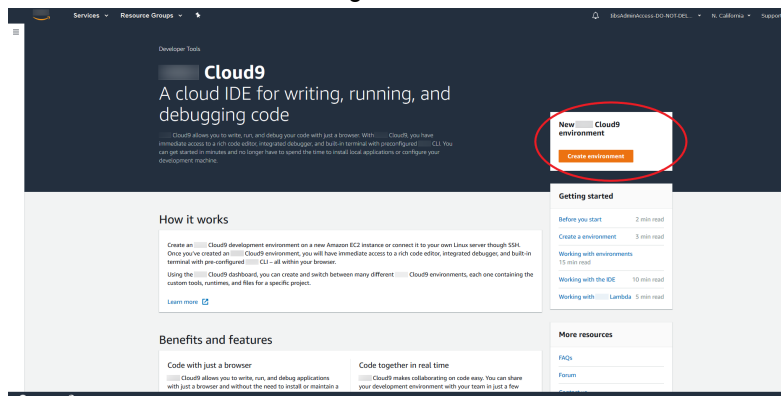
创建 Amazon Virtual Private Cloud (Amazon VPC)

只有在您还没有默认 Amazon VPC 时才需要此步骤。如果未安装，则完成[开始使用 Amazon VPC](#)中的 Amazon VPC User Guide. 这将需要不到五分钟的时间。

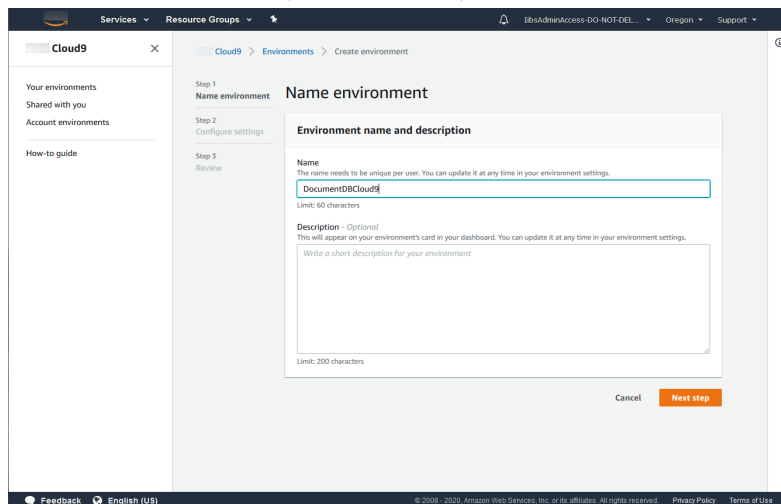
第 1 步：创建 Amazon Cloud9 环境

Amazon Cloud9提供了基于 Web 的终端，您可以使用 mongo shell 连接到 Amazon DocumentDB 集群并使用 Mongo shell 连接到和查询 Amazon DocumentDB 集群。

1. 从Amazon Web Services Management Console导航到Amazon Cloud9控制台并选择创建 环境。



2. 在环境名称和描述部分中，名称在字段中，输入DocumentDBCloud9。



3. 选择 Next step (下一步) 。

Amazon DocumentDB 开发人员指南

第 1 步：创建 Amazon Cloud9 环境

The screenshot shows the 'Name environment' step in the Amazon Cloud9 console. The left sidebar contains links for 'Your environments', 'Shared with you', 'Account environments', and 'How-to guide'. The main content area is titled 'Name environment' and includes a section 'Environment name and description'. Under 'Name', there is a text input field containing 'DocumentDBCloud9' and a note that the name must be unique per user. Under 'Description', there is a larger text area with a placeholder 'Write a short description for your environment'. At the bottom right, there are 'Cancel' and 'Next step' buttons, with the 'Next step' button highlighted by a red circle.

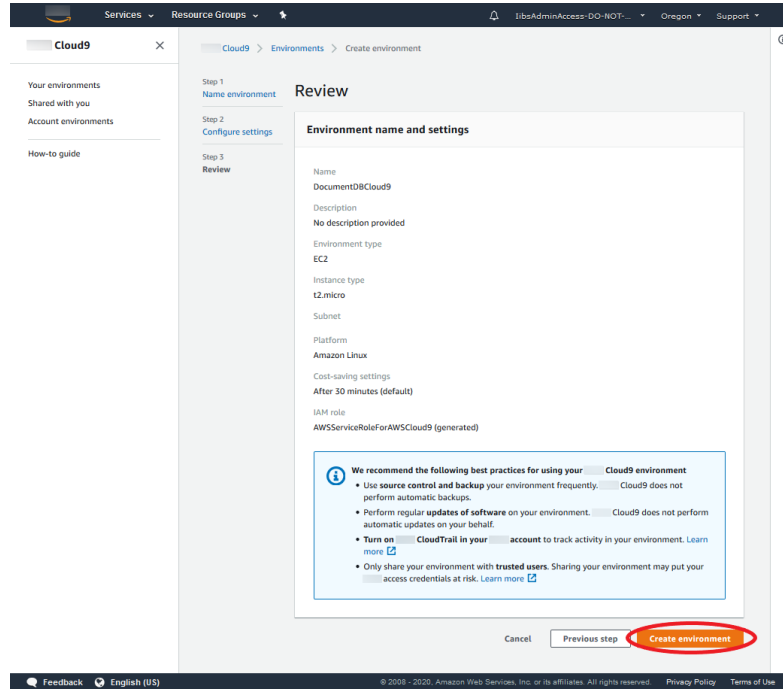
4. 在配置设置部分，选择下一步。

The screenshot shows the 'Configure settings' step in the Amazon Cloud9 console. The left sidebar is the same as in the previous step. The main content area is titled 'Configure settings' and includes a section 'Environment settings'. Under 'Environment type', there are three radio button options: 'Create a new EC2 instance for environment (direct access)' (selected), 'Create a new no-ingress EC2 instance for environment (access via Systems Manager)', and 'Create and run its remote server (SSH connection)'. Under 'Instance type', there are three radio button options: 't2.micro (1 GiB RAM + 1 vCPU)' (selected), 't3.small (2 GiB RAM + 2 vCPU)', and 'm5.large (8 GiB RAM + 2 vCPU)'. Under 'Platform', there are two radio button options: 'Amazon Linux' (selected) and 'Ubuntu Server 18.04 LTS'. Under 'Cost-saving setting', there is a dropdown menu set to 'After 30 minutes (default)'. Under 'IAM role', there is a text field containing 'AWSServiceRoleForAWSCloud9'. At the bottom right, there are 'Cancel', 'Previous step', and 'Next step' buttons, with the 'Next step' button highlighted by a red circle.

5. 在审核部分，选择创建 环境。

Amazon DocumentDB 开发人员指南

第 2 步：创建安全组



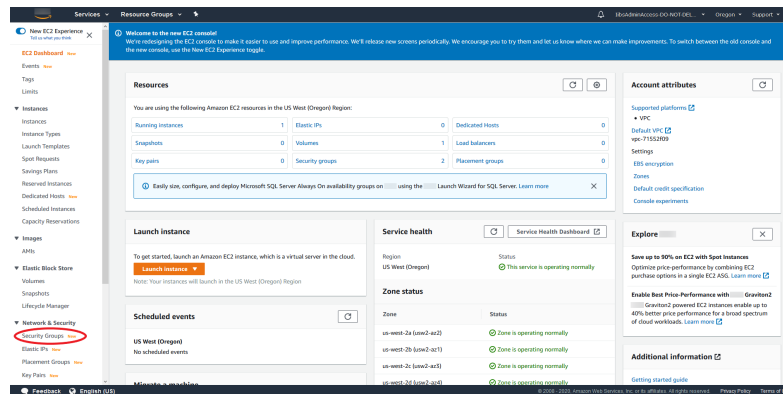
Note

预配置Amazon Cloud9环境耗时最多三分钟。

第 2 步：创建安全组

此安全组将使 Amazon DocumentDB 能够从您的Amazon Cloud9环境。

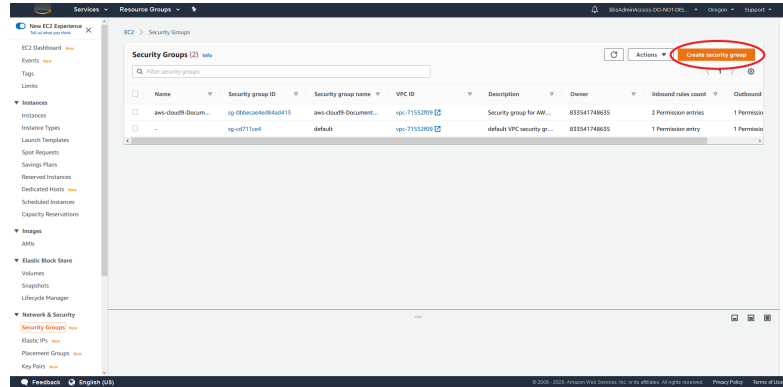
1. 在存储库的Amazon EC2 管理控制台，在网络和安全性，选择安全组。



2. 选择创建安全组。

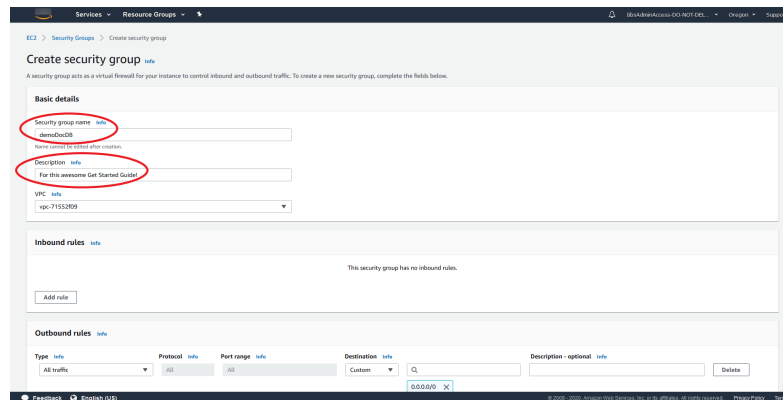
Amazon DocumentDB 开发人员指南

第 2 步：创建安全组

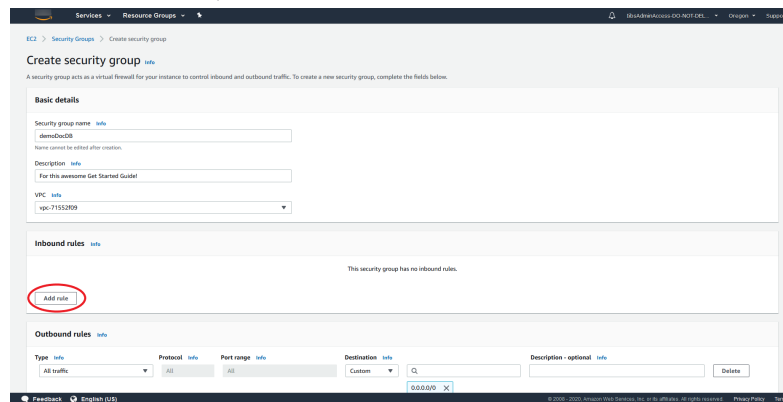


3. 在基本细节框中，为每个字段输入以下内容：

- 对于安全组名称，输入 demoDocDB。
- 对于说明，输入说明。
- 适用于 VPC，接受您的默认 VPC 的使用。



4. 在入站规则部分中，选择添加规则。



5. 对于 Type (类型)，选择 Custom TCP Rule (自定义 TCP 规则)。

Amazon DocumentDB 开发人员指南

第 2 步：创建安全组

The screenshot shows the 'Create security group' page in the Amazon Management Console. The 'Basic details' section is filled out with 'demoDocDB' as the security group name and 'sgc-7152809' as the VPC. In the 'Inbound rules' section, the 'Type' dropdown is set to 'Custom TCP', which is circled in red. The 'Protocol' is 'TCP' and the 'Port range' is '0'. The 'Outbound rules' section is also visible, with 'Type' set to 'All traffic'.

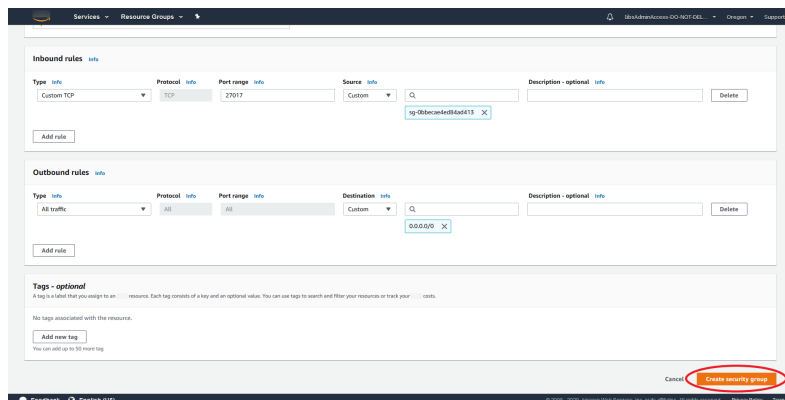
6. 适用于端口范围，输入27017.

This screenshot shows the 'Create security group' page with the 'Port range' field in the 'Inbound rules' section highlighted with a red circle. The value '27017' is entered in this field. The 'Type' is still 'Custom TCP' and the 'Protocol' is 'TCP'.

7. 来源是的安全组Amazon Cloud9您刚刚创建的环境。要查看可用安全组的列表，请输入cloud9在目标字段中。选择名称的安全组aws-cloud9-<environment name>.

This screenshot shows the 'Create security group' page with the 'Source' dropdown in the 'Inbound rules' section highlighted with a red circle. The dropdown menu is open, showing a list of security groups. The first option is 'aws-cloud9-DocumentDB...' with ID 'sg-c8bca4e4b94d413'. The 'Port range' field now contains '27017'.

8. 接受所有其他默认值并选择创建安全组.



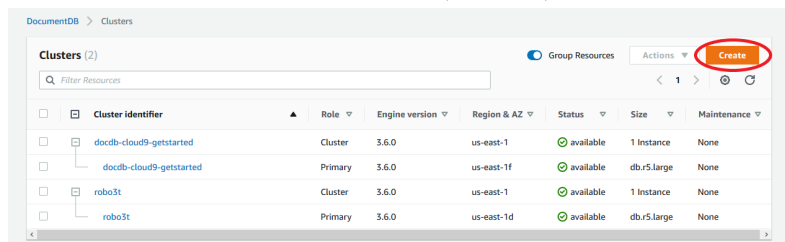
Note

端口 27017 是 Amazon DocumentDB 的默认端口。

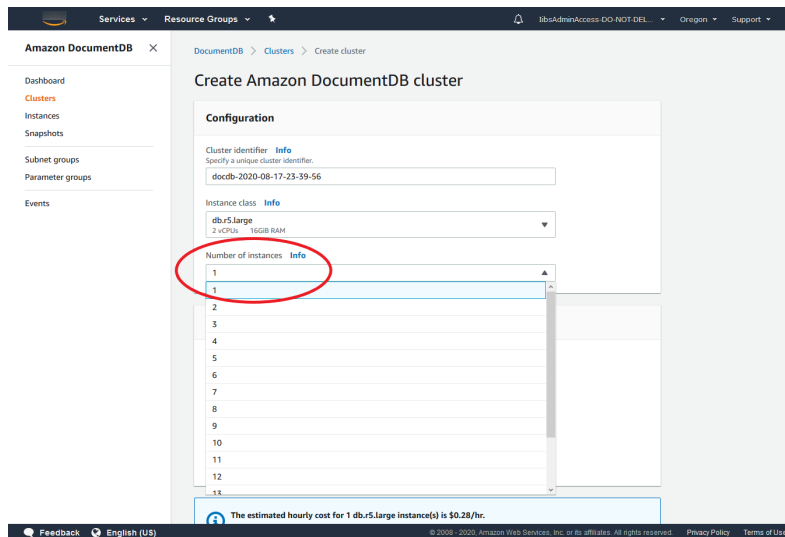
第 3 步：创建 Amazon DocumentDB 集群

在此步骤中，您将使用在上一步中创建的安全组创建 Amazon DocumentDB 集群。

1. 在 Amazon DocumentDB 管理控制台上，在集群，选择Create.



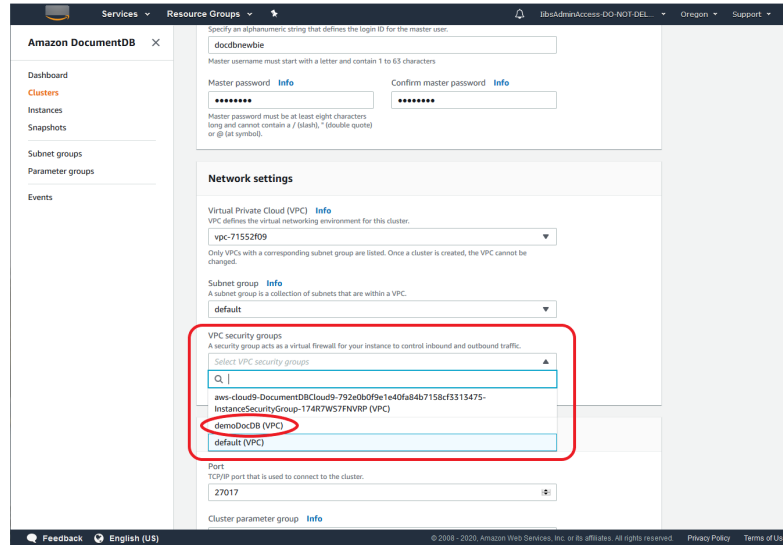
2. 在创建 Amazon DocumentDB 集群页面上的配置部分，选择1为了实例的数量. 选择一个实例有助于降低成本。如果这是生产系统，建议配置三个实例以实现高可用性。您可以将其他设置留在配置部分默认。



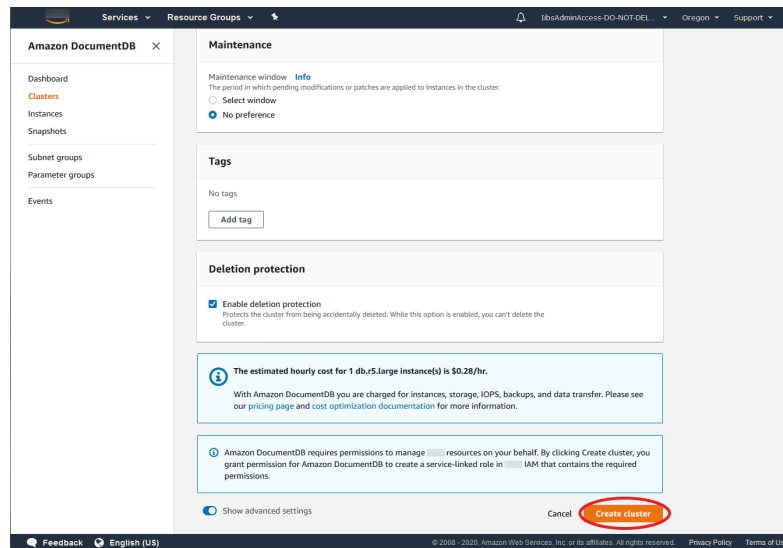
3. 在身份验证部分中，输入用户名和密码。

4. 启用。显示高级设置。

5. 在Network settings (网络设置)部分，VPC 安全组，选择demodocDB.



6. 选择创建集群。

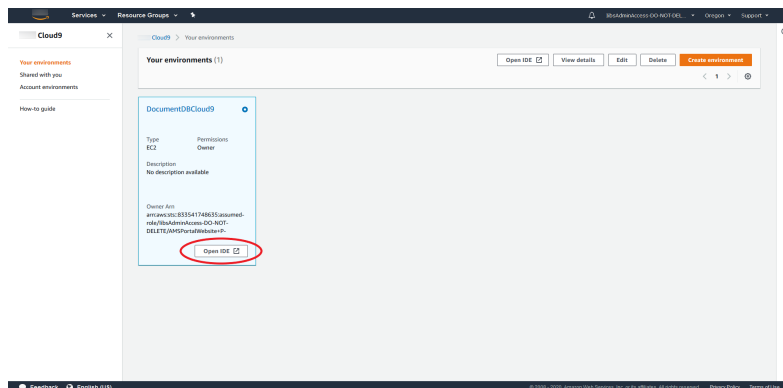


Amazon DocumentDB 现在正在配置集群，此操作可能需要几分钟时间才能完成。当群集和实例状态都显示为时，您可以连接到集群Available。

第 4 步：安装 mongo shell

现在您将在您的中安装 mongo shellAmazon Cloud9您在步骤 1 中创建的环境。mongo shell 是一个命令行应用程序，用于连接和查询 Amazon DocumentDB 集群。

1. 如果您的Amazon Cloud9环境从步骤 1 开始仍处于打开状态，返回到该环境并跳至说明 3。如果你远离你Amazon Cloud9环境一般，在Amazon Cloud9在管理控制台您的环境，查找标记的环境文档dbCloud9. 选择打开 IDE.



2. 在命令提示符处，使用以下命令创建存储库文件：

```
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. 完成后，使用以下命令安装 mongo shell：

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

4. 要加密传输中的数据，请从以下位置下载 Amazon DocumentDB 的公钥。<https://s3.amazonaws.com/rds-downloads/rds-combined-ca-cn-bundle.pem>。此操作将下载名为 rds-combined-ca-bundle.pem 的文件。

```
wget https://s3.amazonaws.com/rds-downloads/rds-combined-ca-cn-bundle.pem
```

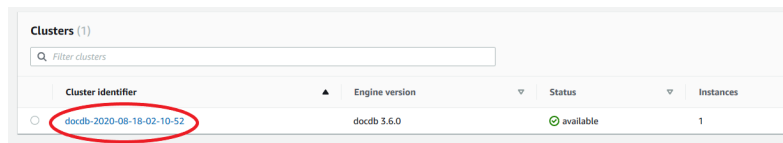
Note

Encryption-in-transit默认情况下，在 Amazon DocumentDB 上启用。您可以选择禁用 TLS。有关更多信息，请参阅 [管理 Amazon DocumentDB 集群 TLS 设置](#)。

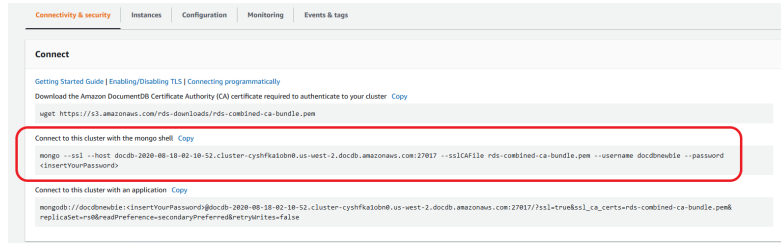
第 5 步：Connect 到您的 Amazon DocumentDB 集群

现在，您将使用步骤 4 中安装的 mongo shell 连接到 Amazon DocumentDB 集群。

1. 在 Amazon DocumentDB 管理控制台上，在集群中，找到你的集群。单击集群标识符来选择您创建的集群。



2. 在安全性选项卡，下使用 mongo shell Connect 到此集群中，复制提供的连接字符串。忽略复制<insertYourPassword>以便在连接时 Mongo shell 提示您输入密码。



3. 返回到您的Amazon Cloud9环境并粘贴连接字符串。

当您输入密码时提示符变为`rs0:PRIMARY>`提示符，您已成功连接到 Amazon DocumentDB 集群。

Note

有关排查故障的信息，请参阅[排查 Amazon DocumentDB](#)。

第 6 步：插入和查询数据

现在您已连接到集群，可以运行几个查询来熟悉文档数据库的使用。

1. 要插入单个文档，请输入以下内容：

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

2. 您会得到以下输出：

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

3. 您可以阅读你用`findOne()`命令（因为它只返回一个文档）。输入以下内容：

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

4. 您会得到以下输出：

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

5. 要执行更多查询，请考虑游戏配置文件使用案例。首先，将一些条目插入标题为`profiles`。输入以下内容：

```
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. 您会得到以下输出：

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

7. 使用`find()`命令返回配置文件集合中的所有文档。输入以下内容：

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

8. 您将获得与步骤 5 中键入的数据匹配的输出。
9. 使用筛选器对单个文档使用查询。输入以下内容：

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

10. 您应该收回这一输出：

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

11. 现在让我们尝试查找个人资料并使用 `findAndModify` 命令。我们将用以下代码给用户 Matt 额外的十点积分：

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

12. 你得到以下输出（请注意，他的分数还没有提高）：

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

13. 您可以通过以下查询验证他的分数是否已更改：

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

14. 您会得到以下输出：

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

步骤 7：了解

恭喜您！您已成功完成 Amazon DocumentDB 入门指南。

下一步是什么？了解如何充分利用此数据库的一些常见功能：

- [管理 Amazon DocumentDB](#)
- [扩展](#)
- [备份和还原](#)

Note

除非您将其删除，否则您在此入门练习中创建的集群将继续产生成本。有关指导，请参阅[删除 Amazon DocumentDB 集群](#)。

Amazon DocumentDB 快速入门使用 Amazon CloudFormation

此节包含使用以下步骤和其他信息，帮助您快速入门的 Amazon DocumentDB (兼容 MongoDB)。 [Amazon CloudFormation](#)。有关 Amazon DocumentDB 的一般信息，请参阅 [什么是 Amazon DocumentDB \(与 MongoDB 兼容\)](#) (p. 1)

这些说明使用 Amazon CloudFormation 以在您的默认 Amazon VPC 中创建集群和实例。有关自行创建这些资源的说明，请参阅 [开始使用 Amazon DocumentDB](#) (p. 28)。

Important

这些区域有：Amazon CloudFormation 通过该模板创建的堆栈创建了一个资源，包括中的资源（例如集群和实例）和 Amazon Elastic Compute Cloud（例如，子网组）。

其中一些资源并非免费套餐资源。有关定价信息，请参阅 [Amazon DocumentDB 定价](#) 和 [Amazon EC2 定价](#)。在完成后，您可以删除堆栈以停止产生任何费用。

此 Amazon CloudFormation 堆栈仅用于教程目的。如果您将此模板用于生产环境，建议您使用更严格的 IAM 策略和安全性。有关保护资源的信息，请参阅 [Amazon VPC 安全性](#) 和 [Amazon EC2 网络 and 安全性](#)。

主题

- [先决条件](#) (p. 40)
- [启动 Amazon DocumentDB Amazon CloudFormation 堆栈](#) (p. 42)
- [访问 Amazon DocumentDB 集群](#) (p. 45)
- [终止保护和删除保护](#) (p. 45)

先决条件

在创建 Amazon DocumentDB (具有 MongoDB 兼容性) 集群之前，您必须具有：

- 默认 Amazon VPC
- 所需的 IAM 权限

所需的 IAM 权限

以下权限允许您为 Amazon CloudFormation 堆栈创建资源：

Amazon 管理的策略

- `AWSCloudFormationReadOnlyAccess`
- `AmazonDocDBFullAccess`

其他 IAM 权限

以下策略概述了创建和删除此 Amazon CloudFormation 堆栈所需的其他权限。

```
{
  "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

上述策略中的加粗权限仅在删除堆栈时需

要：iam:DeleteRole、iam:RemoveRoleFromInstanceProfile、iam:DeleteRolePolicy、iam:DeleteInstanceProfile 和 ec2:DeleteVpcEndpoints。另请注意，ec2:*Vpc 授予 ec2:DeleteVpc 权限。

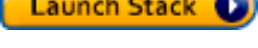
Amazon EC2 密钥对

在您创建 Amazon CloudFormation 堆栈的区域中，您必须具有可用的密钥对（和 PEM 文件）。如果您需要创建 key pair，请参阅[使用 Amazon EC2 创建密钥对](#)中的适用于 Linux 实例的 Amazon EC2 用户指南。

启动 Amazon DocumentDB Amazon CloudFormation 堆栈

本节介绍如何启动和配置 Amazon DocumentDB Amazon CloudFormation 堆栈。

1. 在 <https://console.amazonaws.cn/> 处登录 Amazon Web Services Management Console。
2. 下表列出了每种堆栈模板的 Amazon DocumentDB 堆栈模板。Amazon Web Services 区域。选择启动堆栈(对于)Amazon Web Services 区域要在中启动堆栈。

区域	查看模板	在 Designer 中查看	启动
美国东部 (俄亥俄)	查看模板	在 Designer 中查看	
美国东部 (弗吉尼亚北部)	查看模板	在 Designer 中查看	
美国西部 (俄勒冈)	查看模板	在 Designer 中查看	
亚太地区 (孟买)	查看模板	在 Designer 中查看	
亚太地区 (首尔)	查看模板	在 Designer 中查看	
亚太地区 (新加坡)	查看模板	在 Designer 中查看	
亚太地区 (悉尼)	查看模板	在 Designer 中查看	
亚太地区 (东京)	查看模板	在 Designer 中查看	
加拿大 (中部)	查看模板	在 Designer 中查看	
欧洲 (法兰克福)	查看模板	在 Designer 中查看	
欧洲 (爱尔兰)	查看模板	在 Designer 中查看	
欧洲 (伦敦)	查看模板	在 Designer 中查看	
欧洲 (巴黎)	查看模板	在 Designer 中查看	

3. 创建堆栈— 描述您选择的 Amazon DocumentDB 模板。每个堆栈都基于一个模板 (一个 JSON 或 YAML 文件) 其中包含有关 Amazon 要包含在堆栈中的资源。由于您选择从上面提供的模板启动堆栈, 因此您的模板已被配置为针对 Amazon Web Services 区域你选择了。

当你启动 Amazon CloudFormation 堆栈, [删除保护 \(p. 45\)](#) 对于 Amazon DocumentDB 集群, 在默认情况下处于禁用状态。如果要为集群启用删除保护, 请完成以下步骤。否则, 请选择 Next (下一步) 以继续执行下一步。

要为 Amazon DocumentDB 集群启用删除保护，请执行以下操作

1. 选择在 Designer 中查看来自创建堆栈页。
2. 在控制台生成的 Amazon CloudFormation 设计器页面中，使用集成的 JSON 和 YAML 编辑器修改模板。滚动到 Resources 部分并修改它以包括 DeletionProtection，如下所示。有关使用 Amazon CloudFormation 设计器的更多信息，请参阅[什么是 Amazon CloudFormation 设计器？](#)。

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. 从页面左上角选择 Create Stack (创建堆栈) () 以保存更改并创建启用这些更改的堆栈。
4. 保存更改后，您将被重定向到 Create stack (创建堆栈) 页面。
5. 选择 Next (下一步) 以继续。
4. 指定堆栈信息— 输入模板的堆栈名称和参数。参数在模板中定义，并允许您在创建或更新堆栈时输入自定义值。
 - 在 Stack name (堆栈名称) 下，输入堆栈的名称或接受提供的名称。堆栈名称可以包括字母 (A-Z 和 a—z)、数字 (0—9) 和破折号 (—)。
 - 在 Parameters (参数) 下，输入以下详细信息：
 - dbCluster 名称— 输入 Amazon DocumentDB 集群的名称或接受提供的名称。

集群命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个 Amazon RDS、Neptune 和 Amazon DocumentDB 中的所有集群必须是唯一的 Amazon Web Services 账户每个区域的数字。

- DBInstanceClass— 从下拉列表中，为 Amazon DocumentDB 集群选择实例类。
- dbName— 输入 Amazon DocumentDB 实例的名称或接受提供的名称。

实例命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个 Amazon RDS、Neptune 和 Amazon DocumentDB 中的所有实例必须是唯一的 Amazon Web Services 账户每个区域的数字。
- MasterPassword— 数据库管理员帐户密码。
- MasterUser— 数据库管理员帐户的用户名。MasterUser 必须以字母开头，且只能包含字母数字字符。

选择 Next (下一步) 以保存您的更改并继续。

5. 配置堆栈选项— 配置堆栈的标签、权限和其他选项。

- 标签— 指定要应用于堆栈中资源的标签 (键值) 对。您可以为每个堆栈添加最多 50 个唯一标签。
- Permissions (权限)— 可选。选择 IAM 角色以明确定义如何 Amazon CloudFormation 可以在堆栈中创建、修改或删除资源。如果您不选择角色，Amazon CloudFormation 会根据您的用户凭证使用权限。在指定服务角色之前，请确保您具有传递该角色的权限 (iam:PassRole)。iam:PassRole 权限指定您可以使用哪些角色。

Note

在您指定服务角色后，Amazon CloudFormation 始终对在该堆栈上执行的所有操作使用该角色。拥有对此堆栈执行操作的权限的其他用户将可以使用该角色，即使他们无权传递该角色也是如此。如果该角色包含用户不应具有的权限，则您可能无意中提升了用户的权限。确保该角色授予[最低权限](#)。

- 高级选项— 您可以设置以下高级选项：
 - Stack policy (堆栈策略)— 可选。定义在堆栈更新期间要防止意外更新的资源。默认情况下，堆栈更新期间所有资源都可更新。

您可以直接以 JSON 形式输入堆栈策略，也可以上传包含堆栈策略的 JSON 文件。有关更多信息，请参阅[防止更新堆栈资源](#)。

- 回滚配置— 可选。为指定 CloudWatch Logs 警报 Amazon CloudFormation 以监控创建和更新堆栈的时间。如果操作违反警报阈值，Amazon CloudFormation 将其回滚。
- 通知选项— 可选。指定简单通知系统 (SNS) 的主题。
- 堆栈创建选项— 可选。可以指定以下选项：
 - Rollback on failure (失败时回滚)— 如果堆栈创建失败，是否应回滚堆栈。
 - 超时— 堆栈创建超时之前的分钟数。
 - 终止保护— 防止意外删除堆栈。

Note

Amazon CloudFormationAmazon DocumentDB 删除保护概念不同。有关更多信息，请参阅[终止保护和删除保护 \(p. 45\)](#)。

选择 Next (下一步) 以继续。

6. 回顾 <stack-name>— 审核您的堆栈模板、详细信息和配置选项。您还可以在页面底部打开 quick-create link (快速创建链接)，以创建与此页面具有相同基本配置的堆栈。

- 选择 Create (创建) 以创建堆栈。
- 或者，您也可以选择 Create change set (创建更改集)。更改集是对在创建堆栈之前如何配置此堆栈的预览。这允许您在执行更改集之前检查各种配置。

访问 Amazon DocumentDB 集群

一旦 Amazon CloudFormation 堆栈已完成，您可以使用 Amazon EC2 实例连接到您的 Amazon DocumentDB 集群。有关使用 SSH 连接到 Amazon EC2 实例的信息，请参阅[连接到 Linux 实例](#)中的适用于 Linux 实例的 Amazon EC2 用户指南。

连接后，请参阅以下各节，其中包含了有关使用 Amazon DocumentDB 的信息。

- [第 4 步：安装 mongo shell \(p. 36\)](#)
- [删除 Amazon DocumentDB 集群 \(p. 267\)](#)

终止保护和删除保护

启用删除保护和终止保护是 Amazon DocumentDB 的最佳实践。CloudFormation 终止保护与 Amazon DocumentDB 删除保护功能截然不同。

- 终止保护—您可以通过为 CloudFormation 堆栈启用终止保护来防止堆栈被意外删除。如果用户尝试删除已为其启用终止保护的堆栈，则删除操作会失败，并且堆栈将保持不变。默认情况下，当您使用 CloudFormation 创建堆栈时，终止保护处于禁用状态。可在创建堆栈时对其启用终止保护。有关更多信息，请参阅[设置 Amazon CloudFormation 堆栈选项](#)。
- 删除保护 Amazon DocumentDB 还提供了对集群启用删除保护的功能。如果用户尝试删除已为其启用删除保护的 Amazon DocumentDB 集群，则删除将失败，并且集群保持不变。启用删除保护时，可防止从 Amazon DocumentDB 意外删除。Amazon Web Services Management Console、Amazon CLI 和 CloudFormation。有关为 Amazon DocumentDB 集群启用和禁用删除保护的更多信息，请参阅[删除保护 \(p. 267\)](#)。

MongoDB 4.0 兼容性

Amazon DocumentDB 支持 MongoDB 4.0 兼容性，包括 ACID 交易。MongoDB 4.0 兼容性意味着您目前已经在 MongoDB 4.0 数据库中使用的绝大多数应用程序、驱动程序和工具都可以与 Amazon DocumentDB 4.0 一起使用，几乎没有更改或根本没有更改。本节介绍了您需要了解的关于 Amazon DocumentDB 4.0 的所有信息，包括新功能和功能、入门、迁移路径和功能差异。

主题

- [Amazon DocumentDB 4.0 中的新增功能 \(p. 46\)](#)
- [开始使用 Amazon DocumentDB 4.0 \(p. 47\)](#)
- [升级或迁移到 Amazon DocumentDB 4.0 \(p. 47\)](#)
- [功能差异 \(p. 47\)](#)

Amazon DocumentDB 4.0 中的新增功能

Amazon DocumentDB 4.0 引入了许多新功能和功能，其中包括 ACID 事务和对变更流的改进。下面的摘要介绍了 Amazon DocumentDB 4.0 中引入的一些主要功能。要查看新功能的完整列表，请参阅[发行说明 \(p. 714\)](#)。

- **ACID 事务**：Amazon DocumentDB 现在支持跨多个文档、对账单、集合和数据库执行交易的功能。事务使您能够跨 Amazon DocumentDB 集群中的一个或多个文档执行原子、一致、隔离和持久 (ACID) 操作，从而简化应用程序开发。有关更多信息，请参阅[事务 \(p. 49\)](#)。
- **Change streams**：您现在可以在集群级别打开变更直播 (`client.watch()` 要么 `mongo.watch()`) 和数据库 (`db.watch()`)，您可以指定 `startAtOperationTime` 打开更改流光标，最后，您现在可以将更改流的保留期延长到 7 天 (之前的 24 小时)。有关更多信息，请参阅[在 Amazon DocumentDB 中使用变更流 \(p. 445\)](#)。
- **Amazon Database Migration Service (Amazon DMS)**：您现在可以使用 Amazon DMS 将 MongoDB 4.0 工作负载迁移到 Amazon DocumentDB。Amazon DMS 现在支持 MongoDB 4.0 源代码、Amazon DocumentDB 4.0 目标和用于在 Amazon DocumentDB 3.6 和 4.0 之间执行升级的 Amazon DocumentDB 3.6 源代码。有关更多信息，请参阅[Amazon DMS 文档](#)。
- **性能和索引**：你现在可以使用索引 `$lookup` 中，查找具有包含一个字段或一个字段的投影的查询，`_id` 字段可以直接从索引中提供，而无需从集合中读取 (覆盖的查询)，`hint()` 和 `findAndModify`，针对的性能优化 `$addToSet`，以及减少总体指数规模的改进。有关更多信息，请参阅[发行说明 \(p. 714\)](#)。
- **运算符**：Amazon DocumentDB 4.0 现在支持许多新的聚合运算符：`$ifNull`、`$replaceRoot`、`$setIsSubset`、`$setIntersection`、`$setUnion`、`$setEquals`。您可以在上查看我们支持的所有 MongoDB API、操作和数据类型[支持的 MongoDB API、操作和数据类型 \(p. 93\)](#)。
- **基于角色的访问控制 (RBAC)**：有两者的 `ListCollection` 和 `ListDatabase` 你现在可以选择使用 `authorizedCollections` 和 `authorizedDatabases` 参数允许用户列出他们有权访问的集合和数据库，而无需 `listCollections` 和 `listDatabase` 角色。你也可以杀死自己的光标，而无需 `KillCursor` 角色。

Amazon DocumentDB 不支持所有 MongoDB 4.0 功能。当我们构建 Amazon DocumentDB 4.0 时，我们从客户要求我们构建最多的功能和功能向后工作。我们将根据客户要求我们构建的内容，继续添加额外的 MongoDB 4.0 功能。例如，Amazon DocumentDB 4.0 当前不支持 MongoDB 4.0 中引入的类型转换运算符或字符串运算符。有关受支持的 API 的最新列表，请参阅[支持的 MongoDB API、操作和数据类型 \(p. 93\)](#)。

开始使用 Amazon DocumentDB 4.0

要开始使用 Amazon DocumentDB 4.0，请参阅[入门指南](#)。您可 Amazon DocumentDB 使用 Amazon Web Services Management Console 或者 Amazon 开发工具包 Amazon CLI，或者 Amazon CloudFormation。连接到 Amazon DocumentDB 时，需要使用与 MongoDB 4.0 或更高版本兼容的 MongoDB 驱动程序或实用程序。

Note

当使用 Amazon 开发工具包 Amazon CLI，或者 Amazon CloudFormation，引擎版本默认为 3.6。您必须明确指定参数 `engineVersion=4.0` 创建新的 Amazon DocumentDB 4.0.0 集群。对于给定的 Amazon DocumentDB 集群，您可以使用 Amazon CLI 调用 `describe-db-clusters` 或者使用 Amazon DocumentDB 管理控制台查看特定集群的引擎版本号。

Amazon DocumentDB 4.0 支持 `r5` 和 `t3.medium` 适用于您的集群的实例类型，并在所有支持的区域中提供。使用 Amazon DocumentDB 4.0 不收取任何额外费用。有关定价的更多信息，请参阅[Amazon DocumentDB \(与 MongoDB 兼容\) 定价](#)。

升级或迁移到 Amazon DocumentDB 4.0

您可以 Amazon DocumentDB 用 [Amazon DMS](#) 或者像公用事业 `mongodump`、`mongoexport`、`mongoimport`，和 `mongoexport`。同样，您可以使用相同的工具从 Amazon DocumentDB 3.6 升级到 Amazon DocumentDB 4.0。有关如何迁移的说明，请参阅[使用您的 Amazon DocumentDB 集群从 3.6 升级到 4.0](#) [Amazon Database Migration Service \(p. 112\)](#)。

功能差异

Amazon DocumentDB 3.6 和 4.0 之间的功能差异

随着 Amazon DocumentDB 4.0 的发布，Amazon DocumentDB 3.6 和 Amazon DocumentDB 4.0 之间存在功能差异：

- 嵌套文档的投影：在应用投影时，Amazon DocumentDB 3.6 考虑嵌套文档中的第一个字段。但是，Amazon DocumentDB 4.0 将解析子文档并将投影应用于每个子文档。例如：如果投影是 `"a.b.c": 1`，那么两个版本中的行为都是相同的。但是，如果预测是 `{a:{b:{c:1}}}` 那么 Amazon DocumentDB 3.6 只会将投影应用于 `"a"` 而不是 `"b"` 或 `"c"`。
- 适用于的 `minKey`、`maxKey`：在 Amazon DocumentDB 4.0 中，`{x:{$gt:MaxKey}}` 什么都不返回，而且 `{x:{$lt:MaxKey}}` 返回所有内容。
- 文档比较差异：比较子文档中不同类型（double、int、long）的数值（例如，`b` 在 `{"_id": 1, "a": {"b": 1}}`）现在可以跨数字数据类型和文档的每个级别提供一致的输出。

Amazon DocumentDB 4.0 和 MongoDB 4.0 之间的功能差异

以下是 Amazon DocumentDB 4.0 和 MongoDB 4.0 之间的功能差异。

- 在路径中使用空键查找：当集合中包含数组内有空键的文档时（例如 `{"x": [ {"": 10 }, {"b": 20 } ]}`），以及查询中使用的密键以空字符串结束时（例如 `x.`），那么 Amazon DocumentDB 将返回该文档，因为它会遍历阵列中的所有文档，而 MongoDB 将不返回该文档。

- **\$setOnInsert**以及\$在路径中：现场操作员\$setOnInsert无法与配合使用\$在 Amazon DocumentDB 中的路径中，这也与 MongoDB 4.0 一致。

事务

Amazon DocumentDB (与 MongoDB 兼容) 现在支持 MongoDB 4.0 兼容性, 包括事务。您可以跨多个文档、对账单、集合和数据库执行交易。事务使您能够跨 Amazon DocumentDB 集群中的一个或多个文档执行原子、一致、隔离和持久 (ACID) 操作, 从而简化应用程序开发。交易的常见用例包括财务处理、配送和管理订单以及构建多玩家游戏。

事务不收取额外费用。您只需为作为交易一部分使用的读写 I/O 付费。

主题

- [要求 \(p. 49\)](#)
- [最佳实践 \(p. 49\)](#)
- [限制 \(p. 49\)](#)
- [监控和诊断 \(p. 50\)](#)
- [事务隔离级别 \(p. 50\)](#)
- [使用案例 \(p. 51\)](#)
- [支持的 命令 \(p. 75\)](#)
- [不支持的功能 \(p. 75\)](#)
- [会话 \(p. 76\)](#)
- [事务错误 \(p. 77\)](#)

要求

要使用事务功能, 您需要满足以下要求:

- 您必须使用 Amazon DocumentDB 4.0 引擎。
- 您必须使用与 MongoDB 4.0 或更高版本兼容的驱动程序。

最佳实践

以下是一些最佳实践, 以便您可以最大限度地利用 Amazon DocumentDB 的交易。

- 事务完成后, 始终提交或中止。使事务处于不完整状态会关系数据库资源, 并可能导致写入冲突。
- 建议将事务保持在所需的最小数量的命令。如果您的交易包含多个语句可以分为多个较小的交易, 那么建议这样做以减少超时的可能性。始终致力于创建短交易, 而不是长时间运行的读取。

限制

- Amazon DocumentDB 不支持交易中的游标。
- Amazon DocumentDB 无法在交易中创建新的收藏夹, 也无法针对不存在的收藏进行查询/更新。
- 文档级写入锁定需要 1 分钟的超时, 用户无法配置超时。
- 不支持可重试写入、可重试的提交和可重试中止。
- 每个 Amazon DocumentDB 实例对该实例一次打开的并发交易数量都有上限。有关限制, 请参阅[实例限制 \(p. 500\)](#)。
- 对于给定的事务, 事务日志大小必须小于 32MB。

- Amazon DocumentDB 确实支持`count()`在交易中，但并非所有司机都支持此功能。另一种选择是使用`countDocuments()`API，它将计数查询转换为客户端的聚合查询。
- 交易的执行限制为 1 分钟，会话有 30 分钟的超时时间。如果事务超时，它将被中止，并且在会话中针对现有事务发出的任何后续命令都将产生以下错误：

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

监控和诊断

借助 Amazon DocumentDB 4.0 中对交易的支持，添加了其他 CloudWatch 指标来帮助您监控交易。

新的 CloudWatch 指标

- `DatabaseTransactions`：在一分钟时间内进行的未结交易的数量。
- `DatabaseTransactionsAborted`：在一分钟时间内进行的中止交易的数量。
- `DatabaseTransactionsMax`：一分钟内的最大未结交易数量。
- `TransactionsAborted`：在一分钟内实例中止的交易数。
- `TransactionsCommitted`：在一分钟内在实例上提交的交易数。
- `TransactionsOpen`：在一分钟时间内在实例上打开的交易数。
- `TransactionsOpenMax`：在一分钟内实例上打开的最大交易数。
- `TransactionsStarted`：在一分钟内在实例上开始的交易数。

Note

有关 Amazon DocumentDB 的更多 CloudWatch 指标，请转到[使用 CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)。

此外，两者都添加了新字段`currentOp lsid`、`transactionThreadId`，还有一个新的状态“idle transaction”和`serverStatus`事务：`currentActive`、`currentInactive`、`currentOpen`、`totalAborted`、`totalCommitted`，和`totalStarted`。

事务隔离级别

开始交易时，你可以同时指定`readConcern`和`writeConcern`如以下示例所示：

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

适用于`readConcern`，默认情况下，Amazon DocumentDB 支持快照隔离。如果`readConcern`如果指定了本地、可用或大多数，Amazon DocumentDB 将升级`readConcern`关卡到快照。Amazon DocumentDB 不支持线性化`readConcern`并指定此类读取问题将导致错误。

适用于`writeConcern`，默认情况下，Amazon DocumentDB 支持大多数，当四个数据副本在三个可用区中持久存储时，将达到写法定人数。如果较低`writeConcern`已指定，Amazon DocumentDB 将升级`writeConcern`到大多数。此外，所有 Amazon DocumentDB 写入操作都会被记录，并且不能禁用日志记录。

使用案例

在本节中，我们将介绍两个交易使用案例：多对账单和多重收集。

多对账单交易

Amazon DocumentDB 交易是多语句，这意味着您可以通过显式提交或回退编写跨多个语句的交易。您可以分组insert、update、delete, 和findAndModify作为单个原子操作。

多对账单交易的一个常见用例是借记信贷交易。例如：你欠朋友钱买衣服。因此，您需要从账户中借记（提取）500 美元，并将 500 美元（存款）存入朋友的账户。要执行该操作，您可以在单笔交易中同时执行债务和信贷操作，以确保原子性。这样做可以防止从您的账户中扣除 500 美元但不记入朋友账户的情况。这是这个用例的样子：

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

多项收款交易

我们的交易也是多个收款，这意味着它们可以用来在单个交易中以及跨多个集合执行多项操作。这提供了一致的数据视图，并维护数据的完整性。当你将命令作为单个提交时<>，这些交易是全部或无执行——因此，它们要么全部成功或全部失败。

下面是一个多收款交易的示例，使用相同的场景和多对账单交易示例中的数据。

```
// *** 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
```

回调 API 的交易 API 示例

回调 API 仅适用于 4.2 多个驱动程序。

Javascript

以下代码说明如何在 Javascript 中使用 Amazon DocumentDB 事务 API。

```
// *** 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

以下代码说明如何在 Node.js 中使用 Amazon DocumentDB 事务 API。

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

以下代码说明如何在 C# 中使用 Amazon DocumentDB 事务 API。

```
// 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

以下代码说明如何在 Ruby 中使用 Amazon DocumentDB 事务 API。

```
// 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

以下代码说明了如何在 Go 的情况下使用 Amazon DocumentDB 事务 API。

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

```

Java

以下代码说明如何将 Amazon DocumentDB 事务 API 与 Java 一起使用。

```

// 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

以下代码演示如何在 C 中使用 Amazon DocumentDB 事务 API。

```
// 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

以下代码说明如何在 Python 中使用 Amazon DocumentDB 事务 API。

```
// 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
```

核心 API 的交易 API 示例

Javascript

以下代码说明如何在 Javascript 中使用 Amazon DocumentDB 事务 API。

```
// *** 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#

以下代码说明如何在 C# 中使用 Amazon DocumentDB 事务 API。

```
// 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

以下代码说明如何在 Ruby 中使用 Amazon DocumentDB 事务 API。

```
# 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

以下代码说明如何将 Amazon DocumentDB 事务 API 与 Java 一起使用。

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

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

  MongoDBDatabase 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.TRANIENT_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);

    MongoDB database = mongoClient.getDatabase("bank");
    MongoCollection accountColl = database.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.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
observation", e);
    }
    if (!errors.isEmpty()) {
        throw errors.get(0);
    }
    return this;
}

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

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

```

C

以下代码演示如何在 C 中使用 Amazon DocumentDB 事务 API。

```

// 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

以下代码说明如何将 Amazon DocumentDB 事务 API 与 Scala 一起使用。

```
// 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.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()
}
```

支持的 命令

命令	支持
abortTransaction	是
commitTransaction	是
endSessions	是
killSession	是
killAllSession	是
killAllSessionsByPattern	否
refreshSessions	否
startSession	是

不支持的功能

方法	阶段或命令
db.collection.aggregate()	\$collStats

方法	阶段或命令
	<code>\$currentOp</code> <code>\$indexStats</code> <code>\$listSessions</code> <code>\$out</code>
<code>db.collection.count()</code> <code>db.collection.countDocuments()</code>	<code>\$where</code> <code>\$near</code> <code>\$nearSphere</code>
<code>db.collection.insert()</code>	<code>insert</code> 如果它不是针对现有集合运行，则不受支持。如果此方法以预先存在的集合为目标，则支持此方法。

会话

MongoDB 会话是一个框架，用于支持可重试写入、因果一致性、事务以及跨数据库管理操作。创建会话时，客户端生成逻辑会话标识符 (lsid)，用于在向服务器发送命令时标记该会话中的所有操作。

Amazon DocumentDB 支持使用会话来启用交易，但不支持因果一致性或可重试写入。

在 Amazon DocumentDB 中使用交易时，将使用 `session.startTransaction()` API 和会话一次支持单个交易。同样，事务可以使用提交完成 (`session.commitTransaction()`) 或者中止 (`session.abortTransaction()`) API。

因果一致性

因果一致性保证在单个客户端会话中，客户端将观察到后写后读一致性、单原子读/写操作，并且这些保证适用于群集中的所有实例，而不仅仅是主实例。Amazon DocumentDB 不支持因果一致性，以下陈述将导致错误。

```
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()
```

您可以在会话中禁用因果一致性。请注意，这样做将使您能够利用会话框架，但不会为读取提供因果一致性保证。使用 Amazon DocumentDB 时，从主数据库中的读取将保持一致性，从副本实例进行的读取将最终保持一致。事务是利用会话的主要用例。

```
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 }
```

可重试写入

可重试写入是一种能力，在此功能中，客户端将尝试重试写操作一次，当发生网络错误或者客户端无法找到主写操作时。在 Amazon DocumentDB 中，不支持可重试写入操作，必须禁用。你可以用命令禁用它 (`retryWrites=false`) 在连接字符串中。以下是示例：

```
mongodb://chimera:<insertYourPassword>@docdb-2019-01-29-02-57-28.cluster-ccuszb3pn5e.us-east-1.docdb.amazonaws.com:27017/?ssl=true&ssl_ca_certs=rds-combined-ca-cn-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false
```

事务错误

在使用交易时，有些情况可能会导致错误，指出交易编号与任何正在进行的交易都不匹配。

至少可以在两种不同的情况下生成错误：

- 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.

处理此错误的最佳方法是使事务更新成为幂等的 — 例如，使用 `$set` 变器而不是增量/递减操作。请参阅下文：

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

Amazon DocumentDB 的最佳实践

学习使用 Amazon DB (with MongoDB compatibility) 的最佳实践。随着新的最佳实践的确定，此节将不断更新。

主题

- [基本操作指导方针](#) (p. 78)
- [调整实例大小](#) (p. 79)
- [使用索引](#) (p. 79)
- [安全最佳实践](#) (p. 80)
- [成本优化](#) (p. 81)
- [使用指标确定性能问题](#) (p. 81)
- [TTL 和时间序列工作负载](#) (p. 83)
- [迁移](#) (p. 83)
- [使用集群参数组](#) (p. 83)
- [聚合管道查询](#) (p. 83)
- [batchInsert 和 batchUpdate](#) (p. 83)

基本操作指导方针

以下是使用 Amazon DocumentDB 时每个人都应遵循的基本操作指导方针。Amazon DocumentDB 服务等级协议要求您遵循这些指导方针。

- 部署一个由两个或更多 Amazon DocumentDB 实例一分为二的集群Amazon可用区。对于生产工作负载，我们建议在三个可用区中部署一个由三个或更多 Amazon DocumentDB 实例组成的集群。
- 在规定的服务限制内使用服务。有关更多信息，请参阅 [Amazon DocumentDB 配额和限制](#) (p. 496)。
- 监控您的内存、CPU、连接和存储使用情况。为了帮助您保持系统的性能和可用性，请设置 Amazon CloudWatch 在使用模式发生更改或接近部署容量时向您发送通知。
- 当接近容量限制时，可以向上扩展您的实例。您应为这些实例预配置足够的计算资源（即 RAM、CPU），以满足无法预测的应用程序需求增长。
- 设置您的备份保留期以与您的恢复点目标保持一致。
- 测试您的集群的故障转移，以了解对于您的使用案例而言，该过程需要多长时间。有关更多信息，请参阅 [Amazon DocumentDB 故障转移](#) (p. 313)。
- 使用集群终端节点Connect 您的 Amazon DocumentDB 集群（参见[Amazon DocumentDB D](#) (p. 7)）和在副本集模式下（参见[作为副本集连接到 Amazon DocumentDB](#) (p. 454)）以最大限度地减少故障转移对应用程序的影响。
- 选择一项驱动程序读取首选项设置，以便在满足应用程序的读取一致性要求的同时最大程度地提高读取扩展能力。secondaryPreferred 读取首选项将启用副本读取，并释放主实例以执行更多工作。有关更多信息，请参阅 [读取首选项选项](#) (p. 11)。
- 将您的应用程序设计为在出现网络和数据库错误时能够灵活应对。使用您的驱动程序的错误机制来区分临时错误和持久性错误。适当时使用指数回退机制重试临时错误。确保您的应用程序在实施重试逻辑时考虑数据一致性。
- 为所有生产集群或包含重要数据的任何集群启用集群删除保护。在删除 Amazon DocumentDB 集群之前，请拍摄最终快照。如果要使用 Amazon CloudFormation 部署资源，请启用终止保护。有关更多信息，请参阅 [终止保护和删除保护](#) (p. 45)。

- 创建 Amazon DocumentDB 集群时，`—engine-version` 是一个可选参数，默认为最新的主引擎版本。当前的主引擎版本为 4.0.0。发布新的主要引擎版本时，`—engine-version` 的默认引擎版本将更新，以反映最新的主要引擎版本。因此，对于生产工作负载，尤其是那些依赖脚本编写、自动化或 Amazon CloudFormation 模板，我们建议您将 `—engine` 版本明确指定为预期的主要版本。

调整实例大小

在 Amazon DocumentDB 中选择实例大小的最关键方面之一是缓存的 RAM 量。Amazon DocumentDB 为自己的服务保留了三分之一的 RAM，这意味着只有三分之二的实例 RAM 可用于缓存。因此，Amazon DocumentDB 的最佳做法是选择具有足够内存的实例类型，以容纳内存中的工作集（即数据和索引）。保证实例具有适当大小有助于优化整体性能，并可能最大限度地降低 I/O 成本。您可以使用第三方 [Amazon DocumentDB S](#) 估算特定工作负载的实例大小。

要确定应用程序的工作集是否适合内存，请监视 `BufferCacheHitRatio` 使用 Amazon CloudWatch 对于集群中处于负载状态的每个实例。

这些区域有：`BufferCacheHitRatio` CloudWatch 指标衡量从实例的内存缓存中提供的数据和索引的百分比（相对于存储容量）。一般来说，`BufferCacheHitRatio` 的值应该尽可能高，因为从工作集内存读取数据比从存储卷读取数据更快、更具成本效益。虽然保证 `BufferCacheHitRatio` 尽可能接近 100% 是最理想的，但可实现的最佳值将取决于您的应用程序的访问模式和性能要求。为保证 `BufferCacheHitRatio` 尽可能高，建议为您集群中的实例配置足够 RAM，以便能够在内存中保存索引和工作数据集。

如果您的索引不在内存中，您就会看到一个较低的 `BufferCacheHitRatio`。相比从内存中读取，持续从磁盘中读取会产生额外的 I/O 开销，并且性能也更低。如果您的 `BufferCacheHitRatio` 比率低于预期，请上调集群的实例大小，以提供更多 RAM，以便将工作集数据容纳在内存中。如果上调实例大小会导致 `BufferCacheHitRatio` 大幅提高，就表明应用程序的工作集不在内存中。继续向上扩展，直至 `BufferCacheHitRatio` 在扩展操作后不再大幅上升。有关监控实例的指标的信息，请参阅 [Amazon DocumentDB 指标](#) (p. 376)。

根据您的工作负载和延迟要求，也许可以让您的应用程序在稳定状态使用期间具有较高 `BufferCacheHitRatio` 值，然后定期降低 `BufferCacheHitRatio`，因为分析实例需要扫描实例上运行的整个集合。定期降低 `BufferCacheHitRatio` 可能会很明显，因为后续查询需要将工作集数据从存储卷复制回缓冲区缓存，所以延迟时间较长。我们建议您首先在具有代表性生产工作负载的预生产环境中测试您的工作负载，以便了解性能特征和 `BufferCacheHitRatio`，然后再将工作负载部署到生产环境。

`BufferCacheHitRatio` 是特定于实例的指标，因此同一集群中不同实例可能具有不同的 `BufferCacheHitRatio` 值，具体取决于读取在主实例和副本实例之间的分配方式。如果运行工作负载无法处理因运行分析查询后重新填充工作集缓存而导致的定期延迟增加，应尝试将常规工作负载的缓冲缓存与分析查询的缓冲缓存隔离开。您可以通过将操作查询指向主实例，将分析查询指向仅指向副本实例，实现完全的 `BufferCacheHitRatio` 隔离。您还可以通过将分析查询定向到特定副本实例来实现部分隔离，但要知道有一定百分比的常规查询也将在该副本上运行，并且可能受到影响。

`BufferCacheHitRatio` 的值是否适当取决于您的使用案例和应用程序要求。此指标没有一个最佳值或最小值；只有您可以从成本和性能的角度决定是否可以接受 `BufferCacheHitRatio` 暂时较低的后果。

使用索引

建立索引

将数据导入 Amazon DocumentDB 时，应在导入大型数据集之前创建索引。您可以使用 [Amazon DocumentDB 索引](#) 从正在运行的 MongoDB 实例中提取索引或 `mongodump` 目录，然后在 Amazon DocumentDB 集群中创建这些索引。有关迁移的更多指导，请参阅 [迁移 Amazon DocumentDB](#) (p. 112)。

索引选择性

我们建议您将索引创建限制为其中的重复值数量低于集合中总文档数的 1% 的字段。例如，如果您的集合包含 100,000 个文档，则仅在相同值出现 1000 次或更少的字段上创建索引。

选择具有大量唯一值（即高基数）的索引可确保筛选操作返回少量文档，从而在索引扫描期间产生良好的性能。高基数索引的一个例子是一个唯一索引，它保证相等谓词最多返回单个文档。低基数的示例包括布尔字段上的索引和一周中某天的索引。由于性能差，数据库的查询优化程序不太可能选择低基数索引。同时，低基数索引将继续消耗磁盘空间和 I/O 等资源。作为经验法则，应将索引定位于典型值频率为总集合大小的 1% 或更小的字段。

此外，建议仅在通常用作筛选条件的字段上创建索引，并定期查找未使用的索引。有关更多信息，请参阅 [如何识别未使用的索引？](#) (p. 526)。

索引对数据写入的影响

虽然索引可以通过避免扫描集合中的每个文档来提高查询性能，但这种提高是有代价的。对于集合的每个索引，每次插入、更新或删除文档时，数据库都必须更新集合并将字段写入集合的每个索引。例如，如果某个集合有九个索引，则数据库必须执行十次写入操作，才能将操作通知给客户端。因此，每增加一个索引就会增加一份写入延迟、I/O 开销和占用的总存储空间。

所以应该适当调整集群实例的大小，以确保将所有工作集容纳在内存中。这样可以避免持续从存储卷读取索引页，这会影响性能并产生更高的 I/O 开销。有关更多信息，请参阅 [调整实例大小](#) (p. 79)。

为了获得最佳性能，请尽量减少集合中的索引数量，仅添加必要的索引来提高常用查询的性能。虽然工作负载会变化，但有一个很好的指导原则，就是将每个集合的索引数量保持在五个以内。

识别缺失的索引

识别缺失的索引是我们建议定期执行的最佳做法。有关详细信息，请参阅 [如何识别缺失的索引？](#) (p. 527)。

识别未使用的索引

我们建议您定期识别并删除未使用的索引，这是一种很好的做法。有关详细信息，请参阅 [如何识别未使用的索引？](#) (p. 526)。

安全最佳实践

要获得最佳安全实践，必须使用 Amazon Identity and Access Management (IAM) 账户用于控制对 Amazon DocumentDB API 操作（特别是创建、修改或删除 Amazon DocumentDB 资源的操作）的访问。此类资源包括集群、安全组和参数组。您还必须使用 IAM 可控制执行常见管理任务的操作，例如备份还原集群。创建 IAM 角色时，采用最小权限原则。

- 使用 [基于角色的访问控制](#) (p. 156) 强制执行最低权限。
- 为每个管理 Amazon DocumentDB 资源的人员分配一个单独的 IAM 账户。请勿使用 Amazon Web Services 账户根用户管理 Amazon DocumentDB 资源。为每个人（包括您自己）创建一个 IAM 用户。
- 授予每位 IAM 用户执行其职责所需的最小权限集。
- 使用 IAM 组有效地管理适用于多个用户的权限。有关 IAM 的更多信息，请参阅 [IAM 用户指南](#)。有关 IAM 最佳实践的信息，请参阅 [IAM 最佳实践](#)。
- 定期轮换 IAM 凭证。

- 配置 Amazon Secrets Manager 可自动轮换 Amazon DocumentDB 的密钥。有关更多信息，请参阅 [轮换您的 Amazon Secrets Manager 和 Amazon DocumentDB B 的轮流秘密](#) 在 Amazon Secrets Manager 用户指南。
- 授予每位 Amazon DocumentDB 用户执行其职责所需的最小权限集。有关更多信息，请参阅 [基于角色的访问控制限制数据库访问](#) (p. 156)。
- 使用传输层安全性 (TLS) 加密传输中的数据并使用 Amazon KMS 对静态数据进行加密。

成本优化

以下最佳实践可帮助您管理和最大限度地降低使用 Amazon DocumentDB 时的成本。有关定价的信息，请参阅 [Amazon DocumentDB \(with MongoDB compat](#) 和 [Amazon DocumentDB \(with MongoDB compat](#)。

- 在阈值为该月预期账单的 50% 和 75% 时创建账单提醒。有关创建账单提醒的更多信息，请参阅 [创建账单提醒](#)。
- Amazon DocumentDB 的架构将存储和计算分开，因此即使是单实例集群也非常耐用。集群存储卷在三个可用区中从六个方向复制数据，提供极高的持久性，而不论集群中有多少个实例。典型的生产集群具有三个或更多实例，以提高可用性。但是，您在不需要高可用性时可使用单个实例开发集群，以优化成本。
- 对于开发和测试场景，在不再需要时停止集群，并在恢复开发时启动集群。有关更多信息，请参阅 [停止和启动 Amazon DocumentDB B 集群](#) (p. 265)。
- 写入、读取和删除数据时，TTL 和更改流都会产生 I/O。如果您已启用这些功能，但未在应用程序中使用它们，则禁用这些功能有助于降低开销。

使用指标确定性能问题

要确定资源不足和其他常见瓶颈导致的性能问题，您可以监控可用于 Amazon DocumentDB 集群的指标。

查看性能指标

定期监控性能指标，以查看各种时间范围内的平均值、最大值和最小值。这可帮助您确定性能下降的时间。您还可以将亚马逊设置为 CloudWatch 针对特定指标阈值发出警报，以便在达到这些阈值时向您发出警报。

要排除性能问题，了解系统的基准性能十分重要。设置新集群并让它在典型工作负载下运行之后，您应按一些不同的间隔（例如，1 小时、24 小时、1 周、2 周）来捕获所有性能指标的平均值、最大值和最小值。这将使您能够了解运行状况。这有助于将操作的峰值时间与非峰值时间进行比较。您随后可以利用这些信息确定性能何时降到标准水平以下。

您可以使用 Amazon Web Services Management Console 或 Amazon CLI 查看性能指标。有关更多信息，请参阅下列内容：

- 使用 [Amazon DocumentDB 管理控制台查看 CloudWatch 指标](#) (p. 382)
- 使用 [Amazon CLI](#) (p. 384)

设置 CloudWatch 警报

要设置 CloudWatch 警报，看 [使用 Amazon CloudWatch 警报](#) 在亚马逊 CloudWatch 用户指南。

评估性能指标

一个实例具有多个不同类别的指标。确定可接受的值的方式取决于指标。

CPU

- CPU 利用率— 使用的计算机处理容量的百分比。

内存

- 可用内存— 实例上可用的 RAM 量。
- 交换区使用情况— 实例使用的交换空间量 (以 MB 为单位) 。

输入/输出操作

- 读取 IOPS,写入 IOPS— 每秒平均磁盘读取或写入操作数。
- Read Latency (读取延迟),Write Latency (写入延迟)— 读取或写入操作的平均时间 (以毫秒为单位) 。
- Read Throughput (读取吞吐量),Write Throughput (写入吞吐量)— 每秒对磁盘读取或写入的平均兆字节数。
- Disk Queue Depth (磁盘队列深度)— 等待对磁盘写入或读取的 I/O 操作数。

网络流量

- 网络接收吞吐量,网络传输吞吐量— 每秒出入实例的网络流量速率 (以 MB 为单位) 。

数据库连接

- 数据库连接— 连接到实例的客户端会话数。

一般而言，性能指标的可接受值取决于您的基准性能以及应用程序执行的操作。应调查相对于基准性能的一致或趋势性变化。

以下是有关特定类型的指标的建议：

- 较高的 CPU 消耗— CPU 消耗值高可能是正常情况，只要它们符合您的应用程序目标 (如吞吐量或并发度) 并且符合预期。如果 CPU 消耗始终高于 80%，请考虑扩展您的实例。
- 较高的 RAM 消耗— 如果你的FreeableMemory指标经常降至实例总内存的 10% 以下，请考虑扩大您的实例。有关当您的 DocumentDB 实例遇到高内存压力时会发生什么情况的更多信息，请参阅[Amazon DocumentDB 资源管理 \(p. 11\)](#)。
- 交换区使用情况— 该指标应保持在或接近零。如果交换区使用情况非常大，请考虑扩展您的实例。
- 网络流量— 对于网络流量，应与系统管理员进行讨论，以了解域网络 and Internet 连接的预期吞吐量。如果吞吐量始终低于预期，则应调查网络流量。
- 数据库连接— 如果发现用户连接数较高，同时实例性能下降并且响应时间延长，请考虑约束数据库连接。实例的最佳用户连接数因您的实例类所执行操作的复杂性而异。对于与性能指标有关的问题，提高性能的首要手段之一是优化最常使用和成本最高昂的查询，以了解这是否会减少对系统资源的压力。

如果优化查询后问题仍然存在，请考虑将您的 Amazon DocumentDB 实例类升级为具有更多与所遇问题相关的资源 (CPU、RAM、磁盘空间、网络带宽，I/O 容量) 的实例类。

优化查询

提高集群性能的最佳方式之一是优化最常使用和占用最多资源的查询，以降低其运行成本。

您可以使用分析器 (请参阅 [分析 Amazon DocumentDB 操作 \(p. 388\)](#)) 来记录在您集群上执行的操作的执行时间和详细信息。对于监控集群上速度最慢的操作以帮助提高单个查询的性能和整体集群性能，分析器非常有用。

您还可以使用 `explain` 命令来了解如何分析特定查询的查询计划。您可以使用此信息修改查询或底层集合以提高查询性能（例如，添加索引）。

TTL 和时间序列工作负载

TTL 索引到期后的文档删除是一个工作量巨大的过程。无法保证在任何特定时间段内删除文档。有大量因素会影响 TTL 进程删除过期文档的时间，包括实例大小、实例资源利用率、文档大小、总吞吐量、索引数量以及索引和工作集是否位于内存中，等等。

在 TTL 监视器删除文档时，每次删除都会产生 IO 开销，而这将增加您的账单费用。如果吞吐量和 TTL 删除速率增加，I/O 使用量也会随之增加，从而导致您的账单费用增加。但是，如果您不创建 TTL 索引来删除文档，而是根据时间将文档分成集合，并在不再需要这些集合时将其删除，则不会产生任何 IO 成本。这可能比使用 TTL 索引更具成本效益。

对于时间序列工作负载，您可以考虑创建滚动集合而不是 TTL 索引，因为滚动集合可以更高效地删除数据，同时 I/O 开销也更低。如果您有大型集合（特别是超过 1TB 的集合）或者比较关注 TTL 删除的 I/O 开销，我们建议您根据时间将文档划分成多个集合，并在不再需要集合中的文档时将集合删除。您可以每天或每周创建一个集合，具体取决于您的数据接收速率。虽然要求会因应用程序而异，但一个很好的经验法则是划分数量较多的小型集合，而不是数量较少的大型集合。删除这些集合不会产生 IO 开销，并且比使用 TTL 索引速度更快，成本效益更高。

迁移

作为最佳实践，我们建议在将数据迁移到 Amazon DocumentDB B 时，先在 Amazon DocumentDB B 中创建索引，然后再迁移数据。首先创建索引可以减少总时间并提高迁移速度。为此，您可以使用 Amazon DocumentDB [索引工具](#)。有关迁移的更多信息，请参阅 [Amazon DocumentDB 迁移](#)。

我们还建议，在迁移生产数据库之前，最佳做法是在 Amazon DocumentDB 上全面测试您的应用程序，同时考虑功能、性能、操作和成本。

使用集群参数组

我们建议，在将集群参数组更改应用于生产集群前，您应在测试集群上试验这些更改。有关备份集群的信息，请参阅在 [Amazon DocumentDB 中备份和还原](#) (p. 194)。

聚合管道查询

创建具有多个阶段的聚合管道查询并且仅评估查询中的一个数据子集时，请将该 `$match` 阶段用作第一阶段或管道开头。首先使用 `$match` 将减少聚合管道查询中后续阶段需要处理的文档数量，从而提高查询性能。

batchInsert 和 batchUpdate

当执行高并发速率时 `batchInsert` 和/或 `batchUpdate` 操作数和金额 `FreeableMemory` (CloudWatch Metric) 在您的主实例上变为零，您可以减少批量插入或更新工作负载的并发性，或者，如果无法减少工作负载的并发性，则增加实例大小以增加 `FreeableMemory`。

功能差异：Amazon DocumentDB 和 MongoDB

以下是Amazon DocumentDB (与 MongoDB 兼容) 和 MMongoDB 之间的功能差异。

主题

- [Amazon DocumentDB B 的功能优势 \(p. 84\)](#)
- [更新的功能差异 \(p. 85\)](#)
- [与 MongoDB 之间的功能差异 \(p. 87\)](#)

Amazon DocumentDB B 的功能优势

隐式事务

在Amazon DocumentDB B 中，所有 CRUD 语句 (findAndModify,update,insert,delete) 保证原子性和一致性，即使是修改多个文档的操作也是如此。随着 Amazon DocumentDB 4.0 的推出，现在支持为多语句和多集合操作提供 ACID 属性的显式交易。有关在Amazon DocumentDB B 中使用交易的更多信息，请参阅[事务 \(p. 49\)](#)。

以下是 Amazon DocumentDB 中修改满足原子和一致行为的多个文档的操作示例。

```
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 }]
})
```

组成批量操作 (如 `updateMany` 和 `deleteMany`) 的各个操作是原子操作, 但整个批量操作不是原子操作。例如, 如果各个插入操作成功执行而未出现错误, 则整个 `insertMany` 操作是原子操作。如果 `insertMany` 操作遇到错误, `insertMany` 操作中每个单独的插入语句都将作为原子操作执行。如果你需要 ACID 特性 `insertMany`, `updateMany`, 以及 `deleteMany` 操作, 建议使用事务。

更新的功能差异

Amazon DocumentDB 继续改善与 MongoDB 的兼容性, 它与客户要求我们构建的功能背道而驰。本部分包含我们在 Amazon DocumentDB 中移除的功能差异, 以便我们的客户更轻松地迁移和构建应用程序。

主题

- [数组索引 \(p. 85\)](#)
- [多键索引 \(p. 86\)](#)
- [字符串中的 Null 字符 \(p. 86\)](#)
- [基于角色的访问控制 \(p. 86\)](#)
- [\\$regex 索引 \(p. 86\)](#)
- [嵌套文档的投影 \(p. 87\)](#)

数组索引

截至 2020 年 4 月 23 日, Amazon DocumentDB B 现在支持索引大于 2,048 字节的数组。数组中单个项目的限制仍为 2,048 字节, 这与 MongoDB 一致。

如果您正在创建新索引, 则无需操作即可利用改进的功能。如果您有现有索引, 则可以通过删除索引然后重新创建索引来利用改进的功能。具有改进功能的当前索引版本为 "v" : 3。

Note

对于生产集群, 删除索引可能会影响您的应用程序性能。我们建议您在更改生产系统时首先进行测试并谨慎行事。此外, 重新创建索引所需的时间将是集合的总体数据大小的函数。

您可以使用以下命令查询索引的版本。

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

此操作的输出将类似于下文。在此输出中, 索引的版本是 "v" : 3, 这是最新的索引版本。

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



```
}  
]
```

多键索引

自 2020 年 4 月 23 日起，Amazon DocumentDB B 现在支持使用同一数组中的多个键创建复合索引。

如果您正在创建新索引，则无需操作即可利用改进的功能。如果您有现有索引，则可以通过删除索引然后重新创建索引来利用改进的功能。具有改进功能的当前索引版本为 "v" : 3。

Note

对于生产集群，删除索引可能会影响您的应用程序性能。我们建议您在更改生产系统时首先进行测试并谨慎行事。此外，重新创建索引所需的时间将是集合的总体数据大小的函数。

您可以使用以下命令查询索引的版本。

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

此操作的输出将类似于下文。在此输出中，索引的版本是 "v" : 3，这是最新的索引版本。

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

字符串中的 Null 字符

截至 2020 年 6 月 22 日，Amazon DocumentDB B 现在支持空字符 ('\0') 以字符串表示。

基于角色的访问控制

Amazon DocumentDB 支持内置角色的访问控制 (RBAAC)。要了解更多信息，请参阅 [基于角色的访问控制 \(p. 156\)](#)。Amazon DocumentDB B 尚不支持 RBAC 的自定义角色。

\$regex 索引

截至 2020 年 6 月 22 日，Amazon DocumentDB B 现在支持 \$regex 运算符来使用索引。

要将索引与 \$regex 运算符一起使用，您必须使用 hint() 命令。使用 hint() 时，您必须指定将 \$regex 应用到的字段的名称。例如，如果您在字段 product 上有索引，且索引名称为 p_1，则 db.foo.find({product: /^x.*\$/}).hint({product:1}) 将使用 p_1 索引，但 db.foo.find({product: /^x.*\$/}).hint("p_1") 不使用该索引。您可以通过使用 explain() 命令或使用分析器记录慢速查询来验证是否选择了索引。例如，db.foo.find({product: /^x.*\$/}).hint("p_1").explain()。

Note

hint() 方法一次只能与一个索引一起使用。

为 `$regex` 查询使用索引的功能，针对使用前缀但未指定 `i`、`m` 或 `o` 正则表达式选项的正则表达式查询进行了优化。

将索引与 `$regex` 结合使用时，建议您在具有高选择性的字段上创建索引，这些字段的重复值数量低于集合中总文档数的 1% 的字段。例如，如果您的集合包含 100,000 个文档，则仅在相同值出现 1000 次或更少的字段上创建索引。

嵌套文档的投影

有功能上的区别 `$project` 版本 3.6 中 Amazon DocumentDB B 和 MongoDB 之间的运算符已在 Amazon DocumentDB 4.0 中解析，但在 Amazon DocumentDB 3.6 中仍不支持。

Amazon DocumentDB 3.6 在应用投影时仅考虑嵌套文档中的第一个字段，而 MongoDB 3.6 将解析子文档并将投影应用于每个子文档。

例如：如果投影是 `"a.b.c": 1`，那么这种行为在 Amazon DocumentDB 和 MongoDB 中都能按预期运行。但是，如果投影是 `{a:{b:{c:1}}}` 那么 Amazon DocumentDB B 3.6 只会将投影应用于 `a` 而不是 `b` 要么 `c`。在 Amazon DocumentDB B 4.0 中，投影 `{a:{b:{c:1}}}` 将应用于 `a`、`b`，以及 `c`。

与 MongoDB 之间的功能差异

主题

- [管理数据库和集合 \(p. 87\)](#)
- [cursor.maxTimeMS \(p. 87\)](#)
- [explain\(\) \(p. 88\)](#)
- [字段名称限制 \(p. 88\)](#)
- [索引构建 \(p. 88\)](#)
- [在路径中使用空键查找 \(p. 88\)](#)
- [MongoDB API、操作和数据类型 \(p. 89\)](#)
- [mongodump 和 mongorestore 实用工具 \(p. 89\)](#)
- [结果排序 \(p. 89\)](#)
- [可重试写入 \(p. 89\)](#)
- [稀疏索引 \(p. 89\)](#)
- [存储压缩 \(p. 90\)](#)
- [在 `\$all` 表达式中使用 `\$elemMatch` \(p. 90\)](#)
- [\\$ne, \\$nin, \\$nor, \\$not, \\$exists，以及 \\$elemMatch 索引 \(p. 90\)](#)
- [\\$lookup \(p. 90\)](#)

管理数据库和集合

Amazon DocumentDB B 不支持管理员或本地数据库，也不支持 `MongoDBsystem.*` 要么 `startup_log` 分别是集合。

cursor.maxTimeMS

Amazon DocumentDB `cursor.maxTimeMS` 重置每个计数器 `getMore` 请求。因此，如果 `3000MSmaxTimeMS` 已指定，查询需要 2800MS，每个后续查询 `getMore` 请求需要 300 毫秒，则游标不会超时。只有在单个操作（无论是查询还是个别操作）时，游标才会超时 `getMore` 请求，花费超过指定值 `maxTimeMS`。此外，检查游标执行时间的扫描器以一分钟的粒度运行。

explain()

Amazon DocumentDB 在专门构建的数据库引擎上模拟 MongoDB 4.0 API，该引擎使用分布式、容错、自我修复存储系统。因此，查询计划和输出 `explain()` Amazon DocumentDB B 和 MongoDB 之间可能有所不同。希望控制其查询计划的客户可以使用 `$hint` 运算符强制选择首选索引。

字段名称限制

Amazon DocumentDB B 不支持点号 “.”。例如，在文档字段名称中，`db.foo.insert({'x.1':1})`。

Amazon DocumentDB B 也不支持字段名中的 \$ 前缀。

例如，在 Amazon DocumentDB B 或 MongoDB 中尝试以下命令：

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

MongoDB 将返回以下内容：

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

Amazon DocumentDB B 将返回一个错误：

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

Note

这种功能差异有一个例外。以下以 \$ 前缀开头的字段名称已列入白名单，可以在 Amazon DocumentDB 中成功使用：`$id`、`$ref` 和 `$db`。

索引构建

Amazon DocumentDB 只允许在任何给定时间对一个集合进行一次索引构建。要么在前景中，要么在背景中。如果当前正在构建索引，在同一集合上发生了 `createIndex()` 或 `dropIndex()` 之类的操作，则新尝试的操作将失败。

默认情况下，Amazon DocumentDB B 和 MongoDB 版本 4.0 中的索引构建在后台进行。如果向 `createIndexes` 或其 shell 助手指定 MongoDB 版本 4.2 及更高版本会忽略后台索引构建选项 `createIndex()` 和 `createIndexes()`。

生存时间 (TTL) 索引在索引构建完成后开始使文档过期。

在路径中使用空键查找

当你用一个包含空字符串作为路径一部分的键来查找时（例如 `x.,x..b`），并且该对象有一个空字符串密钥路径（例如 `{"x" : [ { "" : 10 }, { "b" : 20 } ]}`）在数组中，Amazon DocumentDB 返回的结果与你在 MongoDB 中运行相同的查找结果不同。

在 MongoDB 中，当空字符串键不在路径查找的末尾时，在数组内查找空键路径会按预期工作。但是，当空字符串键位于路径查找的末尾时，它不会查看数组。

但是在 Amazon DocumentDB 中，只读取数组中的第一个元素，因为 `getArrayIndexFromKeyString` 将空字符串转换为 0，因此字符串键查找被视为向上查找数组索引。

MongoDB API、操作和数据类型

Amazon DocumentDB B 与 MongoDB 3.6 和 4.0 API 兼容。对于 up-to-date 支持的功能列表，请参阅[支持的 MongoDB API、操作和数据类型](#) (p. 93)。

mongodump和mongoexport实用工具

Amazon DocumentDB 不支持管理员数据库，因此在使用 `mongodump` 或 `mongoexport` 公用事业。当您使用 Amazon DocumentDB。mongoexport，除了恢复操作外，还需要重新创建用户角色。

结果排序

Amazon DocumentDB 不保证结果集的隐式结果排序顺序。要确保结果集的顺序，可使用 `sort()` 显式指定排序顺序。

以下示例根据库存字段按降序对清单集合中的项目排序。

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

在使用时 `$sort` 聚合阶段，排序顺序不会保留，除非 `$sort` 阶段是聚合管道中的最后一个阶段。在使用时 `$sort` 聚合阶段与 `$group` 聚合阶段，`$sort` 聚合阶段仅适用于 `$first` 和 `$last` 蓄能器。在 Amazon DocumentDB B 4.0 中，添加了对以下内容的支持 `$push` 遵守先前的排序顺序 `$sort` 阶段。

可重试写入

从 MongoDB 4.2 可兼容驱动程序开始，默认情况下可重试写入处于启用状态。但是，Amazon DocumentDB B 目前不支持可重试写入。该功能差异将显示在类似以下内容的错误消息中。

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

可以通过连接字符串禁用可重试写入 (例如 `MongoClient("mongodb://my.mongodb.cluster/db?retryWrites=false")`) 或者 `MongoClient` 构造函数的关键字参数 (例如 `MongoClient("mongodb://my.mongodb.cluster/db", retryWrites=False)`)。

下面是一个在连接字符串中禁用可重试写入的 Python 示例。

```
client =
    pymongo.MongoClient('mongodb://<username>:<password>@docdb-2019-03-17-16-49-12.cluster-
ccuszb3pn5e.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0', w='majority', j=True, retryWrites=False)
```

稀疏索引

要使用您在查询中创建的稀疏索引，必须在涵盖索引的字段中使用 `$exists` 子句。如果你省略了 `$exists`，Amazon DocumentDB B 不使用稀疏索引。

以下是示例。

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

对于稀疏的多键索引，如果查找文档生成了一组值而仅缺少一部分索引字段，则 Amazon DocumentDB 不支持唯一键约束。例如，如果输入为 "a" : [{ "b" : 2 }, { "c" : 1 }]，则 `createIndex({ "a.b" : 1 }, { unique : true, sparse : true })` 不受支持，因为 "a.c" 存储在索引中。

存储压缩

Amazon DocumentDB B 目前不支持对存储的数据或索引进行压缩。存储的数据和索引的数据大小可能比您使用其他选项时更大。

在 \$all 表达式中使用 \$elemMatch

Amazon DocumentDB B 目前不支持使用 `$elemMatch` 运算符在 `$all` 表情。一种解决方法是，您可以将 `$and` 运算符与 `$elemMatch` 结合使用，如下所示。

原始运算：

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

更新后的运算：

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

\$ne, \$nin, \$nor, \$not, \$exists，以及 \$elemMatch 索引

Amazon DocumentDB B 目前不支持将索引与 `$ne`, `$nin`, `$nor`, `$not`, `$exists`, `$distinct`，以及 `$elemMatch` 运算符。因此，使用这些运算符将导致集合扫描。通过在使用以上运算符之一之前执行筛选或匹配，将减少需要扫描的数据量，从而提高性能。

\$lookup

Amazon DocumentDB 支持进行相等匹配（例如，左外连接）的功能，但不支持不相关的子查询。

使用索引 \$lookup

现在您可以使用索引了 `$lookup` 阶段操作符。根据您的用例，您可以使用多种索引算法来优化性能。本节将解释不同的索引算法 `$lookup` 并帮助您为自己的工作量选择最佳方案。

默认情况下，Amazon DocumentDB 将在以下情况下使用哈希算法`allowDiskUse:false`使用并在何时进行排序合并`allowDiskUse:true`已使用。对于某些用例，可能需要强制查询优化器使用不同的算法。以下是不同的索引算法`$lookup`聚合运算符可以利用：

- 嵌套循环: A nested loop plan is typically beneficial for a workload if the foreign collection is <1 GB and the field in the foreign collection has an index. If the nested loop algorithm is being used, the explain plan will show the stage as `NESTED_LOOP_LOOKUP`.
- 排序合并: A sort merge plan is typically beneficial for a workload if the foreign collection does not have an index on the field used in lookup and the working dataset doesn't fit in memory. If the sort merge algorithm is being used, the explain plan will show the stage as `SORT_LOOKUP`.
- 哈希: A hash plan is typically beneficial for a workload if the foreign collection is < 1GB and the working dataset fits in memory. If the hash algorithm is being used, the explain plan will show the stage as `##_LOOKUP`.

您可以识别正在使用的索引算法`$lookup`运算符在查询中使用 `explain`。下面是一个示例。

```
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
}
```

作为使用的一种替代方法`explain()`方法，你可以使用分析器来查看你在使用`$lookup`运算符。有关分析器的更多信息，请参阅[分析 Amazon DocumentDB 操作 \(p. 388\)](#)。

使用 `planHint`

如果你想强制查询优化器使用不同的索引算法`$lookup`，您可以使用`planHint`。为此，请使用聚合阶段选项中的注释来强制执行不同的计划。下面是注释语法示例：

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

下面是使用`planHint`强制查询优化器使用`HASH`索引算法：

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

要测试哪种算法最适合您的工作负载，可以使用`executionStats`的参数`explain`测量执行时间的方法`$lookup`在修改索引算法时暂停（即`HASH/SORT/NESTED_LOOP`）。

下面的示例演示如何使用`executionStats`来测量`$lookup`阶段，使用`SORT`算法。

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

支持的 MongoDB API、操作和数据类型

Amazon DocumentDB (与 MongoDB 兼容) 是一项快速、可扩展、高度可用且完全托管的文档数据库服务，它支持 MongoDB 工作负载。Amazon DocumentDB 与 MongoDB 3.6 和 4.0 API 兼容。本部分列出了支持的功能。有关使用 MongoDB API 和驱动程序的支持，请查阅 MongoDB 社区论坛。如需使用 Amazon DocumentDB 服务的支持，请联系相应的 Amazon 支持团队。有关 Amazon DocumentDB 和 MongoDB 之间的功能差异，请参阅[功能差异：Amazon DocumentDB 和 MongoDB \(p. 84\)](#)。

仅供内部使用或不适用于完全托管的服务的 MongoDB 命令和运算符不受支持，并且未包含在支持的功能列表中。

自发布以来，我们已添加 50 多种附加功能，并将继续向客户学习以提供他们所需的功能。有关最近发布的信息，请参阅[Amazon DocumentDB](#)。

如果您希望我们构建一项不受支持的功能，请向我们发送一封包含您的 accountID、请求的功能和使用案例的电子邮件来告知我们此情况[Amazon DocumentDB 服务](#)。

主题

- [数据库命令 \(p. 93\)](#)
- [查询和投影运算符 \(p. 97\)](#)
- [更新运算符 \(p. 99\)](#)
- [地理空间 \(p. 101\)](#)
- [游标方法 \(p. 101\)](#)
- [聚合管道运算符 \(p. 102\)](#)
- [数据类型 \(p. 109\)](#)
- [索引和索引属性 \(p. 110\)](#)

数据库命令

主题

- [管理命令 \(p. 93\)](#)
- [聚合 \(p. 94\)](#)
- [身份验证 \(p. 95\)](#)
- [诊断命令 \(p. 95\)](#)
- [查询和写入操作 \(p. 95\)](#)
- [角色管理命令 \(p. 96\)](#)
- [会话命令 \(p. 96\)](#)
- [用户管理 \(p. 97\)](#)

管理命令

命令	3.6	4.0
受限集合	否	否

命令	3.6	4.0
cloneCollectionAs上限	否	否
collMod	部分	部分
collMod: expireAfterSeconds	是	是
convertToCapped	否	否
copydb	否	否
create	是	是
createView	否	否
createIndexes	是	是
currentOp	是	是
drop	是	是
dropDatabase	是	是
dropIndexes	是	是
filemd5	否	否
killCursors	是	是
killOp	是	是
listCollections	是	是
listDatabases	是	是
listIndexes	是	是
reIndex	否	否
renameCollection	是	是

聚合

命令	3.6	4.0
aggregate	是	是
count	是	是
区分	是	是
mapReduce	否	否

身份验证

命令	3.6	4.0
authenticate	是	是
logout	是	是

诊断命令

命令	3.6	4.0
buildInfo	是	是
collStats	是	是
connPoolStats	否	否
connectionStatus	是	是
dataSize	是	是
dbHash	否	否
dbStats	是	是
explain	是	是
executionStats	是	是
features	否	否
hostInfo	是	是
listCommands	是	是
profiler	是	是
serverStatus	是	是
top	是	是

查询和写入操作

命令	3.6	4.0
delete	是	是
find	是	是
findAndModify	是	是
getLastError	否	否
getMore	是	是

命令	3.6	4.0
getPrevError	否	否
insert	是	是
parallelCollectionScan	否	否
resetError	否	否
update	是	是
Change streams	是	是
GridFS	否	否

角色管理命令

命令	3.6	4.0
createRole	是	是
dropRole	是	是
dropAllRolesFromDatabase	否	否
grantRolesTo角色	是	是
revokePrivilegesFrom角色	是	是
revokeRolesFrom角色	是	是
updateRole	是	是
rolesInfo	否	否

会话命令

命令	3.6	4.0
中止事务	否	是
commitTransaction	否	是
结束会话	否	是
killAllSessions	否	是
killAllSessionsByPattern	否	否
杀死会话	否	是
刷新会话	否	否
Start	否	是

用户管理

命令	3.6	4.0
createUser	是	是
dropAllUsersFromDatabase	是	是
dropUser	是	是
grantRolesTo用户	是	是
revokeRolesFrom用户	是	是
updateUser	是	是
userInfo	是	是

查询和投影运算符

主题

- [数组运算符 \(p. 97\)](#)
- [按位运算符 \(p. 97\)](#)
- [注释运算符 \(p. 98\)](#)
- [比较运算符 \(p. 98\)](#)
- [元素运算符 \(p. 98\)](#)
- [评估查询运算符 \(p. 98\)](#)
- [逻辑运算符 \(p. 99\)](#)
- [投影运算符 \(p. 99\)](#)

数组运算符

命令	3.6	4.0
\$all	是	是
\$elemMatch	是	是
\$size	是	是

按位运算符

命令	3.6	4.0
\$bitsAllSet	是	是
\$bitsAnySet	是	是

命令	3.6	4.0
\$bitsAllClear	是	是
\$bitsAnyClear	是	是

注释运算符

命令	3.6	4.0
\$comment	是	是

比较运算符

命令	3.6	4.0
\$eq	是	是
\$gt	是	是
\$gte	是	是
\$lt	是	是
\$lte	是	是
\$ne	是	是
\$in	是	是
\$nin	是	是

元素运算符

命令	3.6	4.0
\$exists	是	是
\$type	是	是

评估查询运算符

命令	3.6	4.0
\$expr	否	否
\$jsonSchema	否	否
\$mod	是	是

命令	3.6	4.0
\$regex	是	是
\$text	否	否
\$where	否	否

逻辑运算符

命令	3.6	4.0
\$or	是	是
\$and	是	是
\$not	是	是
\$nor	是	是

投影运算符

命令	3.6	4.0
\$	是	是
\$elemMatch	是	是
\$meta	否	否
\$slice	是	是

更新运算符

主题

- [数组运算符 \(p. 99\)](#)
- [按位运算符 \(p. 100\)](#)
- [字段操作员 \(p. 100\)](#)
- [更新修改器 \(p. 100\)](#)

数组运算符

命令	3.6	4.0
\$	是	是
\$[]	是	是

命令	3.6	4.0
\$[<identifier>]	是	是
\$addToSet	是	是
\$pop	是	是
\$pullAll	是	是
\$pull	是	是
\$push	是	是

按位运算符

命令	3.6	4.0
\$bit	是	是

字段操作员

操作符	3.6	4.0
\$inc	是	是
\$mul	是	是
\$rename	是	是
\$setOnInsert	是	是
\$set	是	是
\$unset	是	是
\$min	是	是
\$max	是	是
\$currentDate	是	是

更新修改器

操作符	3.6	4.0
\$each	是	是
\$slice	是	是
\$sort	是	是
\$position	是	是

地理空间

几何体说明符

查询选择器	3.6	4.0
\$box	否	否
\$center	否	否
\$centerSphere	否	否
\$NearSphere	是	是
\$几何体	是	是
\$MaxDistan	是	是
\$minDistan	是	是
\$polyg	否	否
\$uniqueDocs	否	否

查询选择器

命令	3.6	4.0
\$geoIntersects	是	是
\$geoWithin	是	是
\$near	否	否
\$NearSphere	是	是
\$polyg	否	否
\$uniqueDocs	否	否

游标方法

命令	3.6	4.0
cursor.batchSize()	是	是
cursor.close()	是	是
cursor.isClosed()	是	是
cursor.collation()	否	否
cursor.comment()	是	是

命令	3.6	4.0
cursor.count()	是	是
cursor.explain()	是	是
cursor.forEach()	是	是
cursor.hasNext()	是	是
cursor.hint()	是	是
cursor.isExhausted()	是	是
cursor.itcount()	是	是
cursor.limit()	是	是
cursor.map()	是	是
cursor.maxScan()	是	是
cursor.maxTimeMS()	是	是
cursor.max()	否	否
cursor.min()	否	否
cursor.next()	是	是
光标。noCursorTimeout()	否	否
光标。objsLeftInBatch ()	是	是
cursor.pretty()	是	是
cursor.readConcern()	是	是
cursor.readPref()	是	是
cursor.returnKey()	否	否
光标。showRecordId()	否	否
cursor.size()	是	是
cursor.skip()	是	是
cursor.sort()	是	是
cursor.tailable()	否	否
cursor.toArray()	是	是

聚合管道运算符

主题

- [累加器表达式 \(p. 103\)](#)
- [算术运算符 \(p. 103\)](#)
- [数组运算符 \(p. 104\)](#)

- [布尔运算符 \(p. 105\)](#)
- [比较运算符 \(p. 105\)](#)
- [条件表达式运算符 \(p. 105\)](#)
- [数据类型运算符 \(p. 105\)](#)
- [日期运算符 \(p. 105\)](#)
- [文字运算符 \(p. 106\)](#)
- [合并运算符 \(p. 106\)](#)
- [自然运算符 \(p. 106\)](#)
- [集合运算符 \(p. 107\)](#)
- [阶段操作符 \(p. 107\)](#)
- [字符串运算符 \(p. 108\)](#)
- [系统变量 \(p. 108\)](#)
- [文本搜索运算符 \(p. 109\)](#)
- [类型转换运算符 \(p. 109\)](#)
- [变量运算符 \(p. 109\)](#)

累加器表达式

表达式	3.6	4.0
\$sum	是	是
\$avg	是	是
\$first	是	是
\$last	是	是
\$max	是	是
\$min	是	是
\$push	是	是
\$addToSet	是	是
\$stdDevPop	否	否
\$stdDevSamp	否	否

算术运算符

命令	3.6	4.0
\$abs	是	是
\$add	是	是
\$ceil	否	否
\$divide	是	是

命令	3.6	4.0
\$exp	否	否
\$floor	否	否
\$ln	否	否
\$log	否	否
\$log10	否	否
\$mod	是	是
\$multiply	是	是
\$pow	否	否
\$sqrt	否	否
\$subtract	是	是
\$trunc	否	否

数组运算符

命令	3.6	4.0
\$arrayElemAt	是	是
\$arrayToObject	是	是
\$concatArrays	是	是
\$filter	是	是
\$indexOfArray	是	是
\$isArray	是	是
\$objectToArray	是	是
\$range	是	是
\$reverseArray	是	是
\$reduce	是	是
\$size	是	是
\$slice	是	是
\$zip	是	是
\$in	是	是

布尔运算符

命令	3.6	4.0
\$and	是	是
\$or	是	是
\$not	是	是

比较运算符

命令	3.6	4.0
\$cmp	是	是
\$eq	是	是
\$gt	是	是
\$gte	是	是
\$lt	是	是
\$lte	是	是
\$ne	是	是

条件表达式运算符

命令	3.6	4.0
\$cond	是	是
\$ifNull	是	是
\$switch	否	否

数据类型运算符

命令	3.6	4.0
\$type	是	是

日期运算符

命令	3.6	4.0
\$dayOfYear	是	是

命令	3.6	4.0
\$dayOfMonth	是	是
\$dayOfWeek	是	是
\$year	是	是
\$month	是	是
\$week	是	是
\$hour	是	是
\$minute	是	是
\$second	是	是
\$millisecond	是	是
\$dateToString	是	是
\$isoDayOfWeek	是	是
\$isoWeek	是	是
\$dateFromParts	否	否
\$dateToParts	否	否
\$dateFromString	是	是
\$isoWeekYear	是	是

文字运算符

命令	3.6	4.0
\$literal	是	是

合并运算符

命令	3.6	4.0
\$mergeObjects	是	是

自然运算符

命令	3.6	4.0
\$natural	是	是

集合运算符

命令	3.6	4.0
\$setEquals	是	是
\$setIntersection	是	是
\$setUnion	是	是
\$setDifference	否	否
\$setIsSubset	是	是
\$anyElementTrue	否	否
\$allElementsTrue	否	否

阶段操作符

命令	3.6	4.0
\$collStats	否	否
\$project	是	是
\$match	是	是
\$redact	是	是
\$limit	是	是
\$skip	是	是
\$unwind	是	是
\$group	是	是
\$sample	是	是
\$sort	是	是
\$geoNear	是	是
\$lookup	是	是
\$out	是	是
\$indexStats	是	是
\$facet	否	否
\$bucket	否	否
\$bucketAuto	否	否
\$sortByCount	否	否
\$addFields	是	是

命令	3.6	4.0
\$replaceRoot	是	是
\$count	是	是
\$currentOp	是	是
\$listLocalSessions	否	否
\$listSessions	否	否
\$graphLookup	否	否

字符串运算符

命令	3.6	4.0
\$concat	是	是
\$indexOfBytes	是	是
\$indexOfCP	是	是
\$ltrim	否	否
\$rtrim	否	否
\$split	是	是
\$strcasecmp	是	是
\$strLenBytes	是	是
\$strLenCP	是	是
\$substr	是	是
\$substrBytes	是	是
\$substrCP	是	是
\$toLower	是	是
\$toUpper	是	是
\$trim	否	否

系统变量

命令	3.6	4.0
\$\$CURRENT	否	否
\$\$DESCEND	是	是
\$\$KEEP	是	是

命令	3.6	4.0
\$\$PRUNE	是	是
\$\$REMOVE	否	否
\$\$ROOT	是	是

文本搜索运算符

命令	3.6	4.0
\$meta	否	否

类型转换运算符

命令	3.6	4.0
\$转换	否	否
\$tobool	否	否
\$toDate	否	否
\$toDecimal	否	否
\$toDouble	否	否
\$toint	否	否
\$toLong	否	否
\$toObjectId	否	否
\$toString	否	否

变量运算符

命令	3.6	4.0
\$map	是	是
\$let	否	否

数据类型

命令	3.6	4.0
Double	是	是

命令	3.6	4.0
字符串	是	是
对象	是	是
数组	是	是
二进制数据	是	是
ObjectId	是	是
布尔值	是	是
日期	是	是
Null	是	是
32位整数 (int)	是	是
时间戳	是	是
64 位整数 (长)	是	是
MinKey	是	是
MaxKey	是	是
Decimal128	是	是
正则表达式	是	是
JavaScript	否	否
JavaScript (有范围)	否	否
未定义	否	否
符号	否	否
DBPointer	否	

索引和索引属性

主题

- [索引 \(p. 110\)](#)
- [索引属性 \(p. 111\)](#)

索引

命令	3.6	4.0
单个字段索引	是	是
复合索引	是	是
多键索引	是	是

命令	3.6	4.0
文本索引	否	否
2dsphere	是	是
2d 索引	否	否
哈希索引	否	否

索引属性

命令	3.6	4.0
TTL	是	是
唯一	是	是
部分	否	否
区分大小写	否	否
稀疏	是	是
背景	是	是

迁Amazon DocumentDB

Amazon DocumentDB (兼容 MongoDB) 是一项完全托管的与 MongoDB API 兼容的完全托管的数据库服务。您可以使用本部分中详述的过程将数据从在本地或亚马逊弹性计算云 (Amazon EC2) 上运行的 MongoDB 数据库迁移到 Amazon DocumentDB。

主题

- [使用将您的 Amazon DocumentDB 集群从 3.6 升级到 4.0Amazon Database Migration Service \(p. 112\)](#)
- [迁移工具 \(p. 119\)](#)
- [发现 \(p. 120\)](#)
- [规划：Amazon DocumentDB 集群要求 \(p. 122\)](#)
- [迁移方法 \(p. 124\)](#)
- [迁移源 \(p. 127\)](#)
- [迁移连接 \(p. 128\)](#)
- [测试 \(p. 130\)](#)
- [性能测试 \(p. 131\)](#)
- [故障转移测试 \(p. 132\)](#)
- [其他资源 \(p. 132\)](#)

使用将您的 Amazon DocumentDB 集群从 3.6 升级到 4.0Amazon Database Migration Service

Important

Amazon DocumentDB 不遵循与 MongoDB 和 MongoDB 相同的支持生命周期 end-of-life 时间表不适用于 Amazon DocumentDB。目前没有计划 end-of-life 适用于 Amazon DocumentDB 3.6 以及你现有的 MongoDB 3.6 驱动程序、应用程序和工具，将继续与 Amazon DocumentDB 配合使用。

您可以使用将您的 Amazon DocumentDB 3.6 集群升级到 4.0，停机时间最短Amazon DMS.Amazon DMS是一项完全托管的服务，可将关系数据库和非关系数据库轻松迁移到 Amazon DocumentDB。

主题

- [第 1 步：启用变更流 \(p. 113\)](#)
- [第 2 步：修改更改流保留期限 \(p. 113\)](#)
- [第 3 步：迁移Migration \(p. 113\)](#)
- [第 4 步：创建Amazon DMS复制实例 \(p. 114\)](#)
- [第 5 步：创建Amazon DMS源终端节点 \(p. 115\)](#)
- [第 6 步：创建Amazon DMS目标终端节点 \(p. 117\)](#)
- [步骤 7：创建和运行迁移任务 \(p. 118\)](#)
- [步骤 8：将应用程序终端节点更改为 Amazon DocumentDB 群集 4.0 \(p. 119\)](#)

第 1 步：启用变更流

要执行停机时间最短的迁移，Amazon DMS 需要访问集群的变更流。[Amazon DocumentDB 变更流](#) 提供按时间顺序排列的更新事件，这些事件在您的集群集合和数据库内发生。从变更流中读取启用 Amazon DMS 执行更改数据捕获 (CDC) 并将增量更新应用到目标 Amazon DocumentDB 集群。

要为特定数据库上的所有集合启用更改流，请使用 mongo shell 向您的 Amazon DocumentDB 集群进行身份验证，然后执行以下命令：

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

第 2 步：修改更改流保留期限

接下来，根据您希望在变更流中保留变更事件的时间来修改更改流保留期。例如，如果您期望您的 Amazon DMS 从 Amazon DocumentDB v3.6 迁移到 v4.0 要花费 12 个小时，您应将更改流保留时间设置为大于 12 小时的值。Amazon DocumentDB 集群的默认保留期为三小时。您可以使用 Amazon DocumentDB 集群的变更流日志的保留时间修改为 1 小时到 7 天之间的任何时间 Amazon Web Services Management Console 或者 Amazon CLI。有关详细信息，请参考[修改更改流日志保留期限](#)。

第 3 步：迁移 Migration

在 Amazon DocumentDB 4.0 集群上创建的索引与在 Amazon DocumentDB 3.6 集群上创建的索引相同。虽然 Amazon DMS 处理数据的迁移，它不迁移索引。要迁移索引，请使用 Amazon DocumentDB 索引工具从 Amazon DocumentDB 3.6 集群中导出索引。您可以通过创建 Amazon DocumentDB 工具的克隆来获得该工具 GitHub repo，按照其中的说明操作[README.md](#)。您可以从 Amazon EC2 实例或 Amazon Cloud9 环境与您的 Amazon DBC 集群在同一 Amazon VPC 中运行。

以下代码从您的 Amazon DocumentDB v3.6 集群中转储索引：

```
python migrationtools/documentdb_index_tool.py --dump-indexes
--dir ~/index.js/
--host docdb-36-xx.cluster-xxxxxxx.us-west-2.docdb.amazonaws.com:27017
--tls --tls-ca-file ~/rds-ca-2019-root.pem
--username user
--password <password>

2020-02-11 21:51:23,245: Successfully authenticated to database: admin
2020-02-11 21:46:50,432: Successfully connected to instance docdb-36-xx.cluster-
xxxxxxx.us-west-2.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/
```

成功导出索引后，将这些索引还原到您的 Amazon DocumentDB 4.0 集群中。要恢复您在上一步中导出的索引，请使用 Amazon DocumentDB 索引工具。以下命令会从指定目录恢复 Amazon DocumentDB 4.0 集群中的索引。

```
python migrationtools/documentdb_index_tool.py --restore-indexes
--dir ~/index.js/
--host docdb-40-xx.cluster-xxxxxxx.us-west-2.docdb.amazonaws.com:27017
--tls --tls-ca-file ~/rds-ca-2019-root.pem
--username user
```

```
--password <password>

2020-02-11 21:51:23,245: Successfully authenticated to database: admin
2020-02-11 21:51:23,245: Successfully connected to instance docdb-40-xx.cluster-
xxxxxxxx.us-west-2.docdb.amazonaws.com:27017
2020-02-11 21:51:23,264: testdb.coll: added index: _id
```

要确认是否正确还原了索引，请使用 mongo shell 连接到您的 Amazon DocumentDB 4.0 集群，然后列出给定集合的索引。请看下面的代码：

```
mongo --ssl
--host docdb-40-xx.cluster-xxxxxxxx.us-west-2.docdb.amazonaws.com:27017
--sslCAFile rds-ca-2019-root.pem --username documentdb --password documentdb

db.coll.getIndexes()
```

第 4 步：创建 Amazon DMS 复制实例

网络 ACL 和安全组都允许（因此可到达您的实例）的发起 ping 的 Amazon DMS 复制实例连接并从您的源（在本例中为 Amazon DocumentDB 3.6 集群）读取数据，然后将其写入目标（Amazon DocumentDB 4.0 集群）。这些区域有：Amazon DMS 复制实例可以执行批量加载和 CDC 操作。大部分这种处理发生在内存中。但是，大型操作可能需要部分缓冲到磁盘上。缓存事务和日志文件也会写入磁盘。迁移数据后，复制实例还会流式传输所有更改事件，以确保源和目标保持同步。

创建 Amazon DMS 复制实例：

1. 打开 Amazon DMS [控制台](#)。
2. 在导航窗格中，选择复制实例。
3. 选择 Create replication instance (创建复制实例)，并输入以下信息：
 - 对于 Name (名称)，输入您选择的名称。例如，docdb36todocdb40。
 - 适用于说明，输入您选择的描述。对于 listItem，Amazon DocumentDB 3.6 到 Amazon DocumentDB 4.0
 - 适用于实例类，根据您的需求选择尺寸。
 - 适用于引擎版本，选择 3.4.1。
 - 适用于 Amazon VPC，选择容纳您的 Amazon DocumentDB 3.6 和 4.0 集群的 Amazon VPC。
 - 适用于分配的存储空间 (GiB)，请使用默认值 50 GiB。如果您的工作负载写入吞吐量较高，请增加此值以匹配您的工作负载。
 - 适用于多可用区，选择是如果您需要高可用性和故障转移支持。
 - 对于 Publicly accessible (公开访问)，请启用此选项。

Replication instance configuration

Name
The name must be unique among all of your replication instances in the current region.

docdb36todocdb40

Replication instance name must not start with a numeric value

Description

Amazon DocumentDB v3.6 to Amazon DocumentDB v4.0 replicatio

The description must only have unicode letters, digits, whitespace, or one of these symbols: _-/+=@. 1000 maximum character.

Instance class [Info](#)
Choose an appropriate instance class for your replication needs. Each instance class provides differing levels of compute, network and memory capacity. [DMS pricing](#)

dms.c4.xlarge
16 vCPUs 30 GiB Memory

☒ Include previous-generation instance classes

Engine version
Choose an DMS version to run on your replication instance. [DMS versions](#)

3.4.1 (Beta)

☒ Include Beta DMS versions

Allocated storage (GiB)
Choose the amount of storage space you want for your replication instance. DMS uses this storage for log files and cached transactions while replication tasks are in progress.

50

VPC
Choose an Amazon Virtual Private Cloud (VPC) where your replication instance should run.

vpc-2bf12540

☒ **Multi AZ**
If you choose this option, DMS will perform a multi-AZ deployment, with a primary instance in one availability zone (AZ) and a standby instance in another AZ. This configuration provides a highly available, fault-tolerant replication environment. Billing is based on [DMS pricing](#)

☒ **Publicly accessible**
If you choose this option, DMS will assign a public IP address to your replication instance, and you'll be able to connect to databases outside of your Amazon VPC.

4. 选择创建复制实例。

第 5 步：创建Amazon DMS源终端节点

源终端节点用于您希望升级到 4.0 的 Amazon DocumentDB 3.6 集群。

创建源终端节点

1. 打开 Amazon DMS [控制台](#)。
2. 在导航窗格中，选择 Endpoints (端点)。
3. 选择 Create endpoint，然后输入以下信息：
 - 对于 Endpoint type (终端节点类型)，请选择 Source (源)。
 - >对于终端节点标识符，请输入容易记住的名称，例如docdb-source。
 - 适用于源引擎，选择docdb。
 - 适用于服务器名称，输入您的 Amazon DocumentDB v3.6 集群的 DNS 名称。

- 适用于端口，输入您的 Amazon DocumentDB v3.6 集群的端口号。
- 适用于SSL 模式，选择verify-full。
- 适用于CA 证书，选择添加新 CA 证书。下载 [新 CA 证书](#) 以创建 TLS 连接捆绑包。适用于证书标识符，输入 rds-combined-ca-cn-bundle。对于 Import certificate file (导入证书文件)，选择 Choose file (选择文件)，然后导航到之前下载的 .pem 文件。选择并打开此文件。选择导入证书，然后选择rds-combined-ca-bundle来自 的选择证书下拉
- 适用于用户名中，输入您的 Amazon DocumentDB v3.6 集群的主用户名。
- 适用于密码，输入您的 Amazon DocumentDB v3.6 集群的主密码。
- 适用于数据库名称中，输入要升级的数据库名称。

Endpoint configuration

Endpoint identifier [Info](#)
A label for the endpoint to help you identify it.

Source engine
The type of database engine this endpoint is connected to.

Server name

Port
The port the database runs on for this endpoint.

Secure Socket Layer (SSL) mode
The type of Secure Socket Layer enforcement

CA certificate
 [Add new CA certificate](#)

User name [Info](#)

Password [Info](#)

Database name

4. 测试您的连接以验证其是否成功设置。

▼ Test endpoint connection (optional)

VPC

Replication instance
A replication instance performs the database migration

[Run test](#)

Endpoint identifier	Replication instance	Status	Message
docdb36-source	docdb36todocdb40	successful	

5. 选择 Create Endpoint (创建端点)。

Note

Amazon DMS 一次只能迁一个数据库。

第 6 步：创建 Amazon DMS 目标终端节点

目标终端节点适用于您的 Amazon DocumentDB 4.0 集群。

创建目标终端节点：

1. 打开 [Amazon DMS 控制台](#)。
2. 在导航窗格中，选择 Endpoints (端点)。
3. 选择 Create endpoint (创建终端节点)，然后输入以下信息：
 - 对于 Endpoint type (终端节点类型)，请选择 Target (目标)。
 - 对于 Endpoint identifier (终端节点标识符)，请输入容易记住的名称，例如 docdb-target。
 - 适用于源引擎，选择 docdb。
 - 适用于服务器名称，输入您的 Amazon DocumentDB v4.0 集群的 DNS 名称。
 - 适用于端口，输入您的 Amazon DocumentDB v4.0 集群的端口号。
 - 适用于 SSL 模式，选择 verify-full。
 - 适用于 CA 证书，选择现有的 rds-combined-ca-cn-bundle 来自的证书选择证书下拉。
 - 适用于用户名中，输入您的 Amazon DocumentDB v4.0 集群的主用户名。
 - 适用于密码，输入您的 Amazon DocumentDB v4.0 集群的主密码。
 - 适用于数据库名称中，输入用于设置源终端节点的相同数据库名称。

Endpoint configuration

Endpoint identifier [Info](#)
A label for the endpoint to help you identify it.

Target engine
The type of database engine this endpoint is connected to.

Server name

Port
The port the database runs on for this endpoint.

Secure Socket Layer (SSL) mode
The type of Secure Socket Layer enforcement

CA certificate
 [Add new CA certificate](#)

User name [Info](#)

Password [Info](#)

Database name

4. 测试您的连接以验证其已成功设置。

▼ Test endpoint connection (optional)

VPC

vpc-2bf12540

Replication instance

A replication instance performs the database migration

docdb36todocdb40

Run test

Endpoint identifier	Replication instance	Status	Message
docdb36-target	docdb36todocdb40	successful	

5. 选择 Create Endpoint (创建端点)。

步骤 7：创建和运行迁移任务

网络 ACL 和安全组都允许 (因此可到达您的实例) 的发起 ping 的 Amazon DMS 任务将复制实例与源实例和目标实例绑定。在创建迁移任务时，您可以指定源终端节点、目标终端节点、复制实例以及任何所需的迁移设置。网络 ACL 和安全组都允许 (因此可到达您的实例) 的发起 ping 的 Amazon DMS 可以创建具有三种不同迁移类型的任务：迁移现有数据、迁移现有数据并复制持续更改或仅复制数据更改。由于本演练的目的是在停机时间最短的情况下将 Amazon DocumentDB 3.6 集群升级到 Amazon DocumentDB 4.0，因此这些步骤利用该选项迁移现有数据和复制正在进行的更改。利用该选项，Amazon DMS 可在迁移现有数据时捕获更改。甚至在加载批量数据后，Amazon DMS 也会继续捕获和应用更改。最终，源数据库和目标数据库将保持同步，从而实现停机时间最少的迁移。

以下是创建迁移任务以实现最短停机时间迁移的步骤：

1. 打开 Amazon DMS [控制台](#)。
2. 在导航窗格中，选择任务。
3. 选择 Create task (创建任务)，然后输入以下信息：
 - 对于 Task name (任务名称)，请输入容易记住的名称，例如 my-dms-upgrade-task。
 - 适用于复制实例，请选择您在中创建的复制实例 [第 3 步：创建 Amazon Database Migration Service 复制实例](#)
 - 适用于源终端节点，请选择您在中创建的源终端节点 [步骤 4: 创建 Amazon Database Migration Service 源终端节点](#)
 - 适用于目标终端节点，请选择您在中创建的目标终端节点 [第 5 步：创建 Amazon Database Migration Service 目标终端节点](#)
 - 适用于迁移类型，选择迁移现有数据并复制持续更改。

Task configuration

Task identifier
my-dms-upgrade-task

Replication instance
docdb36todocdb40 - vpc-b06365ca

Source database endpoint
docdb36-source

Target database endpoint
docdb40-target

Migration type [Info](#)
Migrate existing data and replicate ongoing changes

4. 在任务设置部分，启用CloudWatch 圆木。
5. 适用于表映射部分中，将所有内容保留为原定设置。这将确保迁移数据库中的所有集合。
6. 适用于迁移任务启动配置，选择 A 创建时自动执行。这将在您创建迁移任务后自动启动迁移任务。
7. 选择创建任务。

Amazon DMS 现在开始将数据从 Amazon DocumentDB 3.6 集群迁移到您的 Amazon DocumentDB 4.0 集群。任务状态应从 Starting (正在启动) 更改为 Running (正在运行)。您可以通过在 Amazon DMS 控制台。几分钟/小时 (取决于迁移的大小)，状态应从更改为加载完成，复制正在进行。这意味着 Amazon DMS 已完成将 Amazon DocumentDB 3.6 集群的完整负载迁移到 Amazon DocumentDB 4.0 集群，现在正在复制更改事件。

Summary			
Status	Type	Source	Target
🟢 Load complete, replication ongoing	Full load, ongoing replication	docdb36source	docdb40target

最终您的源和目标将保持同步。您可以通过运行来验证它们是否同步 `count()` 操作以验证所有更改事件是否已迁移。

步骤 8：将应用程序终端节点更改为 Amazon DocumentDB 群集 4.0

在完全加载完成并且 CDC 流程持续复制后，您可以将应用程序的数据库连接终端节点从 Amazon DocumentDB 3.6 集群更改为 Amazon DocumentDB 群集 4.0 集群。

迁移工具

在迁移到 Amazon DocumentDB 时，大多数客户使用的两个主要工具是 [Amazon Database Migration Service \(Amazon DMS\)](#) 和命令行实用程序如 `mongodump` 和 `mongoexport`。作为最佳做法，对于上述任一选项，我们建议您在开始迁移之前首先在 Amazon DocumentDB 中创建索引，因为这可以减少总体时间并提高迁移速度。为此，您可以使用 [Amazon DocumentDB 索引工具](#)。

Amazon Database Migration Service

Amazon Database Migration Service(Amazon DMS) 是一项云服务，可将关系数据库和非关系数据库轻松迁移到 Amazon DocumentDB。您可以使用Amazon DMS将数据从本地或 EC2 上托管的数据库迁移到 Amazon DocumentDB。与Amazon DMS，可以执行一次性迁移，也可以复制持续更改以保持源和目标同步。

有关使用的更多信息Amazon DMS要迁移到Amazon DocumentDB，请参阅：

- [将 MongoDB 作为 源Amazon DMS](#)
- [将 Amazon DocumentDB 作为 Amazon DB 的目标Amazon Database Migration Service](#)
- [演练：从 MongoDB 迁移到 Amazon DocumentDB](#)

命令行实用程序

用于将数据迁移到以及从中迁移数据的常用实用程序，包括mongodump、mongoexport、mongoimport、mongoexport、mongoimport。通常，mongodump 和 mongoexport 是最有效的实用程序，因为它们使用二进制格式在数据库中存储和还原数据。这通常是性能最高的选项，而且与逻辑导出相比，产生的数据量更少。如果您要使用 JSON 或 CSV 等逻辑格式导出和导入数据，可以使用 mongoexport 和 mongoimport，因为此时数据是人类可读格式，但速度通常比 mongodump/mongoexport 慢，而且得到的数据量也较多。

这些区域有：[迁移方法 \(p. 124\)](#)以下部分将讨论何时最适合使用Amazon DMS和命令行实用程序，基于您的使用案例和要求。

发现

对于您的每个 MongoDB 部署，您应该确定并记录两组数据：架构详情和运行特征。该信息将帮助您选择适当的迁移方法和集群大小调整。

架构详情

- 名称

选择用于跟踪该部署的唯一名称。

- Version

记录部署正在运行的 MongoDB 版本。要查找版本，请使用 mongo shell 连接到副本集成员并运行 `db.version()` 操作。

- Type

记录您的部署是独立的 mongo 实例、副本集还是分片集群。

- 成员

记录每个集群、副本集或独立成员的主机名、地址和端口。

对于集群化部署，您可以通过使用 mongo shell 连接到 mongo 主机并运行 `sh.status()` 操作来查找分片成员。

对于副本集，您可以通过使用 mongo shell 连接到副本集成员并运行 `rs.status()` 操作来获取成员。

- Oplog 大小

对于副本集或分片集群，记录每个副本集成员的 oplog 大小。要查找成员的 oplog 大小，请使用 mongo shell 连接到副本集成员并运行 `ps.printReplicationInfo()` 操作。

- 副本集成员优先级

对于副本集或分片集群，记录每个副本集成员的优先级。要查找副本集成员优先级，请使用 mongo shell 连接到副本集成员并运行 `rs.conf()` 操作。优先级显示为 `priority` 键的值。

- 是否使用 TLS/SSL

记录是否在每个节点上使用传输层安全性 (TLS)/安全套接字层 (SSL) 进行传输加密。

运行特征

- 数据库统计数据

对于每个集合，记录以下信息：

- 名称
- 数据大小
- 集合数量

要查找数据库统计数据，请使用 mongo shell 连接到数据库并运行命令 `db.runCommand({dbstats: 1})`。

- 集合统计数据

对于每个集合，记录以下信息：

- 命名空间
- 数据大小
- 索引数量
- 集合是否有上限

- 索引统计数据

对于每个集合，记录以下索引信息：

- 命名空间
- ID

- Size

- 密钥
- TTL
- 稀疏
- 背景

要查找索引信息，请使用 mongo shell 连接到数据库并运行命令 `db.collection.getIndexes()`。

- Opcounters

该信息可帮助您了解当前的 MongoDB 工作负载模式（读取操作繁重、写入操作繁重还是平衡）。它还能够为您的初始选择 Amazon DocumentDB 实例提供指导。

以下是要在监测期间收集的关键信息（以计数/秒为单位）：

- 查询
- 插入
- 更新
- 删除

您可以通过绘制一段时间内 `db.serverStatus()` 命令的输出来获取该信息。也可以使用 `mongostat` 工具获取这些统计数据的即时值。但使用该选项时，您的迁移计划可能无法涵盖峰值负载。

- 网络统计数据

该信息可帮助您了解当前的 MongoDB 工作负载模式（读取操作繁重、写入操作繁重还是平衡）。它还能够为您的初始选择 Amazon DocumentDB 实例提供指导。

以下是要在监测期间收集的关键信息（以计数/秒为单位）：

- 连接
- 网络输入字节数
- 网络输出字节数

您可以通过绘制一段时间内 `db.serverStatus()` 命令的输出来获得该信息。也可以使用 `mongostat` 工具获取这些统计数据的即时值。但使用该选项时，您的迁移计划可能无法涵盖峰值负载。

规划：Amazon DocumentDB 集群要求

要实现成功迁移，您需要仔细考虑 Amazon DocumentDB 集群的配置以及应用程序访问集群的方式。在确定集群要求时，请考虑以下每个维度：

- 可用性

Amazon DocumentDB 通过部署副本实例提供高可用性，副本实例可在称为的过程中提升为主实例故障转移。通过将副本实例部署到不同的可用区，您可以实现更高级别的可用性。

下表提供了 Amazon DocumentDB 部署配置指南，以满足特定的可用性目标。

可用性目标	实例总数	副本	可用区
99%	1	0	1
99.9%	2	1	2
99.99%	3	2	3

整体系统可靠性必须考虑所有组件，而不仅仅是数据库。有关满足整体系统可靠性需求的最佳实践和建议，请参阅[Amazon Well-Architected 白皮书](#)。

- 性能

Amazon DocumentDB 实例允许您读写集群的存储卷。集群实例有多种类型，它们具有不同的内存和 vCPU，这将影响集群的读写性能。使用您在发现阶段收集的信息，选择可支持您的工作负载性能要求的实例类型。有关受支持实例类型的列表，请参阅[管理实例类 \(p. 283\)](#)。

为 Amazon DocumentDB 集群选择实例类型时，请考虑工作负载性能要求的以下方面：

- vCPU— 需要更高连接数的架构可能会受益于具有更多 vCPUS 的实例。
- 内存— 如果可能，将工作数据集保留在内存中可提供最高的性能。一项基本准则是为 Amazon DocumentDB 引擎预留三分之一的实例内存，将剩余的三分之二用于工作数据集。
- 连接— 最小的最佳连接数是每个 Amazon DocumentDB 实例 vCPU 8 个连接。虽然 Amazon DocumentDB 实例的连接限制要高得多，但在连接数超过每 vCPU 8 个连接后，更多连接带来的性能优势将下降。
- 网络— 具有大量客户端或连接的工作负载应考虑插入和检索数据所需的聚合网络性能。批量操作可更有效地使用网络资源。
- 插入性能— 单个文档插入通常是将数据插入到 Amazon DocumentDB 的最慢方法。批量插入操作要比单个插入快得多。
- 阅读表现从工作内存读取始终比从存储卷返回的读取要快。因此，优化实例内存大小以将工作集保留在内存中是最佳的做法。

除了从主实例提供读取之外，Amazon DocumentDB 集群还会自动配置为副本集。之后，您可以通过在 MongoDB 驱动程序中设置读取首选项来将只读查询路由到只读副本。您可以通过添加副本来扩展读取流量，从而减少主实例上的总体负载。

可以在同一集群中部署不同实例类型的 Amazon DocumentDB 副本。一个作为示例的使用案例是启动一个具有较大实例类型的副本来处理临时分析流量。如果部署一组混合实例类型，请确保为每个实例配置故障转移优先级。这有助于确保故障转移事件始终提升足够大小的副本来处理写入负载。

- 恢复

Amazon DocumentDB 会在写入数据时会持续备份数据。它提供了以下功能 point-in-time 在 1 到 35 天可配置的时间段内恢复 (PITR) 功能，称为备份保留期。默认备份保留期为一天。Amazon DocumentDB 还会自动创建存储卷的每日快照，这些快照也在配置的备份保留期内保留。

如果您需要保留超出备份保留期的快照，还可以使用 Amazon Web Services Management Console 和 Amazon Command Line Interface (Amazon CLI) 随时启动手动快照。有关更多信息，请参阅 [在 Amazon DocumentDB 中备份和还原 \(p. 194\)](#)。

在规划迁移时，请考虑以下事项：

- 选择满足恢复点目标 (RPO) 要求的备份保留期 (1-35 天)。
- 确定是否需要手动快照，如果需要，确定时间间隔。

迁移方法

主要有三种方法可用于将数据迁移到 Amazon DocumentDB。

Note

虽然您可以随时在 Amazon DocumentDB 中创建索引，但整体来说，先创建索引，然后再导入大型数据集的速度会更快。作为最佳做法，我们建议对于以下每种方法，在执行迁移之前首先在 Amazon DocumentDB 中创建索引。为此，您可以使用 [Amazon DocumentDB 索引工具](#)。

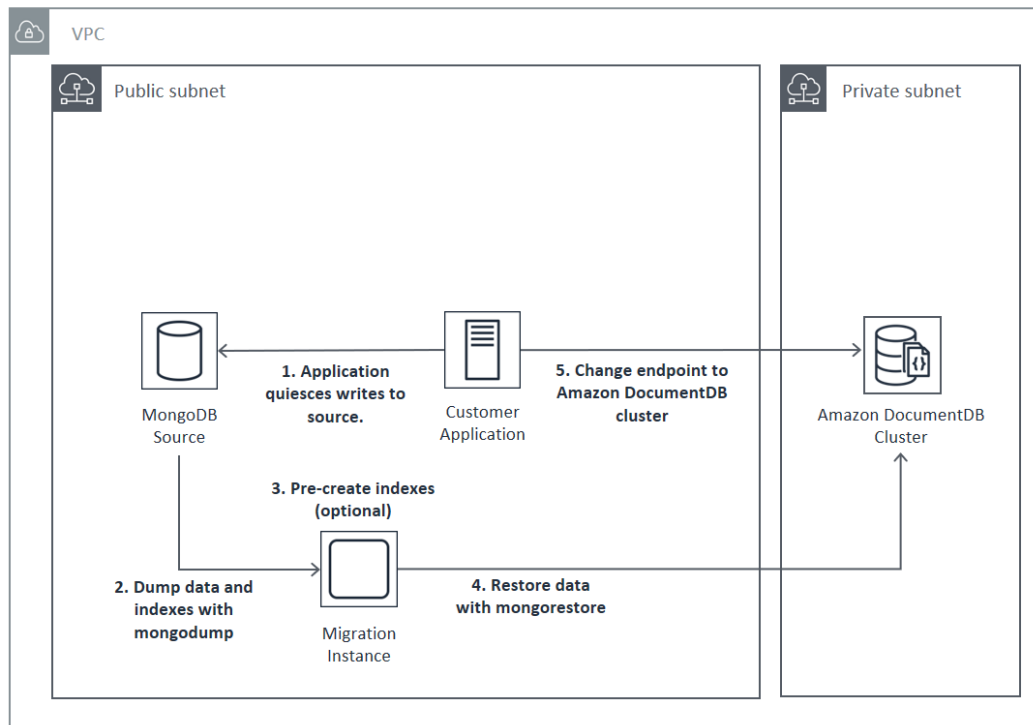
离线

这些区域有：离线方法使用 `mongodump` 和 `mongorestore` 工具将数据从源 MongoDB 部署迁移到 Amazon DocumentDB 集群。离线方法是最简单的迁移方法，但是它也会给集群带来最多的停机时间。

离线迁移的基本过程如下所示：

1. Quiesce 对 MongoDB 源进行写入。
2. 从源 MongoDB 部分转储集合数据和索引。
3. 将索引还原到 Amazon DocumentDB 集群。
4. 将集合数据还原到 Amazon DocumentDB 集群。
5. 将您的应用程序终端节点更改为写入 Amazon DocumentDB 集群。

Offline Migration Approach



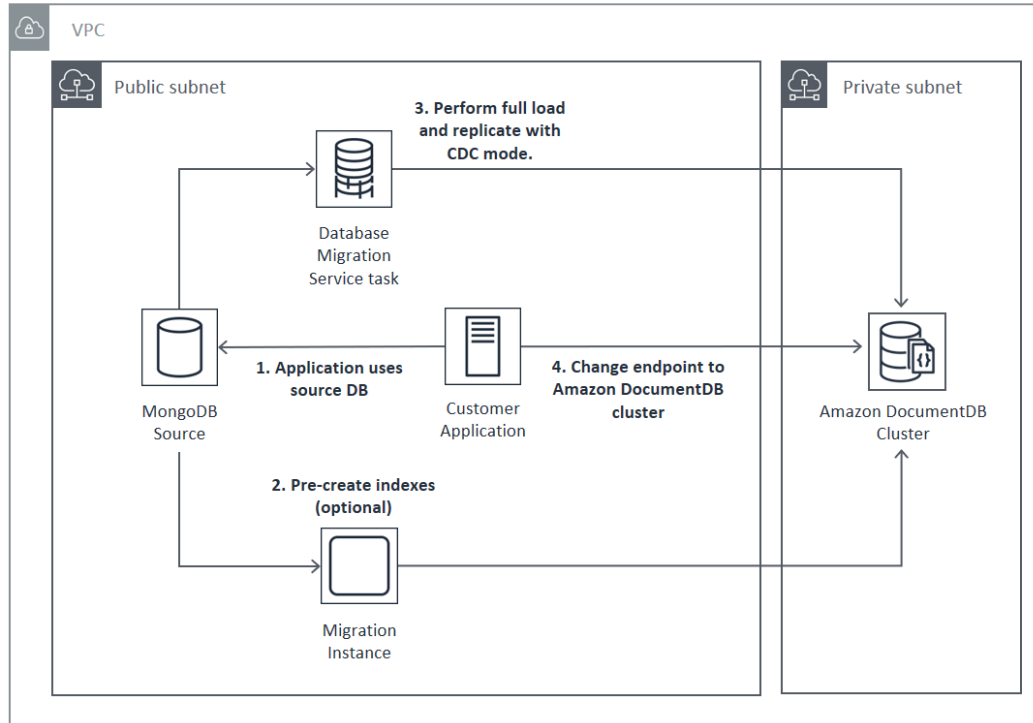
在线

在线 方法使用 Amazon Database Migration Service (Amazon DMS)。它执行从源 MongoDB 部署到源 MongoDB Amazon DocumentDB 集群的数据完全加载。然后，它会切换至更改数据捕获 (CDC) 模式来复制更改。在线方法可最大限度减少集群停机时间，但它是三种方法中速度最慢的一个。

在线迁移的基本过程如下所示：

1. 您的应用程序正常使用源数据库。
2. (可选) 在 Amazon DocumentDB 集群中预先创建索引。
3. 创建 Amazon DMS 任务执行完全加载，然后启用从源 MongoDB 部署到 Amazon DocumentDB 集群的 CDC。
4. After Amazon DMS 任务已完成完全加载，并且正在将更改复制到 Amazon DocumentDB，将应用程序的终端节点切换到 Amazon DocumentDB 集群。

Online Migration Approach



有关使用的更多信息Amazon DMS要迁移，请参阅[将 Amazon DocumentDB 作为 Amazon DB 的目标](#) [Amazon Database Migration Service](#)和相关的[教程](#)中的Amazon Database Migration Service用户指南.

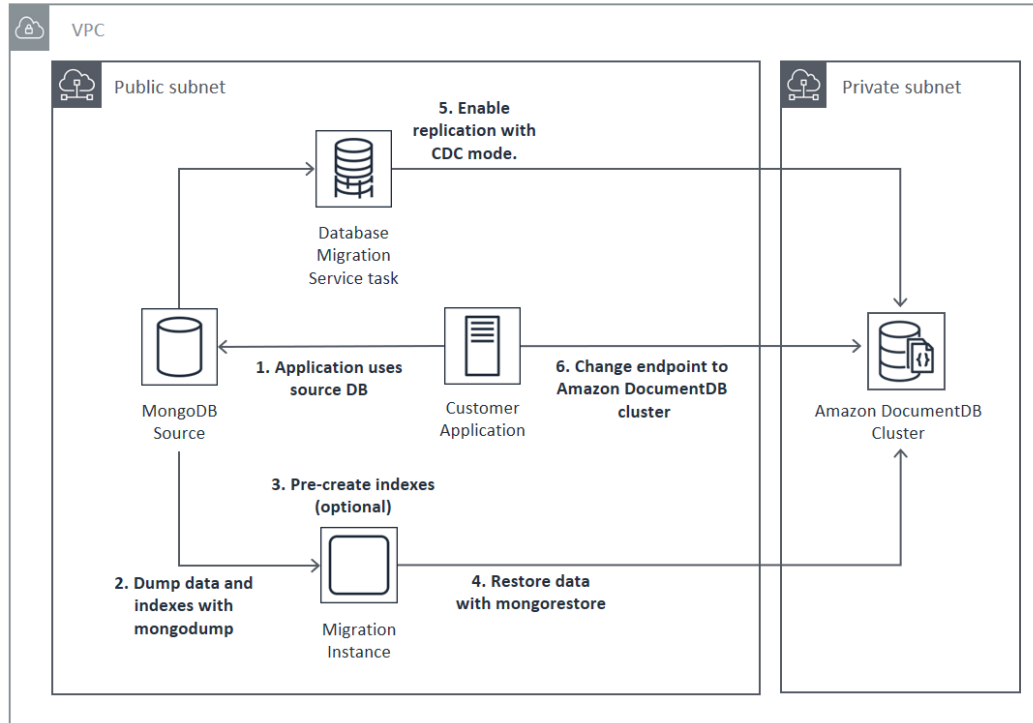
混合

这些区域有：混合动力方法使用mongodump和mongoexport工具将数据从源 MongoDB 部署迁移到 Amazon DocumentDB 集群。然后，它在 CDC 模式下使用 Amazon DMS 复制更改。混合方法使迁移速度和停机时间达到平衡，但它是这三种方法中最复杂的方法。

混合迁移的基本过程如下所示：

1. 您的应用程序正常使用源 MongoDB 部署。
2. 从源 MongoDB 部分转储集合数据和索引。
3. 将索引还原到 Amazon DocumentDB 集群。
4. 将集合数据还原到 Amazon DocumentDB 集群。
5. 创建Amazon DMS任务来启用从源 MongoDB 部署到 Amazon DocumentDB 集群的 CDC。
6. 当Amazon DMS任务是在可接受时段内复制更改后，将您的应用程序终端节点更改为 Amazon DocumentDB 集群。

Hybrid Migration Approach



Important

Amazon DMS 任务当前只能迁移单个数据库。如果 MongoDB 源具有大量数据库，则可能需要实现迁移任务的自动创建，或考虑使用离线方法。

不管您选择何种迁移方法，在迁移数据之前，在 Amazon DocumentDB 集群中预先创建索引都是一种最有效的措施。这是因为 Amazon DocumentDB 索引是parallel 插入的数据，但对现有数据创建索引是单线程操作。

Amazon DMS 不迁移索引（仅迁移数据），因此无需额外步骤来避免二次创建索引。

迁移源

如果您的 MongoDB 源是一个独立的 mongo 进程，并且您想要使用在线或混合迁移方法，请先将独立的 mongo 转换为副本集，以便创建 oplog 来用作 CDC 源。

如果您要从 MongoDB 副本集或分片集群进行迁移，请考虑为每个副本集或分片创建链接或隐藏的辅助副本集或分片来用作迁移源。执行数据转储可能会迫使工作集数据超出内存并影响生产实例的性能。您可以通过从不提供生产数据的节点迁移来降低这一风险。

迁移源版本

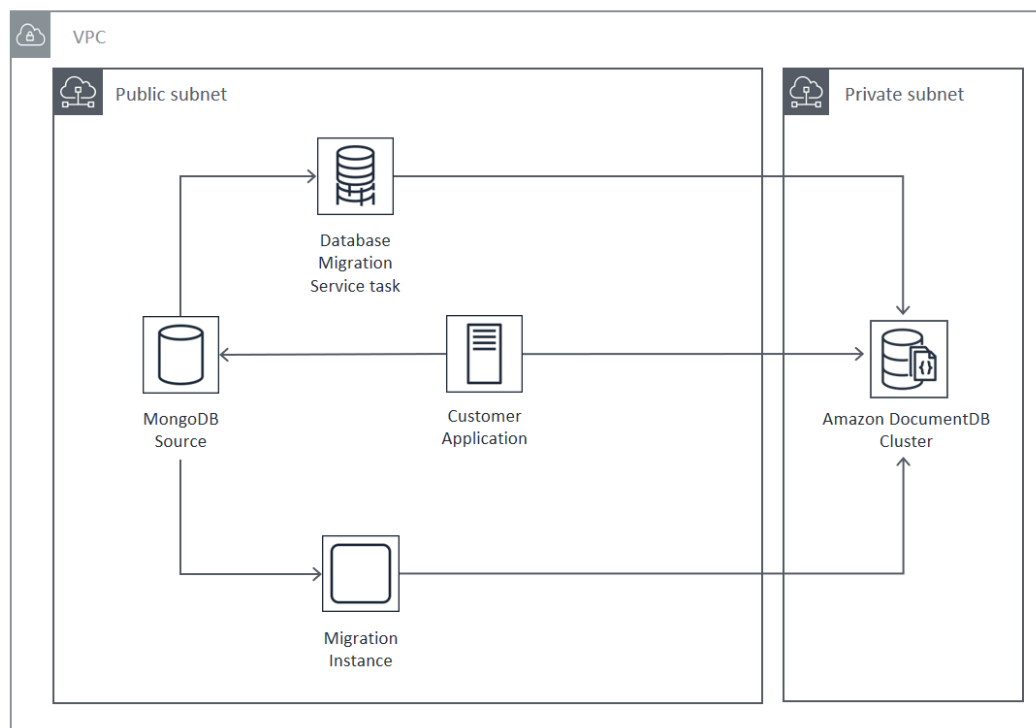
如果源 MongoDB 数据库版本与目标 Amazon DocumentDB 集群的兼容版本不同，您可能需要采取其他准备步骤来确保成功迁移。最常遇到的两个要求是将源 MongoDB 安装升级到支持的迁移版本（MongoDB 3.0 或更高版本），和升级应用程序驱动程序以支持目标 Amazon DocumentDB 版本。

如果您的迁移具有上述任一要求，请务必在迁移计划中包括这些步骤，以升级和测试任何驱动程序更改。

迁移连接

您可以从在数据中心运行的源 MongoDB 部署或从 Amazon EC2 实例上运行的 MongoDB 部署迁移到 Amazon DocumentDB。从在 EC2 上运行的 MongoDB 迁移很简单，只需要正确配置安全组和子网。

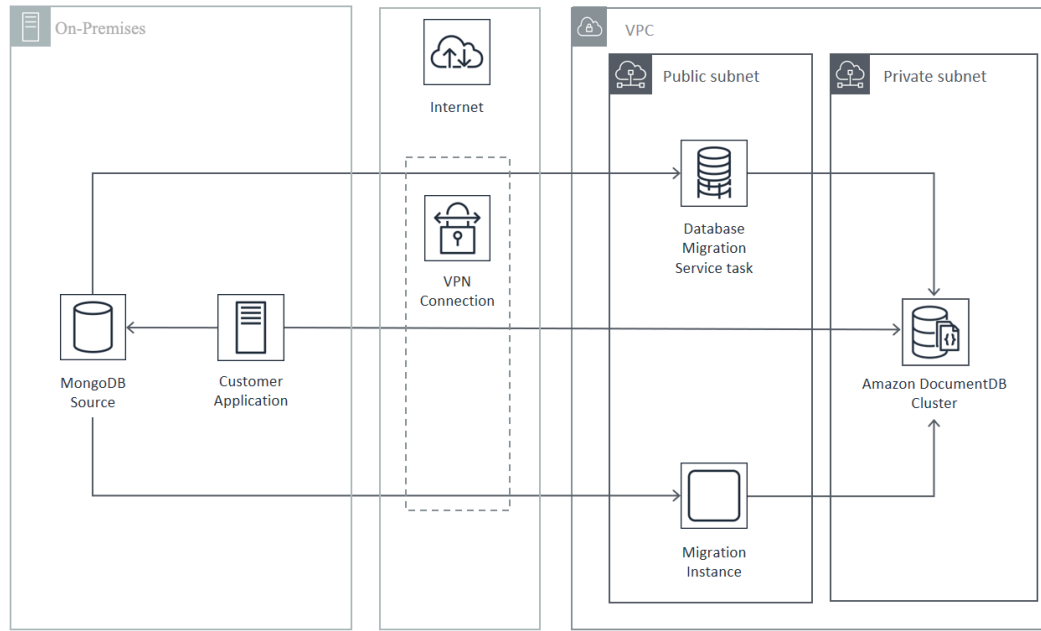
Migrating from EC2 Source



从本地数据库迁移需要 MongoDB 部署与 Virtual Private Cloud (VPC) 之间的连接。您可以通过虚拟专用网络 (VPN) 连接或使用 Amazon Direct Connect 服务来完成该操作。虽然您可以通过 Internet 迁移到 VPC，但从安全角度来看，这种连接方法是最不可取的。

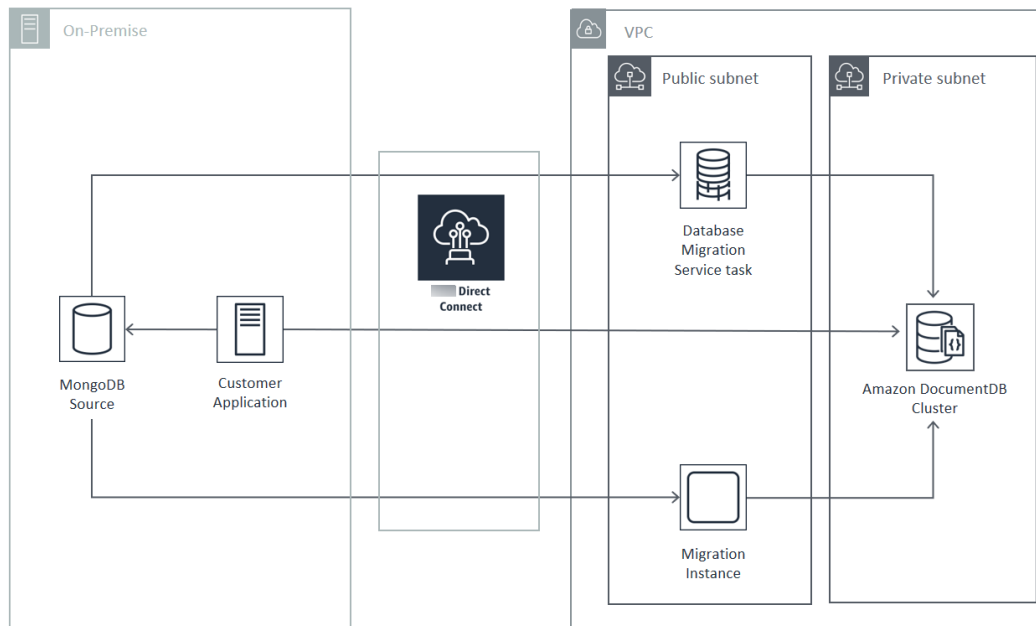
下图说明了通过 VPN 连接从本地源到 Amazon DocumentDB 的迁移。

Migrating from On-Premise Source (VPN)



下图表示使用从本地源到 Amazon DocumentDB 的迁移Amazon Direct Connect.

Migrating from On-Premise Source (Direct Connect)



在线和混合迁移方法需要使用Amazon DMS实例，它必须在Amazon VPC中的Amazon EC2上运行。所有方法都需要迁移服务器来运行 `mongodump` 和 `mongoexport`。通常，在启动 Amazon DocumentDB 集群的 VPC 中的 Amazon EC2 实例上运行迁移服务器会更容易，因为这能够大幅简化与 Amazon DocumentDB 集群的连接。

测试

以下是迁移前测试的目标：

- 验证所选方法能否实现期望的迁移结果。
- 验证实例类型和读取首选项选择是否满足应用程序的性能要求。
- 验证应用程序在故障转移期间的行为。

迁移计划测试注意事项

在测试您的 Amazon DocumentDB 迁移计划时，请考虑以下事项。

主题

- [还原索引 \(p. 130\)](#)
- [转储数据 \(p. 130\)](#)
- [还原数据 \(p. 130\)](#)
- [Oplog 大小调整 \(p. 130\)](#)
- [Amazon Database Migration Service 配置 \(p. 131\)](#)
- [从分片集群迁移 \(p. 131\)](#)

还原索引

默认情况下，mongorestore 会为转储的集合创建索引，但是在还原数据后创建它们。在将数据还原到集群之前，在 Amazon DocumentDB 中创建索引总体上更快。这是因为索引操作在数据加载期间是并行化的。

如果您选择预先创建索引，则在使用 mongorestore 还原数据时可通过提供 `--noIndexRestore` 选项跳过索引创建步骤。

转储数据

mongodump 工具是从源 MongoDB 部署转储数据的首选方法。根据迁移实例上的可用资源，您也许可以通过使用 mongodump 选项增加转储的并行连接数（默认为 4）来加速 `--numParallelCollections`。

还原数据

这些区域有：mongorestore 工具是将转储数据还原到 Amazon DocumentDB 实例的首选方法。您可以通过使用 `--numInsertionWorkersPerCollection` 选项增加还原期间每个集合的工作线程数来提高还原性能。Amazon DocumentDB 集群主实例上的每个 vCPU 分配 1 个工作线程是一个不错的起点。

Amazon DocumentDB 目前不支持 mongorestore 工具的 `--oplogReplay` 选项。

默认情况下，mongorestore 将跳过插入错误并继续还原过程。如果您将不支持的数据还原到 Amazon DocumentDB 实例，就会发生这种情况。例如，您的文档包含带有空字符串的键或值。如果您希望在遇到任何还原错误时完全停止 mongorestore 操作，请使用 `--stopOnError` 选项。

Oplog 大小调整

MongoDB 操作日志 (oplog) 是一种受限集合，其中包含对数据库的所有数据修改。您可以通过对副本集或分片成员运行 `db.printReplicationInfo()` 操作来查看 oplog 的大小及其包含的时间范围。

如果您使用的是在线或混合方法，请确保每个副本集或分片上的 oplog 足够大，以包含在整个数据迁移过程中所做全部更改（无论是通过 mongodump 还是 Amazon DMS 任务完全加载），以及合理的缓冲区。

有关更多信息，请参阅 MongoDB 文档中的 [Check the Size of the Oplog](#) (查看 Oplog 的大小)。通过记录 `mongodump` 或 `mongorestore` 进程或 Amazon DMS 完全加载任务首次测试运行所用的时间来确定所需的最小 oplog 大小。

Amazon Database Migration Service 配置

这些区域有：[Amazon Database Migration Service 用户指南](#) 涵盖了将 MongoDB 源数据迁移到 Amazon DocumentDB 集群所需的组件和步骤。以下是使用 Amazon DMS 执行在线或混合迁移的基本过程：

使用 Amazon DMS 执行迁移

1. 创建 MongoDB 源终端节点。有关更多信息，请参阅[使用 MongoDB 作为 Amazon DMS 源](#)。
2. 创建一个 Amazon DocumentDB 目标终端节点。有关更多信息，请参阅[使用 Amazon DMS 终端节点](#)。
3. 创建至少一个 Amazon DMS 复制实例。有关更多信息，请参阅[使用 Amazon DMS 复制实例](#)。
4. 创建至少一个 Amazon DMS 复制任务。有关更多信息，请参阅[使用 Amazon DMS 任务](#)。

对于在线迁移，迁移任务使用迁移类型 `Migrate existing data and replicate ongoing changes` (迁移现有数据并复制持续更改)。

对于混合迁移，迁移任务使用迁移类型 `Replicate data changes only` (仅复制数据更改)。您可以选择 CDC 开始时间以与 `mongodump` 操作中的转储时间保持一致。MongoDB oplog 是幂等的。为避免遗漏更改，最好在 `mongodump` 完成时间与 CDC 开始时间之间保留几分钟的重叠时间。

从分片集群迁移

将数据从分片集群迁移到 Amazon DocumentDB 实例的过程基本上是 parallel 迁移多个副本集的过程。测试分片集群迁移时的一个关键考虑因素是某些分片可能比其他分片使用得更频繁。这种情况会导致数据迁移具有不同的运行时间。请务必评估每个分片的 oplog 规划和测试时的要求。

以下是迁移分片集群时要考虑的一些配置问题：

- 在运行 `mongodump` 或启动 Amazon DMS 迁移任务之前，必须禁用分片集群均衡器并等待所有正在进行的迁移完成。有关更多信息，请参阅 MongoDB 文档中的 [Disable the Balancer](#) (禁用均衡器)。
- 如果您使用 Amazon DMS 复制数据，请在运行迁移任务之前在每个分片上运行 `cleanupOrphaned` 命令。如果不运行该命令，任务可能会因为重复的文档 ID 而失败。请注意，该命令可能会影响性能。有关更多信息，请参阅 MongoDB 文档中的 [cleanupOrphaned](#)。
- 如果您使用 `mongodump` 工具转储数据，则应为每个分片运行一个 `mongodump` 进程。最节省时间的方法可能需要多个迁移服务器来最大限度提高转储性能。
- 如果您使用 Amazon Database Migration Service 复制数据，则必须为每个分片创建源终端节点。同时为要迁移的每个分片运行至少一个迁移任务。最节省时间的方法可能需要多个复制实例来最大限度提高迁移性能。

性能测试

将数据成功迁移到测试 Amazon DocumentDB 集群后，请对集群执行测试工作负载。通过亚马逊验证 CloudWatch 性能达到或超过 MongoDB 源部署的当前吞吐量的指标。

验证以下关键的 Amazon DocumentDB 指标：

- 网络吞吐量
- 写入吞吐量
- 读取吞吐量
- 副本滞后

有关更多信息，请参阅 [监控 Amazon DocumentDB \(p. 365\)](#)。

故障转移测试

验证应用程序在 Amazon DocumentDB 故障转移事件期间的行为是否满足您的可用性要求。要在控制台上启动 Amazon DocumentDB 群集的手动故障转移，请在集群页面上，选择故障转移对的操作操作Menu。

也可以通过从 Amazon CLI 执行 `failover-db-cluster` 操作来启动故障转移。有关更多信息，请参阅 [failover-db-cluster](#) 在亚马逊的 DocumentDB 部分中 Amazon CLI 引用。

其他资源

请参阅 Amazon Database Migration Service 用户指南 中的以下主题：

- [将 Amazon DocumentDB 作为 Amazon DB 的目标 Amazon Database Migration Service](#)
- [演练：从 MongoDB 迁移到 Amazon DocumentDB](#)

Amazon DocumentDB 中的安全性

Amazon 的云安全性具有优先级最高。作为 Amazon 客户，您将从专为满足大多数安全敏感型组织的要求而打造的数据中心和网络架构中受益。

安全性是 Amazon 和您的共同责任。此文档帮助您了解如何在使用 Amazon DocumentDB 时应用责任共担模式。[责任共担模型](#)将其描述为云的安全性和云中的安全性：

- 云的安全性—Amazon负责保护运行的基础设施Amazon中的服务Amazon云。Amazon还向您提供可安全使用的服务。作为 [Amazon 合规性计划](#)的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于 Amazon DocumentDB 的合规性计划（兼容 MongoDB），请参阅[Amazon合规性计划范围内的服务](#)。
- 云中的安全性—您的责任由Amazon你使用的服务。您还需要对其他因素负责，包括您的数据的敏感性、您组织的要求以及适用的法律法规。

您还将了解如何使用其他Amazon帮助您监控和保护 Amazon DocumentDB 资源的服务。以下主题说明如何配置 Amazon DocumentDB 以实现您的安全性和合规性目标。

主题

- [Amazon DocumentDB 中的数据保护 \(p. 133\)](#)
- [Amazon DocumentDB 中的 Identity and Access Management \(p. 142\)](#)
- [Amazon DocumentDB 用户 \(p. 154\)](#)
- [基于角色的访问控制限制数据库访问 \(p. 156\)](#)
- [Amazon DocumentDB 中的日志记录和监控 \(p. 172\)](#)
- [更新您的 Amazon DocumentDB TLS 证书 \(cn-northwest-1\) \(p. 173\)](#)
- [\(p. 178\)](#)
- [Amazon DocumentDB 中的合规性验证 \(p. 183\)](#)
- [Amazon DocumentDB 中的恢复功能 \(p. 184\)](#)
- [Amazon DocumentDB 中的基础设施安全性 \(p. 184\)](#)
- [Amazon DocumentDB 的安全最佳实践 \(p. 185\)](#)
- [审计Amazon DocumentDB \(p. 185\)](#)

Amazon DocumentDB 中的数据保护

这些区域有：Amazon [责任共担模式](#)适用于中的数据保护。如该模式中所述，Amazon 负责保护运行所有 Amazon Web Services 云 的全球基础设施。您负责维护对托管在此基础设施上的内容的控制。此内容包括您所使用的Amazon Web Services的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。

出于数据保护目的，我们建议您保护 Amazon Web Services 账户 凭证并使用 Amazon Identity and Access Management (IAM) 设置单独的用户账户。这仅向每个用户授予履行其工作职责所需的权限。我们还建议您通过以下方式保护您的数据：

- 对每个账户使用 Multi-Factor Authentication (MFA)。
- 使用 SSL/TLS 与 Amazon 资源进行通信。建议使用 TLS 1.2 或更高版本。
- 使用 Amazon CloudTrail 设置 API 和用户活动日志记录。
- 使用 Amazon 加密解决方案以及 Amazon 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Simple Storage Service (Amazon S3) 中的个人数据。

- 如果在通过命令行界面或 API 访问 Amazon 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅 [《美国联邦信息处理标准 \(FIPS\) 第 140-2 版》](#)。

我们强烈建议您切勿将机密信息或敏感信息（例如您客户的电子邮件地址）放入标签或自由格式字段（例如名称字段）。这包括使用 Amazon DocumentDB 或其他工作时 Amazon 使用控制台、API、Amazon CLI，或者 Amazon 开发工具包。您在用于名称的标签或自由格式字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，我们强烈建议您不要在 URL 中包含凭证信息来验证您对该服务器的请求。

主题

- [静 Amazon DocumentDB 数据 \(p. 134\)](#)
- [加密传输中的数据 \(p. 136\)](#)
- [密钥管理 \(p. 142\)](#)

静 Amazon DocumentDB 数据

Note

Amazon KMS 正将术语客户托管密钥 (CMK) 替换为 Amazon KMS key 和 KMS 密钥。这一概念并未改变。为防止破坏性更改，Amazon KMS 保留了此术语的一些变体。

您可以通过在创建集群时指定存储加密选项来在 Amazon DocumentDB 集群中加密静态数据。存储加密在整个集群范围内启用，应用于所有实例，包括主实例和任何副本。它还应用于集群的存储卷、数据、索引、日志、自动备份和快照。

Amazon DocumentDB AES-256 Amazon Key Management Service (Amazon KMS)。在使用已启用静态加密的 Amazon DocumentDB Amazon DocumentDB

Amazon DocumentDB Amazon KMS 并使用称为信封加密的方法来保护您的数据。当 Amazon DocumentDB 集群使用加密 Amazon KMS，Amazon DocumentDB Amazon KMS 使用您的 KMS 密钥 [生成密文数据密钥](#) 对存储卷进行加密。密钥使用您定义的 KMS 密钥进行加密，并与加密数据和存储元数据一起存储。当 Amazon DocumentDB 需要访问您的加密数据时，它会请求 Amazon KMS 使用 KMS 密钥解密数据密钥，并将明文数据密钥缓存到内存中，以高效地加密和解密存储卷中的数据。

Amazon DocumentDB Amazon Web Services 区域 Amazon DocumentDB

为 Amazon DocumentDB

在 Amazon DocumentDB Amazon Web Services Management Console 或者 Amazon Command Line Interface (Amazon CLI)。默认情况下，您使用控制台创建的集群启用了静态加密功能。默认情况下，您使用 Amazon CLI 创建的集群禁用了静态加密功能。因此，您必须使用 `--storage-encrypted` 参数显式启用静态加密。无论哪种情况，在创建集群后，都无法更改静态加密选项。

Amazon DocumentDB Amazon KMS 检索和管理加密密钥，并定义控制这些密钥的使用方式的策略。如果您未指定 Amazon KMS 密钥标识符，Amazon DocumentDB 使用默认的 Amazon 托管服务 KMS 密钥。Amazon DocumentDB 为每个文件创建单独的 KMS 密钥 Amazon Web Services 区域在你的 Amazon Web Services 账户。有关更多信息，请参阅 [Amazon Key Management Service 概念](#)。

要开始创建自己的 KMS 密钥，请参阅 [开始使用](#) 中的 Amazon Key Management Service 开发人员指南。

Important

您必须使用对称加密 KMS 密钥加密您的集群，因为 Amazon DocumentDB 请勿使用非对称 KMS 密钥，尝试对 Amazon DocumentDB 有关更多信息，请参阅 [中的非对称密钥 Amazon KMS 中的 Amazon Key Management Service 开发人员指南](#)。

如果 Amazon DocumentDB 在此情况下，您只能从备份还原集群。对于 Amazon DocumentDB

此外，如果禁用加密的 Amazon DocumentDB 当 Amazon DocumentDB 在此状态下，集群不再可用，并且数据库的当前状态无法恢复。要还原集群，您必须重新启用对 Amazon DocumentDB

Important

在已创建加密的集群的 KMS 密钥，您无法更改它。请确保先确定您的加密密钥要求，然后再创建加密的集群。

使用 Amazon Web Services Management Console

您可在创建集群时，指定静态加密选项。默认情况下，当您使用 Amazon Web Services Management Console 创建集群时，静态加密处于启用状态。集群创建之后无法修改该选项。

在创建集群时指定静态加密选项

1. 创建 AAmazon DocumentDB [开始使用](#) 部分。但在步骤 6 中，不要选择创建集群。
2. 在 Authentication (身份验证) 部分下，选择 Show advanced settings (显示高级设置)。
3. 向下滚动到 Encryption-at-rest 部分。
4. 选择要进行静态加密的选项。无论您选择哪个选项，都无法在创建集群后更改它。
 - 要对此集群中的静态数据进行加密，请选择启用加密。
 - 如果您不想对此集群中的静态数据进行加密，请选择 Disable encryption (禁用加密)。
5. 选择您想要的主密钥。Amazon DocumentDB Amazon Key Management Service (Amazon KMS) 检索和管理加密密钥，并定义控制这些密钥的使用方式的策略。如果您未指定 Amazon KMS 密钥标识符，Amazon DocumentDB 使用默认的 Amazon 托管服务 KMS 密钥。有关更多信息，请参阅 [Amazon Key Management Service 概念](#)。

Note

创建加密的集群后，您无法更改该集群的 KMS 密钥。请确保先确定您的加密密钥要求，然后再创建加密的集群。

6. 根据需要完成其他部分，然后创建您的集群。

使用 Amazon CLI

使用加密 Amazon DocumentDB 集群 Amazon CLI，则必须指定 `--storage-encrypted` 选项在创建集群时使用。使用创建的 Amazon DocumentDB 集群 Amazon CLI 默认情况下不启用存储加密。

以下示例创建启用了存储加密的 AAmazon DocumentDB

Example

对于 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --port 27017 \  
  --engine docdb \  
  --master-username yourMasterUsername \  
  --master-user-password yourMasterPassword \  
  --storage-encrypted
```

对于 Windows：

```
aws docdb create-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --port 27017 ^  
  --engine docdb ^  
  --master-username yourMasterUsername ^
```



```
--master-user-password yourMasterPassword ^  
--storage-encrypted
```

在创建加密的 Amazon DocumentDB Amazon KMS 密钥标识符，如以下示例所示。

Example

对于 Linux、macOS 或 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
```

对于 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

创建加密的集群后，您无法更改该集群的 KMS 密钥。请确保先确定您的加密密钥要求，然后再创建加密的集群。

Amazon DocumentDB

Amazon DocumentDB

- 您只能在创建集群 Amazon DocumentDB 在创建之后启用或禁用静态加密。但是，您可以通过创建未加密的集群的快照，然后将未加密的快照还原为新集群，同时指定静态加密选项，来创建未加密的集群的加密副本。

有关更多信息，请参阅以下主题：

- [创建手动集群快照 \(p. 201\)](#)
- [从集群快照还原 \(p. 213\)](#)
- [复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)
- 不能通过对已启用存储加密的 Amazon DocumentDB
- Amazon DocumentDB

加密传输中的数据

您可以使用传输层安全性 (TLS) 加密应用程序和 Amazon DocumentDB 集群之间的连接。默认情况下，为新创建的 Amazon DocumentDB 集群启用传输中的加密。可以在创建集群时或稍后禁用它。启用传输中加密后，需要使用 TLS 进行安全连接才能连接到集群。有关使用 TLS 连接到 Amazon DocumentDB 的更多信息，请参阅[以编程方式连接到 Amazon DocumentDB \(p. 428\)](#)。

管理 Amazon DocumentDB 集群 TLS 设置

Amazon DocumentDB 集群的传输中加密通过 [集群参数组](#)。Amazon DocumentDB 使用 Amazon Web Services Management Console 或者 Amazon Command Line Interface (Amazon CLI)。有关如何验证和修改当前 TLS 设置的信息，请参阅以下部分。

使用 Amazon Web Services Management Console

请按照以下步骤使用控制台执行 TLS 加密的管理任务，例如识别参数组、验证 TLS 值以及进行必要的修改。

Note

除非您在创建集群时另行指定，否则将使用默认集群参数组创建集群。中的参数 `default` 无法修改集群参数组（例如，`tls` 启用/禁用）。因此，如果您的集群使用 `default` 集群参数组，则需要修改集群来使用非默认集群参数组。首先，您可能需要创建自定义集群参数组。有关更多信息，请参阅 [创建 Amazon DocumentDB 集群参数组](#) (p. 327)。

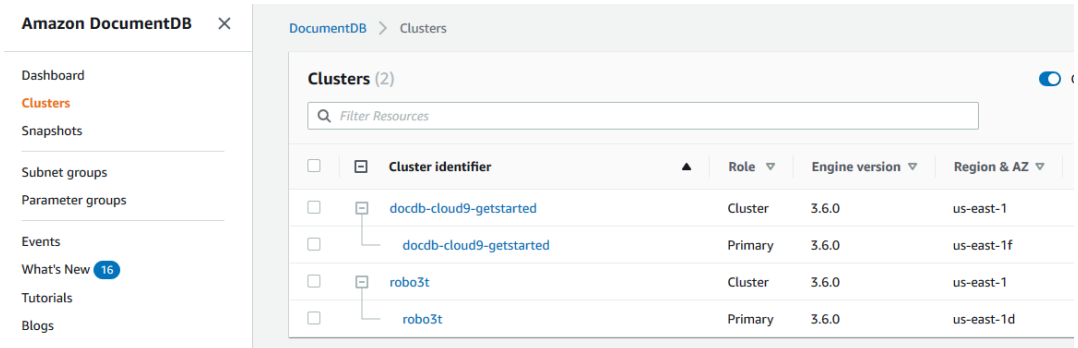
1. 确定集群使用的集群参数组。

- 从打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
- 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

- 请注意，在“群集”导航框中，列 Cluster Identifier 同时显示集群和实例。实例在集群下面列出。请参考下面的屏幕截图。



Cluster identifier	Role	Engine version	Region & AZ
docdb-cloud9-getstarted	Cluster	3.6.0	us-east-1
docdb-cloud9-getstarted	Primary	3.6.0	us-east-1f
robo3t	Cluster	3.6.0	us-east-1
robo3t	Primary	3.6.0	us-east-1d

- 选择您感兴趣的集群。
- 向下滚动到底部群详细信息然后找到集群参数组。请注意集群参数组的名称。

如果集群参数组的名称为 `default`（例如，`default.docdb3.6`），则您必须创建一个自定义集群参数组，并将其设置为集群的参数组，然后才能继续。有关更多信息，请参阅下列内容：

- [创建 Amazon DocumentDB 集群参数组](#) (p. 327) — 如果没有可以使用的自定义集群参数组，请创建一个。
- [修改 Amazon DocumentDB 集群](#) (p. 259) — 修改您的集群以使用自定义集群参数组。

2. 确定的当前值 `tls` 集群参数。

- 从打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
- 在导航窗格中，选择参数组。
- 在集群参数组列表中，选择您感兴趣的集群参数组的名称。
- 找到集群参数部分。在群集参数列表中，找到 `tls` 集群参数行。此时，以下四栏很重要：

- 集群参数名称— 集群参数的名称。对于管理 TLS，需要关注的是 `tls` 集群参数。
- 值— 每个集群参数的当前值。
- 允许的值— 可应用到集群参数的值的列表。
- 应用类型— 或者静态的要么动态。对静态集群参数的更改仅在重启实例时应用。对动态集群参数的更改可以立即应用，也可以在重启实例时应用。

3. 修改的值`tls`集群参数。

如果`tls`不是必需的，则为此集群参数组修改其值。更改的值`tls`集群参数，请按照以下步骤从前一部分继续执行。

- a. 选择集群参数名称 (`tls`) 左侧的按钮。
- b. 选择 Edit (编辑)。
- c. 更改 的值`tls`，在修改`tls`对话框中，请在下拉列表中为此集群参数选择所需的值。
- d. 选择 Modify cluster parameter (修改集群参数)。更改将在重启时应用到每个集群实例。

4. 重启 Amazon DocumentDB 实例。

重启集群的每个实例，以便将更改应用到集群中的所有实例。

- a. 从打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
- b. 在导航窗格中，选择 Instances (实例)。
- c. 要指定待重启的实例，请在实例列表中找到实例，并选择该实例名称左侧的按钮。
- d. 选择操作，然后重启。确认您要重启，请选择重启。

使用 Amazon CLI

请按照以下步骤执行 TLS 加密的管理任务，使用 Amazon CLI 例如识别参数组、验证 TLS 值以及进行必要的修改。

Note

除非您在创建集群时另行指定，否则将使用默认集群参数组创建集群。中的参数`default`无法修改集群参数组（例如，`tls`启用/禁用）。因此，如果您的集群使用`default`集群参数组，则需要修改集群来使用非默认集群参数组。您可能需要先创建自定义集群参数组。有关更多信息，请参阅[创建 Amazon DocumentDB 集群参数组](#) (p. 327)。

1. 确定集群使用的集群参数组。

使用`describe-db-clusters`带以下参数的命令：

- **`--db-cluster-identifier`** – 必需。所需集群的名称。
- **`--query`** – 可选。此查询将输出限制为仅显示感兴趣的字段，在本例中为集群名称及其集群参数组名称。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier docdb-2019-05-07-13-57-08 \
  --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[
  [
    "docdb-2019-05-07-13-57-08",
```

```
        "custom3-6-param-grp"  
    ]  
]
```

如果集群参数组的名称为default (例如, default.docdb3.6), 则您必须有一个自定义集群参数组, 并将其设置为集群的参数组, 然后才能继续。有关更多信息, 请参阅以下主题:

1. [创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#)— 如果没有可以使用的自定义集群参数组, 请创建一个。
 2. [修改Amazon DocumentDB 集群 \(p. 259\)](#)— 修改您的集群以使用自定义集群参数组。
2. 确定的当前值**tls**集群参数。

要获得有关此集群参数组的更多信息, 请使用describe-db-cluster-parameters按照以下参数调用操作:

- **--db-cluster-parameter-group-name** – 必需。使用上一个命令输出中的集群参数组名称。
- **--query** – 可选。此查询将输出限制为仅显示感兴趣的字段, 在本例中为ParameterName、ParameterValue、AllowedValues, 和ApplyType。

```
aws docdb describe-db-cluster-parameters \  
    --db-cluster-parameter-group-name custom3-6-param-grp \  
    --query 'Parameters[*].[ParameterName,ParameterValue,AllowedValues,ApplyType]'
```

此操作的输出将类似于下文 (JSON 格式)。

```
[  
  [  
    "audit_logs",  
    "disabled",  
    "enabled,disabled",  
    "dynamic"  
  ],  
  [  
    "tls",  
    "disabled",  
    "disabled,enabled",  
    "static"  
  ],  
  [  
    "ttl_monitor",  
    "enabled",  
    "disabled,enabled",  
    "dynamic"  
  ]  
]
```

3. 修改的值**tls**集群参数。

如果tls不是必需的, 则为此集群参数组修改其值。要更改 tls 集群参数的值, 请使用带以下参数的modify-db-cluster-parameter-group 操作。

- **--db-cluster-parameter-group-name** – 必需。要修改的集群参数组的名称。不能是default.* 集群参数组。
- **--parameters** – 必需。要修改的集群参数组的参数列表。
 - **ParameterName** – 必需。要修改的集群参数的名称。
 - **ParameterValue** – 必需。此集群参数的新值。必须是集群参数的 AllowedValues 之一。

- **enabled**— 集群仅接受使用 TLS 的安全连接。
- **disabled**— 集群不接受使用 TLS 的安全连接。
- **ApplyMethod**— 何时应用此修改。对于静态群集参数，例如 `tls`，此值必须为 `pending-reboot`。
- **pending-reboot**— 仅在重启后才将更改应用到实例。您必须分别重启每个集群实例，以便此更改在所有集群实例上生效。

以下代码禁用 `tls`，在重启时将更改应用到每个数据库实例。

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

以下代码启用 `tls`，在重启时将更改应用到每个数据库实例。

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters "ParameterName=tls,ParameterValue=enabled,ApplyMethod=pending-reboot"
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

4. 重启 Amazon DocumentDB 实例。

重启集群的每个实例，以便将更改应用到集群中的所有实例。要重启 Amazon DocumentDB 实例，请使用 `reboot-db-instance` 按照以下参数调用操作：

- **--db-instance-identifier** – 必需。要重启的实例的标识符。

以下代码将重启实例 `sample-db-instance`。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb reboot-db-instance \  
  --db-instance-identifier sample-db-instance
```

对于 Windows：

```
aws docdb reboot-db-instance ^  
  --db-instance-identifier sample-db-instance
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "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"
}
```

实例重启需要几分钟时间。只有在实例状态为 `available` (可用) 时，才能使用实例。您可以使用控制台或 Amazon CLI 监控实例状态。有关更多信息，请参阅[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

密钥管理

Amazon DocumentDB 使用 Amazon Key Management Service (Amazon KMS) 来检索和管理加密密钥。Amazon KMS 将安全、高度可用的硬件和软件结合起来，提供可扩展到云的密钥管理系统。利用 Amazon KMS，您可创建加密密钥并定义控制这些密钥的使用方式的策略。Amazon KMS 支持 Amazon CloudTrail，因此，您可审核密钥使用情况以验证密钥是否使用得当。

您的 Amazon KMS 您的密钥可与 Amazon DocumentDB 结合使用并支持 Amazon 服务如 Amazon Simple Storage Service (Amazon S3)、Amazon Relational Database Service (Amazon RDS)、Amazon Elastic Block Store (Amazon EBS) 和 Amazon Redshift。有关支持的服务的列表 Amazon KMS，请参阅[如何 Amazon 服务使用 Amazon KMS](#) 中的 Amazon Key Management Service 开发人员指南。有关 Amazon KMS 的信息，请参阅[什么是 Amazon Key Management Service ?](#)

Amazon DocumentDDB 中的 Identity and Access Management y

要管理集群、实例和集群参数组等 Amazon DocumentDB (兼容 MongoDB) 资源，需要具备以下证书 Amazon 可供用来验证您的请求。这些凭证必须有权访问 Amazon 资源，如 Amazon DocumentDB 实例。以下各部分提供有关可如何使用的详细信息 [Amazon Identity and Access Management \(IAM\)](#) 和 Amazon DocumentDB 通过控制可以访问资源的对象来保护您的资源。

主题

- [身份验证 \(p. 142\)](#)
- [管理您的 Amazon DocumentDB 资源的访问权限概述 \(p. 143\)](#)
- [使用策略管理访问 \(p. 146\)](#)
- [为 Amazon DocumentDocumentDB 使用基于身份的策略 \(IAM 策略\) \(p. 146\)](#)
- [Amazon DocumentDB API 权限 操作、资源和条件参考 \(p. 149\)](#)

身份验证

您可以以下面任一类型的身份访问 Amazon：

- Amazon Web Services 账户 根用户

当您创建 Amazon Web Services 账户时，最初使用的是一个对账户中所有 Amazon Web Services 和资源拥有完全访问权限的登录身份。此身份称为 Amazon Web Services 账户根用户，使用您创建账户时所用的电子邮件地址和密码登录，即可获得该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《Amazon 一般参考》中的[需要根用户凭证的任务](#)。

- IAM 用户和组

IAM 用户是 Amazon Web Services 账户内对某个人员或应用程序具有特定权限的一个身份。IAM 用户可能具有长期凭证，例如用户名和密码或一组访问密钥。要了解如何生成访问密钥，请参阅 IAM 用户指南中的[管理 IAM 用户的访问密钥](#)。为 IAM 用户生成访问密钥时，请确保查看并安全保存密钥对。您以后无法找回秘密访问密钥，而是必须生成新的访问密钥对。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM 用户指南中的[何时创建 IAM 用户（而不是角色）](#)。

- IAM 角色

IAM 角色是可在账户中创建的一种具有特定权限的 IAM 身份。IAM 角色类似于 IAM 用户，因为它是一个 Amazon 身份，具有确定其在 Amazon 中可执行和不可执行的操作的权限策略。但是，角色旨在让需要它的任何人代入，而不是唯一地与某个人员关联。此外，角色没有关联的标准长期凭证（如密码或访问密钥）。相反，当您代入角色时，它会为您提供角色会话的临时安全凭证。具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 您可以不创建 IAM 用户，而是使用来自 Amazon Directory Service、您的企业用户目录、Web 身份提供者或 IAM Identity Center 身份存储的现有身份。这些身份称为联合身份。要向联合身份分配权限，您可以创建角色并为角色定义权限。当外部身份进行身份验证时，该身份将与角色相关联并被授予其定义的权限。如果您使用 IAM Identity Center，则需要配置权限集。IAM Identity Center 将权限集与 IAM 中的角色相关联，以控制您的身份在进行身份验证后可以访问的内容。有关身份联合验证的更多信息，请参阅《IAM 用户指南》中的[为第三方身份提供商创建角色](#)。有关 IAM Identity Center 的更多信息，请参阅《Amazon IAM Identity Center (successor to Amazon Single Sign-On) 用户指南》中的[什么是 IAM Identity Center？](#)。
- Amazon Web Service 访问 – 服务角色是一个 **IAM 角色**，服务代入该角色以代表您执行操作。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅 IAM 用户指南中的[创建向 Amazon Web Service 委派权限的角色](#)。
- 在 Amazon EC2 上运行的应用程序 – 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 Amazon CLI 或 Amazon API 请求的应用程序的临时凭证。这优先于在 EC2 实例中存储访问密钥。要将 Amazon 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅 IAM 用户指南中的[使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

管理您的 Amazon DocumentDB 资源的访问权限概述

每种 Amazon 资源归某人所有 Amazon Web Services 账户，创建或访问这些资源的权限由权限策略控制。账户管理员可以向 IAM 身份（即：用户、组和角色）附加权限策略，某些服务（如 Amazon Lambda）也支持向资源附加权限策略。

Note

账户管理员（或管理员用户）是具有管理员权限的用户。有关更多信息，请参阅 IAM 用户指南中的[IAM 最佳实践](#)。

在授予权限时，您要决定谁获得权限，获得对哪些资源的权限，以及您允许对这些资源执行的具体操作。

主题

- [Amazon DocumentDB B 资源和运营](#) (p. 144)
- [了解资源所有权](#) (p. 144)
- [管理对资源的访问](#) (p. 144)
- [指定策略元素：操作、效果、资源和委托人](#) (p. 146)
- [在策略中指定条件](#) (p. 146)

Amazon DocumentDB B 资源和运营

在 Amazon DocumentDB 中，主要资源是聚集。Amazon DocumentDB 支持可与主要资源配合使用的其他资源，如实例、参数组，以及事件订阅。这些资源称作子资源。

这些资源和子资源具有与其关联的唯一 Amazon Resource Name (ARN)，如下表所示。

资源类型	ARN 格式
Cluster	arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i>
集群参数组	arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i>
集群快照	arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i>
实例	arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i>
安全组	arn:aws:rds: <i>region</i> : <i>account-id</i> :secgrp: <i>security-group-name</i>
子网组	arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i>

Amazon DocumentDB 提供了一组操作来使用 Amazon DocumentDB 资源。有关可用操作的列表，请参阅[操作](#)。

了解资源所有权

一个资源拥有者是 Amazon Web Services 账户它创建了资源。也就是说，资源所有者是 Amazon Web Services 账户的主要实体（根账户、IAM 用户或 IAM 角色），用于对创建资源的请求进行身份验证。以下示例说明了它的工作原理：

- 如果您使用的 root 账户凭证 Amazon Web Services 账户要创建 Amazon DocumentDB 资源，例如实例，您的 Amazon Web Services 账户是 Amazon DocumentDB B 资源的所有者。
- 如果您在中创建 IAM 用户 Amazon Web Services 账户并对该用户授予创建 Amazon DocumentDB 资源的权限，该用户也可以创建 Amazon DocumentDB 资源。但是，您的 Amazon Web Services 账户，该用户所属的 Amazon DocumentDB 资源。
- 如果您在中创建 IAM 角色 Amazon Web Services 账户拥有创建 Amazon DocumentDB 资源的权限，能够担任该角色的任何人都可以创建 Amazon DocumentDB 资源。您的 Amazon Web Services 账户该角色所属的 Amazon DocumentDB 资源。

管理对资源的访问

权限策略规定谁可以访问哪些内容。下一节介绍创建权限策略时的可用选项。

Note

本节讨论如何在 Amazon DocumentDB 范围内使用 IAM。这里不提供有关 IAM 服务的详细信息。有关完整的 IAM 文档，请参阅 IAM 用户指南中的[什么是 IAM ?](#)。有关 IAM policy 语法和说明的信息，请参阅[Amazon IAM 策略参考](#)在里面 IAM 用户指南。

附加到 IAM 身份的策略称作基于身份的策略 (IAM policy)。附加到资源的策略称作基于资源的策略。Amazon DocumentDB 只支持基于身份的策略 (IAM policy)。

主题

- [基于身份的策略 \(IAM 策略\) \(p. 145\)](#)
- [基于资源的策略 \(p. 146\)](#)

基于身份的策略 (IAM 策略)

您可以向 IAM 身份附加策略。例如，您可以执行以下操作：

- 将权限策略附加到账户中的用户或群组— 账户管理员可以使用与特定用户关联的权限策略为该用户授予创建 Amazon DocumentDB 资源（如实例）的权限。
- 向角色附加权限策略（授予跨账户权限）– 您可以向 IAM 角色附加基于身份的权限策略，以授予跨账户的权限。例如，账户 A 中的管理员可以创建一个角色，向其他账户授予跨账户权限 Amazon Web Services 账户（例如，账户 B）或 Amazon 服务如下：
 1. 账户 A 管理员可以创建一个 IAM 角色，然后向该角色附加授予其访问账户 A 中资源的权限策略。
 2. 账户 A 管理员可以把信任策略附加至用来标识账户 B 的角色，账户 B 由此可以作为主体代入该角色。
 3. 之后，账户 B 管理员可以委派权限，指定账户 B 中的任何用户担任该角色。这样，账户 B 中的用户就可以创建或访问账户 A 中的资源了。Amazon 如果您要授予权限，请向服务委托人 Amazon 服务来代入该角色。

有关使用 IAM 委托权限的更多信息，请参阅 IAM 用户指南中的[访问权限管理](#)。

以下是允许 ID 为的用户的示例策略 123456789012 为你创建实例 Amazon Web Services 账户。新实例必须使用以 default 开头的选项组和参数组，并且它必须使用 default 子网组。

```
{
  "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"
      ]
    }
  ]
}
```

有关对 Amazon DocumentDB 使用基于身份的策略的更多信息，请参阅[Amazon DocumentDB 使用基于身份的策略 \(IAM 策略\) \(p. 146\)](#)。有关用户、组、角色和权限的更多信息，请参阅 IAM 用户指南中的身份（用户、组和角色）。

基于资源的策略

其他服务（如 Amazon Simple Storage Service (Amazon S3)）也支持基于资源的权限策略。例如，您可以将策略附加到 Amazon S3 存储桶以管理对该存储桶的访问权限。Amazon DocumentDB 不支持基于资源的策略。

指定策略元素：操作、效果、资源和委托人

对于每个 Amazon DocumentDB 资源（请参阅 [Amazon DocumentDB B 资源和运营 \(p. 144\)](#)），该服务定义了一组 API 操作。有关更多信息，请参阅 [操作](#)。为授予这些 API 操作的权限，Amazon DocumentDB 定义了一组您可以在策略中指定的操作。执行一个 API 操作可能需要多个操作的权限。

以下是基本的策略元素：

- **Resource (资源)** - 在策略中，您可以使用 Amazon Resource Name (ARN) 标识策略应用到的资源。
- **操作** - 您可以使用操作关键字标识要允许或拒绝的资源操作。例如，`rds:DescribeDBInstances` 权限允许用户执行 `DescribeDBInstances` 操作。
- **Effect (效果)** - 您可以指定当用户请求特定操作（可以是允许或拒绝）时的效果。如果没有显式授予（允许）对资源的访问权限，则隐式拒绝访问。您也可显式拒绝对资源的访问，这样可确保用户无法访问该资源，即使有其他策略授予了访问权限的情况下也是如此。
- **主体** - 在基于身份的策略（IAM policy）中，附加了策略的用户是隐式主体。对于基于资源的策略，您可以指定要接收权限的用户、账户、服务或其他实体（仅适用于基于资源的策略）。Amazon DocumentDB 不支持基于资源的策略。

有关 IAM policy 语法和介绍的更多信息，请参阅 IAM 用户指南中的 [Amazon IAM policy 参考](#)。

有关显示所有 Amazon DocumentDB API 操作及它们适用的资源的表，请参阅 [Amazon DocumentDB API 权限 操作、资源和条件参考 \(p. 149\)](#)。

在策略中指定条件

当您授予权限时，可使用 IAM policy 语言来指定规定策略何时生效的条件。例如，您可能希望策略仅在特定日期后应用。有关使用策略语言指定条件的更多信息，请参阅 IAM 用户指南中的 [条件](#)。

要表示条件，您可以使用预定义的条件键。Amazon DocumentDB 没有可在 IAM 策略中使用的服务特定上下文键。有关适用于所有服务的全局条件上下文键列表，请参阅 [条件的可用密钥](#) 在里面 IAM 用户指南。

使用策略管理访问

您可以使用有效的凭证来对自己的请求进行身份验证，但如果您没有权限，则无法创建或访问 Amazon DocumentDB 资源。例如，您必须拥有权限才能创建 Amazon DocumentDB 集群、创建集群快照、修改集群参数组等。

下面几节介绍如何管理 Amazon DocumentDB 的权限。我们建议您先阅读概述。

- [管理您的 Amazon DocumentDB 资源的访问权限概述 \(p. 143\)](#)
- [为 Amazon DocumentDB 使用基于身份的策略 \(IAM 策略\) \(p. 146\)](#)

为 Amazon DocumentDB 使用基于身份的策略 (IAM 策略)

Important

对于某些管理功能，Amazon DocumentDB 使用与 Amazon RDS 共享的操作技术。Amazon DocumentDB 控制 Amazon CLI，并且 API 调用将记录为对 Amazon RDS API 的调用。

我们建议您首先阅读以下介绍性主题，这些主题讲解了管理 Amazon DocumentDB 资源访问的基本概念和选项。有关更多信息，请参阅 [管理您的 Amazon DocumentDB 资源的访问权限概述 \(p. 143\)](#)。

本主题提供了基于身份的策略的示例，在这些策略中，账户管理员可以向 IAM 身份（即：用户、组和角色）附加权限策略。

以下是 IAM policy 示例。

```
{
  "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"
      ]
    }
  ]
}
```

该策略包含一个为 IAM 用户指定以下权限的语句：

- 该策略允许 IAM 用户使用创建实例 `CreateDBInstance` 操作（这也适用于 `create-db-instance` Amazon CLI 操作和 Amazon Web Services Management Console）。
- `Resource` 元素指定用户可以执行操作的资源。使用 Amazon 资源名称 (ARN) 指定资源。此 ARN 包含资源所属服务的名称 (`rds`)，Amazon Web Services 区域 (* 表示本示例中的任何区域)、用户账号 (123456789012 是本示例中的用户 ID) 和资源的类型。

该示例中的 `Resource` 元素为用户指定有关资源的以下策略限制：

- 新实例的实例标识符必须以 `test` 开头（例如，`testCustomerData1`、`test-region2-data`）。
- 新实例的集群参数组必须以 `default` 开头。
- 新实例的子网组必须是 `default` 子网组。

该策略不指定 `Principal` 元素，因为在基于身份的策略中，您未指定获取权限的委托人。附加了策略的用户是隐式委托人。向 IAM 角色附加权限策略后，该角色的信任策略中标识的委托人将获取权限。

有关显示所有 Amazon DocumentDB API 操作及其适用于的资源列表，请参阅 [Amazon DocumentDB API 权限操作、资源和条件参考 \(p. 149\)](#)。

使用 Amazon DocumentDB 控制台所需的权限

对于要使用 Amazon DocumentDB 控制台的用户，该用户必须拥有一组最小权限。这些权限允许用户描述他们的 Amazon DocumentDB 资源 Amazon Web Services 账户并提供其他相关信息（包括 Amazon EC2 安全和网络信息）。

如果创建比必需的最低权限更为严格的 IAM policy，对于附加了该 IAM policy 的用户，控制台将无法按预期正常运行。要确保这些用户仍可使用 Amazon DocumentDB 控制台，也可附加 `AmazonDocDBConsoleFullAccess` 向用户提供托管策略，如中所述 [Amazon Amazon DocumentDB 的托管（预定义）策略 \(p. 148\)](#)。

对于只需要调用的用户，无需为其提供最低控制台权限Amazon CLI或者Amazon DocumentDB API。

AmazonAmazon DocumentDB 的托管 (预定义) 策略

Amazon通过提供由创建和管理的独立 IAM policy来满足许多常用案例的要求。Amazon托管式策略可授予常用案例的必要权限，因此，您可以免去调查都需要哪些权限的工作。有关更多信息，请参阅 IAM 用户指南中的[Amazon托管策略](#)。

以下Amazon托管策略 (可附加到您账户中的用户) 特定于 Amazon DocumentDB：

- AmazonDocDBReadOnlyAccess — 授予对根的所有 Amazon DocumentDB 资源的只读访问权Amazon Web Services 账户。
- AmazonDocDBFullAccess — 授予对根的所有 Amazon DocumentDB 资源的完全访问权限Amazon Web Services 账户。
- AmazonDocDBConsoleFullAccess — 授予使用 Amazon DocumentDB 资源的完全访问权限Amazon Web Services Management Console。

此外，您还可以创建自定义 IAM policy 以允许用户访问所需的 Amazon DocumentDB API 操作和资源。您可以将这些自定义策略附加到需要这些权限的 IAM 用户或组。

客户托管策略示例

本节的用户策略示例介绍如何授予对各 Amazon DocumentDB 操作的权限。当您使用 Amazon DocumentDB API 操作时，可以使用这些策略，Amazon软件开发工具包或Amazon CLI。当您使用控制台时，您需要授予特定于控制台的其他权限，使用[Amazon DocumentDB 控制台所需的权限 \(p. 147\)](#)中对此进行了讨论。

对于某些管理功能，Amazon DocumentDB 使用了与Amazon Relational Database Service (Amazon RDS) 和Amazon Neptune

Note

所有示例均使用美国东部 (弗吉尼亚北部) 区域 (us-east-1) 并包含虚构的账户 ID。

示例

- [示例 1：允许用户对任何 Amazon DocumentDB 资源执行任何描述操作 \(p. 148\)](#)
- [示例 2：防止用户删除实例 \(p. 149\)](#)
- [示例 3：除非启用存储加密，否则防止用户创建集群 \(p. 149\)](#)

示例 1：允许用户对任何 Amazon DocumentDB 资源执行任何描述操作

以下权限策略对用户授予权限以运行以 Describe 开头的所有操作。这些操作显示有关 Amazon DocumentDB 资源 (如实例) 的信息。中的通配符 (*)Resource元素表示可对账户拥有的所有 Amazon DocumentDB 资源执行操作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRDSDescribe",
      "Effect": "Allow",
      "Action": "rds:Describe*",
      "Resource": "*"
    }
  ]
}
```

示例 2：防止用户删除实例

以下权限策略授予权限以防止用户删除特定实例。例如，您可能想禁止任何非管理员用户删除您的生产实例。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyDelete1",
      "Effect": "Deny",
      "Action": "rds:DeleteDBInstance",
      "Resource": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
    }
  ]
}
```

示例 3：除非启用存储加密，否则防止用户创建集群

除非启用存储加密，否则以下权限策略拒绝授予用户创建 Amazon DocumentDB 集群的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventUnencryptedDocumentDB",
      "Effect": "Deny",
      "Action": "RDS:CreateDBCluster",
      "Condition": {
        "Bool": {
          "rds:StorageEncrypted": "false"
        }
      },
      "StringEquals": {
        "rds:DatabaseEngine": "docdb"
      }
    },
    {
      "Resource": "*"
    }
  ]
}
```

Amazon DocumentDB API 权限 操作、资源和条件参考

设置时，请使用以下各节作为参考[使用策略管理访问 \(p. 146\)](#)并编写可附加到 IAM 身份的权限策略（基于身份的策略）。

以下列出了每项 Amazon DocumentDB B API 操作。列表中包括您可授权执行该操作的相应操作，Amazon 您可以授予权限的资源以及您可以针对细粒度访问控制包含的条件键。您需要在策略的 `Action` 字段中指定操作、在策略的 `Resource` 字段中指定资源值、在策略的 `Condition` 字段中指定条件。有关条件的更多信息，请参阅[在策略中指定条件 \(p. 146\)](#)。

您可以使用 Amazon 在您的 Amazon DocumentDB 策略中使用范围的条件键来表达条件。有关 Amazon 范围内的键的完整列表，请参阅《IAM 用户指南》https://docs.amazonaws.cn/IAM/latest/UserGuide/reference_policies_elements.html#AvailableKeys中的可用键。

您可以使用 IAM policy simulator 测试 IAM 策略。它会自动提供资源列表和每个资源所需的参数 Amazon 操作，包括 Amazon DocumentDB B 操作。IAM 策略模拟器确定您指定的每项操作所需的权限。有关 IAM policy simulator 的信息，请参阅[使用 IAM 策略模拟器测试 IAM 策略](#)在里面 IAM 用户指南。

Note

要指定操作，请在 API 操作名称之前使用 `rds:` 前缀 (例如，`rds:CreateDBInstance`)。

下面列出了 Amazon RDS API 操作及其相关操作、资源和条件键。

主题

- [Support 资源级权限的Amazon DocumentDB B 操作 \(p. 150\)](#)
- [不Support 资源级权限的Amazon DocumentDB B 操作 \(p. 154\)](#)

Support 资源级权限的Amazon DocumentDB B 操作

资源级权限提供了指定允许用户对哪些资源执行操作的能力。Amazon DocumentDB 部分支持资源级权限。这意味着对于某些 Amazon DocumentDB 操作，您可以控制何时允许用户执行操作（基于必须满足的条件）或是允许用户使用的特定资源。例如，您可以向用户授予仅修改特定实例的权限。

以下列出Amazon DocumentDB API 操作及其相关操作、资源和条件键。

对于某些管理功能，Amazon DocumentDB 使用与 Amazon RDS 共享的操作技术。

Amazon DocumentDB API 操作和操作	资源	条件键
AddTagsToResource <code>rds:AddTagsToResource</code>	实例 <code>arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:db-tag</code>
	子网组 <code>arn:aws:rds:region:account-id:subgrp:subnet-group-name</code>	<code>rds:subgrp-tag</code>
ApplyPendingMaintenanceAction <code>rds:ApplyPendingMaintenanceAction</code>	实例 <code>arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:db-tag</code>
CopyDBClusterSnapshot <code>rds:CopyDBClusterSnapshot</code>	集群快照 <code>arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name</code>	<code>rds:cluster-snapshot-tag</code>
CreateDBCluster <code>rds:CreateDBCluster</code>	Cluster <code>arn:aws:rds:region:account-id:cluster:db-cluster-name</code>	<code>rds:cluster-tag</code>
	集群参数组 <code>arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
	子网组 <code>arn:aws:rds:region:account-id:subgrp:subnet-group-name</code>	<code>rds:subgrp-tag</code>

Amazon DocumentDB API 操作和操作	资源	条件键
创建数据库 ClusterParameterGroup rds:CreateDBClusterParameterGroup	集群参数组 arn:aws:rds:region:account-id:clusterpg:cluster-parameter-group-name	rds:cluster-pg-tag
创建数据库 ClusterSnapshot rds:CreateDBClusterSnapshot	Cluster arn:aws:rds:region:account-id:cluster:db-cluster-name	rds:cluster-tag
	集群快照 arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name	rds:cluster-snapshot-tag
CreateDBInstance rds:CreateDBInstance	实例 arn:aws:rds:region:account-id:db:db-instance-name	rds:DatabaseClass rds:db-tag
	Cluster arn:aws:rds:region:account-id:cluster:db-cluster-name	rds:cluster-tag
创建数据库SubnetGroup rds:CreateDBSubnetGroup	子网组 arn:aws:rds:region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
DeleteDBInstance rds:DeleteDBInstance	实例 arn:aws:rds:region:account-id:db:db-instance-name	rds:db-tag
deletedBSubnetGroup rds:DeleteDBSubnetGroup	子网组 arn:aws:rds:region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
describedBClusterParameterGroup rds:DescribeDBClusterParameterGroups	集群参数组 arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name	rds:cluster-pg-tag
describedBClusterParameterGroup rds:DescribeDBClusterParameters	集群参数组 arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name	rds:cluster-pg-tag

Amazon DocumentDB API 操作和操作	资源	条件键
DescribeDBClusters rds:DescribeDBClusters	Cluster arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-instance-name</i>	rds:cluster-tag
describeDBClusterSnapshots rds:DescribeDBClusterSnapshots	集群快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i>	rds:cluster-snapshot-tag
describeDBSubnetGroups rds:DescribeDBSubnetGroups	子网组 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i>	rds:subgrp-tag
DescribePendingMaintenanceActions rds:DescribePendingMaintenanceActions	实例 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i>	rds:DatabaseClass rds:db-tag
FailoverDBCluster rds:FailoverDBCluster	Cluster arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-instance-name</i>	rds:cluster-tag
ListTagsForResource rds:ListTagsForResource	实例 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i>	rds:db-tag
	子网组 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i>	rds:subgrp-tag
ModifyDBCluster rds:ModifyDBCluster	Cluster arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster: <i>db-cluster-name</i>	rds:cluster-tag
	集群参数组 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i>	rds:cluster-pg-tag
ModifyDBClusterParameterGroup rds:ModifyDBClusterParameterGroup	集群参数组 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i>	rds:cluster-pg-tag

Amazon DocumentDB API 操作和操作	资源	条件键
ModifyDBClusterSnapshotAttributes rds:ModifyDBClusterSnapshotAttributes	集群快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i>	rds:cluster-snapshot-tag
ModifyDBInstance rds:ModifyDBInstance	实例 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i>	rds:DatabaseClass rds:db-tag
RebootDBInstance rds:RebootDBInstance	实例 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i>	rds:db-tag
RemoveTagsFromResource rds:RemoveTagsFromResource	实例 arn:aws:rds: <i>region</i> : <i>account-id</i> :db: <i>db-instance-name</i>	rds:db-tag
	子网组 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i>	rds:subgrp-tag
resetDBClusterParameterGroup rds:ResetDBClusterParameterGroup	集群参数组 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-pg: <i>cluster-parameter-group-name</i>	rds:cluster-pg-tag
已恢复 DBClusterFromSnapshot rds:RestoreDBClusterFromSnapshot	Cluster arn:aws:rds: <i>region</i> : <i>account-id</i> :db-cluster: <i>db-cluster-instance-name</i>	rds:cluster-tag
	集群快照 arn:aws:rds: <i>region</i> : <i>account-id</i> :cluster-snapshot: <i>cluster-snapshot-name</i>	rds:cluster-snapshot-tag
已恢复 DBClusterToPointInTime rds:RestoreDBClusterToPointInTime	Cluster arn:aws:rds: <i>region</i> : <i>account-id</i> :db-cluster: <i>db-cluster-instance-name</i>	rds:cluster-tag
	子网组 arn:aws:rds: <i>region</i> : <i>account-id</i> :subgrp: <i>subnet-group-name</i>	rds:subgrp-tag

不支持资源级权限的Amazon DocumentDB B 操作

您可以使用 IAM 策略中的所有 Amazon DocumentDB 操作来授予或拒绝用户使用该操作的权限。但是，并非所有 Amazon DocumentDB 操作都支持资源级权限，这使您能够指定可以在哪些资源上执行操作。以下 Amazon DocumentDB API 操作目前不支持资源级权限。因此，要在 IAM 策略中使用这些操作，您必须授予用户使用所有资源的权限，方法是使用*的通配Resource您的陈述中的元素。

- `rds:DescribeDBClusterSnapshots`
- `rds:DescribeDBInstances`

Amazon DocumentDB 用户

在 Amazon DocumentDB 中，用户与密码一起使用才能通过集群的身份验证。每个集群都有一个在集群创建过程中建立的主用户和密码。

Note

所有在 2020 年 3 月 26 日之前创建的新用户都已被授予 `dbAdminAnyDatabase`、`readWriteAnyDatabase` 和 `clusterAdmin` 角色。建议您重新评估所有用户并根据需要修改角色，以便为集群中所有用户强制执行最低权限。有关更多信息，请参阅 [基于角色的访问控制限制数据库访问](#) (p. 156)。

主用户和 serviceadmin 用户

新创建的 Amazon DocumentDB 集群有两个用户：主用户和serviceadmin用户。

主用户 是单个特权用户，可以执行管理任务并创建具有角色的其他用户。当您首次连接到 Amazon DocumentDB 集群时，必须使用主用户名和密码进行身份验证。在创建 Amazon DocumentDB 集群时，主用户将获得该集群的这些管理权限，并被授予以下角色`root`。

在创建集群时会隐式创建 `serviceadmin` 用户。每个Amazon DocumentDB B 集群都有一个`serviceadmin`提供的用户Amazon管理集群的能力。您无法登录、删掉、重命名、更改密码，或者修改 `serviceadmin` 的权限。任何此类尝试都会导致错误。

Note

主`serviceadmin`无法删除 Amazon DocumentDB 集群的用户，其角色为`root`主用户的无法撤销。

如果您忘记了主用户密码，则可以使用 Amazon Web Services Management Console 或 Amazon CLI 重置密码。

创建其他用户

作为主用户（或具有 `createUser` 角色的任何用户）连接后，您可以创建新用户，如下所示。

```
db.createUser(
  {
    user: "sample-user-1",
    pwd: "password123",
    roles:
      [{"db":"admin", "role":"dbAdminAnyDatabase" }]
  }
)
```

要查看用户的详细信息，可以使用 `show users` 命令，如下所示。您还可以使用 `dropUser` 命令删除用户。有关更多信息，请参阅 [常见命令 \(p. 164\)](#)。

```
show users
{
  "_id" : "serviceadmin",
  "user" : "serviceadmin",
  "db" : "admin",
  "roles" : [
    {
      "role" : "root",
      "db" : "admin"
    }
  ]
},
{
  "_id" : "myMasterUser",
  "user" : "myMasterUser",
  "db" : "admin",
  "roles" : [
    {
      "role" : "root",
      "db" : "admin"
    }
  ]
},
{
  "_id" : "sample-user-1",
  "user" : "sample-user-1",
  "db" : "admin",
  "roles" : [
    {
      "role" : "dbAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
}
```

在上面的示例中，新用户 `sample-user-1` 受 `admin` 数据库的限制。对于新用户来说，情况总是如此。Amazon DocumentDB 没有 `authenticationDatabase` 因此，所有身份验证都是在下文中的 `admin` 数据库。

创建用户时，如果省略 `db` 字段在指定角色时，Amazon DocumentDB 会将该角色隐式归因于发出连接的数据库。例如，如果您的连接是针对数据库 `sample-database` 发出的，并且您运行以下命令，则用户 `sample-user-2` 将在 `admin` 数据库中创建并具有对数据库 `sample-database` 的 `readWrite` 权限。

```
db.createUser(
{
  user: "sample-user-2",
  pwd: "password123",
  roles:
    ["readWrite"]
}
)
```

创建具有跨所有数据库范围的角色用户（例如，`readInAnyDatabase`）需要在创建用户时处于 `admin` 数据库上下文，或者在创建用户时明确指定角色的数据库。

要切换数据库上下文，可以使用以下命令。

```
use admin
```

要了解有关基于角色的访问控制以及对集群用户强制执行最低权限的更多信息，请参阅[基于角色的访问控制限制数据库访问](#) (p. 156)。

自动轮换Amazon DocumentDB B 的密码

借助 Amazon Secrets Manager，您可以将代码中的硬编码凭证（包括密码）替换为对 Secrets Manager 的 API 调用，从而以编程方式检索密钥。这有助于确保检查您的代码的人不会泄露密钥，因为其中根本不包含密钥。此外，您还可以配置 Secrets Manager 以根据指定的计划自动轮换密钥。这使您能够将长期密钥替换为短期密钥，这有助于显著减少泄露风险。

使用 Secrets Manager，您可以自动轮换您的Amazon DocumentDB B 密码（即秘密）使用Amazon LambdaSecrets Manager 提供的功能。

有关更多信息Amazon Secrets Manager以及与 Amazon DocumentDB 的本地集成，请参阅以下内容：

- [博客：如何轮换Amazon DocumentDB B 和Amazon Redshift 证书AmazonSecrets Manager](#)
- [什么是AmazonSecrets Manager？](#)
- [轮换 Amazon DocumentDB 的密钥](#)

基于角色的访问控制限制数据库访问

您可以使用以下方式限制对用户可对数据库执行的操作的访问基于角色的访问控制Amazon DocumentDB 中的（RBAC）（与 MongoDB 兼容）。RBAC 的原理是向用户授予一个或多个角色。这些角色决定了用户可以对数据库资源执行的操作。Amazon DocumentDB 目前支持数据库级别范围内的两个内置角色，例如：`read`、`readWrite`、`readAnyDatabase`、`clusterAdmin`，以及用户定义的角色，这些角色可以限定为特定操作以及基于您的要求的精细资源（例如集合）。

RBAC 的常见使用案例包括通过创建对集群中数据库或集合具有只读访问权限的用户来强制执行最低权限，以及允许单个用户访问集群中给定数据库或集合的多租户应用程序设计。

Note

所有在 2020 年 3 月 26 日之前创建的新用户都已被授予 `dbAdminAnyDatabase`、`readWriteAnyDatabase` 和 `clusterAdmin` 角色。建议您重新评估所有现有用户并根据需要修改角色，以便为您的集群强制执行最低权限。

主题

- [RBAC 概念](#) (p. 156)
- [RBAC 内置角色入门](#) (p. 158)
- [RBAC 用户定义的角色入门](#) (p. 160)
- [以用户身份连接到 Amazon DocumentDB](#) (p. 163)
- [常见命令](#) (p. 164)
- [功能差异](#) (p. 167)
- [限制](#) (p. 167)
- [基于角色的访问控制限制数据库访问](#) (p. 168)

RBAC 概念

以下是与基于角色的访问控制相关的重要术语和概念。有关 Amazon DocumentDB 用户的更多信息，请参阅[Amazon DocumentDB 用户](#) (p. 154)。

- 用户— 可以对数据库进行身份验证并执行操作的单个实体。
- 密码— 用于对用户进行身份验证的密钥。
- 角色— 授权用户对一个或多个数据库执行操作。
- 管理数据库— 用户在其中存储和授权的数据库。
- 数据库 (**db**) — 集群中包含用于存储文档的集合的命名空间。

以下命令创建名为 `sample-user` 的用户。

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

在此示例中：

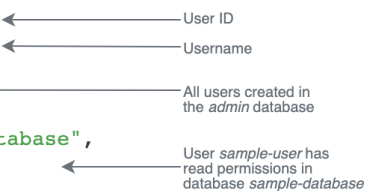
- `user: "sample-user"`— 表示用户名。
- `pwd: "abc123"`— 表示用户密码。
- `role: "read", db: "sample-database"`— 表示用户 `sample-user` 将在中具有读取权限 `sample-database`。



```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

以下示例显示您为用户 `sample-user` 指定 `db.getUser(sample-user)` 后的输出。在此示例中，用户 `sample-user` 驻留在 `admin` 数据库中，但具有 `sample-database` 数据库的读取角色。

```
{
  "_id" : "sample-user",
  "user" : "sample-user",
  "db" : "admin",
  "roles" : [
    {
      "db" : "sample-database",
      "role" : "read"
    }
  ]
}
```



创建用户时，如果省略 `db` 在字段中，Amazon DocumentDB 会隐式地将该角色归属于发出连接的数据库。例如，如果您的连接是针对数据库 `sample-database` 发出的，并且您运行以下命令，则用户 `sample-user` 将在 `admin` 数据库中创建并具有对数据库 `sample-database` 的 `readWrite` 权限。

```
db.createUser({user: "sample-user", pwd: "abc123", roles: ["readWrite"]})
```

此操作的输出将类似于下文。

```
{
  "user": "sample-user",
  "roles": [
    {
      "db": "sample-database",
      "role": "readWrite"
    }
  ]
}
```

创建具有跨所有数据库范围的角色用户（例如，`readAnyDatabase`）需要在创建用户时处于 `admin` 数据库的上下文，或者在创建用户时明确指定角色的数据库。要针对 `admin` 数据库发出命令，可以使用命令 `use admin`。有关更多信息，请参阅[常见命令](#) (p. 164)。

RBAC 内置角色入门

为帮助您掌握基于角色的访问控制，本部分将向您介绍通过为三个用户创建具有不同职能的角色来强制执行最低权限的示例方案。

- `user1` 是一位新上任的经理，需要能够查看和访问集群中的所有数据库。
- `user2` 是一位新员工，只需访问同一集群中的一个数据库 `sample-database-1`。
- `user3` 是现有员工，需要查看和访问同一集群中其以前无权访问的不同数据库 `sample-database-2`。

在这之后，`user1` 和 `user2` 都离开了公司，所以必须撤销他们的访问权限。

要创建用户和授予角色，进入集群用于进行身份验证的用户必须具有可以为 `createUser` 和 `grantRole` 执行操作的关联角色。例如，角色 `admin` 和 `userAdminAnyDatabase` 都可以授予执行上述功能的权限。有关每个角色的操作，请参阅[基于角色的访问控制限制数据库访问](#) (p. 168)。

Note

在 Amazon DocumentDB 中，所有用户和角色操作（例如，`create`、`get`、`drop`、`grant`、`revoke`等）隐式执行在 `admin` 数据库无论您是否针对 `admin` 数据库。

首先，要了解集群中的当前用户和角色，可以运行 `show users` 命令，如下示例所示。您将看到两个用户，即集群的 `serviceadmin` 和主用户。这两个用户始终存在，无法将它们删除。有关更多信息，请参阅[Amazon DocumentDB 用户](#) (p. 154)。

```
show users
```

对于 `user1`，使用以下命令创建具有对整个集群中所有数据库的读写访问权限的角色。

```
db.createUser({user: "user1", pwd: "abc123", roles: [{role: "readWriteAnyDatabase", db: "admin"}]})
```

此操作的输出将类似于下文。

```
{
  "user": "user1",
  "roles": [
    {
      "role": "readWriteAnyDatabase",
      "db": "admin"
    }
  ]
}
```

对于 `user2`，使用以下命令创建对数据库 `sample-database-1` 具有只读访问权限的角色。

```
db.createUser({user: "user2", pwd: "abc123", roles: [{role: "read", db: "sample-database-1"}]})
```

此操作的输出将类似于下文。

```
{
  "user": "user2",
```

```
    "roles":[
      {
        "role":"read",
        "db":"sample-database-1"
      }
    ]
  }
}
```

要模拟 user3 是现有用户的方案，请先创建用户 user3，然后将新角色分配给 user3。

```
db.createUser({user: "user3", pwd: "abc123", roles: [{role: "readWrite", db: "sample-
database-1"}]})
```

此操作的输出将类似于下文。

```
{
  "user":"user3",
  "roles":[
    {
      "role":"readWrite",
      "db":"sample-database-1"
    }
  ]
}
```

现在已创建了用户 user3，请为 user3 分配可对 sample-database-2 进行 read 的角色。

```
db.grantRolesToUser("user3", [{role: "read", db: "sample-database-2"}])
```

最后，user1 和 user2 都离开了公司，需要撤销他们对集群的访问权限。您可以通过删除用户来执行此操作，如下所示。

```
db.dropUser("user1")
db.dropUser("user2")
```

要确保所有用户都具有适当角色，可使用以下命令列出所有用户。

```
show users
```

此操作的输出将类似于下文。

```
{
  "_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"
    }
  ]
}
```

RBAC 用户定义的角色入门

为帮助您掌握用户定义的角色，本部分将向您介绍通过为三个用户创建具有不同职能的角色来强制执行最低权限的示例方案。

在此示例中，以下条件适用：

- user1 是一位新上任的经理，需要能够查看和访问集群中的所有数据库。
- user2 是一位新员工，只需要对一个数据库执行“查找”操作，sample-database-1，在同一个集群中。
- user3 是现有员工，需要查看和访问不同数据库中的特定集合，col2，sample-database-2 在同一集群中，他们以前无权访问。
- 对于 user1，使用以下命令创建具有对整个集群中所有数据库的读写访问权限的角色。

```
db.createUser(
{
  user: "user1", pwd: "abc123",
  roles: [{role: "readWriteAnyDatabase", db: "admin"}]
}
```

此操作的输出将类似于下文。

```
{
  "user": "user1",
  "roles": [
    {
      "role": "readWriteAnyDatabase",
      "db": "admin"
    }
  ]
}
```

适用于 user2，创建一个对数据库中所有集合具有“查找”权限的角色 sample-database-1 使用以下命令。请注意，此角色将确保所有关联的用户只能运行查找查询。

```
db.createRole(
{
  role: "findRole",
```

```
    privileges: [
      {
        resource: {db: "sample-database-1", collection: ""}, actions: ["find"]
      },
      roles: []
    ]
  }
}
```

此操作的输出将类似于下文。

```
{
  "role": "findRole",
  "privileges": [
    {
      "resource": {
        "db": "sample-database-1",
        "collection": ""
      },
      "actions": [
        "find"
      ]
    }
  ],
  "roles": [
  ]
}
```

下一步，创建用户 (user2) 并附加最近创建的角色findRole给用户。

```
db.createUser(
{
  user: "user2",
  pwd: "abc123",
  roles: []
})

db.grantRolesToUser("user2", ["findRole"])
```

为了模拟那个场景user3是现有用户，首先创建用户user3，然后创建一个名为 CollectionRole 的新角色，我们将在下一步中分配给该角色user3。

现在你可以将新角色分配给user3。这个新角色将允许user3以便能够在中插入、更新、删除和查找对一个特定集合的访问权限 col2sample-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: []
  ]
}
```

```
)
```

此操作的输出将类似于下文。

```
{
  "role": "collectionRole",
  "privileges": [
    {
      "resource": {
        "db": "sample-database-2",
        "collection": "col2"
      },
      "actions": [
        "find",
        "update",
        "insert",
        "remove"
      ]
    }
  ],
  "roles": [
  ]
}
```

现在用户user3已创建，您可以授予user3角色collectionFind。

```
db.grantRolesToUser("user3", ["collectionRole"])
```

最后，user1 和 user2 都离开了公司，需要撤销他们对集群的访问权限。您可以通过删除用户来执行此操作，如下所示。

```
db.dropUser("user1")
db.dropUser("user2")
```

要确保所有用户都具有适当角色，可使用以下命令列出所有用户。

```
show users
```

此操作的输出将类似于下文。

```
{
  "_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"
      }
    ]
  }
}
```

以用户身份连接到 Amazon DocumentDB

在连接到 Amazon DocumentDB 集群时，需要在特定数据库的上下文中进行连接。默认情况下，如果未在连接字符串中指定数据库，则会在 `test` 数据库上下文中自动连接到集群。所有集合级别命令（如 `insert` 和 `find`）都是针对 `test` 数据库中的集合发出的。

要查看您所在上下文或者换句话说对其发出命令的数据库，请使用 `db` 命令在 mongo shell 中执行，如下所示。

查询：

```
db
```

输出：

```
test
```

尽管默认连接可能位于 `test` 数据库的上下文中，但这并不一定意味着与连接关联的用户有权对 `test` 数据库执行操作。在上述示例方案中，如果您用于进行身份验证的用户 `user3` 是具有对 `sample-database-1` 数据库的 `readWrite` 权限的角色，则连接的默认上下文是 `test` 数据库。但是，如果您尝试将文档插入 `test` 数据库中的集合中，您将收到 `Authorization failure`（授权失败）错误消息。这是因为该用户未获得对该数据库执行该命令的授权，如下所示。

查询：

```
db
```

输出：

```
test
```

查询：

```
db.col.insert({x:1})
```

输出：

```
WriteCommandError({ "ok" : 0, "code" : 13, "errmsg" : "Authorization failure" })
```

如果您将连接上下文更改为 `sample-database-1` 数据库，则可以写入到集合中，因为该用户具有执行此操作的授权。

查询：

```
use sample-database-1
```

输出：

```
switched to db sample-database-1
```

查询：

```
db.col.insert({x:1})
```

输出：

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

使用特定用户对集群进行身份验证时，还可以在连接字符串中指定数据库。如果这样做，则用户在通过 admin 数据库的身份验证后，不再需要执行 use 命令。

以下连接字符串根据 admin 数据库对用户进行身份验证，但连接上下文则是 sample-database-1 数据库。

```
mongo "mongodb://user3:abc123@sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/sample-database-2"
```

常见命令

本部分提供在 Amazon DocumentDB 中使用基于角色的访问控制的常见命令示例。您必须位于 admin 数据库的上下文中，才能创建和修改用户和角色。您可以使用 use admin 命令切换到 admin 数据库。

Note

对用户和角色的修改将隐式发生在 admin 数据库中。创建具有跨所有数据库范围的角色用户（例如，readAnyDatabase）需要在创建用户时处于 admin 数据库上下文（即 use admin），或者在创建用户时明确指定角色的数据库（如本节的示例 2 所示）。

示例 1：使用创建用户 read 数据库的角色 foo。

```
db.createUser({user: "readInFooBar", pwd: "abc123", roles: [{role: "read", db: "foo"}]})
```

此操作的输出将类似于下文。

```
{
  "user": "readInFooBar",
  "roles": [
    {
      "role": "read",
      "db": "foo"
    }
  ]
}
```

示例 2：创建具有所有数据库的读取访问权限的用户。

```
db.createUser({user: "readAllDBs", pwd: "abc123", roles: [{role: "readAnyDatabase", db: "admin"}]})
```

此操作的输出将类似于下文。

```
{
  "user": "readAllDBs",
  "roles": [
    {
      "role": "readAnyDatabase",
      "db": "admin"
    }
  ]
}
```

示例 3：Grantread角色到新数据库中现有用户。

```
db.grantRolesToUser("readInFooBar", [{role: "read", db: "bar"}])
```

示例 4：更新用户的角色。

```
db.updateUser("readInFooBar", {roles: [{role: "read", db: "foo"}, {role: "read", db:
  "baz"}]})
```

示例 5：撤销用户对数据库的访问权限。

```
db.revokeRolesFromUser("readInFooBar", [{role: "read", db: "baz"}])
```

示例 6：描述内置角色。

```
db.getRole("read", {showPrivileges:true})
```

此操作的输出将类似于下文。

```
{
  "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"
    ]
}
}
```

示例 7 示例：从集群中删除用户。

```
db.dropUser("readInFooBar")
```

此操作的输出将类似于下文。

```
true
```

示例 8：创建具有特定集合的读写访问权限的角色

```
db.createRole(
{
  role: "collectionRole",
  privileges: [
    {
      resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
"update", "insert", "remove"]
    },
    roles: []
  ]
})
```

此操作的输出将类似于下文。

```
{
  "role":"collectionRole",
  "privileges":[
    {
      "resource":{"
        "db":"sample-database-2",
        "collection":"col2"
      },
      "actions":[
        "find",
        "update",
        "insert",
        "remove"
      ]
    }
  ],
  "roles":[]
}
```

示例 9.：创建用户并分配用户定义的角色

```
db.createUser(
{
  user: "user3",
  pwd: "abc123",
  roles: []
})

db.grantRolesToUser("user3",["collectionRole"])
```

示例 10.：向用户定义的角色授予额外权限

```
db.grantPrivilegesToRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col1" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

示例 11 示例：从用户定义的角色中删除权限

```
db.revokePrivilegesFromRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col2" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

示例 12.：更新现有用户定义的角色

```
db.updateRole(
  "collectionRole",
  {
    privileges: [
      {
        resource: {db: "sample-database-3", collection: "sample-collection-3"}, actions:
["find", "update", "insert", "remove"]
      },
    ],
    roles: []
  }
)
```

功能差异

在 Amazon DocumentDB 中，用户和角色定义存储在admin数据库和用户将根据admin数据库。此功能与 MongoDB 社区版不同，但与 MongoDB Atlas 一致。

Amazon DocumentDB 还支持变更流，该功能提供按时间顺序排列的更改事件，这些事件在您的集群集合内发生。这些区域有：listChangeStreams操作在集群级别（即跨所有数据库）应用，而modifyChangeStreams操作可以在数据库级别和群集级别应用。

限制

下表包含 Amazon DocumentDB 中基于角色的访问控制的限制。

描述	限制
每个集群的用户数	1000
与用户关联的角色数	1000
用户定义的角色数	100
与权限关联的资源数	100

基于角色的访问控制限制数据库访问

借助基于角色的访问控制，您可以创建一个用户并向其授予一个或多个角色，以确定该用户可以在数据库或集群中执行哪些操作。

以下是 Amazon DocumentDB 中目前支持的内置角色的列表。

Note

在 Amazon DocumentDB 4.0 中，`ListCollection`和`ListDatabase`命令可以选择使用`authorizedCollections`和`authorizedDatabases`用于列出用户有权访问的集合和数据库的参数`listCollections`和`listDatabase`角色，分别。此外，用户现在可以杀死自己的光标，而无需`KillCursor`角色。

角色类型	角色名称	描述	操作
数据库用户	<code>read</code>	授予用户对指定数据库的读取权限。	<code>changeStreams</code> <code>collStats</code> <code>dbStats</code> <code>find</code> <code>killCursors</code> <code>listIndexes</code> <code>listCollections</code>
	<code>readWrite</code>	授予用户对指定数据库的读取和写入访问权限。	使用 <code>read</code> 权限的所有操作。 <code>createCollection</code> <code>dropCollection</code> <code>createIndex</code> <code>dropIndex</code> <code>insert</code> <code>killCursors</code> <code>listIndexes</code> <code>listCollections</code>

角色类型	角色名称	描述	操作
集群用户			remove update
	readAnyDatabase	授予用户对集群中所有数据库的读取权限。	使用 read 权限的所有操作。 listChangeStreams listDatabases
	readWriteAnyDatabase	授予用户对集群中所有数据库的读写权限。	使用 readWrite 权限的所有操作。 listChangeStreams listDatabases
	userAdminAnyDatabase	授予用户为所有用户分配或修改对指定数据库的角色或权限的能力。	changeCustomData changePassword createUser dropRole dropUser grantRole listDatabases revokeRole viewRole viewUser
	dbAdminAnyDatabase	授予用户对所有指定数据库执行数据库管理角色的能力。	使用 dbAdmin 权限的所有操作。 dropCollection listDatabases listChangeStreams modifyChangeStreams
Superuser	root	授予用户对以下所有角色的合并资源和操作的访问权限： readWriteAnyDatabase、dbAdminAnyDatabase、和 backup。	使用 readWriteAnyDatabase、dbAdminAnyDatabase、和 backup 的所有操作。

角色类型	角色名称	描述	操作
数据库管理	dbAdmin	授予用户对指定数据库执行管理任务的能力。	collMod collStats createCollection createIndex dropCollection dropDatabase dropIndex dbStats find killCursors listIndexes listCollections modifyChangeStreams
	dbOwner	通过合并角色 dbAdmin 和 readWrite 授予用户对指定数据库执行任何管理任务的能力。	使用 dbAdmin 和 readWrite 的所有操作。
集群管理	clusterAdmin	通过合并 clusterManager、clusterMonitor 和 hostManager 角色授予用户最大的集群管理访问权限。	使用 clusterManager、clusterMonitor 和 hostManager 的所有操作。 listChangeStreams dropDatabase modifyChangeStreams
	clusterManager	授予用户对指定集群执行管理和监控操作的能力。	listChangeStreams listSessions modifyChangeStreams replSetGetConfig

角色类型	角色名称	描述	操作
	clusterMonitor	授予用户对监控工具可具有只读访问权限的能力。	collStats dbStats find getParameter hostInfo indexStats killCursors listChangeStreams listCollections listDatabases listIndexes listSessions replSetGetConfig serverStatus top
	hostManager	授予用户监视和管理服务器的能力。	killCursors killAnyCursor killAnySession killop
备份管理	backup	授予用户备份数据所需的访问权限。	getParameter insert find listChangeStreams listCollections listDatabases listIndexes update

角色类型	角色名称	描述	操作
	restore	授予用户还原数据所需的访问权限。	<code>changeCustomData</code> <code>changePassword</code> <code>collMod</code> <code>createCollection</code> <code>createIndex</code> <code>createUser</code> <code>dropCollection</code> <code>dropRole</code> <code>dropUser</code> <code>getParameter</code> <code>grantRole</code> <code>find</code> <code>insert</code> <code>listCollections</code> <code>modifyChangeStreams</code> <code>revokeRole</code> <code>remove</code> <code>viewRole</code> <code>viewUser</code> <code>update</code>

Amazon DocumentDB 中的日志记录和监控

Amazon DocumentDB (与 MongoDB 兼容) 提供了一些 Amazon CloudWatch 指标，您可以监控这些指标以确定 Amazon DocumentDB 集群和实例的运行状况和性能。您可以使用各种工具查看 Amazon DocumentDB 指标，其中包括 Amazon DocumentDB 控制台、Amazon CLI、Amazon CloudWatch 控制台和 CloudWatch API。有关监控的更多信息，请参阅[监控 Amazon DocumentDB \(p. 365\)](#)。

除 Amazon CloudWatch 指标之外，您可以使用分析器来记录在您集群上执行的操作的执行时间和详细信息。对于监控集群上速度最慢的操作以帮助提高单个查询的性能和整体集群性能，分析器非常有用。一旦启用，操作将记录到 Amazon CloudWatch Logs 中，并且您可以使用 CloudWatch Insight 来分析、监控和存档 Amazon DocumentDB 分析数据。有关更多信息，请参阅[分析 Amazon DocumentDB 操作 \(p. 388\)](#)。

Amazon DocumentDB 也与集成 Amazon CloudTrail 服务，提供 IAM 用户、IAM 角色或 Amazon Amazon DocumentDB 中的服务 (与 MongoDB 兼容)。CloudTrail 捕获所有 Amazon CLI 将 Amazon DocumentDB 作为事件调用，包括来自 Amazon DocumentDB 的调用 Amazon Web Services Management Console 也从对 Amazon DocumentDB SDK 的代码调用。有关更多信息，请参阅[使用记录 Amazon DocumentDB API 调用 Amazon CloudTrail \(p. 387\)](#)。

使用 Amazon DocumentDB，您可以审核在您集群中执行的事件。记录的事件的示例包括成功和失败的身份验证尝试、删除数据库中的集合或创建索引。默认情况下，Amazon DocumentDB 上的审核处于禁用状态，并要求您选择使用此功能。有关更多信息，请参阅[审计 Amazon DocumentDB \(p. 185\)](#)。

更新您的 Amazon DocumentDB TLS 证书 (cn-northwest-1)

将更新为 Amazon DocumentDB (CA) 证书（具有 MongoDB 兼容性）集群的证书。2022 年 6 月 1 日，如果您使用启用了传输层安全 (TLS)（默认设置）的 Amazon DocumentDB 集群，而且您没有轮换您的客户端应用程序和服务器证书，则需要执行以下步骤以避免您的应用程序与 Amazon DocumentDB 集群之间发生连接性问题。

作为 Amazon DocumentDB 的标准维护和安全最佳实践的一部分，CA 和服务器证书已更新。之前的 CA 证书将于 2022 年 6 月 1 日到期。客户端应用程序必须将新的 CA 证书添加到其信任存储中，且现有 Amazon DocumentDB 实例必须更新为在此到期日期之前使用新的 CA 证书。

更新您的应用程序和 Amazon DocumentDB 集群

按照此部分中的步骤更新应用程序的 CA 证书捆绑包（[步骤 1](#)），以及您集群的服务器证书（[步骤 2](#)）。在将变更应用于生产环境之前，我们强烈建议您在开发环境或登台环境中测试这些步骤。

Note

必须在每个步骤中完成步骤 1 和 2。Amazon Web Services 区域您在其中拥有 Amazon DocumentDB 集群。

第 1 步：下载新的 CA 证书并更新申请

下载新的 CA 证书并更新您的应用程序，以便使用新的 CA 证书创建与 Amazon DocumentDB 的 TLS 连接。从 <https://rds-truststore.s3.cn-north-1.amazonaws.com.cn/cn-northwest-1/cn-northwest-1-bundle.pem> 下载新的 CA 证书捆绑包。此操作将下载名为 cn-northwest-1-bundle.pem 的文件。

```
wget https://rds-truststore.s3.cn-north-1.amazonaws.com.cn/cn-northwest-1/cn-northwest-1-bundle.pem
```

接下来，更新应用程序以使用新的证书捆绑包。新的 CA 捆绑包中包含旧的 CA 证书 (rds-ca-2017-root) 以及新的 CA 证书 (rds-ca-rsa2048-g1.pem)。通过在新的 CA 捆绑包中同时包含这两个 CA 证书，您可以通过两个步骤来更新应用程序和集群。

在 2021 年 12 月 21 日之后，任何下载 CA 证书捆绑包的操作都应使用新的 CA 证书捆绑包。要验证您的应用程序使用的是否是最新的 CA 证书捆绑包，请参阅[我如何确定我使用的是最新的 CA 捆绑包？ \(p. 177\)](#)。如果您已在应用程序中使用最新的 CA 证书捆绑包，则可跳至步骤 2。

有关将 CA 捆绑与您的应用程序结合使用的示例，请参阅[加密传输中的数据 \(p. 136\)](#)和[启用了 TLS 的情况下的连接 \(p. 430\)](#)。

Note

目前，MongoDB Go 驱动程序 1.2.1 仅在中接受一个 CA 服务器证书。sslcertificateauthorityfile。有关如何在启用 TLS 时使用 Go 连接到 Amazon DocumentDB，请参阅[启用了 TLS 的情况下的连接 \(p. 430\)](#)。

第 2 步：更新服务器证书

在更新应用程序以使用新的 CA 捆绑包后，下一步是通过修改 Amazon DocumentDB 集群中的每个实例来更新服务器证书。要修改实例以使用新的服务器证书，请参阅以下说明。

Note

更新您的实例需要重新启动，这可能会导致服务中断。在更新服务器证书之前，请确保您已完成步骤 1。

使用 Amazon Web Services Management Console

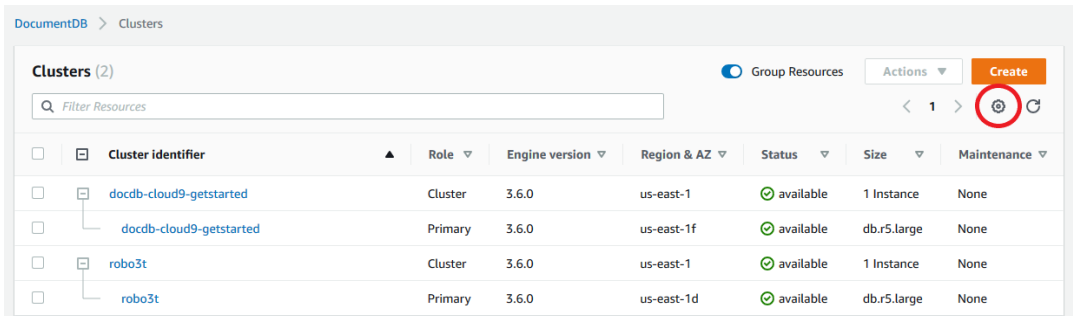
完成以下步骤，以使用为现有 Amazon DocumentDB 实例标识和轮换旧服务器证书。Amazon Web Services Management Console.

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>.
2. 在屏幕右上角的“区域”列表中，选择Amazon Web Services 区域你的集群所在的地方。
3. WH

在控制台左侧的导航窗格中，选择集群。

4. 您可能需要识别哪些实例仍在使用旧的服务器证书 (rds-ca-2017)。您可以在中执行此操作。证书颁发机构默认情况下处于隐藏状态的列。要显示 Certificate authority (证书颁发机构) 列，请执行以下操作：

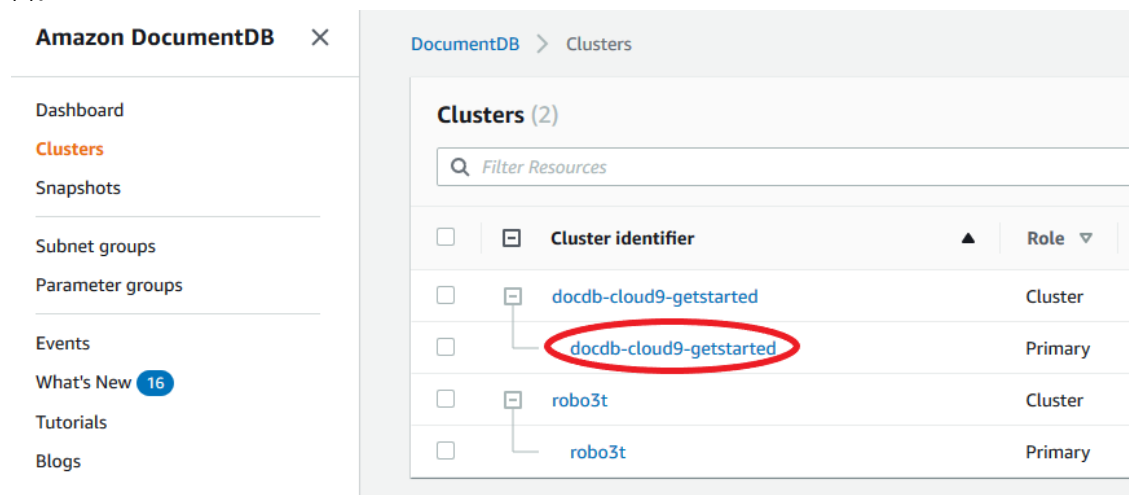
- a. 选择 Settings (设置) 选项卡。



- b. 在可见列列表下，选择证书颁发机构column.

- c. 选择 Confirm (确认) 以保存所做的更改。

5. 现在在 Cluster 导航框中，您将看到该列。Cluster Identifier. 您的实例列在群集下，类似于下面的屏幕截图。



6. 选中您感兴趣的实例左侧的框。
7. 选择 Actions (操作)，然后选择 Modify (修改)。
8. UNDER证书颁发机构中，选择新的服务器证书 (rds-ca-rsa2048-g1) 对于这个例子。

9. 在下一页上可以看到所做更改的摘要。请注意，有额外的警报要提醒您，在修改实例之前，确保应用程序使用最新的证书 CA 捆绑包，以避免导致连接中断。
10. 可以选择在下一个维护时段内应用修改，也可以立即应用。如果您打算立即修改服务器证书，请使用 Apply Immediately (立即应用) 选项。
11. 选择修改实例以完成更新。

使用 Amazon CLI

完成以下步骤，以使用为现有 Amazon DocumentDB 实例标识和轮换旧服务器证书。Amazon CLI。

1. 要立即修改实例，请对集群中的每个实例执行以下命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. 要修改集群中的实例，以便在集群的下一个维护时段中使用新的 CA 证书，请对集群中的每个实例执行以下命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

常见问题

以下是有关 TLS 证书的一些常见问题的答案。

如果我有疑问或问题，应该怎么办？

如果您有疑问或疑问，请联系。[Amazon Web Services Support](#)。

我如何知道我是否使用 TLS 连接到我的 Amazon DocumentDB 集群？

您可以通过检查 `tls` 参数用于集群的集群参数组。如果 `tls` 参数设置为 `enabled`，您正在使用 TLS 证书连接到集群。有关更多信息，请参阅 [管理 Amazon DocumentDB 集群参数组 \(p. 322\)](#)。

为什么要更新 CA 和服务器证书？

作为 Amazon DocumentDB 的标准维护和安全最佳实践的一部分，Amazon DocumentDB CA 和服务器证书已更新。当前 CA 和服务器证书将在 2022 年 6 月 1 日到期。

如果我在到期日之前没有采取任何行动，会怎么样？

如果您正在使用 TLS 连接到 Amazon DocumentDB 集群，并且您未使用以下方式进行更改：2022 年 6 月 1 日，通过 TLS 连接的应用程序将无法再与 Amazon DocumentDB 集群进行通信。

到期之前，Amazon DocumentDB 将不会自动轮换您的数据库证书。您必须在到期日期之前或之后更新应用程序和集群以使用新的 CA 证书。

如何知道哪些 Amazon DocumentDB 实例正在使用旧/新的服务器证书？

要识别仍在使用旧服务器证书的 Amazon DocumentDB 实例，您可以使用 Amazon DocumentDB Amazon Web Services Management Console 或者 Amazon CLI。

使用 Amazon Web Services Management Console

要识别集群中正在使用旧证书的实例，请执行以下操作：

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在屏幕右上角的“区域”列表中，选择Amazon Web Services 区域您的实例所在的位置。
3. 在控制台左侧的导航窗格中，选择 Instances (实例)。
4. 这些区域有：证书颁发机构列（默认情况下处于隐藏状态）显示哪些实例仍在使用旧的服务器证书 (rds-ca-2017) 和新的服务器证书 (rds-ca-rsa2048-g1)。要显示 Certificate authority (证书颁发机构) 列，请执行以下操作：
 - a. 选择 Settings (设置) 选项卡。
 - b. 在可见列列表下，选择证书颁发机构column.
 - c. 选择 Confirm (确认) 以保存所做的更改。

使用 Amazon CLI

要识别集群中正在使用旧服务器证书的实例，请使用describe-db-clusters命令包含以下内容。

```
aws docdb describe-db-instances \
  --filters Name=engine,Values=docdb \
  --query 'DBInstances[*]'.
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

如何修改 Amazon DocumentDB 集群中的单个实例以更新服务器证书？

建议您在给定集群中同时更新所有实例的服务器证书。要修改集群中的实例，可以使用控制台或Amazon CLI。

Note

更新您的实例需要重新启动，这可能会导致服务中断。在更新服务器证书之前，请确保您已完成[步骤 1](#)。

使用 Amazon Web Services Management Console

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在屏幕右上角的“区域”列表中，选择Amazon Web Services 区域你的集群所在的地方。
3. 在控制台左侧的导航窗格中，选择 Instances (实例)。
4. Certificate authority (证书颁发机构) 列（默认情况下处于隐藏状态）会显示哪些实例仍在使用旧的服务器证书 (rds-ca-2017)。要显示 Certificate authority (证书颁发机构) 列，请执行以下操作：
 - a. 选择 Settings (设置) 选项卡。
 - b. 在可见列列表下，选择证书颁发机构column.
 - c. 选择 Confirm (确认) 以保存所做的更改。
5. 选择要修改的实例。
6. 选择 Actions (操作)，然后选择 Modify (修改)。
7. UNDER证书颁发机构中，选择新的服务器证书 (rds-ca-rsa2048-g1)) 对于这个例子。

8. 在下一页上可以看到所做更改的摘要。请注意，有额外的警报要提醒您，在修改实例之前，确保应用程序使用最新的证书 CA 捆绑包，以避免导致连接中断。
9. 可以选择在下一个维护时段内应用修改，也可以立即应用。
10. 选择修改实例以完成更新。

使用 Amazon CLI

完成以下步骤，以使用为现有 Amazon DocumentDB 实例标识和轮换旧服务器证书。Amazon CLI。

1. 要立即修改实例，请对集群中的每个实例执行以下命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. 要修改集群中的实例，以便在集群的下一个维护时段中使用新的 CA 证书，请对集群中的每个实例执行以下命令。

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

如果我向现有集群中添加新的实例，会怎么样？

如果我的集群发生实例替换或故障转移，会怎么样？

如果集群中有实例替换，则创建的新实例将继续使用该实例之前使用的服务器证书。我们建议您同时更新所有实例的服务器证书。如果群集中发生故障转移，则使用新主服务器上的服务器证书。

如果我未使用 TLS 连接到集群，是否仍需要更新每个实例？

如果您未使用 TLS 连接到 Amazon DocumentDB 集群，则不需要执行操作。

如果我目前没有使用 TLS 连接到集群，但计划将来这样做，该怎么办？

如果您在 2022 年 4 月 4 日之前创建了集群，请按照[步骤 1](#)和[步骤 2](#)以确保您的应用程序使用的是更新过的 CA 捆绑包，且每个 Amazon DocumentDB 实例使用的都是最新的服务器证书。如果您的集群是在 2022 年 4 月 4 日之后创建的，那么您的集群已经具有最新的服务器证书。要验证您的应用程序使用的是否是最新的 CA 捆绑包，请参阅[如果我未使用 TLS 连接到集群，是否仍需要更新每个实例？](#) (p. 177)

截止日期可以延长到 2022 年 6 月 1 日之后吗？

如果您的应用程序通过 TLS 连接，截止日期不能延长到 2022 年 6 月 1 日以后。

我如何确定我使用的是最新的 CA 捆绑包？

出于兼容性原因，旧的和新的 CA 捆绑包文件的文件名都为 `rds-combined-ca-bundle.pem`。你也可以使用类似的工具 `openssl` 或 `keytool` 来检查 CA 捆绑包。

为什么我在 CA 捆绑包的名称中看到“RDS”？

对于某些管理功能，如证书管理，Amazon DocumentDB 使用与 Amazon Ration Database Service (Amazon RDS) 共享的操作技术。

新的服务器证书将在格林威治标准时间 2024 年 8 月 22 日到期。

如果我应用了新的服务器证书，我可以恢复为使用旧的服务器证书吗？

如果您需要将实例恢复为使用旧的服务器证书，建议您对集群中的所有实例都执行此操作。您可以通过使用 Amazon Web Services Management Console 或 Amazon CLI 来还原集群中每个实例的服务器证书。

使用 Amazon Web Services Management Console

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在屏幕右上角的“区域”列表中，选择 Amazon Web Services 区域您的集群所在的地方。
3. 在控制台左侧的导航窗格中，选择 Instances (实例)。
4. 选择要修改的实例。选择 Actions (操作)，然后选择 Modify (修改)。
- 5.
6. 选择 Continue (继续) 以查看修改摘要。
7. 在显示的页面中，您可以选择安排在下一个维护时段中应用修改，或立即应用修改。进行选择，然后选择修改实例。

Note

如果您选择立即应用修改，则该操作也将同时应用等待修改队列中的所有更改。如果任何待处理修改需要停机，选择此选项可导致意外停机。

使用 Amazon CLI

如果选择 `--no-apply-immediately`，所做的更改将在集群的下一个维护时段内应用。

如果我从快照还原或执行时间点还原，它会有新的服务器证书吗？

如果您在 2022 年 4 月 4 日之后还原快照或执行时间点还原，则创建的新集群将使用新 CA 证书。

如果我在从 Mac OS X Catalina 直接连接到我的 Amazon DocumentDB 集群时遇到问题，该怎么办？

Mac OS X Catalina 已更新对可信证书的要求。受信任证书现在必须有效期为 825 天或更短（请参阅 <https://support.apple.com/en-us/HT210176>）。Amazon DocumentDB 实例证书的有效期是四年以上，已超过 Mac OS X 的最大值。要从运行 Mac OS X Catalina 的计算机直接连接到 Amazon DocumentDB 集群，您必须在创建 TLS 连接时允许使用无效证书。在这种情况下，无效证书是指其有效期超过 825 天。在连接到 Amazon DocumentDB 集群时，您应在允许使用无效证书之前了解风险。

要从 OS X Catalina 连接到 Amazon DocumentDB 集群，请使用 Amazon CLI，使用 `tlsAllowInvalidCertificates` 参数。

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017 --  
tlsAllowInvalidCertificates
```

Note

此信息仅适用于 GovCloud (美国西部) 区域的用户。

更新您的应用程序和 Amazon DocumentDB 集群

按照此部分中的步骤更新应用程序的 CA 证书捆绑包（[步骤 1](#)），以及您集群的服务器证书（[步骤 2](#)）。在将变更应用于生产环境之前，我们强烈建议您在开发环境或登台环境中测试这些步骤。

Note

您必须在每个步骤中完成步骤 1 和 2 步。Amazon Web Services 区域其中有 Amazon DocumentDB 集群。

第 1 步：下载新 CA 证书并更新应用程序

有关将 CA 捆绑与您的应用程序结合使用的示例，请参阅 [加密传输中的数据 \(p. 136\)](#) 和 [启用了 TLS 的情况下的连接 \(p. 430\)](#)。

Note

目前，MongoDB Go 驱动程序 1.2.1 仅接受在中的一个 CA 服务器证书。sslcertificateauthorityfile. 有关如何在启用 TLS 时使用 Go 连接到 Amazon DocumentDB，请参阅[启用了 TLS 的情况下的连接 \(p. 430\)](#)。

第 2 步：更新服务器证书

在更新应用程序以使用新的 CA 捆绑包后，下一步是通过修改 Amazon DocumentDB 集群中每个实例来更新服务器证书。要修改实例以使用新的服务器证书，请参阅以下说明。

Note

更新您的实例需要重新启动，这可能会导致服务中断。在更新服务器证书之前，请确保您已完成[步骤 1](#)。

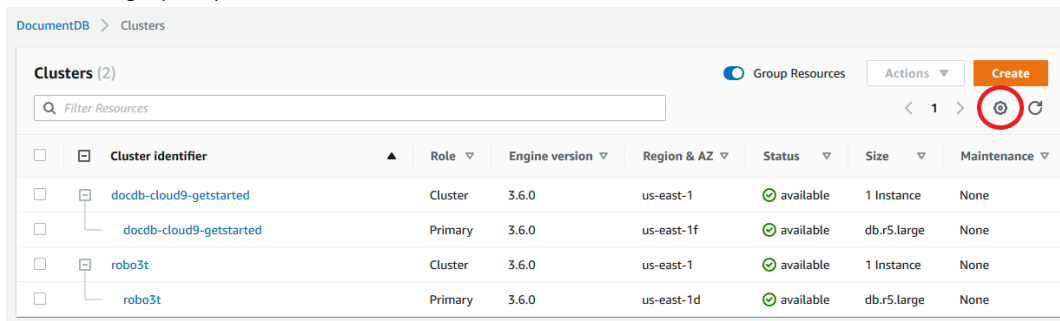
使用 Amazon Web Services Management Console

完成以下步骤，以使用为现有 Amazon DocumentDB 实例标识和轮换旧服务器证书。Amazon Web Services Management Console.

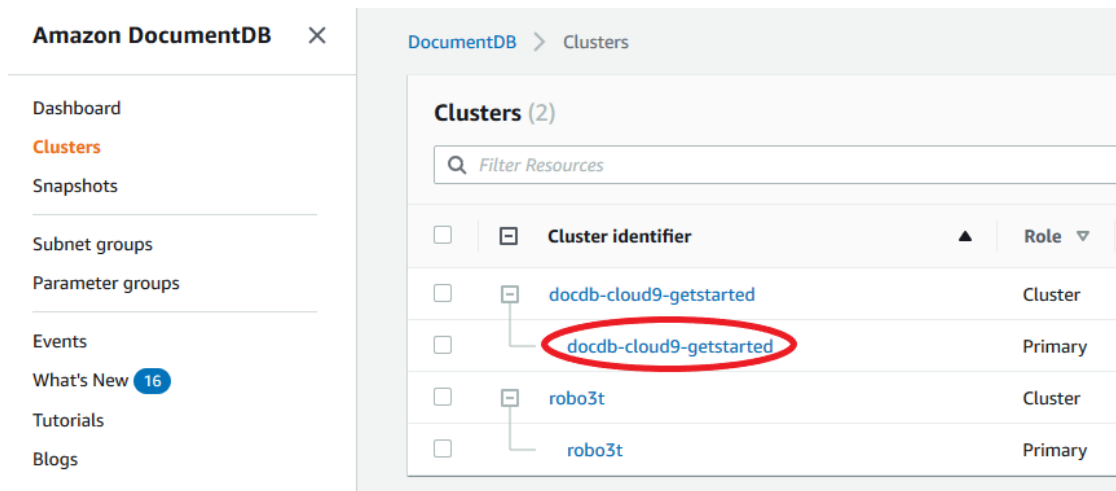
1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>.
2. 在屏幕右上角的“区域”列表中，选择Amazon Web Services 区域集群所在的位置。
3. WH

在控制台左侧的导航窗格中，选择集群。

4. 您可能需要识别哪些实例仍在使用旧的服务器证书 (rds-ca-2017)。您可以在证书颁发机构默认情况下处于隐藏状态的列。要显示 Certificate authority (证书颁发机构) 列，请执行以下操作：
 - a. 选择 Settings (设置) 选项卡。



- b. 在可见列列表下，选择证书颁发机构column.
 - c. 选择 Confirm (确认) 以保存所做的更改。
5. 现在，在 Cluster 导航框中，您将看到该列。Cluster Identifier. 您的实例列在群集下，类似于下面的屏幕截图。



6. 选中您感兴趣的实例左侧的框。
7. 选择 Actions (操作)，然后选择 Modify (修改)。
8. 在下一页上可以看到所做更改的摘要。请注意，有额外的警报要提醒您，在修改实例之前，确保应用程序使用最新的证书 CA 捆绑包，以避免导致连接中断。
9. 可以选择在下一个维护时段内应用修改，也可以立即应用。如果您打算立即修改服务器证书，请使用 Apply Immediately (立即应用) 选项。
10. 选择修改实例以完成更新。

使用 Amazon CLI

完成以下步骤，以使用为现有 Amazon DocumentDB 实例标识和轮换旧服务器证书。Amazon CLI。

1. 要立即修改实例，请对集群中的每个实例执行下面的命令。
2. 要修改集群中的实例，以便在集群的下一个维护时段中使用新的 CA 证书，请对集群中的每个实例执行以下命令。

常见问题

以下是有关 TLS 证书的一些常见问题的答案。

如果我有疑问或问题，应该怎么办？

如果您有任何疑问或疑问，请联系 [Amazon Web Services Support](#)。

我如何知道我是否使用 TLS 连接到我的 Amazon DocumentDB 集群？

您可以通过检查 `tls` 参数集群的集群参数组。如果 `tls` 参数设置为 `enabled`，您正在使用 TLS 证书连接到集群。有关更多信息，请参阅 [管理 Amazon DocumentDB 集群参数组 \(p. 322\)](#)。

为什么要更新 CA 和服务器证书？

如果我在到期日之前没有采取任何行动，会怎么样？

如果您使用 TLS 连接到 Amazon DocumentDB 集群，但在 2022 年 5 月 18 日之前没有进行更改，则通过 TLS 连接的应用程序将无法再与 Amazon DocumentDB 集群通信。

在过期之前，Amazon DocumentDB 将不会自动轮换您的数据库证书。您必须在到期日期之前或之后更新应用程序和集群以使用新的 CA 证书。

如何知道我的哪些 Amazon DocumentDB 实例正在使用旧/新的服务器证书？

要识别仍使用旧服务器证书的 Amazon DocumentDB 实例，您可以使用 Amazon DocumentDB Amazon Web Services Management Console 或者 Amazon CLI。

使用 Amazon Web Services Management Console

要识别集群中正在使用旧证书的实例，

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在屏幕右上角的“区域”列表中，选择 Amazon Web Services 区域实例所在的位置。
3. 在控制台左侧的导航窗格中，选择 Instances (实例)。
4.
 - a. 选择 Settings (设置) 选项卡。
 - b. 在可见列列表下，选择证书颁发机构 column。
 - c. 选择 Confirm (确认) 以保存所做的更改。

使用 Amazon CLI

要识别集群中正在使用旧服务器证书的实例，请使用 `describe-db-clusters` 命令如下。

```
aws docdb describe-db-instances \
  --filters Name=engine,Values=docdb \
  --query 'DBInstances[*]'.
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

如何修改 Amazon DocumentDB 集群中的单个实例以更新服务器证书？

我们建议您同时更新给定集群中所有实例的服务器证书。要修改集群中的实例，您可以使用控制台或 Amazon CLI。

Note

更新您的实例需要重新启动，这可能会导致服务中断。在更新服务器证书之前，请确保您已完成 [步骤 1](#)。

使用 Amazon Web Services Management Console

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在屏幕右上角的“区域”列表中，选择 Amazon Web Services 区域集群所在的位置。
3. 在控制台左侧的导航窗格中，选择 Instances (实例)。
4. Certificate authority (证书颁发机构) 列 (默认情况下处于隐藏状态) 会显示哪些实例仍在使用旧的服务器证书 (rds-ca-2017)。要显示 Certificate authority (证书颁发机构) 列，请执行以下操作：
 - a. 选择 Settings (设置) 选项卡。
 - b. 在可见列列表下，选择证书颁发机构 column。
 - c. 选择 Confirm (确认) 以保存所做的更改。

5. 选择要修改的实例。
6. 选择 Actions (操作), 然后选择 Modify (修改)。
- 7.
8. 在下一页上可以看到所做更改的摘要。请注意, 有额外的警报要提醒您, 在修改实例之前, 确保应用程序使用最新的证书 CA 捆绑包, 以避免导致连接中断。
9. 可以选择在下一个维护时段内应用修改, 也可以立即应用。
10. 选择修改实例以完成更新。

使用 Amazon CLI

完成以下步骤, 以使用为现有 Amazon DocumentDB 实例标识和轮换旧服务器证书。Amazon CLI.

1. 要立即修改实例, 请对集群中的每个实例执行下面的命令。
2. 要修改集群中的实例, 以便在集群的下一个维护时段中使用新的 CA 证书, 请对集群中的每个实例执行以下命令。

如果我向现有集群中添加新的实例, 会怎么样?

如果我的集群发生实例替换或故障转移, 会怎么样?

如果集群中有实例替换, 则创建的新实例将继续使用该实例之前使用的服务器证书。我们建议您同时更新所有实例的服务器证书。如果群集中发生故障转移, 则使用新主服务器上的服务器证书。

如果我目前没有使用 TLS 连接到集群, 是否仍需要更新每个实例?

如果您未使用 TLS 连接到 Amazon DocumentDB 集群, 则不需要执行操作。

如果我目前没有使用 TLS 连接到集群, 但计划将来这样做, 该怎么办?

我如何确定我使用的是最新的 CA 捆绑包?

出于兼容性原因, 旧的和新的 CA 捆绑包文件的文件名都为 us-gov-west-1-bundle.pem。你也可以使用类似的工具 openssl 或 keytool 来检查 CA 捆绑包。

为什么我在 CA 捆绑包的名称中看到“RDS”?

对于某些管理功能, 例如证书管理, Amazon DocumentDB 使用与 Amazon Ration Database Service (Amazon RDS) 共享的操作技术。

如果我应用了新的服务器证书, 我可以恢复为使用旧的服务器证书吗?

如果您需要将实例恢复为使用旧的服务器证书, 建议您对集群中的所有实例都执行此操作。您可以通过使用 Amazon Web Services Management Console 或 Amazon CLI 来还原集群中每个实例的服务器证书。

使用 Amazon Web Services Management Console

1. 登录到 Amazon Web Services Management Console, 然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>.

2. 在屏幕右上角的“区域”列表中，选择Amazon Web Services 区域集群所在的位置。
3. 在控制台左侧的导航窗格中，选择 Instances (实例)。
4. 选择要修改的实例。选择 Actions (操作)，然后选择 Modify (修改)。
- 5.
6. 选择 Continue (继续) 以查看修改摘要。
7. 在显示的页面中，您可以选择安排在下一个维护时段中应用修改，或立即应用修改。进行选择，然后选择修改实例。

Note

如果您选择立即应用修改，则该操作也将同时应用等待修改队列中的所有更改。如果任何待处理修改需要停机，选择此选项可导致意外停机。

使用 Amazon CLI

如果选择`--no-apply-immediately`，将在集群的下一个维护时段内应用所做的更改。

如果我从快照还原或执行时间点还原，它会有新的服务器证书吗？

如果我在从 Mac OS X Catalina 直接连接到我的 Amazon DocumentDB 集群时遇到问题，该怎么办？

Mac OS X Catalina 已更新对可信证书的要求。受信任证书现在必须有效期为 825 天或更短（请参阅<https://support.apple.com/en-us/HT210176>）。Amazon DocumentDB 实例证书的有效期是四年以上，超过 Mac OS X 的最大值。要从运行 Mac OS X Catalina 的计算机直接连接到 Amazon DocumentDB 集群，您必须在创建 TLS 连接时允许使用无效证书。在这种情况下，无效证书是指其有效期超过 825 天。在连接到 Amazon DocumentDB 集群时，您应在允许使用无效证书之前了解风险。

要使用从 OS X Catalina 连接到 Amazon DocumentDB 集群，请使用Amazon CLI，使用`tlsAllowInvalidCertificates`参数。

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017 --  
tlsAllowInvalidCertificates
```

Amazon DocumentDB 中的合规性验证

作为多个项目的一部分，第三方审计员将评估 Amazon DocumentDB 的安全性和合规性（具有 MongoDB 兼容性）Amazon合规计划，包括以下内容：

- 系统和组织控制 (SOC) 1、2 和 3。有关更多信息，请参阅 [SOC](#)。
- 支付卡行业数据安全标准 (PCI DSS)。有关更多信息，请参阅 [PCI DSS](#)。
- ISO 9001、27001、27017 和 27018。有关更多信息，请参阅 [ISO 认证](#)。
- 《健康保险流通与责任法案》商业伙伴协议 (HIPAA BAA)。有关更多信息，请参阅 [HIPAA 合规性](#)。

Amazon 在[合规性计划范围内的 Amazon 服务](#)中提供特定合规性计划范围内经常更新的 Amazon 服务列表。

提供第三方审计报告，可使用 Amazon Artifact 下载。有关更多信息，请参阅[下载 Amazon Artifact 中的报告](#)。

有关 Amazon 合规性计划的更多信息，请参阅 [Amazon 合规性计划](#)。

您在使用 Amazon DocumentDB 时的合规性责任由您数据的敏感性、您组织的合规性目标以及适用的法律法规决定。如果您对 Amazon DocumentDB 的使用需遵守 HIPAA 或 PCI 等标准，Amazon提供帮助的资源：

- [Amazon 合规性资源](#) - 可能适用于您的行业和所在地的业务手册和指南集合。
- [安全性与合规性 Quick Start 指南](#) - 部署指南讨论架构注意事项，提供在 Amazon 部署侧重安全性和合规性的基准环境的步骤。
- [AmazonConfig](#) - 一项服务，可评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- [Amazon Security Hub](#) - Amazon 安全状态的全面视图，可帮助检查是否符合安全行业标准和最佳实践。
- [HIPAA 安全性和合规性架构设计白皮书](#) - 介绍公司如何使用的白皮书Amazon以创建符合 HIPAA 要求的应用程序。

Amazon DocumentDB 中的恢复功能

Amazon全球基础设施围绕Amazon Web Services 区域和可用区构建。Amazon Web Services 区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

只能在 Amazon VPC 中创建 Amazon DocumentDB 集群，该集群在至少两个可用区中包含至少两个子网。Amazon DocumentDB 通过跨至少两个可用区分配您的集群实例，可帮助确保集群中有可用的实例，避免出现可用区故障。Amazon DocumentDB 集群的集群卷始终跨三个可用区提供持久性存储，数据丢失的可能性很小。

有关 Amazon Web Services 区域和可用区的更多信息，请参阅 [Amazon 全球基础设施](#)。

除了AmazonAmazon DocumentDB 提供了多种功能，以帮助支持您的数据弹性和备份需求。

具有容错和自我修复能力的存储

存储卷的每 10 GB 部分在三个可用区之间的 6 个方向进行复制。Amazon DocumentDB 使用容错存储，可透明地处理最多两个数据副本的丢失，而不影响数据库写入可用性，还使用多达三个副本，避免影响读取可用性。Amazon DocumentDB 存储也是自我修复的；数据块和磁盘会持续扫描是否存在错误并自动更换。

手动备份和还原

Amazon DocumentDB 提供了创建集群的完整备份以实现长期保留和恢复的功能。有关更多信息，请参阅在 [Amazon DocumentDB 中备份和还原 \(p. 194\)](#)。

时间点恢复

时间点恢复有助于保护 Amazon DocumentDB 集群免遭意外写入或删除操作。使用时间点恢复，您不必担心创建、维护或计划按需备份。有关更多信息，请参阅[还原到某个时间点 \(p. 217\)](#)。

Amazon DocumentDB 中的基础设施安全性

作为托管服务，Amazon DocumentDB 受Amazon中描述的全局网络安全程序[Amazon Web Services : 安全过程概述](#)白皮书。

你使用Amazon可以使用发布的 API 调用通过网络访问 Amazon DocumentDB。客户端必须支持 TLS (传输层安全性) 1.0。建议使用 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。大多数现代系统 (如 Java 7 及更高版本) 都支持这些模式。此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用[AmazonSecurity Token Service](#)(AmazonSTS) 生成临时安全凭证来对请求进行签名。

您可以从任何网络位置调用这些 API 操作，但 Amazon DocumentDB 不支持基于资源的访问策略，其中可以包含基于源 IP 地址的限制。您还可以使用 Amazon DocumentDB 策略来控制来自特定 Amazon Virtual Private Cloud (Amazon VPC) 终端节点或特定 VPC 的访问。事实上，这隔离了在中仅从一个特定 VPC 到给定 Amazon DocumentDB 资源的网络访问。Amazon网络。

Amazon DocumentDB 的安全最佳实践

出于安全考虑，您必须使用 Amazon Identity and Access Management 使用 IAM 账户可控制对 Amazon DocumentDB API 操作（特别是创建、修改或删除 Amazon DocumentDB 资源的操作）的访问。此类资源包括集群、安全组和参数组。此外，您还必须使用 IAM 来控制执行常见管理任务的操作，例如备份和还原集群。创建 IAM 角色时，请采用最小权限原则。

- 使用[基于角色的访问控制](#) (p. 156) 强制执行最低权限。
- 为每个管理 Amazon DocumentDB 资源的人员分配个人 IAM 账户。请勿使用 Amazon Web Services 账户根用户来管理 Amazon DocumentDB 资源。为每个人（包括您自己）创建一个 IAM 用户。
- 授予每位用户履行其职责所需的最小权限集。
- 使用 IAM 组有效地管理适用于多个用户的权限。有关 IAM 的更多信息，请参阅[IAM 用户指南](#)。有关 IAM 最佳实践的信息，请参阅[IAM 最佳实践](#)。
- 定期轮换 IAM 凭证。
- 配置 Amazon Secrets Manager 可自动轮换 Amazon DocumentDB 的密钥。有关更多信息，请参阅[旋转 Amazon Secrets Manager 密钥和轮换 Amazon DocumentDB 的密钥](#)中的 Amazon Secrets Manager 用户指南。
- 使用传输层安全性 (TLS) 和静态加密来加密您的数据。

审计 Amazon DocumentDB

使用 Amazon DocumentDB (with Amazon DocumentDB (with MongoDB))，您可以审核在集群中执行的事件。记录的事件的示例包括成功和失败的身份验证尝试、删除数据库中的集合或创建索引。默认情况下，在 Amazon DocumentDB 上禁用审核，并要求您选择使用该功能。

启用审核后，Amazon DocumentDB 会将数据定义语言 (DDL)、数据处理语言 (DML)、身份验证、授权和用户管理事件记录到 Amazon CloudWatch 日志。启用审核后，Amazon DocumentDB 会将集群的审核记录（JSON 文档）导出到 Amazon CloudWatch 日志。您可以使用亚马逊 CloudWatch 分析、监控和存档您的 Amazon DocumentDB 审核事件的日志。

尽管 Amazon DocumentDB 不对启用审核额外收费，但您需要为使用 CloudWatch 日志。了解有关的信息 CloudWatch 日志，请参阅[亚马逊 CloudWatch 定价](#)。

Amazon DocumentDB 审计功能与使用监控的服务资源使用情况有明显的不同 Amazon CloudTrail。CloudTrail 记录使用执行的操作 Amazon Command Line Interface (Amazon CLI) 或 Amazon Web Services Management Console 在集群、实例、参数组和快照等资源上。审计 Amazon 资源与 CloudTrail 默认处于启用状态，并且无法禁用。Amazon DocumentDB 审核功能是一种可选功能。它记录在集群中对对象（例如数据库、集合、索引和用户）采取的操作。

主题

- [支持的事件](#) (p. 185)
- [启用审核](#) (p. 187)
- [禁用审核](#) (p. 191)
- [访问审计事件](#) (p. 193)

支持的事件

Amazon DocumentDB 审计支持以下事件类别：

- 数据定义语言 (DDL)-包括数据库管理操作、连接、用户管理和授权。

- 数据操纵语言读取事件 (DML 读取) -包括find()以及各种聚合运算符、算术运算符、布尔运算符和其他读取查询运算符。
- 数据操纵语言写入事件 (DML 写入) -包括insert(), update(), delete(),和bulkWrite()运营商

事件类型如下所示。

Event type	类别	描述
authCheck	Authorization	未经授权的执行操作的尝试。
authenticate	连接	对新连接进行的成功或失败的身份验证尝试。
createDatabase	DDL	创建新数据库。
createCollection	DDL	在数据库中创建新集合。
createIndex	DDL	在集合中创建新索引。
dropCollection	DDL	删除数据库中的集合。
dropDatabase	DDL	删除数据库。
dropIndex	DDL	删除集合中的索引。
modifyChangeStreams	DDL	创建变更流。
renameCollection	DDL	重命名数据库中的集合。
createRole	管理角色	创建角色。
dropAllRolesFromDatabase	管理角色	删除数据库中的所有角色。
dropRole	管理角色	丢掉一个角色。
grantPrivilegesToRole	管理角色	向授予权限。
grantRolesToRole	管理角色	向授予用户定义角色。
revokePrivilegesFromRole	管理角色	撤消角色的权限。
revokeRolesFromRole	管理角色	撤消用户定义角色。
updateRole	管理角色	更新角色。
createUser	用户管理	创建新用户。
dropAllUsersFromDatabase	用户管理	删除数据库中的所有用户。
dropUser	用户管理	删除现有用户。
grantRolesToUser	用户管理	向用户授予角色。
revokeRolesFromUser	用户管理	撤销用户的角色。

Event type	类别	描述
updateUser	UserManagement	更新现有用户。
insert	DML 写入	将一个或多个文档插入到集合中。
delete	DML 写入	从集合中删除一个或多个文档。
update	DML 写入	修改集合中的一个或多个现有文档。
bulkWrite	DML 写入	使用执行顺序控件执行多个写入操作。
count	DML 已读	返回与集合或视图的 find() 查询相匹配的文档计数。
countDocuments	DML 已读	返回与集合或视图的查询相匹配的文档计数。
find	DML 已读	选择集合或视图中的文档，并将光标返回到所选文档。
findAndModify	DML 读取和 DML 写入	修改并返回单个文档。
findOneAndDelete	DML 读取和 DML 写入	根据筛选条件和排序标准删除单个文档，返回已删除的文档。
findOneAndReplace	DML 读取和 DML 写入	根据指定的筛选器替换单个文档。
findOneAndUpdate	DML 读取和 DML 写入	根据筛选条件和排序条件更新单个文档。
aggregate	DML 读取和 DML 写入	支持聚合管道中的 API。
distinct	DML 已读	在单个集合或视图中查找指定字段的不同值，并在数组中返回结果。

Note
此时不审计 TTL 删除事件。

启用审核

在集群上启用审核是一个两步过程。请确保完成这两个步骤，否则审计日志不会发送到 CloudWatch 日志。

第 1 步 启用 audit_logs 集群参数

要启用审计，您需要修改 audit_logs 参数组中的参数。audit_logs 是要记录的事件列表，以逗号分隔。事件必须以小写形式指定，列表元素之间不应有空格。

您可以设置以下参数组值：

值	描述	
ddl	设置此选项将启用对 DDL 事件的审计，例如 createDatabase、dropDatabase、createCollection、dropCollection、身份验证、createUser、dropUser、grantRolesToUser、revokeRolesFromUser、updateUser 和 dropAllUsersFromDatabase	
dml_read	设置此选项将启用对 DML 读取事件的审计，例如查找、排序计数、distinct、分组、项目、展开、GeoNear、GeoIntersects、GeoIntersects、GeoWithin 和其他 MongoDB 读取查询运算符。	
dml_write	设置此项将启用对 DML 写入事件的审计，例如 insert ()、update ()、delete () 和 BulkWrite ()	
all	设置此项将启用对数据库事件的审计，例如读取查询、写入查询、数据库操作和管理员操作。	
none	设置此项将禁用审计	
enabled (传统)	这是一个等同于“ddl”的传统参数设置。设置此选项将启用对 DDL 事件的审计，例如 createDatabase、dropDatabase、createCollection、dropCollection、身份验证、createUser、dropUser、grantRolesToUser、revokeRolesFromUser、updateUser 和 dropAllUsersFromDatabase。我们不建议使用该设置，因为它是传统设置。	
disabled (传统)	这是一个等同于“none”的传统参数设置。我们不建议使用该设置，因为它是传统设置。	

您也可以组合使用上面提到的值。

值	描述	
ddl, dml_read	设置此项将启用 DDL 事件和 DML 读取事件的辅助功能。	
ddl, dml_write	设置此项将启用 DDL 事件和 DML 写入的辅助功能	
dml_read, dml_write	设置此项将启用所有 DML 事件的审计	

Note

您无法修改默认参数组。

有关更多信息，请参阅以下内容：

- [创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#)

创建自定义参数组后，通过将 `audit_logs` 参数值更改为 `enabled` 来修改它。

- [修改 Amazon DocumentDB 集群参数组 \(p. 328\)](#)

第 2 步 启用Amazon CloudWatch 导出日志

当的值`audit_logs`集群参数是`enabled`，您还必须使将日志导出到亚马逊 CloudWatch。如果省略上述步骤中的任何一个，就不会将审核日志发送到 CloudWatch。

创建集群时，执行 `point-in-time-restore`，或者恢复快照，您可以启用 CloudWatch 执行以下步骤。

使用 Amazon Web Services Management Console

启用Amazon DocumentDB 将日志导出到 CloudWatch 使用控制台，请参阅以下主题：

- 创建集群时— 在 [使用 Amazon Web Services Management Console创建集群和主实例 \(p. 248\)](#)，请参阅创建集群：其他配置（第 5 步日志导出）
- 修改现有集群时—[使用 Amazon Web Services Management Console \(p. 259\)](#)。
- 执行集群快照还原时— [使用 Amazon Web Services Management Console 从集群快照进行还原 \(p. 213\)](#)，第 9 步。
- 当执行 `point-in-time` 恢复— [使用 Amazon Web Services Management Console还原到时间点 \(p. 217\)](#)（步骤 9）。

使用 Amazon CLI

在创建新集群时启用审计日志

以下代码创建集群`sample-cluster`并启用 CloudWatch 审核日志。

Example

对于 Linux、macOS 或 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
```

对于 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
```

在修改现有集群时启用审计日志

以下代码修改集群 `sample-cluster` 并启用 CloudWatch 审核日志。

Example

对于 Linux、macOS 或 Unix :

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

对于 Windows :

```
aws docdb modify-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

这些操作的输出将类似于下文 (JSON 格式) 。

```
{  
  "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"
  }
}
```

禁用审核

您可以禁用审核功能 CloudWatch 日志导出和禁用 `audit_logs` 参数。

禁用 CloudWatch 导出日志

您可以使用 Amazon Web Services Management Console 或 Amazon CLI 禁止导出审核日志。

使用 Amazon Web Services Management Console

以下过程使用 Amazon Web Services Management Console 禁用 Amazon DocumentDB 将日志导出到 CloudWatch。

禁用审计日志

1. 登录到 Amazon Web Services Management Console，然后打开亚马逊 DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。然后，选择要禁用导出日志的集群名称左侧的按钮。
3. 选择 Actions (操作)，然后选择 Modify (修改)。
4. 向下滚动到 Log exports (日志导出) 部分并选择 Disabled (已禁用)。
5. 选择 Continue (继续)。
6. 检查更改，然后选择何时将该更改应用到集群。
 - Apply during the next scheduled maintenance window (在下一个计划的维护时段内应用)
 - Apply immediately (立即应用)
7. 选择修改集群。

使用 Amazon CLI

以下代码修改集群 `sample-cluster` 并禁用 CloudWatch 审核日志。

Example

对于 Linux、macOS 或 Unix：


```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

对于 Windows：

```
aws docdb modify-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "DBCluster": {  
    "DBClusterParameterGroup": "default.docdb4.0",  
    "HostedZoneId": "ZNXKH85TT8WVW",  
    "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"  
      }  
    ]  
  }  
}
```

禁用 audit_logs 参数

要禁用集群的 `audit_logs` 参数，您可以修改集群，使其使用 `audit_logs` 参数值为 `disabled` 的参数组。或者，您可以修改集群的参数组中的 `audit_logs` 参数值，使其为 `disabled`。

有关更多信息，请参阅以下主题：

- [修改 Amazon DocumentDB 集群 \(p. 259\)](#)
- [修改 Amazon DocumentDB 集群参数组 \(p. 328\)](#)

访问审计事件

使用以下步骤访问您在Amazon 上的审核事件 CloudWatch.

1. 打开 CloudWatch 控制台位于<https://console.aws.amazon.com/cloudwatch/>.
2. 请确保您与您的 Amazon DocumentDB 集群位于同一区域。
3. 在导航窗格中，选择日志。
4. 要查找集群的审核日志，请从列表中找到并选择 `/aws/docdb/yourClusterName/audit`。

此时，每个实例名称的下方将显示该实例的审计事件。

在 Amazon DocumentDB 中备份和还原

Amazon DocumentDB (具有 MongoDB 兼容性) 会不断将您的数据备份到亚马逊 Simple Storage Service (Amazon S3) (为期 1 至 35 天) , 这样您可以快速还原到备份保留期内的任何时间点。作为此连续备份过程的一部分 , Amazon DocumentDB 还会自动拍摄您的数据快照。

Note

这些是服务管理的 Amazon S3 存储桶, 您将无法访问备份文件。如果您希望控制自己的备份, 请按照中的说明操作。 [转储、还原、导入和导出数据](#)。

还可以通过为集群的数据创建手动快照来超出备份保留期保留备份数据。备份过程不会影响您的集群的性能。

本节讨论 Amazon DocumentDB 中备份功能的使用案例, 并为您展示如何管理 Amazon DocumentDB 集群的备份。

主题

- [备份和还原：概念 \(p. 194\)](#)
- [了解备份存储使用量 \(p. 195\)](#)
- [转储、还原、导入和导出数据 \(p. 196\)](#)
- [集群快照注意事项 \(p. 199\)](#)
- [比较自动快照和手动快照 \(p. 201\)](#)
- [创建手动集群快照 \(p. 201\)](#)
- [复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)
- [共享 Amazon DocumentDB 集群快照 \(p. 209\)](#)
- [从集群快照还原 \(p. 213\)](#)
- [还原到某个时间点 \(p. 217\)](#)
- [删除集群快照 \(p. 221\)](#)

备份和还原：概念

名词	描述	API (动词)
备份保留期	介于 1 到 35 天之间的时间段, 在此时间段内, 您可以执行时间点还原。	<code>create-db-cluster</code> <code>modify-db-cluster</code> <code>restore-db-cluster-to-point-in-time</code>
Amazon DocumentDB 存储卷	高可用性和高持久性的存储卷, 可以跨三个	<code>create-db-cluster</code> <code>delete-db-cluster</code>

名词	描述	API (动词)
	可用区以六种方式复制数据。无论集群中的实例数量是多少，Amazon DocumentDB 集群都具有高持久性。	
备份时段	拍摄自动快照的一天中的时间段。	create-db-cluster describe-db-cluster modify-db-cluster
自动快照	每日快照是集群的完整备份，由 Amazon DocumentDB 中的连续备份过程自动创建。	restore-db-cluster-from-snapshot describe-db-cluster-snapshot-attributes describe-db-cluster-snapshots
手动快照	您手动创建的快照，用于超出备份期间保留集群的完整备份。	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

了解备份存储使用量

Amazon DocumentDB 备份存储包含备份保留期内的连续备份和保留期外的手动快照。要控制备份存储使用量，可以缩短备份保留间隔和/或删除不再需要的旧的手动快照。有关 Amazon DocumentDB 备份的一般信息，请参阅在 [Amazon DocumentDB 中备份和还原 \(p. 194\)](#)。有关 Amazon DocumentDB 备份存储的定价信息，请参阅[Amazon DocumentDB 定价](#)。

要控制成本，您可以监控持续备份和存在时间超出保留期的手动快照所消耗的存储量。然后，您可以缩短备份保留间隔，并在不再需要手动快照时删除它们。

您可以使用 Amazon CloudWatch 指标TotalBackupStorageBilled、SnapshotStorageUsed，和BackupRetentionPeriodStorageUsed以查看和监控 Amazon DocumentDB 备份使用的存储量，如下所示：

- BackupRetentionPeriodStorageUsed 表示目前用于存储连续备份的备份存储量。此度量值取决于集群卷的大小和您在保留期内所做的更改的数量。但是，出于计费目的，此指标不会超过保留期内的集群卷的累计大小。例如，如果您的集群大小为 100 GiB 且保留期为两天，则BackupRetentionPeriodStorageUsed 的最大值为 200 GiB (100 GiB + 100 GiB)。

- `SnapshotStorageUsed` 表示用于存储超出备份保留期的手动快照的备份存储量。在保留期内拍摄的手动快照不计入备份存储。同样，自动快照不会计入备份存储。每个快照的大小是您拍摄快照时的集群卷的大小。`SnapshotStorageUsed` 值取决于您保留的快照数和每个快照的大小。例如，假设您有一个保留期外的快照，并且在拍摄快照时集群卷大小为 100 GiB。`SnapshotStorageUsed` 的数量将为 100 GiB。
- `TotalBackupStorageBilled` 表示 `BackupRetentionPeriodStorageUsed` 与 `SnapshotStorageUsed` 之和再减去免费备份存储量（等于一天的集群卷的大小）。例如，如果集群大小为 100 GiB，您有一天的保留期，并且您有一个保留期外的快照，`TotalBackupStorageBilled` 是 100 GiB（ $100 \text{ GiB} + 100 \text{ GiB} - 100 \text{ GiB} - 100 \text{ GiB}$ ）。
- 这些指标是针对每个 Amazon DocumentDB 集群单独计算的。

您可以通过以下方式使用 CloudWatch 指标监控 Amazon DocumentDB 集群并生成报告。[CloudWatch 控制台](#)。有关如何使用 CloudWatch 指标的更多信息，请参阅 [监控 Amazon DocumentDB \(p. 365\)](#)。

转储、还原、导入和导出数据

您可以使用 `mongodump`、`mongorestore`、`mongoexport` 和 `mongoimport` 用于将数据移入和移出 Amazon DocumentDB 集群的实用工具。此部分讨论了所有这些工具和配置的用途，以帮助您实现更好的性能。

主题

- [mongodump \(p. 196\)](#)
- [mongorestore \(p. 197\)](#)
- [mongoexport \(p. 197\)](#)
- [mongoimport \(p. 197\)](#)
- [教程 \(p. 198\)](#)

mongodump

`mongodump` 实用工具创建 MongoDB 数据库的二进制 (BSON) 备份。这些区域有：`mongodump` 由于以二进制格式存储数据可实现较高的大小效率，因此，在希望将数据还原到 Amazon DocumentDB 集群中时，工具是转储源 MongoDB 部署中的数据的首选方法。

根据您用来执行命令的实例或计算机上的可用资源，您可以加快您的速度。`mongodump` 方法是增加从默认值 1 中转储的并行连接数量使用 `--numParallelCollections` 选项。一个好的经验法则是，在 Amazon DocumentDB 集群的主实例上，为每个 vCPU 启动一个工作线程。

示例用法

下面是的使用示例。`mongodump` Amazon DocumentDB 集群中的实用程序，`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=sample-password
```

```
--password=abc0123 \  
--sslCAFile rds-combined-ca-cn-bundle.pem
```

mongorestore

利用 mongorestore 实用工具，您可以还原使用 mongodump 实用工具创建的数据库的二进制 (BSON) 备份。您可以通过使用 `--numInsertionWorkersPerCollection` 选项（默认值为 1）增加还原期间每个集合的工作线程数来提高还原性能。一个好的经验法则是，在 Amazon DocumentDB 集群的主实例上，为每个 vCPU 启动一个工作线程。

示例用法

下面是使用的示例 mongorestore Amazon DocumentDB 集群中的实用程序，sample-cluster。

```
mongorestore --ssl \  
  --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --username=sample-user \  
  --password=abc0123 \  
  --sslCAFile rds-combined-ca-cn-bundle.pem <fileToBeRestored>
```

mongoexport

这些区域有：mongoexport 工具将 Amazon DocumentDB 中的数据导出为 JSON、CSV 或 TSV 文件格式。mongoexport 工具是导出要求可供人类或机器读取的数据的首选方法。

Note

mongoexport 无法直接支持并行导出。但是，可以通过同时为不同的集合执行多个 mongoexport 作业来提高性能。

示例用法

下面是的使用示例。mongoexport Amazon DocumentDB 集群中的工具，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 rds-combined-ca-cn-bundle.pem
```

mongoimport

这些区域有：mongoimport 工具将 JSON、CSV 或 TSV 文件的内容导入 Amazon DocumentDB 集群中。可以使用 `--numInsertionWorkers` 参数实现并行化和加快导入速度（默认值为 1）。

示例用法

下面是使用的示例 mongoimport Amazon DocumentDB 集群中的工具，sample-cluster。

```
mongoimport --ssl \  
  --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=sample-collection \  
  --db=sample-database \  
  --file=<yourFile> \  
  --numInsertionWorkers=1
```

```
--numInsertionWorkers 4 \  
--username=sample-user \  
--password=abc0123 \  
--sslCAFile rds-combined-ca-cn-bundle.pem
```

教程

下面的教程介绍如何使用 `mongodump`、`mongoexport`、`mongoimport` 用于将数据移入和移出 Amazon DocumentDB 集群的实用工具。

1. 先决条件— 在开始之前，请确保您的 Amazon DocumentDB 集群已配置，并且您有权访问与您的集群所在的同一 VPC 中的 Amazon EC2 实例。有关更多信息，请参阅[使用 Amazon EC2 Connect](#) (p. 466)。

要能够使用 mongo 实用工具，您必须已在 EC2 实例中安装 `mongodb-org-tools` 包，如下所示。

```
sudo yum install mongodb-org-tools-4.0.18
```

由于 Amazon DocumentDB 默认使用传输层安全性 (TLS) 加密，因此您还必须下载 Amazon RDS 证书颁发机构 (CA) 文件以使用 mongo shell 进行连接，如下所示。

```
wget https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem
```

2. 下载示例数据在本教程中，您将下载一些包含餐厅相关信息的示例数据。

```
wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/datasets/  
restaurant.json
```

3. 将示例数据导入 Amazon DocumentDB— 由于数据采用的是逻辑 JSON 格式，因此您将使用 `mongoimport` 用于将数据导入 Amazon DocumentDB 集群的实用程序。

```
mongoimport --ssl \  
--host="tutorialCluster.amazonaws.com:27017" \  
--collection=restaurants \  
--db=business \  
--file=restaurant.json \  
--numInsertionWorkers 4 \  
--username=<yourUsername> \  
--password=<yourPassword> \  
--sslCAFile rds-combined-ca-cn-bundle.pem
```

4. 用转储数据 `mongodump` — 现在，您的 Amazon DocumentDB 集群中有数据，您可以使用将数据转储到的二进制转储。 `mongodump` 实用程序。

```
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 rds-combined-ca-cn-bundle.pem
```

5. 删除 `restaurants` collection— 在恢复之前 `restaurants` 在 `business` 数据库中，您必须先删除数据库中已有的集合，如下所示。

```
use business
```

```
db.restaurants.drop()
```

6. 使用恢复数据**mongorestore**— 使用步骤 3 中的数据二进制转储，您现在可以使用**mongorestore**用于将数据还原到 Amazon DocumentDB 集群的实用程序。

```
mongorestore --ssl \  
  --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \  
  --numParallelCollections 4 \  
  --username=<yourUsername> \  
  --password=<yourPassword> \  
  --sslCAFile rds-combined-ca-cn-bundle.pem restaurantDump.bson
```

7. 使用导出数据**mongoexport**— 要完成本教程，请以 JSON 文件的格式从集群中导出数据，与您在步骤 1 中导入的文件没有什么不同。

```
mongoexport --ssl \  
  --host="tutorialCluster.node.us-east-1.docdb.amazonaws.com:27017" \  
  --collection=restaurants \  
  --db=business \  
  --out=restaurant2.json \  
  --username=<yourUsername> \  
  --password=<yourPassword> \  
  --sslCAFile rds-combined-ca-cn-bundle.pem
```

8. 验证— 您可以使用以下命令验证步骤 5 的输出是否会产生与步骤 1 相同的结果。

```
wc -l restaurant.json
```

此命令的输出：

```
2548 restaurant.json
```

```
wc -l restaurant2.json
```

此命令的输出：

```
2548 restaurant2.json
```

集群快照注意事项

Amazon DocumentDB 会在您的集群的备份时段内创建集群的每日自动快照。Amazon DocumentDB 根据您的指定的备份保留期保存集群的自动快照。如果需要，您可以将集群恢复到备份保留期中的任意时间点。在相同区域中对相同集群执行复制操作时，不会拍摄自动快照。

除了自动集群快照之外，您还可以手动创建集群快照。您可以同时复制自动快照和手动快照。有关更多信息，请参阅 [创建手动集群快照 \(p. 201\)](#) 和 [复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)。

Note

您的集群必须处于可用 状态，才能拍摄自动快照。

您无法共享 Amazon DocumentDB 自动集群快照。解决方法是通过复制自动快照来创建手动快照，然后共享该副本。有关复制快照的更多信息，请参阅[复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)。有关从快照还原集群的更多信息，请参阅[从集群快照还原 \(p. 213\)](#)。

备份存储

每个存储空间的 Amazon DocumentDB 备份存储 Amazon Web Services 区域由备份保留期所需的备份存储组成，其中包括该区域中的自动和手动集群快照。默认备份保留期为 1 天。有关备份存储定价的更多信息，请参阅[Amazon DocumentDB 定价](#)。

在删除集群时，所有自动快照都将被删除且无法恢复。但是，在删除集群时不会删除手动快照。如果您选择让让在删除集群之前让 Amazon DocumentDB 创建最终快照（手动快照），则可以使用该最终快照来恢复集群。

有关快照和存储的更多信息，请参阅 [了解备份存储使用量 \(p. 195\)](#)。

备份时段

自动快照在每天的首选备份时段中进行。如果快照所需的时间超过了分配到备份时段的时间，那么即使备份时段已结束，备份过程也会继续直至完成。备份时段不能与集群的每周维护时段重叠。

如果创建集群时未指定首选备份时段，Amazon DocumentDB 将分配 30 分钟的默认备份时段。该时段是随机从与集群区域关联的 8 小时时间数据块中选择出来的。通过修改集群来更改您的首选备份时段。有关更多信息，请参阅[修改 Amazon DocumentDB 集群 \(p. 259\)](#)。

区域	UTC 时间数据块
美国东部（俄亥俄）	03:00-11:00
美国东部（弗吉尼亚北部）	03:00-11:00
美国西部（俄勒冈）	06:00-14:00
亚太地区（孟买）	17:30-01:30
亚太地区（首尔）	13:00-21:00
亚太地区（新加坡）	14:00-22:00
亚太地区（悉尼）	12:00-20:00
亚太地区（东京）	13:00-21:00
加拿大（中部）	22:00-06:00
欧洲（法兰克福）	23:00-07:00
欧洲（爱尔兰）	22:00-06:00
欧洲（巴黎）	22:00-06:00
Amazon GovCloud (US)	22:00-06:00

备份保留期

备份保留期是自动备份在被自动删除之前保留的天数。Amazon DocumentDB 支持 1 到 35 天的备份保留期。

您可以在创建集群时设置备份保留期。如果您未明确设置备份保留期，则会为集群分配 1 天的默认备份保留期。创建集群后，您通过使用 Amazon Web Services Management Console 或 Amazon CLI 修改集群来修改备份保留期。有关更多信息，请参阅[修改 Amazon DocumentDB 集群 \(p. 259\)](#)。

比较自动快照和手动快照

以下是 Amazon DocumentDB (与 MongoDB 兼容) 自动快照和手动快照的主要特征。

Amazon DocumentDB 自动快照具有以下主要特征：

- 自动快照命名— 自动快照名称遵循模式 `rds:<cluster-name>-yyyy-mm-dd-hh-mm`，与 `yyyy-mm-dd-hh-mm` 表示创建快照的日期和时间。
- 按计划自动创建— 在创建或修改集群时，您可以设置备份保留期转换为介于 1 到 35 天之间的整数值。默认情况下，新集群的备份保留期为 1 天。备份保留期定义自动快照在被自动删除之前保留的天数。您无法在 Amazon DocumentDB 集群上禁用自动备份。

除了设置备份保留期，您还可以设置备份时段，即一天中创建自动快照的时间。

- 删除自动快照— 当您删除自动快照的集群时，将会删除自动快照。您不能手动删除自动快照。
- 增量式在备份保留期内会记录数据库更新，因此存在增量更改记录。
- 从自动快照还原— 您可以使用从自动快照还原 Amazon Web Services Management Console 或者 Amazon CLI。在使用 Amazon CLI 从快照还原时，必须在集群可用 后单独添加实例。
- 共享— 您无法共享 Amazon DocumentDB 自动集群快照。解决方法是通过复制自动快照来创建手动快照，然后共享该副本。有关复制快照的更多信息，请参阅[复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)。有关从快照还原集群的更多信息，请参阅[从集群快照还原 \(p. 213\)](#)。
- 您可以在备份保留期内的任何时间点进行恢复— 由于数据库更新采用增量式记录，所以您可以将集群还原到备份保留期内的任何时间点。

在使用 Amazon CLI 从自动快照或从时间点还原进行还原时，必须在集群可用 后单独添加实例。

Amazon DocumentDB 手动快照具有以下主要特征：

- 按需创建 Amazon DocumentDB 手动快照是使用 Amazon DocumentDB 管理控制台或按需创建的。Amazon CLI。
- 删除手动快照仅当您使用 Amazon DocumentDB 控制台或显式删除手动快照时，才会将其删除。Amazon CLI。在删除手动快照的集群时，不会删除手动快照。
- 完整备份拍摄手动快照时，会创建并存储您的集群数据的完整备份。
- 手动快照命名— 您可以指定手动快照名称。Amazon DocumentDB 不添加 `datetime` 在名称上加时间戳，因此，如果要在名称中包含此信息，则必须添加此信息。
- 从手动快照还原您可以使用控制台或从手动快照还原 Amazon CLI。在使用 Amazon CLI 从快照还原时，必须在集群可用 后单独添加实例。
- Service Quotas— 每个最多支持 100 个手动快照 Amazon Web Services 区域。
- 共享— 您可以共享手动集群快照，这些快照可以通过授权复制。Amazon Web Services 账户。您可以共享加密或未加密的手动快照。有关复制快照的更多信息，请参阅[复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)。
- 你还原到手动快照的拍摄时间从手动快照还原时，将还原到拍摄手动快照的时间。

在使用 Amazon CLI 从快照还原时，必须在集群可用 后单独添加实例。

创建手动集群快照

您可以使用 Amazon Web Services Management Console 或 Amazon CLI 创建手动快照。创建快照所用时间因数据库大小而异。在创建快照时，您必须执行以下操作：

1. 确定要备份的集群。
2. 为快照指定名称。这样，您以后便可从中还原。

使用 Amazon Web Services Management Console 创建集群快照

要使用 Amazon Web Services Management Console 创建手动快照，您可以按照以下任一方法进行操作。

1. 方法 1：

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择快照。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在 Snapshots (快照) 页面上，选择 Create (创建)。
4. 在 Create cluster snapshot (创建集群快照) 页面上：
 - a. 集群标识符— 从集群的下拉列表中，选择要为其创建快照的集群。
 - b. 快照标识符— 输入快照的名称。

快照命名约束：

- 长度为 [1-255] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个集群 (跨 Amazon RDS、Amazon Neptune 和 Amazon DocumentDB) 必须是唯一的。Amazon 账户，每个区域。

- c. 选择 Create (创建)。

2. 方法 2：

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在 Clusters (集群) 页面上，选择要拍摄快照的集群左侧的按钮。
4. 从 Actions (操作) 菜单中，选择 Take snapshot (拍摄快照)。
5. 在 Create cluster snapshot (创建集群快照) 页面上：
 - a. 快照标识符— 输入快照的名称。

快照命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个集群 (跨 Amazon RDS、Amazon Neptune 和 Amazon DocumentDB) 必须是唯一的。Amazon 账户，每个区域。

- b. 选择 Create (创建)。

使用 Amazon CLI 创建集群快照

要使用 Amazon CLI 创建集群快照，请使用带以下参数的 `create-db-cluster-snapshot` 操作。

参数

- `--db-cluster-identifier` – 必需。要拍摄快照的集群的名称。该集群必须存在且可用。
- `--db-cluster-snapshot-identifier` – 必需。正在创建的手动快照的名称。

以下示例为名为 `sample-cluster` 的集群创建名为 `sample-cluster-snapshot` 的快照。

对于 Linux、macOS 或 Unix：

```
aws docdb create-db-cluster-snapshot \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

对于 Windows：

```
aws docdb create-db-cluster-snapshot ^  
  --db-cluster-identifier sample-cluster ^  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

此操作的输出将类似于下文。

```
{  
  "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"  
  }  
}
```

复制 Amazon DocumentDB 集群快照

在 Amazon DocumentDB 中，您可以同时复制手动和自动快照 Amazon Web Services 区域或者不同的 Amazon Web Services 区域在同一账户中。您还可以共享其他人拥有的快照 Amazon Web Services 账户在

相同的Amazon Web Services 区域。但是，您不能跨复制集群快照Amazon Web Services 区域和Amazon Web Services 账户在单一步骤中。这些操作必须单独执行。

作为对复制的替代，您也可与其他人共享手动快照。Amazon Web Services 账户。有关更多信息，请参阅[共享 Amazon DocumentDB 集群快照 \(p. 209\)](#)。

Note

Amazon DocumentDB 根据您保留的备份和快照数据量和您保留的时间段来对您收费。有关与 Amazon DocumentDB 备份和快照关联的存储的更多信息，请参阅[了解备份存储使用量 \(p. 195\)](#)。有关 Amazon DocumentDB 存储的定价信息，请参阅[Amazon DocumentDB 定价](#)。

主题

- [复制共享快照 \(p. 204\)](#)
- [跨复制快照Amazon Web Services 区域 \(p. 204\)](#)
- [限制 \(p. 204\)](#)
- [处理加密 \(p. 204\)](#)
- [参数组注意事项 \(p. 205\)](#)
- [复制集群快照 \(p. 205\)](#)

复制共享快照

您可以复制其他人与您共享的快照Amazon Web Services 账户。如果您要从另一个快照复制共享的加密快照Amazon Web Services 账户，您必须有权访问Amazon KMS加密快照使用的加密密钥。

您只能在同一个中复制共享快照Amazon Web Services 区域快照是否加密。有关更多信息，请参阅[the section called “处理加密” \(p. 204\)](#)。

跨复制快照Amazon Web Services 区域

将快照复制到Amazon Web Services 区域这不同于源快照Amazon Web Services 区域，每个副本都是完整的快照。完整快照副本包含还原 Amazon DocumentDB 集群需要的所有数据和元数据。

取决于Amazon Web Services 区域涉及的数据量和要复制的数据量，可能需要数小时才能完成跨区域快照复制。有时，某一给定的源可能会发出大量跨区域快照复制请求。Amazon Web Services 区域。在这种情况下，Amazon DocumentDB 可能会发出来自该源的新跨区域复制请求。Amazon Web Services 区域直到一些正在进行的副本完成为止。当复制请求在队列中时，不显示有关这些复制请求的进度信息。复制开始后即显示进度信息。

限制

复制快照时，存在以下一些限制：

- 如果您在目标快照可用之前删除了源快照，则快照复制将失败。在删除源快照之前，请确保目标快照的状态为 AVAILABLE。
- 每个账户最多可以同时进行到同一目标区域的五个快照复制请求。
- 根据所涉及的区域和要复制的数据量，可能需要数小时才能完成跨区域快照复制。有关更多信息，请参阅[跨复制快照Amazon Web Services 区域 \(p. 204\)](#)。

处理加密

您可以复制已使用 Amazon KMS 加密密钥加密的快照。如果您复制加密的快照，则此快照的副本也必须加密。如果你在同一个中复制加密的快照Amazon Web Services 区域，你可以用相同的方式加密副本Amazon

KMS加密密钥作为原始快照，也可指定不同的快照。Amazon KMS加密密钥。如果您跨区域复制加密的快照，则不能使用相同的快照。Amazon KMS用于源快照的副本的加密密钥，因为Amazon KMS密钥是特定于区域的。而是必须指定Amazon KMS密钥在目的地有效Amazon Web Services 区域。

源快照在复制过程中保持加密状态。有关更多信息，请参阅[Amazon DocumentDB 中的数据保护 \(p. 133\)](#)。

Note

对于 Amazon DocumentDB 集群快照，在复制快照时，您无法对未加密的集群快照进行加密。

参数组注意事项

跨区域复制快照时，复制不包括由原始 Amazon DocumentDB 集群使用的参数组。当您还原快照来创建新集群时，该集群获取的默认参数组。Amazon Web Services 区域它是在中创建的。要为新集群提供与原始集群相同的参数组，您必须执行以下操作：

1. 在目的地Amazon Web Services 区域、[创建 Amazon DocumentDB 集群参数组](#)设置与原始集群相同。如果新中已存在选项组Amazon Web Services 区域，也可以使用它。
2. 在目标中恢复快照之后Amazon Web Services 区域，修改新 Amazon DocumentDB 集群，并添加上一步中的新参数组或现有参数组。有关更多信息，请参阅[修改Amazon DocumentDB 集群 \(p. 259\)](#)。

复制集群快照

Amazon DocumentDB 可以使用Amazon Web Services Management Console或者Amazon CLI，如下所示。

使用 Amazon Web Services Management Console复制集群快照

要使用创建集群快照副本Amazon Web Services Management Console，请完成以下步骤。此过程适用于复制加密或未加密的集群快照。Amazon Web Services 区域或跨区域。

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择快照，然后选择要复制的快照左侧的按钮。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在 Actions 菜单中，选择 Copy。
4. 在结果中创建群集快照的副本页面，完成设置部分。
 - a. 目标区域— 可选。要将集群快照复制到其他快照Amazon Web Services 区域，选择那个Amazon Web Services 区域为了目标区域。
 - b. 新快照标识符— 输入新快照的名称。

目标快照命名约束：

- 不能是现有快照的名称。
- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个 Amazon DocumentDB 的每个 Amazon Neptune 和 Amazon DocumentDB 的所有集群 (必须是唯一的) Amazon Web Services 账户每区域数。

- c. 复制标签— 要将您在源快照上拥有的任何标签复制到快照副本，请选择复制标签。
5. 完成 Encryption-at-rest (静态加密) 部分。
 - a. 静态加密— 如果您的快照已加密，则这些选项对您不适用，因为您无法从加密的快照创建未加密的副本。如果您的快照未加密，请选择以下选项之一：
 - 要加密所有集群的数据，请选择启用静态加密。如果您选择此选项，则必须指定一个主密钥。
 - 要让集群的数据不加密，请选择禁用静态加密。如果您选择此选项，您的快照的副本数据不会被加密，并且您已完成加密部分。
 - b. 主密钥— 从下拉列表中，选择以下选项之一：
 - (默认) aws/rds— 账号和Amazon KMS此选项的后面列出密钥 ID。
 - <some-key-name>— 如果您创建了密钥，它将列出并可供您选择。
 - 输入键 ARN— 在进行筛选框中，输入您的 Amazon 资源名称 (ARN)Amazon KMS键。ARN 的格式为 `arn:aws:kms:<region>:<accountID>:key/<key-id>`。
6. 要制作所选快照的副本，请选择 Copy snapshot (复制快照)。或者，您也可以选择Cancel不制作快照的副本。

使用复制未加密的集群快照Amazon CLI

要使用复制未加密的集群快照Amazon CLI，使用copy-db-cluster-snapshot操作带以下参数。如果您将快照复制到另一个快照Amazon Web Services 区域在中运行命令Amazon Web Services 区域快照将被复制到的地方。

- **--source-db-cluster-snapshot-identifier** – 必需。要制作副本的集群快照的标识符。集群快照必须存在并且处于可用 状态。如果您将快照复制到另一个快照Amazon Web Services 区域对于源，此标识符必须采用 ARN 格式Amazon Web Services 区域. 此参数不区分大小写。
- **--target-db-cluster-snapshot-identifier** – 必需。要从源集群快照创建的新集群快照标识符。此参数不区分大小写。

目标快照命名约束：

- 不能是现有快照的名称。
- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个 Amazon DocumentDB 的每个 Amazon Neptune 和 Amazon DocumentDB 的所有集群 (必须是唯一的) Amazon Web Services 账户每区域数。
- **--source-region**— 如果您将快照复制到另一个快照Amazon Web Services 区域，请指定Amazon Web Services 区域加密集群快照将从中复制到。

如果你要将快照复制到另一个Amazon Web Services 区域而且你不指定--source-region，则必须指定pre-signed-url请改用选项。这些区域有：pre-signed-url值必须是包含签名版本 4 签名的请求的 URL。CopyDBClusterSnapshot在源代码中调用的操作Amazon Web Services 区域从中复制群集快照。要了解有关的更多信息pre-signed-url请参阅[CopyDBClusterSnapshot](#)。

- **--kms-key-id**— 用于对集群快照副本进行加密的密钥的 KMS 密钥标识符。

如果您要将加密的集群快照复制到另一个快照Amazon Web Services 区域，此为必需参数。您必须为目标指定 KMS 密钥。Amazon Web Services 区域。

如果你要在同一个中复制加密的群集快照Amazon Web Services 区域，Amazon KMS关键参数是可选的。集群快照的副本使用相同的加密Amazon KMSkey 作为源群集快照。如果你想指定一个新的Amazon KMS 用于对副本进行加密的加密密钥，您可以使用此参数来加密。

- **--copy-tags** – Optional. The tags and values to be copied over.

要在正在进行复制时取消操作，您可以删除由以下标识的目标集群快照。--target-db-cluster-snapshot-identifier要么TargetDBClusterSnapshotIdentifier当该群集快照在中时复制状态。

Example

示例 1：将未加密快照复制到同一区域

以下Amazon CLI示例创建的副本sample-cluster-snapshot被命名sample-cluster-snapshot-copy在相同的Amazon Web Services 区域作为源快照。创建副本时，原始快照上的所有标签都将复制到快照副本。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文。

```
{  
  "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

示例 2：跨复制未加密的快照Amazon Web Services 区域

以下Amazon CLI示例创建的副本sample-cluster-snapshot，其中有 ARNarn:aws:rds:us-east-1:123456789012:cluster-snapshot:sample-cluster-snapshot. 该副本被命名为sample-cluster-snapshot-copy而且在Amazon Web Services 区域在其中运行命令。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文。

```
{  
  "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

示例 3：跨复制加密的快照Amazon Web Services 区域

以下Amazon CLI示例创建的副本sample-cluster-snapshot从 us-west-2 区域到 us-east-1 区域。在 us-east-1 区域调用此命令。

对于 Linux、macOS 或 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 \  
  --target-region us-east-1
```

```
--kms-key-id sample-us-east-1-key
```

对于 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
```

此操作的输出将类似于下文。

```
{
  "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",
  }
}
```

共享 Amazon DocumentDB 集群快照

在 Amazon DocumentDB 中，您可以共享手动集群快照，这些快照可以通过授权复制。Amazon Web Services 账户。您可以共享加密或未加密的手动快照。共享未加密的快照时，已授权 Amazon Web Services 账户可以直接从快照还原集群，而不用创建快照的副本并通过它来还原。但是，您无法从共享并且加密的快照还原集群。您可以改为创建集群的副本并通过该副本来还原集群。有关复制快照的更多信息，请参阅[复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)。

Note

您无法共享 Amazon DocumentDB 自动集群快照。解决方法是通过复制自动快照来创建手动快照，然后共享该副本。有关复制快照的更多信息，请参阅[复制 Amazon DocumentDB 集群快照 \(p. 203\)](#)。有关从快照还原集群的更多信息，请参阅[从集群快照还原 \(p. 213\)](#)。

您可以与最多 20 个其他快照共享 Amazon Web Services 账户。您也可以将未加密的手动快照作为公有快照进行共享，这样所有账户均可使用此快照。当以公有快照形式共享快照时，确保不要将您的私有信息包含在任何公有快照之中。

与其他人共享手动快照时 Amazon Web Services 账户并使用从共享快照还原集群 Amazon CLI 或 Amazon DocumentDB API，您必须指定共享快照的 Amazon 资源名称 (ARN) 作为快照标识符。

共享加密的快照

以下限制适用于共享加密快照：

- 您无法公开共享加密的快照。
- 如果某个快照已使用共享该快照的账户的默认 Amazon KMS 加密密钥进行加密，则您无法共享该快照。

按照以下步骤共享加密的快照。

1. 与您希望允许其访问快照的任何账户共享用于加密快照的 Amazon Key Management Service (Amazon KMS) 加密密钥。

您可以共享 Amazon KMS 用另一个加密密钥 Amazon 通过将其他账户添加到 Amazon KMS 密钥策略。有关更新密钥策略的详细信息，请参阅[使用以下密钥策略 Amazon 自杀](#)中的 Amazon Key Management Service 开发人员指南。有关创建密钥策略的示例，请参阅本主题下文中的[创建 IAM 策略来启用加密快照的复制功能 \(p. 211\)](#)。

2. 使用 Amazon CLI ([如下所示 \(p. 212\)](#)) 与其他账户共享加密的快照。

允许访问 Amazon KMS 加密密钥

另一个 Amazon Web Services 账户要从您的账户复制共享的加密快照，您与之共享快照的账户必须有权访问 Amazon KMS 加密了快照的密钥。要允许另一账户访问 Amazon KMS 密钥，请使用您要与之共享的账户的 ARN 将 Amazon KMS 密钥的密钥策略更新为 Amazon KMS 密钥策略中的委托人。然后允许 `kms:CreateGrant` 操作。

在您授予账户访问权限之后 Amazon KMS 加密密钥，要复制加密的快照，该账户必须创建 Amazon Identity and Access Management (IAM) 用户（如果还没有）。此外，该账户还必须将 IAM 策略附加到该 IAM 用户，以允许此用户使用您的 Amazon KMS 键。此账户必须是 IAM 用户，不能是 root 用户 Amazon Web Services 账户由于身份 Amazon KMS 安全限制。

在以下密钥策略示例中，用户 123451234512 是 Amazon KMS 加密密钥的所有者。用户 123456789012 是要与之共享密钥的账户。此更新的密钥策略向该账户提供了对 Amazon KMS 密钥的访问权限。它通过包含根目录的 ARN 来实现此目的 Amazon Web Services 账户作为策略的委托人的用户 123456789012 的身份，并允许 `kms:CreateGrant` 操作。

```
{
  "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 }}
    }
}
```

创建 IAM 策略来启用加密快照的复制功能

当外部 Amazon Web Services 账户有权访问你的 Amazon KMSkey, 该账户的所有者可创建一个策略来允许为该账户创建的 IAM 用户能够复制使用该账户加密的加密快照。Amazon KMS 键。

以下示例显示了一个可以附加到 IAM 用户的策略。Amazon Web Services 账户 123456789012。该策略使 IAM 用户能够从账户 123451234512 复制已使用 Amazon KMS 密钥 c989c1dd-a3f2-4a5d-8d96-e793d082ab26 在 us-west-2 区域。

```
{
  "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
        }
      }
    }
  ]
}
```

有关更新密钥策略的详细信息，请参阅[使用以下密钥策略Amazon KMS](#)中的 Amazon Key Management Service 开发人员指南。

共享快照

要共享快照，请使用 `Amazon DocumentDBmodify-db-snapshot-attribute` 操作。使用 `--values-to-add` 参数添加 ID 列表。Amazon Web Services 账户有权恢复手动快照。

以下示例允许两个 Amazon Web Services 账户标识符（123451234512 和 123456789012）用于还原名为 `manual-snapshot1` 的快照。它还会删除 `all` 属性值以将该快照标记为私有。

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier sample-cluster-snapshot \
  --attribute-name restore \
  --values-to-add ['123451234512','123456789012']'
```

对于 Windows：

```
aws docdb modify-db-cluster-snapshot-attribute ^
  --db-cluster-snapshot-identifier sample-cluster-snapshot ^
  --attribute-name restore ^
  --values-to-add ['123451234512','123456789012']'
```

此操作的输出将类似于下文。

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123451234512",
          "123456789012"
        ]
      }
    ]
  }
}
```

删除 Amazon Web Services 账户列表中的标识符，请使用 `--values-to-remove` 参数。以下示例阻止 Amazon Web Services 账户中的 ID 123456789012 从还原快照。

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier sample-cluster-snapshot \
  --attribute-name restore \
  --values-to-remove ['123456789012']'
```

对于 Windows：

```
aws docdb modify-db-cluster-snapshot-attribute ^
  --db-cluster-snapshot-identifier sample-cluster-snapshot ^
  --attribute-name restore ^
  --values-to-remove ['123456789012']'
```

此操作的输出将类似于下文。

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123451234512"
        ]
      }
    ]
  }
}
```

从集群快照还原

Amazon DocumentDB (与 MongoDB 兼容) 能够创建存储卷的集群快照。可通过从集群快照还原来创建新集群。在还原集群时，您需提供用于还原的集群快照的名称以及还原创建的新集群的名称。您无法从快照还原到现有集群，因为还原时将新建一个新集群。

当从集群快照还原集群时：

- 此操作仅还原集群，而不还原集群的实例。您必须调用 `create-db-instance` 操作为还原的集群创建实例，并在 `--db-cluster-identifier` 中指定还原的集群的标识符。您只能在集群可用 后才能创建实例。
- 您无法将加密快照还原到未加密集群。但是，您可以通过指定 Amazon KMS 密钥将未加密快照还原到加密集群。
- 要从加密快照还原集群，您必须有权访问 Amazon KMS 密钥。

Note

您无法将 3.6 集群还原到 4.0 集群，但可以从一个群集版本迁移到另一个群集版本。有关更多信息，请转至 [Amazon DocumentDB \(p. 112\)](#)。

使用 Amazon Web Services Management Console 从集群快照进行还原

以下过程说明如何使用 Amazon DocumentDB 管理控制台从集群快照还原 Amazon DocumentDB 集群。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择快照，然后选择要用于还原集群的快照左侧的按钮。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在 Actions (操作) 菜单上，选择 Restore (还原)。
4. 在还原快照页面上，填写配置部分。
 - a. 集群标识符— 新集群的名称。您可以接受 Amazon DocumentDB 提供的名称或键入您喜欢的名称。Amazon DocumentDB 提供的名称格式为 `docdb-加上 UTC 时间戳`；例如，`docdb-yyyy-mm-dd-hh-mm-ss`。

- b. 实例类— 新集群的实例类。您可以接受默认实例类或从下拉列表中选择实例类。
 - c. 实例的数量— 您希望使用此集群创建的实例的数量。您可以接受默认 3 个实例（1 个读/写主实例和 2 个只读副本），或从下拉列表中选择实例数。
5. 如果您对集群配置满意，请选择 Restore cluster (还原集群) 并等待集群还原。
6. 如果您更希望更改某些配置（如指定非默认 Amazon VPC 或安全组），请选择显示高级设置从页面左下角开始，然后继续执行以下步骤。
 - a. 完成 Network settings (网络设置) 部分。
 - Virtual Private Cloud (VPC)— 接受当前 VPC，或者从下拉列表中选择一个 VPC。
 - 子网组— 接受 default 子网组，或者从下拉列表中选择一个。
 - VPC 安全组— 接受 default (VPC) 安全组，或者从列表选择一个。
 - b. 完成集群选项部分。
 - 数据库端口— 接受默认端口，27017 或者使用向上或向下箭头来设置要用于应用程序连接的端口。
 - c. 完成加密部分。
 - 静态加密— 如果您的快照已加密，那么这些选项对您不可用。如果它未加密，您可以选择以下选项之一：
 - 要加密所有集群的数据，请选择启用静态加密。如果您选择此选项，则必须指定一个主密钥。
 - 要让集群的数据不加密，请选择禁用静态加密。如果您选择此选项，您便已完成加密部分。
 - 主密钥— 从下拉列表中选择以下选项之一：
 - (默认) aws/rds— 账号和 Amazon KMS 此选项的后面列出密钥 ID。
 - 客户管理的密钥— 此选项仅当您在创建 IAM 加密密钥后可用。Amazon Identity and Access Management (IAM) 控制台。您可以选择该密钥来加密集群。
 - 输入键 ARN— 在进行筛选框中，输入您的 Amazon 资源名称 (ARN) Amazon KMS 键。ARN 的格式为 `arn:aws:kms:<region>:<accountID>:key/<key-id>`。
 - d. 完成 Log exports (日志导出) 部分。
 - 选择要发布到 CloudWatch 的日志类型— 选择下列项之一：
 - Enabled (已启用)— 使您的集群能够将 DDL 日志记录导出到 Amazon CloudWatch Logs。
 - Disabled— 阻止您的集群将 DDL 日志导出到 Amazon CloudWatch Logs。Disabled (已禁用) 为默认值。
 - IAM 角色— 从列表中，选择 RDS 服务相关角色。
 - e. 完成 Tags (标签) 部分。
 - 添加标签— 在密钥框中，输入集群标签的名称。在 Value (值) 框中，可以选择输入标签值。标签用于 Amazon Identity and Access Management (IAM) 策略来管理对 Amazon DocumentDB 资源的访问并控制可将什么操作应用于资源。
 - f. 完成 Deletion protection (删除保护) 部分。
 - 启用删除保护— 防止意外删除集群。启用该选项后，您将无法删除集群。
7. 选择 Restore cluster (还原集群)。

使用 Amazon CLI 从集群快照进行还原

要使用 Amazon CLI 从快照还原集群，请使用带以下参数的 `restore-db-cluster-from-snapshot` 操作。有关更多信息，请参阅 [RestoreDBClusterFromSnapshot \(p. 645\)](#)。

- **--db-cluster-identifier** – 必需。操作创建的集群的名称。在执行此操作之前，不能存在此名称的集群。

集群命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个 Amazon DocumentDB 的每个 Amazon Neptune 和 Amazon DocumentDB 的所有集群 (必须是唯一的) Amazon Web Services 账户每区域数。
- **--snapshot-identifier** – 必需。用于自其还原的快照的名称。具有此名称的快照必须存在并且处于可用 状态。
- **--engine** – 必需。必须是 docdb。
- **--kms-key-id** – 可选。在还原已加密快照或在从未加密快照还原的同时加密集群时，要使用的 Amazon KMS 密钥标识符的 ARN。如果提供 Amazon KMS 密钥 ID，则会导致使用 Amazon KMS 密钥加密还原的集群 (无论快照是否加密)。

--kms-key-id 的格式为 arn:aws:kms:<region>:<accountID>:key/<key-id>。如果不为 --kms-key-id 参数指定值，则会出现以下情况：

- 如果 --snapshot-identifier 中的快照已加密，将使用用于加密快照的同一 Amazon KMS 密钥加密还原的集群。
- 如果 --snapshot-identifier 中的快照未加密，还原的集群也不会加密。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文。

```
{  
  "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-ABCDEFGHIJKLMNQRSTUWXYZ",
"DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-restore",
"AssociatedRoles": [],
"ClusterCreateTime": "2020-04-01T01:43:40.871Z",
"DeletionProtection": true
}
}
```

在集群状态为可用 后，请为集群创建至少一个实例。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文。

```
{
  "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-ABCDEFGHIJKLMNQRSTUWXYZ",
    "CACertificateIdentifier": "rds-ca-2019",
    "PromotionTier": 2,
    "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-cluster-restore-
instance"
  }
}
```

还原到某个时间点

使用 Amazon Web Services Management Console 或 Amazon Command Line Interface (Amazon CLI) 可将集群还原到该集群的备份保留期内的任何时间点。

Note

您无法将 3.6 集群的时间点还原到 4.0 集群，但可以从一个群集版本迁移到另一个群集版本。有关更多信息，请转至 [迁 Amazon DocumentDB \(p. 112\)](#)。

在将集群还原至某个时间点时，请牢记以下几点。

- 创建新集群时将使用与源集群相同的配置，区别在于创建新集群时将使用默认参数组。要将新集群的参数组设置为源集群的参数组，请在新集群可用后对其进行修改。有关修改集群的更多信息，请参阅 [修改 Amazon DocumentDB 集群 \(p. 259\)](#)。

使用 Amazon Web Services Management Console 还原到时间点

可以使用 Amazon Web Services Management Console 完成以下操作，以将集群还原到该集群的备份保留期内的某个时间点。

- 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
- 在导航窗格中，选择 Clusters。在集群列表中，选择要还原的集群左侧的按钮。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在 Actions (操作) 菜单上，选择 Restore to point in time (还原到时间点)。
4. 完成 Restore time (恢复时间) 部分，用于指定将还原到的日期和时间。
 - a. 还原日期— 选择或输入的日期介于最早还原时间和最晚还原时间。
 - b. 恢复时间— 选择或输入介于最早还原时间和最晚还原时间。
5. 填写 Configuration (配置) 部分。
 - a. 集群标识符— 接受默认标识符，或者输入您喜欢的标识符。

集群命名约束：

 - 长度为 [1-63] 个字母、数字或连字符。
 - 第一个字符必须是字母。
 - 不能以连字符结束或包含两个连续连字符。
 - 对于每个 Amazon DocumentDB 的所有集群 (在、和 Neptune) 必须是唯一的。Amazon Web Services 账户每区域数。
 - b. 实例类— 从下拉列表中，选择要用于集群实例的实例类。
 - c. 实例的数量— 从下拉列表中，选择还原集群时您希望创建的实例的数量。
6. 可选。要配置网络设置、集群选项和启用日志导出，请选择 Show advanced settings (显示高级设置)，然后完成以下各部分。否则，请继续下一步。
 - Network settings (网络设置)
 1. Virtual Private Cloud (VPC)— 从下拉列表中，选择要用于此集群的 VPC。
 2. 子网组— 从下拉列表中，为此集群选择子网组。
 3. VPC 安全组— 从下拉列表中，为此集群选择 VPC 安全组。
 - Cluster options (集群选项)
 1. 端口— 接受默认端口 (27017)，或者使用向上或向下箭头来设置此集群进行通信的端口。
 - 日志导出
 1. 审核日志— 选择此选项可启用将审计日志导出到 Amazon CloudWatch Logs 的功能。如果您选择此选项，则必须在集群的自定义参数组中启用 audit_logs。有关更多信息，请参阅[审计 Amazon DocumentDB \(p. 185\)](#)。
 2. Profiler 日志— 选择此选项可启用将操作分析器日志导出到 Amazon CloudWatch Logs 的功能。如果您选择此选项，还必须在集群的自定义参数组中修改以下参数：
 - profiler— 设置为 enabled。
 - profiler_threshold_ms— 设置为值 [0-INT_MAX] 以设置分析操作的阈值。
 - profiler_sampling_rate— 设置为值 [0.0-1.0] 以设置要分析的缓慢操作的百分比。有关更多信息，请参阅[分析 Amazon DocumentDB 操作 \(p. 388\)](#)。
 3. Profiler 日志— 将分析器日志导出到 Amazon CloudWatch
 4. IAM 角色— 从下拉列表中，选择 RDS 服务相关角色。
 - 标签
 1. 添加标签— 在密钥框中，输入集群标签的名称。在 Value (值) 框中，可以选择输入标签值。标签用于 Amazon Identity and Access Management (IAM) 策略来管理对 Amazon DocumentDB 资源的访问并控制可将什么操作应用于资源。

- 删除保护

1. 启用删除保护— 防止意外删除集群。启用该选项后，您将无法删除集群。

7. 要还原集群，请选择 Create cluster (创建集群)。或者，您可以选择 Cancel (取消) 以取消操作。

使用 Amazon CLI 还原到时间点

要使用快照的备份保留期将集群还原到某个时间点，请使用带有以下参数的 `restore-db-cluster-to-point-in-time` 操作。

- **--db-cluster-identifier**— 必需。要创建的新集群的名称。在执行该操作之前，不能存在此集群。参数值必须满足以下约束。

集群命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
 - 第一个字符必须是字母。
 - 不能以连字符结束或包含两个连续连字符。
 - 对于每个 Amazon DocumentDB 的所有集群（在、和 Neptune）必须是唯一的。Amazon Web Services 账户每区域数。
- **--restore-to-time**— 要将集群还原到的 UTC 日期和时间。例如 2018-06-07T23:45:00Z。

时间约束：

- 必须在集群的最新可还原时间之前。
 - 如果未提供 **--use-latest-restorable-time** 参数，则必须指定。
 - 如果 **--use-latest-restorable-time** 参数为 `true`，则无法指定。
 - 如果 **--restore-type** 参数值为 `copy-on-write`，则无法指定。
- **--source-db-cluster-identifier**— 要从中还原的源集群的名称。该集群必须存在且可用。
 - **--use-latest-restorable-time** 要么 **--no-use-latest-restorable-time**— 是否还原至最近的可还原备份时间。如果未提供 **--restore-to-time** 参数，则无法指定。

Amazon CLI 操作 `restore-db-cluster-to-point-in-time` 仅还原集群，而不还原该集群的实例。您必须调用 `create-db-instance` 操作为还原的集群创建实例，并在 **--db-cluster-identifier** 中指定还原的集群的标识符。仅当完成 `restore-db-cluster-to-point-in-time` 操作并且还原的集群可用后，您才能创建实例。

Example

以下示例从快照 `sample-cluster-snapshot` 创建到最近可还原时间的 `sample-cluster-restored`。

对于 Linux、macOS 或 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
```

对于 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

以下示例从快照 `sample-cluster-snapshot` 创建到 2018 年 12 月 11 日 03:15 (UTC) (在 `sample-cluster` 的备份保留期内) 的 `sample-cluster-restored`。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文。

```
{
  "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
  }
}
```

删除集群快照

手动快照是仅在您使用 Amazon Web Services Management Console 或 Amazon CLI 手动删除时才会被删除的完整备份。无法手动删除自动快照，因为自动快照仅在快照保留期已过或删除快照的集群时才会被删除。

使用 Amazon Web Services Management Console 删除集群快照

要使用 Amazon Web Services Management Console 删除手动集群快照，请完成以下步骤。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择快照。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在快照列表中，选择要删除的快照左侧的按钮。快照的类型必须为手动。
 1. 你可以验证快照的类型是手册通过检查它是否被列为 `manual` 要么 `automatic` 在类型 column。
4. 从 Actions 菜单中选择 Delete。如果 Delete (删除) 选项不可用，您可能选择了自动快照。
5. 要删除快照，请在删除确认屏幕中选择 Delete (删除)。要保留快照，请选择 Cancel (取消)。

使用 Amazon CLI 删除集群快照

Amazon DocumentDB 手动集群快照是完整备份，您可以使用 Amazon CLI。您不能手动删除自动快照。

要使用 Amazon CLI 删除手动集群快照，请使用带以下参数的 `delete-db-cluster-snapshot` 操作。

参数

- `--db-cluster-snapshot-identifier` - 必需。要删除的手动快照的名称。

以下示例将删除集群快照 `sample-cluster-snapshot`。

对于 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster-snapshot \
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

对于 Windows：

```
aws docdb delete-db-cluster-snapshot ^
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

此操作的输出将列出您删除的集群快照的详细信息。

管理 Amazon DocumentDB 资源

以下各节介绍各种组件及其用于管理 Amazon DocumentDB 实现的相关任务。

主题

- [Amazon DocumentDB 操作任务概述](#) (p. 222)
- [Amazon DocumentDB 全球集群概览](#) (p. 227)
- [管理 Amazon DocumentDB](#) (p. 243)
- [管理 Amazon DocumentDB 实例](#) (p. 282)
- [管理 Amazon DocumentDB 子网组](#) (p. 303)
- [Amazon DocumentDB 高可用性和复制](#) (p. 311)
- [管理 Amazon DocumentDB 事件](#) (p. 316)
- [选择区域和可用区](#) (p. 320)
- [管理 Amazon DocumentDB 集群参数组](#) (p. 322)
- [了解 Amazon DocumentDB 终端节点](#) (p. 343)
- [了解 Amazon DocumentDB Amazon 资源名称 \(ARN\)](#) (p. 349)
- [为 Amazon DocumentDB 资源添加标签](#) (p. 352)
- [维护 Amazon DocumentDB](#) (p. 356)
- [了解服务相关角色](#) (p. 361)

Amazon DocumentDB 操作任务概述

本部分讲述 Amazon DocumentDB (具有 MongoDB 兼容性) 集群的操作任务以及如何使用完成这些任务。Amazon CLI.

主题

- [给 Amazon DocumentDB 集群添加副本](#) (p. 222)
- [描述集群和实例](#) (p. 223)
- [创建集群快照](#) (p. 224)
- [从快照还原](#) (p. 225)
- [从集群中删除实例](#) (p. 226)
- [删除集群](#) (p. 226)

给 Amazon DocumentDB 集群添加副本

在您创建您的 Amazon DocumentDB 集群的主实例后，您可以添加一个或多个实例。复制品. 副本是只读实例，它有两个用途：

- 可扩展性— 如果您有大量需要同时访问的客户端，则可以添加多个副本以进行读取扩展。

- 高可用性— 如果主实例发生故障，Amazon DocumentDB 会自动故障转移到一个副本实例并将该副本实例指定为新的主实例。如果副本发生故障，则集群中的其他实例仍能够处理请求，直到发生故障的节点恢复为止。

每个 Amazon DocumentDB 集群可支持多达 15 个副本。

Note

为实现最大容错能力，您应在不同的可用区中部署副本。这可以确保您的 Amazon DocumentDB 集群即使在整个可用区变得不可用时也能够继续正常运行。

以下 Amazon CLI 示例说明如何添加新副本。--availability-zone 参数将副本置于指定的可用区中。

```
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
```

描述集群和实例

以下 Amazon CLI 示例列出区域中的所有 Amazon DocumentDB 集群。对于某些管理功能（如集群和实例周期管理），Amazon DocumentDB 利用与 Amazon RDS 共享的操作技术。这些区域有：filterName=engine,Values=docdb 筛选器参数仅返回 Amazon DocumentDB 集群。

有关描述和修改集群的更多信息，请参阅[Amazon DocumentDB 集 \(p. 246\)](#)。

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

此操作的输出将类似于下文。

```
{  
  "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",
      ...
    }
  ]
}

```

以下 Amazon CLI 示例列出 Amazon DocumentDB 集群中的实例。有关描述和修改集群的更多信息，请参阅 [Amazon DocumentDB 实例生命周期](#) (p. 288)。

```

aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].DBClusterMembers'

```

输出如下所示。此输出中有两个实例。主实例是 sample-instance-1 ("IsClusterWriter": true)。此外，还有一个副本实例，即 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
      }
    ]
  ]
]

```

创建集群快照

一个集群快照是对您的 Amazon DocumentDB 集群中的数据进行完整备份。创建快照后，Amazon DocumentDB 将直接从集群卷读取数据。因此，即使您的集群当前没有任何实例在运行，您也可以创建快照。创建快照所用时间因集群卷大小而异。

Amazon DocumentDB 支持自动备份，自动备份在每天的首选备份时段（白天的一个 30 分钟时段进行。以下 Amazon CLI 示例显示如何查看您的集群的备份时段：

```

aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].PreferredBackupWindow'

```

此输出显示备份时段（UTC 时间）：

```
[
  "00:18-00:48"
]
```

您可以在创建 Amazon DocumentDB 集群时定义备份时段。您还可以更改备份时段，如下示例所示：如果您不定义备份时段，Amazon DocumentDB 会自动给您的集群分配一个时段。

```
aws docdb modify-db-cluster \
  --db-cluster-identifier sample-cluster \
  --preferred-backup-window "02:00-02:30"
```

除了自动备份以外，您还可以随时手动创建集群快照。当您执行此操作时，您指定要备份的集群以及快照的唯一名称，以便稍后从此快照还原。

以下 Amazon CLI 示例说明如何创建数据快照。

```
aws docdb create-db-cluster-snapshot \
  --db-cluster-identifier sample-cluster \
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

从快照还原

您可以将集群快照还原到新的 Amazon DocumentDB 集群。为此，您需要提供快照的名称和新集群的名称。您无法从快照还原到现有集群；还原时，Amazon DocumentDB 将新建一个集群，然后使用快照数据填充到该集群中。

以下示例显示集群 `sample-cluster` 的所有快照。

```
aws docdb describe-db-cluster-snapshots \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotType,Status]'
```

输出如下所示。手动快照就是您手动创建的快照，而自动快照则由 Amazon DocumentDB 在集群备份时段内由创建。

```
[
  [
    "sample-cluster-snapshot",
    "manual",
    "available"
  ],
  [
    "rds:sample-cluster",
    "automated",
    "available"
  ]
]
```

以下示例显示如何从快照还原 Amazon DocumentDB 集群。

```
aws docdb restore-db-cluster-from-snapshot \
  --engine docdb \
  --db-cluster-identifier new-sample-cluster \
  --snapshot-identifier sample-cluster-snapshot
```

新的集群没有任何实例与之相关联；因此，如果您想要与集群进行交互，您必须给它添加实例。

```
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
```

您可以使用以下 Amazon CLI 操作来监控集群和实例的创建进度。当集群和实例处于可用状态时，您可连接到新集群的终端节点并访问您的数据。

```
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]'
```

从集群中删除实例

Amazon DocumentDB 将所有数据都存储在集群卷中。即使您从集群中删除所有实例，数据仍会保留在该集群卷中。如果您需要再次访问数据，则可随时向该集群中添加实例，并在停止处恢复操作。

以下示例显示如何从 Amazon DocumentDB 集群中删除实例。

```
aws docdb delete-db-instance \  
  --db-instance-identifier sample-instance
```

删除集群

您必须先删除其所有实例，然后才能删除 Amazon DocumentDB 集群。以下 Amazon CLI 示例返回关于集群中实例的信息。如果此操作会返回任何实例标识符，您必须删除各个实例。有关更多信息，请参阅[从集群中删除实例](#) (p. 226)。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].DBClusterMembers[*].DBInstanceIdentifier'
```

当没有任何剩余的实例时，您就可以删除集群了。此时，您必须选择以下选项之一：

- 创建最终快照— 将所有集群数据捕获到一个快照中，以便您可以使用该数据重新创建一个新实例。下例说明具体做法：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --final-db-snapshot-identifier sample-cluster-snapshot
```

- 跳过最终快照— 永久舍弃所有集群数据。此操作无法撤消。下例说明具体做法：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --skip-final-snapshot
```

Amazon DocumentDB 全球集群概览

什么是全局集群？

全局集群由一个主区域和最多五个只读辅助区域组成。您可将写入操作直接发送到主区域中的主集群，Amazon DocumentDB 使用专用基础设施自动发送到辅助区域。延迟通常不到一秒。

全局集群有何用处？

- 从区域范围的停机中恢复— 如果发生区域范围的停机，您可以在几分钟内将其中一个辅助集群升级为主集群，典型的恢复时间目标 (RTO) 不到一分钟。恢复点目标 (RPO) 通常以秒为单位进行测量，但这取决于发生故障时整个网络的滞后时间。
- 具有本地延迟的全局读取— 如果您在世界各地设有办事处，则可以使用全局集群在主区域将其主要信息来源保持最新。您其他区域的办事处可以访问各自区域中的信息，存在本地延迟。
- 可扩展的辅助集群— 您可以通过向辅助区域添加更多只读实例来扩展辅助集群。辅助集群为只读模式，因此它最多可以支持 16 个只读副本实例，而非 15 个单集群的通常限制。
- 从主集群到辅助集群的快速复制— 全局集群执行的复制对主数据库集群造成的性能影响不大。数据库实例的资源完全专用于承担应用程序读取和写入工作负载。

全局集群当前的局限性是什么？

- Amazon DocumentDB v3.6 不支持全局集群。
- t3、t4g 和 r4 实例类型不支持全局集群。
- 全局集群可以在以下区域中使用：Amazon GovCloud（美国西部）、南美洲（圣保罗）、欧洲（米兰）、中国（北京）和中国（宁夏）。
- 如果发生区域故障转移，您必须手动将辅助集群提升为主集群，然后修改应用程序以指向新的主集群。
- 只有主集群才能执行写入操作。执行写入操作的客户端连接到主集群的集群终端节点。
- 您的集群最多可以有五个辅助区域和一个主区域。
- 辅助集群无法停止。如果主集群具有与之关联的辅助集群，则无法停止该主集群。只有没有辅助集群的区域集群才能停止。
- 在某些情况下，附加到辅助集群的副本可能会重新启动。如果主区域的实例重新启动或发生故障转移，辅助区域中的副本也会重新启动。随后集群将不可用，直到所有副本与主数据库集群的写入器实例恢复同步。此行为是预期的。在更改主集群之前，请务必了解对全局集群的影响。
- 您无法使用 Amazon CloudFormation 来管理全局集群。
- 您不能在辅助集群上使用更改流。

主题

- [快速入门指南：全球集群 \(p. 227\)](#)
- [管理 Amazon DocumentDB 全局集群 \(p. 236\)](#)
- [Connect 到 Amazon DocumentDB 全局集群 \(p. 240\)](#)
- [监控 Amazon DocumentDB 全局集群 \(p. 241\)](#)
- [灾难恢复和 Amazon DocumentDB 全球集群 \(p. 241\)](#)

快速入门指南：全球集群

主题

- [配置 \(p. 228\)](#)

- [创建Amazon DocumentDB 全局集群 \(p. 228\)](#)
- [添加Amazon Web Services 区域到Amazon DocumentDB B 全球集群 \(p. 233\)](#)
- [对 Amazon DocumentDB 全局集群使用快照 \(p. 236\)](#)

配置

Amazon DocumentDB 全局集群跨越至少两个Amazon Web Services 区域。主区域支持具有一个主（写入）实例和最多十五个副本实例的集群，而辅助区域运行的只读集群最多由十六个副本实例组成。每个全局集群最多可以有五个辅助区域。该表列出了全局集群中允许的集群、实例和副本的最大数量。

描述	主 Amazon Web Services 区域	辅助 Amazon Web Services 区域
集群	1	5 (最大值)
写入器实例	1	0
每个集群的只读实例 (Amazon DocumentDB 副本)	15 (最大值)	16 (合计)
只读实例 (允许的最大值，辅助区域的指定实际数量)	15 - s	s = 辅助 Amazon Web Services 区域的总数

集群有以下特定要求：

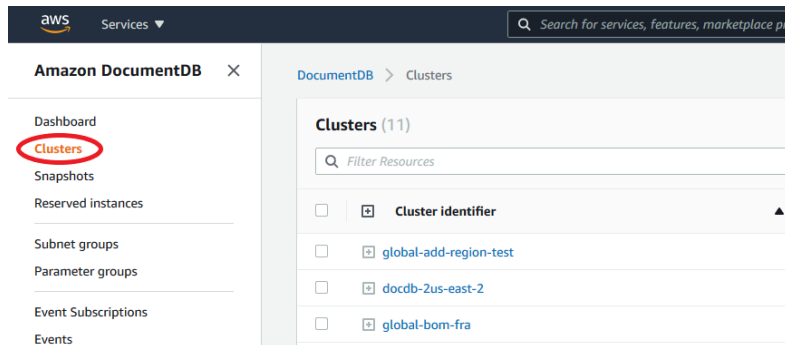
- 数据库实例类要求— 您只能使用db.r5和db.r6实例类。
- Amazon Web Services 区域要求— 主集群必须位于一个区域，并且至少有一个辅助集群必须位于同一账户的不同区域。您最多可以创建五个辅助（只读）集群，它们必须各在不同的区域中。换句话说，没有两个集群可位于同一个区域中。
- 命名要求— 为每个集群选择的名称在所有区域中必须唯一。即使不同的集群位于不同的区域中，也不能对它们使用相同的名称。

创建Amazon DocumentDB 全局集群

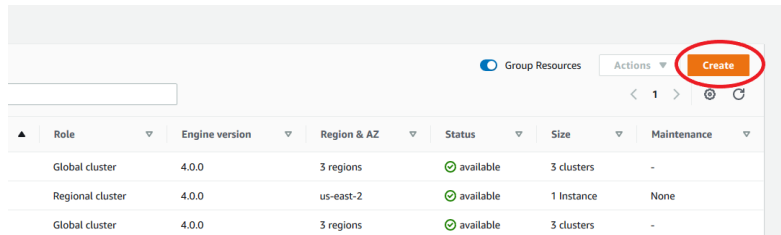
你准备好建立你的第一个全球集群了吗？在本节中，我们将说明如何使用新的数据库集群和实例创建一个全新的全局集群，使用以下任一方法Amazon Web Services Management Console要么Amazon CLI按照以下说明进行操作。

使用 Amazon Web Services Management Console

1. 在里面Amazon Web Services Management Console，导航到Amazon DocumentDB。
2. 当您进入Amazon DocumentDB B 控制台时，选择集群。

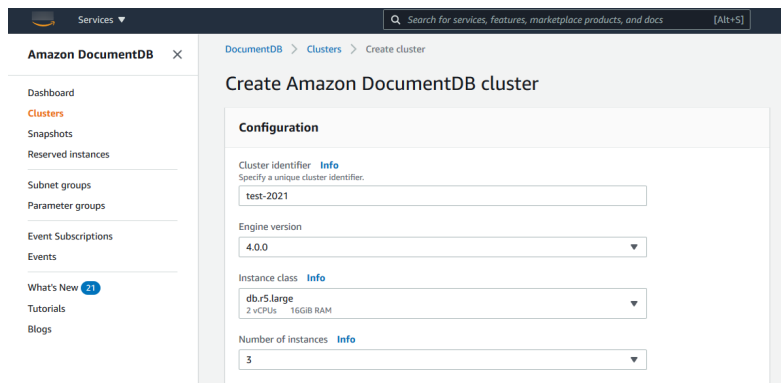


3. 选择Create（创建）。

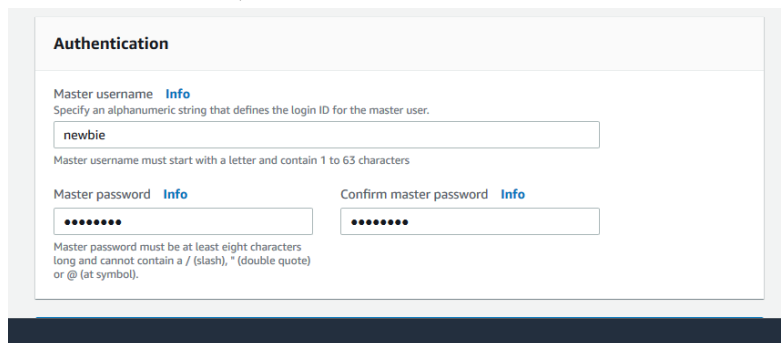


4. 填写配置该部分的创建Amazon DocumentDB相应地形成：

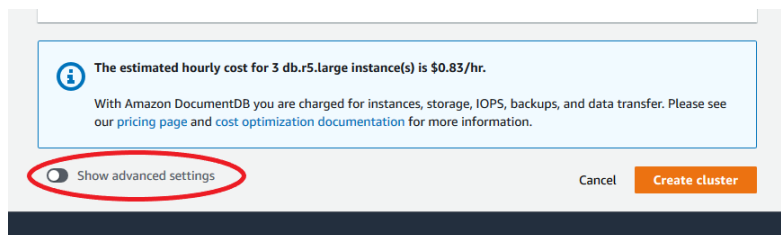
- 集群标识符：您可以输入此实例的唯一标识符，也可以允许 Amazon DocumentDB 根据集群标识符提供实例标识符。
- 引擎版本：选择4.0.0
- 实例类：选择db.r5.large
- 实例数：选择3.



5. 在里面身份验证部分，填写主用户名和主密码。



6. 选择显示高级设置.



7. 在里面网络设置“”

- 保留默认选项虚拟私有云和子网组。

Network settings

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

vpc-d5bf5fbc

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group [Info](#)
A subnet group is a collection of subnets that are within a VPC.

default

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups

default (VPC) X

- 对于VPC 安全组,默认 VPC应该已经添加了。

Network settings

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

vpc-d5bf5fbc

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group [Info](#)
A subnet group is a collection of subnets that are within a VPC.

default

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups

default (VPC) X

- 类型DocDB进入VPC 安全组字段并选择入站 DOCDB (VPC).

Network settings

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

vpc-d5bf5fbc

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group [Info](#)
A subnet group is a collection of subnets that are within a VPC.

default

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups

DocDB-Inbound (VFC) X

8. 对于集群选项和南美洲ncryption-at-rest，保留默认选项。

Cluster options

Port
TCP/IP port that is used to connect to the cluster.

27017

Cluster parameter group [Info](#)

default.docdb4.0

Encryption-at-rest

Encryption-at-rest [Info](#)

☒ Enable encryption

☐ Disable encryption

Master key

(default) aws/rds

Account

827630067164

KMS key ID

5e5db66b-e29d-4cfd-bfe5-585582908728

9. 对于备份和日志导出，保留默认选项。

Backup

Backup retention period [Info](#)
A period between 1 and 35 days in which you can perform a point-in-time restore and for which automated backups are retained.

1 day

Backup window
The daily time range (in UTC) during which automated backups are created.

Start time : 00 : 00 UTC Duration : 0.5 hours

Log exports

Select the log types to publish to Amazon CloudWatch Logs

☐ Audit logs

☐ Profiler logs

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS Service Linked Role

To enable auditing, ensure that both exporting auditing logs to Amazon CloudWatch is enabled and the Cluster Parameter "Auditing" is enabled. [Learn more](#)

10. 对于维护,标签和删除保护，保留默认选择。

The screenshot shows three sections of the Amazon DocumentDB console:

- Maintenance:** Includes a "Maintenance window" section with an "Info" link and a description: "The period in which pending modifications or patches are applied to instances in the cluster." There are two radio buttons: "Select window" (unselected) and "No preference" (selected).
- Tags:** Shows "No tags" and an "Add tag" button.
- Deletion protection:** Includes a checkbox "Enable deletion protection" which is checked. Below it is a description: "Protects the cluster from being accidentally deleted. While this option is enabled, you can't delete the cluster."

11. 现在点击上面写着的按钮Create.

The screenshot shows a summary box with the following information:

- An information icon and text: "The estimated hourly cost for 3 db.r5.large instance(s) is \$0.83/hr."
- A paragraph: "With Amazon DocumentDB you are charged for instances, storage, IOPS, backups, and data transfer. Please see our [pricing page](#) and [cost optimization documentation](#) for more information."
- A toggle switch for "Show advanced settings" which is currently turned off.
- At the bottom right, there are two buttons: "Cancel" and "Create cluster". The "Create cluster" button is highlighted with a red circle.

使用 Amazon CLI

要创建 Amazon DocumentDB 区域集群，请调用`create-db-cluster` Amazon CLI。以下Amazon CLI命令创建一个名为 Amazon DocumentDB 集群`global-cluster-id`。有关删除保护的更多信息，请参见[删除 Amazon DocumentDB 集群](#) (p. 267)。

以及`--engine-version`是一个可选参数，默认为最新的主引擎版本。当前的主要引擎版本是4.0.0。发布新的主要引擎版本时，的默认引擎版本`--engine-version`将进行更新以反映最新的主要引擎版本。因此，对于生产工作负载，尤其是那些依赖脚本编写、自动化或Amazon CloudFormation模板，我们建议您明确指定`--engine-version`到预期的主版本。

如果 `adb-subnet-group-name` 要么 `vpc-security-group-id` 未指定，Amazon DocumentDB 将使用给定区域的默认子网组和 Amazon VPC 安全组。

在以下示例中，将每个 `#####` 替换为您自己的信息。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "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-KQSGI4MHU4NTDDRVLNTU7XVAY",
    "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
    "Port": 27017,
    "Status": "creating",
    "ReaderEndpoint": "cluster-id.cluster-ro-sfcrcljcoroz.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-sfcrcljcoroz.us-east-1.docdb.amazonaws.com"
  }
}
```

创建集群需要几分钟时间。您可以使用 Amazon Web Services Management Console 或 Amazon CLI 监控集群的状态。有关更多信息，请参阅 [监控 Amazon DocumentDB 集群的状态 \(p. 366\)](#)。

Important

当您使用以下应用程序时：Amazon CLI 要创建 Amazon DocumentDB 区域集群，不创建任何实例。因此，您必须显式创建主实例和所需的任何副本实例。您可以使用控制台或 Amazon CLI 来创建实例。有关更多信息，请参阅 [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#) 和 [CreateDBCluster \(p. 542\)](#) Amazon DocumentDB API 参考中。

您的区域集群可用后，您可以按照以下说明在其他区域添加辅助集群：[添加 Amazon Web Services 区域到 Amazon DocumentDB B 全球集群 \(p. 233\)](#)。当您添加区域时，您的区域集群将成为您的主集群，并且您在此所选区域中有一个新的辅助集群。

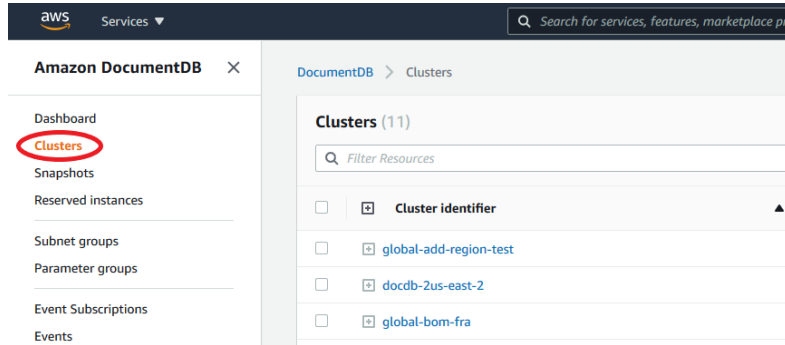
添加 Amazon Web Services 区域到 Amazon DocumentDB B 全球集群

一个全局集群需要在与主集群不同的区域中至少有一个辅助集群，并且您最多可以添加五个辅助集群。请注意，对于添加的每个辅助集群，必须将主集群中允许的副本数量减少一个。例如，如果您的全局集群有五个

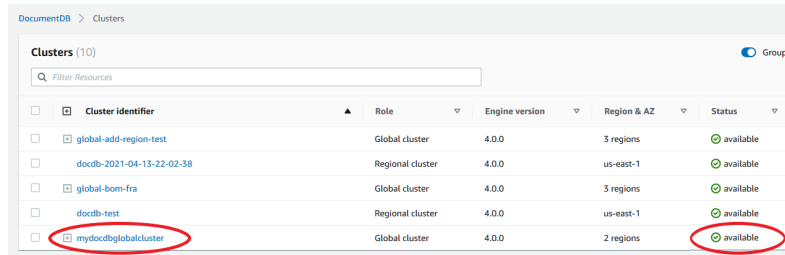
辅助区域，则主集群只能有十个（而不是十五个）副本。有关更多信息，请参阅。[Amazon DocumentDB 全局集群的配置要求](#)。

使用 Amazon Web Services Management Console

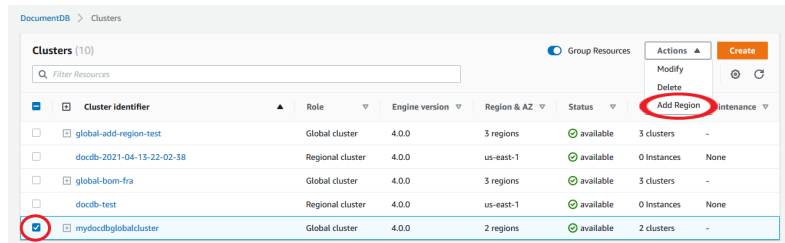
1. 登录Amazon Web Services Management Console然后打开Amazon DocumentDB B 控制台。
2. 在导航窗格中，选择 Clusters。



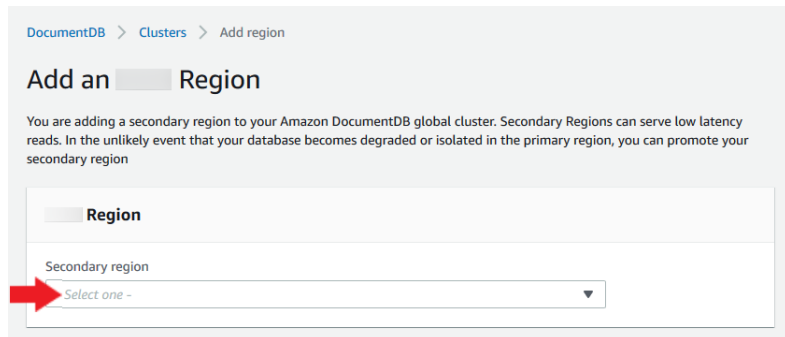
3. 选择要向其中添加辅助集群的集群。确保集群是Available。



4. 选择下拉菜单操作然后选择添加区域。



5. 在添加区域页面上，选择辅助区域。请注意，您不能为同一个全局集群选择在该区域中已有辅助集群的区域。此外，该区域也不能是主集群所在的同一个区域。如果这是您要添加的第一个区域，则还必须指定您选择的全局集群标识符。



6. 在新区域中填写辅助集群的其余字段，然后选择创建集群。完成添加区域后，它会出现以下列表中：集群在里面Amazon Web Services Management Console。

Configuration

Global Cluster Id
firstregion

Cluster identifier [Info](#)
Specify a unique cluster identifier.
docdb-2021-04-26-20-09-49

Instance class [Info](#)
db.r5.large
2 vCPUs 16GiB RAM

Number of instances [Info](#)
3

The estimated hourly cost for 3 db.r5.large instance(s) is \$0.83/hr.
With Amazon DocumentDB you are charged for instances, storage, IOPS, backups, and data transfer. Please see our [pricing page](#) and [cost optimization documentation](#) for more information.

☐ Show advanced settings

Cancel **Create cluster**

使用 Amazon CLI

- 使用 `create-db-cluster` 带有名称的 CLI 命令 (`--global-cluster-identifier`) 您的全局集群。对于其他参数，请执行以下操作：
 - 对于 `--region`，请选择不同的 Amazon Web Services 区域而不是你的主要地区。
 - 为以下项选择特定的值 `--engine` 和 `--engine-version` 参数。
 - 对于加密集群，请将主 Amazon Web Services 区域 指定为 `--source-region` 以进行加密。

以下示例创建了一个新 Amazon DocumentDB 集群，并将其作为只读辅助集群挂载到全局集群。在最后一步中，实例将添加到新集群。

在以下示例中，将每个 `#####` 替换为您自己的信息。

对于 Linux、macOS 或 Unix：

```
aws docdb --region secondary-region-id \  
  create-db-cluster \  
    --db-cluster-identifier cluster-id \  
    --global-cluster-identifier global-cluster-id \  
    --engine-version version  
  
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
```

对于 Windows：

```
aws docdb --region secondary-region-id ^  
  create-db-cluster ^  
    --db-cluster-identifier cluster-id ^  
    --global-cluster-identifier global-cluster-id ^  
    --engine-version version
```

```
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
```

对 Amazon DocumentDB 全局集群使用快照

您可以还原 Amazon DocumentDB 集群的快照以作为全局集群的起点。为此，您必须还原快照并创建新集群。这将作为您的全局集群的主集群。然后，您可以向恢复的集群添加另一个区域，从而将其转换为全局集群。

管理 Amazon DocumentDB 全局集群

您可以对构成全局集群的各个集群执行大多数的管理操作。当您选择集团相关资源在集群在控制台中，您可以看到主集群和辅助集群分组到关联的全局集群之下。

这些区域有：配置全局集群的选项卡显示Amazon Web Services 区域集群的运行位置、版本和全局集群标识符。

主题

- [修改 Amazon DocumentDB 全局集群 \(p. 236\)](#)
- [修改 Amazon DocumentDB 全局集群的参数 \(p. 236\)](#)
- [从 Amazon DocumentDB 全局集群删除集群 \(p. 236\)](#)
- [从 Amazon DocumentDB 全局集群删除集群 \(p. 238\)](#)
- [在辅助区域中创建无管控 Amazon DocumentDB 集群 \(p. 239\)](#)

修改 Amazon DocumentDB 全局集群

这些区域有：集群页面中的Amazon Web Services Management Console列出您所有的全局集群，同时显示每个全局集群的主集群和辅助集群。全局集群有自己的配置设置。具体来说，它具有与其主集群和辅助集群关联的区域。

当您对全局集群进行更改时，您有机会取消更改。

选择 Continue (继续) 时，即表示您确认更改。

修改 Amazon DocumentDB 全局集群的参数

您可以为全局集群中的每个集群独立配置集群参数组。大多数参数的工作方式与其他类型的 Amazon DocumentDB 集群相同。我们建议您在全局数据库中使所有集群之间的设置保持一致。在将辅助集群提升为主集群时，此操作有助于避免意外的行为变化。

例如，对于时区和字符集使用相同设置，可避免在不同集群作为主集群时出现不一致的行为。

从 Amazon DocumentDB 全局集群删除集群

有几种情况下，您可能希望从全局集群中删除集群。例如，如果主集群变为降级或隔离后，您可能希望从全局集群中移除集群。然后，它将成为独立的预置集群，可用于创建新的全局集群。要了解更多信息，请参阅手动从计划外停机中恢复全局集群。

您也可能会想要删除集群，因为您想要删除不再需要的全局集群。在分离所有关联集群之后，您才能删除全局集群，仅保留主集群。有关更多信息，请参阅删除 Amazon DocumentDB 全局集群。

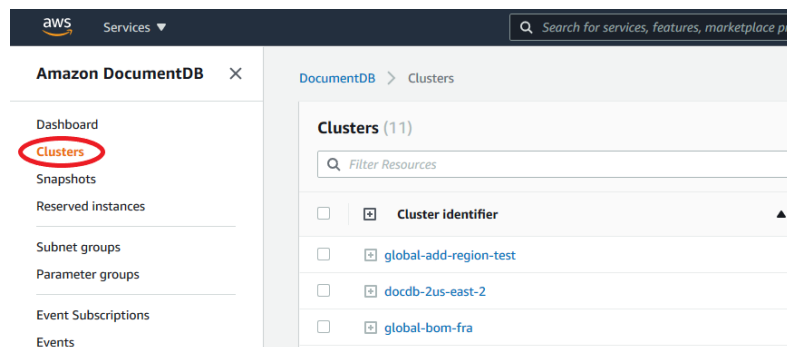
Note

当集群从全局集群中分离时，它将不再与主集群同步。它将成为具有完全读/写功能的独立预置集群。此外，它在 Amazon DocumentDB 控制台中不可见。只有在控制台中选择集群所在的区域时，才可见。

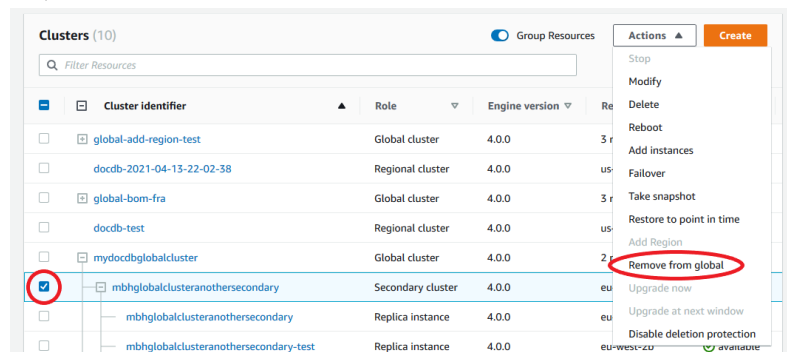
您可以使用从全局集群中移除集群 Amazon Web Services Management Console，Amazon CLI 或 RDS API。

使用 Amazon Web Services Management Console

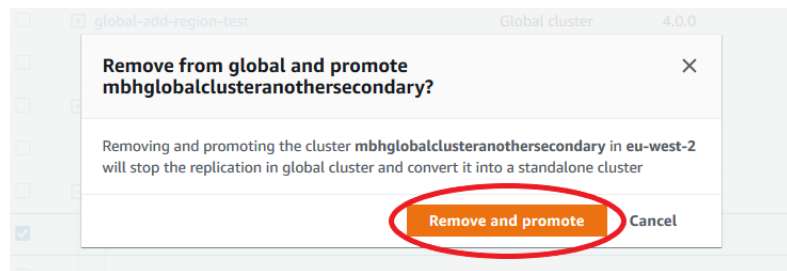
1. 登录到 Amazon Web Services Management Console 然后导航到 Amazon DocumentDB 控制台。
2. 选择集群在左侧导航。



3. 扩展全局集群，以便您可以查看所有辅助集群。选择要删除的辅助群集。选择操作，然后在下拉菜单中，选择从全局中删除。



4. 系统会显示一条提示，要求您确认是否要将辅助集群从全局集群中分离。选择删除和升级从全局集群中删除集群。



现在，该集群不再作为辅助集群，也不再与主集群同步。这个集群为一个有完全读/写功能的独立集群。

在移除或删除所有辅助集群后，您可以按同样方式移除主集群。在删除所有辅助集群之前，您无法从全局集群中分离或删除主集群。全局集群可能保留在集群列表中，其中有零个区域和可用区。如果不想再使用此全局集群，则可以删除。

使用 Amazon CLI

要从全局集群中删除集群，请运行`remove-from-global-cluster`带有下列参数的 CLI 命令：

- `--global-cluster-identifier`— 全局集群的名称（标识符）。
- `--db-cluster-identifier`— 要从全局集群中移除的每个集群的名称。

以下示例先从全局集群中移除辅助集群，然后移除主集群。

对于 Linux、macOS 或 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
```

重复`remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN`命令适用于全局集群中的每个辅助区域。

对于 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
```

重复`remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN`命令适用于全局集群中的每个辅助区域。

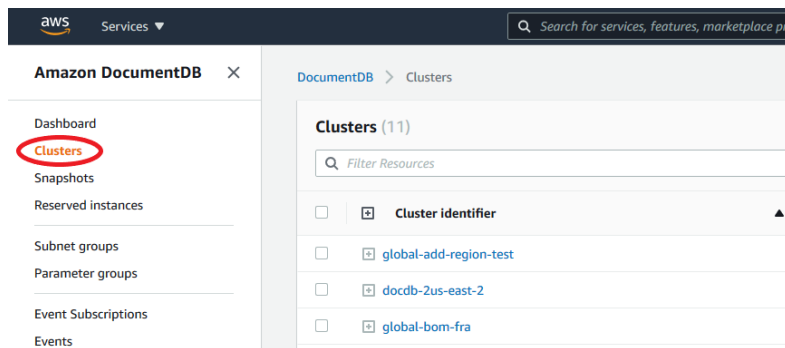
从 Amazon DocumentDB 全局集群删除集群

要删除全局集群，请执行以下操作：

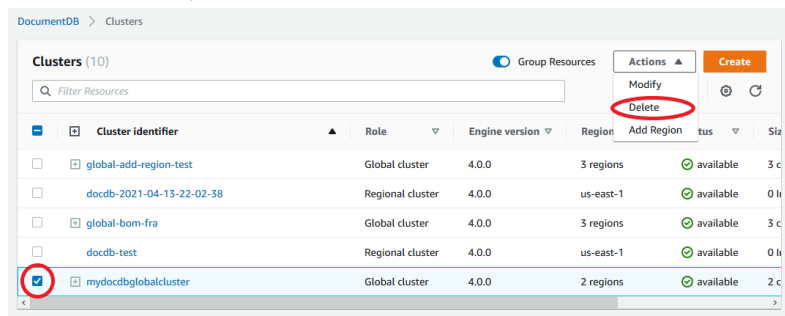
- 从全局集群中删除所有辅助集群。每个集群都会变成独立的集群。请参阅上一部分“移除全局集群”。
- 从每个独立的集群中，删除所有副本。
- 从全局集群中移除主集群。这将成为独立的集群。
- 从主集群中，首先删除所有副本，然后删除主实例。从新近独立集群中删除主实例通常还会同时删除集群和全局集群。

使用 Amazon Web Services Management Console

1. 登录到Amazon Web Services Management Console然后导航到 Amazon DocumentDB 控制台。
2. 选择集群然后找到要删除的全局集群。



3. 选择全局集群后，选择Delete来自的操作菜单。



确认所有集群从全局集群中移除。全局集群应显示零个区域和可用区且集群的大小为零。如果全局集群包含任何集群，则无法将其删除。你首先必须按照上一步中的说明进行操作，移除全局集群。

使用 Amazon CLI

要删除全局集群，请运行delete-global-cluster带有名称的 CLI 命令Amazon Web Services 区域以及全局集群标识符，如以下示例所示。

对于 Linux、macOS 或 Unix：

```
aws docdb --region primary_region delete-global-cluster \
  --global-cluster-identifier global_cluster_id
```

对于 Windows：

```
aws docdb --region primary_region delete-global-cluster ^
  --global-cluster-identifier global_cluster_id
```

在辅助区域中创建无管控 Amazon DocumentDB 集群

尽管 Amazon DocumentDB 全局群集至少需要另一个辅助群集中至少有一个辅助群集Amazon Web Services 区域与主集群相比，您可以对辅助集群使用无管控配置。无管控辅助 Amazon DocumentDB 集群是没有实例的集群。此类型的配置可以降低全局集群的开支。在 Amazon DocumentDB 集群中，计算和存储是分离的。如果没有实例，您就无需为计算付费，而只需为存储付费。如果设置正确，无管控辅助存储卷将与主集群保持同步。

您可以像平常创建 Amazon DocumentDB 全局集群一样添加辅助集群。但是，在主集群开始复制到辅助集群之后，您将从辅助集群中删除该只读实例。此辅助集群现在被视为“无管控”集群，因为它不再有实例。但是，存储卷与主 Amazon DocumentDB 集群保持同步。

Important

我们只建议那些能够容忍区域范围内故障 15 分钟以上的客户使用无头集群。这是因为使用无头辅助集群从区域范围内的故障中恢复需要用户在故障转移后创建新实例。新实例可能需要大约 10-15 分钟才能使用。

如何向全局集群添加无头辅助集群

1. 登录到Amazon Web Services Management Console然后打开[Amazon DocumentDB 控制台](#)。
2. 选择集群在左侧导航。
3. 选择需要辅助集群的全局集群。确保主集群为Available。
4. 对于 Actions (操作)，选择 Add region (添加区域)。
5. 在存储库的添加区域页面上，选择辅助区域。

Note

您不能为同一个全局集群选择在该区域中已有辅助集群的区域。此外，它不能是主集群所在的同一个区域。

6. 在新区域中填写辅助集群的其余字段。这些配置选项与任何集群实例的配置选项相同。
7. 添加区域。完成将区域添加到全局集群中的操作后，您将会看到它会出现出现在Clusters中的Amazon Web Services Management Console。
8. 继续操作之前，请先使用Amazon Web Services Management Console或者Amazon CLI. 如果你使用Amazon CLI：

```
$ aws docdb describe-db-clusters --db-cluster-identifier secondary-cluster-id --query '*[].[Status]' --output text
```

新添加的辅助集群的状态可能需要几分钟的时间才能从创建到可用。当集群处于可用状态时，您可以删除读取器实例。

9. 在辅助集群中选择读取器实例，然后选择Delete。
10. 删除读取器实例后，辅助集群仍然是全局集群的组成部分。它不应该有与之关联的任何实例。

Note

如果主区域发生计划外停机事件，您可以使用此无管控辅助 Amazon DocumentDB 集群手动恢复 Amazon DocumentDB 全局集群。

Connect 到 Amazon DocumentDB 全局集群

连接到全局集群的方式取决于您是需要写入集群还是从集群读取：

- 对于只读请求或查询，您连接到Amazon Web Services 区域。
- 要运行数据操作语言 (DML) 或数据定义语言 (DDL) 语句，应连接到主集群的集群终端节点。此端点可能与您的应用程序不在同一个 Amazon Web Services 区域 中。

当您在控制台中查看全局集群时，您可以看到与其所有集群关联的所有通用终端节点。

连接到全局集群的方式取决于您是需要写入数据库还是从数据库读取。对于要从主区域提供的 DDL、DML 和读取操作，应连接到主集群。我们建议您使用集群终端节点以副本集模式（读取首选项）连接到主集群secondaryPreferred=true. 这会将写入流量路由到主集群的写入器实例，并将读取流量路由到主集群的副本实例。

对于跨区域、只读流量，您应该连接到一个辅助集群。我们建议您以副本集模式使用集群终端节点连接到辅助集群。由于所有实例都是只读副本实例，因此无需指定读取首选项。为了最大限度地减少延迟，请选择您所在区域或离您最近的区域中的任何读取器终端节点。

监控Amazon DocumentDB 全局集群

Amazon DocumentDB (与 MongoDB 兼容) 与 CloudWatch 因此您可以为您的集群收集和分析运行指标。可用监控这些指标 CloudWatch控制台、Amazon DocumentDB 控制台、Amazon Command Line Interface(Amazon CLI)，或者 CloudWatch API。

要监视全局集群，请使用以下 CloudWatch 指标。

指标	描述
GlobalClusterReplicatedWriteIO	从主集群卷复制的计费写入 I/O 操作的平均数量Amazon Web Services 区域到辅助卷中的群集卷Amazon Web Services 区域，每隔 5 分钟报告一次。已复制的数量ReplicatedWriteIOs到每个次区域的数量与区域内的数量相同VolumeWriteIOPs由主要区域执行。
GlobalClusterDataTransferBytes	从主集群传输的数据量Amazon Web Services 区域到辅助集群Amazon Web Services 区域，以字节为单位。
GlobalClusterReplicationLag	从主集群的复制更改事件时的滞后总量（以毫秒为单位）Amazon Web Services 区域到辅助集群Amazon Web Services 区域

有关如何查看这些指标的更多信息，请参阅[查看 CloudWatch 数据](#)。

灾难恢复和 Amazon DocumentDB 全球集群

通过使用全局集群，您可以快速从灾难（例如区域故障）中恢复。灾难恢复通常以 RTO 和 RPO 的值来衡量。

- 恢复时间目标 (RTO)— 灾难后系统恢复工作状态所需的时间。换言之，RTO 用于衡量停机时间。对于全球集群，RTO 大约为数分钟。
- 恢复点目标 (RPO)— 可能丢失的数据量（按时间衡量）。对于全球集群，RPO 通常以秒为单位进行测量。
- 要从计划外停机中恢复，您可以执行跨区域故障转移到全局集群中的一个辅助区域。当全局集群具有多个辅助区域时，如果是主区域，请确保分离所有辅助区域Amazon Web Services 区域发生中断。然后，您将其中一个辅助区域提升为新的主区域Amazon Web Services 区域。最后，在其他每个辅助区域中创建新的集群，并将这些集群附加到全局集群。
- 将辅助集群提升为主集群时，还需要更新应用程序用于连接到全局集群的终端节点。要从新提升的集群获取新的写入器终端节点，可以通过从终端节点字符串中删除 `-ro` 来转换以前的读取器终端节点。例如，如果以前的读取器终端节点为 `global-16rr-test-cluster-1.cluster-ro-12345678901.us-west-2.docdb.amazonaws.com`，则新提升的写入器终端节点为 `global-16rr-test-cluster-1.cluster-cps2igpwyrrwa.us-west-2.rds.amazonaws.com`。

Amazon DocumentDB 全局集群发生

如果整个集群合而为一 Amazon Web Services 区域变得不可用，则可以提升全局集群中的另一个集群，使之具有读/写功能。

如果集群位于另一个中，则可以手动启用故障转移机制 Amazon Web Services 区域是作为主集群的更好选择。例如，您可以提高其中一个辅助集群的容量，然后将其提升成为主集群。或者是活动之间的平衡 Amazon Web Services 区域可能会发生变化，因此将主集群切换到其他群集 Amazon Web Services 区域可能会降低写入操作的延迟。

以下过程概述了如何提升 DocumentDB 全局集群中的一个辅助集群。

提升辅助集群：

1. 停止向中的主集群发出 DML 语句和其他写入操作 Amazon Web Services 区域停机。
2. 从辅助群集标识群集 Amazon Web Services 区域用作新的主群集。如果你有两个（或更多）中学 Amazon Web Services 区域在您的全局集群中，选择滞后时间最短的辅助集群。
3. 从全局集群分离您所选的辅助集群。

从全局集群删除辅助集群会立即停止从主集群到该辅助集群的复制过程，并会将其提升为拥有完全读/写功能的独立预置的集群。与该停机区域中的主集群关联的任何其他辅助集群仍然可用，并且可以接受应用程序的调用。它们还会消耗资源。由于您要重新创建全局集群，为避免分裂大脑和其他问题，请先删除其他辅助集群，再在后续步骤中创建新的全局集群。

有关分离的详细步骤，请参阅 [从 Amazon DocumentDB 全局集群删除集群 \(p. 236\)](#)。

4. 重新配置应用程序，使用新的终端节点将所有写入操作发送到现在的独立集群。如果您在创建全局集群时接受了提供的名称，则可以在应用程序中从集群的终端节点字符串中删除-ro。

例如，辅助集群的终端节点 `my-global.cluster-ro-aaaaaabbabbbb.us-west-1.docdb.amazonaws.com` 变成 `my-global.cluster-aaaaaabbabbbb.us-west-1.docdb.amazonaws.com` 当该集群从全局集群分离时。

在下一步中，当您开始向新的全局集群添加区域时，该集群将成为新的全局集群的主集群。

5. 添加 Amazon Web Services 区域到群集。执行此操作后，从主数据库集群到辅助数据库集群的复制过程将会开始。
6. 添加更多 Amazon Web Services 区域根据需要重新创建为了支持应用程序所需的拓扑。确保在做出这些更改之前、更改期间和更改之后，将应用程序写入内容发送到正确的集群，以避免全局集群中集群之间的数据不一致（大脑分裂问题）。
7. 当停机问题已解决且您已准备好分配原件时 Amazon Web Services 区域再次作为主集群，请按相反顺序执行相同步骤。
8. 从全局集群移除其中一个辅助集群。这将使其能够为读/写流量提供服务。
9. 将所有写入流量重定向到原始集群中的主集群 Amazon Web Services 区域。
10. 添加 Amazon Web Services 区域可以在同一个区域中设置一个或多个辅助集群 Amazon Web Services 区域和以前一样。

Amazon DocumentDB 全局集群可以通过以下方式进行管理 Amazon SDK，使您能够为灾难恢复和业务连续性规划使用情形创建自动化全局群集故障转移流程的解决方案。其中一个解决方案是在 Apache 2.0 许可下为我们的客户提供的，并且可以从我们的工具存储库中访问 [这里](#)。该解决方案利用 Amazon Route53 进行终端节点管理，并提供 Amazon 可以根据相应事件触发的 Lambda 函数。

管理Amazon DocumentDB

要管理 Amazon DocumentDB 集群，您必须具有相应的 Amazon DocumentDB 控制平面权限的 IAM 策略。这些权限使您能够创建、修改和删除集群和实例。这些区域有：AmazonDocDBFullAccess策略提供了管理 Amazon DocumentDB 集群所需的所有权限。

以下主题介绍在使用 Amazon DocumentDB 集群时如何执行各种任务，包括创建、删除、修改、连接和查看集群。

主题

- [了解集群 \(p. 243\)](#)
- [Amazon DocumentDB 集 \(p. 244\)](#)
- [确定集群的状态 \(p. 245\)](#)
- [Amazon DocumentDB 集 \(p. 246\)](#)
- [扩展Amazon DocumentDB \(p. 271\)](#)
- [克隆 Amazon DocumentDB 集群卷 \(p. 272\)](#)
- [了解Amazon DocumentDB B 集群容错能力 \(p. 282\)](#)

了解集群

Amazon DocumentDB 将计算和存储分开，并将数据复制和备份转移到集群卷。集群卷提供持久、可靠且高度可用的存储层，可跨三个可用区以六种方式复制数据。副本可实现更高的数据可用性和读取扩展。每个集群最多可以扩展到 15 个副本。

名词	描述	API 操作 (动词)
Cluster	包含一个或多个实例和一个管理这些实例的数据的集群存储卷。	<code>create-db-cluster</code> <code>delete-db-cluster</code> <code>describe-db-clusters</code> <code>modify-db-cluster</code>
实例	读取集群存储卷以及向其中写入数据将通过实例进行。在给定的集群中，有两种类型的实例：主实例和副本实例。一个集群始终有一个主实例，可以有 0—15 个副本。	<code>create-db-instance</code> <code>delete-db-instance</code> <code>describe-db-instances</code> <code>modify-db-instance</code> <code>describe-orderable-db-instance-options</code> <code>reboot-db-instance</code>
集群卷	跨三个可用区的虚拟数据库存储卷，每个可用区均有两个集群数据副本。	不适用
主实例	支持读取和写入操作，并执行针对集群卷的所有数据修改。每个集群均有一个主实例。	不适用

名词	描述	API 操作 (动词)
副本实例	仅支持读取操作。除主实例之外，每个 Amazon DocumentDB 集群最多可拥有 15 个副本实例。多个副本共同分担读取工作负载。通过将副本置于单独的可用区中，您还可以提高数据库可用性。	不适用
集群终端节点	Amazon DocumentDB 集群的终端节点，它连接到该集群的当前主实例。每个 Amazon DocumentDB 集群均有一个集群终端节点和一个主实例。	不适用
读取器终端节点	Amazon DocumentDB 集群的终端节点，它连接到该集群的可用副本之一。每个 Amazon DocumentDB B 集群都有一个读取器终端节点。如果有多个副本，则读取器终端节点将每个连接请求定向到其中一个 Amazon DocumentDB 副本。	不适用
实例终端节点	Amazon DocumentDB 集群中连接到特定实例的实例的终端节点。集群中的每个实例，不论其实例类型，都有各自唯一的实例终端节点。	不适用

Amazon DocumentDB 集

创建或修改集群时，了解哪些参数不可改变，哪些参数在创建集群后可修改，这一点很重要。下表列出了特定于集群的所有设置或参数。如表中所示，有些是可修改的，有些不可修改。

Note

不应将这些设置与 Amazon DocumentDB 集群参数组及其参数混淆。有关集群参数组的更多信息，请参阅 [管理 Amazon DocumentDB 集群参数组 \(p. 322\)](#)。

参数	可修改	备注
DBClusterIdentifier	是	命名约束： - 长度为 [1-63] 个字母、数字或连字符。 - 第一个字符必须是字母。 - 不能以连字符结束或包含两个连续连字符。 - 亚马逊 RDS、Amazon Neptune 和 Amazon DocumentDB 中的所有集群必须是唯一的 Amazon Web Services 账户，每个区域。
Engine	否	必须是 docdb。
BackupRetentionPeriod	是	必须介于 [1-35] 天之间。

参数	可修改	备注
DBClusterParameterGroupName	是	命名约束： - 长度为 [1—255] 个字母数字字符。 - 第一个字符必须是字母。 - 不能以连字符结束或包含两个连续连字符。
DBSubnetGroupName	否	创建集群后，无法修改集群的子网。
EngineVersion	否	必须是 4.0.0。
KmsKeyId	否	如果选择加密您的集群，则不能更改用于加密集群的 Amazon KMS 密钥。
MasterUsername	否	创建集群后，无法修改 MasterUsername。 命名约束： - 长度为 [1—63] 个字母数字字符。 - 第一个字符必须是字母。 - 不能是数据库引擎的保留字。
MasterUserPassword	是	约束： - 长度为 [8—100] 个可打印的 ASCII 字符。 - 可以使用任何可打印 ASCII 字符，以下字符除外： / (正斜杠) " (双引号) @ (@ 符号)
Port	是	端口号适用于集群中的所有实例。
PreferredBackupWindow	是	
PreferredMaintenanceWindow	是	
StorageEncrypted	否	如果选择加密集群，则不能对其进行加密。
Tags	是	
VpcSecurityGroupIds	否	创建集群后，您不能修改集群所在的 VPC。

确定集群的状态

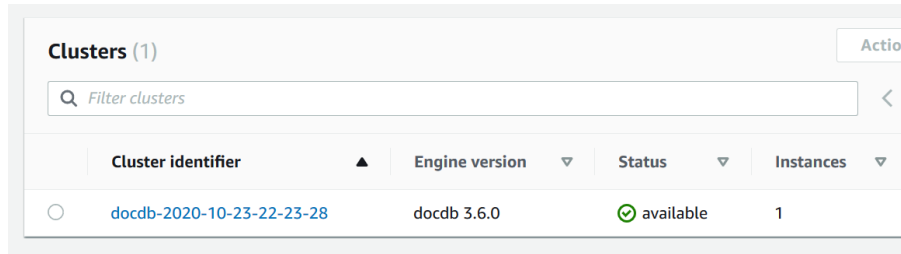
您可以使用 Amazon Web Services Management Console 或 Amazon CLI 确定集群的状态。

使用 Amazon Web Services Management Console

使用以下过程查看您的 Amazon DocumentDB 集群的状态 Amazon Web Services Management Console

1. 登录 Amazon Web Services Management Console，然后打开 Amazon DocumentDB B 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

3. 在 Cluster identifier (集群标识符) 列中，找到您感兴趣的集群名称。然后，要查找集群的状态，请跨该行阅读至 Status (状态) 列，如下所示。



	Cluster identifier ▲	Engine version ▼	Status ▼	Instances ▼
☐	docdb-2020-10-23-22-23-28	docdb 3.6.0	✔ available	1

使用 Amazon CLI

使用describe-db-clusters使用操作查看您的 Amazon DocumentDB 集群的状态Amazon CLI.

以下代码可查找集群 sample-cluster 的状态。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

对于 Windows：

```
aws docdb describe-db-clusters ^
  --db-cluster-identifier sample-cluster ^
  --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

此操作的输出将类似于下文 (JSON 格式)。

```
[
  [
    "sample-cluster",
    "available"
  ]
]
```

Amazon DocumentDB 集

Amazon DocumentDB 集群的生命周期包括创建、描述、修改和删除集群。本节提供有关如何完成这些过程的信息。

主题

- [创建Amazon DocumentDB 集群 \(p. 247\)](#)
- [Amazon DocumentDB 集群描述 \(p. 255\)](#)
- [修改Amazon DocumentDB 集群 \(p. 259\)](#)
- [确定待处理维护 \(p. 261\)](#)
- [更新集群的引擎版本 \(p. 263\)](#)
- [停止和启动Amazon DocumentDB B 集群 \(p. 265\)](#)
- [删除 Amazon DocumentDB 集群 \(p. 267\)](#)

创建 Amazon DocumentDB 集群

Amazon DocumentDB 集群包含实例和一个代表集群的数据的集群卷。集群卷作为单个虚拟卷在三个可用区之间的 6 个方向进行复制。集群包含一个主实例，以及可选的最多 15 个副本实例。

以下部分显示如何使用 Amazon DocumentDB 集群 Amazon Web Services Management Console 或者 Amazon CLI。然后，您可为该集群添加更多副本实例。当您使用控制台创建 Amazon DocumentDB 集群时，系统会同时自动为您创建一个主实例。如果您将 Amazon CLI 在集群的状态为之后创建您的 Amazon DocumentDB 集群可用，然后必须为该集群创建主实例。

先决条件

以下是创建 Amazon DocumentDB 集群的先决条件。

如果您还没有 Amazon Web Services 账户，请完成以下步骤来创建一个。

注册 Amazon Web Services 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

VPC 先决条件

您只能在 Amazon Virtual Private Cloud (Amazon VPC) 中创建 Amazon DocumentDB B 集群。要在 Amazon DocumentDB 集群中使用您的 Amazon VPC，该 VPC 必须最少在两个可用区中均拥有一个子网。通过在可用区之间分配集群实例，您可以确保实例在可能出现可用区故障的情况下在集群中可用。

子网先决条件

创建 Amazon DocumentDB 集群时，您必须在该 VPC 中选择一个 VPC 和相应的子网组来启动您的集群。子网确定可用区以及该可用区内要用于启动实例的 IP 范围。为进行此讨论，我们将互换使用术语子网 和 可用区。子网组是一组指定的子网（或可用区）。子网组允许您做的是指定要用于启动 Amazon DocumentDB 实例的可用区域。例如，在包含三个实例的集群中，为了实现高可用性，建议在单独的可用区中配置各个实例。因此，如果单个可用区出现故障，它只会影响单个实例。

目前，Amazon DocumentDB 实例最多可以在三个可用区中进行配置。即使一个子网组有三个以上的子网，您也只能使用其中的三个子网来创建 Amazon DocumentDB 集群。因此，在创建子网组时，建议仅选择要将实例部署到的三个子网。在美国东部（弗吉尼亚北部），您的子网组可以有六个子网（或可用区）。但是，在配置 Amazon DocumentDB 集群时，Amazon DocumentDB 会选择其用来预置实例的三个可用区。

例如，假设您在创建集群时，Amazon DocumentDB 选择了可用区 {1A、1B 和 1C}。如果您尝试在可用区 {1D} 中创建实例，API 调用将失败。但是，如果您选择不指定特定可用区的情况下创建实例，则 Amazon DocumentDB 会代表您选择一个可用区。Amazon DocumentDB 使用算法在可用区间对实例进行负载均衡，以帮助实现高可用性。例如，如果配置了三个实例，则在默认情况下，将在三个可用区中配置它们，而不会在单个可用区中配置。

建议：

- 除非您有特殊原因，否则请始终创建包含三个子网的子网组。这样做可帮助确保包含三个或更多实例的集群能够实现更高的可用性，因为将在三个可用区中预配置实例。
- 始终将实例分散在多个可用区中以实现高可用性。切勿将集群的所有实例放在单个可用区中。
- 由于故障转移事件随时可能发生，您不应假定主实例或副本实例始终位于特定可用区中。

其他先决条件

以下是创建 Amazon DocumentDB 集群的一些其他先决条件：

- 如果你正在连接到Amazon使用Amazon Identity and Access Management(IAM) 证书，您的 IAM 账户必须拥有 IAM 政策来授予执行 Amazon DocumentDB 操作所需的权限。

如果要使用 IAM 账户访问 Amazon DocumentDB 控制台，必须先登录Amazon Web Services Management Console使用您的 IAM 账户。然后转至上的Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。

- 如果要定制您的集群的配置参数，您必须指定集群参数组和具有必需参数设置的参数组。有关创建或修改集群参数组的信息，请参阅[管理 Amazon DocumentDB 集群参数组 \(p. 322\)](#)。
- 您必须确定要为集群指定的 TCP/IP 端口号。有些公司的防火墙不允许连接到Amazon DocumentDB B 的默认端口。如果您公司的防火墙阻止该默认端口，请为您的集群选择其他端口。集群中的所有实例都使用同一端口。

使用 Amazon Web Services Management Console创建集群和主实例

以下过程介绍如何使用控制台启动具有一个或多个实例的 Amazon DocumentDB 集群。

创建集群：使用默认设置

使用 Amazon Web Services Management Console创建包含使用默认设置的实例的集群

1. 登录Amazon Web Services Management Console，然后打开Amazon DocumentDB B 控制台<https://console.aws.amazon.com/docdb>。
2. 如果你想在创建集群Amazon Web Services 区域除美国东部（弗吉尼亚北部）区域之外，请从控制台右上角的列表中选择该区域。
3. 在导航窗格中，选择 Clusters (集群)，然后选择 Create (创建)。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

4. 在创建Amazon DocumentDB页面，完成配置pane。
 - a. 集群标识符— 接受 Amazon DocumentDB 提供的名称，或输入您集群的名称；例如**sample-cluster**。

集群命名约束：
 - 长度为 [1-63] 个字母、数字或连字符。
 - 第一个字符必须是字母。
 - 不能以连字符结束或包含两个连续连字符。
 - Amazon RDS、Neptune 和 Amazon DocumentDB 中的所有集群必须是唯一的Amazon Web Services 账户，每个区域。
 - b. 引擎版本—接受 4.0.0 的默认引擎版本，或者选择选择 3.6.0。
 - c. 实例类— 接受默认值db.r5.large，或者从列表中选择所需的实例类。
 - d. 实例实例实例实例实例实例实例— 在列表中，选择要使用此集群创建的实例数。第一个实例为主实例，所有其他实例为只读副本实例。您可以在稍后添加和删除实例（如果需要）。默认情况下，Amazon DocumentDB 集群使用三个实例（一个主实例和两个副本）启动。
5. 完成 Authentication (身份验证) 窗格。
 - a. 主用户名— 输入主用户的名称。要登录您的集群，您必须使用主用户名。

主用户名命名约束：

- 长度为 [1—63] 个字母数字字符。
- 第一个字符必须是字母。

- 不能是数据库引擎的保留字。
- b. 主密码— 输入主用户的密码，然后确认。要登录您的集群，您必须使用主密码。

主密码约束：

- 长度为 [8-100] 个可打印 ASCII 字符。
- 可以使用任何可打印 ASCII 字符，以下字符除外：
 - / (正斜杠)
 - " (双引号)
 - @ (@ 符号)

6. 在屏幕的底部，选择以下选项之一：

- 要立即创建集群，请选择 Create cluster (创建集群)。
- 要不创建集群，请选择 Cancel (取消)。
- 要在创建集群前进一步配置集群，请选择 Show additional configurations (显示其他配置)，然后在 [创建集群：其他配置 \(p. 250\)](#) 上继续。

Additional Configurations (其他配置) 部分中包含的配置如下：

- 网络设置— 默认情况下，使用 default VPC security group (VPC 安全组)。
- 集群选项— 默认使用端口为 27017，默认参数组。
- 加密— 默认为使用启用加密 (default) aws/rds 钥匙。

Important

集群加密后，就不能取消加密。

- 备份— 默认是将备份保留 1 天，然后让 Amazon DocumentDB 选择备份窗口。
- 日志导出— 默认为不将审核日志导出到 CloudWatch 日志。
- 维护— 默认为由 Amazon DocumentDB 选择维护时段。
- 删除保护— 保护您的集群免遭意外删除。使用控制台创建的集群的默认设置为已启用。

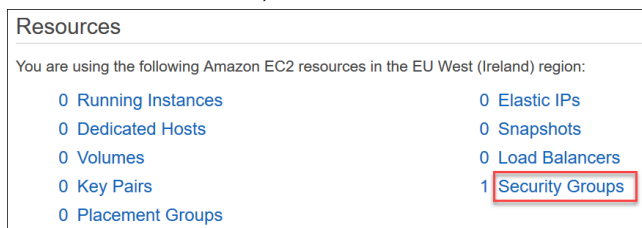
如果您现在接受默认设置，则可以稍后通过修改集群来更改其中的大部分。

7. 为集群的安全组启用入站连接。

如果未更改集群的默认设置，则会使用给定区域中的默认 VPC 的默认安全组创建一个集群。要连接到 Amazon DocumentDB，您必须在端口 27017 (或您选择的端口) 上为集群的安全组启用入站连接。

将入站连接添加到集群的安全组

- a. 登录到 Amazon Web Services Management Console 并打开 Amazon EC2 控制台 (<https://console.aws.amazon.com/ec2/>) 。
- b. 在主窗口的资源部分中，选择安全组。



- c. 从安全组列表中，找到您在创建集群时使用的安全组 (很可能是 default 安全组)，然后选中安全组名称左侧的框。

Name	Group ID	Group Name	VPC ID
☒	sg-06b2ad61	default	vpc-d833a4bc
☐	sg-07443a112c70a5282	test-sg	vpc-d833a4bc

- d. 从操作菜单中，选择编辑入站规则，然后选择或输入规则限制。
 - i. 类型—从列表中，选择要向网络流量开放的协议。
 - ii. 协议—从列表中选择协议类型。
 - iii. 端口范围—对于自定义规则，输入端口号或端口范围。确保端口号或范围包括您在创建集群时指定的端口（默认值：27017）。
 - iv. 源—指定可以到达您的实例的流量。从列表中，选择流量源。如果选择自定义，请指定单个 IP 地址或以 CIDR 格式表示的 IP 地址范围（例如，203.0.113.5/32）。
 - v. 说明—输入对此规则的描述。
 - vi. 在创建完规则时，选择保存。

创建集群：其他配置

如果要接受集群的默认设置，您可以跳过以下步骤并选择 Create cluster (创建集群)。

1. 完成 Network settings (网络设置) 窗格。

Network settings

a

Virtual Private Cloud (VPC) [Info](#)

VPC defines the virtual networking environment for this cluster.

vpc-91280df6

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

b

Subnet group [Info](#)

A subnet group is a collection of subnets that are within a VPC.

default

c

VPC security groups

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups

default (VPC) X

- a. Virtual Private Cloud (VPC)—在列表中，选择要在其中启动此集群的 Amazon VPC。
 - b. 子网组—在列表中，选择要用于此集群的子网组。
 - c. VPC 安全组—在列表中，选择此集群的 VPC 安全组。
2. 完成 Cluster options (集群选项) 窗格。

Cluster options

Port

TCP/IP port that is used to connect to the cluster.

27017

Cluster parameter group [Info](#)

default.docdb4.0

- a. 数据库端口—使用向上和向下箭头设置应用程序用于连接到您的实例的 TCP/IP 端口。
 - b. 集群参数组—在参数组列表中，选择此集群的集群参数组。
3. 完成 Encryption (加密) 窗格。

- a. 启用加密—选择以下选项之一：
 - 启用加密—Default 所有静态数据都会加密。如果您选择加密您的数据，则无法撤消此操作。
 - 禁用加密—您的数据未加密。
- b. 主密钥—这仅在您加密数据时可用。在列表中，选择要用于加密此集群中的数据的密钥。默认为 (default) aws/rds。

如果您选择 Enter a key ARN (输入一个密钥 ARN)，则必须为密钥输入一个 Amazon 资源名称 (ARN)。

备份保留期—在列表中，选择您希望自动备份的保留天数。

4. 完成 Backup (备份) 窗格。

- a. 备份保留期—在列表中，选择在删除集群前保留自动备份的天数。
 - b. 备份时段—设置 Amazon DocumentDB 每天备份此集群的时间和持续时间。
 - i. 开始时间-在第一个列表中，选择开始自动备份的开始时间 (UTC)。在第二个列表中，选择您希望自动备份开始的时间 (分钟)。
 - ii. Duration-在列表中，选择分配给创建自动备份的小时数。
5. 完成日志导出窗格，选择要导出到的日志类型 CloudWatch 日志。

- 审核日志—选择此选项可启用将审计日志导出到亚马逊 CloudWatch 日志。如果您选择 Audit logs (审计日志)，则必须在集群的自定义参数组中启用 audit_logs。有关更多信息，请参阅 [审计 Amazon DocumentDB \(p. 185\)](#)。

- 分析器日志—选择此选项可启用将操作分析器日志导出到亚马逊 CloudWatch 日志。如果您选择 Profiler logs (分析器日志)，则还必须在集群的自定义参数组中修改以下参数：
 - profiler—将设置为 enabled。
 - profiler_threshold_ms—设置为一个值 [0-INT_MAX] 设置分析操作的阈值。
 - profiler_sampling_rate—设置为一个值 [0.0-1.0] 将慢速操作的百分比设置为 profile。

有关更多信息，请参阅 [分析 Amazon DocumentDB 操作 \(p. 388\)](#)。

6. 完成 Maintenance (维护) 窗格。

- 选择以下选项之一
 - 选择窗口—您可以指定 Amazon DocumentDB 在集群上执行维护的某一周中的一天、UTC 开始时间和持续时间。
 - 开始日— 在列表中，选择一周中的哪一天开始群集维护。
 - 开始时间— 在列表中，选择开始维护的小时和分钟 (UTC)。
 - Duration— 在列表中，选择分配多少时间用于群集维护。如果无法在指定时间内完成维护，维护过程将在指定时间过后继续进行，直到完成。
 - 没有首选项— Amazon DocumentDB 可选择一周中的一天、开始时间和执行维护的持续时间。

7. 如果您要将一个或多个标签添加到此集群，请填写标签窗格。

对于每个要添加到集群的标签，请重复以下步骤。您最多可以在一个集群上具有 10 个集群。

- 选择添加标签。
- 键入标签的键。
- (可选) 键入标签的值。

要删除标签，请选择删除标签。

- 在使用控制台创建集群时，将默认启用删除保护。要禁用删除保护，请清除启用删除保护。启用后，删除保护可以防止删除集群。要删除某个受到删除保护的集群，您必须先修改集群以禁用删除保护。

有关删除保护的更多信息，请参阅 [删除 Amazon DocumentDB 集群 \(p. 267\)](#)。

9. 要创建集群，请选择创建集群。否则，选择取消。

使用 Amazon CLI 创建集群

以下过程介绍如何使用 Amazon CLI 启动 Amazon DocumentDB B 集群并创建 Amazon DocumentDB B 副本。

参数

- **--db-cluster-identifier** – 必填项。标识此集群的一个小写字母字符串。

集群命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 所有集群（跨亚马逊 RDS、Amazon Neptune 和 Amazon DocumentDB）必须是唯一的 Amazon 账户，每个区域。

- **--engine** – 必填项。必须是 **docdb**。
- **--deletion-protection** | **--no-deletion-protection** – 可选。启用删除保护后，可以防止删除集群。当您使用 Amazon CLI 时，默认设置为已禁用删除保护。

有关删除保护的更多信息，请参阅 [删除 Amazon DocumentDB 集群 \(p. 267\)](#)。

- **--master-username** – 必填项。用于对用户进行身份验证的用户名。

主用户命名约束：

- 长度为 [1-63] 个字母数字字符。
- 第一个字符必须是字母。
- 不能是数据库引擎的保留字。

- **--master-user-password** – 必填项。用于对用户进行身份验证的用户密码。

主密码约束：

- 长度为 [8-100] 个可打印 ASCII 字符。
- 可以使用任何可打印 ASCII 字符，以下字符除外：
 - / (正斜杠)
 - " (双引号)
 - @ (@ 符号)

有关更多参数，请参阅 [CreateDBCluster \(p. 542\)](#)。

要启动 Amazon DocumentDB 集群，请使用 Amazon CLI

要创建 Amazon DocumentDB 集群，请调用 `create-db-cluster` Amazon CLI。以下 Amazon CLI 命令创建一个名为 Amazon DocumentDB 集群 `sample-cluster` 启用了删除保护。有关删除保护的更多信息，请参见 [删除 Amazon DocumentDB 集群 \(p. 267\)](#)。

以及`--engine-version`是一个可选参数，默认为最新的主引擎版本。当前的主要引擎版本为 4.0.0。发布新的主要引擎版本时，的默认引擎版本`--engine-version`将进行更新以反映最新的主要引擎版本。因此，对于生产工作负载，尤其是那些依赖脚本编写、自动化或Amazon CloudFormation模板，我们建议您明确指定`--engine-version`改为预期的主要版本。

Note

如果 `adb-subnet-group-name` 要么 `vpc-security-group-id` 未指定，Amazon DocumentDB 将使用给定区域的默认子网组和 Amazon VPC 安全组。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "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-KQSGI4MHU4NTDDRVLNTU7XVAY",  
    "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",  
    "Port": 27017,  
    "Status": "creating",  
    "ReaderEndpoint": "sample-cluster.cluster-ro-sfcr1cjcoroz.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"
  }
}
```

创建集群需要几分钟时间。您可以使用 Amazon Web Services Management Console 或 Amazon CLI 监控集群的状态。有关更多信息，请参阅 [监控 Amazon DocumentDB 集群的状态 \(p. 366\)](#)。

Important

当您使用 Amazon CLI 要创建 Amazon DocumentDB 集群，不创建任何实例。因此，您必须显式创建主实例和所需的任何副本实例。您可以使用控制台或 Amazon CLI 来创建实例。有关更多信息，请参阅 [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)。

有关更多信息，请参阅 [CreateDBCluster](#) 在 Amazon DocumentDB。

Amazon DocumentDB 集群描述

您可以使用 Amazon DocumentDB 管理控制台或 Amazon CLI 查看与您的 Amazon DocumentDB 集群相关的连接终端节点、安全组、VPC 和参数组等详细信息。

有关更多信息，请参阅下列内容：

- [监控 Amazon DocumentDB 集群的状态 \(p. 366\)](#)
- [查找集群的终端节点 \(p. 344\)](#)

使用 Amazon Web Services Management Console

执行以下程序通过控制台查看指定 Amazon DocumentDB 集群的详细信息。

1. 登录 Amazon Web Services Management Console，然后打开 Amazon DocumentDB B 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在集群列表中，选择要查看其详细信息的集群的名称。有关集群的信息分为以下几组：
 - 摘要— 有关集群的一般信息，包括引擎版本、集群状态、待维护及其参数组的状态。
 - 连接与安全— 连接部分列出了使用 mongo shell 或应用程序连接到此集群的连接端点。Security Groups (安全组) 部分列出与此集群关联的安全组及其 VPC ID 和描述。
 - 配置— 这个集群详细信息部分列出了集群的详细信息，包括集群的 Amazon Resource Name (ARN)、终端节点和参数组。其中还列出集群的备份信息、维护详细信息以及安全和网络设置。Cluster instances (集群实例) 部分列出属于该集群的实例，其中包括每个实例的角色和集群参数组状态。
 - 监控— 亚马逊 CloudWatch 记录此集群的指标。有关更多信息，请参阅 [使用 CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)。
 - 活动和标签— 这个近期事件部分列出了该集群的最新事件。Amazon DocumentDB 将记录与集群、实例、快照、安全组和集群参数组相关的事件。此信息包括与每个事件关联的日期、时间和消息。Tags (标签) 部分列出附加该集群的标签。

使用 Amazon CLI

要查看您的 Amazon DocumentDB B 集群的详细信息，请使用 Amazon CLI，`describe-db-clusters` 命令如下示例所示。有关更多信息，请参阅 [DescribeDBClusters](#) 在 Amazon DocumentDB B 资源管理 API 参考。

Note

对于某些管理功能，例如集群和实例生命周期管理，Amazon DocumentDB 利用了与 Amazon RDS 共享的操作技术。这些区域有：`filterName=engine,Values=docdb` 筛选参数仅返回 Amazon DocumentDB B 集群。

Example

示例 1：列出所有 Amazon DocumentDB B 集群

以下 Amazon CLI 代码列出了某个区域内所有 Amazon DocumentDB 集群的详细信息。

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

此操作的输出将类似于下文。

```
{
  "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

示例 2：列出指定 Amazon DocumentDB 集群的所有详细信息

以下 Amazon CLI 代码列出集群 `sample-cluster` 的详细信息。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \
  --filter Name=engine,Values=docdb \
  --db-cluster-identifier sample-cluster
```

对于 Windows：

```
aws docdb describe-db-clusters ^
  --filter Name=engine,Values=docdb ^
  --db-cluster-identifier sample-cluster
```

此操作的输出将类似于下文。

```
{
  "DBClusters": [
    {
      "AvailabilityZones": [
        "us-east-1c",
        "us-east-1b",
        "us-east-1a"
      ],
      "BackupRetentionPeriod": 1,
      "DBClusterIdentifier": "sample-cluster",
      "DBClusterParameterGroup": "sample-parameter-group",
      "DBSubnetGroup": "default",
      "Status": "available",
      "EarliestRestorableTime": "2020-03-09T00:04:26.844Z",
      "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": "4.0.0",
      "LatestRestorableTime": "2020-03-10T20:38:17.456Z",
      "Port": 27017,
      "MasterUsername": "<user-name>",
      "PreferredBackupWindow": "00:00-00:30",
      "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
      "DBClusterMembers": [
        {
          "DBInstanceIdentifier": "sample-instance-1",
          "IsClusterWriter": true,
          "DBClusterParameterGroupStatus": "in-sync",
          "PromotionTier": 1
        },
        {
          "DBInstanceIdentifier": "sample-instance-2",
          "IsClusterWriter": false,
          "DBClusterParameterGroupStatus": "in-sync",
          "PromotionTier": 1
        }
      ],
      "VpcSecurityGroups": [
```

```

        {
          "VpcSecurityGroupId": "sg-abcd0123",
          "Status": "active"
        }
      ],
      "HostedZoneId": "ABCDEFGHIJKLMN",
      "StorageEncrypted": true,
      "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
      "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
      "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
      "AssociatedRoles": [],
      "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
      "EnabledCloudwatchLogsExports": [
        "profiler"
      ],
      "DeletionProtection": true
    }
  ]
}

```

Example

示例 3：列出 Amazon DocumentDB B 集群的具体详细信息

要使用 Amazon CLI 列出集群的部分详细信息，请添加 `--query`，以指定 `describe-db-clusters` 操作要列出哪个集群成员。`--db-cluster-identifier` 参数是要显示其详细信息的特定集群的标识符。有关查询的更多信息，请参阅[如何使用过滤输出--query选项](#)在 Amazon Command Line Interface 用户指南。

以下示例列 Amazon DocumentDB 集群中的实例。

对于 Linux、macOS 或 Unix：

```

aws docdb describe-db-clusters \
  --filter Name=engine,Values=docdb \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[DBClusterMembers]'

```

对于 Windows：

```

aws docdb describe-db-clusters ^
  --filter Name=engine,Values=docdb ^
  --db-cluster-identifier sample-cluster ^
  --query 'DBClusters[*].[DBClusterMembers]'

```

此操作的输出将类似于下文。

```

[
  [
    [
      {
        "DBInstanceIdentifier": "sample-instance-1",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "sample-instance-2",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      }
    ]
  ]
]

```

```
}  
]  
]  
]
```

修改 Amazon DocumentDB 集群

要修改集群，该集群必须处于 可用 状态。您无法修改已停止的集群。如果集群已停止，则首先启动集群，等待集群进入可用状态，然后进行所需修改。有关更多信息，请参阅 [停止和启动 Amazon DocumentDB B 集群](#) (p. 265)。

使用 Amazon Web Services Management Console

使用控制台，通过以下过程修改特定的 Amazon DocumentDB 集群。

修改 Amazon DocumentDB 集群

1. 登录 Amazon Web Services Management Console，然后打开 Amazon DocumentDB B 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 通过选择集群名称左侧的按钮，指定要修改的集群。
4. 选择 Actions (操作)，然后选择 Modify (修改)。
5. 在 Modify Cluster: cluster-name (修改集群: cluster-name) 窗格中，进行所需更改。您可以更改以下方面：
 - 集群规格—集群的名称、安全组和密码。
 - 集群选项—集群的端口和参数组。
 - 备份— 集群的备份保留期和备份时段。
 - 日志导出—启用或禁用导出审计或分析器日志。
 - 维护— 设置集群的维护时段。
 - 删除保护—在群集上启用或禁用删除保护。默认情况下，将启用删除保护。
6. 完成后，选择 Continue (继续) 以查看更改摘要。
7. 如果您对所做更改满意，可以选择 Modify cluster (修改集群) 以修改集群。或者，您可以选择 Back (返回) 或 Cancel (取消) 来编辑或取消您的更改。

需要几分钟时间才能应用更改。只有在实例状态为 available (可用) 时，才能使用集群。您可以使用控制台或 Amazon CLI 监控集群状态。有关更多信息，请参阅 [监控 Amazon DocumentDB 集群的状态](#) (p. 366)。

使用 Amazon CLI

使用 `modify-db-cluster` 操作以使用 Amazon CLI 修改指定的集群。有关更多信息，请参阅 [ModifyDBCluster](#) 在 Amazon DocumentDB。

参数

- `--db-cluster-identifier`—必填项。您要修改的 Amazon DocumentDB 集群的标识符。
- `--backup-retention-period`—可选。自动备份的保留天数。有效值为 1-35。
- `--db-cluster-parameter-group-name`—可选。用于集群的集群参数组的名称。
- `--master-user-password`—可选。主数据库用户的新密码。

密码约束：

- 长度为 [8—100] 个可打印的 ASCII 字符。
- 可以使用任何可打印 ASCII 字符，以下字符除外：
 - / (正斜杠)
 - " (双引号)
 - @ (@ 符号)
- **--new-db-cluster-identifier**—可选。重命名集群时集群的新集群标识符。此值以一个小写字母存储。

命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 亚马逊 RDS、Amazon Neptune 和 Amazon DocumentDB 中的所有集群必须是唯一的 Amazon Web Services 账户，每个区域。
- **--preferred-backup-window**—可选。创建自动备份的每日时间范围，以通用协调时间 (UTC) 表示。
 - 格式：hh24:mm-hh24:mm
- **--preferred-maintenance-window**—可选。可能进行系统维护的每周时间范围，以 UTC 表示。
 - 格式：ddd:hh24:mm-ddd:hh24:mm
 - 有效值：Sun、Mon、Tue、Wed、Thu、Fri 和 Sat。
- **--deletion-protection**要么**--no-deletion-protection**—可选。是否应在此集群上启用删除保护。删除保护可以防止集群被意外删除，直到修改集群以禁用删除保护。有关更多信息，请参阅 [删除 Amazon DocumentDB 集群 \(p. 267\)](#)。
- **--apply-immediately**要么**--no-apply-immediately**—使用**--apply-immediately**立即进行更改。使用 **--no-apply-immediately** 在集群的下一个维护时段期间进行更改。

Example

以下代码将更改集群 sample-cluster 的备份保留期。

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --apply-immediately \
    --backup-retention-period 7
```

对于 Windows :

```
aws docdb modify-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --apply-immediately ^
    --backup-retention-period 7
```

此操作的输出将类似于下文。

```
{
  "DBCluster": {
    "BackupRetentionPeriod": 7,
    "DbClusterResourceId": "cluster-VDP53QEWS7YHM36TTXOPJT5YE",
    "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
}
```

需要几分钟时间才能应用更改。只有在实例状态为 `available` (可用) 时, 才能使用集群。您可以使用控制台或 Amazon CLI 监控集群状态。有关更多信息, 请参阅 [监控 Amazon DocumentDB 集群的状态 \(p. 366\)](#)。

确定待处理维护

您可以通过确定是否有待处理的集群维护来确定您是否拥有最新的 Amazon DocumentDB 引擎版本。

使用 Amazon Web Services Management Console

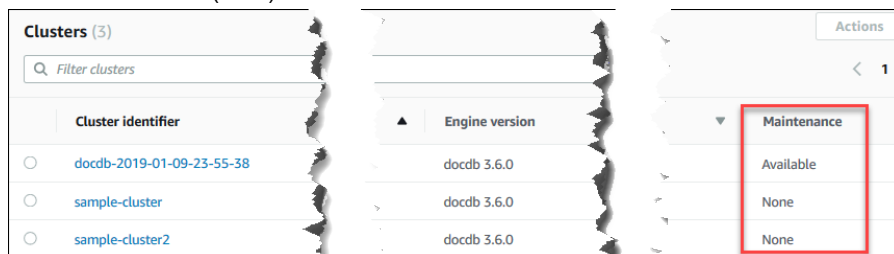
您可以使用 Amazon Web Services Management Console 确定集群是否有待处理维护。

1. 登录 Amazon Web Services Management Console, 然后打开 Amazon DocumentDB B 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中, 选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格, 请在页面左上角选择菜单图标 (≡)。

3. 找到 Maintenance (维护) 列以确定集群是否有待处理维护。



Cluster identifier	Engine version	Maintenance
docdb-2019-01-09-23-55-38	docdb 3.6.0	None
sample-cluster	docdb 3.6.0	Available
sample-cluster2	docdb 3.6.0	None

None (无) 表示集群运行的是最新的引擎版本。Available (可用) 表示集群有待处理维护，这可能意味着需要升级引擎。

4. 如果您的集群有待处理维护，请继续执行[更新集群的引擎版本 \(p. 263\)](#)中的步骤。

使用 Amazon CLI

您可以在 Amazon CLI 中使用具有以下参数的 `describe-pending-maintenance-actions` 操作确定集群是否拥有最新的引擎版本。

参数

- **--resource-identifier**—可选。资源 (集群) 的 ARN。如果省略该参数，则列出所有集群的待处理维护操作。
- **--region**—可选。例如，您要其中运行此操作的 AWS 区域，`us-east-1`。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb describe-pending-maintenance-actions \
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster \
  --region us-east-1
```

对于 Windows：

```
aws docdb describe-pending-maintenance-actions ^
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
  --region us-east-1
```

此操作的输出将类似于下文。

```
{
  "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"
        }
      ]
    }
  ]
}
```

```
}  
]  
}
```

如果您的集群有待处理维护，请继续执行[更新集群的引擎版本](#) (p. 263) 中的步骤。

更新集群的引擎版本

我们提供两种类型的更新：

- 补丁更新：这是同一引擎版本中的更新。
- 引擎升级：这会将您从一个引擎版本升级到另一个引擎版本，例如，从版本 3.6 升级到版本 4.0。

在本文中，我们将解释如何使用 Amazon Web Services Management Console 或者 Amazon CLI。您可以立即对其进行更新，也可以在集群的下一个维护时段内进行更新。要确定您的引擎是否需要更新，请参阅[确定待处理维护](#) (p. 261)。请注意，当您应用更新时，您的集群将经历一些停机时间。

如果您正在尝试从主要引擎版本升级到另一个引擎版本，例如 3.6 升级到 4.0，请参阅[使用将您的 Amazon DocumentDB 集群从 3.6 升级到 4.0 Amazon Database Migration Service](#) (p. 112)。

要获取集群引擎版本的最新补丁更新，有两个配置要求：

- 集群状态必须为可用。
- 该集群必须运行较早的引擎版本。

使用 Amazon Web Services Management Console

以下过程使用控制台将补丁更新应用于集群的引擎版本。您可以选择立即更新，也可以在集群的下一个维护时段内更新。

1. 登录 Amazon Web Services Management Console，然后打开 Amazon DocumentDB B 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。在集群列表中，选择要升级的集群左侧的按钮。集群状态必须为可用。

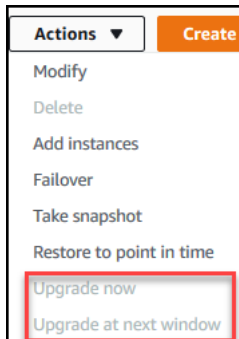
Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标



)。

3. 从 Actions (操作) 菜单中，选择以下选项之一。仅当您选择的集群未运行最新的引擎版本时，才可以选择这些菜单选项。



- 立即升级— 立即启动升级过程。集群升级到最新引擎版本时，集群将离线一段时间。
- 在下一个窗口升级— 在集群的下一维护时段内启动升级过程。集群升级到最新引擎版本时，它将离线一段时间。

4. 当确认窗口打开时，选择以下选项之一：
- 升级— 根据在上一步中选择的时间表将集群升级到最新的引擎版本。
 - Cancel— 取消集群的引擎升级并继续使用集群的当前引擎版本。

使用 Amazon CLI

您可以使用对集群应用补丁更新Amazon CLI还有`apply-pending-maintenance-action`操作带以下参数。

参数

- **--resource-identifier** – 必填项。您要升级的Amazon DocumentDB B 集群的 ARN。
- **--apply-action** – 必填项。允许使用以下值。要升级集群的引擎版本，请使用 `db-upgrade`。
 - **db-upgrade**
 - **system-update**
- **--opt-in-type** – 必填项。允许使用以下值。
 - `immediate`— 立即应用维护操作。
 - `next-maintenance`— 在下一维护时段内应用维护操作。
 - `undo-opt-in`—取消所有现有的`next-maintenance`选择加入请求。

Example

以下示例补丁更新了的引擎版本`sample-cluster`升级至版本 4.0.0。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于以下内容：

```
{  
  "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"  
      }  
    ]  
  }  
}
```

停止和启动Amazon DocumentDB B 集群

停止和启动 Amazon DocumentDB 集群可以帮助您控制开发和测试环境的成本。您无需在每次使用 Amazon DocumentDB 时都创建和删除集群和实例，而是在不需要时暂时停止集群中的所有实例。然后，当您恢复测试时，可以再次启动它们。

主题

- [停止和启动集群概述 \(p. 265\)](#)
- [使用 Amazon Web Services Management Console \(p. 265\)](#)
- [使用 Amazon CLI \(p. 266\)](#)
- [可以在已停止的集群上执行的操作 \(p. 267\)](#)

停止和启动集群概述

在不需要使用 Amazon DocumentDB 集群期间，您可以同时停止该集群中的所有实例。然后，您可以在需要使用时再次启动集群。启动和停止简化了用于开发、测试或不需要持续可用性的类似活动的集群的设置和停用过程。您可以使用 Amazon Web Services Management Console 或 Amazon CLI 通过单个操作停止和启动集群，而不考虑集群中的实例数量。

当您的集群停止后，集群存储卷保持不变。您只需在指定的保留时段内为集群存储、手动快照和自动备份存储付费。您无需为任何实例小时数付费。Amazon DocumentDB 会在七天后自动启动您的集群，以使其不会落后于任何所需的维护更新。当您的集群在 7 天后启动时，您将再次开始为集群中的实例付费。当您的集群停止时，您无法查询您的存储卷，因为查询需要实例处于可用状态。

当 Amazon DocumentDB 集群停止时，不能以任何方式修改该集群及其实例。这包括添加或删除实例，或者删除集群。

使用 Amazon Web Services Management Console

以下过程显示如何停止有一个或多个实例处于可用状态的集群或启动已停止的集群。

停止或启动 Amazon DocumentDB 集群

1. 登录 Amazon Web Services Management Console，然后打开 Amazon DocumentDB B 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

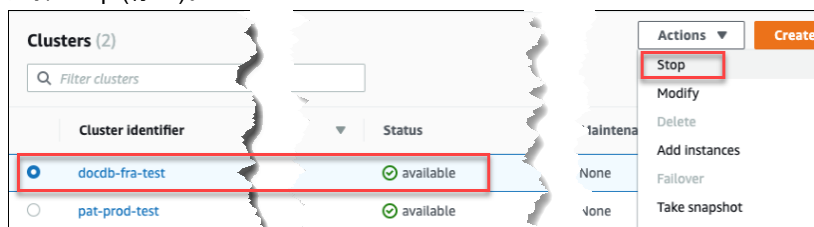
Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在集群列表中，选择要停止或启动的集群名称左侧的按钮。
4. 选择操作，然后选择您要在集群上执行的操作。

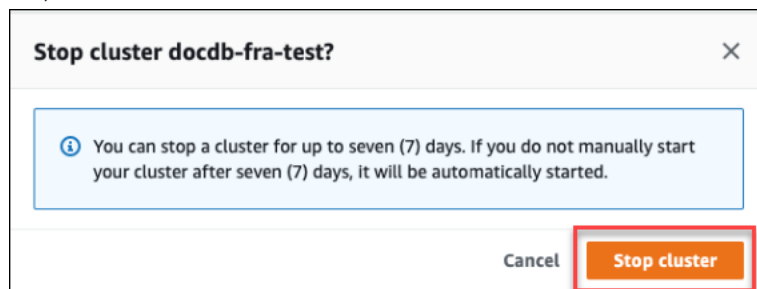
- 如果您要停止集群且集群可用，则：

- a. 选择 Stop (停止)。

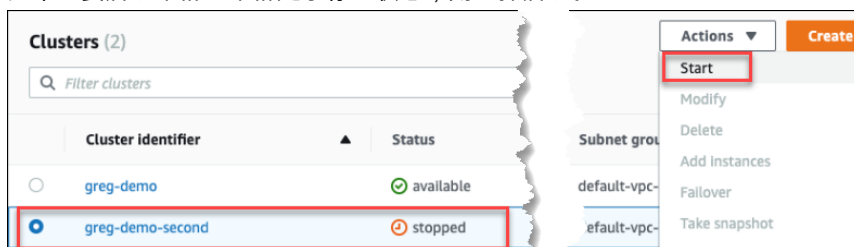


为避免激活故障转移机制，停止操作先停止副本实例，然后停止主实例。

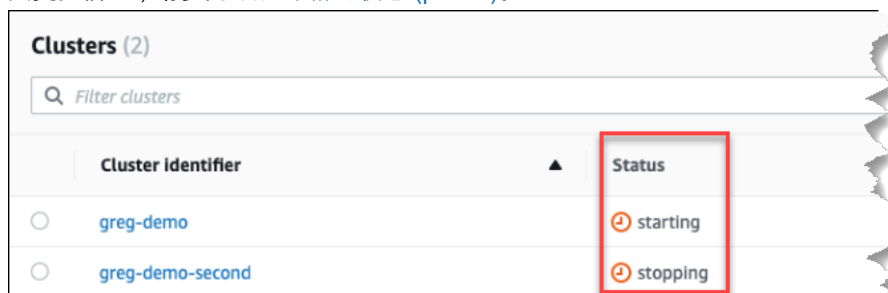
- b. 在确认对话框中，确认您要通过选择 Stop cluster (停止集群) 停止该集群；或者要保持集群运行，则选择取消。



- 如果您要启动集群且集群处于停止状态，则选择启动。



5. 监控集群的状态及其实例。如果您启动集群，则当集群及其实例处于可用时，您可以继续使用集群。有关更多信息，请参阅 [确定集群的状态 \(p. 245\)](#)。



使用 Amazon CLI

以下代码示例显示如何停止有一个或多个实例处于可用状态的集群或启动已停止的集群。

要使用 Amazon CLI 停止有一个或多个可用实例的集群，请使用 `stop-db-cluster` 操作。要启动已停止的集群，请使用 `start-db-cluster` 操作。这两个操作都使用 `--db-cluster-identifier` 参数。

参数：

- `--db-cluster-identifier` – 必填项。要停止或启动的集群的名称。

Example — 使用停止集群Amazon CLI

以下代码停止集群 `sample-cluster`。集群必须有一个或多个实例处于可用状态。

对于 Linux、macOS 或 Unix：

```
aws docdb stop-db-cluster \
```

```
--db-cluster-identifier sample-cluster
```

对于 Windows :

```
aws docdb stop-db-cluster ^  
  --db-cluster-identifier sample-cluster
```

Example — 要启动集群，请使用Amazon CLI

下面的代码启动集群 *sample-cluster*。集群当前必须处于停止状态。

对于 Linux、macOS 或 Unix :

```
aws docdb start-db-cluster \  
  --db-cluster-identifier sample-cluster
```

对于 Windows :

```
aws docdb start-db-cluster ^  
  --db-cluster-identifier sample-cluster
```

可以在已停止的集群上执行的操作

当 Amazon DocumentDB 集群停止时，您可以执行一个 point-in-time 还原到指定的自动备份保留时段中的任意时间点。有关执行以下操作的详细信息 point-in-time 恢复，请参阅[还原到某个时间点 \(p. 217\)](#)。

在停止 Amazon DocumentDB 集群后，您无法修改该集群或其任何实例的配置。您也无法在该集群中添加或删除实例，或者，如果仍具有任何关联的实例，则无法删除该集群。您必须在执行任何此类管理操作之前启动该集群。

Amazon DocumentDB 仅在再次启动后，才会将任何计划的维护应用于停止的集群。在 7 天后，Amazon DocumentDB 自动启动停止的集群，以使其维护状态不会落后太多。当集群重新启动后，您将再次开始为集群中的实例付费。

当集群停止时，Amazon DocumentDB 不会执行任何自动备份，也不会延长备份保留期。

删除 Amazon DocumentDB 集群

您可以使用删除 Amazon DocumentDB 集群Amazon Web Services Management Console或者Amazon CLI。要删除集群，集群必须处于可用 状态，且不得有任何与其关联的实例。如果集群已停止，则首先启动集群，等待集群进入可用 状态，然后删除集群。有关更多信息，请参阅 [停止和启动Amazon DocumentDB B 集群 \(p. 265\)](#)。

删除保护

为了防止您的集群遭到意外删除，您可以启用删除保护。在使用控制台创建集群时，将默认启用删除保护。但是，如果您使用 Amazon CLI 创建集群，将默认禁用删除保护。

无论您是使用控制台还是使用控制台执行删除操作，Amazon DocumentDB 都会对集群强制执行删除保护 Amazon CLI。如果已启用删除保护，则无法删除集群。要删除启用了删除保护的实例，首先请修改该集群并禁用删除保护。

当在集群上启用数据保护的情况下使用控制台时，无法删除该集群的最后一个实例，因为这样也会删除集群。您可以使用 Amazon CLI 删除受删除保护的集群的最后一个实例。但是，集群本身仍然存在，您的数据将被保留。您可以通过为集群创建新实例来访问数据。有关启用和禁用删除保护的更多信息，请参阅：

- [创建Amazon DocumentDB 集群 \(p. 247\)](#)
- [修改Amazon DocumentDB 集群 \(p. 259\)](#)

使用删除Amazon DocumentDB B 集群Amazon Web Services Management Console

要使用 Amazon Web Services Management Console删除集群，必须禁用删除保护。

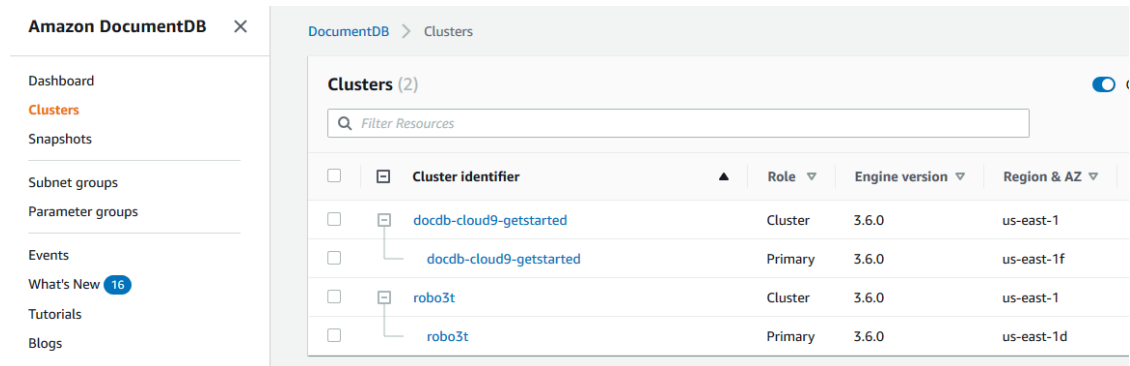
确定集群是否已启用删除保护：

1. 登录Amazon Web Services Management Console，然后打开Amazon DocumentDB B 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 请注意，在“群集”导航框中，列集群标识符显示集群和实例。实例列在群集下方，类似于下面的屏幕截图。



4. 选择集群的名称，然后选择 Configuration (配置) 选项卡。在 Cluster details (集群详细信息) 部分中，找到 Deletion protection (删除保护)。如果已启用删除保护，则修改集群以禁用删除保护。有关修改集群的信息，请参阅 [修改Amazon DocumentDB 集群 \(p. 259\)](#)。

禁用删除保护后，您就可以删除该集群了。

删除集群：

1. 在导航窗格中，选择 Clusters。
2. 通过检查实例列确定集群是否具有任何实例。在删除集群之前，必须先删除其所有实例。有关更多信息，请参阅 [删除 Amazon DocumentDB 实例 \(p. 300\)](#)。
3. 根据您的集群是否具有任何实例，请执行以下步骤之一。
 - 如果集群没有实例，请选择集群名称左侧的按钮，然后选择 Actions (操作)。从下拉菜单中，选择 Delete (删除)。填写删除 <集群名称> 对话框，然后选择删除。
 - 如果集群具有一个或多个实例，请执行以下操作：
 - a. 在导航窗格中，选择 Instances (实例)。
 - b. 删除集群中的每个实例。当您删除最后一个实例时，集群也将被删除。有关删除实例的信息，请参阅 [删除 Amazon DocumentDB 实例 \(p. 300\)](#)。

删除集群需要几分钟时间。要监控集群的状态，请参阅[监控 Amazon DocumentDB 集群的状态 \(p. 366\)](#)。

使用删除Amazon DocumentDB B 集群Amazon CLI

您不能删除具有任何关联实例的集群。要确定哪些实例与您的集群关联，请运行 `describe-db-clusters` 命令并删除集群的所有实例。然后，在需要时禁用集群的删除保护，最后删除集群。

1. 首先，删除集群的所有实例。

要确定需要删除的实例，请运行以下命令。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[
  [
    "sample-cluster",
    [
      "sample-instance-1",
      "sample-instance-2"
    ]
  ]
]
```

如果要删除的集群包含任何实例，请如下所示删除它们。

```
aws docdb delete-db-instance \
  --db-instance-identifier sample-instance
```

2. 接下来，禁用删除保护。

使用 Amazon CLI 删除集群的所有实例将不会删除该集群。您还必须删除集群，但仅当删除保护处于禁用状态时才可以执行此操作。

要确定集群是否已启用删除保护，请运行以下命令。

Tip

要查看所有Amazon DocumentDB B 集群的删除保护状态，请省略`--db-cluster-identifier`参数。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[DBClusterIdentifier,DeletionProtection]'
```

此操作的输出将类似于下文。

```
[
  [
    "sample-cluster",
    "true"
  ]
]
```

如果集群已启用删除保护，则修改集群并禁用删除保护。要对集群禁用删除保护，请运行以下命令。

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --no-deletion-protection \  
  --apply-immediately
```

3. 最后，删除集群。

禁用删除保护后，您就可以删除该集群了。要删除集群，请使用带以下参数的 `delete-db-cluster` 操作。

- **--db-cluster-identifier** – 必填项。要删除的集群的标识符。
- **--final-db-snapshot-identifier**—可选。如果您需要最终快照，则必须包含该参数和最终快照的名称。必须包含 `--final-db-snapshot-identifier` 或 `--skip-final-snapshot`。

命名约束：

- 长度为 [1-63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 亚马逊 RDS、Amazon Neptune 和 Amazon DocumentDB 中的所有集群必须是唯一的 Amazon Web Services 账户，每个区域。
- **--skip-final-snapshot**—可选。仅当您不想在删除集群之前拍摄最终快照时，才使用此参数。默认设置是拍摄最终快照。必须包含 `--final-db-snapshot-identifier` 或 `--skip-final-snapshot`。

以下 Amazon CLI 代码删除具有最终快照的集群 `sample-cluster`。如果有任何实例与集群关联，或者如果已启用删除保护，则此操作将失败。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --final-db-snapshot-identifier sample-cluster-final-snapshot
```

对于 Windows：

```
aws docdb delete-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Example

以下 Amazon CLI 代码将删除集群 `sample-cluster` 而不拍摄最终快照。

对于 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --skip-final-snapshot
```

对于 Windows :

```
aws docdb delete-db-cluster ^  
  --db-cluster-identifier sample-cluster ^  
  --skip-final-snapshot
```

`delete-db-cluster` 操作的输出是您要删除的集群。

删除集群需要几分钟时间。要监控集群的状态，请参阅[监控集群状态 \(p. 366\)](#)。

扩展Amazon DocumentDB

Amazon DocumentDB 使您能够根据需要扩展集群中的存储和计算。本节介绍如何使用存储扩展、实例扩展和读取扩展来管理 Amazon DocumentDB 集群和实例的性能和扩展。

主题

- [存储扩展 \(p. 271\)](#)
- [实例扩展 \(p. 271\)](#)
- [读取扩展 \(p. 271\)](#)
- [写入扩展 \(p. 271\)](#)

存储扩展

Amazon DocumentDB 存储自动使用您的集群卷中的数据进行扩展。当数据量增多时，您的集群卷存储将以 10 GiB 增量递增，最长达 64 TiB。

实例扩展

您可以通过修改集群中每个实例的实例类，来按需扩展 Amazon DocumentDB 集群。Amazon DocumentDB 支持多个针对Amazon DocumentDB B 进行了优化的实例类。

有关更多信息，请参阅 [修改 Amazon DocumentDB 实例 \(p. 295\)](#)。

读取扩展

您可以通过在 Amazon DocumentDB 集群中最多创建 15 个 Amazon DocumentDB 副本来实现该集群的读取扩展。每个 Amazon DocumentDB 副本均返回集群卷中的相同数据，且副本滞后时间最短 — 通常少于主实例写入更新后的 100 毫秒。当读取流量增大时，可创建额外 Amazon DocumentDB 副本并直接连接到这些副本以为集群分配读取负载。Amazon DocumentDB 副本不必具有与主实例相同的实例类。

有关更多信息，请参阅 [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)。

要使用 Amazon DocumentDB 读取规模，我们建议您以副本集的形式连接到集群，并使用驱动程序的内置读取首选项功能将读取分配到副本实例。有关详细信息，请参阅 [作为副本集连接到 Amazon DocumentDB \(p. 454\)](#)。

写入扩展

您可以通过增加集群主实例的大小来扩展 Amazon DocumentDB 集群的写入容量。此部分提供了两种方法来根据您的需求扩展集群的主实例。第一种方法旨在最大程度地减小应用程序影响，但需要执行更多步骤才能完成。第二种方法经过了优化，需要的步骤更少，因此更简单，但它会对应用程序产生更多的潜在影响，这需要您做出权衡。

根据您的应用程序，可以从下面选择最适合您的方法。有关可用实例大小和成本的更多信息，请参阅[Amazon DocumentDB](#)页面。

1. 针对高可用性和性能进行优化— 如果您要在中连接到您的集群[副本集模式 \(p. 454\)](#) (推荐)，您可以使用以下过程来最大程度地减小在扩展主实例时对应用程序的影响。此方法可以最大程度地减小影响，因为它可确保集群的可用性保持一个较高的水平甚至更高，并且读取扩展目标将作为实例添加到集群中，而不是进行就地更新。
 - a. 将较大的实例类型的一个或多个副本添加到集群中 (请参阅[??? \(p. 288\)](#))。我们建议所有副本具有与主实例相同的实例类型甚至更大的实例类型。这可避免因故障转移到较小的实例类型而意外降低写入性能。对于大多数客户而言，这意味着暂时将其集群中的实例数增加一倍，然后在扩展完成后删除较小的副本。
 - b. 将所有新副本上的故障转移层设置为优先级零，并确保较小实例类型的副本具有最高的故障转移优先级。有关更多信息，请参阅[??? \(p. 313\)](#)。
 - c. 启动手动故障转移，这会将其中一个新副本提示为主实例。有关更多信息，请参阅[??? \(p. 314\)](#)。

Note

这将导致您的集群停机约 30 秒。请相应地做好规划。

- d. 从集群中删除小于新主实例的实例类型的所有副本。
- e. 将所有实例的故障转移层设置回相同的优先级 (通常，这意味着将其设置回 1)。

例如，假设您的集群当前包含三个 `r5.large` 实例 (一个主实例和两个副本实例)，并且您希望扩展到一个 `r5.xlarge` 实例类型。为此，您首先将三个 `r5.xlarge` 副本实例添加到集群中，然后将新 `r5.xlarge` 副本的故障转移层设置为零。接下来，您将启动手动故障转移 (这意味着您的应用程序将停机约 30 秒)。在故障转移完成后，您将从集群中删除所有三个 `r5.large` 实例，并让集群扩展到 `r5.xlarge` 实例。

为了帮助优化成本，Amazon DocumentDB 实例以一秒为增量计费，在计费状态发生变化 (例如创建、修改或删除实例) 后，最低收费十分钟。有关更多信息，请参阅最佳实践文档中的[成本优化 \(p. 81\)](#)。

2. 优化以实现简单性— 为了简单起见，这种方法进行了优化。它不会扩展和收缩集群，但可能会暂时减少您的读取容量。

更改副本的实例类别可能会导致该实例在短时间内 (从几秒钟变为不到 30 秒) 内无法处理请求。如果您要在中连接到您的集群[副本集模式 \(p. 454\)](#) (推荐)，那么在扩展操作期间，这会将您的读取容量减少一个副本 (例如，在 3 节点集群中减少到 66% 的容量，或 4 节点集群的 75% 容量等)。

- a. 扩展集群中的一个副本实例。有关更多信息，请参阅[管理实例类 \(p. 283\)](#)。
- b. 等到实例可用 (参见[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#))。

Note

这将导致您的集群停机约 30 秒。请相应地做好规划。

- c. 继续执行步骤 1 和 2，直到所有副本实例逐一扩展。
- d. 启动手动故障转移。这会将其中一个副本提升为主实例。有关更多信息，请参阅[Amazon DocumentDB 故障转移 \(p. 313\)](#)。

Note

这将导致您的集群停机长达 30 秒，但花费的时间通常少于该时间。请相应地做好规划。

- e. 扩展以前的主实例 (现在是副本) 实例。

克隆 Amazon DocumentDB 集群卷

使用 Amazon DocumentDB 克隆功能，您可以创建一个新集群，该集群使用相同的 Amazon DocumentDB 集群卷并具有与原始集群卷相同的数据。该过程旨在快速且经济高效。我们将新集群及其关联的数据卷称为克隆。与使用其他技术（如还原快照）实际复制数据相比，创建克隆速度更快且空间利用效率更高。

Amazon DocumentDB B 支持从已配置的 Amazon DocumentDB B 集群创建 Amazon DocumentDB B 预配置的克隆。使用与源不同的部署配置创建克隆时，源的 Amazon DocumentDB 引擎的最新版本将被用于创建克隆。

当您从 Amazon DocumentDB 集群创建克隆时，将在您的 Amazon 账户-拥有源 Amazon DocumentDB 集群的同一个账户。

主题

- [Amazon DocumentDB 克隆 \(p. 273\)](#)
- [Amazon DocumentDB B 克隆的限制 \(p. 273\)](#)
- [Amazon DocumentDB B 克隆的工作原理 \(p. 274\)](#)
- [创建 Amazon DocumentDB 克隆 \(p. 276\)](#)

Amazon DocumentDB 克隆

Amazon DocumentDB 使用 copy-on-write protocol 创建克隆。此机制占用最少的额外空间来创建初始克隆。首次创建克隆时，Amazon DocumentDB 将保留源数据库集群和新（克隆的）Amazon DocumentDB 集群使用的数据的单个副本。只有当源 Amazon DocumentDB 集群或 Amazon DocumentDB 集群克隆对数据（在 Amazon DocumentDB 存储卷上）进行更改时，才会分配额外的存储空间。了解相关更多信息 copy-on-write 协议，请参阅[Amazon DocumentDB B 克隆的工作原理 \(p. 274\)](#)。

Amazon DocumentDB 克隆非常适合使用您的生产数据快速设置测试环境，且不会有损坏数据的风险。您可以将克隆用于多种类型的应用程序，例如：

- 对潜在的变化（例如模式变化和参数组变化）进行试验，以评估所有影响。
- 执行工作负载密集型操作，例如导出数据或在克隆上运行分析查询。
- 为开发、测试或其他用途创建生产数据库集群的副本。

您可以从同一个 Amazon DocumentDB 集群创建多个克隆。您还可以从另一个克隆创建多个克隆。

创建 Amazon DocumentDB 克隆后，您可以以不同于源 Amazon DocumentDB 集群的方式配置 Amazon DocumentDB 实例。例如，您可能不需要用于开发目的的克隆来满足与源生产 Amazon DocumentDB 集群相同的高可用性要求。在这种情况下，您可以使用单个 Amazon DocumentDB 实例来配置克隆，而不是使用 Amazon DocumentDB 集群使用的多个数据库实例。

当克隆完成测试、开发等使用目的时，您可以将其删除。

Amazon DocumentDB B 克隆的限制

Amazon DocumentDB; 克隆具有以下限制：

- 您可以根据需要创建任意数量的克隆，最多为 Amazon Web Services 区域中允许的最大数据库集群数。但是，在创建 15 个克隆后，下一个克隆是完整副本。克隆操作的作用类似于 point-in-time 恢复。
- 您不能在其他版本中创建克隆 Amazon 来自源 Amazon DocumentDB 集群的区域。
- 您无法从没有数据库实例的 Amazon DocumentDB 集群创建克隆。您只能克隆具有至少一个数据库实例的 Amazon DocumentDB 集群。
- 您可以在与 Amazon DocumentDB 集群不同的 Virtual Private Cloud (VPC) 中创建克隆。不过，这些 VPC 的子网必须映射到相同的可用区。

Amazon DocumentDB B 克隆的工作原理

Amazon DocumentDB 克隆运行于 Amazon DocumentDB 集群的存储层。它使用 copy-on-write 该协议在支持 Amazon DocumentDB 存储卷的底层持久介质方面既快速又节省空间。有关 Amazon DocumentDB 集群卷的更多信息 [管理 Amazon DocumentDB \(p. 243\)](#)。

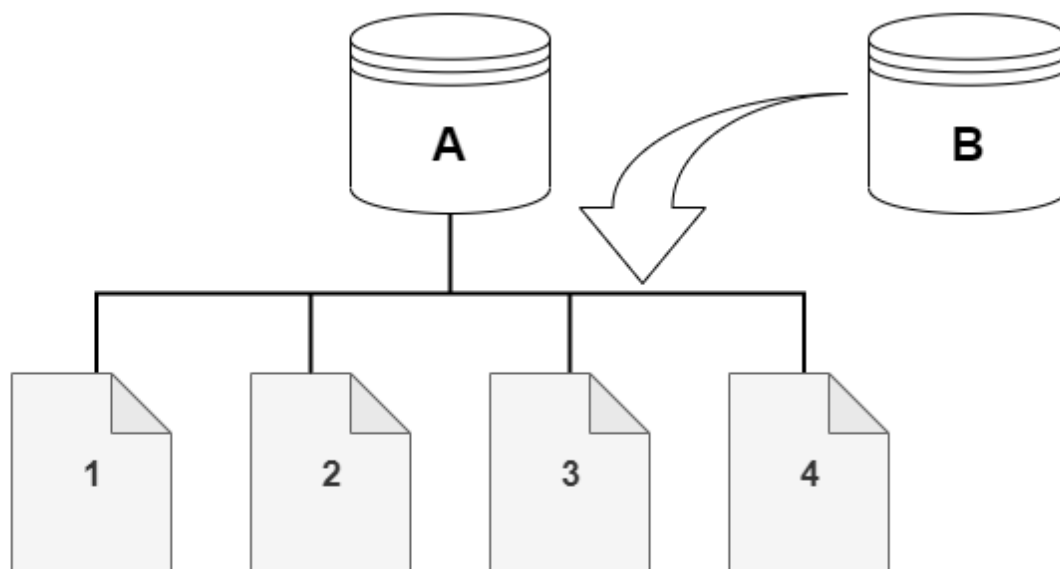
主题

- [了解 copy-on-write protocol \(p. 274\)](#)
- [删除源集群卷 \(p. 276\)](#)

了解 copy-on-write protocol

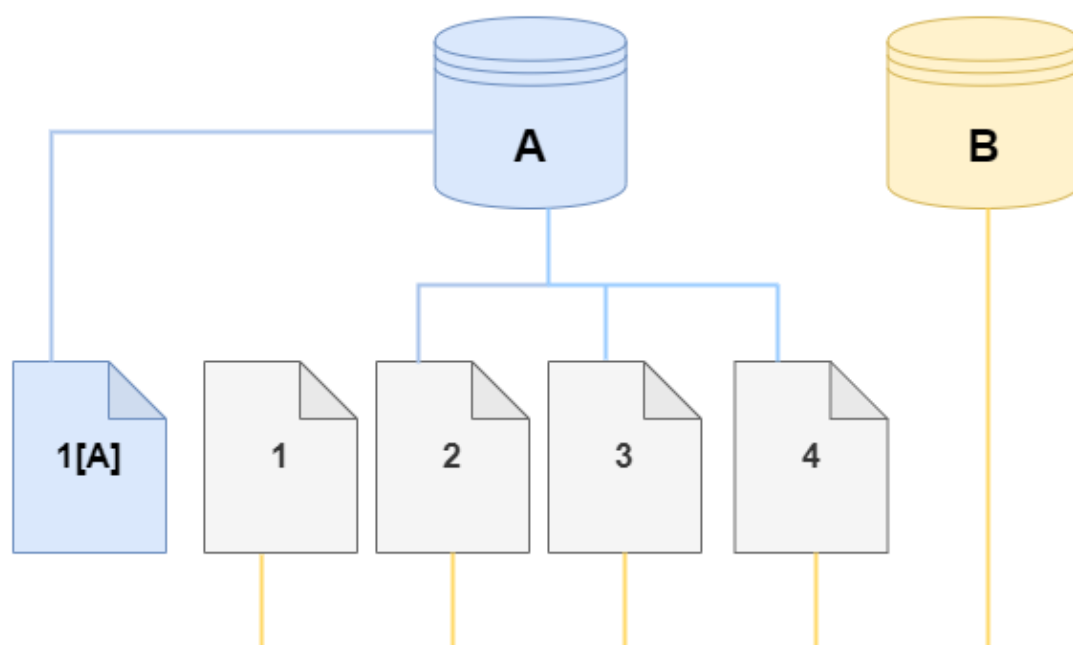
Amazon DocumentDB 集群将数据存储在底层 Amazon DocumentDB 存储卷中的页面中。

例如，在下图中，您可以找到拥有四个数据页（1、2、3 和 4）的 Amazon DocumentDB 集群（A）。假设从 Amazon DocumentDB 集群创建了一个克隆 B。创建克隆时，未复制任何数据。相反，克隆指向与源 Amazon DocumentDB 集群相同的页面集。

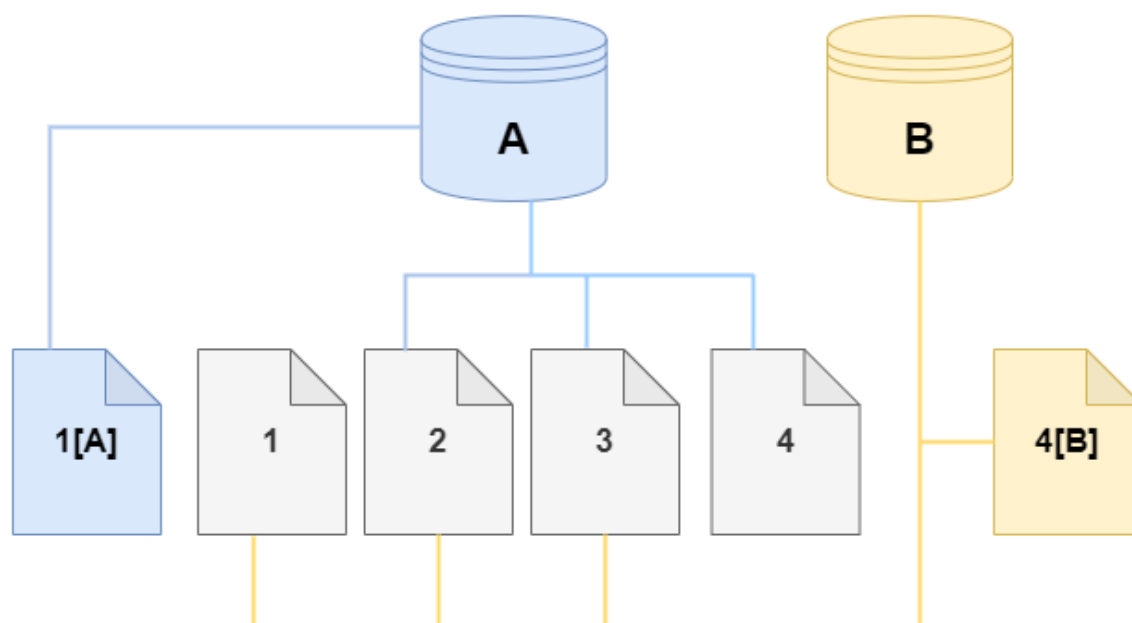


创建克隆时，通常不需要额外的存储空间。这些区域有：copy-on-write 协议在物理存储介质上使用与源段相同的数据段。只有当源段的容量不足以容纳整个克隆段时，才需要额外的存储空间。如果是这种情况，源段将被复制到另一个物理设备。

在下图中，您可以找到 copy-on-write 使用相同集群 A 及其克隆 B 的协议，如前所示。如果您对 Amazon DocumentDB 集群（A）进行更改，那么第 1 页上保存的数据也将随之而发生改变。Amazon DocumentDB 没有写入原始页面 1，而是创建了一个新页面 1 [A]。集群（A）的 Amazon DocumentDB 集群卷现在指向页面 1 [A]、2、3 和 4，而克隆（B）仍引用原始页面。



在克隆上，对存储卷的第 4 页进行了更改。Amazon DocumentDB 没有写入原始页面 4，而是创建了一个新页面 4 [B]。克隆现在指向页面 1、2、3 和页面 4[B]，而集群 (A) 继续指向 1[A]、2、3 和 4。



随着时间推移，当源 Amazon DocumentDB 集群卷和克隆上出现了更多更改时，因此需要更多存储空间来捕获和存储更改。

删除源集群卷

删除与一个或多个克隆关联的源集群卷时，克隆不会受到影响。克隆继续指向以前由源集群卷拥有的页面。

创建Amazon DocumentDB 克隆

您可以在相同的 clone 中创建克隆Amazon账户作为源 Amazon DocumentDB 集群。为此，您可以使用 Amazon Web Services Management Console 或 Amazon CLI 并按照以下步骤进行操作。

通过使用Amazon DocumentDB B 克隆，您可以从预配置的 Amazon DocumentDB 集群中创建预配置的 Amazon DocumentDB 集群克隆

控制台

以下过程介绍了如何使用克隆 Amazon DocumentDB 集群Amazon Web Services Management Console.

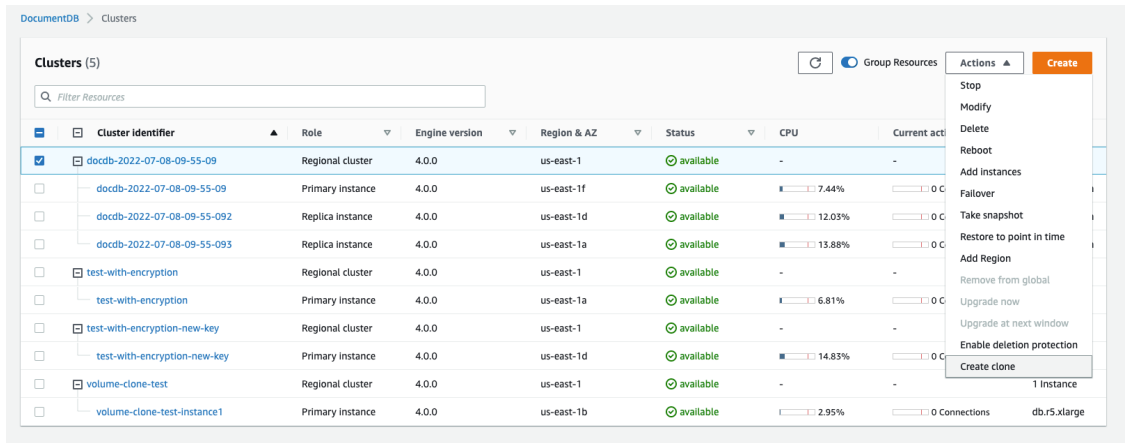
使用创建克隆Amazon Web Services Management Console生成一个包含一个 Amazon DocumentDB 实例的 Amazon DocumentDB B 集群。

这些说明适用于创建克隆的同一 Amazon 账户所拥有的数据库集群。数据库集群必须由同一数据库集群拥有 AmazonAmazon DocumentDB B 不支持跨账户克隆账户。

使用 Amazon 创建 Amazon Web Services Management Console 账户拥有的数据库集群的克隆

1. 登录Amazon Web Services Management Console，然后打开Amazon DocumentDB B 控制台<https://console.aws.amazon.com/docdb>.

- 在导航窗格中，选择 Clusters。
- 从列表中选择您的 Amazon DocumentDB 集群，操作，选择创建克隆。



创建克隆页面打开，您可以在其中配置集群标识符还有一个实例类，以及Amazon DocumentDB B 集群克隆的其他选项。

- 在设置部分，执行以下操作：
 - 对于集群标识符，输入您要为克隆的 Amazon DocumentDB 集群拟定的名称。
 - 对于实例配置，请选择合适的实例类用于您克隆的Amazon DocumentDB B 集群。

Create Clone

You are cloning a DocumentDB cluster. This will create a new DB cluster that includes all of the data from the existing database as well as a writer DB instance.

Settings

Source cluster identifier
docdb-2022-07-08-09-55-09

Cluster identifier
Specify a unique cluster identifier.

Instance configuration

Instance class

2 vCPUs 16GiB RAM

- 对于网络设置，请选择一个子网组用于您的用例和关联的 VPC 安全组。
- 对于南美洲ncryption-at-rest，如果源群集（正在克隆的群集）已启用加密，则克隆的群集也必须启用加密。如果这种情况是真的，那么启用加密选项显示为灰色（已禁用），但是启用加密已选择选项。相反，如果源集群未启用加密，则启用加密选项可用，您可以选择启用或禁用加密。

Network settings

Subnet group
A subnet group is a collection of subnets that are within a VPC.

default ▼

VPC security groups
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.

Select VPC security groups ▼

default ✕

Encryption-at-rest

Enable encryption

☒ Enable encryption

☐ Disable encryption

KMS key ID

(default) aws/rds ▼

Account
12345678910

KMS key ID
example-key-abcdef123

- e. 完成新的群集克隆配置，方法是选择要导出的日志类型（可选），输入用于连接群集的特定端口，并启用防止意外删除群集的保护（默认情况下处于启用状态）。

Log exports

Select the log types to publish to Amazon CloudWatch Logs

☐ Audit logs

☐ Profiler logs

Cluster options

Port

TCP/IP port that is used to connect to the cluster.

27017

Deletion protection

☒ Enable deletion protection

Protects the cluster from being accidentally deleted. While this option is enabled, you can't delete the cluster.

Tags

No tags associated with the cluster.

Add new tag

You can add 50 more tags.

Cancel>Create

- f. 完成 Amazon DocumentDB 集群克隆的所有设置。要了解有关 Amazon DocumentDB 集群和实例设置的更多信息，请参阅[管理 Amazon DocumentDB \(p. 243\)](#)。
5. 选择创建克隆启动您选择的 Amazon DocumentDB B 集群的 Amazon DocumentDB B 克隆。

克隆创建完成后，它将与您的其他 Amazon DocumentDB 集群一起列在控制台中数据库部分并显示其当前状态。当其状态为可用时，您的克隆即可以使用。

Amazon CLI

使用 Amazon CLI 克隆您的 Amazon DocumentDB 集群需要执行几个步骤。

这些区域有：`restore-db-cluster-to-point-in-time` Amazon CLI 你使用的命令会导致一个空的 Amazon DocumentDB 集群中有 0 个 Amazon DocumentDB 实例。也就是说，此命令仅还原 Amazon DocumentDB 集群，而不还原该集群的数据库实例。在克隆可用之后，您可以单独执行此操作。该过程的两个步骤如下：

1. 使用创建克隆 `restore-db-cluster-to-point-in-time` CLI 命令。与此命令一起使用的参数控制正在创建的空 Amazon DocumentDB 集群（克隆）的容量类型和其他详细信息。

2. 使用以下命令为克隆创建 Amazon DocumentDB 实例 `create-db-instance` CLI 命令用于在恢复的 Amazon DocumentDB B 集群中重新创建 Amazon DocumentDB 实例

以下命令假定 Amazon CLI 和您的 Amazon 区域被设置为默认值。这种方法使您无需在每个命令中传递 `--region` 名称。有关更多信息，请参阅[配置 Amazon CLI](#)。您还可以在后面的每个 CLI 命令中指定 `--region`。

主题

- [创建克隆 \(p. 280\)](#)
- [检查状态并获取克隆的详细信息 \(p. 281\)](#)
- [为克隆创建 Amazon DocumentDB 实例 \(p. 281\)](#)
- [用于克隆的参数 \(p. 281\)](#)

创建克隆

您传递给 `restore-db-cluster-to-point-in-time` CLI 命令的特定参数会有所不同。您传递的内容取决于您要创建的克隆类型。

使用以下步骤从预配置的 Amazon DocumentDB 集群中创建预配置的 Amazon DocumentDB 克隆。

创建与源 Amazon DocumentDB 集群引擎模式相同的克隆

- 使用 `restore-db-cluster-to-point-in-time` CLI 命令并指定以下参数的值：
 - `--db-cluster-identifier` – 为克隆选择一个有意义的名称。当您使用克隆时，可以为克隆命名 `restore-db-cluster-to-point-in-time` CLI 命令。
 - `--restore-type` – 使用 `copy-on-write` 创建源数据库集群的克隆。如果没有这个参数，`restore-db-cluster-to-point-in-time` 恢复 Amazon DocumentDB 集群，而不是创建克隆。默认为 `restore-type` 是 `full-copy`。
 - `--source-db-cluster-identifier` – 使用要克隆的源 Amazon DocumentDB 集群的名称。
 - `--use-latest-restorable-time` – 此值指向克隆的最新可还原卷数据。此参数是必需的 `restore-type copy-on-write`，但是，你不能使用 `restore-to-time` parameter 用它。

以下示例从名为 `my-source-cluster` 的集群创建一个名为 `my-clone` 的克隆。

对于 Linux、macOS 或 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
```

对于 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
```

该命令返回包含克隆详细信息的 JSON 对象。在尝试为您的克隆创建数据库实例之前，请检查以确保您的克隆数据库集群可用。有关更多信息，请参阅[检查状态并获取克隆的详细信息 \(p. 281\)](#)。

检查状态并获取克隆的详细信息

您可以使用以下命令检查新创建的空数据库集群的状态。

```
$ aws docdb describe-db-clusters --db-cluster-identifier my-clone --query '*[].[Status]' --output text
```

或者，您可以使用下面的 Amazon CLI 查询获取[为您的克隆创建数据库实例 \(p. 281\)](#)所需的状态和其他值。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone \
  --query '*[].[Status:Status,Engine:Engine,EngineVersion:EngineVersion]'
```

对于 Windows：

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone ^
  --query "*[].[Status:Status,Engine:Engine,EngineVersion:EngineVersion]"
```

此查询返回类似于下述信息的输出：

```
[
  {
    "Status": "available",
    "Engine": "docdb",
    "EngineVersion": "4.0.0",
  }
]
```

为克隆创建 Amazon DocumentDB 实例

使用[create-db-instance](#)CLI 命令为您的克隆创建数据库实例。

这些区域有：--db-instance-class参数仅用于预配置的Amazon DocumentDB B 集群。

对于 Linux、macOS 或 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
```

对于 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
```

用于克隆的参数

下表汇总[restore-db-cluster-to-point-in-time](#)克隆Amazon DocumentDB B 集群。

参数	描述
<code>--source-db-cluster-identifier</code>	使用要克隆的源 Amazon DocumentDB 集群的名称。
<code>--db-cluster-identifier</code>	为克隆选择一个有意义的名称。您可以使用 <code>restore-db-cluster-to-point-in-time</code> 命令为您的克隆命名。然后将此名称传递给 <code>create-db-instance</code> 命令。
<code>--restore-type</code>	指定 <code>copy-on-write</code> 作为 <code>--restore-type</code> 创建源数据库集群的克隆，而不是还原源 Amazon DocumentDB 集群。
<code>--use-latest-restorable-time</code>	此值指向克隆的最新可还原卷数据。

了解 Amazon DocumentDB B 集群容错能力

Amazon DocumentDB 集群设计为具有容错能力。每个集群的卷在一个集群中跨越多个可用区 Amazon Web Services 区域，每个可用区均包含一个群卷数据副本。此功能意味着您的集群可容忍可用区故障，而不发生任何数据丢失，只是会短暂中断服务。

如果集群中的主实例失败，则 Amazon DocumentDB 将通过两种方式之一来自动故障转移到新的主实例：

- 将现有 Amazon DocumentDB 副本升级为根据每个副本的升级级别设置选择的新主实例，然后为以前的主实例创建替代实例。故障转移到副本实例通常不到 30 秒。在此期间，读取和写入操作可能会短暂中断。要提高集群的可用性，建议您在两个或更多的不同的可用区中创建至少一个或多个 Amazon DocumentDB 副本。
- 创建新的主实例。只有在您的集群中没有副本实例时才会发生这种情况，并且可能需要几分钟才能完成。

如果集群具有一个或多个 Amazon DocumentDB 副本，则 Amazon DocumentDB 副本将在故障事件期间被提升为主实例。故障事件将导致短暂中断，其间的读取和写入操作将失败并引发异常。不过，服务通常会在 120 秒内（经常在 60 秒内）还原。要提高集群的可用性，建议您在两个或更多的不同的可用区中创建至少一个或多个 Amazon DocumentDB 副本。

您可以通过为每个副本分配一个优先级来自定义发生故障后将 Amazon DocumentDB 副本提升为主实例的顺序。优先级介于 0（最高优先级）和 15（最低优先级）之间。如果主实例失败，则具有最高优先级的 Amazon DocumentDB 副本将被提升为新的主实例。您可以随时修改 Amazon DocumentDB 副本的优先级。修改优先级不会触发故障转移。您可以对 `modify-db-instance` 操作使用 `--promotion-tier` 参数。有关自定义实例的故障转移优先级的更多信息，请参阅 [Amazon DocumentDB 故障转移 \(p. 313\)](#)。

多个 Amazon DocumentDB 副本可共享同一个优先级，这会产生多个提升层。如果两个或更多 Amazon DocumentDB 副本共享同一个优先级，则最大的副本将被提升为主副本。如果两个或更多 Amazon DocumentDB 副本共享同一个优先级和大小，则将提升同一提升层中的任意副本。

如果集群不包含任何 Amazon DocumentDB 副本，则将在故障事件期间重新创建主实例。故障事件将导致中断，其间的读取和写入操作将失败并引发异常。创建新的主实例时将还原服务，该操作所需的时间通常在 10 分钟内。将 Amazon DocumentDB 副本提升为主实例要比创建新的主实例快得多。

管理 Amazon DocumentDB 实例

以下主题提供了帮助您管理 Amazon DocumentDB 实例的信息。其中包括有关实例类和状态的详细信息，以及如何创建、删除和修改实例。

主题

- [管理实例类 \(p. 283\)](#)
- [确定实例的状态 \(p. 288\)](#)
- [Amazon DocumentDB 实例生命周期 \(p. 288\)](#)

管理实例类

实例类确定 Amazon DocumentDB (兼容 MongoDB) 实例的计算和内存容量。您需要的实例类取决于您的处理能力和内存要求。

Amazon DocumentDB 支持 R4、R5、R6G、T3 和 T4G 系列的实例类。这些是最新一代的实例类，针对内存密集型应用程序进行了优化。有关这些实例类的规格，请参阅 [实例类规格 \(p. 286\)](#)。

主题

- [确定实例类 \(p. 283\)](#)
- [更改实例的类 \(p. 285\)](#)
- [不同区域支持的实例类 \(p. 285\)](#)
- [实例类规格 \(p. 286\)](#)

确定实例类

要确定实例的类，可以使用 Amazon Web Services Management Console 或 `describe-db-instances` Amazon CLI 操作。

使用 Amazon Web Services Management Console

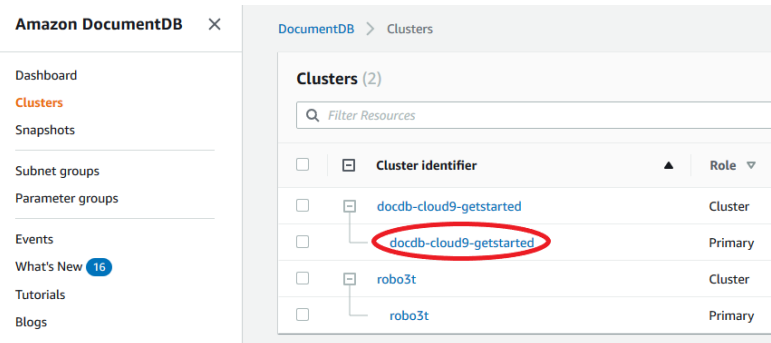
要确定集群实例的实例类，请在控制台中完成以下步骤。

1. 登录到 Amazon Web Services Management Console，然后打开 Amazon DocumentDB 控制台，网址为 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择集群找到您感兴趣的实例。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在集群导航框中，将看到列 Cluster Identifier。您的实例列在 clusters 下，与下面的屏幕截图类似。



4. 在实例列表中，展开集群以查找感兴趣的实例。找到所需的实例。然后，看看 Size 列的实例类，以查看其实例类。

在下图中，实例 robo3t 的实例类为 db.r5.4xlarge。

The screenshot shows the Amazon DocumentDB Clusters console. At the top, there's a breadcrumb 'DocumentDB > Clusters'. Below it, a search bar labeled 'Filter Resources' and a 'Group Resources' toggle are visible. A table lists the clusters. The table has columns: Cluster identifier, Role, Engine version, Region & AZ, Status, Size, and Maintenance. There are two clusters: 'docdb-cloud9-getstarted' and 'robo3t'. Each cluster has a 'Primary' instance. The 'robo3t' primary instance is highlighted with a red circle, showing its 'Size' as 'db.r5.large'.

Cluster identifier	Role	Engine version	Region & AZ	Status	Size	Maintenance
docdb-cloud9-getstarted	Cluster	3.6.0	us-east-1	available	1 Instance	None
docdb-cloud9-getstarted	Primary	3.6.0	us-east-1f	available	db.r5.large	None
robo3t	Cluster	3.6.0	us-east-1	available	1 Instance	None
robo3t	Primary	3.6.0	us-east-1d	available	db.r5.large	None

使用 Amazon CLI

要使用 Amazon CLI 确定实例的类，可使用带有以下参数的 `describe-db-instances` 操作。

- **--db-instance-identifier**— 可选。指定要查找其实例类的实例。如果省略该参数，`describe-db-instances` 将返回最多 100 个实例的描述。
- **--query**— 可选。指定结果中要包含的实例成员。如果省略该参数，则返回所有实例成员。

Example

以下示例将查找实例的实例名称和类 `sample-instance-1`。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
  --db-instance-identifier sample-instance-1
```

对于 Windows：

```
aws docdb describe-db-instances ^
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
  --db-instance-identifier sample-instance-1
```

此操作的输出将类似于下文。

```
[
  [
    "sample-instance-1",
    "db.r5.large"
  ]
]
```

Example

以下示例将查找最多 100 个 Amazon DocumentDB 实例的实例名称和类。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
  --filter Name=engine,Values=docdb
```

对于 Windows：

```
aws docdb describe-db-instances ^
--query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
--filter Name=engine,Values=docdb
```

此操作的输出将类似于下文。

```
[
  [
    "sample-instance-1",
    "db.r5.large"
  ],
  [
    "sample-instance-2",
    "db.r5.large"
  ],
  [
    "sample-instance-3",
    "db.r5.4xlarge"
  ],
  [
    "sample-instance-4",
    "db.r5.4xlarge"
  ]
]
```

有关更多信息，请参阅 [描述 Amazon DocumentDB 实例 \(p. 292\)](#)。

更改实例的类

您可以使用 Amazon Web Services Management Console 或 Amazon CLI 更改实例的实例类。有关更多信息，请参阅 [修改 Amazon DocumentDB 实例 \(p. 295\)](#)。

不同区域支持的实例类

Amazon DocumentDB 支持以下实例类：

- **R6G**— 最新一代的内存优化实例，基于 ARM 的内存优化实例 Amazon Graviton2 处理器的性能提高了多达 30%，成本降低了 5%。仅 4.0.0 引擎支持 Graviton 实例类。
- **R5**— 内存优化实例，在相同实例成本的情况下，其性能比 R4 实例提高 100%。
- **R4**— 上一代内存优化实例。
- **T4G**— 最新一代低成本可突增的通用实例类型，由 ARM 提供支持 Amazon Graviton2 处理器提供基准水平的 CPU 性能，与 T3 实例相比，性价比提高了 35%，非常适合运行 CPU 使用率适中、使用量暂时出现峰值的应用程序。仅 4.0.0 引擎支持 Graviton 实例类。
- **T3**— 低成本可突增通用实例类型，可突增通用实例类型，可根据需要随时突增 CPU 利用率。

有关实例类的详细规格，请参阅 [实例类规格 \(p. 286\)](#)。

特定实例类可能只在部分区域受到支持。下表指定了在每个区域支持 Amazon DocumentDB 的实例类。

不同区域支持的实例类

Region	R6G	R5	R4	T4G	T3
美国东部（俄亥俄）	支持	支持	支持	支持	支持
美国东部（弗吉尼亚北部）	支持	支持	支持	支持	支持

Region	R6G	R5	R4	T4G	T3
美国西部 (俄勒冈)	支持	支持	支持	支持	支持
南美洲 (圣保罗)	支持	支持		支持	支持
亚太地区 (孟买)	支持	支持		支持	支持
亚太地区 (首尔)	支持	支持		支持	支持
亚太地区 (悉尼)	支持	支持		支持	支持
亚太地区 (新加坡)	支持	支持		支持	支持
亚太地区 (东京)	支持	支持		支持	支持
加拿大 (中部)	支持	支持		支持	支持
欧洲 (法兰克福)	支持	支持		支持	支持
欧洲 (爱尔兰)	支持	支持	支持	支持	支持
欧洲 (伦敦)	支持	支持		支持	支持
欧洲 (米兰)	支持	支持		支持	支持
欧洲 (巴黎)	支持	支持		支持	支持
中国 (北京) 区域	支持	支持		支持	支持
中国 (宁夏)	支持	支持		支持	支持
Amazon GovCloud (US)	支持	支持			支持

实例类规格

下表提供了 Amazon DocumentDB 实例类的详细信息。可在表下方找到每个表列的说明。

支持的 Amazon DocumentDB 实例类

实例类	vCPU ¹	内存 (GiB) ²	最大带宽 (Mbps) ³	网络性能 ⁴	支持引擎 ⁵
R6G — 基于 Graviton2 的新一代内存优化实例类					
db.r6g.large	2	16	最多 4,750	最高 10 Gbps	只有 4.0.0
db.r6g.xlarge	4	32	最多 4,750	最高 10 Gbps	只有 4.0.0

实例类	vCPU ¹	内存 (GiB) ²	最大带宽 (Mbps) ³	网络性能 ⁴	支持引擎 ⁵
db.r6g.2xlarge	8	64	最多 4,750	最高 10 Gbps	只有 4.0.0
db.r6g.4xlarge	16	128	4750	最高 10 Gbps	只有 4.0.0
db.r6g.8xlarge	32	256	9,000	12 Gbps	只有 4.0.0
db.r6g.12xlarge	48	384	13,500	20 Gbps	只有 4.0.0
db.r6g.16xlarge	64	512	19000	25 Gbps	只有 4.0.0
R5 — 上一代内存优化实例类					
db.r5.large	2	16	最高 3,500	最高 10 Gbps	3.6.0 和 4.0.0
db.r5.xlarge	4	32	最高 3,500	最高 10 Gbps	3.6.0 和 4.0.0
db.r5.2xlarge	8	64	最高 3,500	最高 10 Gbps	3.6.0 和 4.0.0
db.r5.4xlarge	16	128	3,500	最高 10 Gbps	3.6.0 和 4.0.0
db.r5.8xlarge	32	256	6800	10Gbps	3.6.0 和 4.0.0
db.r5.12xlarge	48	384	7,000	10Gbps	3.6.0 和 4.0.0
db.r5.16xlarge	64	512	13600	20 Gbps	3.6.0 和 4.0.0
db.r5.24xlarge	96	768	14,000	25 Gbps	3.6.0 和 4.0.0
R4 — 上一代内存优化实例类					
db.r4.large	2	15.25	437	最高 10 Gbps	仅限 3.6.0
db.r4.xlarge	4	30.5	875	最高 10 Gbps	仅限 3.6.0
db.r4.2xlarge	8	61	875	最高 10 Gbps	仅限 3.6.0
db.r4.4xlarge	16	122	875	最高 10 Gbps	仅限 3.6.0
db.r4.8xlarge	32	244	875	10Gbps	仅限 3.6.0
db.r4.16xlarge	64	488	14,000	25 Gbps	仅限 3.6.0
T4G — 基于 Graviton2 的最新一代可突增性能实例类					
db.t4g.medium	2	4	最多 2,085	最高 5 Gbps	只有 4.0.0
T3 — 上一代可突增性能实例类					

实例类	vCPU ¹	内存 (GiB) ²	最大带宽 (Mbps) ³	网络性能 ⁴	支持引擎 ⁵
db.t3.medium	2	4	最多 1,536	最高 5 Gbps	3.6.0 和 4.0.0

1. vCPU— 虚拟中央处理器 (CPU) 数。虚拟 CPU 是可用于比较实例类的容量单位。您不再购买或租用特定的处理器并用上数月或数年，而是以小时为单位租用容量。我们的目标是提供一致的 CPU 容量，无论实际的底层硬件是什么。

2. 内存 (GiB)— 为实例分配的 RAM (GB)。内存与 vCPU 之间通常具有一致的比率。

3. 最大带宽 (Mbps)— 以 MB/s 为单位的最大带宽。除以 8 可获得预期吞吐量 (MB/s)。

4. 网络性能— 与其他实例类有关的网络速度。

5. 支持引擎— 支持实例类的 Amazon DocumentDB 引擎。

确定实例的状态

要查看有效实例的状态、其含义以及如何确定您的实例的状态，请参阅[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

Amazon DocumentDB 实例生命周期

Amazon DocumentDB 实例的生命周期包括创建、修改、维护和升级、执行备份和还原、重启和删除实例。本节提供有关如何完成这些过程的信息。

主题

- [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)
- [描述 Amazon DocumentDB 实例 \(p. 292\)](#)
- [修改 Amazon DocumentDB 实例 \(p. 295\)](#)
- [重新启动 Amazon DocumentDB 实例 \(p. 298\)](#)
- [删除 Amazon DocumentDB 实例 \(p. 300\)](#)

将 Amazon DocumentDB 实例添加到集群

您可以使用创建新的 Amazon DocumentDB 实例 Amazon Web Services Management Console 或者 Amazon CLI。要将实例添加到集群，该集群必须处于可用 状态。您无法将实例添加到已停止的集群。如果集群已停止，则首先启动集群，等待集群进入可用 状态，然后添加实例。有关更多信息，请参阅[停止和启动 Amazon DocumentDB B 集群 \(p. 265\)](#)。

Note

如果您使用控制台创建 Amazon DocumentDB 集群，则会同时自动为您创建实例。如果您要创建其他实例，请使用以下过程之一。

使用 Amazon Web Services Management Console 添加实例

使用以下过程使用 Amazon DocumentDB 控制台为集群创建实例。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标

(≡

)。

3. 要选择希望将实例添加到的集群，请选择该集群名称左侧的按钮。
4. 选择 Actions (操作)，然后选择 Add instance (添加实例)。
5. 在 Add instance to: (将实例添加到:)<cluster-name> 页面中，对要添加到集群的每个实例重复以下步骤。您最多可以有 15 个。
 - a. 实例标识符您可以输入此实例的唯一标识符，或允许 Amazon DocumentDB 基于集群标识符提供实例标识符。

实例命名约束：

- 长度为 [1—63] 个字母、数字或连字符。
 - 第一个字符必须是字母。
 - 不能以连字符结束或包含两个连续连字符。
 - 对于每个实例的 Amazon RDS、Neptune 和 Amazon DocumentDB 必须是唯一的。Amazon Web Services 账户，每区域数。
- b. 实例类— 从下拉列表中，为该实例选择所需的实例类型。
 - c. 提升层— 从下拉列表中，为您的实例选择提升层，或者选择没有首选项允许 Amazon DocumentDB 为您的实例设置促销套餐。数字越小，意味着优先级越高。有关更多信息，请参阅[控制故障转移目标 \(p. 313\)](#)。
 - d. 要添加更多实例，请选择 Add additional instances (添加额外实例)，并重复步骤 a、b 和 c。
6. 完成此操作。
 - 要将实例添加到集群，请选择 Create (创建)。
 - 要取消操作，请选择取消。

创建实例需要几分钟时间。您可以使用控制台或 Amazon CLI 查看实例的状态。有关更多信息，请参阅[监控实例的状态 \(p. 368\)](#)。

使用 Amazon CLI 添加实例

将 create-db-instance Amazon CLI 操作与以下参数一起使用，为您的集群创建主实例。

- **--db-instance-class** – 必需。实例的计算和内存容量，例如，db.m4.large。并非所有实例类都可用 Amazon Web Services 区域。
- **--db-instance-identifier** – 必需。标识实例的一个字符串。

实例命名约束：

- 长度为 [1—63] 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。
- 对于每个实例的 Amazon RDS、Neptune 和 Amazon DocumentDB 必须是唯一的。Amazon Web Services 账户，每区域数。
- **--engine** – 必需。必须是 docdb。
- **--availability-zone** – 可选。要在其中创建此实例的可用区。使用此参数在不同可用区中找到您的实例，以增强容错能力。有关更多信息，请参阅[Amazon DocumentDB 高可用性和复制 \(p. 311\)](#)。
- **--promotion-tier** – 可选。此实例的故障转移优先级层。必须介于 0 与 15 之间，数字越小，优先级越高。有关更多信息，请参阅[控制故障转移目标 \(p. 313\)](#)。

1. 首先，确定您可以在哪些可用区中创建您的实例。

如果要在创建实例前指定可用区，请运行以下命令确定哪些可用区可用于您的 Amazon DocumentDB 集群。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
    --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

对于 Windows：

```
aws docdb describe-db-clusters ^  
    --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

此操作的输出将类似于下文。

```
[  
  [  
    "sample-cluster",  
    [  
      "us-east-1c",  
      "us-east-1b",  
      "us-east-1a"  
    ]  
  ]  
]
```

2. 其次，确定您可以在您的区域中创建哪些实例类。

要确定您所在区域中可用的实例类，请运行以下命令。从输出中，为要添加到 Amazon DocumentDB 集群的实例选择实例类。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-orderable-db-instance-options \  
    --engine docdb \  
    --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

对于 Windows：

```
aws docdb describe-orderable-db-instance-options ^  
    --engine docdb ^  
    --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

此操作的输出将类似于下文。

```
[  
  "db.r5.16xlarge",  
  "db.r5.2xlarge",  
  "db.r5.4xlarge",  
  "db.r5.8xlarge",  
  "db.r5.large",  
  "db.r5.xlarge"  
]
```

3. 最后，将实例添加到您的 Amazon DocumentDB 集群。

要将实例添加到您的 Amazon DocumentDB 集群，请运行以下命令。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文。

```
{  
  "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-ABCDEFGHJKLMNOPQRSTUVWXYZ",
    "CACertificateIdentifier": "rds-ca-2019",
    "PromotionTier": 2,
    "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2"
  }
}

```

创建实例需要几分钟时间。您可以使用控制台或 Amazon CLI 查看实例的状态。有关更多信息，请参阅[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

描述 Amazon DocumentDB 实例

您可以使用 Amazon DocumentDB 管理控制台或 Amazon CLI 以查看详细信息，例如与 Amazon DocumentDB 实例相关的连接终端节点、安全组 VPC、证书颁发机构和参数组。

使用 Amazon Web Services Management Console

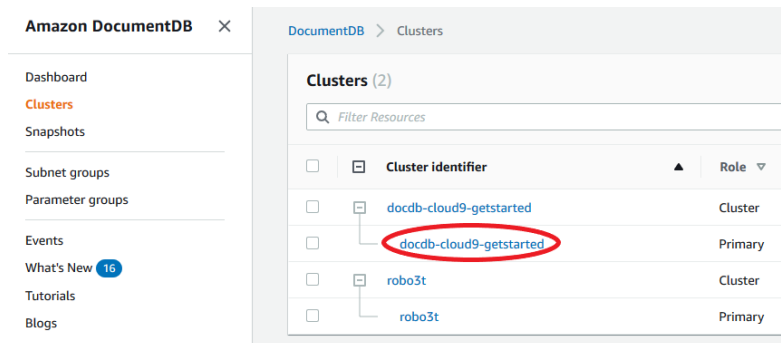
要使用 Amazon Web Services Management Console 查看实例的详细信息，请执行以下步骤。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在 Cluster 导航框中，将看到列 Cluster Identifier。您的实例列在群集下，类似于下面的屏幕截图。



4. 在实例列表中，选择要查看其详细信息的实例的名称。有关实例的信息分为以下几组：
 - 摘要有关实例的一般信息，包括引擎版本、类、状态和所有待处理维护。
 - 连接和安全—Connect (连接)部分列出要使用 mongo shell 或应用程序连接到此实例的连接终端节点。Security Groups (安全组) 部分列出与此实例关联的安全组及其 VPC ID 和描述。
 - 配置—详细信息部分列出实例的配置和状态，包括实例的 Amazon 资源名称 (ARN)、终端节点、角色、类和证书颁发机构。其中还列出实例的安全性和网络设置以及备份信息。Cluster details (集群详细信息) 部分列出此实例所属集群的详细信息。Cluster instances (集群实例) 部分列出属于集群的所有实例，其中包括每个实例的角色和集群参数组状态。

Note

您可以通过选择修改与实例关联的集群修改旁边的簇详细信息标头。有关更多信息，请参阅 [修改 Amazon DocumentDB 集群 \(p. 259\)](#)。

- 监控— 此实例的 CloudWatch Logs 指标。有关更多信息，请参阅 [使用 CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)。
- 活动 & 标签— 近期事件部分列出了此实例的最近事件。Amazon DocumentDB 会记录与集群、实例、快照、安全组和集群参数组相关的事件。此信息包括与每个事件关联的日期、时间和消息。这些区域有：
`<guilabel>Tags</guilabel>`
部分列出附加到此集群的标签。有关更多信息，请参阅 [为 Amazon DocumentDB 资源添加标签 \(p. 352\)](#)。

使用 Amazon CLI

要使用查看 Amazon DocumentDB 实例的详细信息，请使用 Amazon CLI，使用 `describe-db-clusters` 命令，如下示例所示。有关更多信息，请参阅 [DescribeDBInstances](#) 中的 Amazon DocumentDB 资源管理 API 参考。

Note

对于某些管理功能，例如集群和实例生命周期管理，Amazon DocumentDB 利用与 Amazon RDS 共享的操作技术。这些区域有：`filterName=engine,Values=docdb` 筛选器参数仅返回 Amazon DocumentDB 集群。

1. 列出所有 Amazon DocumentDB 实例。

以下 Amazon CLI 代码列出区域中所有 Amazon DocumentDB 实例的详细信息。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --filter Name=engine,Values=docdb
```

对于 Windows：

```
aws docdb describe-db-instances \  
  --filter Name=engine,Values=docdb
```

2. 列出指定 Amazon DocumentDB 实例的所有详细信息

以下代码列出了 `sample-cluster-instance` 的详细信息。将 `--db-instance-identifier` 参数包含在实例名称中，会将输出限制为该特定实例的信息。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-cluster-instance
```

对于 Windows：

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-cluster-instance
```

此操作的输出将类似于以下内容：

```
{
  "DBInstances": [
    {
      "DbiResourceId": "db-BJKKB54PIDV5QFKGV5T3S6GM",
      "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"
}
]
```

修改 Amazon DocumentDB 实例

您可以使用以下任一方法修改 Amazon DocumentDB 实例。Amazon Web Services Management Console 或者 Amazon CLI。要修改某个实例，该实例必须处于可用 状态。您无法修改已停止的实例。如果集群已停止，则首先启动集群，等待实例进入可用 状态，然后进行所需修改。有关更多信息，请参阅[停止和启动 Amazon DocumentDB B 集群](#) (p. 265)。

使用 Amazon Web Services Management Console

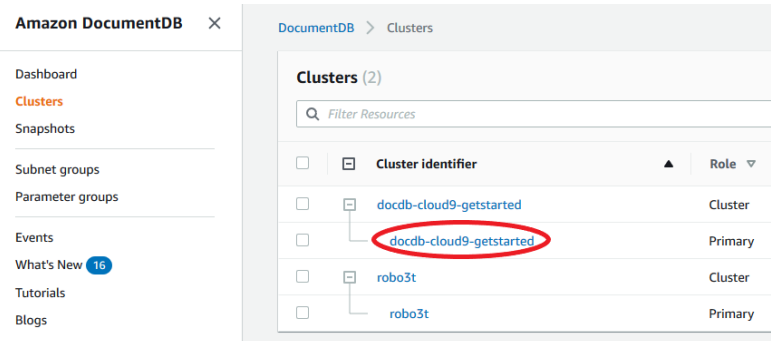
要使用控制台修改特定的 Amazon DocumentDB 实例，请完成以下步骤。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在 Cluster 导航框中，将看到列 Cluster Identifier。您的实例列在群集下，类似于下面的屏幕截图。



4. 选中您要修改的实例左侧的框。
5. 选择 Actions (操作)，然后选择 Modify (修改)。
6. 在 Modify instance: <instance-name> (修改实例: <instance-name>) 窗格中，进行所需更改。您可以进行以下更改：

- 实例规格—实例标识符和类。实例标识符命名约束：

- 实例标识符— 输入您拥有的所有实例的唯一名称。Amazon Web Services 账户在当前区域中。实例标识符必须包含 [1—63] 个字母数字字符或连字符，以字母作为第一个字符，并且不能以连字符结尾，也不能包含两个连续的连字符。
 - 实例类— 从下拉菜单中，为您的 Amazon DocumentDB 实例选择实例类。有关更多信息，请参阅[管理实例类](#) (p. 283)。
 - 证书颁发机构— 此实例的服务器证书。有关更多信息，请参阅[更新您的 Amazon DocumentDB TLS 证书 \(cn-northwest-1\)](#) (p. 173)。
 - 故障转移在故障转移期间，具有最高提升层的实例将被提升为主实例。有关更多信息，请参阅[Amazon DocumentDB 故障转移](#) (p. 313)。
 - 维护将待处理的修改或修补程序应用于集群中的实例的维护时段。
7. 完成后，选择 Continue (继续) 以查看更改摘要。
 8. 在确认您的更改后，您可以立即应用这些更改，也可以在 Scheduling of modifications (修改计划) 下的下一个维护时段内应用这些更改。选择 Modify instance (修改实例) 以保存更改。或者，您可以选择 Cancel (取消) 以放弃更改。

需要几分钟时间才能应用更改。只有在实例状态为 available (可用) 时，才能使用实例。您可以使用控制台或 Amazon CLI 监控实例状态。有关更多信息，请参阅[监控 Amazon DocumentDB 实例的状态](#) (p. 368)。

使用 Amazon CLI

使用修改特定的 Amazon DocumentDB 实例 Amazon CLI，使用 `modify-db-instance` 具有以下参数。有关更多信息，请参阅 [ModifyDBInstance](#)。以下代码为实例 `sample-instance` 将实例类修改为 `db.r5.large`。

参数

- **--db-instance-identifier** – 必需。要修改的实例的标识符。
- **--db-instance-class** – 可选。实例的新计算和内存容量；例如，`db.r5.large`。并非所有实例类都可用 Amazon Web Services 区域。如果您修改实例类，则在更改期间会发生中断。更改在下一个维护时段内应用，除非 `ApplyImmediately` 对于此请求，则指定为 `true`。
- **--apply-immediately** 要么 **--no-apply-immediately** – 可选。指定应立即应用此修改，还是等到下一个维护时段。如果省略此参数，则会在下一个维护时段执行修改。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-instance \
  --db-instance-identifier sample-instance \
  --db-instance-class db.r5.large \
  --apply-immediately
```

对于 Windows：

```
aws docdb modify-db-instance ^
  --db-instance-identifier sample-instance ^
  --db-instance-class db.r5.large ^
  --apply-immediately
```

此操作的输出将类似于下文。

```
{
  "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": "ABCDEFGHJKLMN"
    },
    "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/wJa1rXUtnFEMI/K7MDENG/
bPxRficyEXAMPLEKEY",
    "DbiResourceId": "db-ABCDEFGHJKLMNQRSTUvwxyz",
    "CACertificateIdentifier": "rds-ca-2019",
    "PromotionTier": 1,
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:sample-instance-1",
    "EnabledCloudwatchLogsExports": [
      "profiler"
    ]
  }
}

```

需要几分钟时间才能应用修改。只有在实例状态为 `available` (可用) 时, 才能使用实例。您可以使用 Amazon Web Services Management Console 或 Amazon CLI 监控实例状态。有关更多信息, 请参阅 [监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

重新启动 Amazon DocumentDB 实例

有时您可能需要重启 Amazon DocumentDB 实例，通常是出于维护目的。如果您进行了某些更改，例如更改与集群关联的集群参数组，则必须重启集群中的实例才能使更改生效。您可以使用 Amazon Web Services Management Console 或 Amazon CLI 重启指定的实例。

重启实例会重新启动数据库引擎服务。重启将导致短暂中断，在此期间，实例的状态将设置为 `rebooting`。重新启动完成后，即会创建 Amazon DocumentDB 事件。

重启实例不会导致故障转移。要对 Amazon DocumentDB 集群进行故障转移，请使用 Amazon Web Services Management Console 或者 Amazon CLI 的 `failover-db-cluster`。有关更多信息，请参阅 [Amazon DocumentDB 故障转移 \(p. 313\)](#)。

如果实例未处于 `available` (可用) 状态，则无法重启该实例。数据库可能会由于几个原因而不可用，例如，以前请求的修改或维护时段操作。有关实例状态的更多信息，请参阅 [监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

使用 Amazon Web Services Management Console 重启实例

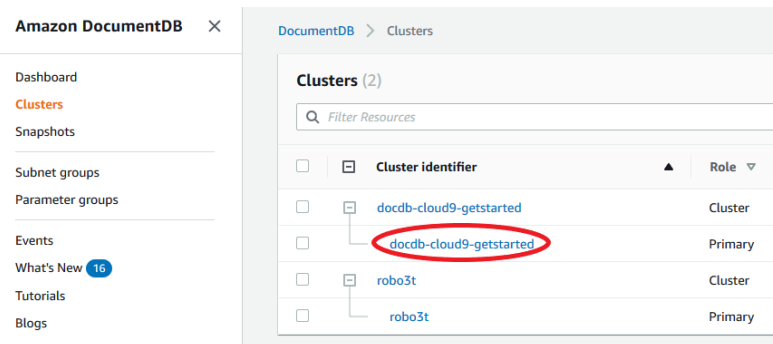
以下过程将使用控制台重启您指定的实例。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在 Cluster 导航框中，将看到列 Cluster Identifier。您的实例列在群集下，类似于下面的屏幕截图。



4. 选中您要重启的实例左侧的框。
5. 选择 Actions (操作)、Reboot (重启)，然后选择 Reboot (重启) 以确认重启。

实例重启需要几分钟时间。只有在实例状态为 `available` (可用) 时，才能使用实例。您可以使用控制台或 Amazon CLI 监控实例状态。有关更多信息，请参阅 [监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

使用 Amazon CLI 重启实例

要重启 Amazon DocumentDB 实例，请使用 `reboot-db-instance` 使用 `--db-instance-identifier` 参数。此参数指定要重启的实例的标识符。

以下代码将重启实例 `sample-instance`。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb reboot-db-instance \
  --db-instance-identifier sample-instance
```

对于 Windows：

```
aws docdb reboot-db-instance ^
  --db-instance-identifier sample-instance
```

此操作的输出将类似于下文。

```
{
  "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-ABCDEFGHIJKLMNQRSTUWXYZ",
    "CACertificateIdentifier": "rds-ca-2019",
```

```
"PromotionTier": 1,
"DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance",
"EnabledCloudwatchLogsExports": [
  "profiler"
]
}
```

实例重启需要几分钟时间。只有在实例状态为 `available` (可用) 时, 才能使用实例。您可以使用控制台或 Amazon CLI 监控实例状态。有关更多信息, 请参阅[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

删除 Amazon DocumentDB 实例

您可以使用删除 Amazon DocumentDB 实例 Amazon Web Services Management Console 或者 Amazon CLI。要删除某个实例, 该实例必须处于可用 状态。您无法删除已停止的实例。如果包含您的实例的 Amazon DocumentDB 集群已停止, 则首先启动集群, 等待实例进入。可用, 然后删除实例。有关更多信息, 请参阅[停止和启动 Amazon DocumentDB B 集群 \(p. 265\)](#)。

Note

Amazon DocumentDB 会将所有数据都存储在集群卷中。即使您从集群中删除所有实例, 数据仍会保留在该集群卷中。如果您需要再次访问数据, 则可随时向该集群中添加实例, 并在停止处恢复操作。

删除保护

删除 Amazon DocumentDB 集群的最后一个实例也会删除集群以及与该集群关联的自动快照和持续备份。无论您使用还是执行删除操作, Amazon DocumentDB 都会为集群强制执行删除保护。Amazon Web Services Management Console 或者 Amazon CLI。如果已启用删除保护, 则无法删除集群。

要删除启用了删除保护的实例, 首先请修改该集群并禁用删除保护。有关更多信息, 请参阅[删除 Amazon DocumentDB 集群 \(p. 267\)](#)。

使用 Amazon Web Services Management Console 删除实例

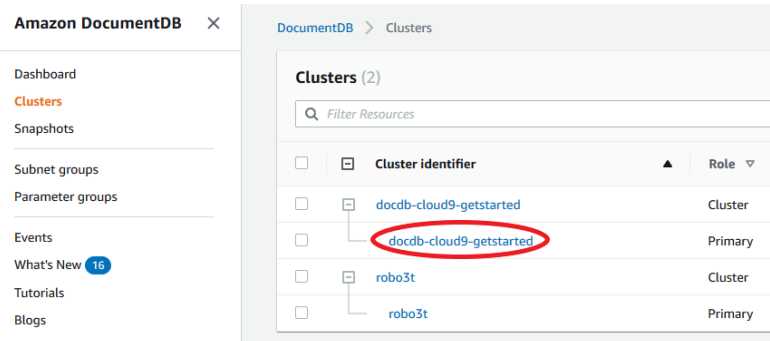
以下过程使用控制台删除指定的 Amazon DocumentDB 实例。

1. 登录到 Amazon Web Services Management Console, 然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中, 选择 Clusters。

Tip

如果您在屏幕左侧没有看到导航窗格, 请在页面左上角选择菜单图标 (≡)。

3. 在 Cluster 导航框中, 将看到列 Cluster Identifier。您的实例列在群集下, 类似于下面的屏幕截图。



4. 选中您要删除的实例左侧的框。
5. 选择 Actions (操作)，然后选择 Delete (删除)。
 1. 如果您要删除集群中的最后一个实例：
 - Create final cluster snapshot? (是否创建最终集群快照?) — 选择是如果要在删除集群前创建最后一个快照。否则，请选择 No。
 - 最终快照名称如果您选择创建最终快照，请输入所创建的新集群快照的集群快照标识符。
 - Delete <instance-name> instance? (是否删除 <instance-name> 实例?) — 输入短语删除整个集群在此字段中以确认删除。
 2. 如果您不删除集群中的最后一个实例：
 - Delete <instance-name> instance? (是否删除 <instance-name> 实例?) — 输入短语删除我在此字段中以确认删除。
6. 选择 Delete (删除) 以删除实例。

删除实例需要几分钟时间。要监控实例的状态，请参阅[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

使用 Amazon CLI 删除实例

以下过程使用删除 Amazon DocumentDB 实例：Amazon CLI。

1. 首先，确定您的 Amazon DocumentDB 集群中有多少个实例：

要确定集群中有多少个实例，请运行 `describe-db-clusters` 命令，如下所示。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

此操作的输出将类似于下文。

```
[
  [
    "sample-cluster",
    [
      "sample-instance-1",
      "sample-instance-2"
    ]
  ]
]
```

2. 如果您的 Amazon DocumentDB 集群中有多个实例：

要删除指定的 Amazon DocumentDB 实例，请使用 `delete-db-instance` 命令使用 `--db-instance-identifier` 参数，如下所示。删除实例需要几分钟时间。要监控实例的状态，请参阅[监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)。

```
aws docdb delete-db-instance \
  --db-instance-identifier sample-instance-2
```

此操作的输出将类似于下文。

```
{
  "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-ABCDEFGHIJKLMNQRSTUvwxyz",
    "CACertificateIdentifier": "rds-ca-2019",
    "PromotionTier": 1,
    "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2",
    "EnabledCloudwatchLogsExports": [
      "profiler"
    ]
  }
}

```

3. 如果要删除的实例是 Amazon DocumentDB 集群中的最后一个实例：

如果您删除 Amazon DocumentDB 集群中的最后一个实例，则还会删除该集群以及与该集群关联的自动快照和持续备份。

要删除集群中的最后一个实例，您可以删除此集群并可以选择创建最终快照。有关更多信息，请参阅[删除 Amazon DocumentDB 集群](#) (p. 267)。

管理 Amazon DocumentDB 子网组

Virtual Private Cloud (VPC) 是仅适用于您的虚拟网络。Amazon Web Services 账户。它在逻辑上与 Amazon 云中的其他虚拟网络隔绝。您可以启动 Amazon DocumentDB 集群等资源。您可以为 VPC 指定 IP 地址范围、添加子网、关联安全组以及配置路由表。

子网是 Amazon VPC 中的 IP 地址范围。您可以在指定子网内启动 Amazon 资源。对于必须连接到 Internet 的资源，请使用公有子网。对于不连接到 Internet 的资源，请使用私有子网。有关公有子网和私有子网的更多信息，请参阅[VPC 和子网基础知识](#)中的 Amazon Virtual Private Cloud 用户指南。

数据库子网组是在 VPC 中创建的一组子网，并随后为集群指定这些子网。通过使用子网组，您可以在创建集群时指定特定的 VPC。如果使用 default 子网组，它将涵盖 VPC 中的所有子网。

每个数据库子网组应包含给定区域中至少两个可用区的子网。在 VPC 中创建数据库集群时，您必须选择一个数据库子网组。Amazon DocumentDB 使用该数据库子网组和首选的可用区选择要与集群相关联的子网和该子网中的 IP 地址。如果主实例发生故障，Amazon DocumentDB 可将相应的副本实例提升为新的主实例。然后，它可以使用先前主实例所在子网的 IP 地址创建新的副本实例。

Amazon DocumentDB 在 VPC 中创建实例时，会使用从数据库子网组中选择的 IP 地址将网络接口分配给集群。我们强烈建议您使用 DNS 名称，因为基本 IP 地址在故障转移期间可能会发生变化。有关更多信息，请参阅[Amazon DocumentDB D \(p. 7\)](#)。

有关创建您自己的 VPC 和子网的信息，请参阅[使用 VPC 和子网](#)中的 Amazon Virtual Private Cloud 用户指南。

主题

- [创建 Amazon DocumentDB 子网组 \(p. 303\)](#)
- [描述 Amazon DocumentDB 子网组 \(p. 307\)](#)
- [修改 Amazon DocumentDB 子网组 \(p. 308\)](#)
- [删除 Amazon DocumentDB 子网组 \(p. 310\)](#)

创建 Amazon DocumentDB 子网组

创建 Amazon DocumentDB 集群时，您必须选择一个 Amazon VPC 和该 Amazon VPC 中的对应子网组来启动您的集群。子网组确定可用区中要用于启动实例的可用区和 IP 范围。

子网组是指定的子网集，允许您指定要用于启动 Amazon DocumentDB 实例的可用区。例如，在包含三个实例的集群中，建议在单独的 Amazon 中预置各个实例 — 这样做会针对高可用性进行优化。因此，如果单个可用区出现故障，它只会影响单个实例。

Amazon DocumentDB 实例目前可在多达三个可用区中预置。即使子网组拥有三个以上的子网，您也只能使用这三个子网来创建 Amazon DocumentDB 集群。因此，在创建数据库时，建议仅选择要部署实例的三个子网。

例如：即创建集群，Amazon DocumentDB 选择可用区 {1A、1B 和 1C}。如果您尝试在可用区 {1D} 中创建实例，API 调用将失败。但是，如果您选择创建实例而不指定特定的可用区，Amazon DocumentDB 将代表您选择一个可用区。Amazon DocumentDB 使用某种算法对可用区中的实例进行负载均衡以帮助实现高可用性。默认情况下，如果预置了三个实例，将在三个可用区中预置它们，而不会在单个可用区中预置。

最佳实践

- Unless you have a specific reason, always create a subnet group with three subnets. This ensures that clusters with three or more instances will be able to achieve higher availability as instances will be provisioned across three AZs.

- Always spread instances across multiple AZs to achieve high availability. Never place all instances for a cluster in a single AZ.
- Because failover events can happen at any time, you should not assume that a primary instance or replica instances will always be in a particular AZ.

如何创建子网组

您可以使用 Amazon Web Services Management Console 要么 Amazon CLI 创建 Amazon DocumentDB 子网组。

主题

- [使用 Amazon Web Services Management Console \(p. 304\)](#)
- [使用 Amazon CLI \(p. 305\)](#)

使用 Amazon Web Services Management Console

按照以下步骤创建 Amazon DocumentDB 子网组。

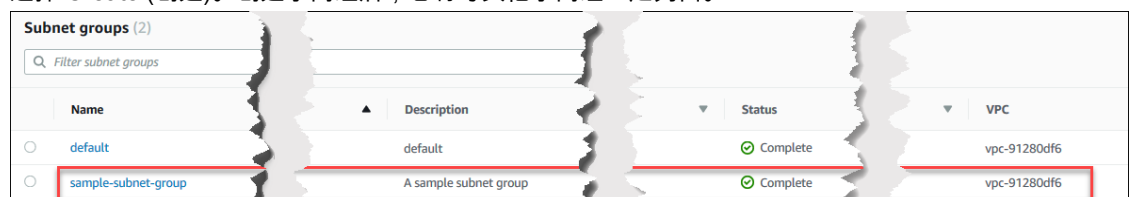
创建 Amazon DocumentDB 子网组

1. 登录 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Subnet groups (子网组)，然后选择 Create (创建)。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在创建子网组页面中：
 - a. 在子网组详细信息部分中：
 - i. 名称为子网组输入有意义的名称。
 - ii. 描述—输入子网组描述。
 - b. 在添加子网部分中：
 - i. VPC— 在列表中，为该子网组选择一个 VPC。
 - ii. 请执行下列操作之一：
 - 要包括所选的 VPC 中的所有子网，请选择添加与此 VPC 相关的所有子网。
 - 要为该子网组指定子网，请为要包含子网的每个可用区执行以下操作。您必须包含至少两个可用区。
 - A. 可用区— 在列表中，选择一个可用区。
 - B. 子网— 在列表中，从为该子网组所选的可用区中选择一个子网。
 - C. 选择 Add subnet (添加子网)。
4. 选择 Create (创建)。创建子网组后，它将与其他子网组一起列出。



Name	Description	Status	VPC
default	default	Complete	vpc-91280df6
sample-subnet-group	A sample subnet group	Complete	vpc-91280df6

使用 Amazon CLI

在使用 Amazon CLI 创建子网组之前，您必须先确定可用的子网。运行以下 Amazon CLI 操作以列出可用区及其子网。

参数：

- **--db-subnet-group**— 可选。如果指定特定的子网组，将列出该组的可用区和子网。如果省略该参数，将列出所有子网组的可用区和子网。如果指定 `default` 子网组，将列出 VPC 的所有子网。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-subnet-groups \
  --db-subnet-group-name default \
  --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

对于 Windows：

```
aws docdb describe-db-subnet-groups ^
  --db-subnet-group-name default ^
  --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[
  [
    "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"
      ]
    ]
  ]
]
```

通过使用上一操作中的输出，您可以创建新的子网组。新子网组必须包含至少两个可用区中的子网。

参数：

- **--db-subnet-group-name** – 必填项。该子网组的名称。
- **--db-subnet-group-description** – 必填项。该子网组的描述。
- **--subnet-ids** – 必填项。要包含在该子网组中的子网的列表。示例：subnet-53ab3636。
- **— 标签 —** 可选。要附加到该子网组的标签（键/值对）的列表。

以下代码创建具有三个子网（subnet-4e26d263、subnet-afc329f4 和 subnet-b3806e8f）的子网组 sample-subnet-group。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "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"  
  }  
}
```

描述 Amazon DocumentDB 子网组

您可以使用Amazon Web Services Management Console或者Amazon CLI以获取 Amazon DocumentDB 子网组的详细信息。

使用 Amazon Web Services Management Console

以下过程介绍如何获取 Amazon DocumentDB 子网组的详细信息。

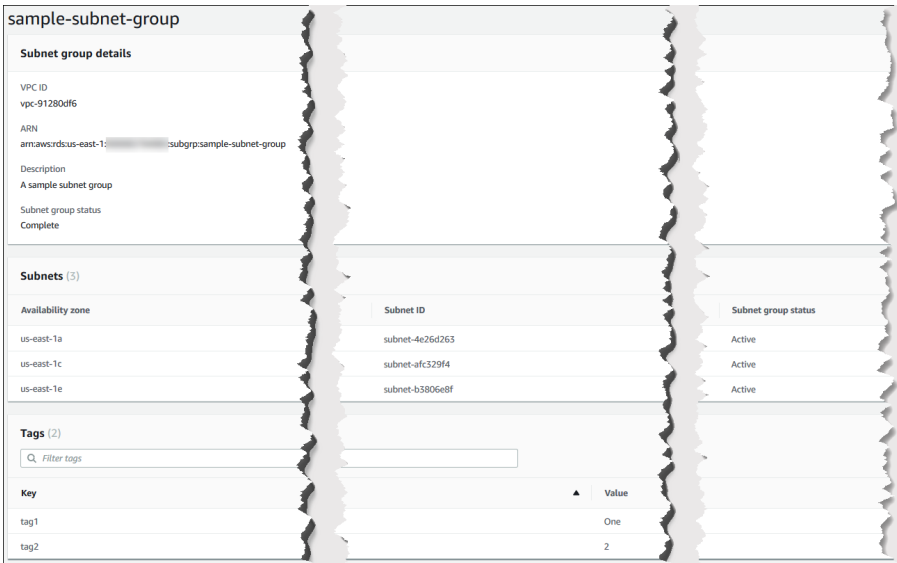
查找子网组的详细信息

1. 登录Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>.
2. 在导航窗格中，选择子网组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 要查看子网组的详细信息，请选择该子网组的名称。



使用 Amazon CLI

要查找 Amazon DocumentDB 子网组的详细信息，请使用describe-db-subnet-groups按照以下参数操作。

参数

- --db-subnet=group-name— 可选。如果包含，则列出指定的子网组的详细信息。如果省略，则列出最多 100 个子网组的详细信息。

Example

以下代码列出了在[创建 Amazon DocumentDB 子网组 \(p. 303\)](#)部分中创建的 `sample-subnet-group` 子网组的详细信息。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-subnet-groups \  
  --db-subnet-group-name sample-subnet-group
```

对于 Windows：

```
aws docdb describe-db-subnet-groups ^  
  --db-subnet-group-name sample-subnet-group
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "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"  
  }  
}
```

修改 Amazon DocumentDB 子网组

您可以使用 Amazon Web Services Management Console 要么 Amazon CLI 修改子网组描述或在 Amazon DocumentDB 子网组中添加或删除子网。不过，您无法修改 `default` 子网组。

主题

- [使用 Amazon Web Services Management Console \(p. 309\)](#)
- [使用 Amazon CLI \(p. 309\)](#)

使用 Amazon Web Services Management Console

您可以使用 Amazon Web Services Management Console 更改子网组描述或添加和删除子网。请记住，在完成后，您必须具有至少两个与子网组关联的可用区。

修改子网组

1. 登录 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择子网组。然后，选择子网组名称左侧的按钮。请记住，您无法修改 default 子网组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 选择 Actions (操作)，然后选择 Modify (修改)。
4. 说明要更改子网组描述，请输入新描述。
5. 要更改与子网组关联的子网，请在添加子网部分中执行一个或多个以下操作：
 - 要从该子网组中删除所有子网，请选择全部删除。
 - 要从该子网组中删除特定的子网，请为要删除的每个子网选择删除。
 - 要添加与该 VPC 关联的所有子网，请选择添加与此 VPC 相关的所有子网。
 - 要将特定子网添加到该子网组中，请为要添加子网的每个可用区执行以下操作。
 - a. 可用区— 在列表中，选择新的可用区。
 - b. 子网— 在列表中，从为该子网组所选的可用区中选择一个子网。
 - c. 选择 Add subnet (添加子网)。
6. 在确认对话框中：
 - 要对子网组进行这些更改，请选择 Modify (修改)。
 - 要将子网组保持不变，请选择取消。

使用 Amazon CLI

您可以使用 Amazon CLI 更改子网组描述或添加和删除子网。请记住，在完成后，您必须具有至少两个与子网组关联的可用区。您无法修改 default 子网组。

参数：

- `--db-subnet-group-name` – 必填项。您正在修改的 Amazon DocumentDB 子网组的名称。
- `--subnet-ids` – 必填项。完成此更改后，子网组中所需的所有子网的列表。

Important

将从子网组中删除当前位于子网组中并且未包含在该列表中的任何子网。如果要保留当前位于子网组中的任何子网，您必须将其包含在该列表中。

- `--db-subnet-group-description` – 可选。子网组的描述。

Example

以下代码修改描述并将现有子网替换为 `subnet-991cb8d0`、`subnet-53ab3636` 和 `subnet-29ab1025` 子网。

对于 Linux、macOS 或 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"
```

对于 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"
```

此操作的输出将类似于下文 (JSON 格式)。请注意,这是在[创建 Amazon DocumentDB 子网组 \(p. 303\)](#)部分中创建的子网组。但是,此子网组中的子网将替换为 modify-db-subnet-group 操作中列出的子网。

```
{  
  "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"  
  }  
}
```

删除 Amazon DocumentDB 子网组

您可以使用 Amazon Web Services Management Console 要么 Amazon CLI 删除 Amazon DocumentDB 子网组。不过,您无法删除 default 子网组。

主题

- [使用 Amazon Web Services Management Console \(p. 311\)](#)
- [使用 Amazon CLI \(p. 311\)](#)

使用 Amazon Web Services Management Console

您可以使用 Amazon Web Services Management Console 删除子网组。但是，您无法删除 default 子网组。

删除子网组

1. 登录 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择子网组。然后，选择子网组名称左侧的按钮。请记住，您无法删除 default 子网组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 选择 Actions，然后选择 Delete。
4. 在确认对话框中：
 - 要删除子网组，请选择删除。
 - 要保留子网组，请选择取消。

使用 Amazon CLI

要删除 Amazon DocumentDB 子网组 Amazon CLI，使用 `delete-db-subnet-group` 按照以下参数操作。

参数

- `--db-subnet-group-name` – 必填项。要删除的 Amazon DocumentDB 子网组的名称。请记住，您无法删除 default 子网组。

Example

以下代码删除 `sample-subnet-group`。

对于 Linux、macOS 或 Unix：

```
aws docdb delete-db-subnet-group \
  --db-subnet-group-name sample-subnet-group
```

对于 Windows：

```
aws docdb delete-db-subnet-group ^
  --db-subnet-group-name sample-subnet-group
```

该操作不会生成任何输出。

Amazon DocumentDB 高可用性和复制

通过使用副本实例，可以在 Amazon DocumentDB 中实现高可用性和读取扩展（具有 MongoDB 兼容性）。单个 Amazon DocumentDB 集群支持单个主实例和最多 15 个副本实例。这些实例可以分布在集群区域的多个可用区中。主实例接受读取和写入流量，而副本实例仅接受读取请求。

集群卷由该集群的多个数据副本组成。但是，集群卷中的数据，对于集群中的主实例和 Amazon DocumentDB 副本表示为单个逻辑卷。副本实例具有最终一致性。它们返回具有最短副本滞后的查询结果 - 通常远远少于主实例写入更新后的 100 毫秒。副本滞后因数据库更改速率而异。也就是说，在对数据库执行大量写入操作期间，您可能发现副本滞后时间变长。

读取扩展

Amazon DocumentDB 副本十分适用于读取扩展，因为它们完全专用于集群卷上的读取操作。写入操作由主实例进行管理。集群卷在集群中的所有实例之间共享。因此，您不必为 Amazon DocumentDB 副本复制和维护数据副本。

高可用性

在创建 Amazon DocumentDB 集群时，根据子网组中的可用区数（必须至少为两个），Amazon DocumentDB 将在这些可用区中预置实例。在集群中创建实例时，Amazon DocumentDB 会自动在子网组的可用区中分发实例来平衡集群。此操作还会阻止所有实例位于同一可用区中。

示例

为了说明这一点，请考虑以下示例：创建一个集群，其中包含一个具有三个可用区的子网组：AZ1、AZ2，和 AZ3。

在创建集群中的第一个实例时，它是主实例并位于其中的一个可用区中。在本示例中，它位于 AZ1 中。创建的第二个实例是副本实例，它位于另外两个可用区之一中，例如，AZ2。创建的第三个实例是副本实例，它位于剩余的可用区 AZ3 中。如果创建更多的实例，它们将分布在多个可用区中，以便在集群中实现平衡。

如果在主实例 (AZ1) 中发生故障，则会触发故障转移，并将其中的一个现有副本提升为主实例。在旧的主实例恢复时，它将成为预置它的同一可用区 (AZ1) 中的副本。在预置三实例集群后，Amazon DocumentDB 继续保留该三实例集群。Amazon DocumentDB 自动处理实例故障检测、故障转移和恢复，而无需进行任何手动干预。

在 Amazon DocumentDB 执行故障转移并恢复实例时，恢复的实例保留在最初预置它的可用区中。但是，此实例的角色可能从主实例变为副本。这样做是为了防止出现以下情况：一系列故障转移可能导致所有实例位于同一可用区中。

您可以将 Amazon DocumentDB 副本指定为故障转移目标。也就是说，如果主实例失败，指定的 Amazon DocumentDB 副本或某个层中的副本将提升为主实例。提升过程只造成短暂的中断，在此期间，对主实例发出的读写请求将失败，并且会出现异常。如果 Amazon DocumentDB 集群不包含任何 Amazon DocumentDB 副本，则在主实例失败后，会重新创建主实例。提升 Amazon DocumentDB 副本要比重新创建主实例快得多。

对于高可用性场景，建议您创建一个或多个 Amazon DocumentDB 副本。这些副本应具有与主实例相同的实例类，并位于 Amazon DocumentDB 集群的不同可用区中。

有关更多信息，请参阅以下内容：

- [了解 Amazon DocumentDB B 集群容错能力 \(p. 282\)](#)
- [Amazon DocumentDB 故障转移 \(p. 313\)](#)
 - [控制故障转移目标 \(p. 313\)](#)

全球群集的高可用性

跨多个实现高可用性 Amazon Web Services 区域，可以设置 [Amazon DocumentDB 全局集群](#)。每个全局集群跨越多个区域，可在中实现低延迟的全局读取以及从停机中进行灾难恢复。Amazon Web Services 区域。Amazon DocumentDB 自动处理从主区域到每个辅助区域的数据复制和更新。

添加 副本

添加到集群的第一个实例是主实例。在第一个实例之后添加的每一个实例都是副本实例。除主实例之外，每个集群最多可拥有 15 个副本实例。

如果您使用 Amazon Web Services Management Console 创建集群，则同时会自动创建主实例。创建集群和主实例的同时还要创建副本，请选择 [Create replica in different zone](#) (在其他区域创建副本)。有关更多信息，请参阅 [创建 Amazon DocumentDB 集群 \(p. 247\)](#) 中的步骤 4.d。要将更多副本添加到 Amazon DocumentDB 集群，请参阅 [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)。

在使用 Amazon CLI 创建集群时，必须明确地创建主实例和副本实例。有关更多信息，请参阅以下主题的“使用 Amazon CLI”部分：

- [创建 Amazon DocumentDB 集群 \(p. 247\)](#)
- [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)

Amazon DocumentDB 故障转移

在某些情况下，例如某些类型的计划维护，或出现不太可能出现的主节点或可用区故障时，Amazon DocumentDB (具有 MongoDB 兼容性) 将检测故障并替换主节点。在故障转移期间，写入停机时间将最小化。因为主节点的角色故障转移到一个只读副本，而不是必须创建和设置新的主节点。此故障检测和副本提升可确保在提升完成后，您可以马上继续写入新的主节点。

要使故障转移起作用，您的集群必须至少具有两个实例 — 一个主实例和至少一个副本实例。

控制故障转移目标

Amazon DocumentDB 为您提供故障转移层作为方法来控制在发生故障转移时，将哪个副本实例提升为主实例。

故障转移层

每个副本实例都与一个故障转移层 (0—15) 关联。当因为维护或意外硬件故障而发生故障转移时，主实例将故障转移到具有最高优先级的副本 (编号最低层)。如果多个副本具有相同的优先级层，则主实例将故障转移到大小与之前主实例最接近的层的副本。

通过将一组选择副本的故障转移层设置为 0 (最高优先级)，可以确保故障转移将提升该组中的副本之一。您可以通过为特定副本分配低优先级层 (编号高) 来有效阻止这些副本在发生故障转移时被提升为主实例。当特定副本被应用程序大量使用并且故障转移到其中一个副本会对关键应用程序产生不利影响时，此方法很有用。

您可以在创建实例时或稍后通过修改实例来设置它的故障转移层。通过修改实例来设置实例故障转移不会触发故障转移。有关更多信息，请参阅以下主题：

- [将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)
- [修改 Amazon DocumentDB 实例 \(p. 295\)](#)

手动启动故障转移时，可通过两种方法来控制将哪些副本实例提升为主实例：前面介绍的故障转移层和 `--target-db-instance-identifier` 参数。

`--target-db-instance-identifier`

为了进行测试，您可以使用 `failover-db-cluster` 操作强制故障转移事件。您可以使用 `--target-db-instance-identifier` 参数指定将哪个副本实例提升为主实例。使用 `--target-db-instance-`

`identifier` 参数取代故障转移优先级层。如果您不指定 `--target-db-instance-identifier` 参数，主故障转移将与故障转移优先级层保持一致。

故障转移过程中发生的事件

Amazon DocumentDB 将自动处理故障转移，以便应用程序尽快恢复数据库操作而无需管理干预。

- 如果故障转移时，如果您在相同或不同的可用区中有 Amazon DocumentDB 副本实例：Amazon DocumentDB 会翻转您的实例的别名记录 (CNAME)，以指向运行状态正常的副本；相应地，此副本会晋升为新的主实例。从开始到结束，故障转移通常会在 30 秒内完成。
- 如果您没有 Amazon DocumentDB 副本实例（例如，单个实例集群）：Amazon DocumentDB 将尝试在与原始实例相同的可用区中创建新实例。原实例会尽量替换，但可能不会成功，例如出现全面影响该可用区的问题时。

您的应用程序应在连接丢失时重试数据库连接。

测试故障转移

集群的故障转移会将集群中的 Amazon DocumentDB 副本之一（只读实例）提升为主实例（集群写入器）。

当主实例发生故障时，Amazon DocumentDB 将自动故障转移到 Amazon DocumentDB 副本（如果存在）。当您要模拟主实例的故障以进行测试时，可以强制进行故障转移。集群中的每个实例都有自己的终端节点地址。因此，您需要在故障转移完成后清理并重新建立使用这些终端节点地址的任何现有连接。

要强制进行故障转移，请使用带以下参数的 `failover-db-cluster` 操作。

- `--db-cluster-identifier` — 必填项。要进行故障转移的集群的名称。
- `--target-db-instance-identifier` — 可选。要提升为主实例的实例的名称。

Example

以下操作将强制对 `sample-cluster` 集群进行故障转移。它不指定哪个实例要成为新的主实例，因此 Amazon DocumentDB 将根据故障转移层优先级来选择实例。

对于 Linux、macOS 或 Unix：

```
aws docdb failover-db-cluster \  
  --db-cluster-identifier sample-cluster
```

对于 Windows：

```
aws docdb failover-db-cluster ^  
  --db-cluster-identifier sample-cluster
```

以下操作将强制 `sample-cluster` 集群进行故障转移，同时指定将 `sample-cluster-instance` 提升为主角色。（请注意输出中的 `"IsClusterWriter": true`。）

对于 Linux、macOS 或 Unix：

```
aws docdb failover-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --target-db-instance-identifier sample-cluster-instance
```

对于 Windows :

```
aws docdb failover-db-cluster ^
--db-cluster-identifier sample-cluster ^
--target-db-instance-identifier sample-cluster-instance
```

此操作的输出将类似于下文 (JSON 格式) 。

```
{
  "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-ABCDEFGHIJKLMNPOQRSTUVWXYZ"  
  }  
}
```

复制滞后

复制滞后通常为 50 毫秒或更短。副本滞后增加的最常见原因有：

- 主节点的写入率较高，导致只读副本落后于主节点。
- 长时间运行的查询（例如，大型顺序扫描、聚合查询）和传入写复制之间的只读副本的争用。
- 对只读副本进行大量的并发查询。

为了尽量减少复制滞后，请尝试以下故障排除

- 如果您的写入率较高或 CPU 使用率较高，我们建议您扩展集群中的实例。
- 如果只读副本上存在长时间运行的查询，并且对所查询的文档进行频繁更新，请考虑更改长时间运行的查询，或者针对主/写副本运行这些查询，以避免对只读副本发生争用。
- 如果只读副本上存在大量并发查询或 CPU 使用率较高，则另一种选择是扩大只读副本的数量以分散工作负载。
- 由于复制滞后是高写吞吐量和长时间运行的查询的结果，因此我们建议使用 dbClusterReplicaLagMasterWriteThroughputWriteIOPS 指标结合慢查询记录器和 WriteThroughput/WriteIOPS 指标。

一般来说，我们建议您的所有副本都具有相同的实例类型，以便群集故障转移不会导致性能下降。

如果您在扩展和向外扩展之间进行选择（例如 6 个较小的实例与三个较大的实例），我们通常建议在扩展之前尝试首先向上扩展（较大的实例），因为每个数据库实例将获得更大的缓冲缓存。

主动地，您应该设置复制滞后警报并将其阈值设置为一个值，该值是副本实例上的数据在开始影响应用程序功能之前落后多远（或“过时”）的上限。一般来说，由于工作负载临时，我们建议在发出警报之前超过几个数据点的复制延迟阈值。

Note

此外，我们建议您为超过 10 秒的复制滞后设置另一个警报。如果您超过了多个数据点的此阈值，我们建议您扩展实例或降低主实例上的写入吞吐量。

管理 Amazon DocumentDB 事件

Amazon DocumentDB（兼容 MongoDB）保留与集群、实例、快照、安全组和集群参数组相关的事件记录。此信息包含事件的日期和时间、事件的源名称和源类型以及一条与事件相关的消息。

Important

对于某些管理功能，Amazon DocumentDB 使用与 Amazon RDS 和 Amazon Neptune 共享的操作技术。区域限制由区域级别管理，在 Amazon DocumentDB、Amazon RDS 和 Amazon Neptune 之间共享。有关更多信息，请参阅 [区域配额](#) (p. 497)。

主题

- [查看 Amazon DocumentDB 活动类别](#) (p. 317)
- [查看 Amazon DocumentDB 事件](#) (p. 318)

查看 Amazon DocumentDB 活动类别

每种 Amazon DocumentDB 资源类型与特定类型的事件关联。您可以使用 Amazon CLI `describe-event-categories` 操作，查看事件类型和 Amazon DocumentDB 资源类型之间的映射。

参数

- **--source-type**— 可选。可以使用 `--source-type` 参数查看特定源类型的事件类别。以下是允许的值：
 - `db-cluster`
 - `db-instance`
 - `db-parameter-group`
 - `db-security-group`
 - `db-cluster-snapshot`
- **--filters**— 可选。要仅查看 Amazon DocumentDB 的事件类别，请使用筛选器 `--filter Name=engine,Values=docdb`。

Example

以下代码列出了与集群关联的事件类别。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-event-categories \
  --filter Name=engine,Values=docdb \
  --source-type db-cluster
```

对于 Windows：

```
aws docdb describe-event-categories ^
  --filter Name=engine,Values=docdb ^
  --source-type db-cluster
```

此操作的输出将类似于下文（JSON 格式）。

```
{
  "EventCategoriesMapList": [
    {
      "EventCategories": [
        "notification",
        "failure",
        "maintenance",
        "failover"
      ],
      "SourceType": "db-cluster"
    }
  ]
}
```

以下代码列出了与每种 Amazon DocumentDB 源类型关联的事件类别。

```
aws docdb describe-event-categories
```

此操作的输出将类似于下文（JSON 格式）。

```
{
  "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"
      ]
    }
  ]
}
```

查看 Amazon DocumentDB 事件

您可通过 Amazon DocumentDB 控制台（显示过去 24 小时的事件）检索 Amazon DocumentDB 资源的事件。您还可以通过使用[describe-events](#) Amazon CLI 命令，或[DescribeEvents](#) Amazon DocumentDB API 操作。如果您将 Amazon CLI 或者使用 Amazon DocumentDB API 查看事件，则可检索过去长达 14 天内的事件。

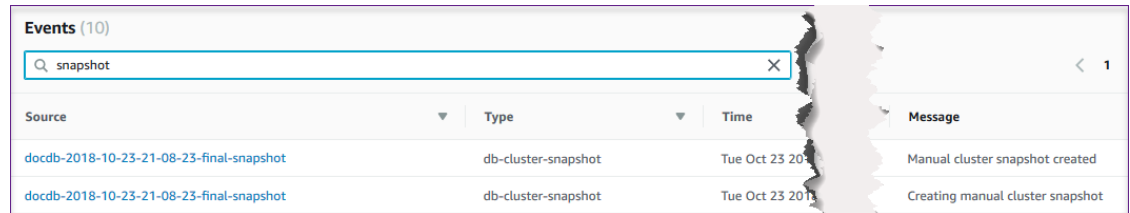
主题

- [使用 Amazon Web Services Management Console \(p. 319\)](#)
- [使用 Amazon CLI \(p. 319\)](#)

使用 Amazon Web Services Management Console

查看过去 24 小时的所有 Amazon DocumentDB 实例事件

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Events (事件)。列表中显示可用的事件。
3. 使用 Filter (筛选条件) 列表按类型筛选事件。在文本框中输入期限以进一步筛选结果。例如，以下屏幕截图显示筛选的所有 Amazon DocumentDB 事件快照事件。



The screenshot shows the Amazon DocumentDB Events console. At the top, there's a search bar with 'snapshot' entered. Below it, a table lists events. The table has columns for Source, Type, Time, and Message. Two events are visible, both of type 'db-cluster-snapshot'.

Source	Type	Time	Message
docdb-2018-10-23-21-08-23-final-snapshot	db-cluster-snapshot	Tue Oct 23 2018	Manual cluster snapshot created
docdb-2018-10-23-21-08-23-final-snapshot	db-cluster-snapshot	Tue Oct 23 2018	Creating manual cluster snapshot

使用 Amazon CLI

查看过去 7 天的所有 Amazon DocumentDB 实例事件

您可通过运行 [describe-events](#) Amazon CLI 使用 `--duration` 设置为的参数 10080 (10,080 分钟)。

```
aws docdb describe-events --duration 10080
```

筛选 Amazon DocumentDB 事件

要查看特定的 Amazon DocumentDB 事件，请使用 `describe-events` 按照以下参数操作。

参数

- `--filter`— 必需，用于限制 Amazon DocumentDB 事件的返回值。使用 `Name=engine,Values=docdb` 以仅筛选 Amazon DocumentDB 的所有事件。
- `--source-identifier`— 可选。为其返回事件的事件源的标识符。如果省略，则在结果中包含来自所有源的事件。
- `--source-type`— 可选，除非 `--source-identifier` 已提供，则为必需。如果提供 `--source-identifier`，则 `--source-type` 必须与 `--source-identifier` 的类型一致。以下是允许的值：
 - `db-cluster`
 - `db-instance`
 - `db-parameter-group`
 - `db-security-group`
 - `db-cluster-snapshot`

以下示例列出您的所有 Amazon DocumentDB 事件。

```
aws docdb describe-events --filters Name=engine,Values=docdb
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
```



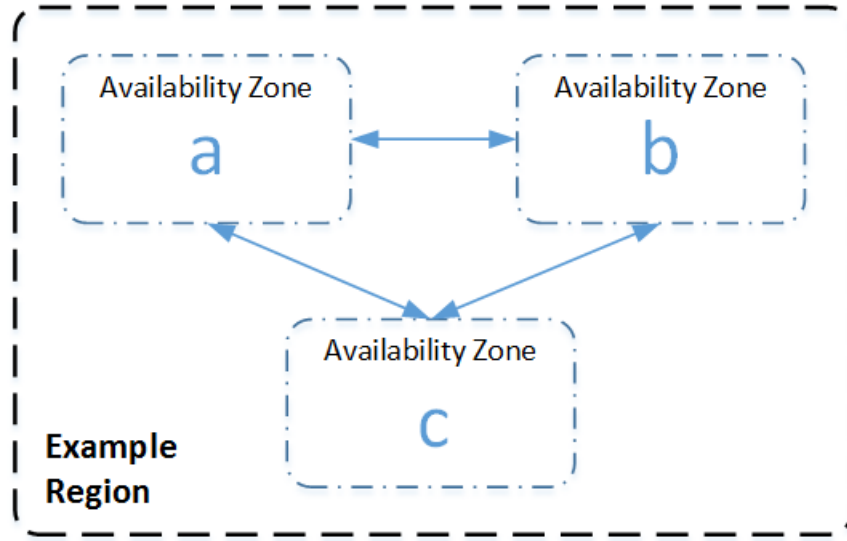
```
"Events": [
  {
    "SourceArn": "arn:aws:rds:us-east-1:123SAMPLE012:db:sample-cluster-instance3",
    "Message": "instance created",
    "SourceType": "db-instance",
    "Date": "2018-12-11T21:17:40.023Z",
    "SourceIdentifier": "sample-cluster-instance3",
    "EventCategories": [
      "creation"
    ]
  },
  {
    "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"
    ]
  }
]
```

有关更多信息，请参阅[审计Amazon DocumentDB \(p. 185\)](#)。

选择区域和可用区

Amazon 云计算资源在全球多个位置托管。这些地点包括Amazon Web Services 区域可用区。EACESAmazon Web Services 区域是一个单独的地理区域。每个区域都有多个相互隔离的位置，称为可用区。借助 Amazon DocumentDB，您可以在多个位置放置资源（例如实例和数据）。资源不会跨复制 Amazon Web Services 区域除非你特意这样做。

Amazon 运行着具有高可用性的高级数据中心。数据中心有时会发生影响托管于同一位置的所有实例的可用性的故障，虽然这种故障极少发生。如果您将所有实例都托管在受此类故障影响的同一个位置，则您的所有实例都将不可用。下图显示了Amazon Web Services 区域有三个可用区。



需要牢记的是，每个区域都是独立的，这一点非常重要。您启动的任何 Amazon DocumentDB 活动（如创建实例或列出可用实例）都只会在您当前的默认情况下运行。Amazon Web Services 区域。您可以通过设置 `EC2_REGION` 环境变量在控制台上更改默认区域。或者，通过使用 Amazon CLI 中的 `--region` 参数进行覆盖。有关更多信息，请参阅 [配置 Amazon Command Line Interface](#) 特别是有关环境变量和命令行选项的章节。

当您使用 Amazon DocumentDB 控制台创建集群并选择在不同的可用区域中创建副本时，Amazon DocumentDB 会创建两个实例。它在一个可用区中创建主实例，在另一个可用区中创建副本实例。集群始终跨三个可用区复制。

要在特定的特定情况下创建或使用 Amazon DocumentDB 实例 Amazon Web Services 区域，使用相应的区域服务终端节点。

区域可用性

Amazon DocumentDB 可在以下内容中找到 Amazon 地区。

Amazon DocumentDB 支持的区域

区域名称	区域	可用区（计算）
美国东部（俄亥俄）	us-east-2	3
美国东部（弗吉尼亚北部）	us-east-1	6
美国西部（俄勒冈）	us-west-2	4
南美洲（圣保罗）	sa-east-1	3
亚太地区（孟买）	ap-south-1	3
亚太地区（首尔）	ap-northeast-2	4
亚太地区（新加坡）	ap-southeast-1	3
亚太地区（悉尼）	ap-southeast-2	3

区域名称	区域	可用区 (计算)
亚太地区 (东京)	ap-northeast-1	3
加拿大 (中部)	ca-central-1	3
中国 (北京) 区域	cn-north-1	3
中国 (宁夏)	cn-northwest-1	3
欧洲 (法兰克福)	eu-central-1	3
欧洲 (爱尔兰)	eu-west-1	3
欧洲 (伦敦)	eu-west-2	3
欧洲 (米兰)	eu-south-1	3
欧洲 (巴黎)	eu-west-3	3
Amazon GovCloud (US)	us-gov-west-1	3

默认情况下，Amazon DocumentDB 集群的时区是协调世界时 (UTC)。

有关查找特定区域中集群和实例的连接终端节点的信息，请参阅[了解 Amazon DocumentDB 终端节点 \(p. 343\)](#)。

管理 Amazon DocumentDB 集群参数组

您可以通过使用集群参数组中的参数来管理 Amazon DocumentDB 引擎配置。一个集群参数组是 Amazon DocumentDB 配置值的集合，它使管理 Amazon DocumentDB 集群的参数变得更加容易。集群参数组就像是引擎配置值的容器，这些值可应用于集群中的所有实例。

本节介绍如何创建、查看和修改集群参数组。它还显示如何确定与给定集群关联的集群参数组。

主题

- [描述 Amazon DocumentDB 集群参数组 \(p. 322\)](#)
- [创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#)
- [修改 Amazon DocumentDB 集群参数组 \(p. 328\)](#)
- [修改 Amazon DocumentDB 集群以使用自定义集群参数组 \(p. 331\)](#)
- [复制 Amazon DocumentDB 集群参数组 \(p. 332\)](#)
- [重置 Amazon DocumentDB 集群参数组 \(p. 334\)](#)
- [删除 Amazon DocumentDB 集群参数组 \(p. 335\)](#)
- [Amazon DocumentDB 集群参数参考 \(p. 337\)](#)

描述 Amazon DocumentDB 集群参数组

当您创建 Amazon DocumentDB 集群时，default.docdb4.0 系统会为该集群自动创建集群参数组。您可以查看此默认集群参数组的详细信息，或者您拥有的任何其他集群参数组的详细信息。您还可以确定与特定集群关联的参数组。

主题

- [描述 Amazon DocumentDB 集群参数组的详细信息 \(p. 323\)](#)
- [确定 Amazon DocumentDB 集群的参数组 \(p. 325\)](#)

描述 Amazon DocumentDB 集群参数组的详细信息

要描述给定集群参数组的详细信息，请使用完成以下步骤：Amazon Web Services Management Console 或者 Amazon Command Line Interface (Amazon CLI)。

使用 Amazon Web Services Management Console

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择参数组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在集群参数组窗格中，选择要查看其详细信息的参数组的名称。
4. 生成的页面上会显示参数组的参数、最近的活动和标签。
 - 在集群参数，您可以查看参数的名称、当前值、允许值、参数是否可修改、应用类型、数据类型和描述。您可以通过选择参数然后选择来修改单个参数编辑中的集群参数部分。有关更多信息，请参阅 [修改 Amazon DocumentDB 集群参数 \(p. 339\)](#)。
 - 在近期事件，您可以看到此参数组的最近事件。您可以使用本节中的搜索栏筛选这些事件。有关更多信息，请参阅 [管理 Amazon DocumentDB 事件 \(p. 316\)](#)。
 - 在 Tags (标签) 下，您可以看到此集群参数组上的标签。您可以通过选择添加或删除标签编辑中的标签部分。有关更多信息，请参阅 [为 Amazon DocumentDB 资源添加标签 \(p. 352\)](#)。

使用 Amazon CLI

您可以使用 `describe-db-cluster-parameter-groups` Amazon CLI 命令可查看 Amazon DocumentDB 的单个群集参数组或所有群集参数组的 Amazon 资源名称 (ARN)、系列、描述和名称。您也可以使用 `describe-db-cluster-parameters` Amazon CLI 命令来查看单个群集参数组中的参数及其详细信息。

- **`--describe-db-cluster-parameter-groups`**— 查看您的所有集群参数组及其详细信息的列表。
 - **`--db-cluster-parameter-group-name`** 可选。您想描述的集群参数组的名称。如果忽略此参数，则将描述所有集群参数组。
- **`--describe-db-cluster-parameters`**— 列出参数组中的所有参数及其值。
 - **`--db-cluster-parameter-group name`** – 必需。您想描述的集群参数组的名称。

Example

以下代码列出了最多 100 个集群参数组及其 ARN、系列、描述和名称。

```
aws docdb describe-db-cluster-parameter-groups
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
```

```
    "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

以下代码列出了的 ARN、系列、描述和名称sample-parameter-group.

对于 Linux、macOS 或 Unix :

```
aws docdb describe-db-cluster-parameter-groups \
  --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows :

```
aws docdb describe-db-cluster-parameter-groups ^
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "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

以下代码列出了中的参数值。sample-parameter-group.

对于 Linux、macOS 或 Unix :

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows :

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文 (JSON 格式)。

```
{  
  "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"  
    }  
  ]  
}
```

确定 Amazon DocumentDB 集群的参数组

要确定与特定集群关联的参数组，请使用Amazon Web Services Management Console或者Amazon CLI。

使用 Amazon Web Services Management Console

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在左侧导航窗格中，选择集群。
3. 在集群列表中，选择感兴趣的集群的名称。
4. 生成的页面将显示所选集群的详细信息。向下滚动到 Cluster details (集群详细信息)。在此部分的底部，在 Cluster parameter group (集群参数组) 的下方找到参数组的名称。

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 ³²⁶

sample-parameter-group

使用 Amazon CLI

以下 Amazon CLI 代码确定管理集群 `sample-cluster` 的参数组。

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

此操作的输出将类似于下文 (JSON 格式)。

```
[
  [
    "sample-cluster",
    "sample-parameter-group"
  ]
]
```

创建 Amazon DocumentDB 集群参数组

Amazon DocumentDB 提供 3.6 和 4.0 版本控制的集群参数组。您可以根据 `docdb4.0` 命名的系列 `default.docdb4.0`。在任何一个版本中，您都无法修改 `default.docdb4.0` 要么 `default.docdb3.6` 直接集群参数组。要自定义集群参数组中的参数并将其与 Amazon DocumentDB 集群关联，您必须首先创建一个新的非默认集群参数组。然后，您可以修改该新的自定义参数组中的参数，然后将其与 Amazon DocumentDB 集群关联。

以下过程引导您通过基于创建自定义集群参数组。 `docdb4.0` 系列。然后您可以 [修改该参数组中的参数](#)。

Note

创建集群参数组之后，您应至少等待 5 分钟，再使用此特定的参数组。这允许 Amazon DocumentDB 完全完成 `create` 在将集群参数组用作新数据库集群的默认设置之前的操作。您可以使用 Amazon Web Services Management Console 或者 `describe-db-cluster-parameter-groups` Amazon CLI 操作，验证是否已创建集群参数组。有关更多信息，请参阅 [描述 Amazon DocumentDB 集群参数组 \(p. 322\)](#)。

使用 Amazon Web Services Management Console

要创建集群参数组

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择参数组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在集群参数组窗格中，选择 `Create`。
4. 在创建集群参数组窗格中，输入以下内容：
 - a. **Group name**— 输入集群参数组的名称。例如 `sample-parameter-group`。集群参数组具有以下命名限制：
 - 长度为 [1—255] 个字母数字字符。
 - 第一个字符必须是字母。
 - 不能以连字符结束或包含两个连续连字符。
 - b. **说明**— 为此集群参数组提供描述。

5. 要创建集群参数组，请选择创建。要取消操作，请选择取消。
6. 选择之后Create，将在页面顶部显示以下文本，以验证是否已成功创建集群参数组：

```
Successfully created cluster parameter group 'sample-parameter-group'.
```

使用 Amazon CLI

要为 Amazon DocumentDB 4.0 集群创建新的集群参数组，请使用Amazon CLI `create-db-cluster-parameter-group`按照以下参数进行操作：

- **--db-cluster-parameter-group-name**— 自定义集群参数组的名称。例如 `sample-parameter-group`。
- **--db-cluster-parameter-group-family**— 用作自定义集群参数组模板的集群参数组系列。目前，必须为 `docdb4.0`。
- **--description**— 用户提供的此集群参数组的描述。下面的示例使用了“Custom docdb4.0 parameter group”。

对于 Linux、macOS 或 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"
```

对于 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"
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "sample-parameter-group",  
    "DBParameterGroupFamily": "docdb4.0",  
    "Description": "Custom docdb4.0 parameter group",  
    "DBClusterParameterGroupArn": "sample-parameter-group-arn"  
  }  
}
```

修改 Amazon DocumentDB 集群参数组

本节介绍如何修改自定义Amazon DocumentDB 参数组。在 Amazon DocumentDB 中，您不能直接修改`default.docdb4.0`集群参数组。如果您的 Amazon DocumentDB 集群正在使用默认集群参数组并且您想修改其中的值，则必须首先[创建新的参数组](#)要么[复制现有参数组](#)请修改它，然后将修改后的参数组应用于您的集群。

要修改非默认集群参数组，请完成以下步骤：修改集群参数组之后，您应至少等待 5 分钟，再使用。这允许 Amazon DocumentDB 完全完成`modify`在使用集群参数组之前的操作。您可以使用Amazon Web Services Management Console或者Amazon CLI `describe-db-cluster-parameters`操作，验证您的集群参数组已修改。有关更多信息，请参阅[描述集群参数组](#) (p. 322)。

使用 Amazon Web Services Management Console

按照以下步骤修改自定义 Amazon DocumentDB 参数组。您无法修改 default 参数组。如果要在 default 参数组，您可以复制默认的集群参数组请修改它，然后将修改后的参数组应用于您的集群。有关将参数组应用于集群的更多信息，请参阅 [修改 Amazon DocumentDB 集群 \(p. 259\)](#)。

修改自定义集群参数组

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在控制台左侧的导航窗格中，选择参数组。在参数组列表中，选择要修改的参数组的名称。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 对于要修改的参数组中的每个参数，执行以下操作：
 - a. 找到要修改的参数，然后通过检查它是否被列为来验证它是否可修改。true 在可修改 column。
 - b. 如果它是可修改的，请选择参数然后选择编辑从控制台页面的右上角。
 - c. 在修改 **<parameter-name>** 对话框中，进行所需的更改。然后，选择 Modify cluster parameter (修改集群参数) 或选择 Cancel (取消) 以放弃更改。

使用 Amazon CLI

您可以修改 ParameterValue、Description，或者 ApplyMethod 在自定义 Amazon DocumentDB 集群参数组中，使用 Amazon CLI。您无法直接修改默认集群参数组。

要修改自定义集群参数组的参数，请使用 modify-db-cluster-parameter-group 按照以下参数进行操作。

- **--db-cluster-parameter-group-name** – 必需。您正在修改的集群参数组的名称。
- **--parameters** – 必需。您正在修改的参数。有关适用于 Amazon DocumentDB 集群中的所有实例的参数的列表，请参阅 [Amazon DocumentDB 集群参数参考 \(p. 337\)](#)。每个参数条目必须包含以下内容：
 - **ParameterName**— 您正在修改的参数的名称。
 - **ParameterValue**— 此参数的新值。
 - **ApplyMethod**— 您希望如何对应用的参数进行更改。允许的值为 immediate 和 pending-reboot。

Note

带 static 的 ApplyType 参数必须具有 pending-reboot 的 ApplyMethod。

Example -修改参数的值

在此示例中，您列出的参数值 sample-parameter-group 并修改 tls 参数。然后，等待 5 分钟后，再次列出 sample-parameter-group 的参数值以查看更改后的参数值。

1. 列出的参数及其值 sample-parameter-group。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
    --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文 (JSON 格式)。

```
{  
  "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. 修改tls参数，以便其值为disabled。

您无法修改ApplyMethod由于ApplyType是static。

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --parameters  
  "ParameterName=tls,\"ParameterValue\"=disabled,\"ApplyMethod\"=pending-reboot
```

对于 Windows：

```
aws docdb modify-db-cluster-parameter-group ^  
  --db-cluster-parameter-group-name sample-parameter-group ^  
  --parameters  
  "ParameterName=tls,\"ParameterValue\"=disabled,\"ApplyMethod\"=pending-reboot
```

此操作的输出将类似于下文 (JSON 格式)。

```
{  
  "DBClusterParameterGroupName": "sample-parameter-group"  
}
```

3. 等至少 5 分钟。
4. 列出的参数值sample-parameter-group确认tls参数已修改。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows：

```
aws docdb describe-db-cluster-parameters ^
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文（JSON 格式）。

```
{
  "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"
    }
  ]
}
```

修改 Amazon DocumentDB 集群以使用自定义集群参数组

当您创建 Amazon DocumentDB 集群时，default.docdb4.0 将为该集群自动创建参数组。您无法修改 default 集群参数组。相反，您可以修改 Amazon DocumentDB 集群以将新的自定义参数组与其关联。

本节介绍如何修改现有 Amazon DocumentDB 集群以使用自定义集群参数组 Amazon Web Services Management Console 和 Amazon Command Line Interface (Amazon CLI)。

使用 Amazon Web Services Management Console

修改 Amazon DocumentDB 集群以使用新的非默认集群参数组。

1. 在开始之前，请确保您已创建 Amazon DocumentDB 集群和集群参数组。请参阅 [创建 Amazon DocumentDB 集群 \(p. 247\)](#) 和 [创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#) 以获取进一步说明。
2. 创建集群参数组后，打开 Amazon DocumentDB 控制台 <https://console.amazonaws.cn/docdb>。在导航窗格中，选择集群以将新参数组添加到集群。

3. 选择要与您的参数组关联的集群。选择操作选择，然后选择修改以修改您的集群。
4. 在Cluster options (集群选项)中，选择要与集群关联的新参数组。
5. 选择 Continue (继续) 以查看修改摘要。
6. 在确认您的更改后，您可以立即应用这些更改，也可以在 Scheduling of modifications (修改计划) 下的下一个维护时段内应用这些更改。
7. 选择 Modify cluster (修改集群) 以使用新参数组更新您的集群。

使用 Amazon CLI

在开始之前，请确保您已创建 Amazon DocumentDB 集群和集群参数组。您可以[创建 Amazon DocumentDB 集群](#)使用Amazon CLI `create-db-cluster`operation. 您可以[创建集群参数组](#)使用Amazon CLI `create-db-cluster-parameter-group`operation.

要将新数据库集群参数组添加到您的集群，请使用Amazon CLI `modify-db-cluster`按照以下参数进行操作。

- `--db-cluster-identifier`— 集群的名称 (例如, `sample-cluster`)。
- `--db-cluster-parameters-group-name`— 您希望与集群关联的参数组的名称 (例如, `sample-parameter-group`)。

Example

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文 (JSON 格式)。

```
"DBCluster": {  
  "AvailabilityZones": [  
    "us-west-2c",  
    "us-west-2b",  
    "us-west-2a"  
  ],  
  "BackupRetentionPeriod": 1,  
  "DBClusterIdentifier": "sample-cluster",  
  "DBClusterParameterGroup": "sample-parameter-group",  
  "DBSubnetGroup": "default",  
  ...  
}
```

复制 Amazon DocumentDB 集群参数组

您可以使用在 Amazon DocumentDB 中创建集群参数组副本。Amazon Web Services Management Console 或者Amazon Command Line Interface(Amazon CLI)。

使用 Amazon Web Services Management Console

以下过程通过创建现有集群参数组副本，引导您通过创建新集群参数组。

复制集群参数组

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>.
2. 在导航窗格中，选择参数组。

3. 在集群参数组窗格中，选择要复制的集群参数组的名称。
4. 选择操作选择，然后选择Copy来复制该参数组。
5. 在复制选项，输入新数据库集群参数组的名称和描述。然后选择 。Copy保存您的更改。

使用 Amazon CLI

要创建集群参数组副本，请使用copy-db-cluster-parameter-group按照以下参数进行操作。

- **--source-db-cluster-parameter-group-identifier** – 必需。您希望创建副本的集群参数组的名称或 Amazon 资源名称 (ARN)。

如果源群集和目标群集参数组位于相同Amazon Web Services 区域，标识符可以是名称，也可以是 ARN。

如果源群集和目标群集参数组位于不同Amazon Web Services 区域，标识符必须是 ARN。

- **--target-db-cluster-parameter-group-identifier** – 必需。集群参数组副本的名称或 ARN。

约束：

- 不能为 null 或空。
 - 必须包含 1-255 个字母、数字或连字符。
 - 第一个字符必须是字母。
 - 不能以连字符结束或包含两个连续连字符。
- **--target-db-cluster-parameter-group-description** – 必需。用户提供的集群参数组副本的描述。

Example

以下代码将创建 sample-parameter-group 的副本，将副本命名为 sample-parameter-group-copy。

对于 Linux、macOS 或 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"
```

对于 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"
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "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"
  }
}
```

重置 Amazon DocumentDB 集群参数组

您可以通过使用将部分或全部 Amazon DocumentDB 集群参数组的值重置为其默认值。Amazon Web Services Management Console 或者 Amazon Command Line Interface (Amazon CLI) 以重置集群参数组。

使用 Amazon Web Services Management Console

要将部分或全部集群参数组的值重置为默认值，请按照以下步骤操作。

要重置集群参数组的参数值

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在控制台左侧的导航窗格中，选择参数组。
3. 在集群参数组窗格中，选择要重置的集群参数组的名称。
4. 选择操作选择，然后选择 Reset 以重置该参数组。
5. 在结果上集群参数组重置确认页面中，确认您要将该参数组的所有集群参数重置为默认值。然后选择。Reset 以重置你的参数组。您还可以选择 Cancel 放弃更改。

使用 Amazon CLI

要将部分或全部集群参数组的参数值重置为其默认值，请使用带有以下参数的 `reset-db-cluster-parameter-group` 操作。

- **--db-cluster-parameter-group-name** – 必需。要重置的集群参数组的名称。
- **--parameters** 可选。集群参数组中要重置为其默认值的 ParameterName 和 ApplyMethod 列表。必须将静态参数设置为 `pending-reboot` 在下一个实例重启时生效或 `reboot-db-instance` 请求。你必须打电话给 `reboot-db-instance` 适用于集群中您希望将更新的静态参数应用到的每个实例。

此参数和 `--reset-all-parameters` 相互排斥：您可以使用这两个参数之一，但不能同时使用二者。

- **--reset-all-parameters** 要么 **--no-reset-all-parameters** 可选。指定是否将所有参数 (`--reset-all-parameters`) 或仅部分参数 (`--no-reset-all-parameters`) 重置为其默认值。`--reset-all-parameters` 参数和 `--parameters` 相互排斥：您可以使用这两个参数之一，但不能同时使用二者。

当您重置整个组时，动态参数立即更新。静态参数设置为 `pending-reboot` 在下一个实例重启时生效或 `reboot-db-instance` 请求。对于集群中您希望更新的静态参数应用于的每个实例，您必须调用 `reboot-db-instance`。

Example

示例 1：将所有参数重置为其默认值

以下代码重置集群参数组中的所有参数。sample-parameter-group 他们的默认值。

对于 Linux、macOS 或 Unix：

```
aws docdb reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --reset-all-parameters
```

对于 Windows：

```
aws docdb reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
```



```
--reset-all-parameters
```

示例 2：将指定的参数重置为其默认值

以下代码重置tls集群参数组中的参数sample-parameter-group设置为默认值。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出将类似于下文（JSON 格式）。

```
{
    "DBClusterParameterGroupName": "sample-parameter-group"
}
```

重新引导集群实例

在更改静态参数的值之前，必须重启集群实例。重启集群中您希望将更新的静态参数应用到的每个实例。

对于 Linux、macOS 或 Unix：

```
aws docdb reboot-db-instance \
    --db-instance-identifier sample-cluster-instance
```

对于 Windows：

```
aws docdb reboot-db-instance ^
    --db-instance-identifier sample-cluster-instance
```

删除 Amazon DocumentDB 集群参数组

您可以使用删除自定义 Amazon DocumentDB 集群参数组。Amazon Web Services Management Console 或者 Amazon Command Line Interface (Amazon CLI)。您无法删除 default.docdb4.0 集群参数组。

使用 Amazon Web Services Management Console

删除集群参数组

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择参数组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在参数组窗格中，选择要删除的集群参数组左侧的单选按钮。

4. 选择 Actions，然后选择 Delete。
5. 在 Delete 选择确认窗格，选择 Delete 删除集群参数组。要保留集群参数组，请选择 Cancel。

使用 Amazon CLI

要删除集群参数组，请使用带以下参数的 `delete-db-cluster-parameter-group` 操作。

- **--db-cluster-parameter-group-name** – 必需。要删除的集群参数组的名称。它必须是现有集群参数组。您不能删除 `default.docdb4.0` 集群参数组。

Example 删除集群参数组

以下示例引导您通过三个步骤删除集群参数组：

1. 找到您要删除的集群参数组的名称。
2. 删除指定的集群参数组。
3. 验证是否已删除集群参数组。

1. 找到要删除的集群参数组的名称。

以下代码列出了所有集群参数组的名称。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \
    --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

对于 Windows：

```
aws docdb describe-db-cluster-parameter-groups ^
    --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

之前操作的输出是集群参数组名称的一个列表，与以下 (JSON 格式) 类似。

```
[
  [
    "default.docdb4.0"
  ],
  [
    "sample-parameter-group"
  ],
  [
    "sample-parameter-group-copy"
  ]
]
```

2. 删除特定的集群参数组。

以下代码删除集群参数组。 `sample-parameter-group-copy`。

对于 Linux、macOS 或 Unix：

```
aws docdb delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group-copy
```

对于 Windows：

```
aws docdb delete-db-cluster-parameter-group ^
--db-cluster-parameter-group-name sample-parameter-group-copy
```

此操作没有输出。

3. 验证指定的集群参数组是否已删除。

以下代码列出了所有剩余集群参数组的名称。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \
--query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

对于 Windows：

```
aws docdb describe-db-cluster-parameter-groups ^
--query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

之前操作的输出是集群参数组列表，与以下（JSON 格式）类似。您刚刚删除的集群参数组不应在列表中。

此操作的输出将类似于下文（JSON 格式）。

```
[
  [
    "default.docdb4.0"
  ],
  [
    "sample-parameter-group"
  ]
]
```

Amazon DocumentDB 集群参数参考

当您更改动态参数并保存集群参数组时，将立即应用更改，而不管 Apply immediately（立即应用）设置。当您更改静态参数并保存集群参数组时，参数更改将在您手动重启实例后生效。您可以使用 Amazon DocumentDB 控制台或通过显式调用重启实例。reboot-db-instance。

下表显示了适用于 Amazon DocumentDB 集群中的所有实例的参数。

Amazon DocumentDB 集群级参数

参数	默认值	有效值	可修改	应用类型	数据类型	描述
audit_logs	disabled	启用，禁用	是	动态	字符串	定义是否启用了 Amazon CloudTrail 审核日志。 enabled—CloudTrail 审核日志已启用。 disabled—CloudTrail 审核日志已禁用。

参数	默认值	有效值	可修改	应用类型	数据类型	描述
change_stream_log_retention_in_hours	168	1-168	是	动态	整数	定义保留更改流日志并且可以使用的持续时间长度（以秒为单位）。
profiler	disabled	启用，禁用	是	动态	字符串	为慢速操作启用分析。 • enabled——超过客户定义阈值（例如 100 毫秒）的操作将记录到 Amazon CloudWatch Logs。 • disabled——缓慢的操作不会记录到 CloudWatch Logs 中。
profiler_sampling_rate	1.0	0.0-1.0	是	动态	Float	定义记录操作的采样率。
profiler_threshold_ms	100	50-2147483646	是	动态	整数	为 profiler 定义阈值。 • 大于所有操作 profiler_threshold 记录到 CloudWatch Logs。

参数	默认值	有效值	可修改	应用类型	数据类型	描述
tls	已启用	启用，禁用	是	静态	字符串	定义是否需要传输层安全性 (TLS) 连接。 • enabled需要 TLS 连接进行连接。 • disabledTLS 连接不能用于连接。
ttl_monitor	已启用	启用，禁用	是	动态	字符串	定义是否为集群启用有效时间 (TTL) 监控。 • enabled—TTL 监控已启用。 • disabled禁用 TTL 监控。

修改 Amazon DocumentDB 集群参数

在 Amazon DocumentDB 中，集群参数组由参数这些默认设置适用于您在集群中创建的所有实例。对于自定义集群参数组，您可以随时修改参数值，或将所有参数值重置为其默认值。本节介绍如何查看构成 Amazon DocumentDB 集群参数组的参数及妻值，以及如何更改或更新这些值。

参数可以是动态或静态的。当您更改动态参数并保存集群参数组时，将立即应用更改，而不管Apply Immediately设置。当您更改静态参数并保存集群参数组时，参数更改仅在您手动重启实例后生效。

查看 Amazon DocumentDB 集群参数组的参数

您可以使用查看 Amazon DocumentDB 集群的参数及妻值。Amazon Web Services Management Console要么Amazon CLI。

使用 Amazon Web Services Management Console

要查看集群参数组的详细信息

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择参数组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (☰)。

3. 在 Parameter groups (参数组) 窗格中，选择要查看其详细信息的集群参数组的名称。

4. 生成的页面显示每个参数的以下值：参数的名称、当前值、允许值、参数是否可修改、应用类型、数据类型和描述。

	Cluster parameter name ▲	Values ▼	Allowed values
☐	audit_logs	disabled	enabled,disabled
☐	tls	enabled	disabled,enabled
☐	ttl_monitor	enabled	disabled,enabled

使用 Amazon CLI

要查看集群参数组的参数及其值，请使用带有以下参数的 `describe-db-cluster-parameters` 操作。

- **--db-cluster-parameter-group-name** – 必需。您想要详细参数列表的集群参数组的名称。
- **--source** 可选。如果提供特定源，仅返回参数。参数源可以是 `engine-default`、`system` 或 `user`。

Example

以下代码列出了 `custom3-6-param-grp` 参数组的所有参数及其值。有关参数组的更多信息，请省略 `--query` 行。有关所有参数组的信息，请省略 `--db-cluster-parameter-group-name` 行。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name custom3-6-param-grp \
  --query 'Parameters[*].[ParameterName,ParameterValue]'
```

对于 Windows：

```
aws docdb describe-db-cluster-parameters ^
  --db-cluster-parameter-group-name custom3-6-param-grp ^
  --query 'Parameters[*].[ParameterName,ParameterValue]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[
  [
    "audit_logs",
    "disabled"
  ],
  [
    "tls",
    "enabled"
  ],
  [
    "ttl_monitor",
    "enabled"
  ]
]
```

修改 Amazon DocumentDB 集群参数组的参数

您可以使用修改参数组的参数。Amazon Web Services Management Console 要么 Amazon CLI。

使用 Amazon Web Services Management Console

要更新集群参数组的参数

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>.
2. 在导航窗格中，选择参数组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在参数组窗格中，选择要更新其参数的集群参数组。
4. 结果页面显示了此集群参数组的参数及其对应的详细信息。选择要更新的参数。
5. 在页面右上角，选择编辑以更改参数的值。有关集群参数类型的更多信息，请参阅[Amazon DocumentDB 集群参数参考 \(p. 337\)](#).
6. 进行更改，然后选择修改集群参数保存更改。要放弃更改，请选择更改，Cancel.

使用 Amazon CLI

要修改集群参数组的参数，请使用带以下参数的 `modify-db-cluster-parameter-group` 操作：

- **--db-cluster-parameter-group-name** – 必需。您正在修改的集群参数组的名称。
- **--parameters** – 必需。您正在修改的参数。每个参数条目必须包含以下内容：
 - **ParameterName**— 您正在修改的参数的名称。
 - **ParameterValue**— 此参数的新值。
 - **ApplyMethod**— 您希望如何对应用的参数进行更改。允许的值为 `immediate` 和 `pending-reboot`。

Note

带 `static` 的 `ApplyType` 参数必须具有 `pending-reboot` 的 `ApplyMethod`。

要更改集群参数组的参数值 (Amazon CLI)

以下示例将更改 `tls` 参数。

1. 列出的参数及其值 **sample-parameter-group**

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows：

```
aws docdb describe-db-cluster-parameters ^
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "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. 修改 **tls** 参数，以便其值为 **disabled**。您无法修改 **ApplyMethod** 由于 **ApplyType** 是 **static**。

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name sample-parameter-group \
  --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

对于 Windows：

```
aws docdb modify-db-cluster-parameter-group ^
  --db-cluster-parameter-group-name sample-parameter-group ^
  --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

此操作的输出将类似于下文（JSON 格式）。

```
{
  "DBClusterParameterGroupName": "sample-parameter-group"
}
```

3. 等至少 5 分钟。
4. 列出 **sample-parameter-group** 的参数值。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows：

```
aws docdb describe-db-cluster-parameters ^
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文（JSON 格式）。

```
{
  "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"
    }
  ]
}
```

了解 Amazon DocumentDB 终端节点

您可以使用 Amazon DocumentDB (与 MongoDB 兼容) 终端节点连接到集群或实例。Amazon DocumentDB 有三种不同类型的终端节点，分别用于不同用途。

主题

- [查找集群的终端节点 \(p. 344\)](#)
- [查找实例的终端节点 \(p. 345\)](#)
- [连接到 终端节点 \(p. 348\)](#)

集群终端节点

集群终端节点是 Amazon DocumentDB 集群的一个终端节点，连接到该集群的当前主实例。每个 Amazon DocumentDB 集群都具有单个集群终端节点和一个主实例。在发生故障转移时，集群终端节点将重新映射到新的主实例。

读取器终端节点

读取器终端节点是 Amazon DocumentDB 集群的一个终端节点，连接到该集群的可用副本之一。每个 Amazon DocumentDB 集群都具有读取器终端节点。如果有多个副本，则读取器终端节点会将每个连接请求定向到 Amazon DocumentDB 副本之一。

实例终端节点

实例终端节点是连接到特定实例的终端节点。集群中的每个实例（不论其是主实例还是副本实例）都有各自唯一的实例终端节点。最好不要使用应用程序中的实例终端节点。这是因为它们在发生故障转移时会更改角色，从而需要在应用程序中更改代码。

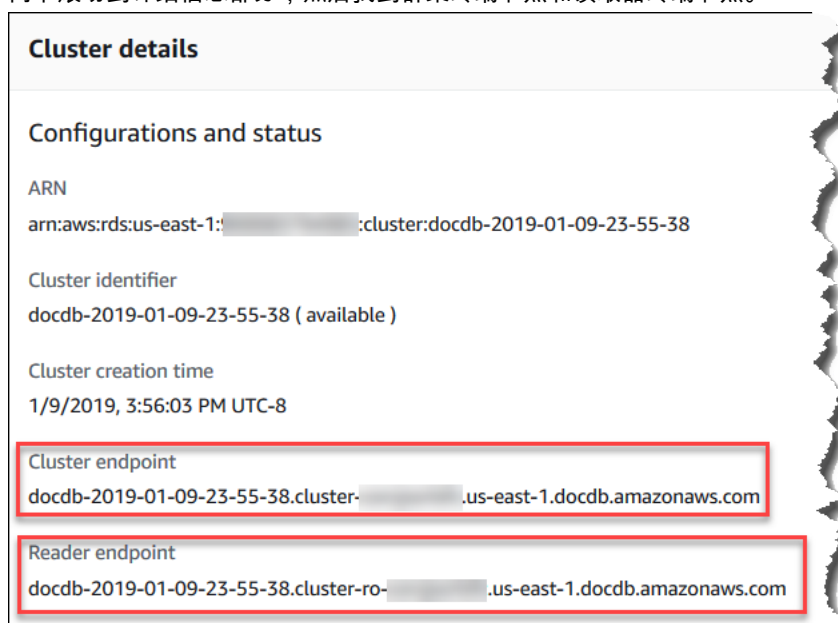
查找集群的终端节点

您可使用 Amazon DocumentDB 控制台查找集群的集群终端节点和读取器终端节点，或 Amazon CLI。

使用控制台

使用控制台查找集群的终端节点

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择集群。
3. 从集群列表中，选择感兴趣的集群的名称。
4. 向下滚动到详细信息部分，然后找到集群终端节点和读取器终端节点。



5. 要连接到此群集，请向上滚动至 Connect (连接) 部分。查找连接字符串 mongoshell 和可以在应用程序代码中使用的连接字符串以连接到集群。



使用 Amazon CLI

要使用 Amazon CLI 查找集群的集群和读取器终端节点，请运行具有以下参数的 `describe-db-clusters` 命令。

参数

- **--db-cluster-identifier**— 可选。指定要为其返回终端节点的群集。如果省略，将返回最多 100 个集群的终端节点。

- **--query**— 可选。指定要显示的字段。减少查找终端节点所需查看的数据量，从而有用。如果省略，则返回有关集群的全部信息。
- **--region**— 可选。使用--region参数指定要将命令应用到的区域。如果省略，则使用默认区域。

Example

以下示例将返回 sample-cluster 的 DBClusterIdentifier、终端节点（集群终端节点）和 ReaderEndpoint。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
  --region us-east-1 \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

对于 Windows：

```
aws docdb describe-db-clusters ^  
  --region us-east-1 ^  
  --db-cluster-identifier sample-cluster ^  
  --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[  
  [  
    "sample-cluster",  
    27017,  
    "sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com",  
    "sample-cluster.cluster-ro-corlsfccjozr.us-east-1.docdb.amazonaws.com"  
  ]  
]
```

现在您已使用集群终端节点，可使用以下两种方式连接到集群。mongo 要么 mongodb。有关更多信息，请参阅[连接到终端节点 \(p. 348\)](#)。

查找实例的终端节点

您可使用 Amazon DocumentDB 控制台或查找实例的终端节点。Amazon CLI。

使用控制台

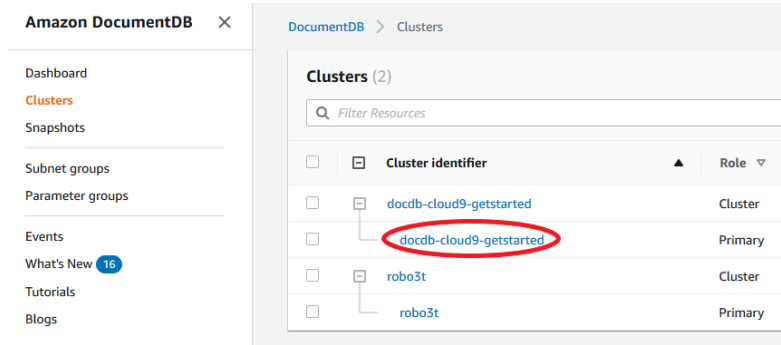
使用控制台查找实例的终端节点

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

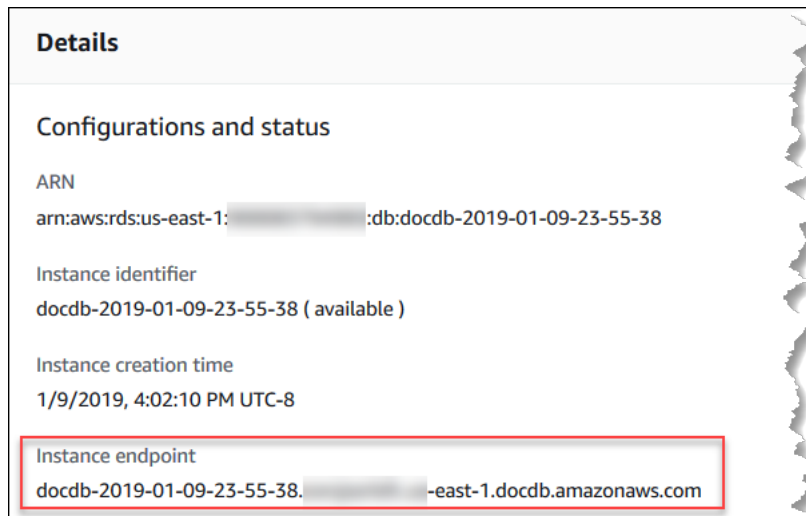
Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标（☰）。

3. 在集群导航框中，将看到该列 Cluster Identifier。您的实例列在群集下，类似于下面的屏幕截图。



4. 选中您感兴趣的实例左侧的框。
5. 向下滚动至 Details (详细信息) 部分，然后找到实例终端节点。



6. 要连接到此实例，请向上滚动至 Connect (连接) 部分。查找连接字符串 mongoshell 和 可以在应用程序代码中使用的连接字符串以连接到实例。



使用 Amazon CLI

要使用 Amazon CLI 查找实例终端节点，请运行带有以下参数的命令。

Arguments

- **--db-instance-identifier**— 可选。指定要为其返回终端节点的实例。如果省略，将返回最多 100 个实例的终端节点。
- **--query**— 可选。指定要显示的字段。减少查找终端节点所需查看的数据量，从而有用。如果省略，将返回实例的所有信息。这些区域有：Endpoint 字段有三个成员，因此在查询中列出它，如以下示例所示，将返回所有三个成员。如果您只对部分 Endpoint 成员感兴趣，请将查询中的 Endpoint 替换为您感兴趣的成员，如第二个示例中所示。
- **--region**— 可选。使用 --region 参数指定要将命令应用到的区域。如果省略，则使用默认区域。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --region us-east-1 \  
  --db-instance-identifier sample-cluster-instance \  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

对于 Windows：

```
aws docdb describe-db-instances ^  
  --region us-east-1 ^  
  --db-instance-identifier sample-cluster-instance ^  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[  
  [  
    "sample-cluster-instance",  
    {  
      "Port": 27017,  
      "Address": "sample-cluster-instance.corcjozrlsfc.us-  
east-1.docdb.amazonaws.com",  
      "HostedZoneId": "Z2R2ITUGPM61AM"  
    }  
  ]  
]
```

减少输出以消除终端节点的 HostedZoneId，可通过指定 Endpoint.Port 和 Endpoint.Address 来修改查询。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \  
  --region us-east-1 \  
  --db-instance-identifier sample-cluster-instance \  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

对于 Windows：

```
aws docdb describe-db-instances ^  
  --region us-east-1 ^  
  --db-instance-identifier sample-cluster-instance ^  
  --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

此操作的输出将类似于下文（JSON 格式）。

```
[  
  [  
    "sample-cluster-instance",  
    27017,  
    "sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com"  
  ]  
]
```

现在您已使用实例终端节点，那么便可使用以下两种方式连接到实例。mongo 要么 mongodb。有关更多信息，请参阅[连接到终端节点](#)（p. 348）。

连接到 终端节点

当您拥有终端节点 (集群或实例) 时, 您可以使用mongoShell 或连接字符串。

使用 mongo Shell 进行连接

使用以下结构来构建连接到集群或实例所需的字符串, 使用mongoShell:

```
mongo \  
  --ssl \  
  --host Endpoint:Port \  
  --sslCAFile rds-combined-ca-cn-bundle.pem \  
  --username UserName \  
  --password Password
```

mongoShell 示例

Connect 到集群 :

```
mongo \  
  --ssl \  
  --host sample-cluster.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \  
  --sslCAFile rds-combined-ca-cn-bundle.pem \  
  --username UserName \  
  --password Password
```

Connect 到实例 :

```
mongo \  
  --ssl \  
  --host sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \  
  --sslCAFile rds-combined-ca-cn-bundle.pem \  
  --username UserName \  
  --password Password
```

使用连接字符串进行连接

使用以下结构构建连接到集群或实例所需的连接字符串。

```
mongodb://UserName:Password@endpoint:port?replicaSet=rs0&ssl_ca_certs=rds-combined-ca-cn-bundle.pem
```

连接字符串示例

Connect 到集群 :

```
mongodb://UserName:Password@sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=rds-combined-ca-cn-bundle.pem
```

Connect 到实例 :

```
mongodb://UserName:Password@sample-cluster-instance.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=rds-combined-ca-cn-bundle.pem
```

了解 Amazon DocumentDB Amazon 资源名称 (ARN)

您在 Amazon 中创建的资源分别使用 Amazon 资源名称 (ARN) 进行唯一标识。对于某些 Amazon DocumentDB (与 MongoDB 兼容) 操作, 您必须通过指定 Amazon DocumentDB 资源的 ARN 来唯一标识该资源。例如, 当您向资源中添加标签时, 必须提供该资源的 ARN。

主题

- [构建 Amazon DocumentDB 资源的 ARN \(p. 349\)](#)
- [查找 Amazon DocumentDB 资源 ARN \(p. 350\)](#)

构建 Amazon DocumentDB 资源的 ARN

您可以使用以下语法为 Amazon DocumentDB 资源构建 ARN。Amazon DocumentDB 共享 Amazon DocumentDB 共享 Amazon Relational Database Service 共享 Amazon DocumentDB 的格式。Amazon DocumentDB ARN 包含 `rds` 而不是 `docdb`。

`arn:aws:rds:region:account_number:resource_type:resource_id`

区域名称	区域	可用区 (计算)
美国东部 (俄亥俄)	us-east-2	3
美国东部 (弗吉尼亚北部)	us-east-1	6
美国西部 (俄勒冈)	us-west-2	4
南美洲 (圣保罗)	sa-east-1	3
亚太地区 (孟买)	ap-south-1	3
亚太地区 (首尔)	ap-northeast-2	4
亚太地区 (新加坡)	ap-southeast-1	3
亚太地区 (悉尼)	ap-southeast-2	3
亚太地区 (东京)	ap-northeast-1	3
加拿大 (中部)	ca-central-1	3
中国 (北京) 区域	cn-north-1	3
中国 (宁夏)	cn-northwest-1	3
欧洲 (法兰克福)	eu-central-1	3
欧洲 (爱尔兰)	eu-west-1	3
欧洲 (伦敦)	eu-west-2	3
欧洲 (米兰)	eu-south-1	3

区域名称	区域	可用区 (计算)
欧洲 (巴黎)	eu-west-3	3
Amazon GovCloud (US)	us-gov-west-1	3

Note

Amazon DocumentDB 架构将存储和计算分开。对于存储层，Amazon DocumentDB 会在三个之间复制 6 个数据副本。Amazon 可用区 (AZ)。上表中列出的 AZ 是指，您可以在给定区域中预置计算实例时使用的 AZ 数。例如，如果要在 ap-northeast-1 中启动 Amazon DocumentDB 集群，将在三个 AZ 之间的 6 个方向复制存储，但仅在两个 AZ 中具有计算实例。

下表显示在构建特定 Amazon DocumentDB 资源的 ARN 时应使用的格式。Amazon DocumentDB 共享 Amazon RDS ARN 的格式。Amazon DocumentDB ARN 包含 rds 而不是 docdb。

资源类型	ARN 格式/示例
实例 (db)	arn:aws:rds:region:account_number:db:resource_id arn:aws:rds:us-east-1:1234567890:db:sample-db-instance
集群 (cluster)	arn:aws:rds:region:account_number:cluster:resource_id arn:aws:rds:us-east-1:1234567890:cluster:sample-db-cluster
集群参数组 (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
安全组 (secgrp)	arn:aws:rds:region:account_number:secgrp:resource_id arn:aws:rds:us-east-1:1234567890:secgrp:sample-public-secgrp
集群快照 (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
子网组 (subgrp)	arn:aws:rds:region:account_number:subgrp:resource_id arn:aws:rds:us-east-1:1234567890:subgrp:sample-subnet-10

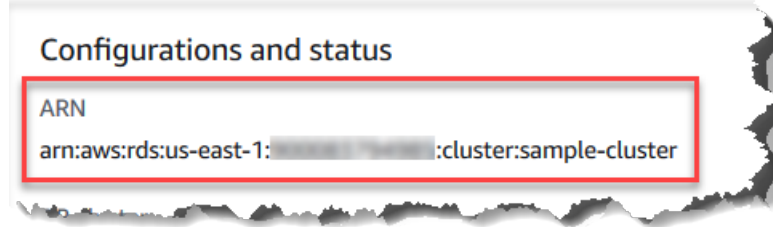
查找 Amazon DocumentDB 资源 ARN

您可以 Amazon DocumentDB 用 Amazon Web Services Management Console 或者 Amazon CLI。

使用 Amazon Web Services Management Console

要使用控制台查找 ARN，请导航到要获取其 ARN 的资源，然后查看该资源的详细信息。

例如，您可以在集群的详细信息窗格中获取集群的 ARN，如以下屏幕截图所示。



使用 Amazon CLI

要 ARN 用 Amazon CLI 对于特定的 Amazon DocumentDB 资源，请使用 `describe` 该资源的操作。下表显示了每个 Amazon CLI 操作，以及与操作配合使用以获取 ARN 的 ARN 属性。

Amazon CLI 命令	ARN 属性
<code>describe-db-instances</code>	<code>DBInstanceArn</code>
<code>describe-db-clusters</code>	<code>DBClusterArn</code>
<code>describe-db-parameter-groups</code>	<code>DBParameterGroupArn</code>
<code>describe-db-cluster-parameter-groups</code>	<code>DBClusterParameterGroupArn</code>
<code>describe-db-security-groups</code>	<code>DBSecurityGroupArn</code>
<code>describe-db-snapshots</code>	<code>DBSnapshotArn</code>
<code>describe-db-cluster-snapshots</code>	<code>DBClusterSnapshotArn</code>
<code>describe-db-subnet-groups</code>	<code>DBSubnetGroupArn</code>

Example - 查找集群的 ARN

以下 Amazon CLI 操作可查找集群 `sample-cluster` 的 ARN。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].DBClusterArn'
```

对于 Windows：

```
aws docdb describe-db-clusters ^  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].DBClusterArn'
```

此操作的输出将类似于下文（JSON 格式）。

```
[
```



```
] "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster"
```

Example - 查找多个参数组的 ARN

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameter-groups \
  --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

对于 Windows：

```
aws docdb describe-db-cluster-parameter-groups ^
  --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

此操作的输出将类似于下文（JSON 格式）。

```
[
  "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"
]
```

为 Amazon DocumentDB 资源添加标签

您可以使用 Amazon DocumentDB（具有 MongoDB 兼容性）标签向 Amazon DocumentDB 资源添加元数据。这些标签可与 Amazon Identity and Access Management(IAM) 策略，以管理对 Amazon DocumentDB 资源的访问并控制可将什么操作应用于资源。您还可以将具有类似标签的资源的费用分组在一起，使用标签来跟踪成本。

您可以为以下 Amazon DocumentDB 资源添加标签：

- 集群
- 实例
- 快照
- 集群快照
- 参数组
- 集群参数组
- 安全组
- 子网组

Amazon DocumentDB 资源标签概述

Amazon DocumentDB 标签是由您定义的名称-值对，与某种 Amazon DocumentDB 资源关联。此名称也叫键。为键提供值为可选操作。您可以使用标签向 Amazon DocumentDB 资源分配任意信息。例如，您可以使用标签键定义一个类别，而标签值可以是该类别中的一个项目。例如，您可以定义一个标签键 `project` 和标签值 `Salix`，表示 Amazon DocumentDB 资源分配给 Salix 项目。您也可以使用标签通过以下键指定 Amazon DocumentDB 资源用于测试或生产。 `environment=test` 要么 `environment=production`。我们建议使用一组具有一致性的标签键，以使跟踪与 Amazon DocumentDB 资源关联的元数据变得更轻松。

您也可以使用标签来组织您的 Amazon 账单，使其反映您的成本结构。要执行此操作，请注册以获取包含标签键值的 Amazon Web Services 账户 账单。然后，如需查看组合资源的成本，请按有同样标签键值的资源组织您的账单信息。例如，您可以将特定的应用程序名称用作几个资源的标签，然后组织账单信息，以查看在数个服务中的使用该应用程序的总成本。有关更多信息，请参阅 [使用成本分配标签](#) 中的 Amazon Billing and Cost Management 用户指南。

每个 Amazon DocumentDB 资源都有一组标签，其中包含分配给该资源的所有标签。一个标签集可以包含多达 10 个标签，也可以为空。如果向 Amazon DocumentDB 资源添加一个标签，而该标签的键与资源上某个现有的标签相同，则新值将覆盖旧值。

Amazon 不会对您的标记应用任何语义意义；所有标记都会严格地作为字符串进行解析。Amazon DocumentDB 可以在实例或其他 Amazon DocumentDB 资源上设置标签，这取决于您创建资源时使用的设置。例如，Amazon DocumentDB 可添加一个标签来指示实例用于生产或测试。

您可以将标签添加到快照中，但您的账单不会反映此分组。

您可以使用 Amazon Web Services Management Console 或者 Amazon CLI 以添加、列出和删除 Amazon DocumentDB 资源上的标签。使用 Amazon CLI 时，必须提供要使用的资源的 Amazon 资源名称 (ARN)。有关 Amazon DocumentDB ARN 的更多信息，请参阅 [了解 Amazon DocumentDB Amazon 资源名称 \(ARN\)](#) (p. 349)。

标签约束

以下限制适用于 Amazon DocumentDB 标签：

- 每个资源的最大标签数-10
- 最大键长度 - 128 个 Unicode 字符
- 最大值长度 - 256 个 Unicode 字符
- 键和值的有效字符 - UTF-8 字符集中的大写和小写字母、位、空格及以下字符：_ . : / = + - 和 @ (Java regex: `"^([\p{L}\p{Z}\p{N}_.:/=+\-]*)$"`)
- 标签键和值区分大小写。
- 前缀 aws：无法用于标签键或值；将其保留供 Amazon 使用。

在 Amazon DocumentDB 资源上添加和更新标签

您最多可以使用 10 个标签添加到资源 Amazon Web Services Management Console 或者 Amazon CLI。

使用 Amazon Web Services Management Console

无论您将标签添加到哪个资源，向资源添加标签的过程都是相似的。在本示例中，您向集群添加标签。

要使用控制台向集群添加标签或更新标签

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 从导航窗格中，选择集群。
3. 选择要将标签添加到的集群的名称。
4. 向下滚动到标签部分，然后选择 编辑。
5. 对于每个要添加到此资源的标签，请执行以下操作：
 - a. 要添加新标签，请在值框中输入标签的名称。要更改标签的值，请在值列中找到标签的名称。
 - b. 要赋予标签新建或更新的值，在值框中，为标签输入一个值。
 - c. 如果要添加多个标签，请选择添加。否则，完成后，选择保存。

使用 Amazon CLI

无论您将标签添加到哪个资源，向资源添加标签的过程都是相似的。在本示例中，您向集群添加三个标签。第二个标签 `key2` 没有值。

使用带有这些参数的 Amazon CLI 操作 `add-tags-to-resource`。

参数

- **--resource-name**— 要向其添加标签的 Amazon DocumentDB 资源的 ARN。
- **--tags**— 以格式列出要添加到此资源的标签（键-值对）`Key=key-name,Value=tag-value`。

Example

对于 Linux、macOS 或 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
```

对于 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
```

此 `add-tags-to-resource` 操作不会生成任何输出。要查看操作结果，请使用 `list-tags-for-resource` 操作。

列出 Amazon DocumentDB 资源上的标签

您可以使用 Amazon Web Services Management Console 或者 Amazon CLI 以获取 Amazon DocumentDB 资源的标签列表。

使用 Amazon Web Services Management Console

无论您将标签添加到哪个资源，在资源上列出标签的过程都是相似的。在本示例中，您为集群列出标签。

要使用控制台列出集群上的标签

1. 在处打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 从导航窗格中，选择集群。
3. 选择要为之列出标签的集群的名称。
4. 要查看此资源上的标签列表，向下滚动到标签部分。

使用 Amazon CLI

无论您为哪个资源列出标签，在资源上列出标签的过程都是相似的。在本示例中，您在集群上列出标签。

使用带有这些参数的 Amazon CLI 操作 `list-tags-for-resource`。

参数

- **--resource-name** – 必填项。要列出其标签的 Amazon DocumentDB 资源的 ARN。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb list-tags-for-resource \  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

对于 Windows：

```
aws docdb list-tags-for-resource ^  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "TagList": [  
    {  
      "Key": "key1",  
      "Value": "value1"  
    },  
    {  
      "Key": "key2",  
      "Value": ""  
    },  
    {  
      "Key": "key3",  
      "Value": "value3"  
    }  
  ]  
}
```

删除 Amazon DocumentDB 资源的标签

您可以使用 Amazon Web Services Management Console 或者 Amazon CLI 以删除 Amazon DocumentDB 资源的标签。

使用 Amazon Web Services Management Console

无论您将标签添加到哪个资源，从资源中删除标签的过程都是相似的。在本示例中，您从集群中删除标签。

使用控制台从集群中删除标签

1. 在处打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 从导航窗格中，选择集群。
3. 选择要从中删除标签的集群的名称。
4. 向下滚动到标签部分，然后选择 编辑。
5. 如果您要从该资源中删除所有标签，请选择 Remove all（删除所有）。否则，对于每个要从此资源删除的标签，请执行以下操作：
 - a. 查找键列中的标签的名称。
 - b. 在同一行上选择删除作为标签值。
 - c. 完成后，选择保存。

使用 Amazon CLI

无论您从哪个资源删除标签，从资源中删除标签的过程都是相似的。在本示例中，您从集群中删除标签。

使用带有这些参数的 Amazon CLI 操作 `remove-tags-from-resource`。

- `--resource-name` – 必填项。要从中删除标签的 Amazon DocumentDB 资源的 ARN。
- `--tag-keys` – 必填项。要从此资源中删除的标签值列表。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb remove-tags-from-resource \  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \  
  --tag-keys key1 key3
```

对于 Windows：

```
aws docdb remove-tags-from-resource ^  
  --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \  
  --tag-keys key1 key3
```

此 `removed-tags-from-resource` 操作不会生成任何输出。要查看操作结果，请使用 `list-tags-for-resource` 操作。

维护 Amazon DocumentDB

Amazon DocumentDB 会定期对 Amazon DocumentDB 资源执行维护。维护最常涉及对数据库引擎（集群维护）或实例的底层操作系统（OS）（实例维护）的更新。

某些维护项目要求 Amazon DocumentDB 使实例脱机一小段时间。要求实例脱机的维护项目包括必需的操作系统或引擎修补。仅对与安全性和实例可靠性相关的修补程序自动安排必需的修补。在集群或实例上执行维护时，将会出现这种情况；如果实例是主实例，将进行故障转移。有关更多信息，请参阅[Amazon DocumentDB 故障转移 \(p. 313\)](#)。

集群和实例维护都有各自的维护时段。默认情况下，Amazon DocumentDB 会为集群和每个单独的实例分配维护时段。您可以在创建集群或实例时选择维护时段。也可以随时修改维护时段以适应您的业务计划或实践。通常，建议选择尽量减少对应用程序的影响的维护时段（例如，晚上或周末）。本指南与您遇到的应用程序类型和使用模式密切相关。

主题

- [确定待处理的 Amazon DocumentDB 维护操作 \(p. 356\)](#)
- [应用 Amazon DocumentDB 更新 \(p. 357\)](#)
- [用户启动的更新 \(p. 360\)](#)
- [管理您的 Amazon DocumentDB 维护窗口 \(p. 360\)](#)

确定待处理的 Amazon DocumentDB 维护操作

您可以使用 Amazon Web Services Management Console 或 Amazon CLI 来查看维护更新是否对集群可用。

如果有可用更新，您可以执行以下操作之一：

- 推迟维护操作。
- 立即应用维护操作。
- 计划下一个维护时段内要开始的维护操作。

- 不执行任何操作。

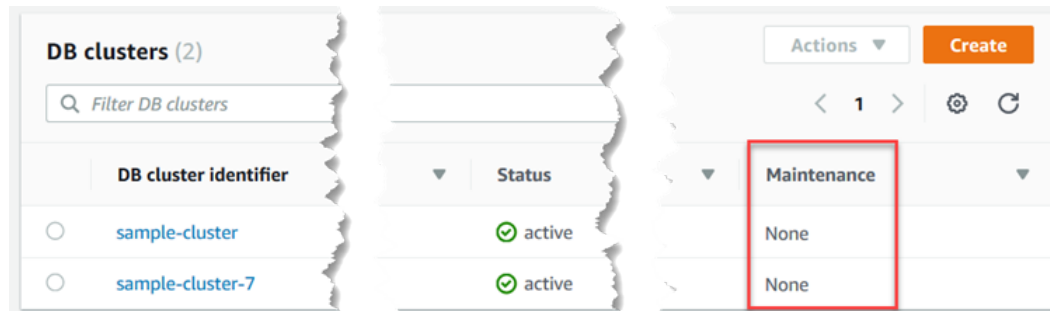
Important

某些 OS 更新标记为必需。如果您推迟某个必需的更新，则会收到一条来自 Amazon DocumentDB 的通知，指示何时对实例或集群执行此更新。其他更新为可用。您可以无限期推迟这些更新。

维护时段确定待处理的操作何时开始，但不限制这些操作的总执行时间。维护操作不保证在维护时段结束前完成，可以在超出指定的结束时间后继续。

使用 Amazon Web Services Management Console

如果有更新，则用单词表示 Available 要么必需中的维护 Amazon DocumentDB 控制台上集群的列，如下所示：



使用 Amazon CLI

使用以下 Amazon CLI 操作确定待处理的维护操作。此处的输出显示没有待处理的维护操作。

```
aws docdb describe-pending-maintenance-actions
```

此操作的输出将类似于下文（JSON 格式）。

```
{
  "PendingMaintenanceActions": []
}
```

应用 Amazon DocumentDB 更新

通过 Amazon DocumentDB，您可以选择何时应用维护操作。您可以使用 Amazon DocumentDB 何时应用更新。Amazon Web Services Management Console 要么 Amazon CLI。

使用此主题中的过程对实例立即执行升级或者计划升级。

使用 Amazon Web Services Management Console

您可以使用控制台管理 Amazon DocumentDB 实例和集群的更新。

管理实例或集群的更新

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择集群。
3. 在集群列表中，选择要应用维护操作的集群名称旁边的按钮。

4. 在 Actions (操作) 菜单中, 选择以下选项之一 :

- Upgrade now (立即升级), 以立即执行挂起的维护任务。
- Upgrade at next window (在下一个时段升级), 以在集群的下一个维护时段中执行挂起的维护任务。

Note

如果没有挂起的维护任务, 上面的两个选项均会处于非活动状态。

使用 Amazon CLI

要将挂起的更新应用于实例或集群, 请使用 `apply-pending-maintenance-action` Amazon CLI 操作。

参数

- **--resource-identifier**— 待处理的维护操作应用于的资源的 Amazon DocumentDB 资源名称 (ARN)。
- **--apply-action**— 应用于此资源的待处理的维护操作。

有效值 : `system-update` 和 `db-upgrade`。

- **--opt-in-type**— 用于指定加入请求类型或撤消加入请求的值。不能撤消 `immediate` 类型的加入请求。

有效值 :

- `immediate`— 立即应用维护操作。
- `next-maintenance`— 在资源的下一个维护时段内应用维护操作。
- `undo-opt-in`— 取消任何现有 `next-maintenance` 加入请求。

Example

对于 Linux、macOS 或 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
```

对于 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
```

要返回至少有一个挂起更新的资源列表, 请使用 `describe-pending-maintenance-actions` Amazon CLI 操作。

Example

对于 Linux、macOS 或 Unix :

```
aws docdb describe-pending-maintenance-actions \  
  --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

对于 Windows :

```
aws docdb describe-pending-maintenance-actions ^
--resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "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"
        }
      ]
    }
  ]
}
```

您还可以通过指定 `--filters` 的参数 `describe-pending-maintenance-actions` Amazon CLI operation. `--filters` 操作的格式是 `Name=filter-name,Values=resource-id,...`

下面是筛选条件的 `Name` 参数可接受的值：

- `db-cluster-id`— 接受集群标识符或 ARN 的列表。返回的列表只包括这些标识符或 ARN 标识的集群的挂起的维护操作。
- `db-instance-id`— 接受实例标识符或 ARN 的列表。返回的列表只包括这些标识符或 ARN 标识的实例的挂起的维护操作。

以下示例返回 `sample-cluster1` 和 `sample-cluster2` 集群的待处理维护操作。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb describe-pending-maintenance-actions \
--filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

对于 Windows：

```
aws docdb describe-pending-maintenance-actions ^
--filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

应用日期

每个维护操作都有一个相应的应用日期，您可以在描述待处理的维护操作时找到它们。当您从 Amazon CLI 读取待处理维护操作的输出时，将列出以下三个日期：

- **CurrentApplyDate**— 将立即应用或在下一个维护时段期间应用维护操作的日期。如果维护是可选的，则该值可以为 `null`。
- **ForcedApplyDate**— 自动应用维护的日期，与维护时段无关。
- **AutoAppliedAfterDate**— 将在该日期后的集群维护时段期间应用维护。

用户启动的更新

作为 Amazon DocumentDB 用户，您可以启动对集群或实例的更新。例如，您可以将实例的类修改为具有更多或更少内存的类，也可以更改集群的参数组。Amazon DocumentDB 查看这些更改与 Amazon DocumentDB 启动的更新不同。有关修改集群或实例的更多信息，请参阅以下内容：

- [修改 Amazon DocumentDB 集群 \(p. 259\)](#)
- [修改 Amazon DocumentDB 实例 \(p. 295\)](#)

要查看待处理的用户启动的修改的列表，请运行以下命令。

Example

查看实例的待处理的用户启动的更改

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
  --query 'DBInstances[*].
  [DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

对于 Windows：

```
aws docdb describe-db-instances ^
  --query 'DBInstances[*].
  [DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

此操作的输出将类似于下文（JSON 格式）。

在本例中，sample-cluster-instance 有针对 db.r5.xlarge 实例类的待处理更改，sample-cluster-instance-2 没有待处理的更改。

```
[
  [
    "sample-cluster",
    "sample-cluster-instance",
    {
      "DBInstanceClass": "db.r5.xlarge"
    }
  ],
  [
    "sample-cluster",
    "sample-cluster-instance-2",
    {}
  ]
]
```

管理您的 Amazon DocumentDB 维护窗口

每个实例和集群都有一个每周维护时段，在此期间会应用任何待处理的更改。在请求或要求的情况下，您可以将维护时段视为控制修改及软件修补程序更新的时间的机会。如果在给定的周内安排了维护事件，则将在您确定的 30 分钟维护时段内启动维护。大部分维护事件也将在 30 分钟的维护时段内完成，但较大的维护事件可能需要 30 分钟以上的时间才能完成。

这个 30 分钟维护时段是随机从每个区域的 8 小时时间段中选择出来的。如果您未在创建实例或集群时指定首选维护时段，Amazon DocumentDB 会在一个星期中随机选择的某天为您分配 30 分钟的维护时段。

区域	UTC 时间数据块
美国东部 (俄亥俄)	03:00-11:00
美国东部 (弗吉尼亚北部)	03:00-11:00
美国西部 (俄勒冈)	06:00-14:00
亚太地区 (孟买)	17:30-01:30
亚太地区 (首尔)	13:00-21:00
亚太地区 (新加坡)	14:00-22:00
亚太地区 (悉尼)	12:00-20:00
亚太地区 (东京)	13:00-21:00
加拿大 (中部)	22:00-06:00
欧洲 (法兰克福)	23:00-07:00
欧洲 (爱尔兰)	22:00-06:00
欧洲 (巴黎)	22:00-06:00
Amazon GovCloud (US)	22:00-06:00

更改维护时段

维护时段应当选在使用量最小的时段上，因而可能必须不时予以更改。您的集群或实例只会在应用系统更改（例如，扩展存储操作或数据库实例类的更改）并且需要中断的期间出现不可用现象，且持续时间只是这些必要更改所需的最少时间。

对于数据库引擎升级，Amazon DocumentDB 会使用集群的首选维护时段，而不是单个实例的维护时段。

更改维护时段

- 对于集群：请参阅[修改 Amazon DocumentDB 集群 \(p. 259\)](#)。
- 对于实例：请参阅[修改 Amazon DocumentDB 实例 \(p. 295\)](#)。

了解服务相关角色

Amazon DocumentDB (与 MongoDB 兼容) 使用 Amazon Identity and Access Management(IAM) 服务相关角色。一个[服务相关角色](#)是一种独特类型的 IAM 角色，它与 Amazon DocumentDB 直接关联。服务相关角色由 Amazon DocumentDB 预定义，并包含服务调用其他角色所需的所有权限。Amazon 服务代表您。

服务相关角色可让您更轻松地使用 Amazon DocumentDB，因为您不必手动添加必要的权限。Amazon DocumentDB 定义了其服务相关角色的权限，除非另外定义，否则只有 Amazon DocumentDB 可以代入该角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其它 IAM 实体的权限策略。

只有在首先删除角色的相关资源后，才能删除角色。这将保护您的 Amazon DocumentDB 资源，因为您不会无意中删除对资源的访问权限。

有关支持服务相关角色的其它服务的信息，请参阅[使用 IAM 的 Amazon 服务](#)并查找 Service-Linked Role (服务相关角色) 列中显示为 Yes (是) 的服务。选择 Yes (是) 和链接，查看该服务的[服务相关角色文档](#)。

Amazon DocumentDB 服务相关角色权限

Amazon DocumentDB (与 MongoDB 兼容) 使用名为的服务相关角色 Amazon 服务角色 ForRDS 允许 Amazon DocumentDB 致电 Amazon 代表您的群集的服务。

AWSServiceRoleForRDS 服务相关角色信任以下服务以担任该角色：

- docdb.amazonaws.com

角色权限策略允许 Amazon DocumentDB 对指定资源完成以下操作：

- 对 ec2 的操作：
 - AssignPrivateIpAddresses
 - AuthorizeSecurityGroupIngress
 - CreateNetworkInterface
 - CreateSecurityGroup
 - DeleteNetworkInterface
 - DeleteSecurityGroup
 - DescribeAvailabilityZones
 - DescribeInternetGateways
 - DescribeSecurityGroups
 - DescribeSubnets
 - DescribeVpcAttribute
 - DescribeVpcs
 - ModifyNetworkInterfaceAttribute
 - RevokeSecurityGroupIngress
 - UnassignPrivateIpAddresses
- 对 sns 的操作：
 - ListTopic
 - Publish
- 对 cloudwatch 的操作：
 - PutMetricData
 - GetMetricData
 - CreateLogStream
 - PullLogEvents
 - DescribeLogStreams
 - CreateLogGroup

Note

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。您可能会遇到以下错误消息：

Unable to create the resource. Verify that you have permission to create service linked role. Otherwise wait and try again later.

如果您看到此错误，请确保您已启用以下权限：

```
{
  "Action": "iam:CreateServiceLinkedRole",
  "Effect": "Allow",
```

```
"Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
"Condition": {
  "StringLike": {
    "iam:AWSServiceName": "rds.amazonaws.com"
  }
}
```

有关更多信息，请参阅 IAM 用户指南中的[服务相关角色权限](#)。

创建 Amazon DocumentDB 服务相关角色

无需手动创建服务相关角色。当您创建集群时，Amazon DocumentDB 将为您创建服务相关角色。

如果删除此服务相关角色然后需要再次创建它，则可以使用相同的流程在您的账户中重新创建此角色。当您创建集群时，Amazon DocumentDB 将再次为您创建服务相关角色。

修改 Amazon DocumentDB 服务相关角色

Amazon DocumentDB 不允许您修改 AWSServiceRoleForRDS 服务相关角色。创建服务相关角色后，将无法更改角色名称，因为可能有多个实体引用该角色。不过，您可以使用 IAM 修改角色的说明。有关更多信息，请参见《IAM 用户指南》中的[编辑服务相关角色](#)。

删除 Amazon DocumentDB 服务相关角色

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。但是，您必须先删除所有 集群，然后才能删除服务相关角色。

清除 Amazon DocumentDB 服务相关角色

必须先确认服务相关角色没有活动会话并删除该角色使用的任何资源，然后才能使用 IAM 删除服务相关角色。

要使用控制台检查服务相关角色是否具有活动会话

1. 在处登录Amazon Web Services Management Console然后在处打开 IAM 控制台：<https://console.amazonaws.cn/iam/>。
2. 在 IAM 控制台的导航窗格中，选择角色，然后选择的名称（不是复选框）。Amazon 服务角色 ForRDS角色。
3. 在所选角色的 Summary 页面上，选择 Access Advisor 选项卡。
4. 在 Access Advisor (访问顾问) 选项卡上，查看服务相关角色的近期活动。

Note

如果您不确定 Amazon DocumentDB 是否正在使用 AWSServiceRoleForRDS 角色，可以尝试删除该角色。如果服务正在使用该角色，则删除操作会失败，并且您可以查看正在使用该角色的 区域。如果该角色已被使用，则您必须等待会话结束，然后才能删除该角色。您无法撤销服务相关角色对会话的权限。

如果您要删除 AWSServiceRoleForRDS 角色，必须先删除。所有实例和集群。有关删除实例和集群的信息，请参阅以下主题：

- [删除 Amazon DocumentDB 实例 \(p. 300\)](#)
- [删除 Amazon DocumentDB 集群 \(p. 267\)](#)

Amazon DocumentDB 服务相关角色支持的区域

Amazon DocumentDB 支持在服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅<https://docs.amazonaws.cn/documentdb/latest/developerguide/regions-and-azs.html#regions-and-azs-availability>。

监控 Amazon DocumentDB

监控Amazon服务是保持系统正常运行并最佳运行的重要组成部分。明智的做法是从你的所有部分收集监控数据Amazon解决方案，以便在发生故障或降级时更容易调试和修复。在开始监控之前Amazon解决方案，我们建议您考虑并制定以下问题的答案：

- 监控目的是什么？
- 您要监控什么资源？
- 您将以什么样的频率监控这些资源？
- 您将使用哪些监控工具？
- 由谁负责执行监控？
- 发生错误时通知谁？以何种方式发送通知？

要了解当前的性能模式，判断性能异常表现，并构想问题解决方法，您应该针对不同时间和不同负载条件设定基准性能指标。当您监视您的Amazon解决方案中，我们建议您存储历史监控数据，以供日后参考，也可帮助您设定基准。

通常，性能指标的可接受值取决于您的基准性能以及应用程序执行的操作。应调查相对于基准性能的一致或趋势性变化。有关特定指标类型的建议如下：

- CPU 或 RAM 使用率过高— 高 CPU 或 RAM 使用较高的 CPU 或 RAM 使用较高的 CPU 或 RAM 使用较高的 CPU 或 RAM 使用较高的 CPU 或 RAM 使用较高的 CPU 或 RAM 使用较高的 CPU
- 存储卷消耗— 调查存储消耗情况 (VolumeBytesUsed) 如果使用的空间始终不低于总存储卷空间的 85%。确定是可以从存储卷中删除数据还是可以将数据存档到其他系统以释放空间。有关更多信息，请参阅 [Amazon DocumentDB D \(p. 9\)](#) 和 [Amazon DocumentDB 配额和限制 \(p. 496\)](#)。
- 网络流量— 对于网络流量，应与系统管理员讨论，以了解域网络和 Internet 连接的预期吞吐量。如果吞吐量始终低于预期，则应调查网络流量。
- 数据库连接— 如果发现用户连接数较高，同时实例性能下降并且响应时间下降，请考虑约束数据库连接。实例的最佳用户连接数因实例类和所执行操作的复杂性而异。
- IOPS 指标— IOPS 指标的预期值取决于磁盘规格和服务器配置，因此，请使用您的基准来了解典型状况。调查一下值是否始终与您的基准不同。为获得最佳 IOPS 性能，请确保典型工作集与内存大小相适，以最大限度地减少读取和写入操作。

Amazon DocumentDB (MongoDB 兼容性) 提供了各种 Amazon DocumentDB CloudWatch 您可监控这些指标以确定 Amazon DocumentDB 集群和实例的运行状况和性能。您可以使用各种工具查看 Amazon DocumentDB 指标，包括 Amazon DocumentDB 控制台，Amazon CLI、CloudWatch API 和性能 Insights。

主题

- [监控 Amazon DocumentDB 集群的状态 \(p. 366\)](#)
- [监控 Amazon DocumentDB 实例的状态 \(p. 368\)](#)
- [使用 Amazon DocumentDB 活动订阅 \(p. 370\)](#)
- [使用 CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)
- [使用记录 Amazon DocumentDB API 调用Amazon CloudTrail \(p. 387\)](#)
- [分析 Amazon DocumentDB 操作 \(p. 388\)](#)
- [使用 Performance Insights 进行监控 \(p. 394\)](#)

监控 Amazon DocumentDB 集群的状态

集群的状态表示集群的运行状况。您可以使用 Amazon DocumentDB 控制台或 Amazon CLI `describe-db-clusters` 命令。

主题

- [集群状态值 \(p. 366\)](#)
- [使用 Amazon Web Services Management Console 监控集群状态 \(p. 366\)](#)
- [使用 Amazon CLI 监控集群状态 \(p. 367\)](#)

集群状态值

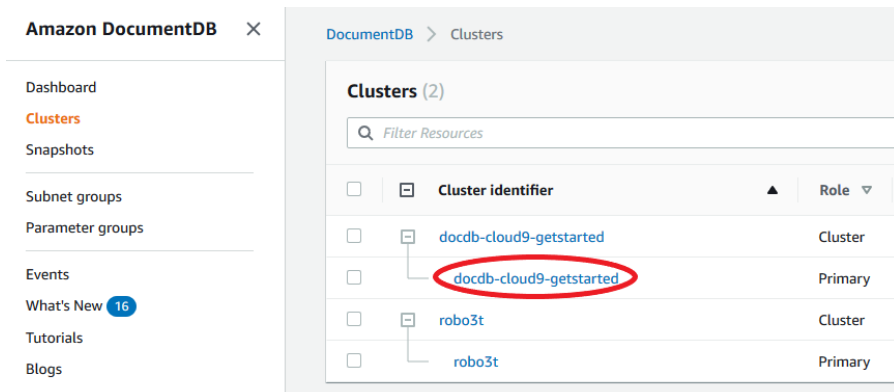
下表列出集群状态的有效值。

集群状态	说明
<code>available</code>	集群状态良好且可用。
<code>backing-up</code>	当前正在备份集群。
<code>creating</code>	正在创建集群。正在创建的过程中无法访问。
<code>deleting</code>	正在删除集群。正在删除的过程中无法访问。
<code>failing-over</code>	正在执行从主实例到 Amazon DocumentDB 副本的故障转移。
<code>inaccessible-encryption-credentials</code>	无法访问用于加密或解密密钥的 Amazon KMS 密钥。
<code>maintenance</code>	正在对集群应用维护更新。此状态用于 Amazon DocumentDB 预先计划的集群级别的维护。
<code>migrating</code>	正将集群快照还原给集群。
<code>migration-failed</code>	迁移失败。
<code>modifying</code>	正在按照客户请求修改集群。
<code>renaming</code>	正在按照客户请求重命名集群。
<code>resetting-master-credentials</code>	正在按照客户请求重置集群的主凭证。
<code>upgrading</code>	集群引擎版本正在升级。

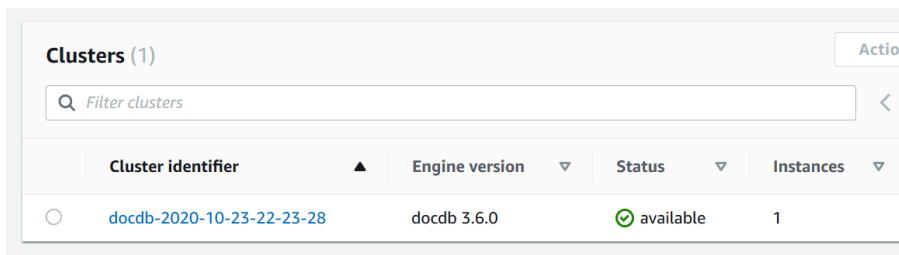
使用 Amazon Web Services Management Console 监控集群状态

在使用 Amazon Web Services Management Console 确定集群状态时，请使用以下过程。

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。
3. 在 Cluster 导航框中，您将看到此列集群标识符。您的实例列在群集下，类似于下面的屏幕截图。



4. 在集群标识符列中，找到您感兴趣的实例名称。然后，要查找实例的状态，请跨该行阅读至状态列，如下所示。



使用Amazon CLI监控集群状态

在使用 Amazon CLI 确定集群的状态时，请使用 `describe-db-clusters` 操作。以下代码可查找集群 `sample-cluster` 的状态。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

对于 Windows：

```
aws docdb describe-db-clusters ^
  --db-cluster-identifier sample-cluster ^
  --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

此操作的输出将类似于下文。

```
[
  [
    "sample-cluster",
    "available"
  ]
]
```


监控 Amazon DocumentDB 实例的状态

实例的状态指示实例的运行状况。您可以使用 Amazon DocumentDB 中查看实例的状态 (具有 MongoDB 兼容性) Amazon Web Services Management Console或者Amazon CLI手术describe-db-instances.

Note

Amazon DocumentDB 还使用另一种名为维护状态，如维护Amazon DocumentDB 控制台的列。此值指示需要应用于实例的任何维护修补程序的状态。维护状态独立于 Amazon DocumentDB 实例状态。有关维护状态的更多信息，请参阅[应用 Amazon DocumentDB 更新 \(p. 357\)](#)。

主题

- [实例状态值 \(p. 368\)](#)
- [使用Amazon Web Services Management Console监控实例的状态 \(p. 369\)](#)
- [使用Amazon CLI监控实例的状态 \(p. 370\)](#)

实例状态值

下表列出实例的可能状态值以及如何对每个状态计费。其中显示是否对实例和存储计费、只对存储向您计费，还是不向您计费。对于所有实例状态，始终会针对备份用量向您计费。

实例状态的有效值

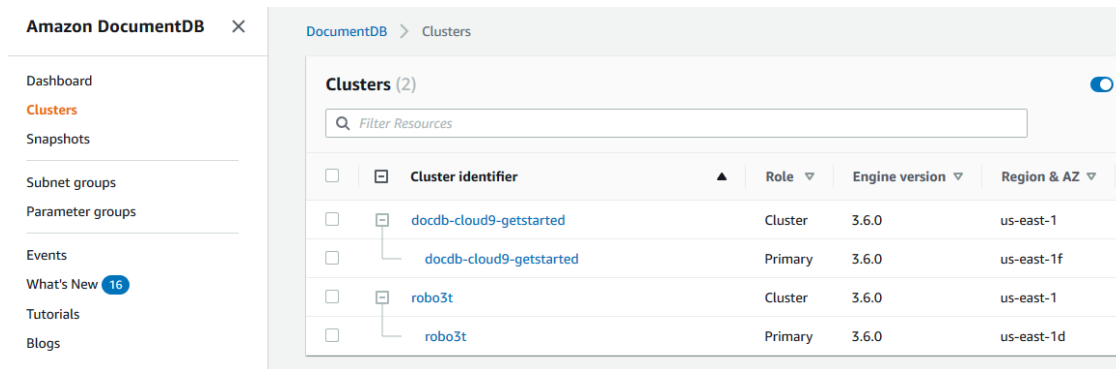
实例状态	已计费	说明
available	已计费	实例正常和可用。
backing-up	已计费	当前正在备份实例。
configuring-log-exports	已计费	向亚马逊发布日志文件 CloudWatch 正在对此实例启用或禁用日志。
creating	不计费	正在创建实例。无法访问正在创建的实例。
deleting	不计费	正在删除实例。
failed	不计费	实例已失败，Amazon DocumentDB 无法恢复它。要恢复数据，请执行 point-in-time 还原至实例的最近可还原时间。
inaccessible-encryption-credentials	不计费	无法访问用于加密或解密实例的 Amazon KMS 密钥。
incompatible-network	不计费	Amazon DocumentDB 正尝试恢复实例，但无法执行此操作，因为 VPC 正处于一种阻止此操作完成的状态。例如，如果子网中的所有可用 IP 地址都在使用中，并且 Amazon DocumentDB 无法为实例获取 IP 地址，就会出现此状态。
maintenance	已计费	Amazon DocumentDB 正在对实例应用维护更新。此状态用于 Amazon DocumentDB 预先计划的实例级别的维护。我们将通过此状态评估向客户公开其他维护操作的方式。
modifying	已计费	按照请求正在修改实例。

实例状态	已计费	说明
rebooting	已计费	正在重启实例，因为需要重启实例的请求或 Amazon DocumentDB 过程正在重启实例。
renaming	已计费	按照请求正在重命名实例。
resetting-master-credentials	已计费	按照请求正在重置实例的主凭证。
restore-error	已计费	实例在尝试还原到 point-in-time 或从快照。
starting	对存储计费	实例正在启动。
stopped	对存储计费	实例已停止。
stopping	对存储计费	正在停止实例。
storage-full	已计费	实例超出了其存储分配容量。这是一种严重状态，应立即修复；请通过修改实例来扩展存储。设置 Amazon Amazon CloudWatch 当存储空间逐渐减小时发送警报，以避免发生这种情况。

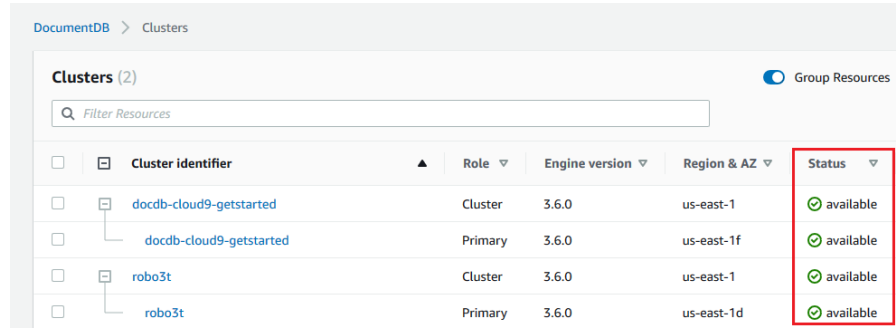
使用Amazon Web Services Management Console监控实例的状态

在使用Amazon CLI确定集群状态时，请使用以下过程。

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。
3. 请注意，在“群集”导航框中，列Cluster Identifier同时显示集群和实例。实例列在群集下方，类似于下面的屏幕截图。



4. 找到您感兴趣的实例名称。然后，要查找实例的状态，请跨该行阅读至 Status (状态) 列，如下所示。



The screenshot shows the 'Clusters' page in the Amazon DocumentDB console. It displays a table with columns: Cluster identifier, Role, Engine version, Region & AZ, and Status. There are four rows of data, all with a status of 'available' indicated by a green checkmark icon. The 'Status' column is highlighted with a red box.

Cluster identifier	Role	Engine version	Region & AZ	Status
docdb-cloud9-getstarted	Cluster	3.6.0	us-east-1	available
docdb-cloud9-getstarted	Primary	3.6.0	us-east-1f	available
robo3t	Cluster	3.6.0	us-east-1	available
robo3t	Primary	3.6.0	us-east-1d	available

使用Amazon CLI监控实例的状态

在使用 Amazon CLI 确定集群的状态时，请使用 `describe-db-instances` 操作。以下代码可查找实例 `sample-cluster-instance-01` 的状态。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance-01 \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

对于 Windows：

```
aws docdb describe-db-instances ^
    --db-instance-identifier sample-cluster-instance-01 ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

此操作的输出将类似于下文。

```
[
  [
    "sample-cluster-instance-01",
    "available"
  ]
]
```

使用 Amazon DocumentDB 活动订阅

Amazon DocumentDB 使用 Amazon Simple Notification Service (Amazon SNS) 在发生 Amazon DocumentDB 事件时提供通知。这些通知可以采用 Amazon SNS 支持的任何形式进行 Amazon Web Services 区域，例如电子邮件、短信或调用 HTTP 终端节点。

Amazon DocumentDB 将这些事件分组为您可以订阅的类别，以便您在该类别事件出现时收取通知。您可以针对实例、集群、快照、集群快照或者参数组订阅事件类别。例如，如果您针对指定实例订阅了 Backup 类别，那么无论何时出现影响该实例的与备份相关的事件，您都将收到通知。您还将在事件订阅更改时收到通知。

事件在集群和实例级别发生。所以，如果您针对集群或实例进行订阅，将可收到事件。

事件订阅会发送到您在创建订阅时提供的地址。您可能希望创建多个不同的订阅，如使用一个订阅接收所有事件通知，并使用另一个订阅接收仅针对生产实例的关键事件。您可以在不删除订阅的情况下轻松关闭通知。为此，请将 Enabled (已启用) 单选按钮否在 Amazon DocumentDB 控制台中。

Important

Amazon DocumentDB 不保证在事件流中发送的事件的顺序。事件顺序可能会发生变化。

Amazon DocumentDB 使用 Amazon SNS 主题的 Amazon 资源名称 (ARN) 标识每个订阅。Amazon DocumentDB 控制台在您创建订阅时为您创建 ARN。

Amazon DocumentDB 活动订阅的账单是通过 Amazon SNS 进行的。Amazon SNS 费用在使用事件通知时适用。有关更多信息，请参阅 Amazon Simple Notification Service 定价。除了 Amazon SNS 费用外，Amazon DocumentDB 不收取活动订阅费用。

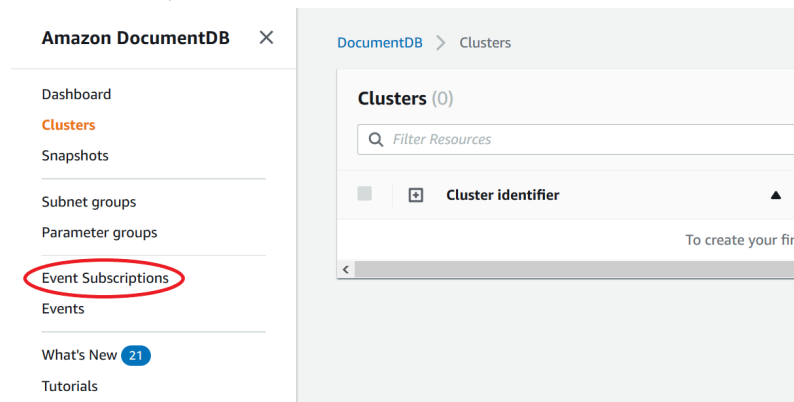
主题

- [订阅 Amazon DocumentDB 事件订阅 \(p. 371\)](#)
- [管理 Amazon DocumentDB 事件通知订阅 \(p. 372\)](#)
- [Amazon DocumentDB 事件类别和消息 \(p. 374\)](#)

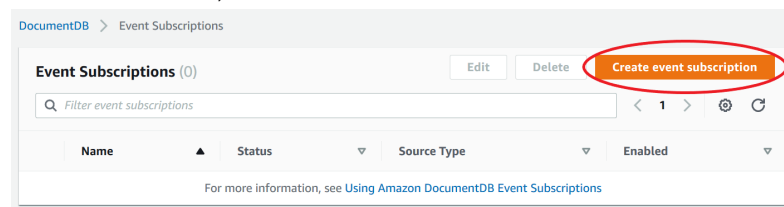
订阅 Amazon DocumentDB 事件订阅

您可以使用 Amazon DocumentDB 控制台订阅活动订阅，如下所示：

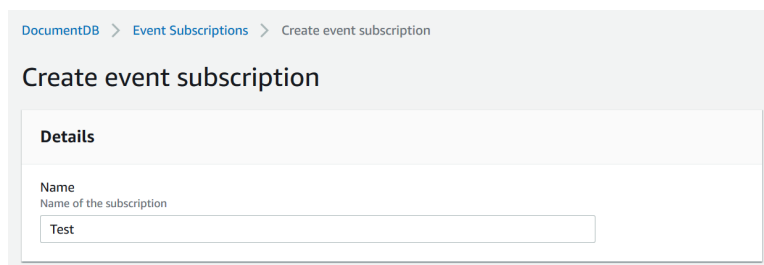
1. 在 <https://console.amazonaws.cn/docdb> 处登录 Amazon Web Services Management Console。
2. 在导航窗格中，选择事件订阅。



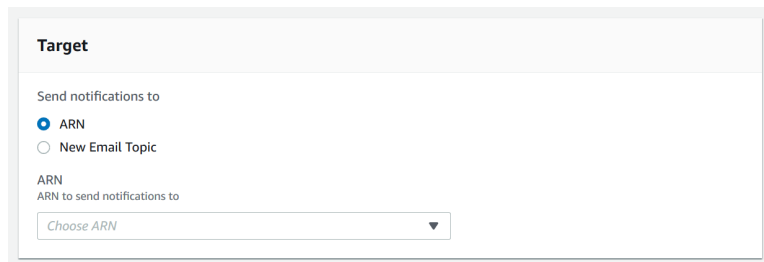
3. 在事件订阅窗格中，选择创建事件订阅。



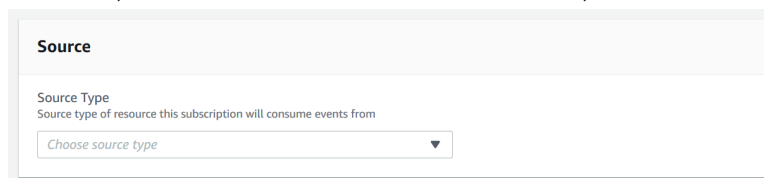
4. 在创建事件订阅对话框中，请执行以下操作：
 - 对于名称，输入事件通知订阅的名称。



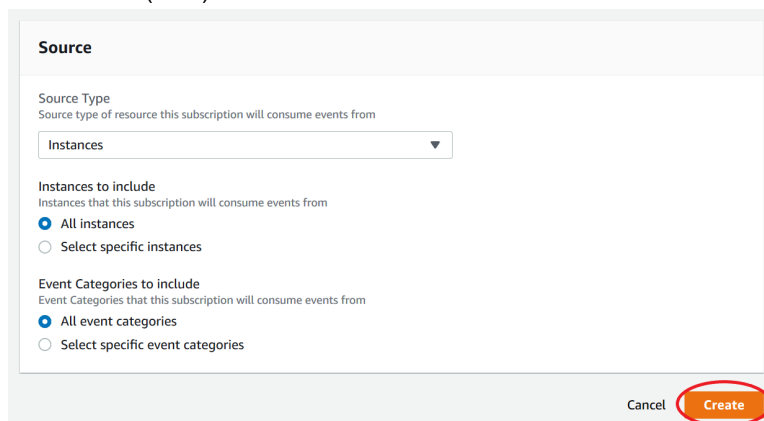
- 适用于目标中，选择要将通知发送到的位置。您可以选择现有进行筛选或者选择新电子邮件主题输入主题的名称和收件人列表。



- 适用于源，选择一种源类型。根据选定源类型的情况，选择要接收事件通知的事件类别和源。



- 选择 Create (创建)。

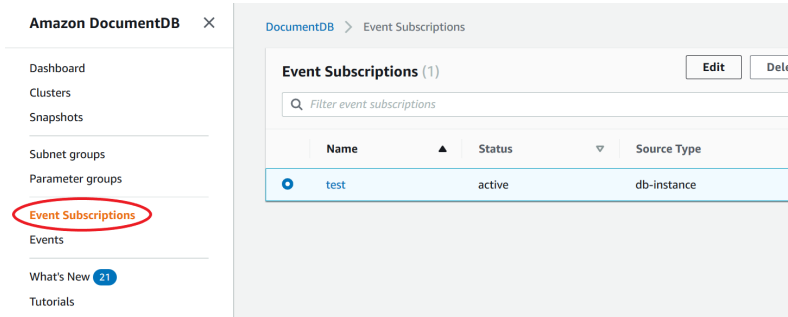


管理 Amazon DocumentDB 事件通知订阅

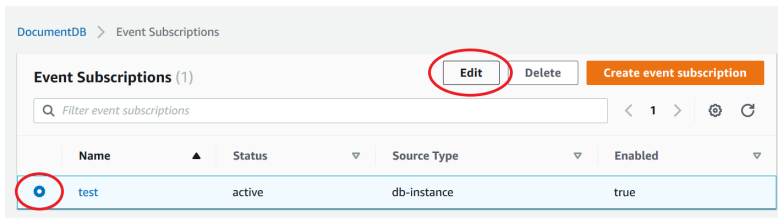
如果选择事件订阅在 Amazon DocumentDB 控制台的导航窗格中，您可以查看订阅类别和当前订阅的列表。您还可以修改或删除特定订阅。

修改当前的 Amazon DocumentDB 事件通知订阅

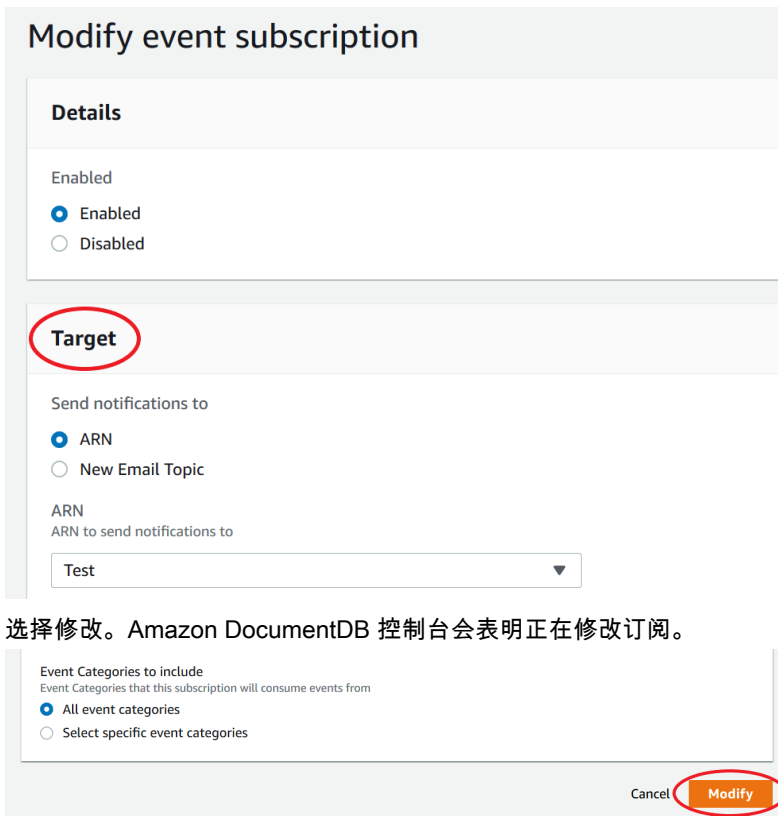
1. 在 <https://console.amazonaws.cn/docdb> 处登录 Amazon Web Services Management Console。
2. 在导航窗格中，选择事件订阅。事件订阅窗格中会显示您的所有事件通知订阅。



3. 在事件订阅窗格中，选择您要修改的订阅，然后选择编辑。



4. 在目标或来源部分中对订阅进行更改。您可以通过在源部分中选择或者取消选择操作来添加或删除源标识符。



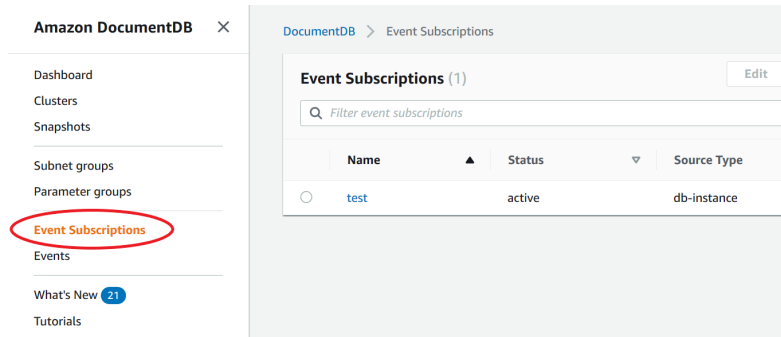
5. 选择修改。Amazon DocumentDB 控制台会表明正在修改订阅。

删除 Amazon DocumentDB 事件通知订阅

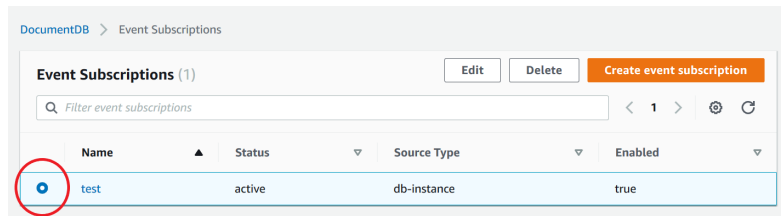
当您不再需要时，可以删除订阅。该主题的所有用户都将再也不会收到订阅指定的事件通知。

1. 在 <https://console.amazonaws.cn/docdb> 处登录 Amazon Web Services Management Console。

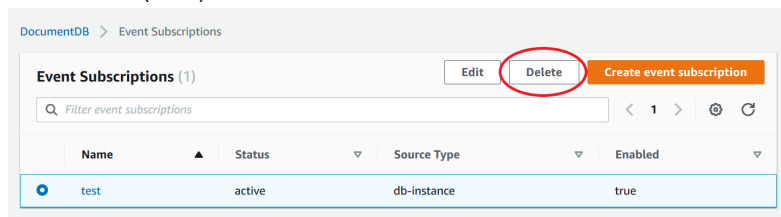
2. 在导航窗格中，选择事件订阅。



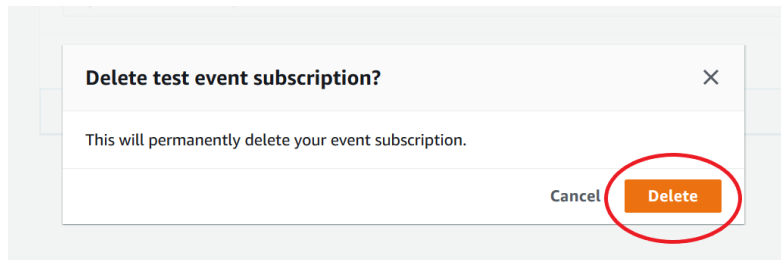
3. 在事件订阅窗格中，选择要删除的订阅。



4. 选择 Delete (删除)。



5. 将出现一个弹出窗口，询问您是否要永久删除此通知。选择 Delete (删除)。



Amazon DocumentDB 事件类别和消息

Amazon DocumentDB 会在各种类型中生成许多事件，您可以使用控制台对它们进行订阅。每个类别应用于一种源类型，可以是实例、快照或者参数组。

Note

Amazon DocumentDB 使用现有 Amazon RDS 事件定义和 ID。

源自实例的 Amazon DocumentDB 事件

类别	描述
可用性	实例已重启。

类别	描述
可用性	实例已关闭。
配置更改	将修改应用于实例类。
配置更改	已完成对实例类应用修改。
配置更改	重置主凭据。
创建	已创建实例。
删除	删除实例
失败	实例由于不兼容配置或底层存储问题而失败。开始对实例进行时间点还原。
通知	实例已停止。
通知	实例已启动。
通知	实例将由于它超过最大允许停止的时间而正被启动。
恢复	已启动实例的恢复。恢复时间会随待恢复数据量的变化而变化。
恢复	已完成实例的恢复。

源自群集的 Amazon DocumentDB 事件

类别	描述
创建	已创建集群
删除	已删除集群。
故障转移	再次推广以前的主。
故障转移	已完成实例故障转移。
故障转移	开始故障转移到数据库实例：%s
故障转移	已开始对数据库实例的相同 AZ 故障转移
故障转移	已开始到数据库实例的跨可用区故障转移
maintenance	集群已修补。
maintenance	数据库集群处于无法升级的状态：%s
通知	已停止集群。
通知	已启动集群。
通知	集群停止失败。
通知	由于它超过最大允许停止的时间而正被启动。
通知	已将集群从 %s 重命名为 %s。

源自群集快照的 Amazon DocumentDB 事件

下表显示了 Amazon DocumentDB 群集快照为源类型时的事件类别和事件列表。

类别	描述
backup	创建手动群集快照。
backup	已创建手动群集快照。
backup	正在创建自动群集快照。
backup	已创建自动群集快照。

源自参数组的 Amazon DocumentDB 事件

下表显示的是参数组为源类型时的事件类型和事件列表。

类别	描述
配置更改	使用应用方法 %s 将参数 %s 更新为 %s

使用 CloudWatch 监控 Amazon DocumentDB

Amazon DocumentDB (兼容 MongoDB) 与 Amazon CloudWatch 集成, 因此您可以为您的群集收集和分析运行指标。您可以使用 CloudWatch 控制台、Amazon DocumentDB 控制台和 Amazon Command Line Interface (Amazon CLI) 或者 CloudWatch API。

CloudWatch 还让您设置警报, 以便在指标值超过您指定的阈值时, 您可以收到通知。您甚至可以设置 Amazon CloudWatch Events, 以便在发生违规情况时采取纠正措施。有关使用 CloudWatch 和警报的更多信息, 请参阅 [Amazon CloudWatch 文档](#)。

主题

- [Amazon DocumentDB 指标 \(p. 376\)](#)
- [查看 CloudWatch 数据 \(p. 382\)](#)
- [Amazon DocumentDB 维度 \(p. 386\)](#)
- [监控 Opcount \(p. 386\)](#)
- [监控数据库连接 \(p. 387\)](#)

Amazon DocumentDB 指标

要监控 Amazon DocumentDB 群集和实例的运行状况和性能, 您可以在 Amazon DocumentDB 控制台中查看以下指标。

资源利用率

指标	描述	
BackupRetentionPeriodStorageUsed	用于支持 Amazon DocumentDB 保留时段内的时间点还原功能的备份存储总量 (以	

指标	描述	
	GiB 为单位)。包含在 TotalBackupStorageBilled 指标报告的总数中。针对每个 Amazon DocumentDB 集群单独计算。	
ChangeStreamLogSize	集群用于存储变更流日志的存储量 (以兆字节为单位)。此值是集群总存储量的子集 (VolumeBytesUsed), 将影响集群的成本。有关存储定价的信息, 请参阅 Amazon DocumentDB 产品页面 。变更流日志大小取决于集群上发生了多少更改以及变更流日志的保留时间。有关变更流的更多信息, 请参阅在 Amazon DocumentDB 中使用变更流 (p. 445)。	
CPUUtilization	实例占用的 CPU 百分比。	
DatabaseConnections	每 1 分钟获取的实例上打开的连接数。	
DatabaseConnectionsMax	在一分钟内, 实例上的最大开放数据库连接数。	
DatabaseCursors	每 1 分钟打开一次的实例上的游标数。	
DatabaseCursorsMax	一分钟内实例上打开的游标的最大数量。	
DatabaseCursorsTimedOut	在一分钟内超时的游标数。	
FreeableMemory	随机存取内存的可用量 (以字节为单位)。	
FreeLocalStorage	此指标报告每个实例中可用于临时表和日志的存储量。此值取决于实例类。您可通过为实例选择较大的实例类来增加对实例可用的存储空间量。	
LowMemThrottleQueueDepth	由于以一分钟频率占用的可用内存不足而受限的请求的队列深度。	
LowMemThrottleMaxQueueDepth	一分钟内由于可用内存不足而受限的请求的最大队列深度。	
LowMemNumOperationsThrottled	在一分钟内, 由于可用内存不足而受到限制的请求数。	

指标	描述	
SnapshotStorageUsed	给定 Amazon DocumentDB 集群的所有快照在其备份保留时段外消耗的备份存储总量（以 GiB 为单位）。包含在 TotalBackupStorageBilled 指标报告的总数中。针对每个 Amazon DocumentDB 集群单独计算。	
SwapUsage	实例上使用的交换空间的大小。	
TotalBackupStorageBilled	为给定 Amazon DocumentDB 集群计费时所针对的备份存储总量（以 GiB 为单位）。包含由 BackupRetentionPeriodStorageUsed 和 SnapshotStorageUsed 指标度量的备份存储。针对每个 Amazon DocumentDB 集群单独计算。	
TransactionsOpen	每 1 分钟的频率在实例上打开的交易数。	
TransactionsOpenMax	在一分钟内实例上打开的最大交易数。	
VolumeBytesUsed	您的集群使用的存储量（以字节为单位）。此值将影响集群的成本。有关定价信息，请参阅 Amazon DocumentDB 产品页面 。	

延迟

指标	描述	
DBClusterReplicaLagMaximum	集群中主实例和每个 Amazon DocumentDB 实例之间的最大滞后量（以毫秒为单位）。	
DBClusterReplicaLagMinimum	集群中主实例和每个副本实例之间的最小滞后量（以毫秒为单位）。	
DBInstanceReplicaLag	在从主实例向副本实例复制更新时的滞后总量（以毫秒为单位）。	
ReadLatency	每个磁盘 I/O 操作所需的平均时间。	
WriteLatency	每个磁盘 I/O 操作所需的平均时间（以毫秒为单位）。	

操作

指标	描述	
DocumentsDeleted	一分钟内删除的文档的数量。	
DocumentsInserted	一分钟时间内插入的文档的数量。	
DocumentsReturned	一分钟内返回的文件数量。	
DocumentsUpdated	一分钟内更新的文档的数量。	
OpcountersCommand	在一分钟内发出的命令数。	
OpcountersDelete	在一分钟内发出的删除操作的数量。	
OpcountersGetmore	在一分钟内发行的 getmores 的数量。	
OpcountersInsert	在一分钟时间内发出的插入操作的数量。	
OpcountersQuery	在一分钟内发出的查询数。	
OpcountersUpdate	在一分钟内发布的更新操作的数量。	
TransactionsStarted	在一分钟内在实例上开始的交易数。	
TransactionsCommitted	在一分钟内在实例上提交的交易数。	
TransactionsAborted	在一分钟内实例中止的交易数。	
TTLDeletedDocuments	TTLMonitor 在一分钟内删除的文档数。	

吞吐量

指标	描述	
NetworkReceiveThroughput	集群中每个实例从客户端接收的网络吞吐量（以每秒字节数为单位）。此吞吐量不包括集群中的实例与集群卷之间的网络流量。	
NetworkThroughput	Amazon DocumentDB 集群中每个实例从客户端接收和传输到客户端的网络吞吐量（以每秒字节数为单位）。此吞吐量不包括集群中的实例与集群卷之间的网络流量。	
NetworkTransmitThroughput	集群中每个实例发送到客户端的网络吞吐量（以每秒字节数为单位）。此吞吐量不包括集群中的实例与集群卷之间的网络流量。	

指标	描述	
ReadIOPS	每秒平均磁盘读取 I/O 操作数。Amazon DocumentDB 每分钟分别报告一次读取和写入 IOPS。	
ReadThroughput	每秒从磁盘读取的平均字节数。	
VolumeReadIOPS	集群卷中计费读取 I/O 操作的平均数量，每隔 5 分钟报告一次。计费读取操作数是在集群卷级别计算的，由集群中的所有实例聚合而来，然后每隔 5 分钟报告一次。此值是通过采用 5 分钟以上的读取操作数指标的值计算得来的。您可通过采用计费读取操作数指标的值并除以 300 秒来确定每秒的计费读取操作数。 例如，如果 <code>VolumeReadIOPS</code> 返回 13686，然后每秒的计费读取操作数为 $45 (13686/300 = 45.62)$。 您累积请求不在缓冲区缓存中因而必须从存储加载的数据库页的查询的计费读取操作数。您可能看到计费读取操作数出现峰值，因为查询结果是从存储中读取然后加载到缓冲区缓存中的。	

指标	描述	
VolumeWriteIOPs	集群卷中计费写入 I/O 操作的平均数量，每隔 5 分钟报告一次。计费写入操作数是在集群卷级别计算的，由集群中的所有实例聚合而来，然后每隔 5 分钟报告一次。此值是通过采用 5 分钟以上的写入操作数指标的值计算得来的。您可通过采用计费写入操作数指标的值并除以 300 秒来确定每秒的计费写入操作数。 例如，如果 VolumeWriteIOPs 返回 13686，然后每秒的计费写入操作数为 45 ($13686/300 = 45.62$)。 请注意，VolumeReadIOPs 和 VolumeWriteIOPs 指标由 DocumentDB 存储层计算，其中包括主实例和副本实例执行的 iOS。每 20-30 分钟汇总一次数据，然后每隔 5 分钟报告一次，从而在该时间段内为该指标发出相同的数据点。如果您正在寻找在 1 分钟间隔内与插入操作关联的指标，则可以使用实例级 WriteIOPS 指标。该指标可在 Amazon DocumentDB 主实例的监控选项卡中找到。	
WriteIOPS	每秒平均磁盘写入 I/O 操作数。在群集级别使用时，WriteIOPS 在群集中的所有实例中进行评估。每分钟分别报告一次读取和写入 IOPS。	
WriteThroughput	每秒写入磁盘的平均字节数。	

系统

指标	描述	
BufferCacheHitRatio	缓冲区缓存提供的请求的百分比。	
DiskQueueDepth	对分布式存储卷的并发写入请求的数量。	
EngineUptime	实例已运行的时间长度（以秒为单位）。	
IndexBufferCacheHitRatio	缓冲区缓存提供的索引请求的百分比。删除索引、集合或数据库后，您可能会看到指标的峰值超过 100%。这将在 60 秒后自动更	

指标	描述	
	正。将来的补丁更新中将修复这一限制。	

T3 实例指标

指标	描述	
CPUCreditUsage	在测量周期内花费的 CPU 积分数。	
CPUCreditBalance	实例产生的 CPU 积分数量。在 CPU 突增以及 CPU 积分的花费速度比获得速度快时，该余额将用完。	
CPUSurplusCreditBalance	在 CPUCreditBalance 值为零时，用于保持 CPU 性能而花费的超额 CPU 积分数。	
CPUSurplusCreditsCharged	超过可在 24 小时内获得的 CPU 积分数上限的超额 CPU 积分数，因而会产生额外的费用。有关更多信息，请参阅 监控 CPU 积分 。	

查看 CloudWatch 数据

您可以使用 Amazon CloudWatch 控制台、Amazon DocumentDB 控制台、Amazon Command Line Interface(Amazon CLI) 或者 CloudWatch API。

使用 Amazon DocumentDB 管理控制台查看 CloudWatch 指标

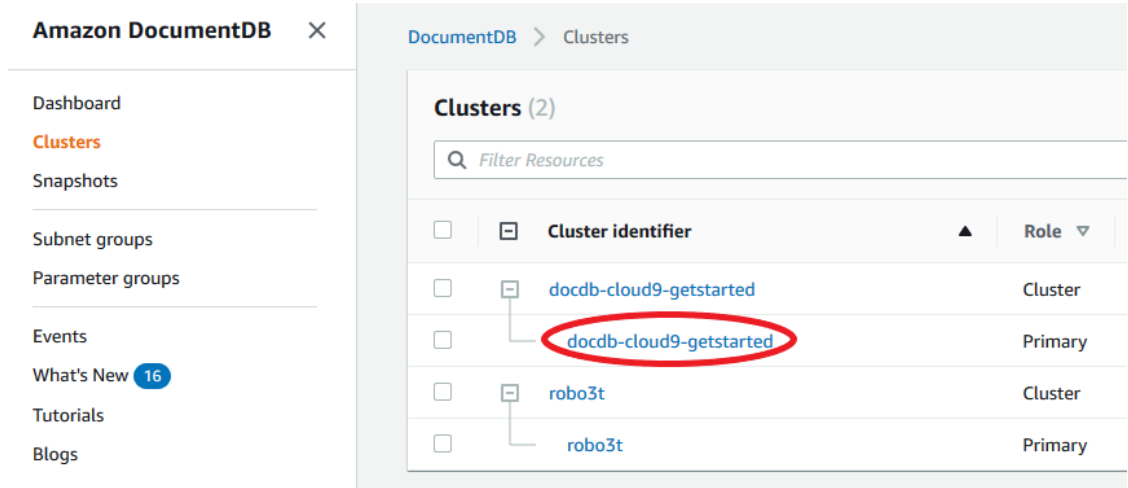
要使用 Amazon DocumentDB 管理控制台查看 CloudWatch 指标，请完成以下步骤。

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。

Tip

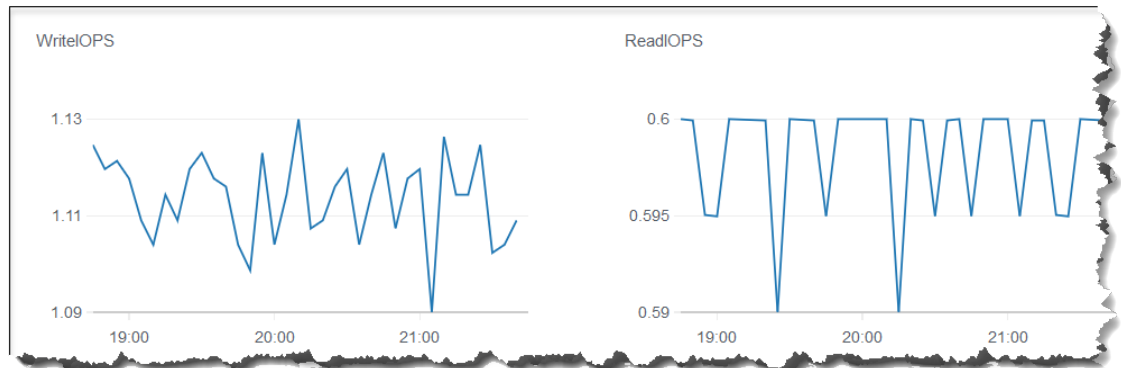
如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标 (≡)。

3. 在 Cluster 导航框中，您将看到该列Cluster Identifier. 您的实例列在群集下，类似于下面的屏幕截图。



4. 从实例列表中，选择要获取其指标的实例的名称。
5. 在生成的实例摘要页面中，选择监控选项卡以查看 Amazon DocumentDB 实例指标的图形表示形式。由于必须为每个指标生成图表，可能需要一些几分钟时间才能填充 CloudWatch 图表。

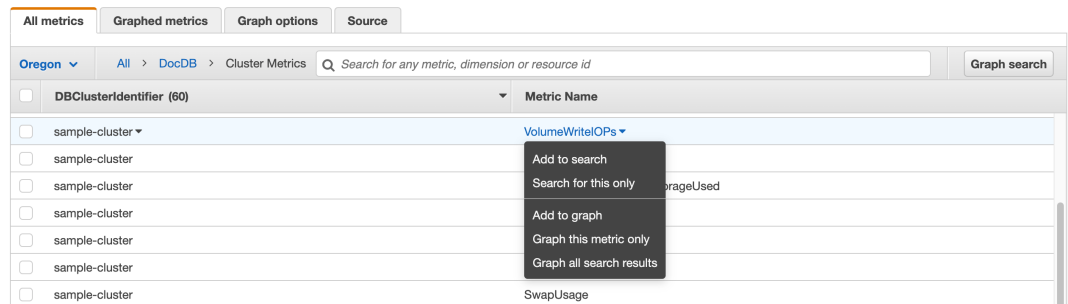
下图显示了 Amazon DocumentDB 控制台中两个 CloudWatch 指标的图形表示方式，WriteIOPS 和 ReadIOPS。



使用 CloudWatch 管理控制台查看 CloudWatch 指标

要使用 CloudWatch 管理控制台查看 CloudWatch 指标，请完成以下步骤。

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.amazonaws.cn/cloudwatch>。
2. 在导航窗格中，选择 Metrics (指标)。然后，从服务名称列表中选择 DocDB。
3. 选择指标维度 (例如，集群指标)。
4. 这些区域有：所有指标选项卡显示中该维度的所有指标 DocDB。
 - a. 要对表进行排序，请使用列标题。
 - b. 要为指标绘制图表，请选中该指标旁的复选框。要选择所有指标，请选中表的标题行中的复选框。
 - c. 要按指标进行筛选，请将鼠标悬停在指标名称上，然后选择指标名称旁边的下拉箭头。然后，选择添加到搜索，如下图所示。



使用 Amazon CLI

要查看 Amazon DocumentDB 的 CloudWatch 数据，请使用 `CloudWatchget-metric-statistics` 按照以下参数进行操作。

参数

- **--namespace** – 必需。需要 CloudWatch 指标的服务命名空间。对于 Amazon DocumentDB，这必须是 `AWS/DocDB`。
- **--metric-name** – 必需。需要其数据的指标的名称。
- **--start-time** – 必需。用于确定要返回的第一个数据点的时间戳。

包含指定的值；结果包括具有指定时间戳的数据点。时间戳必须采用 ISO 8601 UTC 格式（例如，2016-10-03T23:00:00Z）。

- **--end-time** – 必需。用于确定要返回的最后一个数据点的时间戳。

包含指定的值；结果包括具有指定时间戳的数据点。时间戳必须采用 ISO 8601 UTC 格式（例如，2016-10-03T23:00:00Z）。

- **--period** – 必需。返回的数据点的粒度（以秒为单位）。对于具有常规精度的指标，期间可以短到一分钟（60 秒），并且必须为 60 的倍数。对于以小于一分钟的间隔收集的高精度指标，期间可以是 1、5、10、30、60 或 60 的任意倍数。
- **--dimensions** – 可选。如果指标包含多个维度，则必须为每个维度包含一个值。CloudWatch 将不同维度的每种唯一组合视为一个单独的指标。如果未发布某个特定的维度组合，则无法检索该组合的统计数据。您必须指定创建指标时使用的同一维度。
- **--statistics** – 可选。百分位数之外的指标统计数据。对于百分位数统计数据，请使用 `ExtendedStatistics`。调用 `GetMetricStatistics` 时，必须指定 `Statistics` 或 `ExtendedStatistics`，但不能同时指定两者。

允许的值：

- `SampleCount`
- `Average`
- `Sum`
- `Minimum`
- `Maximum`
- **--extended-statistics** – 可选。percentile 统计数据。指定介于 p0.0 到 p100 之间的值。调用 `GetMetricStatistics` 时，必须指定 `Statistics` 或 `ExtendedStatistics`，但不能同时指定两者。
- **--unit** – 可选。给定指标的单位。可以用多个单位报告指标。如果不提供单位，将返回所有单位。如果您仅指定指标不报告的单位，调用的结果将为空。

可能的值：

- 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

以下示例查找 2 小时时段的最大 CPUUtilization，每隔 60 秒采样一次。

对于 Linux、macOS 或 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
```

对于 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
```

此操作的输出类似于以下内容：

```
{
  "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"
    }
  ]
}
```

Amazon DocumentDB 维度

Amazon DocumentDB 的指标由账户或操作的值来限定。您可以使用 CloudWatch 控制台来检索由下表中的任意维度筛选的 Amazon DocumentDB 数据。

维度	描述
DBClusterIdentifier	筛选您为特定 Amazon DocumentDB 集群请求的数据。
DBClusterIdentifier, Role	筛选您为特定 Amazon DocumentDB 集群请求的数据，并按实例角色 (WRITER/READER) 聚合指标。例如，您可以聚合属于某个群集的所有 READER 实例的指标。
DBInstanceIdentifier	筛选您为特定数据库实例请求的数据。

监控 Opcount

对于空闲集群，Opcount 指标具有非零值（通常为 50）。这是因为 Amazon DocumentDB 会定期执行运行状况检查、内部操作和指标收集任务。

监控数据库连接

使用例如数据库引擎命令查看连接数时，`db.runCommand( { serverStatus: 1 })`，您最多可能会看到比中看到的数量多 10 个的连接数。`DatabaseConnections`通过 CloudWatch。发生这种情况的原因是，Amazon DocumentDB 执行定期运行状况检查和指标收集任务，而这些任务不记入`DatabaseConnections.DatabaseConnections`仅代表客户发起的连接。

使用记录 Amazon DocumentDB API 调用 Amazon CloudTrail

Amazon DocumentDB (与 MongoDB 兼容) 与 Amazon CloudTrail，是提供 IAM 用户、IAM 角色或所采取操作的记录的服务。Amazon Amazon DocumentDB 中的服务 (与 MongoDB 兼容)。CloudTrail 捕获所有 Amazon CLI 将 Amazon DocumentDB 作为事件捕获，包括来自 Amazon DocumentDB 控制台的调用和对 Amazon DocumentDB SDK 的代码调用。如果您创建了一个跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶 (包括 Amazon DocumentDB 的事件)。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history (事件历史记录) 中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 Amazon DocumentDB 发出了什么请求 (与 MongoDB 兼容)、从中发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

Important

对于某些管理功能，Amazon DocumentDB 使用与 Amazon Relational Database Service (Amazon RDS) 共享的操作技术。Amazon DocumentDB 控制台，Amazon CLI，并将记录为对 Amazon RDS API 的调用。

要了解有关 Amazon CloudTrail 的更多信息，请参阅 [Amazon CloudTrail 用户指南](#)。

CloudTrail 中的 Amazon DocumentDB 信息

在您创建 Amazon Web Services 账户时，将在该账户上启用 CloudTrail。当 Amazon DocumentDB 中发生活动 (与 MongoDB 兼容) 时，该活动将记录在 CloudTrail 事件中，并与其他事件一起保存在其他事件中。Amazon 中的服务事件记录。您可以在中查看、搜索和下载最新事件 Amazon Web Services 账户。有关更多信息，请参阅 [使用 CloudTrail 事件历史记录查看事件](#)。

要持续记录中的事件 Amazon Web Services 账户，包括 Amazon DocumentDB 的事件 (与 MongoDB 兼容) 创建跟踪。通过跟踪，CloudTrail 可将日志文件传送到 Amazon S3 Bucket。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 Amazon Web Services 区域。此跟踪记录在 Amazon 分区中记录所有区域中的事件，并将日志文件传送到您指定的 Amazon S3 Bucket。此外，您可以配置其他 Amazon 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅 Amazon CloudTrail 用户指南 中的以下主题：

- [创建跟踪记录概述](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [从多个区域接收 CloudTrail 日志文件](#)
- [从多个账户接收 CloudTrail 日志文件](#)

每个事件或日志条目都包含有关生成请求的人员的信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 IAM 用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。

- 请求是否由其他 Amazon 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

分析 Amazon DocumentDB 操作

您可以使用 Amazon DocumentDB 中的分析器（兼容 MongoDB）中的分析器来记录在您集群上执行的操作的执行时间和详细信息。对于监控集群上速度最慢的操作以帮助您提高单个查询的性能和整体集群性能，分析器非常有用。

默认情况下，分析器功能处于禁用状态。在启用时，分析器将用时超过客户定义阈值（例如，100 毫秒）的操作记录到 Amazon CloudWatch Logs 中。记录的详细信息包括分析的命令、时间、计划摘要和客户端元数据。在将操作记录到 CloudWatch Logs 中之后，您可以使用 CloudWatch Logs Insights 来分析、监控和存档 Amazon DocumentDB 分析数据。[常见查询 \(p. 394\)](#)部分中提供了常见的查询。

在启用时，分析器会使用集群中的其他资源。我们建议您从较高的阈值（例如，500 毫秒）开始，然后逐步降低该值以确定缓慢的操作。对于高吞吐量应用程序，从 50 毫秒阈值开始会导致集群性能问题。分析器在集群级别启用，并对集群中的所有实例和数据库执行分析。Amazon DocumentDB 尽最大努力将操作记录到 Amazon CloudWatch Logs 中。

尽管 Amazon DocumentDB 不会对启用分析器施加任何额外的费用，但您需要向使用 CloudWatch Logs 收取的标准费率。有关 CloudWatch Logs 定价的信息，请参阅[Amazon CloudWatch 定价](#)。

主题

- [支持的 操作 \(p. 388\)](#)
- [限制 \(p. 388\)](#)
- [启用 Amazon DocumentDB 档案器 \(p. 389\)](#)
- [禁用 Amazon DocumentDB 档案器 \(p. 391\)](#)
- [禁用分析器日志导出 \(p. 392\)](#)
- [访问您的 Amazon DocumentDB 档案器日志 \(p. 394\)](#)
- [常见查询 \(p. 394\)](#)

支持的 操作

Amazon DocumentDB 分析器支持以下操作：

- aggregate
- count
- delete
- distinct
- find (OP_QUERY 和命令)
- findAndModify
- insert
- update

限制

只有在查询的整个结果集都能容纳在一个批处理中，且结果集低于 16MB（最大 BSON 大小）时，慢查询分析器才能发出分析器日志。大于 16MB 的结果集会自动拆分为多个批次。

大多数驱动程序或 shell 可能会设置较小的默认批处理大小。您可以在查询中指定批量大小。为了捕获慢查询日志，我们建议使用超过预期结果集大小的批处理大小。如果您不确定结果集大小，或者如果不同，您也可以将批处理大小设置为较大的数字（例如 100k）。

但是，使用较大的批处理大小意味着在向客户端发送响应之前，必须从数据库中检索更多结果。对于某些查询，这可能会在获得结果之前造成更长的延迟。如果您不打算使用整个结果集，则可能会花更多的 I/O 来处理查询并丢弃结果。

启用 Amazon DocumentDB 档案器

在集群上启用分析器的过程包含三个步骤。确保所有步骤均完成，否则分析日志不会发送到 CloudWatch Logs。分析器在集群级别设置，对集群的所有数据库和实例执行分析。

在集群上启用分析器

1. 由于您无法修改默认集群参数组，请确保您有可用的自定义集群参数组。有关更多信息，请参阅[创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#)。
2. 使用可用的自定义集群参数组，修改以下参数：`profiler`、`profiler_threshold_ms` 和 `profiler_sampling_rate`。有关更多信息，请参阅[修改 Amazon DocumentDB 集群参数组 \(p. 328\)](#)。
3. 创建或修改集群以使用自定义集群参数组并启用导出 `profiler` 将日志发布到 CloudWatch Logs。

以下部分说明如何使用 Amazon Web Services Management Console 和 Amazon Command Line Interface (Amazon CLI) 实施这些步骤。

使用 Amazon Web Services Management Console

1. 在开始之前，请创建 Amazon DocumentDB 集群和一个自定义集群参数组（如果您没有）。有关更多信息，请参阅[创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#) 和 [创建 Amazon DocumentDB 集群 \(p. 247\)](#)。
2. 使用可用的自定义集群参数组，修改以下参数。有关更多信息，请参阅[修改 Amazon DocumentDB 集群参数组 \(p. 328\)](#)。
 - `profiler`— 启用或禁用查询分析。允许值为 `enabled` 和 `disabled`。默认值为 `disabled`。要启用分析，请将值设置为 `enabled`。
 - `profiler_threshold_ms`— 当 `profiler` 设置为 `enabled`，用时超过的所有命令 `profiler-threshold-ms` 已登录到 CloudWatch。允许值为 `[50-INT_MAX]`。默认值为 100。
 - `profiler_sampling_rate`— 应该分析或记录的缓慢操作的部分。允许值为 `[0.0-1.0]`。默认值为 1.0。
3. 修改集群以使用自定义集群参数组，并将分析器日志导出设置为发布到 Amazon CloudWatch Logs。
 - a. 在导航窗格中，选择 Clusters (集群) 以将自定义参数组添加到集群。
 - b. 选择要与您的参数组关联的集群名称左边的按钮。选择 Actions (操作)，然后选择 Modify (修改) 以修改您的集群。
 - c. 在 Cluster options (集群选项) 下，选择上一步中的自定义参数组以将其添加到集群中。
 - d. 在日志导出，请选择分析器日志以发布到 Amazon CloudWatch Logs。
 - e. 选择 Continue (继续) 以查看修改摘要。
 - f. 在确认您的更改后，您可以立即应用这些更改，也可以在 Scheduling of modifications (修改计划) 下的下一个维护时段内应用这些更改。
 - g. 选择 Modify cluster (修改集群) 以使用新参数组更新您的集群。

使用 Amazon CLI

以下过程对集群 `sample-cluster` 上的所有支持操作启用分析器。

1. 在开始之前，请运行以下命令，并查看对于名称中不包含 `default` 且具有 `docdb3.6` 作为参数组系列的集群参数组的输出，以确保您拥有可用的自定义集群参数组。如果您没有非默认集群参数组，请参阅[创建 Amazon DocumentDB 集群参数组 \(p. 327\)](#)。

```
aws docdb describe-db-cluster-parameter-groups \
  --query 'DBClusterParameterGroups[*].
[DBClusterParameterGroupName,DBParameterGroupFamily]'
```

在以下输出中，仅 `sample-parameter-group` 满足这两个条件。

```
[
  [
    "default.docdb3.6",
    "docdb3.6"
  ],
  [
    "sample-parameter-group",
    "docdb3.6"
  ]
]
```

2. 使用您的自定义集群参数组，修改以下参数。
 - `profiler`— 启用或禁用查询分析。允许的值为 `enabled` 和 `disabled`。默认值为 `disabled`。要启用分析，请将值设置为 `enabled`。
 - `profiler_threshold_ms`— 当 `profiler` 设置为 `enabled`，所有命令都用时超过 `profiler - threshold-ms` 已登录到 CloudWatch。允许的值为 `[0-INT_MAX]`。将此值设置为 0 可分析所有支持的操作。默认值为 100。
 - `profiler_sampling_rate`— 应该分析或记录的缓慢操作的部分。允许的值为 `[0.0-1.0]`。默认值为 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. 修改您的 Amazon DocumentDB 集群，以便它使用 `sample-parameter-group` 从之前步骤获取的自定义集群参数组并设置参数 `--enable-cloudwatch-logs-exports` 到 `profiler`。

以下代码修改集群 `sample-cluster` 要使用的 `sample-parameter-group` 从之前步骤获取的步骤，并添加 `profiler` 到已启用的 CloudWatch Logs 导出。

```
aws docdb modify-db-cluster \
  --db-cluster-identifier sample-cluster \
  --db-cluster-parameter-group-name sample-parameter-group \
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["profiler"]}'
```

此操作的输出将类似于下文。

```
{
  "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-ABCDEFGHIJKLMNORSTUVWXYZ",
    "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster",
    "AssociatedRoles": [],
    "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
    "EnabledCloudwatchLogsExports": [
        "profiler"
    ],
    "DeletionProtection": true
}
}

```

禁用 Amazon DocumentDB 档案器

要禁用分析器，请禁用分析器 `profiler` 参数和导出 `profiler` 将日志发布到 CloudWatch Logs。

禁用 分析器

您可以使用 Amazon Web Services Management Console 或 Amazon CLI 禁用 `profiler` 参数，如下所示。

使用 Amazon Web Services Management Console

以下过程使用Amazon Web Services Management Console禁用 Amazon DocumentDB profiler。

1. 在处登录Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择参数组。然后选择您要在其上禁用分析器的集群参数组的名称。
3. 在生成的 Cluster parameters (集群参数) 页面中，选择 profiler 参数左侧的按钮，然后选择 Edit (编辑)。
4. 在 Modify Profiler (修改分析器) 对话框中，在列表中选择 disabled。
5. 选择 Modify cluster parameter (修改集群参数)。

使用 Amazon CLI

要使用 Amazon CLI 在集群上禁用 profiler，请如下所示修改集群。

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name sample-parameter-group \
  --parameters ParameterName=profiler,ParameterValue=disabled,ApplyMethod=immediate
```

禁用分析器日志导出

你可以禁用导出 profiler 将日志发布到 CloudWatch Logs Amazon Web Services Management Console 要么 Amazon CLI，如下所示。

使用 Amazon Web Services Management Console

以下过程使用Amazon Web Services Management Console禁用 Amazon DocumentDB 将日志导出到 CloudWatch。

1. 在处打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择 Clusters。选择要禁用导出日志的集群名称左侧的按钮。
3. 在 Actions (操作) 菜单上，选择 Modify (修改)。
4. 向下滚动到 Log exports (日志导出) 部分并取消选择 Profiler logs (分析器日志)。
5. 选择 Continue (继续)。
6. 检查更改，然后选择何时将该更改应用到集群：
 - Apply during the next scheduled maintenance window (在下一个计划的维护时段内应用)
 - Apply immediately (立即应用)
7. 选择修改集群。

使用 Amazon CLI

以下代码修改集群 sample-cluster 并禁用 CloudWatch 分析器日志。

Example

对于 Linux、macOS 或 Unix：

```
aws docdb modify-db-cluster \
  --db-cluster-identifier sample-cluster \
```

```
--cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

对于 Windows：

```
aws docdb modify-db-cluster ^  
--db-cluster-identifier sample-cluster ^  
--cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

此操作的输出将类似于下文。

```
{  
  "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-ABCDEFGHIJKLMNQRSTUWXYZ",  
    "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster",  
    "AssociatedRoles": [],  
    "ClusterCreateTime": "2020-01-10T22:13:38.261Z",  
    "DeletionProtection": true  
  }  
}
```

访问您的 Amazon DocumentDB 档案器日志

按照以下步骤在 Amazon CloudWatch Logs 中访问您的分析日志。

1. 访问 <https://console.aws.amazon.com/cloudwatch/>，打开 CloudWatch 控制台。
2. 确保您与 Amazon DocumentDB 集群位于同一区域。
3. 在导航窗格中，选择日志。
4. 要查找集群的分析器日志，请在列表中选择 `/aws/docdb/yourClusterName/profiler`。

此时，每个实例名称的下方将显示该实例的分析日志。

常见查询

以下是您可以用来分析您的已分析命令的常见查询。有关 CloudWatch Logs Insights 的更多信息，请参阅[使用 CloudWatch Logs Insights 分析日志数据](#)和[查询示例](#)。

获取指定集合上最慢的 10 个操作

```
filter ns="test.foo" | sort millis desc | limit 10
```

获取集合上用时超过 60 毫秒的所有更新操作

```
filter millis > 60 and op = "update"
```

获取上个月最慢的 10 个操作

```
sort millis desc | limit 10
```

获取具有 COLLSCAN 计划摘要的所有查询

```
filter planSummary="COLLSCAN"
```

使用 Performance Insights 进行监控

Performance Insights 在现有 Amazon DocumentDB 监控功能的基础上进行了添加，以便通过示意图展示您的集群性能并帮助您分析影响性能。使用 Performance Insights 控制面板，您可以可视化数据库负载并按等待状态、查询语句、主机或应用程序筛选负载。

它有多有用？

- 可视化数据库性能 — 可视化负载以确定负载的时间和位置在数据库上
- 确定导致数据库负载的原因 — 确定哪些查询、主机和应用程序对实例的负载造成了负载
- 确定数据库何时会有负载 — 放大 Performance Insights 仪表板以专注于特定事件，或缩小以查看更长时间范围内的趋势
- 数据库加载警报 — 自动访问新的数据库加载指标 CloudWatch 您可以在其中监控数据库加载指标以及其他 DocumentDB 指标并在这些指标上设置警报

Amazon DocumentDB Performance Insights 有什么限制？

- 中的 Performance InsightsAmazon GovCloud（美国西部）地区尚未提供
- DocumentDB 的 Performance Insights 可保留长达 7 天的性能数据
- 超过 1024kb 的查询不会在 Performance Insights 中聚合

主题

- [Performance Insights 概念 \(p. 395\)](#)
- [启用和禁用 Performance Insights \(p. 397\)](#)
- [为 Performance Insights 配置访问策略 \(p. 399\)](#)
- [使用性能详情控制面板分析指标 \(p. 402\)](#)
- [使用性能详情 API 检索指标 \(p. 416\)](#)
- [亚马逊 CloudWatch Performance Insights 的指标 \(p. 425\)](#)
- [Performance Insights 的计数器指标 \(p. 426\)](#)

Performance Insights 概念

主题

- [平均活动会话数 \(p. 395\)](#)
- [维度 \(p. 396\)](#)
- [最大 vCPU \(p. 397\)](#)

平均活动会话数

数据库负载（数据库负载）衡量数据库中的活动级别。性能详情的关键指标是 DB Load，每秒收集一次。的单位DBLoad指标是“AAS”（AAS）对于 DocumentDB 实例。

网络 ACL 和安全组都允许（因此可到达您的实例）的发起 ping 的活跃会话是已将作业提交到 DocumentDB 实例并且正在等待响应的连接。例如，如果您将查询提交到 DocumentDB 实例，则数据库会话在实例处理该查询时将处于活动状态。

为了获取平均活动会话数，性能详情会对同时运行查询的会话数进行采样。平均活动会话数是会话总数除以样本总数。下表显示了正在运行的查询的五个连续示例。

示例	运行查询的会话数	AAS	计算
1	2	2	2 个会话/1 个样本
2	0	1	2 个会话/2 个样本
3	4	2	6 个会话/3 个样本
4	0	1.5	6 个会话/4 个样本
5	4	2	10 个会话/5 个样本

在上一示例中，从 1-5 的时间间隔的数据库负载为 2 AAS。数据库负载的增加意味着，平均而言数据库上运行的会话更多。

维度

DB Load 指标不同于其他时间序列指标，因为您可以将它分为称为维度的子组件。您可以将维度视为 DB Load 指标的不同特征的类别。诊断性能问题时，最有用的维度是等待状态和主要查询。

等待状态

一个等待状态导致查询语句等待特定事件发生，然后才能继续运行。例如，查询语句可能会一直等到已锁定资源解锁后才会解锁。通过合并 DB Load 使用等待状态，您可以全面了解会话状态。以下是各种 DocumentDB 等待状态：

DocumentDB 等待状态	等待状态描述
门锁	当会话正在等待分页缓冲池时，将出现 Latch 等待状态。当系统经常处理大型查询、集合扫描或缓冲池太小而无法处理工作集时，频繁寻呼缓冲池可能会更频繁地发生。
CPU	当会话在 CPU 上等待时，会出现 CPU 等待状态。
收藏锁	这些区域有：CollectionLock 在会话等待获取集合锁定时，会发生等待状态。当集合上有 DDL 操作时，会发生这些事件。
文档锁	这些区域有：DocumentLock 在会话等待获取文档锁定时，会发生等待状态。对同一文档进行大量并发写入操作将导致该文档出现更多的 DocumentLock 等待状态。
SystemLock	这些区域有：SystemLock 当会话在系统上等待时，会出现等待状态。当系统上经常出现长时间运行的查询、长时间运行的事务或高并发时，可能会发生这种情况。
IO	当会话等待 IO 完成时，会出现 IO 等待状态。
bufferLock	这些区域有：BufferLock 在会话等待获取缓冲区中共享页面上的锁定时，会发生等待状态。如果其他进程在请求的页面上保留打开的光标，则 BufferLock 等待状态可能会延长。
LowMemCottle	这些区域有：LowMemThrottle 由于 Amazon DocumentDB 实例的沉重内存压力导致会话正在等待时，会出现等待状态。如果此状态持续很长时间，请考虑扩展实例以提供额外的内存。有关更多信息，请参阅 资源管理器 。
背景活动	这些区域有：BackgroundActivity 当会话正在等待内部系统进程时，会出现等待状态。
其他	其他等待状态是内部等待状态。如果此状态持续很长一段时间，请考虑终止此查询。有关更多信息，请参阅 如何查找并终止长时间运行或受阻的查询？

主要查询

而等待状态显示瓶颈，而主要查询则显示哪些查询对数据库负载的贡献最大。例如，当前可能正在数据库上运行许多查询，但单个查询可能会占用 99% 的数据库负载。在这种情况下，高负载可能表示查询存在问题。

最大 vCPU

在控制面板中，数据库负载图表会收集、聚合和显示会话信息。要查看活动会话是否超过最大 CPU，请查看它们与最大 vCPU 线的关系。这些区域有：最大 vCPU 值由 DocumentDB 实例的 vCPU (虚拟 CPU) 内核数决定。

如果数据库负载经常高于最大 vCPU 线并且主要等待状态为 CPU，则表示 CPU 过载。在这种情况下，您可能需要限制与实例的连接数，优化具有高 CPU 负载的任何查询，或考虑使用更大的实例类。如果始终有大量实例处于任何等待状态，则表示可能存在要解决的瓶颈或资源争用问题。即使数据库负载未越过最大 vCPU 线，也可能出现此问题。

启用和禁用 Performance Insights

要使用 Performance Insights，请在数据库实例中启用它。如果需要，您可以稍后将其禁用。启用和禁用 Performance Insights 不会导致停机、重新启动或故障转移。

Performance Insights 代理占用数据库主机上有限的 CPU 和内存。当数据库负载较高时，代理将通过降低收集数据的频率来限制性能影响。

在创建集群时启用性能详情

在控制台中，您可以在创建或修改新数据库实例时启用或禁用 Performance Insights。

使用 Amazon Web Services Management Console

在控制台中，可以在创建 DocumentDB 集群时启用 Performance Insights。在创建新 DocumentDB 集群时，通过以下方式启用 Performance Insights 情：Enable Performance Insights (启用 Performance Insights) 中的 Performance Insights 部分。

控制台说明

1. 要创建集群，请按照以下说明操作 [创建 Amazon DocumentDB 集群](#)。
2. SelectEnable Performance Insights (启用 Performance Insights) 在 Performance Insights 部分。

Performance Insights [Info](#)

☒ Enable Performance Insights

KMS Key [Info](#)

(default) aws/rds

Account

KMS key ID

 You can't change the KMS key after enabling Performance Insights.

Note

Performance Insights 数据保留期将为 7 天。

Amazon KMS 密钥— 指定您的 Amazon KMS 密钥。Performance Insights 使用您的 Amazon KMS 密钥来加密所有潜在的敏感数据。正在传输的数据和静态数据都会被加密。有关更多信息，请参阅配置 Amazon KMS Performance Insights 的策略。

在修改实例时启用和禁用

您可以修改数据库实例以使用控制台或禁用 Performance Insights。Amazon CLI。

使用 Amazon Web Services Management Console

控制台说明

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 选择 Clusters (集群)。
3. 选择一个数据库实例，然后选择修改。
4. 在 Performance Insights 部分中选择 Enable Performance Insights (启用 Performance Insights) 要么禁用 Performance Insights。

Note

如果选择 Enable Performance Insights (启用 Performance Insights)，您可以指定您的 Amazon KMS 密钥。Performance Insights 使用您的 Amazon KMS 密钥来加密所有潜在的敏感数据。正在传输的数据和静态数据都会被加密。有关更多信息，请参阅 [加密 Amazon DocumentDB 静态数据](#)。

5. 选择 Continue (继续)。
6. 对于修改计划，选择立即应用。如果选择 Apply during the next scheduled maintenance window (在下一个计划的维护时段内应用)，您的实例将忽略此设置并立即启用性能详情。
7. 选择 Modify instance (修改实例)。

使用 Amazon CLI

当您使用 `create-db-instance` 要么 `modify-db-instance` Amazon CLI 可以通过以下方式启用 Performance Insights 详情：`--enable-performance-insights`，或者通过指定将其禁用 `--no-enable-performance-insights`。

以下过程介绍如何使用为数据库实例启用或禁用 Performance Insights。Amazon CLI。

Amazon CLI 说明

调用 `modify-db-instance` Amazon CLI 命令并提供以下值：

- `--db-instance-identifier`— 数据库实例的名称
- `--enable-performance-insights` 以启用，或 `--no-enable-performance-insights` 以禁用

Example

以下示例为启用 Performance Insights：`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
```

为 Performance Insights 配置访问策略

要访问 Performance Insights，您必须拥有 Amazon Identity and Access Management (IAM) 的相应权限。您可以使用以下选项来授予访问权限：

- 将 AmazonRDSPerformanceInsightsReadOnly 托管策略附加到 IAM 用户或角色。
- 创建自定义 IAM 策略并将其附加到 IAM 用户或角色。

此外，如果您在启用 Performance Insights 时指定了客户托管密钥，请确保账户中的用户对 KMS 密钥具有 kms:Decrypt 和 kms:GenerateDataKey 权限。

Note

适用于 encryption-at-rest 和 Amazon KMS 密钥和安全组管理，Amazon DocumentDB 利用与共享的操作技术。 [Amazon RDS](#).

将 AmazonRDSPerformanceInsights 只读策略附加到 IAM 委托人

AmazonRDSPerformanceInsightsReadOnly 是 Amazon-托管策略，可以授予对 Amazon DocumentDB Performance Insights API 的所有只读操作的访问权限。目前，此 API 中的所有操作均为只读。如果将 AmazonRDSPerformanceInsightsReadOnly 附加到 IAM 用户或角色，收件人可以使用 Performance Insights 以及其他控制台功能。

为 Performance Insights 创建自定义 IAM 策略

对于没有 AmazonRDSPerformanceInsightsReadOnly 策略中，您可以通过创建或修改用户托管 IAM 策略来授予对 Performance Insights 的访问权限。当您策略附加到 IAM 用户或角色时，收件人可以使用 Performance Insights。

创建自定义策略

1. 通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择 Policies (策略)。
3. 选择 Create policy (创建策略)。
4. 在创建策略页面上，选择“JSON”选项卡。
5. 复制并粘贴以下文本，将 `us-east-1` 替换为您的 Amazon 区域的名称，并将 `111122223333` 替换为您的客户账号。


```
{
  "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. 选择 Review policy (审核策略)。
7. 为策略提供名称并可以选择提供描述，然后选择创建策略。

现在，可以将策略附加到 IAM 用户或角色。以下过程假设您已经有一个可用于此目的的 IAM 用户。

将策略附加到 IAM 用户

1. 打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择 Users。
3. 从列表中选择现有用户。

Important

要使用 Performance Insights，请确保除了自定义策略之外，您还有权访问 Amazon DocumentDB。例如，AmazonDocDBReadOnlyAccess 预定义策略提供了对 Amazon DocDB 的只读访问权限。[使用策略管理访问](#)。

- 在 Summary (摘要) 页上，选择 Add permissions (添加权限)。
- 选择直接附加现有策略。对于 Search，键入策略名称的前几个字符，如下所示。

The screenshot shows the 'Add permissions to test' console interface. At the top, there are two numbered tabs: '1' (selected) and '2'. Below the title, the section is 'Grant permissions'. A message states: 'Use IAM policies to grant permissions. You can assign an existing policy or create a new one.' There are three main action buttons: 'Add user to group' (with a group icon), 'Copy permissions from existing user' (with a user icon), and 'Attach existing policies directly' (with a document icon). Below these is a 'Create policy' button and a refresh icon. A search bar labeled 'Filter policies' contains the text 'Perf'. To the right of the search bar, it says 'Showing 1 result'. Below the search bar is a table with the following content:

	Policy name	Type	Used as
☐	PerformanceInsightsCustomPolicy	Customer managed	None

- 选择策略，然后选择后续：审核。
- 选择 Add permissions (添加权限)。

为 Performance Insights 配置 Amazon KMS 策略

Performance Insights 使用 Amazon KMS key 加密敏感数据。当您通过 API 或控制台启用 Performance Insights 时，您可以选择以下选项：

- 选择原定设置 Amazon 托管式密钥。

Amazon DocumentDB 使用 Amazon 托管式密钥为您的新数据库实例。Amazon DocumentDB 创建 Amazon 托管式密钥为您的 Amazon account。您的 Amazon 账户有不同 Amazon 托管式密钥对于每个 Amazon DocumentDB 的 Amazon 区域。

- 选择客户托管密钥。

如果您指定一个客户托管密钥，则您账户中调用 Performance Insights API 的用户需要在 KMS 密钥具有 `kms:Decrypt` 和 `kms:GenerateDataKey` 权限。您可以通过 IAM 策略配置这些权限。但是，我们建议您通过 KMS 密钥策略来管理这些权限。有关更多信息，请参阅在 [Amazon KMS 中使用密钥策略](#)。

Example

以下示例密钥策略显示了如何将语句添加到 KMS 密钥策略。这些语句可以访问 Performance Insights。根据您使用的方式 Amazon KMS，您可能想要更改一些限制。在将语句添加到您的策略之前，请删除所有注释。

```
{
  "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-AAAABBBBBCCCCDDDDDEEEEE"
  }
}
}
```

使用性能详情控制面板分析指标

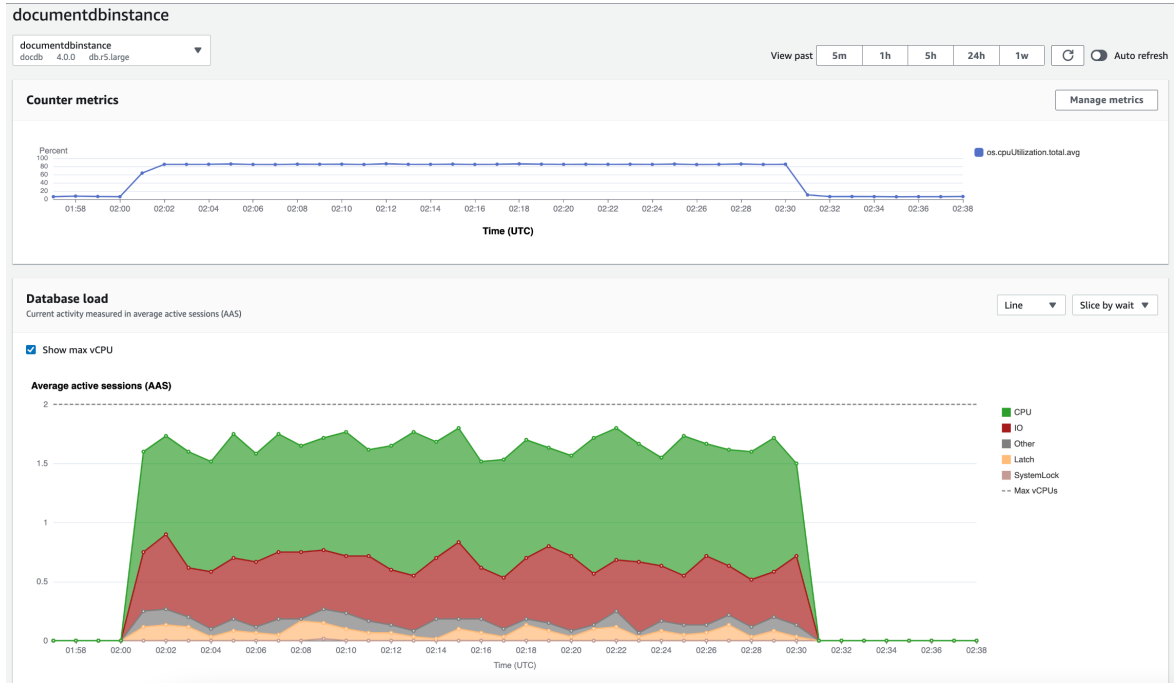
Performance Insights 控制面板包含帮助您分析和排查性能问题的数据库性能信息。在主控制面板页面上，可以查看有关数据库负载（数据库负载）的信息。您可以按维度（例如等待状态或查询）对数据库负载进行“切片”。

主题

- [性能详情控制面板概览 \(p. 402\)](#)
- [打开性能详情控制面板 \(p. 408\)](#)
- [按等待状态分析数据库负载 \(p. 409\)](#)
- [主要查询选项卡概览 \(p. 411\)](#)
- [放大数据库负载图 \(p. 415\)](#)

性能详情控制面板概览

与性能详情进行交互的最简单方式即为控制面板。以下示例显示了 Amazon DocumentDB 实例的控制面板。默认情况下，Performance Insights 控制面板将显示最近一小时的数据。



控制面板分为以下几个部分：

1. 计数器指标— 显示特定性能计数器指标的数据。
2. 数据库负载— 显示数据库负载与由表示的数据库实例容量的比较情况。最大 vCPU 行。
3. 主要维度— 显示导致数据库负载的主要维度。这些维度包括 waits、queries、hosts、databases、和 applications。

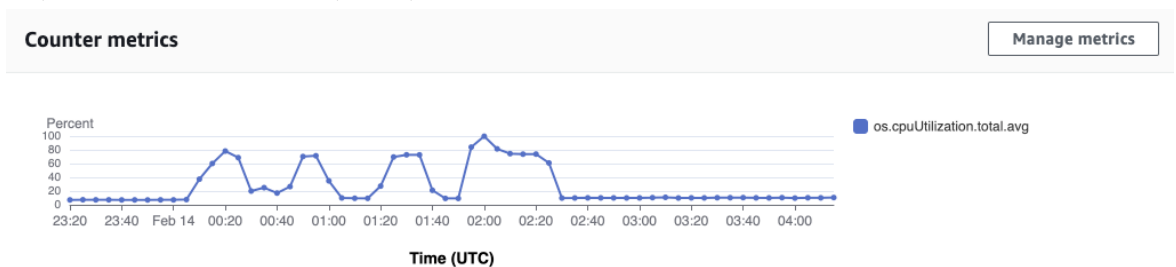
主题

- [计数器指标图表 \(p. 403\)](#)
- [数据库负载图表 \(p. 404\)](#)
- [主要维度表 \(p. 406\)](#)

计数器指标图表

使用计数器指标，您可以自定义 Performance Insights 控制面板来包括最多 10 个其他图表。这些图表显示了所选的数十个操作系统指标。您可将此信息与数据库负载相关联，以帮助识别和分析性能问题。

计数器指标图表显示了性能计数器的数据。



要更改性能计数器，请选择管理指标。您可以选择多个OS 指标如下屏幕截图所示。要查看任何指标的详细信息，请将鼠标悬停在相应指标名称上。

Select metrics shown on the graph ✕

Check the metrics that you want to see on the Performance Insights dashboard.

Find metrics

OS metrics (4)

Clear all selections

▼ general

☐ numVCPU

▼ cpuUtilization

☐ idle☒ system☒ total

☐ user☒ wait

▼ loadAverageMinute

☐ fifteen☐ five☐ one

▼ memory

☐ active☒ buffers☐ cached

☐ dirty☐ free☐ inactive

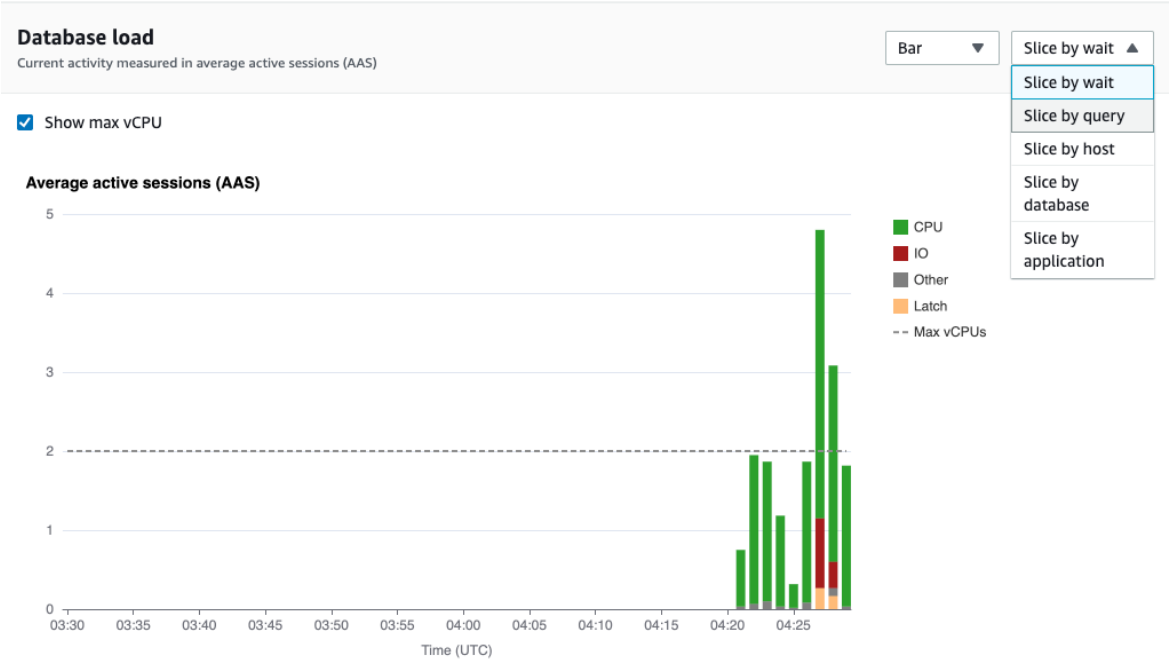
数据库负载图表

这些区域有：数据库负载图表显示数据库活动与由最大 vCPU行。预设情况下，堆叠折线图将以每单位时间的平均活动会话数表示数据库负载。数据库负载按等待状态进行切片（分组）。



按维度切片的数据库负载

您可以选择按任何受支持维度分组的活动会话显示负载。下图显示了 Amazon DocumentDB 实例的维度。

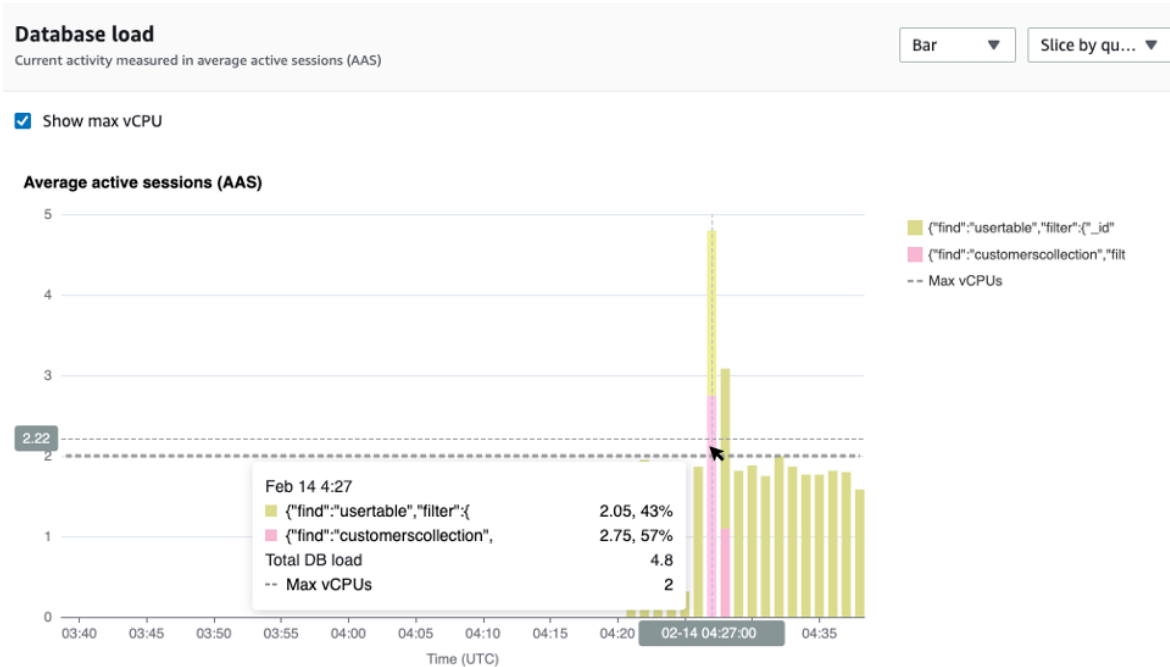


维度项目的数据库负载详细信息

要查看维度中数据库负载项目的详细信息，请将光标悬停在相应项目名称上。下图显示了查询语句的详细信息。



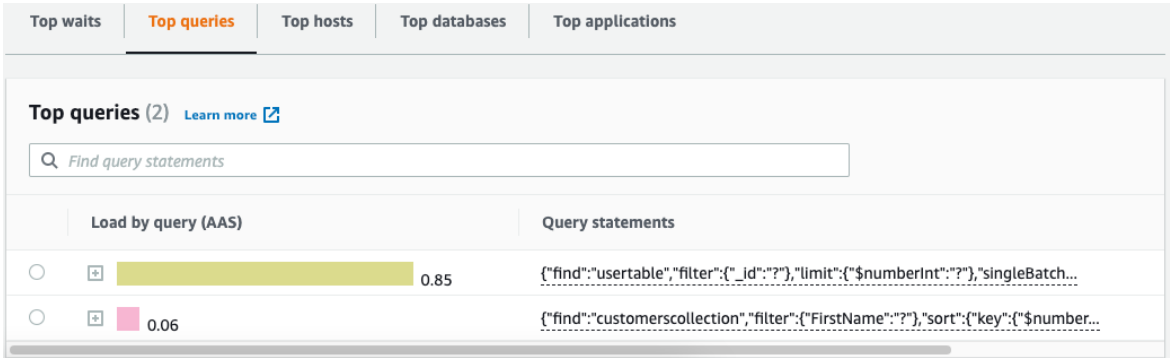
要在图例中查看任何项目在选定时间段内的详细信息，请将鼠标悬停在相应项目上。



主要维度表

这些区域有：主要维度表按不同的维度切割数据库负载。维度是数据库负载不同特征类别或“切片依据”。如果维度是查询，主要查询显示了对数据库负载贡献最大的查询语句。

请选择以下任何一个维度选项卡。



下表提供了每个选项卡的简要描述。

选项卡
数据终端正在等

透明卡
待的事件
当前查询运行的查询语句
需要连接客户端的主机 IP 和端口
需要端端连接的数据库的名称

透明卡
连接数据库的应用程序的名称

要了解如何使用分析查询主要查询选项卡，请参阅[主要查询选项卡概览 \(p. 411\)](#)。

打开性能详情控制面板

要查看性能 Performance Insights 控制面板，请执行以下步骤：

- 1. 在处打开 Performance Insights 控制台<https://console.aws.amazon.com/docdb/>。
- 2. 选择一个数据库实例。将为该 Amazon DocumentDB 实例显示 Performance Insights 详情控制面板。

对于启用 Performance Insights 的 Amazon DocumentDB 实例，您还可以通过选择会话实例列表中的项目。在当前活动下，会话项目显示在过去五分钟内平均活跃会话中的数据库负载。条形图显示负载量。当条形图为空时，实例处于空闲状态。随着负载的增加，条形图会以蓝色填充。当负载超过实例类上的虚拟 CPU (vCPUs) 数量时，条形图变为红色，表示可能出现瓶颈。

Clusters (1)

Group Resources

Actions

Create

Filter Resources

	Cluster identifier	Role	Engine version	Region & AZ	Status	CPU	Current activity
	documentdbinstance	Regional cluster	4.0.0	ap-south-1	available	-	-
	documentdbinstance	Primary instance	4.0.0	ap-south-1c	available	84.99%	5 Connections
	documentdbinstance2	Replica instance	4.0.0	ap-south-1b	available	15.37%	2 Connections
	documentdbinstance3	Replica instance	4.0.0	ap-south-1a	available	14.84%	2 Connections

- 3. (可选) 通过选择右上角的按钮来选择不同的时间间隔。例如，要将间隔更改为 1 小时，请选择 1h。

View past

5m

1h

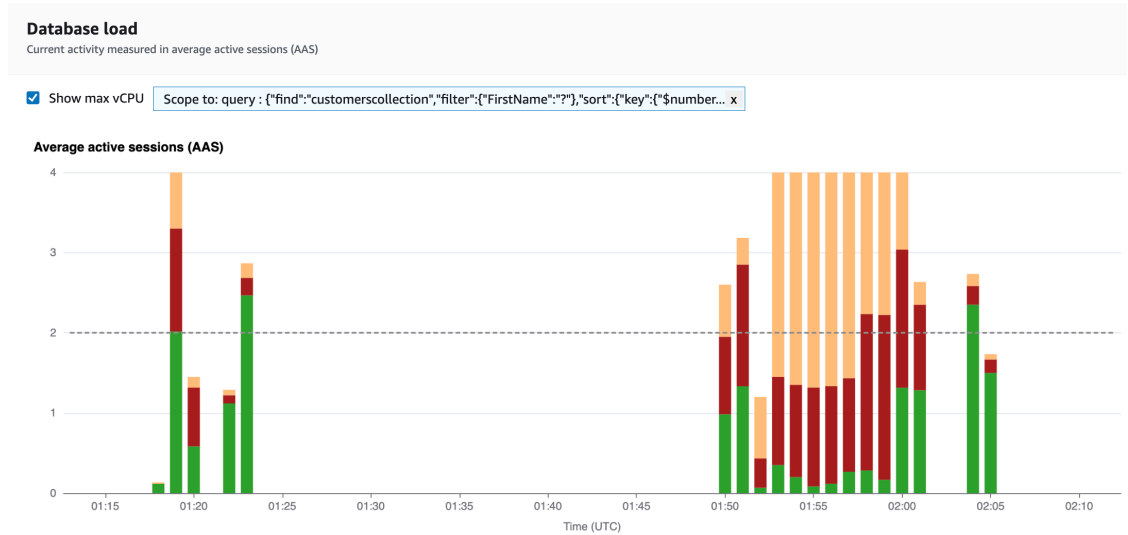
5h

24h

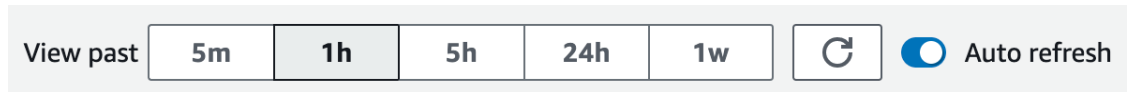
1w

Auto refresh

在以下屏幕截图中，数据库负载间隔为 1 小时。



4. 要自动刷新数据，请启用自动刷新。



Performance Insights 控制面板自动刷新新的数据。刷新速率取决于所显示的数据量：

- 5 分钟则每 5 秒刷新一次。
- 1 小时则每分钟刷新一次。
- 5 小时则每分钟刷新一次。
- 每 5 分钟刷新 24 小时一次。
- 每小时刷新一周一次。

按等待状态分析数据库负载

如果数据库加载（数据库加载）图表显示了一个瓶颈，您可以找出负载的来源。为此，请查看数据库负载图表下方的主要负载项目。选择特定项目（如查询或应用程序）以深入了解该项目并查看有关该项目的详细信息。

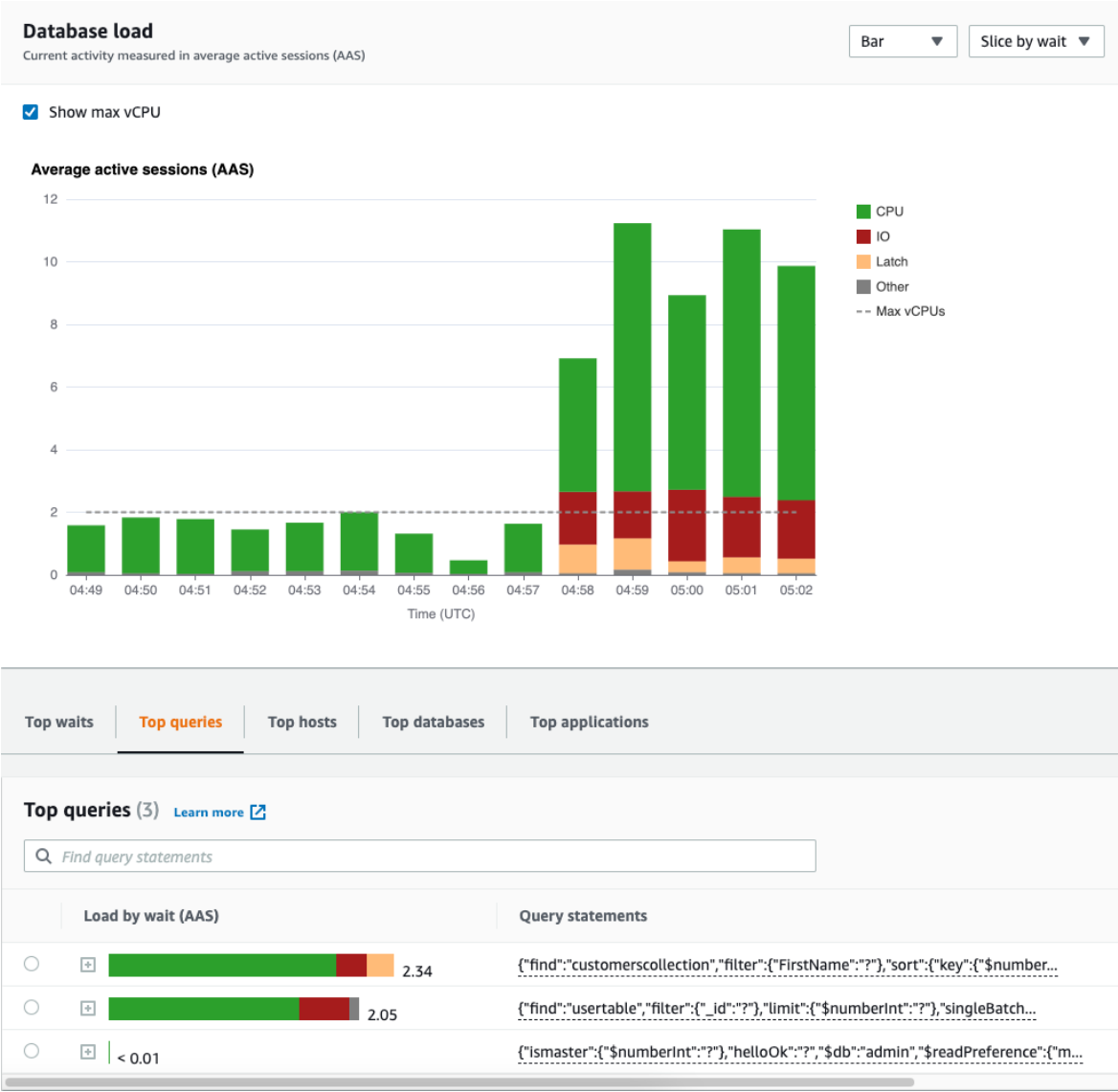
按等待状态和主要查询分组的数据库负载通常提供了最多的性能问题见解。按等待状态分组的数据库负载显示了数据库中是否存在任何资源瓶颈或并发瓶颈。在这种情况下，主要查询主要负载项目表的选项卡显示了驱动该负载的查询。

诊断性能问题的典型工作流程如下：

1. 查看数据库负载图表并了解是否存在数据库负载的事件越过了 Max CPU 线。
2. 如果有，请查看数据库负载图表并确定负主要责任的等待状态。
3. 通过以下方式确定导致负载的摘要查询：主要查询主要负载项目表上的选项卡对于这些等待状态所起的作用最大。您可以通过按等待状态排列负载 (AAS)column。
4. 在主要查询选项卡以展开它并查看它包含的子查询。

您还可以通过选择选择来查看哪些主机或应用程序的负载最大主要主机要么主要应用程序，分别。应用程序名称在到 Amazon DocumentDB 实例的连接字符串中指定。Unknown表示未指定应用程序字段。

例如，在以下仪表板中，CPU等待占大部分数据库负载。选择下面的顶级查询主要查询将限定数据库负载图的范围，以便将重点放在 select 查询贡献的最大负载上。



Top waits

Top queries

Top hosts

Top databases

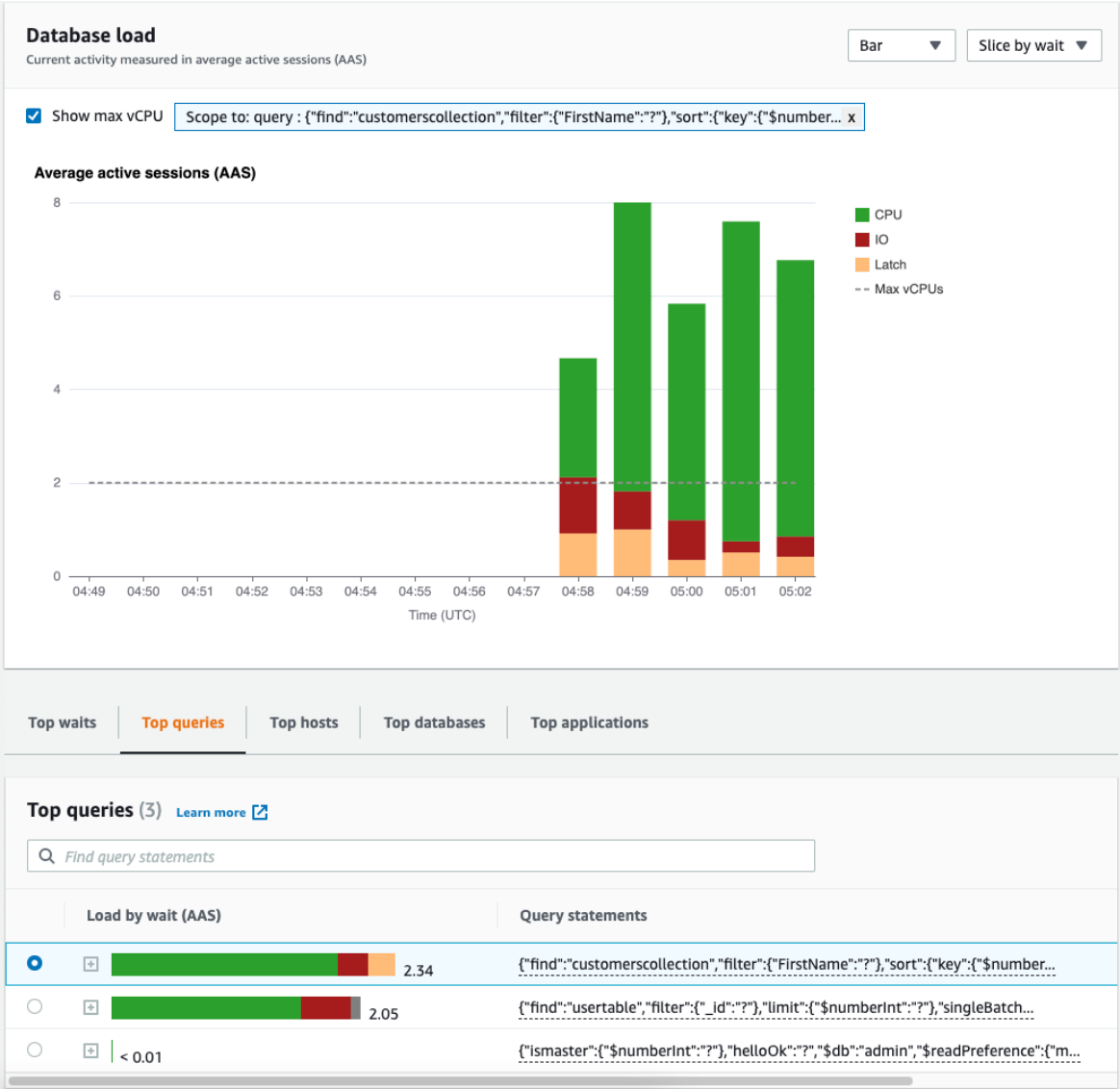
Top applications

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...



主要查询选项卡概览

默认情况下，主要查询选项卡显示了对数据库负载影响最大的查询。您可以分析查询文本以帮助调整查询。

主题

- [查询摘要](#) (p. 411)
- [按等待状态排列的负载 \(AAS\)](#) (p. 412)
- [查看详细的查询信息](#) (p. 413)
- [访问语句查询文本](#) (p. 414)
- [查看和下载对账单查询文本](#) (p. 414)

查询摘要

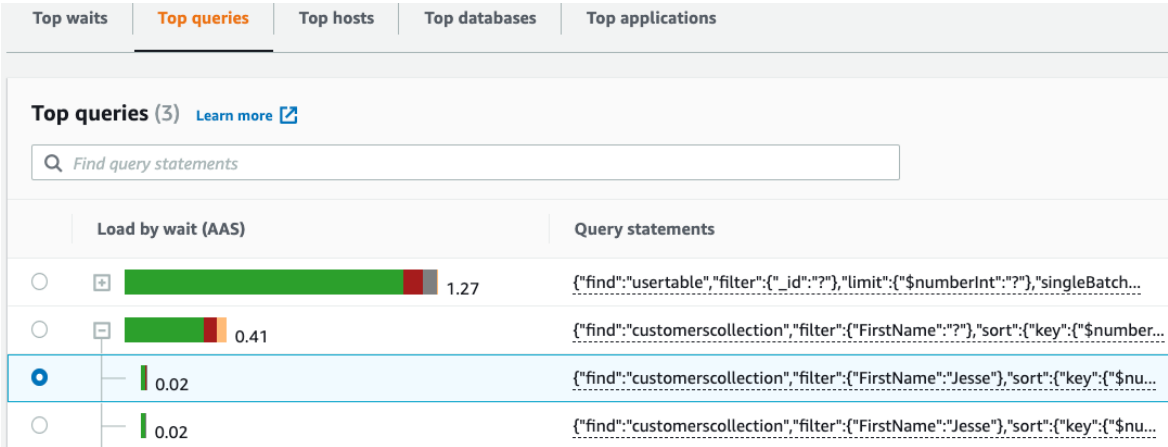
一个查询摘要是在结构上相似但可能具有不同文本值的多个实际查询的组合。摘要用问号替换硬编码值。例如，查询摘要可能类似于下面这样：

```
{ "find": "customerscollection", "filter": { "FirstName": "?" }, "sort": { "key": { "$numberInt": "?" } }, "limit": { "$numberInt": "?" } }
```

此摘要可能包含以下子查询：

```
{ "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" } }
```

要查看摘要中的文字查询语句，请选择查询，然后选择加号 (+)。在下面的屏幕截图中，选定的查询是摘要。

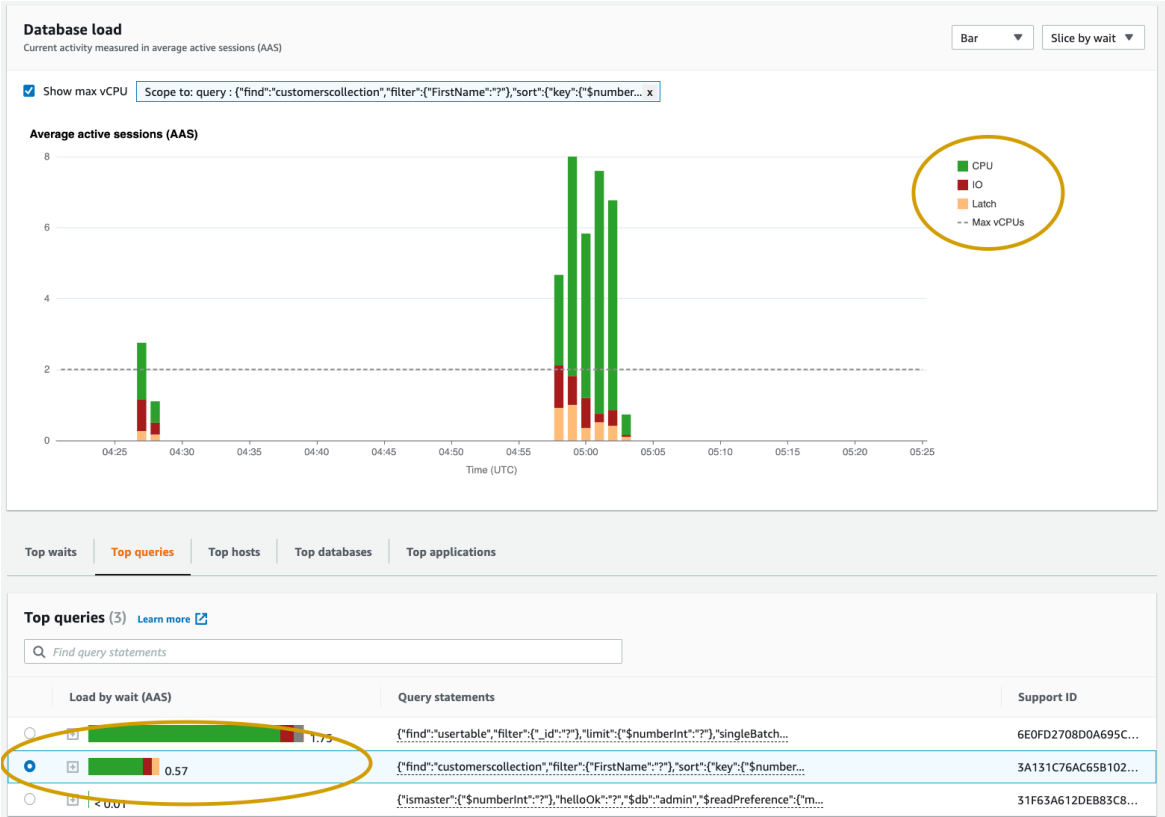


Note

查询摘要将相似的查询语句进行分组，但不会编辑敏感信息。

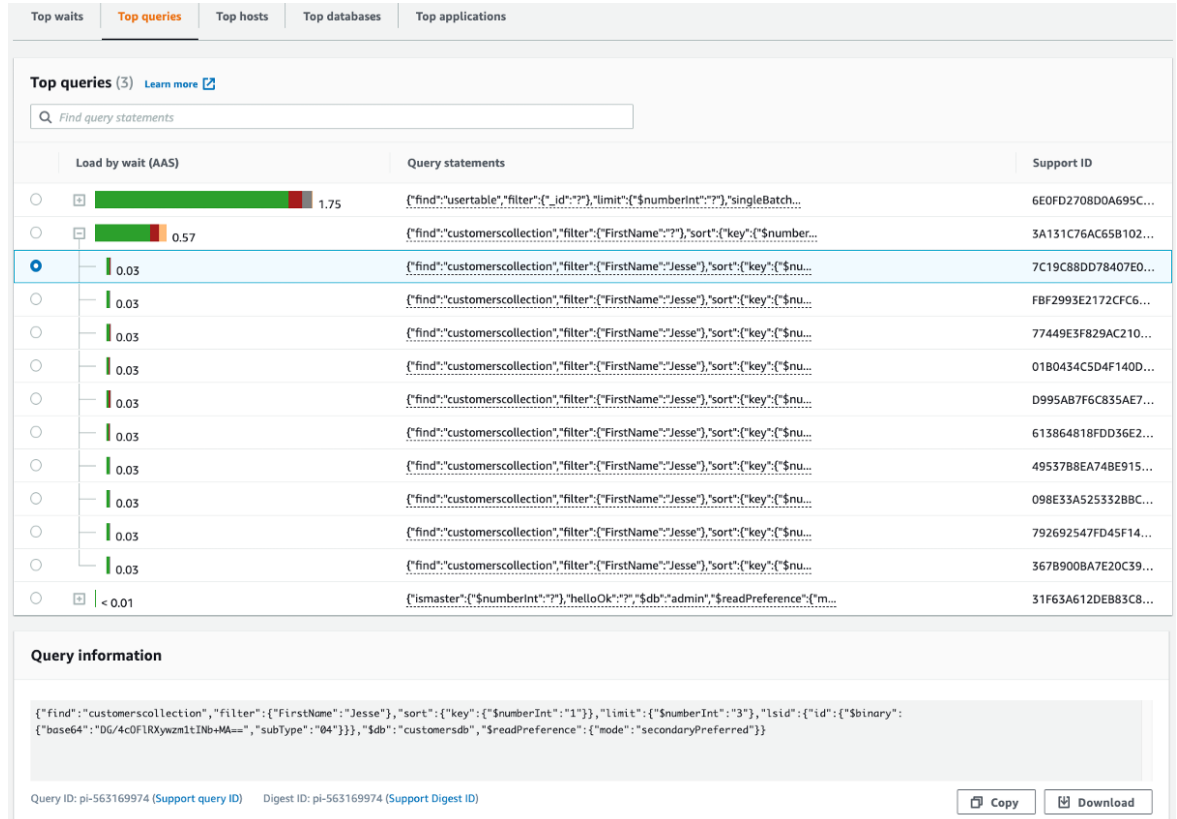
按等待状态排列的负载 (AAS)

In主要查询，按等待状态排列的负载 (AAS)列说明了与每个主要负载项目关联的数据库负载的百分比。此列按当前在中选择的分组方式反映该项目的负载。数据库负载图表. 例如，您可以按等待状态对数据库负载图表进行分组。在这种情况下，系统将对 DB Load by Waits (按等待状态排列的数据库负载) 栏进行大小调整、分段和颜色编码，以显示该查询在导致给定等待状态方面所起的作用大小，它还会显示哪些等待状态正在影响选定的查询。



查看详细的查询信息

在主要查询表格中，你可以打开摘要声明查看它的信息。信息将显示在底部窗格中。



以下类型的标识符 (ID) 与查询语句关联：

1. Support 查询 ID— 查询 ID 的哈希值。此值仅用于当您处理时引用查询 ID。AmazonSupport 无法访问您实际的查询 ID 和查询文本。
2. Support 摘要 ID— 摘要 ID 的哈希值。此值仅用于当您处理 Amazon Support 时引用摘要 ID。AmazonSupport 无法访问您实际的摘要 ID 和查询文本。

访问语句查询文本

默认情况下，中的每个行主要查询表显示了每个查询语句的 500 字节的查询文本。当摘要语句超过 500 字节时，可通过在 Performance Insights 控制面板中打开该语句来查看更多文本。在这种情况下，显示的查询的最大长度为 1 KB。如果查看完整查询语句，也可以选择下载。

查看和下载对账单查询文本

在 Performance Insights 控制面板中，您可以查看或下载查询文本。

在 Performance Insights 控制面板中查看更多查询文本

1. 在处打开 Amazon DocumentDB 控制台：<https://console.aws.amazon.com/docdb/>
2. 在导航窗格中，选择 Performance Insights。
3. 选择一个数据库实例。将为该数据库实例显示 Performance Insights 控制面板。

具有大于 500 字节的文本的查询语句类似于下图所示：

Top queries (3) [Learn more](#)

	Load by wait (AAS)	Query statements	Support ID
☐	1.75	{ "find": "usertable", "filter": { "id": "?" }, "limit": { "\$numberInt": "3" }, "singleBatch": true }	6E0FD2708D0A695C...
☐	0.57	{ "find": "customerscollection", "filter": { "firstName": "?" }, "sort": { "key": { "\$numberInt": "1" } }, "limit": { "\$numberInt": "3" }, "isid": { "\$binary": "base64:0G/4c0F1RXyzmltINb+MA==" }, "subType": "04" } }	3A131C76AC65B102...
☒	0.03	{ "find": "customerscollection", "filter": { "firstName": "Jesse" }, "sort": { "key": { "\$numberInt": "1" } }, "limit": { "\$numberInt": "3" }, "isid": { "\$binary": "base64:0G/4c0F1RXyzmltINb+MA==" }, "subType": "04" } }	7C19C88DD78407E0...
☐	0.03	{ "find": "customerscollection", "filter": { "firstName": "Jesse" }, "sort": { "key": { "\$numberInt": "1" } }, "limit": { "\$numberInt": "3" }, "isid": { "\$binary": "base64:0G/4c0F1RXyzmltINb+MA==" }, "subType": "04" } }	FBF2993E2172CFC6...

4. 检查查询信息部分以查看更多查询文本。

Query information

```
{ "find": "customerscollection", "filter": { "firstName": "Jesse" }, "sort": { "key": { "$numberInt": "1" } }, "limit": { "$numberInt": "3" }, "isid": { "$binary": "base64:0G/4c0F1RXyzmltINb+MA==" }, "subType": "04" } }
```

Query ID: pi-563169974 (Support query ID) Digest ID: pi-563169974 (Support Digest ID)

Copy

Download

Performance Insights 控制面板可以为每个完整查询语句最多显示 1 KB。

Note

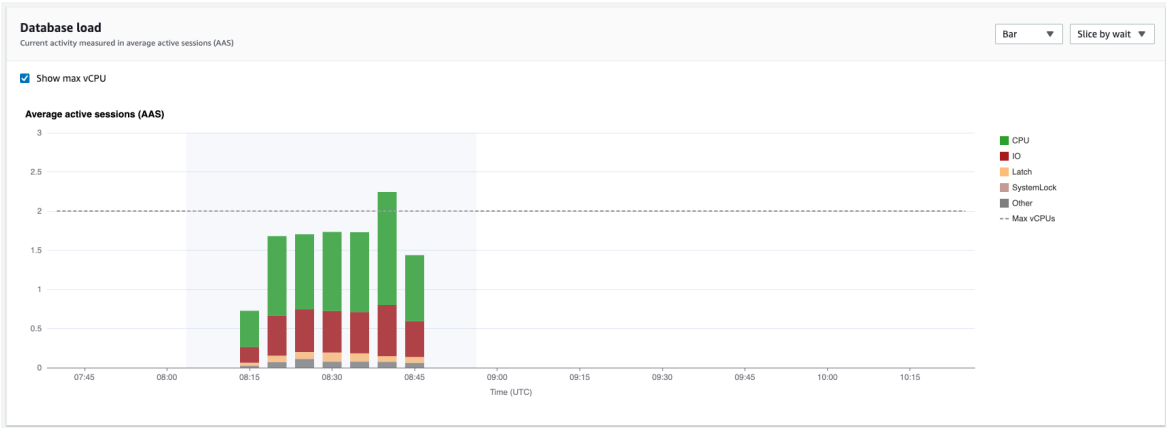
要复制或下载查询语句，请禁用所有弹出窗口阻止程序。

放大数据库负载图

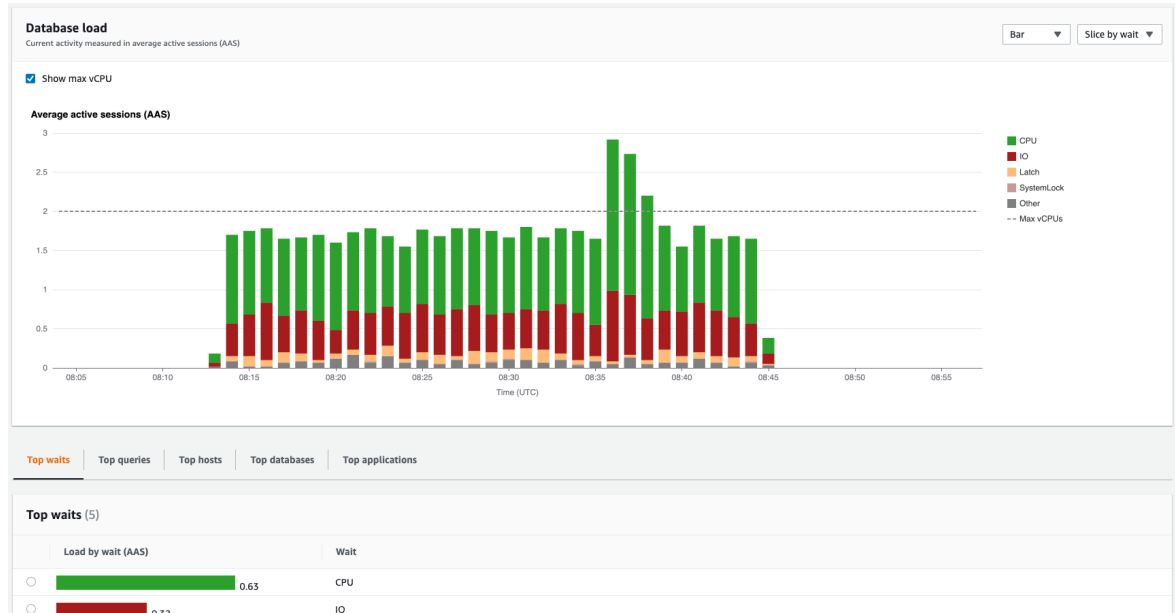
您可以使用 Performance Insights 用户界面的其他功能来帮助分析性能数据。

单击并拖动进行放大

在 Performance Insights 界面中，您可以选择负载图表的一小部分并放大细节。



要放大负载图表的一部分，请选择开始时间并拖动到所需时间段的结尾。执行该操作时，所选区域将突出显示。释放鼠标时，负载图表上的所选区域将放大，并将页首项目表格被重新计算。



使用性能详情 API 检索指标

启用性能详情后，API 将提供实例性能的可见性。亚马逊 CloudWatch 日志为以下方式提供了售卖监控指标的权威源：Amazon 服务。

性能详情提供了按平均活动会话 (AAS) 衡量的数据库负载的特定于域的视图。对 API 使用者而言，此指标看起来像是二维时间序列数据集。数据的时间维度提供所查询时间范围的每个时间点的数据库负载数据。每个时间点将分解与所请求维度相关的整体负载，如相应时间点测量的 Query、Wait-state、Application 或 Host。

Amazon DocumentDB Performance Insights 用于监控您的 Amazon DocumentDB 数据库实例，使您可以分析数据库性能和排查数据库性能问题。查看 Performance Insights 数据的一种方法是在 Amazon Web Services Management Console 中。Performance Insights 还提供公有 API，以便您可以查询自己的数据。您可以使用 API 来执行以下操作：

- 将数据卸载到数据库中
- 将性能详情数据添加到现有监控控制面板
- 构建监控工具

要使用 Performance Insights API，请在您的 Amazon DocumentDB 实例之一上启用 Performance Insights。有关启用 Performance Insights 的信息，请参阅 [启用和禁用 Performance Insights \(p. 397\)](#)。有关 Performance Insights API 的更多信息，请参阅 [Performance Insights API 参考](#)。

Performance Insights API 提供以下操作。

Performance Insights 操作	Amazon CLI command	描述
<code>DescribeDimensionKeys</code>	<code>aws pi describe-dimension-keys</code>	对于特定的时间段，检索指标的前 N 个维度键。
<code>GetDimensionKeyDetails</code>	<code>aws pi get-dimension-key-details</code>	检索数据库实例或数据源的指定维度组的属性。例如，如果您指定了查询 ID，并且有维度详细信息

Performance Insights 操作	Amazon CLI command	描述
		息, <code>GetDimensionKeyDetails</code> 检索维度的完整文本 <code>db.query.statement</code> 与此 ID 关联。此操作很有用, 因为 <code>GetResourceMetrics</code> 和 <code>DescribeDimension</code> 支持检索大型查询语句文本。
<code>GetResourceMetadata</code>	<code>aws pi get-resource-metadata</code>	检索不同功能的元数据。例如, 元数据可以表明特定数据库实例上的某个功能已打开或关闭。
<code>GetResourceMetrics</code>	<code>aws pi get-resource-metrics</code>	检索一组数据源在一段时间内的性能详情指标。您可以提供特定维度组和维度, 并为每个组提供聚合和筛选条件。
<code>ListAvailableResourceDimensions</code>	<code>aws pi list-available-resource-dimensions</code>	检索特定实例上每个特定指标类型可查询的维度。
<code>ListAvailableResourceMetrics</code>	<code>aws pi list-available-resource-metrics</code>	检索指定指标类型的所有可用指标, 指定数据库实例可用该指标进行查询。

主题

- [Amazon CLI Performance Insights 的](#) (p. 417)
- [检索时间序列指标](#) (p. 417)
- [Amazon CLI 性能详情的 示例](#) (p. 419)

Amazon CLI Performance Insights 的

您可以使用 Amazon CLI 查看 Performance Insights 数据。可以通过在命令行上输入以下内容来查看 Performance Insights 的 Amazon CLI 命令的帮助。

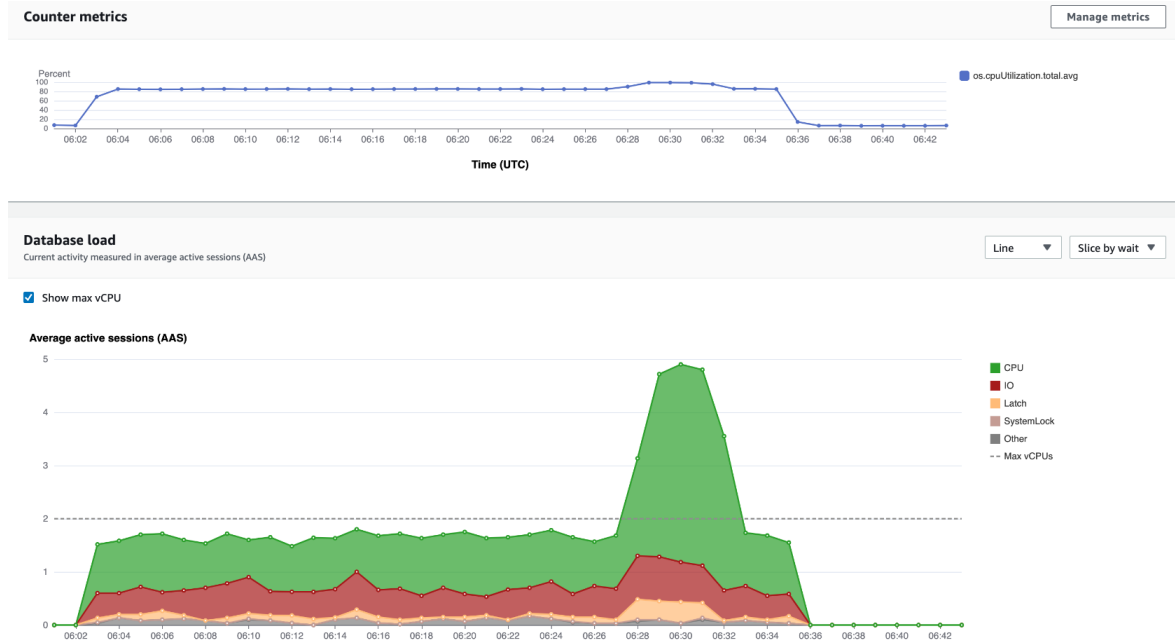
```
aws pi help
```

如果尚未安装 Amazon CLI, 请参阅 Amazon CLI 用户指南中的 [安装 Amazon 命令行界面](#) 来了解安装信息。

检索时间序列指标

`GetResourceMetrics` 操作从 Performance Insights 数据中检索一个或多个时间序列指标。`GetResourceMetrics` 需要指标和时间段, 并返回包含数据点列表的响应。

例如, Amazon Web Services Management Console 使用 `GetResourceMetrics` 来填充 Counter Metrics (计数器指标) 图表和 Database Load (数据库负载) 图表, 如下图所示。



`GetResourceMetrics` 返回的所有指标都是标准的时间序列指标，但 `db.load` 除外。此指标显示在 Database Load (数据库负载) 图表中。`db.load` 指标不同于其他时间序列指标，因为您可以将它分为称为维度的子组件。在上图中，按组成 `db.load` 的等待状态对 `db.load` 进行细分和分组。

Note

`GetResourceMetrics` 也可以返回 `db.sampleload` 指标，但 `db.load` 指标在大多数情况下是合适的。

有关 `GetResourceMetrics` 返回的计数器指标的信息，请参阅[Performance Insights 的计数器指标 \(p. 426\)](#)。

指标支持以下计算：

- 平均值 – 指标在一段时间内的平均值。在指标名称后面附加 `.avg`。
- 最小值 – 指标在一段时间内的最小值。在指标名称后面附加 `.min`。
- 最大值 – 指标在一段时间内的最大值。在指标名称后面附加 `.max`。
- 总计 – 指标值在一段时间内的总计。在指标名称后面附加 `.sum`。
- 样本数 – 在一段时间内收集指标的次数。在指标名称后面附加 `.sample_count`。

例如，假定在 300 秒（5 分钟）时段内收集指标，并且每分钟收集一次指标。各分钟的值为 1、2、3、4 和 5。在本例中，返回以下计算：

- 平均值 – 3
- 最小值 – 1
- 最大值 – 5
- 总计 – 15
- 样本数 – 5

有关使用 `get-resource-metrics` Amazon CLI 命令，请参阅[get-resource-metrics](#)。

对于 `--metric-queries` 选项，请指定一个或多个要获取其结果的查询。每个查询包括必需的 `Metric` 和可选的 `GroupBy` 和 `Filter` 参数。以下是 `--metric-queries` 选项规范的示例。

```
{
  "Metric": "string",
  "GroupBy": {
    "Group": "string",
    "Dimensions": ["string", ...],
    "Limit": integer
  },
  "Filter": {"string": "string"
  ...}
}
```

Amazon CLI性能详情的 示例

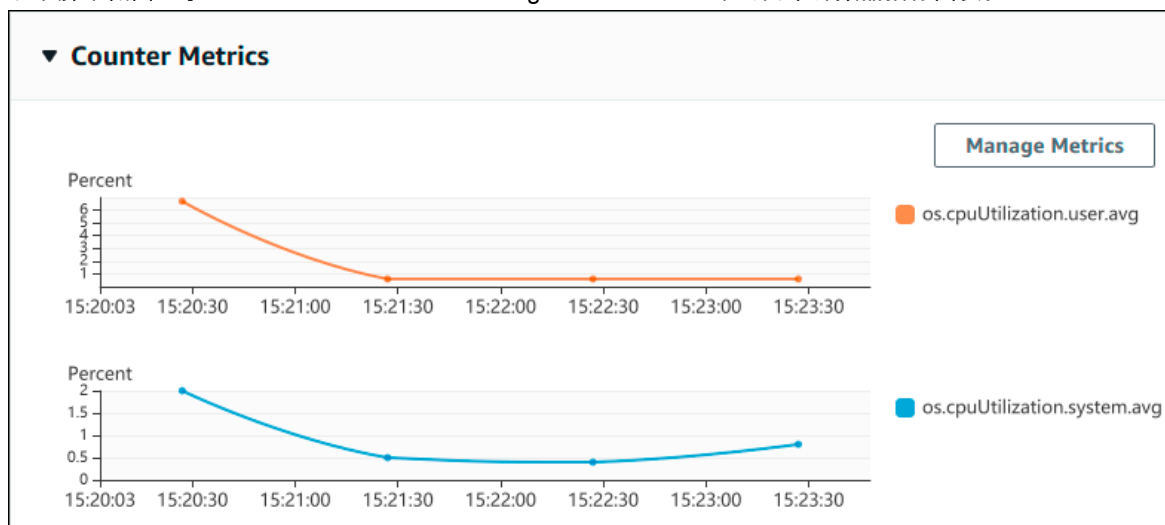
以下示例显示了如何使用 Amazon CLI 获取性能详情。

主题

- [检索计数器指标 \(p. 419\)](#)
- [检索首要等待状态的数据库负载平均值 \(p. 421\)](#)
- [检索首要查询的数据库负载平均值 \(p. 423\)](#)
- [检索按查询筛选的数据库负载平均值 \(p. 425\)](#)

检索计数器指标

以下屏幕截图显示 Amazon Web Services Management Console 中的两个计数器指标图表。



以下示例显示如何收集 Amazon Web Services Management Console 用于生成两个计数器指标图表的相同数据。

对于 Linux、macOS 或 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"}]'
```

对于 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"}]'
```

还可以通过为 `--metrics-query` 选项指定文件来使命令更易于读取。以下示例为该选项使用名为 `query.json` 的文件。此文件具有以下内容。

```
[
  {
    "Metric": "os.cpuUtilization.user.avg"
  },
  {
    "Metric": "os.cpuUtilization.idle.avg"
  }
]
```

运行以下命令来使用此文件。

对于 Linux、macOS 或 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
```

对于 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
```

上一个示例为各选项指定了以下值：

- `--service-type=DOCDB` 对于 Amazon DocumentDB
- `--identifier` – 数据库实例的资源 ID
- `--start-time` 和 `--end-time` – 要查询的期间的 ISO 8601 DateTime 值，支持多种格式

它查询一小时时间范围：

- `--period-in-seconds` – 对于每分钟查询来说为 60
- `--metric-queries` – 两个查询的数组，每个查询只用于一个指标。

指标名称使用点在有用的类别中分类指标，最后一个元素是函数。在示例中，对于每个查询来说，此函数是 `avg`。与 Amazon CloudWatch 一样，支持的函数为 `min`、`max`、`total` 和 `avg`。

响应类似于以下内容。

```
{
  "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
```

响应具有 Identifier、AlignedStartTime 和 AlignedEndTime。但 --period-in-seconds 值为 60，开始和结束时间已与分钟对齐。如果 --period-in-seconds 为 3600，则开始和结束时间已与小时对齐。

响应中的 MetricList 具有许多条目，每个条目具有 Key 和 DataPoints 条目。每个 DataPoint 具有 Timestamp 和 Value。每个 Datapoints 列表具有 60 个数据点，因为查询针对一小时内的每分钟数据，具有 Timestamp1/Minute1、Timestamp2/Minute2 等，一直到 Timestamp60/Minute60。

因为查询用于两个不同的计数器指标，响应 MetricList 中有两个元素。

检索首要等待状态的数据库负载平均值

以下示例是 Amazon Web Services Management Console 用于生成堆栈区域线图的相同查询。此示例检索 db.load.avg 根据前七个等待状态划分负载的最后一个小时。命令与 [检索计数器指标 \(p. 419\)](#) 中的命令相同。不过，query.json 文件具有以下内容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_state", "Limit": 7 }
  }
]
```

运行以下命令。

对于 Linux、macOS 或 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
```

对于 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
```

该示例指定的指标db.load.avg和GroupBy在前七个等待状态中。有关此示例的有效值的详细信息，请参阅 Performance Insights API 参考 中的 [DimensionGroup](#)。

响应类似于以下内容。

```
{
  "AlignedStartTime": "2022-04-04T06:00:00+00:00",
  "AlignedEndTime": "2022-04-04T06:15:00+00:00",
  "Identifier": "db-NQF3TTFQ3GTOKIMJODMC3KQQ4",
  "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
```

在此响应中，中有三个条目MetricList。总共有一个条目db.load.avg，以及各有三个条目db.load.avg根据前三个等待状态之一划分。由于存在分组维度（与第一个示例不同），所以必须具有一个用于每个指标分组的键。不能像在基本计数器指标使用案例中那样每个指标只有一个键。

检索首要查询的数据库负载平均值

以下示例组db.wait_state按前 10 个查询语句。有两个不同的查询语句组：

- db.query— 完整的查询语句，例如{"find":"customers","filter":{"FirstName":"Jesse"},"sort":{"key":{"\$numberInt":"1"}}}
- db.query_tokenized— 令牌化的查询语句，例如{"find":"customers","filter":{"FirstName":"?"},"sort":{"key":{"\$numberInt":"?"}},"limit":{"\$numberInt":"?"}}

在分析数据库性能时，将仅参数不同的查询语句视为一个逻辑项目很有用。因此，您在查询时可以使用db.query_tokenized。但是，尤其是当您感兴趣的时候explain()，有时使用参数检查完整的查询语句更有用。令牌化和完整查询之间存在父-子关系，多个完整查询（子级）分组在同一令牌化查询（父级）下。

此示例中的命令类似于[检索首要等待状态的数据库负载平均值 \(p. 421\)](#) 中的命令。不过，query.json 文件具有以下内容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.query_tokenized", "Limit": 10 }
  }
]
```

下面的示例使用了 db.query_tokenized。

对于 Linux、macOS 或 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
```

对于 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
```

此示例查询 1 小时，以秒为单位的期间为单位的 1 分钟。

该示例指定的指标 `db.load.avg` 和 `GroupBy` 在前七个等待状态中。有关此示例的有效值的详细信息，请参阅 [Performance Insights API 参考](#) 中的 [DimensionGroup](#)。

响应类似于以下内容。

```
{
  "AlignedStartTime": "2022-04-04T06:00:00+00:00",
  "AlignedEndTime": "2022-04-04T06:15:00+00:00",
  "Identifier": "db-NQF3TTFQ3GTOKIMJODMC3KQQ4",
  "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
```

此回复中有 11 个条目 `MetricList` (1 个总计，10 个首要令牌化查询)，其中每个条目具有 24 个每小时 `DataPoints`。

对于令牌化查询，每个维度列表中具有三个条目：

- `db.query_tokenized.statement` — 令牌化的查询语句。
- `db.query_tokenized.db_id` — Performance Insights 为您生成的综合 ID。此示例返回 `pi-1064184600` 合成 ID。
- `db.query_tokenized.id` — Performance Insights 中的查询的 ID。

在 Amazon Web Services Management Console 中，此 ID 称为支持 ID。它如此命名是因为 ID 是 Amazon Support 可检查以帮助解决数据库问题的数据。Amazon 极其重视您的数据的安全性和隐私，几乎所有数据都使用您的 Amazon KMS 客户主密钥 (CMK) 进行加密存储。因此，Amazon 中的任何人都无法查看这些数据。在上一个示例中，`tokenized.statement` 和 `tokenized.db_id` 都进行了加密存储。如果您的数据库出现问题，Amazon Support 可以通过引用支持 ID 来帮助您。

在查询时，在 `Group` 中指定 `GroupBy` 可能很方便。不过，要更精细地控制返回的数据，请指定维度列表。例如，如果所需的所有内容是 `db.query_tokenized.statement`，则可将 `Dimensions` 属性添加到 `query.json` 文件中。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": {
      "Group": "db.query_tokenized",
      "Dimensions": ["db.query_tokenized.statement"],
      "Limit": 10
    }
  }
]
```

检索按查询筛选的数据库负载平均值

此示例中的相应 API 查询类似于 [检索首要查询的数据库负载平均值 \(p. 423\)](#) 中的命令。不过，query.json 文件具有以下内容。

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_state", "Limit": 5 },
    "Filter": { "db.query_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
  }
]
```

在此响应中，根据 query.json 文件中指定的令牌化查询 AKIAIOSFODNN7EXAMPLE 的贡献筛选所有值。AKIAIOSFODNN7EXAMPLE 键还可能遵循与没有筛选条件的查询不同的顺序，因为前五个等待状态影响了筛选的查询。

亚马逊 CloudWatch Performance Insights 的指标

Performance Insights 会自动将指标发布到 Amazon CloudWatch。可以从 Performance Insights 查询相同的数据，但具有中的指标 CloudWatch 使添加变得容易 CloudWatch 警报。还可以轻松地将指标添加到现有 CloudWatch 控制面板中。

指标	描述
DBLoad	Amazon DocumentDB 的活动会话的数量。通常，您需要活动会话的平均数量数据。在 Performance Insights 中，作为 db.load.avg 查询此数据。
DBLoadCPU	等待状态类型为 CPU 的活动会话的数量。在 Performance Insights 中，查询此数据的方式为 db.load.avg，按等待状态类型筛选 CPU。
DBLoadNonCPU	等待状态类型不为 CPU 的活动会话的数量。

Note

这些指标发布到 CloudWatch 仅当数据库实例上有负载时。

您可以使用检查这些指标 CloudWatch 控制台，Amazon CLI，或 CloudWatch API。

例如，可以通过运行 [get-metric-statistics](#) 命令来获取 DBLoad 指标的统计数据。

```
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
```

该示例将生成与下类似的输出。

```
{
  "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"
}
```

有关 CloudWatch 的更多信息，请参阅[什么是 Amazon CloudWatch ?](#) 中的亚马逊 CloudWatch 用户指南。

Performance Insights 的计数器指标

计数器指标是 Performance Insights 控制面板中的操作系统指标。为帮助确定和分析性能问题，您可将计数器指标与数据库负载相关联。

性能详情操作系统计数器

以下操作系统计数器可用于 DocumentDB Performance Insights。

计数器	类型	指标
active	memory	os.memory.active

计数器	类型	指标
buffers	memory	os.memory.buffers
cached	memory	os.memory.cached
dirty	memory	os.memory.dirty
free	memory	os.memory.free
inactive	memory	os.memory.inactive
mapped	memory	os.memory.mapped
pageTables	memory	os.memory.pageTables
slab	memory	os.memory.slab
total	memory	os.memory.total
writeback	memory	os.memory.writeback
idle	cpuUtilization	os.cpuUtilization.idle
system	cpuUtilization	os.cpuUtilization.system
total	cpuUtilization	os.cpuUtilization.total
user	cpuUtilization	os.cpuUtilization.user
wait	cpuUtilization	os.cpuUtilization.wait
one	loadAverageMinute	os.loadAverageMinute.one
fifteen	loadAverageMinute	os.loadAverageMinute.fifteen
five	loadAverageMinute	os.loadAverageMinute.five
cached	swap	os.swap.cached
free	swap	os.swap.free
in	swap	os.swap.in
out	swap	os.swap.out
total	swap	os.swap.total
rx	network	os.network.rx
tx	network	os.network.tx
numVCPUs	general	os.general.numVCPUs

使用 Amazon DocumentDB 进行开发

这些部分涵盖使用 Amazon DocumentDB 进行开发 (与 MongoDB 兼容)。

主题

- [以编程方式连接到 Amazon DocumentDB \(p. 428\)](#)
- [在 Amazon DocumentDB 中使用变更流 \(p. 445\)](#)
- [作为副本集连接到 Amazon DocumentDB \(p. 454\)](#)
- [从 Amazon VPC 外部连接到 Amazon DocumentDB 集群 \(p. 457\)](#)
- [从 Robo 3T 连接到 Amazon DocumentDB 集群 \(p. 458\)](#)
- [从 Studio 3T 连接到 Amazon DocumentDB 集群 \(p. 461\)](#)
- [使用 Amazon EC2 Connect \(p. 466\)](#)
- [使用 Amazon DocumentDB JDBC 驱动程序进行 Connect \(p. 480\)](#)

以编程方式连接到 Amazon DocumentDB

本部分包含说明了如何使用多种不同语言连接到 Amazon DocumentDB (兼容 MongoDB) 的代码示例。根据连接的集群是否启用传输层安全性 (TLS)，这些示例分为两个部分。默认情况下，Amazon DocumentDB 集群启用了 TLS。但是，您可以根据需要关闭 TLS。有关更多信息，请参阅[加密传输中的数据 \(p. 136\)](#)。

如果您想要从集群所在的 VPC 之外连接到 Amazon DocumentDB，请参阅[从 Amazon VPC 外部连接到 Amazon DocumentDB 集群 \(p. 457\)](#)。

在连接到集群之前，您必须知道集群是否启用了 TLS。下一部分介绍如何使用 Amazon Web Services Management Console 或 Amazon CLI 确定集群的 `tls` 参数的值。之后，您可以查找和应用适当的代码示例。

主题

- [确定 tls 参数的值 \(p. 428\)](#)
- [启用了 TLS 的情况下的连接 \(p. 430\)](#)
- [禁用了 TLS 的情况下的连接 \(p. 439\)](#)

确定 tls 参数的值

确定集群是否启用了 TLS 是一个两步过程，您可以使用 Amazon Web Services Management Console 或 Amazon CLI 执行该过程。

1. 确定管理集群的参数组。

使用 Amazon Web Services Management Console

1. 登录到 Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台 <https://console.aws.amazon.com/docdb>。
2. 在左侧导航窗格中，选择集群。
3. 在集群列表中，选择您的集群的名称。
4. 生成的页面将显示所选集群的详细信息。向下滚动到 Cluster details (集群详细信息)。在此部分的底部，在 Cluster parameter group (集群参数组) 的下方找到参数组的名称。

使用 Amazon CLI

使用以下 Amazon CLI 代码可以确定管理您的集群的参数。请确保将 `sample-cluster` 替换为您的集群的名称。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

此操作的输出将类似于以下内容：

```
[
  [
    "sample-cluster",
    "sample-parameter-group"
  ]
]
```

2. 确定您的集群参数组中的 `tls` 参数的值。

使用 Amazon Web Services Management Console

1. 在导航窗格中，选择参数组。
2. 在集群参数组窗口中，选择您的集群参数组。
3. 打开的页面上会显示您的集群参数组中包含的参数。您可以在其中查看 `tls` 参数的值。有关修改此参数的信息，请参阅[修改 Amazon DocumentDB 集群参数组 \(p. 328\)](#)。

使用 Amazon CLI

您可以使用 `describe-db-cluster-parameters` Amazon CLI 命令来查看集群参数组中的参数的详细信息。

- **--describe-db-cluster-parameters**— 列出参数组中的所有参数及其值。
- **--db-cluster-parameter-group name** – 必需。您的集群参数组的名称。

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于以下内容：

```
{
  "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",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot"
    }
  ]
}
```

确定 `tls` 参数的值后，即可使用以下部分中的代码示例之一继续连接到您的集群。

- [启用了 TLS 的情况下的连接 \(p. 430\)](#)
- [禁用了 TLS 的情况下的连接 \(p. 439\)](#)

启用了 TLS 的情况下的连接

要查看以编程方式连接到启用了 TLS 的 Amazon DocumentDB 集群的代码示例，请选择您要使用的语言所对应的选项卡。

要加密传输中的数据，请下载名为的 Amazon DocumentDB 的公钥。 `rds-combined-ca-cn-bundle.pem` 使用以下操作。

```
wget https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem
```

Python

以下代码说明了如何在启用了 TLS 的情况下使用 Python 连接到 Amazon DocumentDB。

```
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=rds-combined-ca-
cn-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred')

##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

以下代码说明了如何在启用了 TLS 的情况下使用 Node.js 连接到 Amazon DocumentDB。

```
var MongoClient = require('mongodb').MongoClient,
    f = require('util').format,
    fs = require('fs');

//Specify the Amazon DocumentDB cert
var ca = [fs.readFileSync("rds-combined-ca-cn-bundle.pem")];

//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?ssl=true&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
  {
    sslValidate: true,
    sslCA:ca,
    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

以下代码说明了如何在启用了 TLS 的情况下使用 PHP 连接到 Amazon DocumentDB。

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

$SSL_DIR = "/home/ubuntu";
$SSL_FILE = "rds-combined-ca-cn-bundle.pem";

//Specify the Amazon DocumentDB cert
$ctx = stream_context_create(array(
  "ssl" => array(
    "cafile" => $SSL_DIR . "/" . $SSL_FILE,
  ))
```



```
);

//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", array("ssl" =>
true), array("context" => $ctx));

//Specify the database and collection to be used
$col = $client->sampldatabase->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

以下代码说明了如何在启用了 TLS 的情况下使用 Go 连接到 Amazon DocumentDB。

Note

从版本 1.2.1 开始，MongoDB Go 驱动程序将仅使用在 `sslcertificateauthorityfile` 中找到的第一个 CA 服务器证书。以下代码示例通过将 `sslcertificateauthorityfile` 中找到的所有服务器证书手动附加到在创建客户端期间使用的自定义 TLS 配置来解决此限制。

```
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 = "rds-combined-ca-cn-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

从 Java 应用程序连接到启用了 TLS 的 Amazon DocumentDB 集群时，您的程序必须使用 Amazon 提供了证书颁发机构 (CA) 文件以验证连接。要使用 Amazon RDS CA 证书，请执行以下操作：

1. 从下载 Amazon RDS CA 文件 <https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem>.
2. 通过执行以下命令，使用该文件中包含的 CA 证书来创建信任存储。请务必将 `<truststorePassword>` 更改为其他内容。如果您要访问同时包含旧 CA 证书 (rds-ca-2015-root.pem) 和新 CA 证书 (rds-ca-2019-root.pem) 的信任存储，可以将证书捆绑包导入该信任存储。

下面是一个示例 Shell 脚本，它将证书捆绑包导入 Linux 操作系统上的信任存储。

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem" > ${mydir}/rds-combined-ca-cn-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/ {split_after=1}{print > "rds-ca-" n ".pem"}' < ${mydir}/rds-combined-ca--cn-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}/rds-combined-ca--cn-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
```

下面是一个示例 Shell 脚本，它将证书捆绑包导入 macOS 上的信任存储。

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-
bundle.pem" > ${mydir}/rds-combined-ca-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/rds-combined-ca-cn-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}/rds-combined-ca-cn-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. 使用keystore请先在您的应用程序中设置以下系统属性，以便在该程序中连接到 Amazon DocumentDB 集群。

```
javax.net.ssl.trustStore: <truststore>
javax.net.ssl.trustStorePassword: <truststorePassword>
```

4. 以下代码说明了如何在启用了 TLS 的情况下使用 Java 连接到 Amazon DocumentDB。

```
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 = MongoClient.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

以下代码说明了如何在启用了 TLS 的情况下使用 C#/.NET 连接到 Amazon DocumentDB。

```
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}/sample-database?
ssl=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_to_rds-combined-ca-cn-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("sample-database");
        var collection = database.GetCollection<BsonDocument>("sample-collection");
        var docToInsert = new BsonDocument { { "pi", 3.14159 } };
        collection.InsertOne(docToInsert);
    }
}
```

mongo shell

以下代码说明了如何在启用了 TLS 的情况下使用 mongo shell 连接和查询 Amazon DocumentDB。

1. 使用 mongo shell Connect 到 Amazon DocumentDB。如果你使用的是 4.2 之前的 mongo shell 版本，请使用以下代码进行连接。

```
mongo --ssl --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017
--sslCAFile rds-combined-ca-cn-bundle.pem --username <sample-user> --
password <password>
```

如果你使用的是等于或大于 4.2 的 mongo shell 版本，请使用以下代码进行连接。

```
mongo --tls --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017
--tlsCAFile rds-combined-ca-cn-bundle.pem --username <sample-user> --
password <password>
```

2. 插入单个文档。

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. 查找以前插入的文档。

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

以下代码说明了如何使用蒙古人使用 R 连接到 Amazon DocumentDB。 <https://jeroen.github.io/mongolite/>) 启用了 TLS 的情况下。

```
#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/rds-combined-ca-bundle.pem>"))

#Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)

#Find the document that was previously written
client$find()
```

Ruby

以下代码说明了如何在启用了 TLS 的情况下使用 Ruby 连接到 Amazon DocumentDB。

```
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/rds-combined-ca-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
```

禁用了 TLS 的情况下的连接

要查看以编程方式连接到禁用了 TLS 的 Amazon DocumentDB 集群的代码示例，请选择您要使用的语言所对应的选项卡。

Python

以下代码说明了如何在禁用了 TLS 的情况下使用 Python 连接到 Amazon DocumentDB。

```
## 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

以下代码说明了如何在禁用了 TLS 的情况下使用 Node.js 连接到 Amazon DocumentDB。

```
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

以下代码说明了如何在禁用了 TLS 的情况下使用 PHP 连接到 Amazon DocumentDB。

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoClient("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->sampldatabase->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

以下代码说明了如何在禁用了 TLS 的情况下使用 Go 连接到 Amazon DocumentDB。

```
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

以下代码说明了如何在禁用了 TLS 的情况下使用 Java 连接到 Amazon DocumentDB。

```
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

以下代码说明了如何在禁用了 TLS 的情况下使用 C#/.NET 连接到 Amazon DocumentDB。

```
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

以下代码说明了如何在禁用了 TLS 的情况下使用 mongo shell 连接和查询 Amazon DocumentDB。

1. 使用 mongo shell Connect 到 Amazon DocumentDB。

```
mongo --host mycluster.node.us-east-1.docdb.amazonaws.com:27017 --username <sample-
user> --password <password>
```

2. 插入单个文档。

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. 查找以前插入的文档。

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

以下代码说明了如何使用蒙古人使用 R 连接到 Amazon DocumentDB。 <https://jeroen.github.io/mongolite/> 当 TLS 被禁用时。

```
#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

以下代码说明了如何在禁用了 TLS 的情况下使用 Ruby 连接到 Amazon DocumentDB。

```
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/rds-combined-ca-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
```

```
end

#Close the connection
client.close
```

在 Amazon DocumentDB 中使用变更流

Amazon DocumentDB 中的变更流功能（兼容 MongoDB）提供按时间顺序排列的更改事件，这些事件在您的集群集合中发生。您可以从变更流中读取事件，实现许多不同的使用案例，包括以下情况：

- 更改通知
- Amazon 全文搜索 OpenSearchService (OpenSearchService)
- Amazon Redshift 分析

应用程序可以使用变更流在各个集合中订阅数据变更。变更流事件在集群上发生时按顺序排列，并在记录事件之后存储 3 个小时（默认情况下）。保留期可以延长至 7 天，使用 `change_stream_log_retention_duration` 参数。要修改变更流的保留期限，请参阅[修改更改流日志保留持续时间](#)。

主题

- [支持的 操作 \(p. 445\)](#)
- [计费 \(p. 445\)](#)
- [限制 \(p. 446\)](#)
- [启用变更流 \(p. 446\)](#)
- [例如：在 Python 中使用变更流 \(p. 447\)](#)
- [完整文档查找 \(p. 449\)](#)
- [恢复变更流 \(p. 449\)](#)
- [使用恢复变更流 `startAtOperationTime` \(p. 451\)](#)
- [变更流中的交易 \(p. 452\)](#)
- [修改更改流日志保留持续时间 \(p. 452\)](#)

支持的 操作

Amazon DocumentDB 支持更改流的以下操作：

- MongoDB 中支持的所有变更事件 `db.collection.watch()`、`db.watch()` 和 `client.watch()` API。
- 完整的文档查找以获取更新。
- 聚合阶段：`$match`、`$project`、`$redact`，和 `$addFields` 和 `$replaceRoot`。
- 从简历令牌恢复更改流
- 使用从时间戳恢复更改流 `startAtOperation`（适用于 Amazon DocumentDB v4.0+）

计费

默认情况下，Amazon DocumentDB 变更流功能处于禁用状态，并且在启用该功能之前不会产生任何额外费用。在集群中使用变更流时，会产生额外的读写 I/O 和存储成本。您可以使用 `modifyChangeStreamsAPI` 操作为您的集群启用此功能。有关定价的更多信息，请参阅[Amazon DocumentDB 定价](#)。

限制

更改流在 Amazon DocumentDB 中有以下限制：

- 变更流只能通过与 Amazon DocumentDB 集群主实例的连接打开。当前不支持从副本实例上的变更流中进行读取。在调用 `watch()` API 操作时，您必须指定 **primary** 读取首选项，以确保所有读取都定向到主实例（请参阅[示例 \(p. 447\)](#)部分）。
- 写入集合的变更流的事件最多可在 7 天内使用（默认为 3 小时）。变更流数据将在日志保留时段过后删除，即使没有发生新更改也是如此。
- 对集合执行长时间的写操作，例如 `updateMany` 要么 `deleteMany` 可以暂时延迟变更流事件的写入，直至长时间运行的写入操作完成为止。
- Amazon DocumentDB 不支持 MongoDB 操作日志 (oplog)。
- 使用 Amazon DocumentDB，您必须明确在给定集合上启用变更流。
- 如果变更流事件的总大小（包括变更数据，在请求的情况下还包括完整文档）大于 16 MB，客户端将在变更流上遇到读取失败情况。
- 使用时目前不支持 Ruby 驱动程序 `db.watch()` 和 `client.watch()` 使用 Amazon DocumentDB v3.6。

启用变更流

您可以为给定数据库中的所有集合启用 Amazon DocumentDB 变更流，也可以仅针对选定集合启用。以下示例说明如何使用 mongo shell 为不同使用案例启用变更流。指定数据库和集合名称时，会将空字符串视为通配符。

```
//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});
```

如果满足以下任意条件，则将为集合启用变更流：

- 数据库和集合都已显式启用。

- 已启用包含该集合的数据库。
- 启用所有数据库。

如果父数据库也启用了更改流，或者如果群集中的所有数据库都已启用，则从数据库中删除集合不会禁用该集合的更改流。如果创建的新收藏夹与已删除的集合名称相同，则将为该集合启用更改流。

您可以使用 `$listChangeStreams` 聚合管道阶段。Amazon DocumentDB 支持的所有聚合阶段都可以在管道中使用进行额外的处理。如果以前启用的某个集合被禁用，则该集合将不会显示在 `$listChangeStreams` 输出中。

```
//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: {}}));
```

例如：在 Python 中使用变更流

以下是在集合级别将 Amazon DocumentDB 更改流与 Python 结合使用的示例。

```
import os
import sys
from pymongo import MongoClient, ReadPreference

username = "DocumentDBusername"
password = <Insert your password>

clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, ssl='true',
    ssl_ca_certs='rds-combined-ca-cn-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': '015daf99d4000000001010000000100009025'},
'clusterTime': Timestamp(1571789268, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 2}}}
"""
```

以下是在数据库级别将 Amazon DocumentDB 更改流与 Python 结合使用的示例。

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, ssl='true',
    ssl_ca_certs='rds-combined-ca-cn-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'}
"""
```

完整文档查找

更新变更事件不包括完整文档；它只包括已执行的更改。如果您的使用案例需要用到受更新影响的完整文档，则可以在打开流时启用完整文档查找。

更新变更流事件的 `fullDocument` 文档会指明文档查找时已更新文档的最新版本。如果在更新操作和 `fullDocument` 查找，`fullDocument` 更新时，文档可能无法指明文档状态。

```
#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': '015daf9b7c00000001010000000100009025'},
'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}}}
```

恢复变更流

您可以在以后通过使用恢复令牌来恢复变更流，该令牌相当于上次检索的变更事件文档的 `_id` 字段。

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, ssl='true',
    ssl_ca_certs='rds-combined-ca-cn-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': '015daf9c5b00000001010000000100009025'}
"""

#Python provides a nice shortcut for getting a stream's resume token
print(stream.resume_token)

"""
Output
{'_data': '015daf9c5b00000001010000000100009025'}
"""

#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': '015f7e8f0c000000060100000006000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602129676, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0ac423bafb9adba2')},
'fullDocument': {'_id': ObjectId('5f7e8f0ac423bafb9adba2'), 'x': 5},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""

#Followed by the insert
print(stream.try_next())

"""
#Output:
{'_id': {'_data': '015f7e8f0c000000070100000007000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602129676, 7),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94')},
'fullDocument': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94'), 'y': 5}}
"""
```

使用恢复变更流startAtOperationTime

以后您可以通过使用特定时间戳恢复更改流startAtOperationTime。

Note

使用的能力startAtOperationTime在 Amazon DocumentDB 4.0+ 中提供。使用startAtOperationTime，更改流光标将只返回在指定时间戳之后或之后发生的更改。这些区域有：startAtOperationTime和resumeAfter命令是互相排斥的，因此不能一起使用。

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint"
client = MongoClient(clusterendpoint, username=username, password=password, ssl='true',
    ssl_ca_certs='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': '015f7e941a000000030100000003000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602130970, 3),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e9417c423bafb9d9adbb1')},
'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('5f7e94fffc423bafb9d9adbb2')},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
```

```
"""
print(stream.try_next())

"""
#Output: And finally the last event will be the insert operation (y:5)
{'_id': {'_data': '015f7e95020000006010000006000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602131202, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e95025c4a569e0f6dde92')},
'fullDocument': {'_id': ObjectId('5f7e95025c4a569e0f6dde92'), 'y': 5}}
"""
```

变更流中的交易

更改流事件将不包含来自未提交和/或已中止的事务的事件。例如，如果你用一个开始交易INSERT操作还有一个UPDATE操作和。如果您的INSERT操作成功了，但UPDATE操作失败，事务将被回滚。由于此交易已回退，因此您的更改流将不包含此交易的任何事件。

修改更改流日志保留持续时间

您可以使用Amazon Web Services Management Console或者Amazon CLI。

使用 Amazon Web Services Management Console

修改更改流日志保留期限的步骤

1. 登录到Amazon Web Services Management Console，然后在以下位置打开 Amazon DocumentDB 控制台<https://console.aws.amazon.com/docdb>。
2. 在导航窗格中，选择参数组。

Tip

如果您在屏幕左侧没有看到导航窗格，请在页面左上角选择菜单图标
(≡)。

3. 在参数组窗格中，选择与您的集群关联的集群参数组。要确定与您的集群关联的集群参数组，请参阅[确定 Amazon DocumentDB 集群的参数组 \(p. 325\)](#)。
4. 打开的页面上会显示您的集群参数组中包含的参数及其相应详细信息。选择 `change_stream_log_retention_duration` 参数。
5. 在页面右上角，选择编辑以更改参数的值。这些区域
有：`change_stream_log_retention_duration`参数可以修改为 1 到 7 天之间。
6. 进行更改，然后选择修改集群参数保存更改。要放弃更改，请选择Cancel。

使用 Amazon CLI

修改您的集群参数组`change_stream_log_retention_duration`参数，请使用`modify-db-cluster-parameter-group`按照以下参数进行操作：

- **--db-cluster-parameter-group-name** – 必需。您正在修改的集群参数组的名称。要确定与您的集群关联的集群参数组，请参阅[确定 Amazon DocumentDB 集群的参数组 \(p. 325\)](#)。
- **--parameters** – 必需。您正在修改的参数。每个参数条目必须包含以下内容：
 - **ParameterName**— 您正在修改的参数的名称。在这种情况下，它是`change_stream_log_retention_duration`

- **ParameterValue**— 此参数的新值。
- **ApplyMethod**— 您希望如何对应用的参数进行更改。允许的值为 `immediate` 和 `pending-reboot`。

Note

带 `static` 的 `ApplyType` 参数必须具有 `pending-reboot` 的 `ApplyMethod`。

1. 更改参数的值 `change_stream_log_retention_duration`，运行以下命令并替换 `parameter-value` 其中包含要修改参数的值。

对于 Linux、macOS 或 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"
```

对于 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"
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "DBClusterParameterGroupName": "sample-parameter-group"  
}
```

2. 等至少 5 分钟。
3. 列出的参数值 `sample-parameter-group` 以确保您的更改已完成。

对于 Linux、macOS 或 Unix：

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name sample-parameter-group
```

对于 Windows：

```
aws docdb describe-db-cluster-parameters ^  
  --db-cluster-parameter-group-name sample-parameter-group
```

此操作的输出将类似于下文（JSON 格式）。

```
{  
  "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"
    }
]
}
```

作为副本集连接到 Amazon DocumentDB

在针对 Amazon DocumentDB 进行开发时（具有 MongoDB 兼容性），我们建议您以副本集形式连接到集群，并使用驱动程序的内置读取首选项功能将读取操作分布到副本实例。本节更深入地介绍了这意味着什么，并介绍了如何使用适用于 Python 的开发工具包作为副本集连接到 Amazon DocumentDB 集群作为示例。

Amazon DocumentDB 有三个终端节点，您可以使用它们连接到集群：

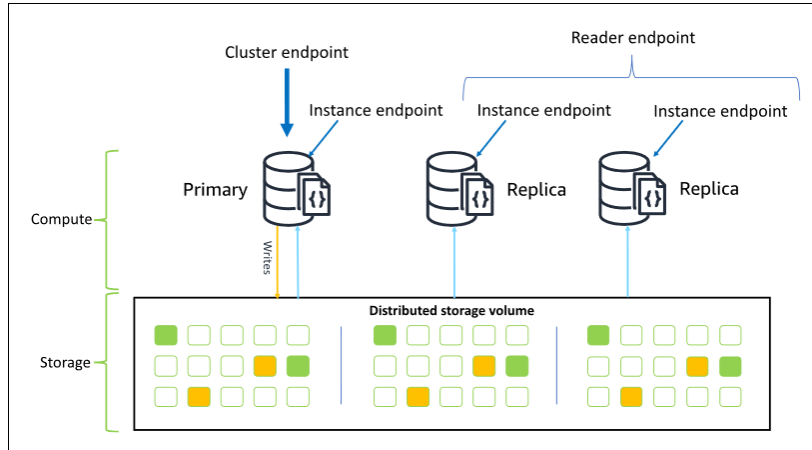
- 集群终端节点
- 读取器终端节点
- 实例终端节点

在大多数情况下，在连接到 Amazon DocumentDB 时，我们建议您使用集群终端节点。这是指向集群中的主实例的别名记录，如下图所示。

使用 SSH 隧道时，我们建议您使用集群终端节点连接到集群，而不要尝试以副本集模式（即在连接字符串中指定 `replicaSet=rs0`）进行连接，因为这会导致错误。

Note

有关 Amazon DocumentDB 终端节点的更多信息，请参阅 [Amazon DocumentDB D \(p. 7\)](#)。



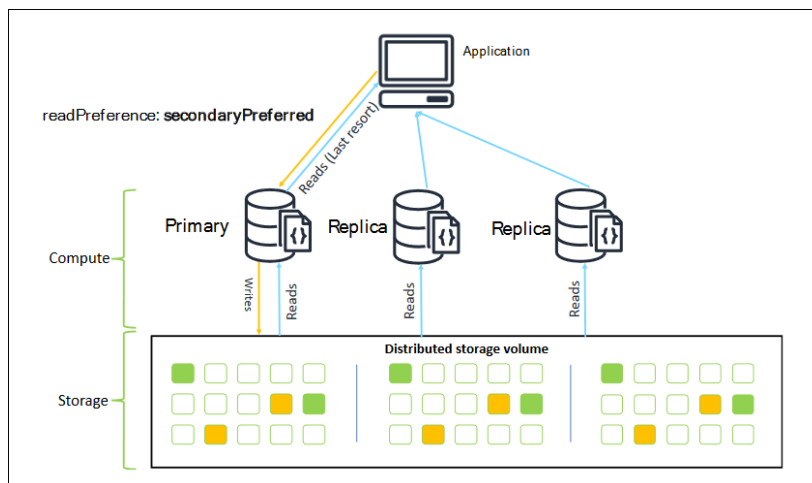
使用集群终端节点，您可以在副本集模式下连接到集群。然后您可以使用内置的读取首选项驱动程序功能。在以下示例中，指定/?replicaSet=rs0表示您要以副本集形式连接到的开发工具包。如果你省略/?replicaSet=rs0'，客户端将所有请求路由到集群终端节点，即您的主实例。

```
## 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')
```

作为副本集进行连接的优势在于，它使您的 SDK 能够自动发现集群地形，包括在群集中添加或移除实例时。然后，您可以通过将读取请求路由到副本实例来更有效地使用集群。

当您作为副本集进行连接时，可以指定readPreference连接。如果指定了读取首选项secondaryPreferred，客户端会将读取查询路由到副本并将查询写入主实例（如下图所示）。这是对群集资源的更好利用。有关更多信息，请参阅[读取首选项选项](#) (p. 11)。

```
## 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')
```



对 Amazon DocumentDB 副本的读取操作具有最终一致性。他们返回数据的顺序与在主数据上写入的顺序相同，而且复制滞后通常不到 50 毫秒。您可以使用 Amazon CloudWatch 指标监控集群的副本

滞后。DBInstanceReplicaLag和DBClusterReplicaLagMaximum。有关更多信息，请参阅[使用 CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)。

与传统的单片数据库体系结构不同，Amazon DocumentDB 将存储和计算分开。鉴于这种现代体系结构，我们鼓励您在副本实例上读取扩展。副本实例上的读取不会阻止从主实例复制写入操作。您可以在群集中添加多达 15 个只读副本实例，并扩展到每秒数百万次读取。

作为副本集进行连接并将读取分发到副本的关键好处是，它增加了群集中可用于应用程序的总体资源。作为最佳实践，我们建议您以副本集形式连接。此外，我们最常建议在以下情况下使用：

- 你在主机上使用了近 100% 的 CPU。
- 缓冲区缓存命中率接近零。
- 您已达到单个实例的连接或光标限制。

扩展集群实例大小是一种选择，在某些情况下，这可能是扩展集群的最佳方法。但是，您还应考虑如何更好地使用集群中已有的副本。这使您可以在不增加使用更大实例类型的成本的情况下增加规模。我们还建议您监控这些限制并提醒这些限制（即CPUUtilization、DatabaseConnections，和BufferCacheHitRatio）使用 CloudWatch 警报，以便您知道资源何时被大量使用。

有关更多信息，请参阅以下主题：

- [Amazon DocumentDB 的最佳实践 \(p. 78\)](#)
- [Amazon DocumentDB 配额和限制 \(p. 496\)](#)

使用群集连接

请考虑使用集群中所有连接的情况。例如，r5.2xlarge实例的连接数限制为 4,500（以及 450 个打开的游标）。如果您创建了三个实例的 Amazon DocumentDB 集群，并且只使用集群终端节点连接到主实例，则打开的连接和游标的集群限制分别为 4,500 和 450。如果您正在构建的应用程序要使用许多在容器中启动的工作程序，那么可能会达到这些限制。容器同时打开多个连接，使集群饱和。

相反，您可以作为副本集连接到 Amazon DocumentDB 集群，然后将读取内容分发到副本实例。然后，您可以有效地将群集中可用连接和游标的数量增加三倍，分别为 13,500 和 1,350 个。向群集添加更多实例只会增加读取工作负载的连接和游标数量。如果您需要增加写入集群的连接数量，我们建议您增加实例大小。

Note

的连接数large、xlarge, 和2xlarge实例随着实例大小高达 4,500 个而增加。每个实例的最大连接数4xlarge实例或更高的实例为 4,500 个。有关实例类型限制的更多信息，请参阅[实例限制 \(p. 500\)](#)。

一般而言，我们不建议使用 secondary 的读取首选项连接到集群。这是因为如果集群中没有副本实例，则读取操作将失败。例如，假设您具有两个实例的 Amazon DocumentDB 集群，其中包含一个主副本和一个副本。如果副本有问题，请从设置为的连接池中读取请求secondary失败。的优势secondaryPreferred是，如果客户端找不到要连接的合适的副本实例，它会回到主实例进行读取。

多个连接池

在某些情况下，应用程序中的读取需要具有写后读一致性，这只能从 Amazon DocumentDB 中的主实例提供。在这些情况下，您可以创建两个客户端连接池：一个用于写入，另一个用于需要后写一致性的读取。要做到这一点，您的代码将类似于下文。

```
## 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')
```

另一种选择是创建单个连接池并覆盖给定集合的读取首选项。

```
##Specify the collection and set the read preference level for that collection
col = db.review.with_options(read_preference=ReadPreference.SECONDARY_PREFERRED)
```

摘要

为了更好地使用集群中的资源，我们建议您使用副本集模式连接到您的集群。如果它适合您的应用程序，则可以通过将读取内容分发到副本实例来读取扩展应用程序。

从 Amazon VPC 外部连接到 Amazon DocumentDB 集群

Amazon DocumentDB (兼容 MongoDB) 集群在 Amazon Virtual Private Cloud (Amazon VPC) 中部署。它们可以由 Amazon EC2 实例或其他实例直接访问 Amazon 部署在同一 Amazon VPC 中的服务。此外，EC2 实例或其他实例可以访问 Amazon DocumentDB Amazon 在同一 VPC 中使用的服务 Amazon Web Services 区域或者通过 VPC 对等互连进行其他区域。

但是，假设您的使用案例要求您（或您的应用程序）从集群的 VPC 外部访问您的 Amazon DocumentDB 资源。在这种情况下，您可以使用 SSH 隧道（也称为端口转发）以访问您的 Amazon DocumentDB 资源。

深入讨论 SSH 隧道超出了本主题的范围。有关 SSH 隧道的更多信息，请参阅以下内容：

- [SSH 隧道](#)
- [SSH 端口转发示例](#)，尤其是[本地转发](#)部分

要创建 SSH 隧道，您需要一个与 Amazon DocumentDB 集群在同一 Amazon VPC 中运行的 Amazon EC2 实例。您可以使用同一 VPC 中的现有 EC2 实例作为集群，或创建一个集群。有关更多信息，请参阅适合您的操作系统的主题：

- [Amazon EC2 Linux 实例入门](#)
- [Amazon EC2 Windows 实例入门](#)

您可能通常会使用以下命令连接到 EC2 实例：

```
ssh -i "ec2Access.pem" ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com
```

如果是这样，您可以设置到 Amazon DocumentDB 集群的 SSH 隧道。sample-cluster.node.us-east-1.docdb.amazonaws.com 在本地计算机上运行以下命令。-L 标志用于转发本地端口。使用 SSH

隧道时，我们建议您使用集群终端节点连接到集群，而不要尝试以副本集模式（即在连接字符串中指定 `replicaSet=rs0`）进行连接，因为这会导致错误。

```
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
```

创建 SSH 隧道后，您发布到的任何命令 `localhost:27017` 被转发到 Amazon DocumentDB 集群 `sample-cluster` 在 Amazon VPC 中运行。如果在 Amazon DocumentDB 群集上启用了传输层安全性 (TLS)，则需要从下载 Amazon DocumentDB 的公钥。以下操作将下载此文件：

```
wget https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem
```

Note

默认情况下会对新的 Amazon DocumentDB 集群启用 TLS。但是，您可以将其禁用。有关更多信息，请参阅 [管理 Amazon DocumentDB 集群 TLS 设置 \(p. 137\)](#)。

要从 Amazon VPC 外部连接到您的 Amazon DocumentDB 集群，请使用以下命令。

```
mongo --sslAllowInvalidHostnames --ssl --sslCAFile rds-combined-ca-cn-bundle.pem --username <yourUsername> --password <yourPassword>
```

从 Robo 3T 连接到 Amazon DocumentDB 集群

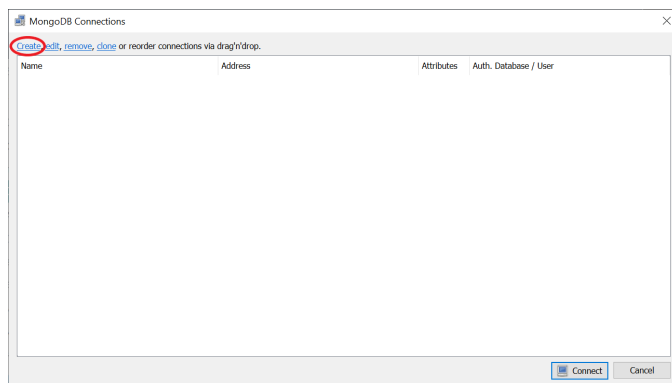
Robo 3T 是一款轻量级、开源、以外壳为中心的跨平台图形用户界面工具，用于管理 MongoDB 工作负载。Robo 3T 使您能够创建数据库、集合、添加用户、文档、通过自动完成执行一次性查询以及可视化 GUI 界面的结果。

先决条件

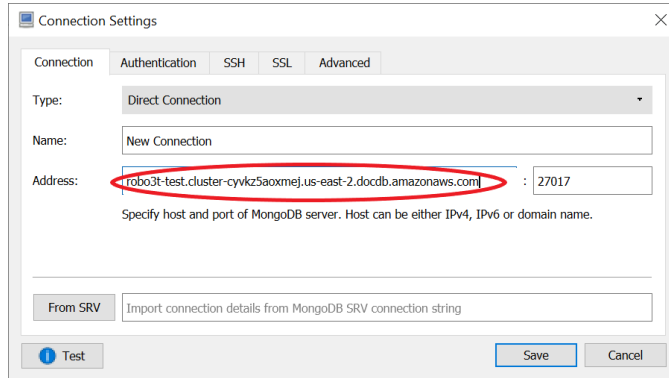
- 如果 Amazon EC2 上尚未运行 Amazon DocumentDB 群集，请按照以下操作的说明操作：[与 Amazon EC2 进行 Connect](#)。
- 如果你没有机器人 3T，[下载并安装](#)。

Connect Robo 3T 联系

1. 打开机器人 3T 然后选择 Create。



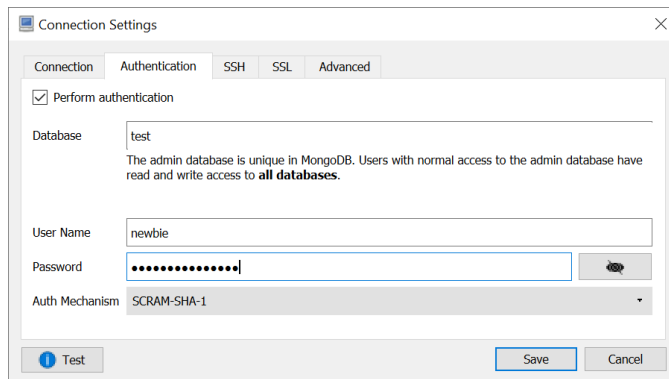
2. 在存储库的 Connection 选项卡，在地址字段中，输入群集终端节点信息。



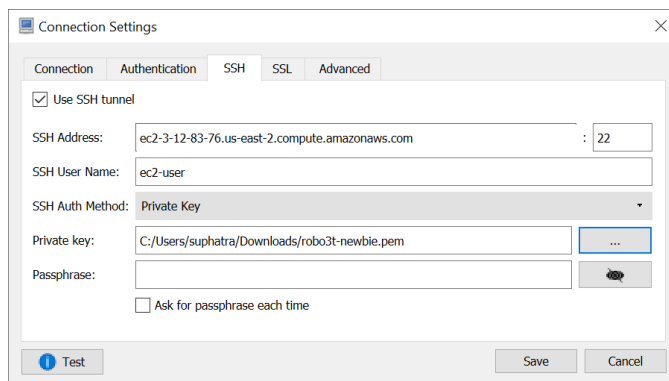
Note

找不到集群终端节点？只需按照步骤操作。[这里](#)。

3. 在存储库的身份验证选项卡，选中对于对身份验证。现在为您的群集输入身份验证信息。确保使用自定义数据库名称，如test。使用admin（默认设置）不适用于没有数据库的群集的 Amazon DocumentDB。创建第一个数据库后，您可以修改连接以使用admin。



4. 在存储库的SSH选项卡，选中对于使用 SSH 隧道，然后添加 EC2 实例的 SSH 地址、用户名和私钥/密码。SSH 地址是您的 EC2 实例的公有 DNS。



5. InSSH 身份验证方法中，选择身份验证方法之一。

- If you chose 私有密钥, then select the “...” button to open up the file finder and select the .pem file for your EC2 instance.
- If you chose Password, you must enter the SSH address, username and private key for your Amazon EC2 instance. You can find this on the Amazon EC2 console.

Note

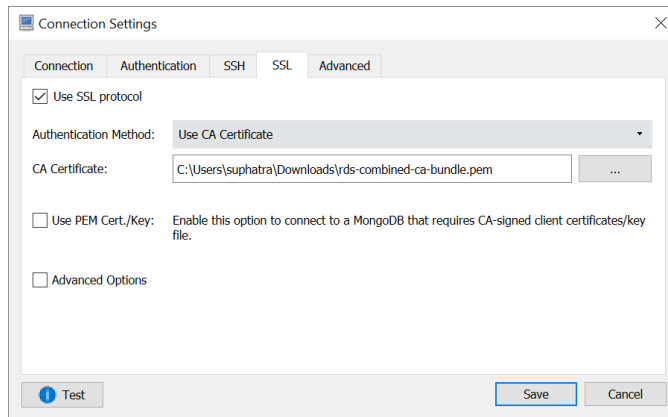
提示: 如果您使用 Linux/macOS 客户端计算机, 则可能需要使用以下命令更改私钥的权限:

```
> chmod 400 /fullPathToYourPemFile/<yourKey>.pem
```

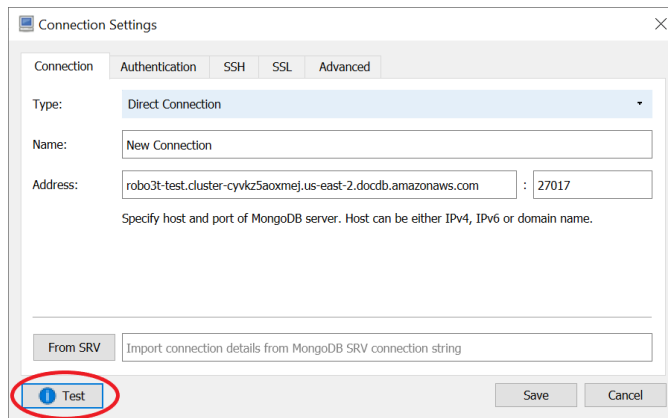
6. 现在选择SSL选项卡, 然后单击下拉菜单身份验证法. 选择使用 CA 证书. Select高级选项而且对于无效的主机名菜单中, 选择已允许.

Note

如果你使用 macOS Catalina 或更高版本, 请选择自签名证书作为身份验证法因为 macOS 不接受有效期超过 825 天的证书。



7. 通过选择来测试连接。测试按钮。



8. 一个诊断应显示带有测试结果的窗口。如果一切都是绿色的, 那么关闭盒子。



9. 现在选择Save.
10. 现在选择你的集群然后选择Connect (连接).

从 Studio 3T 连接到 Amazon DocumentDB 集群

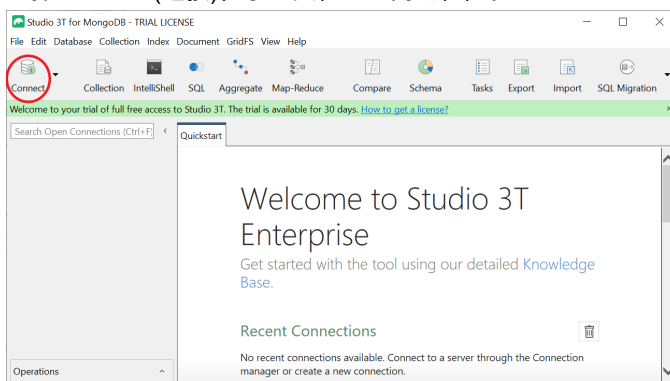
3T Studio 对于使用 MongoDB 的开发人员和数据工程师而言，是一款受欢迎的 GUI 和 IDE。它提供了几种强大的数据功能树、表格和 JSON 视图、CSV、JSON、SQL 和 bson/monGoDump 中的导入/导出、灵活的查询选项、可视化拖放 UI、带自动完成功能的内置 mongo shell、聚合管道编辑器和 SQL 查询支持。

先决条件

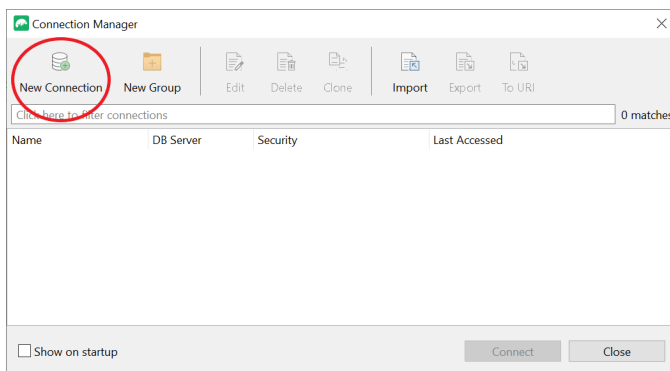
- 如果您尚未在 Amazon EC2 上运行 Amazon DocumentDB 集群，请按照有关如何执行以下操作的说明操作：[Connect Amazon EC2](#)。
- 如果您没有 Studio 3T，[下载并安装它](#)。

Connect 工作室 3T

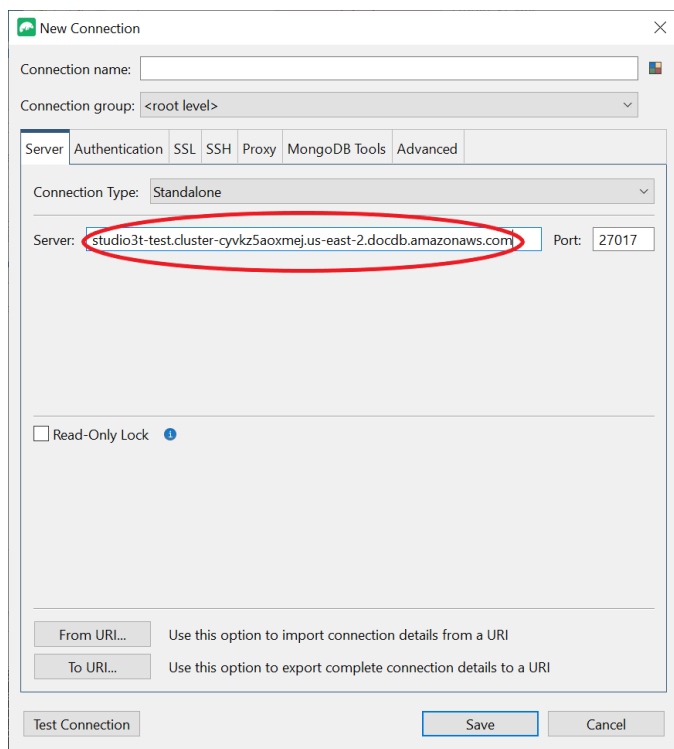
1. 选择 Connect (连接) 位于工具栏左上角的位置。



2. 选择新连接位于工具栏左上角的位置。



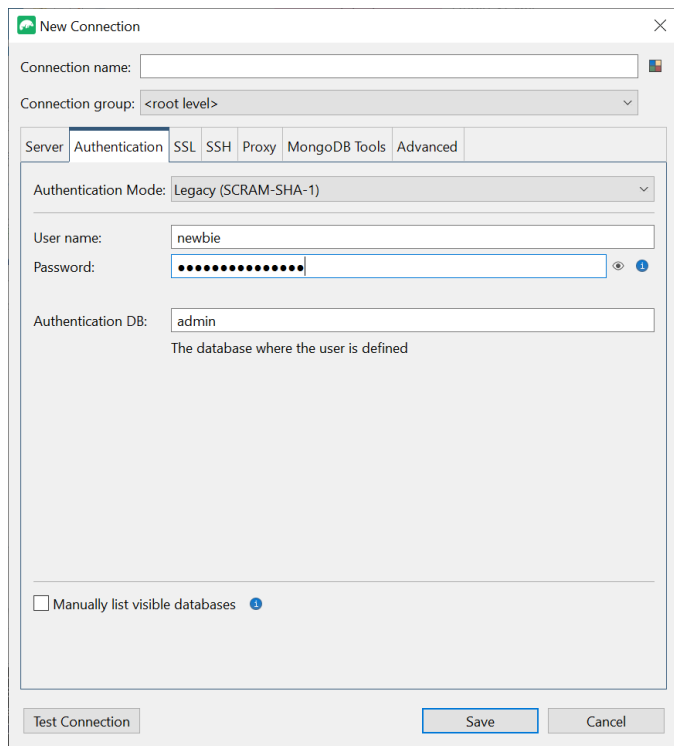
3. 在存储库的服务器选项卡，在服务器字段中，输入群集终端节点信息。



Note

找不到集群终端节点？只需按照步骤操作[这里](#)。

4. 选择身份验证选项卡然后选择传统在下拉菜单中验证模式。



5. 在用户名和密码字段之间没有不同。

6. 选择SSL选项卡然后选中复选框使用 SSL 协议进行连接。

New Connection

Connection name:

Connection group: <root level>

Server | Authentication | **SSL** | SSH | Proxy | MongoDB Tools | Advanced

☒ Use SSL protocol to connect

☒ Use own Root CA file (--sslCAFile)

☐ Accept server SSL certificates trusted by the operating system

☐ Accept any server SSL certificates

☐ Use Client Certificate (--sslPEMKeyFile)

Client Certificate:

Passphrase:

☒ My client certificate is not protected by a passphrase

☒ Allow invalid hostnames (--sslAllowInvalidHostnames)

☐ Use Server Name Indication (Advanced)

SNI Host Name:

Test Connection Save Cancel

7. 选择使用自己的根 CA 文件。然后添加 Amazon DocumentDB 证书 (如果在 DocumentDB 集群上禁用 SSL , 则可以跳过此步骤) 。选中该复选框以允许无效的主机名。

New Connection

Connection name:

Connection group: <root level>

Server | Authentication | **SSL** | SSH | Proxy | MongoDB Tools | Advanced

☒ Use SSL protocol to connect

☒ Use own Root CA file (--sslCAFile)

☐ Accept server SSL certificates trusted by the operating system

☐ Accept any server SSL certificates

☐ Use Client Certificate (--sslPEMKeyFile)

Client Certificate:

Passphrase:

☒ My client certificate is not protected by a passphrase

☒ Allow invalid hostnames (--sslAllowInvalidHostnames)

☐ Use Server Name Indication (Advanced)

SNI Host Name:

Test Connection Save Cancel

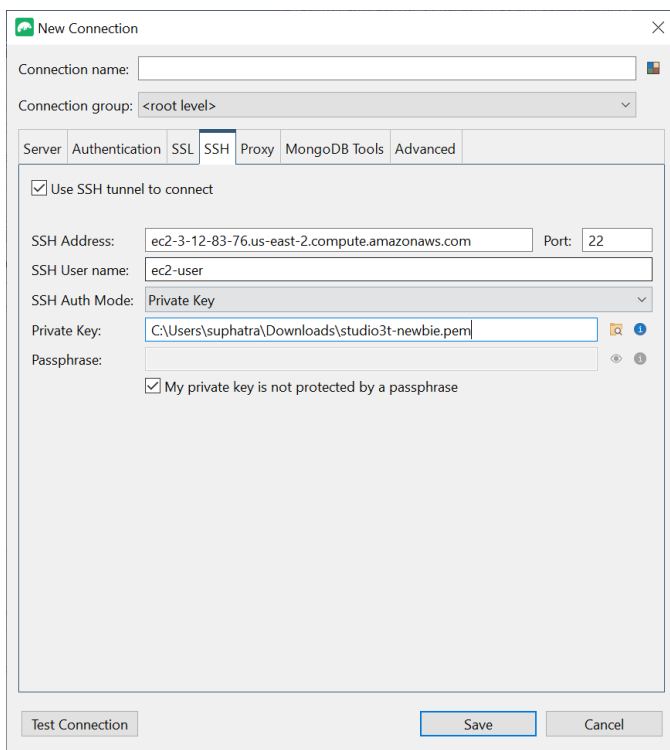
Note

没有证书？您可以使用以下命令下载它：

```
wget https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem
```

8. 如果您从 Amazon VPC 外部的客户端计算机进行连接，则需要创建 SSH 隧道。您将在 SSH 选项卡。
 - a. 选中该复选框使用 SSH 隧道然后在 SSH 地址字段中返回的子位置类型。这是你的实例公有 DNS (IPv4)。您可以从您的位置获取此 URL [Amazon EC2 管理控制台](#)。
 - b. 输入您的用户名。这是 Amazon EC2 实例的用户名
 - c. 适用于 SSH 身份验证模式，请选择私有密钥。在私有密钥字段中，选择文件查找器图标以查找并选择 Amazon EC2 实例的私钥。这是您在 Amazon EC2 控制台中创建实例时保存的 .pem 文件 (key pair)。
 - d. 如果您使用 Linux/macOS 客户端计算机，则可能需要使用以下命令更改私钥的权限：

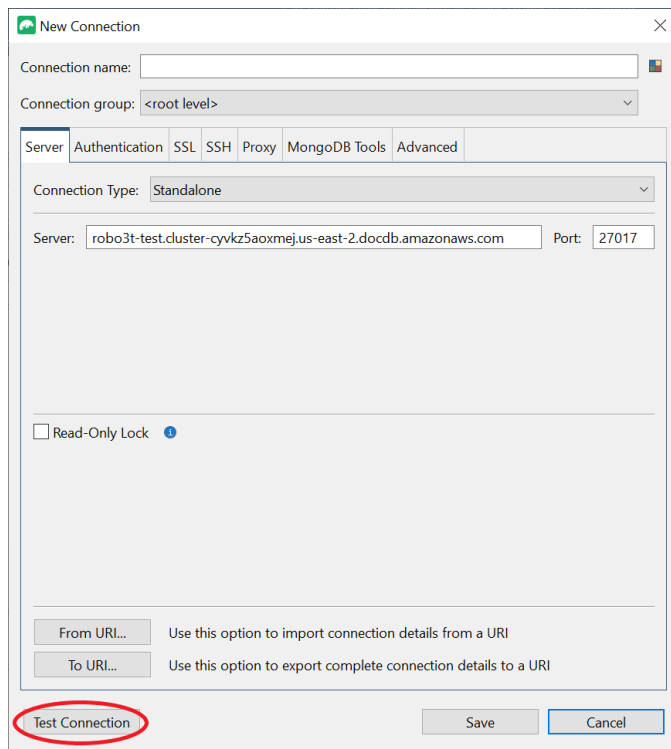
```
chmod 400 /fullPathToYourPemFile/<yourKey>.pem
```



Note

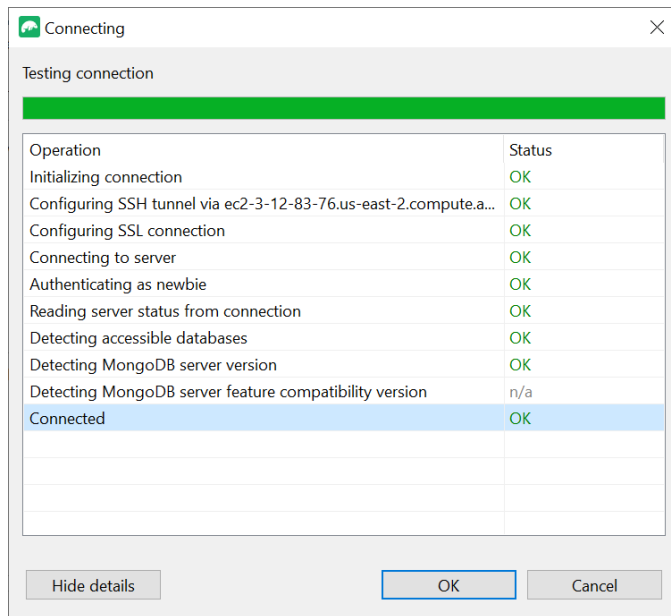
此 Amazon EC2 实例应与您的 DocumentDB 集群位于同一亚马逊 VPC 和安全组中。你可以从你的那里获取 SSH 地址、用户名和私钥 [Amazon EC2 管理控制台](#)。

9. 现在，通过选择测试连接按钮。



The 'New Connection' dialog box is shown with the 'Server' tab selected. The 'Connection name' field is empty. The 'Connection group' is set to '<root level>'. The 'Connection Type' is 'Standalone'. The 'Server' field contains 'robo3t-test.cluster-cyvkz5aoxmej.us-east-2.docdb.amazonaws.com' and the 'Port' is '27017'. There is a 'Read-Only Lock' checkbox which is unchecked. At the bottom, there are buttons for 'From URI...', 'To URI...', 'Test Connection' (highlighted with a red circle), 'Save', and 'Cancel'.

10. 诊断窗口应加载一个绿色条以表示测试成功。现在选择确定关闭诊断窗口。

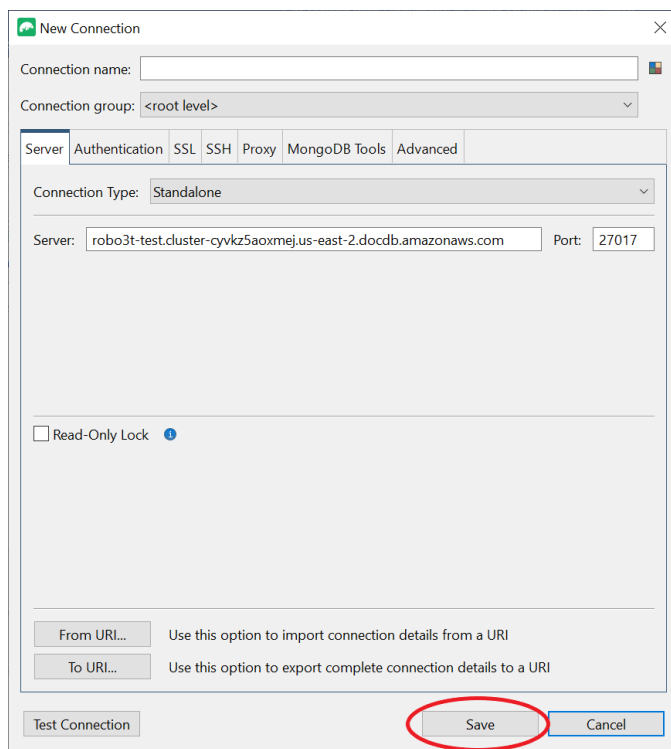


The 'Connecting' dialog box shows the 'Testing connection' status. A green progress bar is at the top. Below it is a table with the following data:

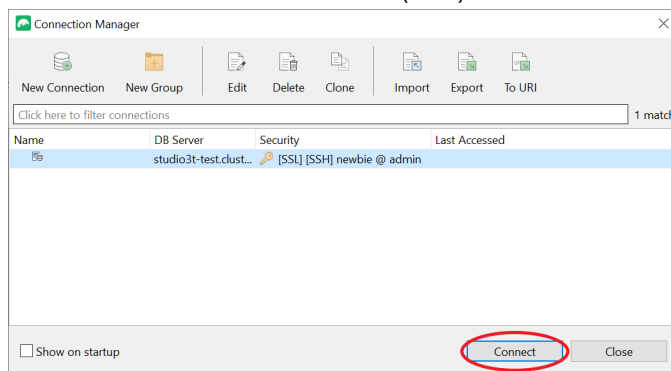
Operation	Status
Initializing connection	OK
Configuring SSH tunnel via ec2-3-12-83-76.us-east-2.compute.a...	OK
Configuring SSL connection	OK
Connecting to server	OK
Authenticating as newbie	OK
Reading server status from connection	OK
Detecting accessible databases	OK
Detecting MongoDB server version	OK
Detecting MongoDB server feature compatibility version	n/a
Connected	OK

At the bottom, there are buttons for 'Hide details', 'OK', and 'Cancel'.

11. 选择Save以保存您的连接以供将来使用。



12. 现在选择你的集群然后选择Connect (连接).



恭喜您！现在，您已通过 Studio 3T 成功连接到 Amazon DocumentDB 集群。

使用Amazon EC2 Connect

本节介绍如何使用 Amazon EC2 启动 Amazon DocumentDB (兼容 MongoDB) 集群并使用 mongo shell 与之交互。以下视频演示了本指南中的步骤。

先决条件

在创建第一个 Amazon DocumentDB 集群之前，您必须先执行以下操作：

创建 Amazon Web Services (Amazon) 账户

在开始使用 Amazon DocumentDB 之前，您必须拥有 Amazon Web Services (Amazon) 账户。Amazon 账户是免费的。您只需为使用的服务和资源付费。

如果您还没有 Amazon Web Services 账户，请完成以下步骤来创建一个。

注册 Amazon Web Services 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

设置所需的 Amazon Identity and Access Management (IAM) 权限。

访问管理 Amazon DocumentDB 资源（例如集群、实例和集群参数组）需要具备以下证书 Amazon 可以用来验证您的请求。有关更多信息，请参阅 [Amazon DocumentDB 中的 Identity and Access Management](#) (p. 142)。

1. 在搜索栏中 Amazon Web Services Management Console，键入 IAM 并选择 IAM 在出现的下拉菜单中。
2. 进入 IAM 控制台后，选择 Users 从导航窗格中。
3. 选择您的用户名。
4. 单击按钮添加权限。
5. 选择 Attach existing policies directly (直接附加现有策略)。
6. 类型 AmazonDocDBFullAccess 在搜索栏中，并在它出现在搜索结果中后将其选中。
7. 点击底部的蓝色按钮，上面写着下一步：审核。
8. 点击底部的蓝色按钮，上面写着添加权限。

创建 Amazon Virtual Private Cloud

取决于哪个 Amazon Web Services 区域您在，您可能已经创建了默认 VPC，也可能没有。如果您没有默认 VPC，则完成中的步骤 1 [Amazon VPC 入门](#) 在 Amazon VPC User Guide. 这将花费不到五分钟。

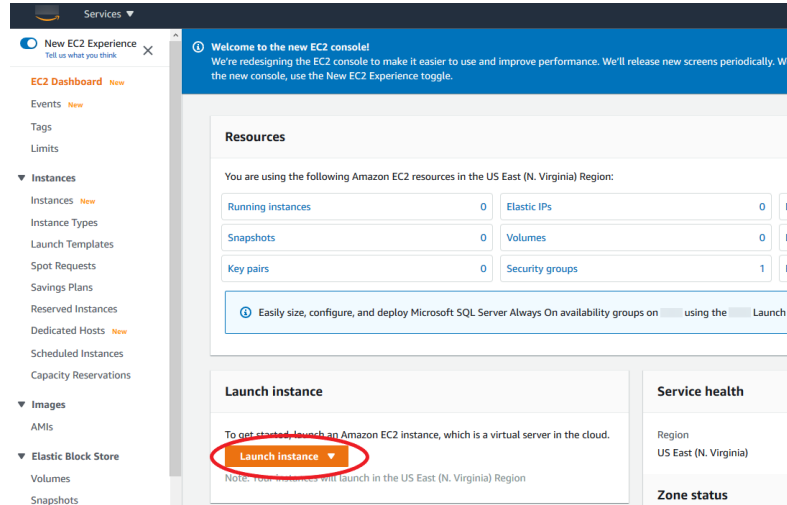
第 1 步：创建 Amazon EC2 实例

在此步骤中，您将在同一区域和亚马逊 VPC 中创建一个 Amazon EC2 实例，稍后将使用该实例来配置您的 Amazon DocumentDB 集群。

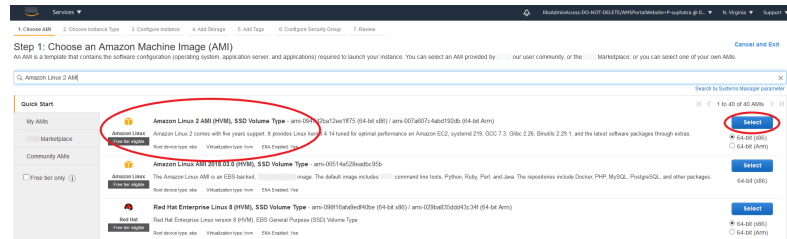
1. 在 Amazon EC2 控制台上，选择启动实例。

Amazon DocumentDB 开发人员指南

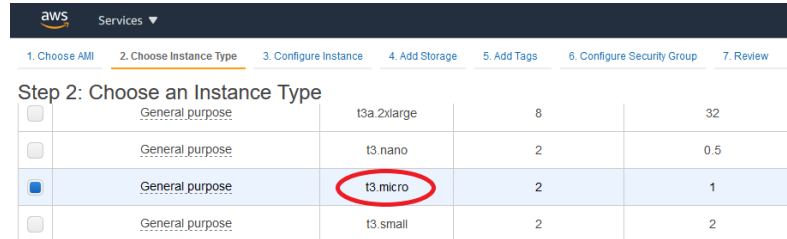
第 1 步：创建 Amazon EC2 实例



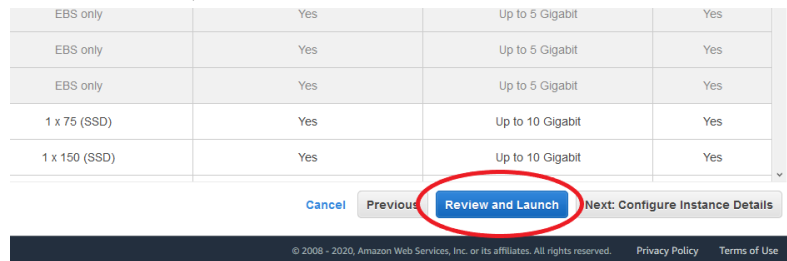
2. 定位Amazon Linux AMI然后选择Select.



3. 选择t3.micro实例的类型。



4. 选择核查和启动，这将允许你跳到控制台的步骤 7：核查实例启动页面。



5. 在“安全组”下，选择编辑安全组。

Amazon DocumentDB 开发人员指南

第 1 步：创建 Amazon EC2 实例

Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.

Warning: Your instance configuration is not eligible for the free usage tier. To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about this usage tier eligibility and usage restrictions.

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0847c2b12ee1ff75

Instance Type

Instance Type	ECU	VCPU	Memory (GB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t3.micro	Variable	2	1	EBS only	Yes	Up to 5 Gbps

Security Groups

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2020-09-28T15:16:42.686-07:00

Type	Protocol	Port Range	Source	Description

This security group has no rules.

Edit security groups

6. 这将默认为创建新的安全组。在安全组名称字段中，writedemoEC2。

Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:

☒ Create a new security group

☐ Select an existing security group

Security group name: demoEC2

Description: launch-wizard-1 created 2020-09-28T15:16:42.686-07:00

Type	Protocol	Port Range
SSH	TCP	22

Add Rule

7. 将类型更改为 SSH。这会自动将端口范围设置为 22。

Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:

☒ Create a new security group

☐ Select an existing security group

Security group name: demoEC2

Description: launch-wizard-1 created 2020-09-28T15:16:42.686-07:00

Type	Protocol	Port Range
SSH	TCP	22

Add Rule

8. 选择下拉菜单源然后选择我的 IP。

Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to it HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:

☒ Create a new security group

☐ Select an existing security group

Security group name: demoEC2

Description: launch-wizard-1 created 2020-09-28T15:16:42.686-07:00

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP

Add Rule

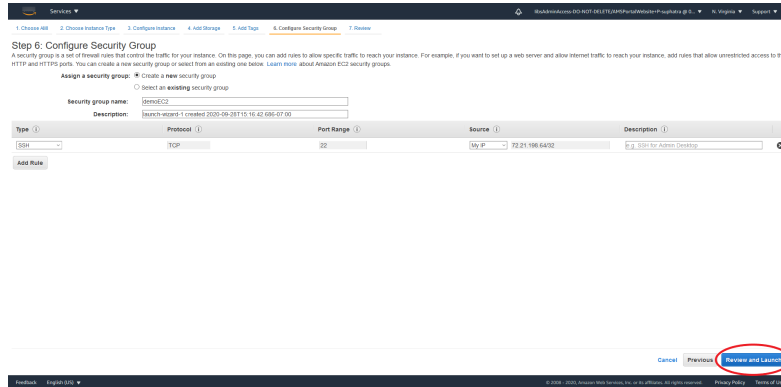
Note

您只能从当前的 IP 地址访问 demoEC2 安全组。如果您的 IP 地址发生变化，则必须更新安全组。

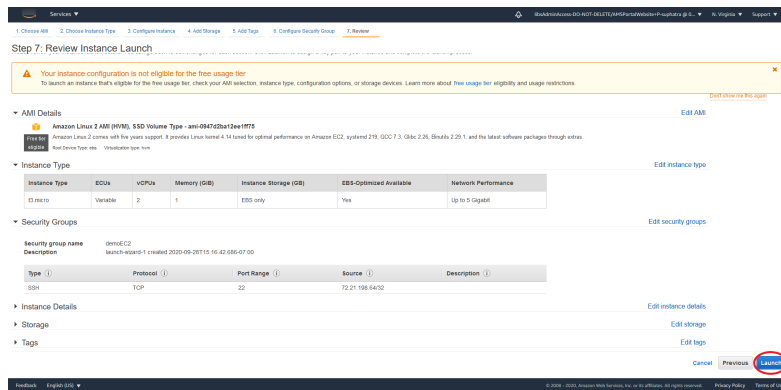
9. NOW 选择核查和启动。现在，您应该看到在 demoEC2 安全组内容安全组部分。

Amazon DocumentDB 开发人员指南

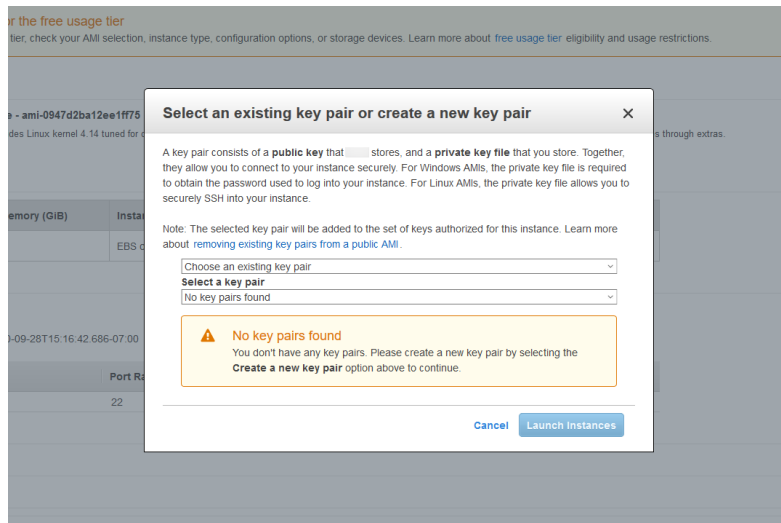
第 1 步：创建 Amazon EC2 实例



10. 验证信息并选择启动。



11. 将弹出一个窗口，标题为选择现有key pair 或创建新的key pair. 它将如下所示：



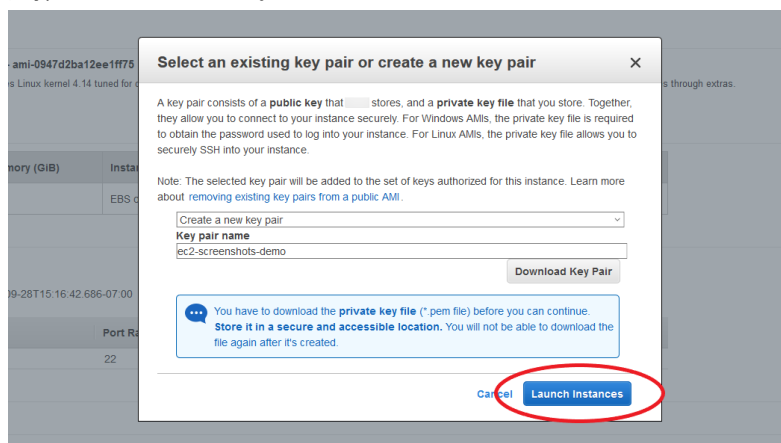
您必须提供 Amazon EC2 key pair。如果您有 Amazon EC2 key pair：

- 选择一个key pair，从列表中选择您的key pair。
- 必须已经有可用私有密钥文件（.pem 文件）才能登录 Amazon EC2 实例。

如果您没有 Amazon EC2 key pair：

- 选择 Create a new key pair。

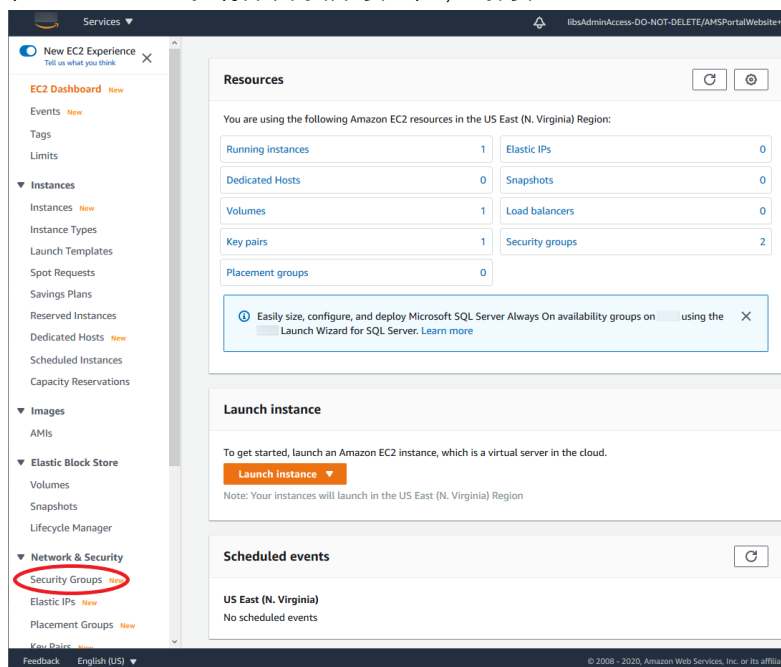
- b. 在字段中为键对写一个名称密钥对名称。
 - c. 下载私有密钥文件（.pem 文件）。稍后登录 Amazon EC2 实例时，将需要此文件。
12. 选择 Launch Instances。



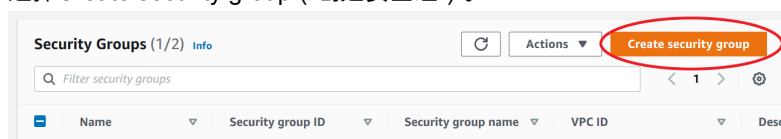
第 2 步：创建安全组

现在，您将在默认 Amazon VPC 中创建一个新的安全组。安全组 demoDocDB 允许您通过端口 27017（Amazon DocumentDB B 的默认端口）从 Amazon EC2 实例连接到您的 Amazon DocumentDB 集群。

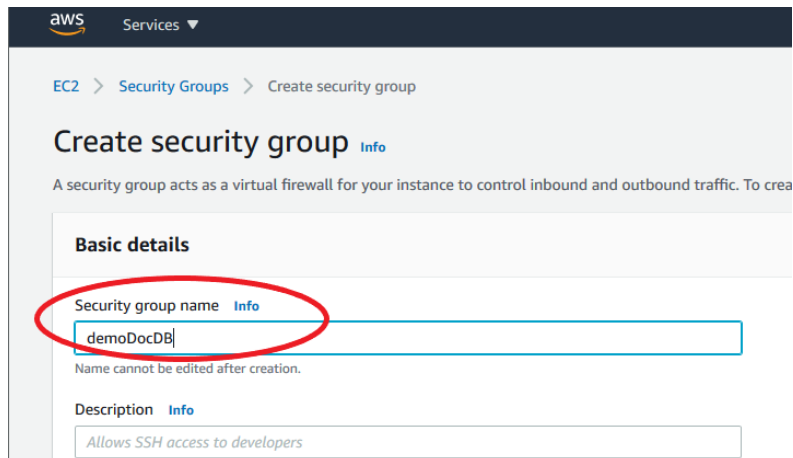
1. 在 Amazon EC2 控制台下网络和安全，选择安全组。



2. 选择 Create security group（创建安全组）。

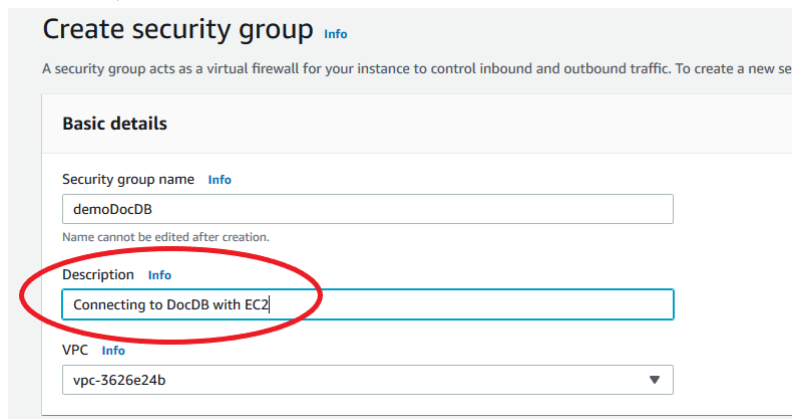


3. 对于安全组名称，输入 demoDocDB。



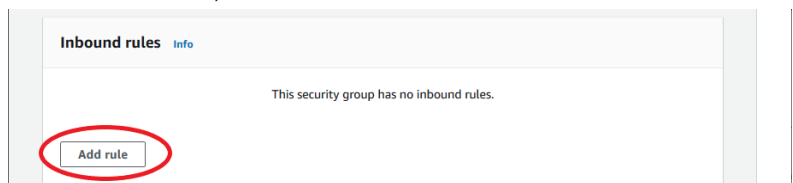
The screenshot shows the 'Create security group' page in the AWS Management Console. The 'Basic details' section is visible. The 'Security group name' field is highlighted with a red circle and contains the text 'demoDocDB'. Below it, a note states 'Name cannot be edited after creation.' The 'Description' field is also visible, containing the text 'Allows SSH access to developers'.

4. 对于说明，输入说明。



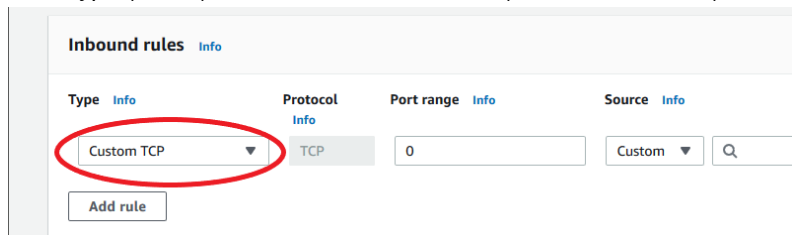
The screenshot shows the 'Create security group' page in the AWS Management Console. The 'Basic details' section is visible. The 'Description' field is highlighted with a red circle and contains the text 'Connecting to DocDB with EC2'. The 'VPC' dropdown menu is also visible, showing 'vpc-3626e24b'.

5. 对于 VPC，接受使用您的默认 VPC。
6. 在入站规则部分中，选择添加规则。



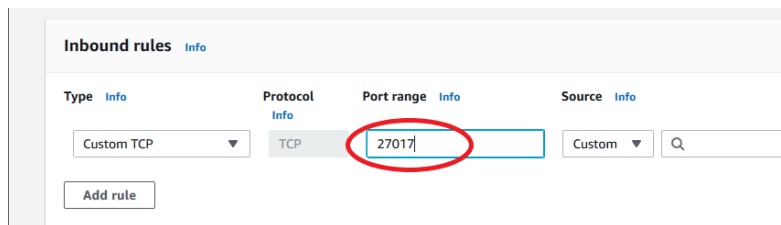
The screenshot shows the 'Inbound rules' section of the 'Create security group' page. The text 'This security group has no inbound rules.' is displayed. The 'Add rule' button is highlighted with a red circle.

7. 对于 Type (类型)，选择 Custom TCP Rule (自定义 TCP 规则)。

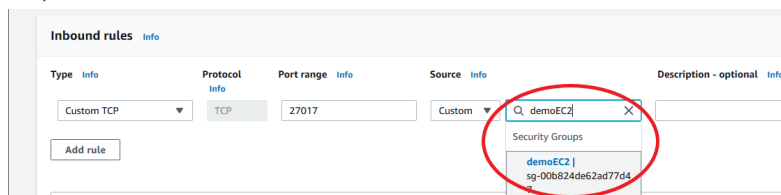


The screenshot shows the 'Inbound rules' section of the 'Create security group' page. The 'Type' dropdown menu is highlighted with a red circle and shows 'Custom TCP'. The 'Protocol' dropdown menu is also visible, showing 'TCP'. The 'Port range' field is empty, and the 'Source' dropdown menu is set to 'Custom'.

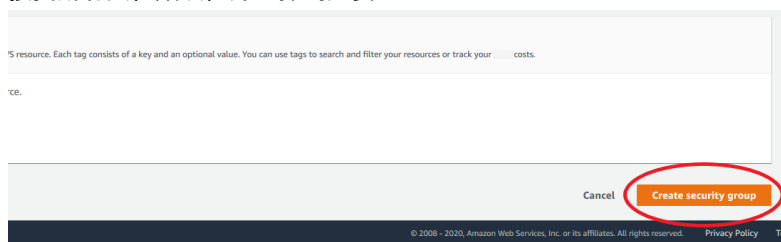
8. 对于端口范围，输入 27017。



9. 对于目标，选择Custom. 在旁边的字段中，搜索你刚才创建的安全组demoEC2. 您可能需要刷新浏览器，让 Amazon EC2 控制台自动填充demoEC2来源名称。



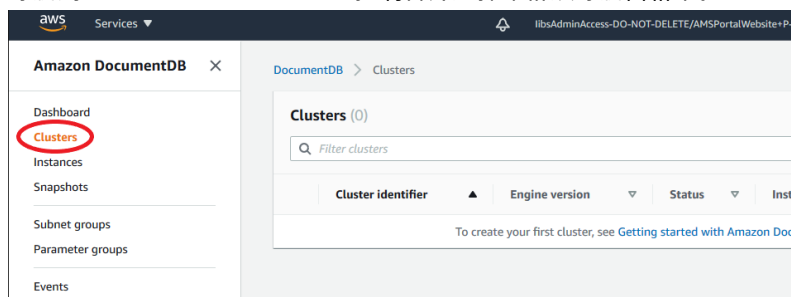
10. 接受所有其他默认值并选择创建安全组。



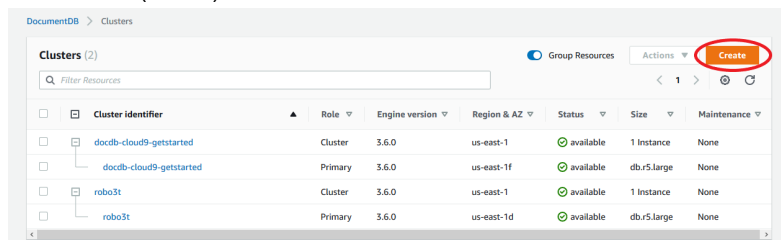
第 3 步：创建 Amazon DocumentDB 集群

在配置 Amazon EC2 实例的同时，您将创建您的 Amazon DocumentDB 集群。

1. 导航到Amazon DocumentDB B 控制台并选择集群从导航窗格中。



2. 选择Create (创建) 。



3. 对于实例数，选择1. 这将最大限度地降低成本。保留其他设置的默认值。

Create Amazon DocumentDB cluster

Configuration

Cluster identifier [Info](#)
Specify a unique cluster identifier.

getstarted-EC2

Engine version
4.0.0

Instance class [Info](#)
db.r5.large
2 vCPUs 16GiB RAM

Number of instances [Info](#)
1

4. 对于身份验证，输入用户名和密码。重要提示：在后面的步骤中，您将需要此用户名和密码来对集群进行身份验证。

Authentication

Master username [Info](#)
Specify an alphanumeric string that defines the login ID for the master user.

docdbnoob

Master username must start with a letter and contain 1 to 63 characters

Master password [Info](#) Confirm master password [Info](#)

Master password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

5. 开显示高级设置.

Amazon DocumentDB requires permissions to manage resources on your behalf. By clicking Create cluster, you grant permission for Amazon DocumentDB to create a service-linked role in IAM that contains the required permissions.

☒ Show advanced settings

Cancel Create cluster

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy

6. 在网络设置部分，用于Amazon VPC 安全组，选择demoDocDB.

Network settings

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this cluster.

vpc-3626e24b

Only VPCs with a corresponding subnet group are listed. Once a cluster is created, the VPC cannot be changed.

Subnet group [Info](#)

demoEC2 (VPC)

demoDocDB (VPC)

default (VPC)

Select VPC security groups

default (VPC) X

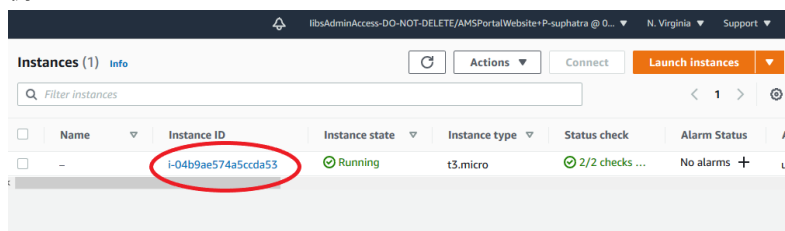
7. 选择创建集群。



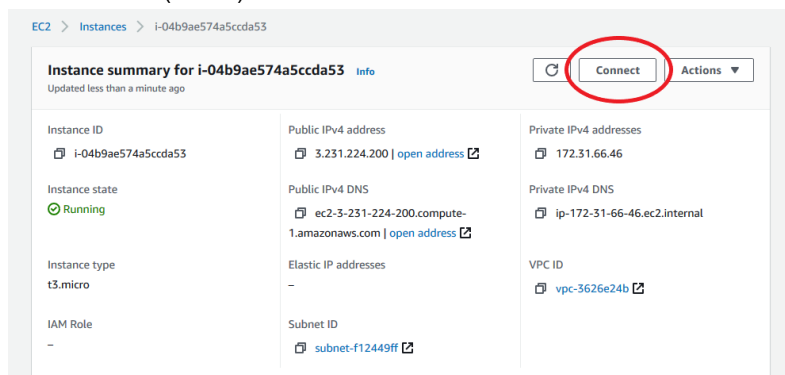
第 4 步：Connect 您的 Amazon EC2 实例

要安装 mongo shell，您必须先连接到您的 Amazon EC2 实例。安装 mongo shell 使您能够连接和查询您的 Amazon DocumentDB 集群。完成以下步骤：

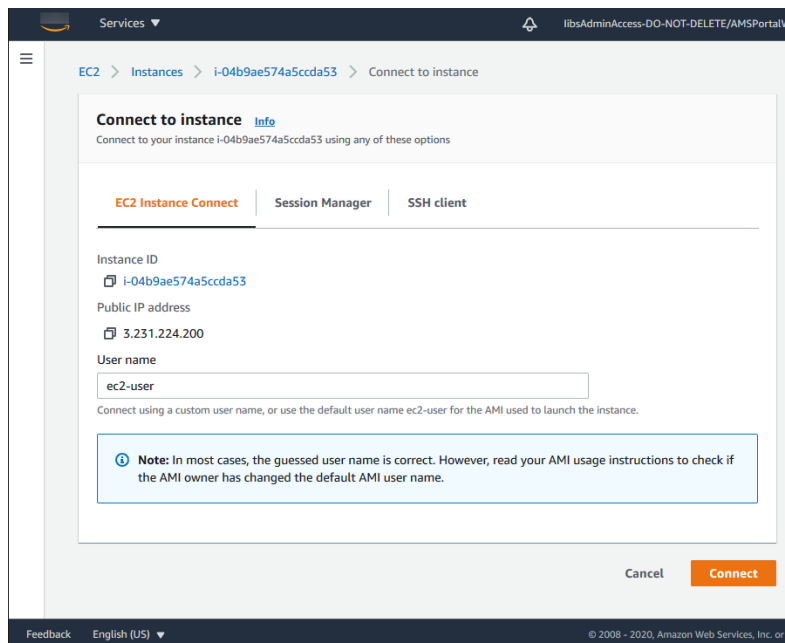
1. 在 Amazon EC2 控制台上，导航到您的实例，看看您刚刚创建的实例是否正在运行。如果是，选择实例。



2. 选择 Connect (连接)。



3. 您将看到三个连接方法选项：Amazon EC2 实例 Connect、会话管理器或 SSH 客户端。您必须选择一个并按照其说明进行操作。



Note

如果您的 IP 地址在您开始本演练后发生了变化，或者您稍后要返回环境，则必须更新您的 demoEC2 安全组入站规则，用于启用来自新 API 地址的入站流量。

第 5 步：安装 mongo shell

您现在可以安装 mongo shell，这是一个命令行实用程序，用于连接和查询 Amazon DocumentDB 集群。为操作系统安装 mongo shell，请按照以下说明安装 mongo shell。

在 Amazon Linux 上

在 Amazon Linux 上安装 mongo shell

1. 创建存储库文件。在 EC2 实例的命令行中，执行以下命令：

```
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. 完成后，通过执行以下命令安装 mongo shell：

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

在 Ubuntu 18.04 上

在 Ubuntu 18.04 上安装 mongo shell

1. 导入包管理系统将使用的公有密钥。

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. 使用适合您的 Ubuntu 版本的命令创建用于 MongoDB 的列表文件 `/etc/apt/sources.list.d/mongodb-org-3.6.list`。

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

以上命令将同时为 Bionic 和 Xenial 安装 mongo 3.6 shell。

3. 使用以下命令重新加载本地程序包数据库：

```
sudo apt-get update
```

4. 安装 MongoDB shell。

```
sudo apt-get install -y mongodb-org-shell
```

有关在您的 Ubuntu 系统上安装早期版本的 MongoDB 的信息，请参阅[在 Ubuntu 中安装 MongoDB Community Edition](#)。

在其他操作系统上

要在其他操作系统上安装 mongo shell，请参阅 MongoDB 文档中的[安装 MongoDB Community Edition](#)。

第 6 步：管理 Amazon DocumentDB T

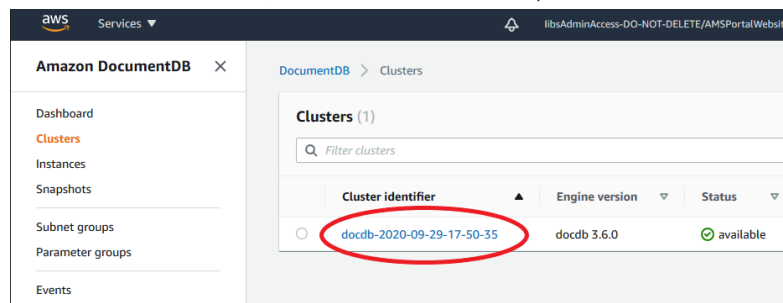
使用以下代码下载 Amazon DocumentDB B 的 CA 证书：`wget https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem`

Note

对于任何新的 Amazon DocumentDB 集群，传输层安全性 (TLS) 均处于启用状态。有关更多信息，请参阅[管理 Amazon DocumentDB B 集群 TLS 设置](#)。

步骤 7：Connect 您的 Amazon DocumentDB 集群

1. 在 Amazon Document 数据库控制台的集群下方，找到您的集群。选择您创建的集群。



2. 导航到Connection盒子。它会像这样。



复制提供的连接字符串。



将其粘贴到终端并对其进行以下更改后：首先，确保字符串中的用户名正确。然后，省略<insertYourPassword>这样当你连接时，mongo shell 会提示你输入密码。您的连接字符串应类似于以下内容：

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile rds-combined-ca-cn-
bundle.pem --username demoUser --password
```

3. 现在在终端中按 Enter 键。现在将提示您输入密码。输入您的密码。
4. 当您输入密码时，可以看到rs0:PRIMARY>提示，您已成功连接到您的Amazon DocumentDB B 集群。

连接有问题吗？请参阅[对 Amazon DocumentDB](#)。

第 8 步：插入和查询数据

现在您已连接到集群，可以运行一些查询来熟悉如何使用文档数据库。

1. 要插入单个文档，输入以下内容：

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

2. 您将得到以下输出：

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

3. 您可以使用以下方式阅读你写的文档findOne()命令（因为它只返回一个文档）。输入以下内容：

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

4. 您将得到以下输出：

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

5. 要再执行一些查询，请考虑游戏配置文件用例。首先，将几个条目插入一个名为profiles。输入以下内容：

```
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. 您将得到以下输出：

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

7. 使用find()命令返回配置文件集中的所有文档。输入以下内容：

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

8. 您将获得与您步骤 5 中键入的数据相匹配的输出。

9. 使用筛选器对单个文档进行查询。输入以下内容：

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

10. 您应该得到此输出内容：

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

11. 现在，让我们尝试查找配置文件并使用以下命令对其进行修改findAndModify命令。我们将使用以下代码给用户 Matt 额外十分：

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

12. 你会得到以下输出（注意他的分数还没有增加）：

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

13. 您可以使用以下查询来验证他的分数是否已更改：

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

14. 您将得到以下输出：


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

第 9 步：探索

恭喜您！您已成功完成 Amazon DocumentDB B 快速入门指南。

下一步是什么？了解如何充分利用这个强大的数据库及其一些常用功能：

- [管理 Amazon DocumentDB](#)
- [扩展](#)
- [备份和还原](#)

Note

为了节省成本，您可以停止您的 Amazon DocumentDB 集群以降低成本，也可以删除该集群。默认情况下，在闲置 30 分钟后，您的 Amazon Cloud9 环境将停止底层 Amazon EC2 实例。

使用 Amazon DocumentDB JDBC 驱动程序进行 Connect

适用于 Amazon DocumentDB 的 JDBC 驱动程序为开发人员提供了 SQL 关系界面，并支持通过 Tableau 和 DBVisualizer 等 BI 工具进行连接。

有关更多详细信息，请参阅 [GitHub 上的 Amazon DocumentDB JDBC 驱动程序文档](#)。

主题

- [入门 \(p. 480\)](#)
- [从 Tableau Desktop Connect 到 Amazon DocumentDB \(p. 481\)](#)
- [从 DBVisualizer Connect 到 Amazon DocumentDB \(p. 487\)](#)
- [自动生成模式 \(p. 489\)](#)
- [SQL Support 和限制 \(p. 495\)](#)
- [故障排除 \(p. 495\)](#)

入门

第 1 步 创建 Amazon DocumentDB 集群

如果您没有创建 Amazon DocumentDB 集群，请使用 [开始使用](#) 《Amazon DocumentDB 开发人员指南》中的部分。

Note

DocumentDB 是唯一的 Virtual Private Cloud (VPC) 服务。如果您是从集群的 VPC 之外的本地计算机进行连接，则需要创建到 Amazon EC2 实例的 SSH 连接。在这种情况下，请使用中的说明启动集群 [Connect EC2](#)。请参阅 [使用 SSH 隧道 Connect 到 Amazon DocumentDB](#) 了解有关 SSH 隧道的更多信息以及何时可能需要它。

第 2 步 JRE 或 JDK 安装

根据您的 BI 应用程序，您可能需要确保计算机上安装 64 位 JRE 或 JDK 安装版本 8 或更高版本。您可以下载 Java SE 运行时环境 [8 这里](#)。

第 3 步 下载 DocumentDB JDBC 驱动程序

从下载 DocumentDB JDBC 驱动程序[这里](#)。驱动程序被打包为单个 JAR 文件（例如 ocumentdb-jdbc-1.0.0-all.jar）。

第 4 步 使用 SSH 隧道 Connect 到 Amazon DocumentDB

在 Amazon DocumentDB irtual Private Cloud (Amazon MongoDB) 中部署在 Amazon Virtual Private Cloud (Amazon VPC) 中。它们可以由 Amazon EC2 实例或其他实例直接访问Amazon部署在同一 Amazon VPC 中的服务。此外，可以通过 EC2A 实例或其他实例访问 Amazon DocumentDBAmazon同一 VPC 中的服务Amazon通过 VPC 对等互连进行区域或其他区域。

您可以使用 SSH 隧道（也称为“端口转发”）从集群 VPC 以外访问您的 Amazon DocumentDB 资源。对于大多数没有在与 DocumentDB 集群所在的 VPC 中的虚拟机上运行应用程序的用户来说，这将是这种情况。

要创建 SSH 隧道，您需要一个与 Amazon DocumentDB 集群在同一 Amazon VPC 中运行的 Amazon EC2 实例。您可以使用同一 VPC 中的现有 EC2 实例作为集群，或创建一个集群。您可以设置到 Amazon DocumentDB 集群的 SSH 隧道。sample-cluster.node.us-east-1.docdb.amazonaws.com在本地计算机上运行以下命令。

```
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
```

-L 标志用于转发本地端口。这是连接到 VPC 外部客户端上运行的任何 BI 工具的先决条件。运行上述步骤后，您可以继续进入所选 BI 工具的后续步骤。

有关 SSH 隧道的更多信息，请参阅以下文档：[使用 SSH 隧道连接到 Amazon DocumentDB](#)。

从 Tableau Desktop Connect 到 Amazon DocumentDB

主题

- [添加 Amazon DocumentDB JDBC 驱动程序 \(p. 481\)](#)
- [使用 Tableau 连接到 Amazon DocumentDB-SSH 隧道 \(p. 482\)](#)

添加 Amazon DocumentDB JDBC 驱动程序

要从 Tableau Desktop 连接到 Amazon DocumentDB，您必须下载并安装 DocumentDB JDBC 驱动程序和 DocumentDB Tableau 连接器。

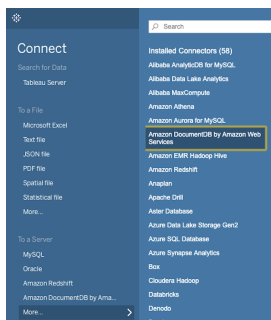
1. 下载 DocumentDB JDBC 驱动程序 JAR 文件并根据操作系统将其复制到以下目录之一：
 - Windows-C:\Program Files\Tableau\Drivers
 - MacOS--/Library/Tableau/Drivers
2. 下载 DocumentDB Tableau 连接器（TACO 文件）并将其复制到我的Tableau 存储库/连接器目录。
 - Windows-C:\Users\[user]\Documents\My Tableau Repository\Connectors
 - MacOS-/Users/[user]/Documents/My Tableau Repository/Connectors

有关更多信息，请参阅[Tableau 文档](#)。

使用 Tableau 连接到 Amazon DocumentDB-SSH 隧道

要从 DocumentDB 集群的 VPC 之外的客户端计算机连接到 Tableau，必须先设置 SSH 隧道，然后执行以下步骤：

1. 启动 Tableau 桌面应用程序。
2. 导航到 Connect (连接) > 转至服务器 > 更多。
3. 选择 Amazon DocumentDB Amazon Web Services 下安装连接器。



使用 Tableau 连接到 Amazon DocumentDB-外部 SSH 隧道

1. 输入所需的连接参数 Hostname、端口、数据库、用户名和密码。下面示例中的连接参数等效于 JDBC 连接字符串：

```
jdbc:documentdb://localhost:27019/test?  
tls=true&tlsAllowInvalidHostnames=true&scanMethod=random&scanLimit=1000&loginTimeoutSe
```

属性集合中分别传递用户名和密码参数。有关连接字符串参数的更多信息，请参阅[Amazon DocumentDB JDBC 驱动程序 github 文档](#)。

Amazon DocumentDB by

×

General

Advanced

Hostname

localhost

Port

27019

Database

customer

Username

jsmith

Password

.....

☒ Enable TLS

☒ Allow Invalid Hostnames (enabling option is less secure)

For support, contact ⓘ

Sign In

2. (可选) 更高级的选项可以在Advanced选项卡。

The screenshot shows the 'Advanced' tab of the 'Amazon DocumentDB by Tableau' configuration window. The window has a title bar with a close button (X). Below the title bar are two tabs: 'General' and 'Advanced', with 'Advanced' being the active tab. The settings include:

- ☐ Enable SSH Tunnel
- ☒ Enable Retry Reads
- ☐ Enable Replica Set Mode
- Read Preference: Primary (dropdown menu)
- Scan Method: Random (dropdown menu)
- Scan Limit: 1000 (text input)
- Login Timeout: 0 (text input)
- Schema Name: _default (text input)

At the bottom, there is a link 'For support, contact' followed by an information icon (i), and a blue 'Sign In' button.

3. 选择 Sign in (登录)。

使用 Tableau 连接到 Amazon DocumentDB-内部 SSH 隧道

Note

如果您不希望使用终端设置 SSH 隧道，则可以使用 Tableau GUI 指定 EC2 实例详细信息，JDBC 驱动程序本质上将使用这些详细信息来创建 SSH 隧道。

1. 在存储库的 Advanced 选项卡，选择启用 SSH 隧道选项以查看进一步的房产。

The screenshot shows the 'Advanced' tab of the 'Amazon DocumentDB by Tableau' connection window. The 'General' tab is also visible. The 'Advanced' tab contains the following settings:

- ☒ Enable SSH Tunnel
- SSH User: ec2-user
- SSH Hostname: [redacted].us-east-2.compute.amazonaws.com
- SSH Private Key File (~/.documentdb/): ec2-literal.pem
- ☒ SSH Strict Host Key Check (disabling option is less secure)
- ☒ Enable Retry Reads
- Read Preference: Primary
- Scan Method: Random

At the bottom, there is a link 'For support, contact' with an information icon, and a 'Sign In' button.

2. 输入SSH 用户、SSH 主机名, 和SSH 私钥文件.
3. (可选) 您可以禁用SSH 严格主机密钥检查选项, 该选项绕过对已知主机文件进行的主机密钥检查。

Note

禁用此选项不太安全, 因为它可能会导致中间人攻击。

The screenshot shows the 'Advanced' tab of the 'Amazon DocumentDB by Tableau' configuration window. The 'General' tab is also visible. The 'Advanced' tab contains the following settings:

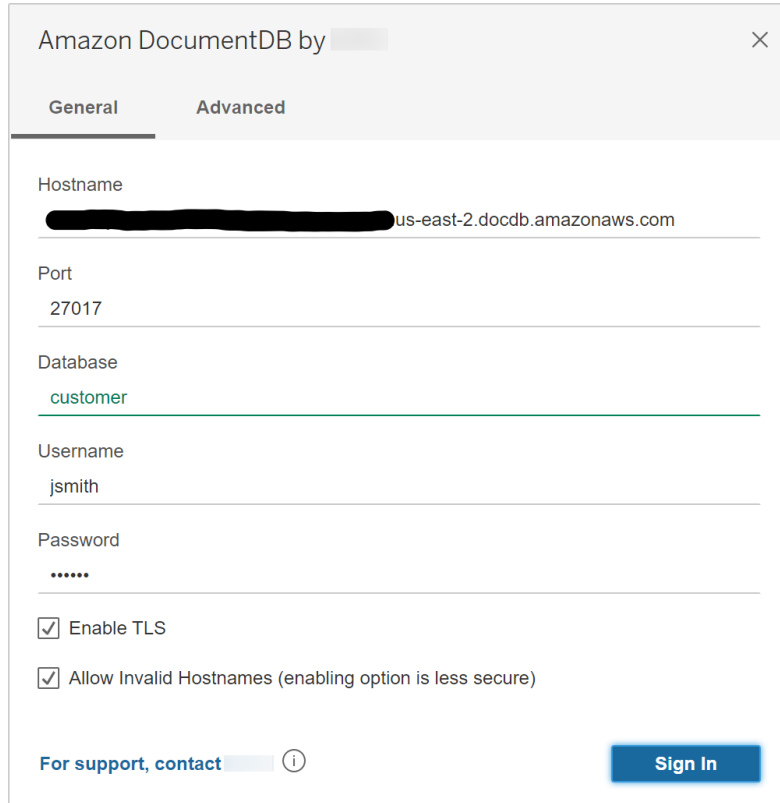
- ☒ Enable SSH Tunnel
- SSH User: ec2-user
- SSH Hostname: [redacted]us-east-2.compute.amazonaws.com
- SSH Private Key File (~/.documentdb/): ec2-literal.pem
- ☐ SSH Strict Host Key Check (disabling option is less secure)
- ☒ Enable Retry Reads
- Read Preference: Primary
- Scan Method: Random

At the bottom, there is a link 'For support, contact' with an information icon, and a blue 'Sign In' button.

4. 输入所需的参数；Hostname、端口、数据库、用户名和密码。

Note

使用内部 SSH 隧道选项时，请确保使用 DocumentDB 群集终端节点而不是本地主机。



Amazon DocumentDB by [redacted] X

General Advanced

Hostname
[redacted]us-east-2.docdb.amazonaws.com

Port
27017

Database
customer

Username
jsmith

Password
.....

☒ Enable TLS

☒ Allow Invalid Hostnames (enabling option is less secure)

For support, contact [redacted] ⓘ

Sign In

5. 选择 Sign in (登录)。

从 DBVisualizer Connect 到 Amazon DocumentDB

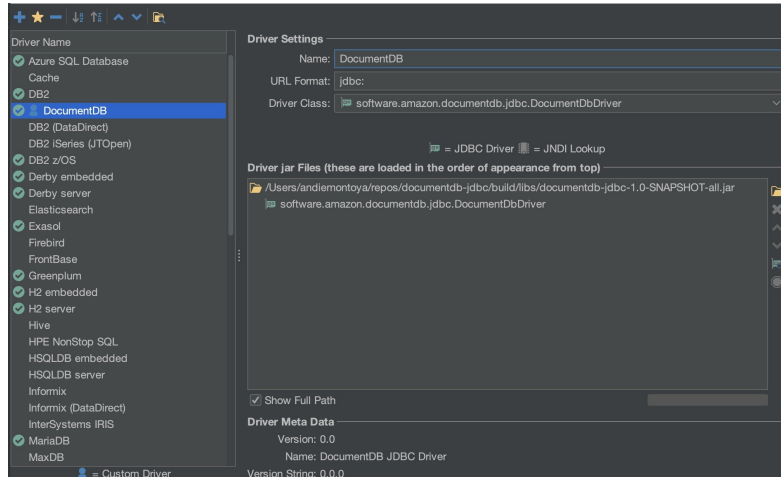
主题

- [添加 Amazon DocumentDB JDBC 驱动程序 \(p. 487\)](#)
- [使用 DBVisualizer 连接到 Amazon DocumentDB \(p. 488\)](#)

添加 Amazon DocumentDB JDBC 驱动程序

要从 DBVisualizer 连接到 Amazon DocumentDB，您必须首先导入 Amazon DocumentDB JDBC 驱动程序

1. 启动 DBVisualizer 应用程序并导航到菜单路径：工具 > 驱动程序管理器...
2. 选择+ (或在菜单中，选择司机 > 创建驱动程序)。
3. 将 Name (名称) 设置为 DocumentDB。
4. SetURL 格式到 `jdbc:documentdb://<host>[:port]/<database>[?option=value[&option=value[...]]]`
5. 选择文件夹按钮，然后选择 Amazon DocumentDB JDBC 驱动程序 JAR 文件，然后选择打开按钮。
6. 验证驱动程序类已将字段设置为 `software.amazon.documentdb.jdbc.DocumentDbDriver`。你的驱动程序管理器设置 DocumentDB 应类似于以下示例。



7. 关闭对话框。将设置 Amazon DocumentDB JDBC 驱动程序并准备使用。

使用 DBVisualizer 连接到 Amazon DocumentDB

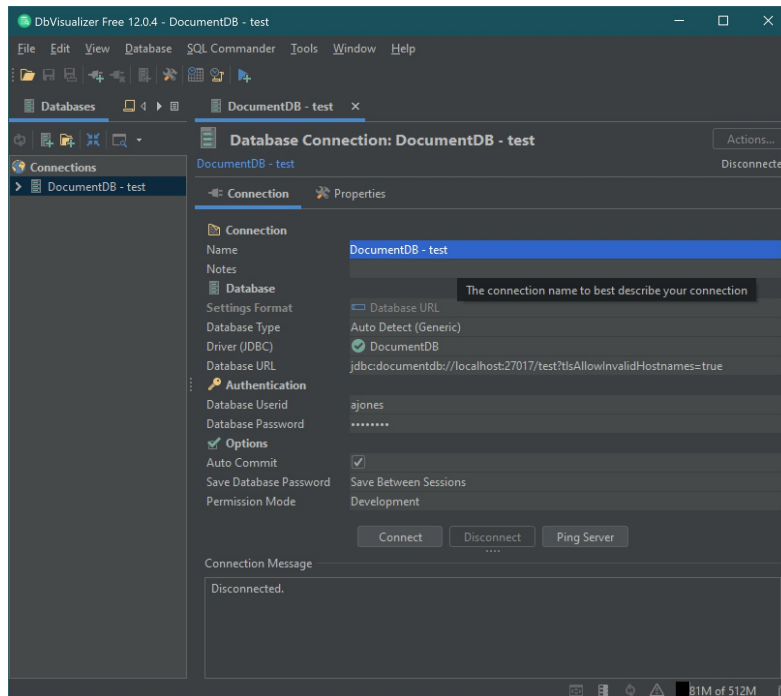
使用 DBVisualizer Connect 到 Amazon DocumentDB

1. 如果您是从 Amazon DocumentDB 集群的 VPC 外部进行连接，请确保已设置 SSH 隧道。
2. 选择数据库 > 创建数据库连接从顶级菜单中。
3. 输入一个描述性的名称名称字段中返回的子位置类型。
4. Set驱动程序 (JDBC) 转到在上一部分中创建的 DocumentDB 驱动程序。
5. Set数据库 URL转入 JDBC 连接字符串。

例如：`jdbc:documentdb://localhost:27017/database?
tlsAllowInvalidHostnames=true。`

6. Set数据库 Userid转到您的 Amazon DocumentDB 用户 ID。
7. Set数据库密码对于用户 ID 的相应密码。

您的“数据库连接”对话框应类似于以下对话框：



8. 选择 Connect (连接)。

自动生成模式

Amazon DocumentDB 是一个文档数据库，因此没有表和架构的概念。但是，Tableau 之类的 BI 工具将希望它连接的数据库能够呈现模式。具体来说，当 JDBC 驱动程序连接需要获取数据库中集合的模式时，它将轮询数据库中的所有集合。驱动程序将确定该集合的模式缓存版本是否已存在。如果缓存版本不存在，它将对文档的集合进行样本并根据以下行为创建架构。

主题

- [模式生成限 \(p. 489\)](#)
- [扫描方法选项 \(p. 489\)](#)
- [DocumentDB 数据类型 \(p. 490\)](#)
- [映射标量文档字段 \(p. 490\)](#)
- [对象和数组数据类型处理 \(p. 493\)](#)

模式生成限

DocumentDB JDBC 驱动程序将标识符的长度限制为 128 个字符。模式生成器可能会截断生成的标识符（表名和列名）的长度，以确保它们符合该限制。

扫描方法选项

可以使用连接字符串或数据源选项修改采样行为。

- ScanMethod =<option>
 - random- (默认) -样本文档按随机顺序返回。
 - idForward-样本文件按照 ID 的顺序返回。
 - idReverse-样本文件按照 ID 的相反顺序返回。

- 所有-对集合中的所有文档进行样本。
- ScanLimit=<n>-要抽样的文件数量。该值必须为正整数。默认值是 。1000. 如果扫描方法设置为所有将忽略此选项。

DocumentDB 数据类型

DocumentDB 服务器支持多种 MongoDB 数据类型。下面列出了受支持的数据类型及其关联的 JDBC 数据类型。

MongoDB 数据类型	DocumentDB 支持	JDBC 数据类型
二进制数据	是	VARBINARY
Boolean	是	BOOLEAN
Double	是	DOUBLE
32 位整数	是	INTEGER
64 位整数	是	BIGINT
字符串	是	VARCHAR
ObjectId	是	VARCHAR
日期	是	TIMESTAMP
Null	是	VARCHAR
正则表达式	是	VARCHAR
时间戳	是	VARCHAR
MinKey	是	VARCHAR
MaxKey	是	VARCHAR
对象	是	虚拟表
数组	是	虚拟表
Decimal128	否	DECIMAL
JavaScript	否	VARCHAR
JavaScript (带范围)	否	VARCHAR
未定义	否	VARCHAR
符号	否	VARCHAR
dbPointer (4.0+)	否	VARCHAR

映射标量文档字段

在从集合中扫描文档样本时，JDBC 驱动程序将创建一个或多个架构来表示集合中的样本。一般来说，文档中的标量字段映射到表架构中的列。例如，在名为团队的集合中和单个文档{ "_id" : "112233", "name" : "Alastair", "age": 25 }，这将映射到模式：

表名称	列名称	数据类型	密钥
球队	团队 ID	VARCHAR	PK
球队	名称	VARCHAR	
球队	年龄	INTEGER	

数据类型冲突促销

扫描抽样文档时，字段的数据类型可能在文档之间不一致。在这种情况下，JDBC 驱动程序会将 JDBC 数据类型提升为适合采样文档中所有数据类型的通用数据类型。

例如：

```
{
  "_id" : "112233",
  "name" : "Alastair", "age" : 25
}

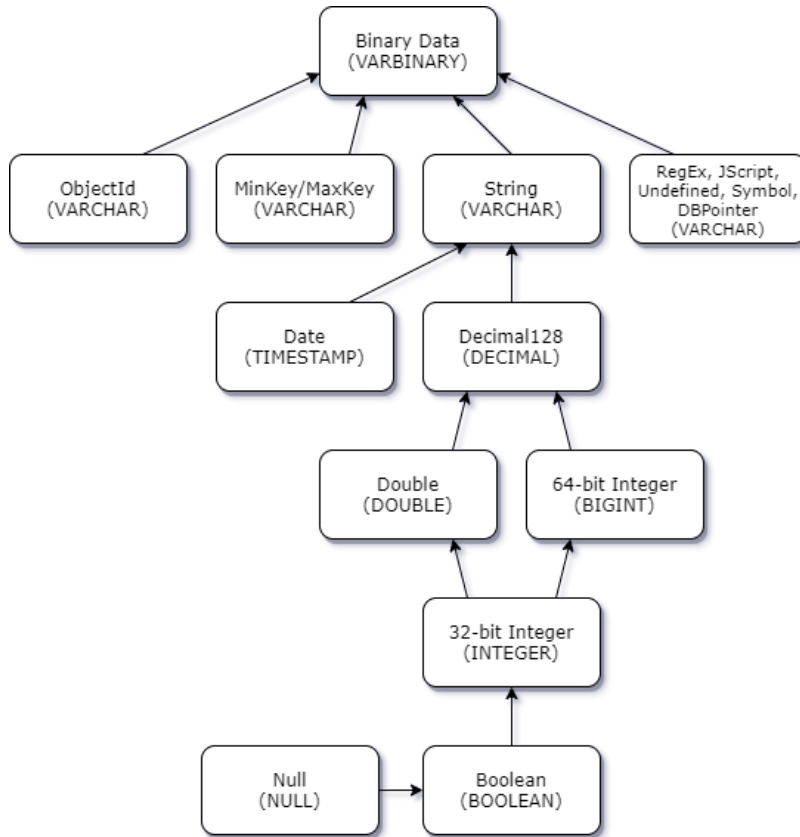
{
  "_id" : "112244",
  "name" : "Benjamin",
  "age" : "32"
}
```

这些区域有：年龄字段在第一个文档中为 32 位整数，但在第二个文档中为字符串。在这里，JDBC 驱动程序将 JDBC 数据类型提升为 VARCHAR，以便在遇到时处理任何一种数据类型。

表名称	列名称	数据类型	密钥
球队	团队 ID	VARCHAR	PK
球队	名称	VARCHAR	
球队	年龄	VARCHAR	

Scalar-Scalar 冲突促销

下图显示了标量标量数据类型冲突的解决方法。



Scalar-复杂类型冲突促销

与标量类型冲突一样，不同文档中的同一字段在复数（数组和对象）和标量（整数、布尔值等）之间可能存在冲突的数据类型。对于这些领域，所有这些冲突都被解决（提升）到 VARCHAR。在这种情况下，数组和对象数据将作为 JSON 表示形式返回。

嵌入式数组-字符串字段冲突示例：

```
{
  "_id": "112233",
  "name": "George Jackson",
  "subscriptions": [
    "Vogue",
    "People",
    "USA Today"
  ]
}
{
  "_id": "112244",
  "name": "Joan Starr",
  "subscriptions": 1
}
```

上面的示例映射到了 ustomer2 表的模式：

表名称	列名称	数据类型	密钥
客户 2	客户 2	VARCHAR	PK

表名称	列名称	数据类型	密钥
客户 2	名称	VARCHAR	
客户 2	订阅	VARCHAR	

和 ustomer1_订阅虚拟表：

表名称	列名称	数据类型	密钥
客户 1_ 订阅	客户 1	VARCHAR	PK/FK
客户 1_ 订阅	订阅_index_lvl0	BIGINT	PK
客户 1_ 订阅	值	VARCHAR	
客户_地址	city	VARCHAR	
客户_地址	region	VARCHAR	
客户_地址	country	VARCHAR	
客户_地址	code	VARCHAR	

对象和数组数据类型处理

到目前为止，我们只描述了标量数据类型的映射方式。对象和数组数据类型（当前）映射到虚拟表。JDBC 驱动程序将创建一个虚拟表来表示文档中的对象或数组字段。映射的虚拟表的名称将连接原始集合的名称，后跟字段的名称，用下划线字符（“_”）分隔。

基表的主键（“_id”）在新虚拟表中使用新名称，并作为关联基表的外键提供。

对于嵌入式数组类型字段，将生成索引列以表示数组中每个级别的数组中的索引。

嵌入式对象字段示例

对于文档中的对象字段，JDBC 驱动程序创建虚拟表的映射。

```
{
  "Collection: customer",
  "_id": "112233",
  "name": "George Jackson",
  "address": {
    "address1": "123 Avenue Way",
    "address2": "Apt. 5",
    "city": "Hollywood",
    "region": "California",
    "country": "USA",
    "code": "90210"
  }
}
```

上面的示例映射到客户表的架构：

表名称	列名称	数据类型	密钥
顾客	客户 ID	VARCHAR	PK

表名称	列名称	数据类型	密钥
顾客	名称	VARCHAR	

和 ustomer_address 虚拟表：

表名称	列名称	数据类型	密钥
客户_地址	客户 ID	VARCHAR	PK/FK
客户_地址	地址 1	VARCHAR	
客户_地址	地址 2	VARCHAR	
客户_地址	city	VARCHAR	
客户_地址	region	VARCHAR	
客户_地址	country	VARCHAR	
客户_地址	code	VARCHAR	

嵌入式数组字段示例

对于文档中的数组字段，JDBC 驱动程序还会创建虚拟表的映射。

```
{
  "Collection: customer1",
  "_id": "112233",
  "name": "George Jackson",
  "subscriptions": [
    "Vogue",
    "People",
    "USA Today"
  ]
}
```

上面的示例映射到了 customer1 表的模式：

表名称	列名称	数据类型	密钥
客户 1	客户 1	VARCHAR	PK
客户 1	名称	VARCHAR	

和 ustomer1_订阅虚拟表：

表名称	列名称	数据类型	密钥
客户1_订阅	客户 1	VARCHAR	PK/FK
客户1_订阅	订阅_index_lvl0	BIGINT	PK
客户1_订阅	值	VARCHAR	

表名称	列名称	数据类型	密钥
客户_地址	city	VARCHAR	
客户_地址	region	VARCHAR	
客户_地址	country	VARCHAR	
客户_地址	code	VARCHAR	

SQL Support 和限制

Amazon DocumentDB JDBC 驱动程序是支持 SQL-92 的子集和一些常见扩展的只读驱动程序。请参阅[SQL 限制文档](#)和[JDBC 限制文档](#)有关更多信息。

故障排除

如果您在使用 Amazon DocumentDB JDBC 驱动程序时遇到问题，请参阅[故障排除指南](#)。

Amazon DocumentDB 配额和限制

本主题介绍 Amazon DocumentDB 的资源配额、限制和命名约束（与 MongoDB 兼容性）。

对于某些管理功能，Amazon DocumentDB 使用与 Amazon Relational Database Service (Amazon RDS) 和 Amazon Neptune 共享的操作技术。

主题

- [支持的 实例类型](#) (p. 496)
- [支持的区域](#) (p. 497)
- [区域配额](#) (p. 497)
- [聚合限制](#) (p. 499)
- [集群限制](#) (p. 499)
- [实例限制](#) (p. 500)
- [命名约束](#) (p. 502)
- [TTL 约束](#) (p. 503)

支持的 实例类型

Amazon DocumentDB 支持按需实例和以下实例类型：

- 内存优化：
 - R6G 实例类型：
db.r6g.large、db.r6g.2xlarge、db.r6g.4xlarge、db.r6g.8xlarge、db.r6g.12xlarge、db.r6g.16xlarge
 - R5 实例类型：
db.r5.large、db.r5.2xlarge、db.r5.4xlarge、db.r5.8xlarge、db.r5.12xlarge、db.r5.16xlarge、db.r5.24xlarge
 - R4 实例类型：
db.r4.large、db.r4.2xlarge、db.r4.4xlarge、db.r4.8xlarge、db.r4.16xlarge
- 突增性能：
 - T4G 实例类型：db.t4g.medium
 - T3 实例类型：db.t3.medium

有关支持的实例类型及其规范的更多信息，请参阅[实例类规格](#) (p. 286)。

支持的区域

Amazon DocumentDB 可在以下内容中找到Amazon地区：

区域名称	区域	可用区 (计算)
美国东部 (俄亥俄)	us-east-2	3
美国东部 (弗吉尼亚北部)	us-east-1	6
美国西部 (俄勒冈)	us-west-2	4
南美洲 (圣保罗)	sa-east-1	3
亚太地区 (孟买)	ap-south-1	3
亚太地区 (首尔)	ap-northeast-2	4
亚太地区 (新加坡)	ap-southeast-1	3
亚太地区 (悉尼)	ap-southeast-2	3
亚太地区 (东京)	ap-northeast-1	3
加拿大 (中部)	ca-central-1	3
中国 (北京) 区域	cn-north-1	3
中国 (宁夏)	cn-northwest-1	3
欧洲 (法兰克福)	eu-central-1	3
欧洲 (爱尔兰)	eu-west-1	3
欧洲 (伦敦)	eu-west-2	3
欧洲 (米兰)	eu-south-1	3
欧洲 (巴黎)	eu-west-3	3
Amazon GovCloud (US)	us-gov-west-1	3

区域配额

对于某些管理功能，Amazon DocumentDB 使用与 Amazon Relational Database Service (Amazon RDS) 和 Amazon Neptune 共享的操作技术。下表包含在 Amazon DocumentDB、Amazon RDS 和 Neptune 之间共享的区域限制。

下面是每个 Amazon 账户在每个区域的限制。

资源	Amazon 默认限制
集群	40
集群参数组	50

资源	Amazon 默认限制
事件订阅	20
实例	40
手动集群快照	100
每个集群的只读副本数	15
子网组	50
每个子网组的子网数	20
每个资源的标签	50
每个实例的 VPC 安全组数	5

如果配额是可调整的，则可以使用服务配额来请求增加配额。有些请求会自动解决，而其他请求则会提交给 Amazon Web Services Support。您可以跟踪提交给的服务配额增加请求的状态。提高服务配额的请求没有得到优先支持。如果您有紧急请求，请联系 [Amazon Web Services Support](#)。有关服务配额的更多信息，请参阅 [什么是服务配额？](#)

要请求提高 Amazon DocumentDB 的配额：

1. 通过 <https://console.amazonaws.cn/servicequotas> 打开服务配额控制台，如有必要，请登录。
2. 在导航窗格中，选择 Amazon services (Amazon 服务)。
3. 从列表中选择 Amazon DocumentDB，或在搜索字段中键入 Amazon DocumentDB。
4. 如果配额是可调整的，您可以选择其单选按钮或其名称，然后从页面右上角选择 Request quota increase (请求增加配额)。
5. 对于 Change quota value (更改配额值)，输入新值。新值必须大于当前值。
6. 选择 Request (请求)。解决请求后，配额的 Applied quota value (应用的配额值) 设置为新值。
7. 要查看任何待处理或最近解决的请求，请从导航窗格选择 Dashboard (控制面板)。对于待处理的请求，请选择请求状态以打开收到的请求。请求的初始状态为 Pending。在状态更改为 Quota requested，您将看到案例编号 Amazon Web Services Support。选择案例编号以打开请求服务单。

聚合限制

下表介绍 Amazon DocumentDB 中的聚合限制。

资源	限制
受支持的最大阶段数	500

集群限制

下表介绍了 Amazon DocumentDB 集群的限制。

资源	限制
集群大小 (所有集合的索引的总和)	64 TiB
集合大小 (所有集合的总和不能超过集群限制) – 不包含索引大小	32 TB
每集群的集合数	100000
每集群的数据库数	100000
数据库大小 (所有数据库的总和不能超过集群限制)	64 TiB
文档嵌套深度	100 个级别
文档大小	16 MB
索引键大小	2,048 字节
每个集合索引一次	64
复合索引中的键	32
单个批处理命令中的最大写入次数	100000
每个集群的用户数	1000

实例限制

下表介绍了每个实例的 Amazon DocumentDB 限制。

实例类型	实例内存 (GiB)	连接限制	光标限制 (每分钟)	开放交易限额 (每分钟)
T3.medium	4	500	30	50
t4g. 中	4	500	30	50
r4.large	15.25	1700	450	不适用
r4.xlarge	30.5	3400	450	不适用
r4.2xlarge	61	6800	450	不适用
r4.4xlarge	122	13600	725	不适用
r4.8xlarge	288	27200	1450	不适用
r4.16xlarge	488	30000	2900	不适用
r5.large	16	1700	450	200
r5.xlarge	32	3500	450	400
r5.2xlarge	64	7100	450	800
r5.4xlarge	128	14200	760	1600
r5.8xlarge	256	28400	1520	3200
r5.12xlarge	383	30000	2280	4800
r5.16xlarge	512	30000	3040	6400
r5.24xlarge	768	30000	4560	9600
r6g. 大	16	1700	450	200
r6g.xLARGE	32	3500	450	400
r6g.2xLL	64	7100	450	800
r6g.4xLL	128	14200	760	1600
r6g.8xLL	256	28400	1520	3200
r6G.12x 大	383	30000	2280	4800
r6g.16x 大	512	30000	3040	6400

您可以使用以下内容监控和警报每个实例的限制 CloudWatch 指标。有关 Amazon DocumentDB 的更多信息 CloudWatch 指标，请参阅[使用 CloudWatch 监控 Amazon DocumentDB \(p. 376\)](#)。

限制	CloudWatch 指标
实例内存	FreeableMemory

限制	CloudWatch 指标
连接	数据库连接最大
游标	数据库游标最大
事务	交易 OpenMAX

命名约束

下表介绍 Amazon DocumentDB 中的命名约束。

资源	默认限制
集群标识符	- 长度为 [1-63] 个字母、数字或连字符。 - 第一个字符必须是字母。 - 不能以连字符结束或包含两个连续连字符。 - 对于每个集群（在 Amazon RDS、Amazon Neptune 和 Amazon DocumentDB 中）必须是唯一的 Amazon 在每个区域中的账户。
集合名称：<col>	长度为 [1-57] 个字符。
数据库名称：<db>	长度为 [1-63] 个字符。
完全限定集合名称：<db>.<col>	长度为 [3—120] 个字符。
完全限定索引名称：<db>.<col>.\$<index>	长度为 [6—127] 个字符。
索引名称：<col>.\$<index>	长度为 [3—63] 个字符。
实例标识符	- 长度为 [1-63] 个字母、数字或连字符 - 第一个字符必须是字母 - 不能以连字符结束或包含两个连续连字符 - 对于每个实例（在 Amazon RDS、Amazon Neptune 和 Amazon DocumentDB 中）必须是唯一的 Amazon 在每个区域中的账户。
主密码	- 长度为 [8-100] 个可打印 ASCII 字符。 - 可以使用任何可打印 ASCII 字符，以下字符除外： / (正斜杠) " (双引号) @ (@ 符号)
主用户名	- 长度为 [1-63] 个字母数字字符。 - 第一个字符必须是字母。 - 不能是数据库引擎的保留字。
参数组名称	- 长度为 [1-255] 个字母数字字符。 - 第一个字符必须是字母。 - 不能以连字符结束或包含两个连续连字符。

TTL 约束

在特定时间范围内无法保证从 TTL 索引中删除，只能尽力而为。实例资源利用率、文档大小和总体吞吐量等因素会影响 TTL 删除的时间。

查询

本节介绍了使用 Amazon DocumentDB 进行查询的所有方面。

主题

- [查询文档 \(p. 504\)](#)
- [使用 Amazon DocumentDB 查询地理空间数据 \(p. 506\)](#)
- [查询计划 \(p. 509\)](#)
- [EXplain 结果 \(p. 510\)](#)

查询文档

有时，您可能需要查看在线商店的库存，这样客户就能看到并购买您销售的物品。查询集合相对容易，无论您想要集合中的所有文档，还是仅需要那些满足特定标准的文档。

要查询文档，请使用 `find()` 操作。`find()` 命令具有单个文档参数，该参数定义了在选择要返回的文档时要使用的标准。`find()` 的输出是一个文档，其格式为一行文本，不含换行符。要格式化输出文档，从而更加轻松地读取，请使用 `find().pretty()`。本主题中的所有示例都使用 `.pretty()` 设置输出的格式。

以下代码示例使用您插入到 `example` 前两个练习中的收藏 — `insertOne()` 和 `insertMany()` 位于的“添加文档”部分 [使用文档](#)。

主题

- [检索集合中的所有文档 \(p. 504\)](#)
- [检索与字段值匹配的文档 \(p. 504\)](#)
- [检索与嵌入文档匹配的文档 \(p. 505\)](#)
- [检索与嵌入文档中的字段值匹配的文档 \(p. 505\)](#)
- [检索与数组匹配的文档 \(p. 505\)](#)
- [检索与数组中的值匹配的文档 \(p. 505\)](#)
- [使用运算符检索文档 \(p. 505\)](#)

检索集合中的所有文档

要检索集合中的所有文档，请将 `find()` 操作和空查询文档结合使用。

以下查询返回 `example` 集合中的所有文档。

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

检索与字段值匹配的文档

要检索与字段和值匹配的所有文档，请将 `find()` 操作和查询文档（标识要匹配的字段和值）结合使用。

通过使用前述文档，此查询将返回其中“Item”字段等于“Pen”的所有文档。

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

检索与嵌入文档匹配的文档

要查找与嵌入文档匹配的所有文档，请将 `find()` 操作和查询文档（指定嵌入文档名称和嵌入文档的所有字段和值）结合使用。

在与嵌入文档匹配时，该文档的嵌入文档的名称必须与查询中的名称相同。此外，嵌入文档中的字段和值必须与查询匹配。

以下查询仅返回“Poster Paint”文档。这是因为“Pen”具有不同的“OnHand”和“MinOnHand”值，并且“Spray Paint”比查询文档多一个字段（OrderQty）。

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

检索与嵌入文档中的字段值匹配的文档

要查找与嵌入文档匹配的所有文档，请将 `find()` 操作和查询文档（指定嵌入文档名称和嵌入文档的所有字段和值）结合使用。

考虑到上述文档，以下查询使用“点表示法”来指定嵌入文档和感兴趣的字段。将返回所有与这些内容匹配的文档，而不管嵌入文档中可能存在哪些其他字段。此查询将返回“Poster Paint”和“Spray Paint”，因为它们与指定的字段和值匹配。

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

检索与数组匹配的文档

要查找所有与数组匹配的文档，请将 `find()` 操作和您感兴趣的数组名称以及数组中的所有值结合使用。此查询将返回所有包含带该名称的数组（其中数组值和顺序与查询中的完全相同）的文档。

以下查询仅返回“Pen”，因为“Poster Paint”具有其他颜色（White），并且“Spray Paint”具有顺序不同的颜色。

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

检索与数组中的值匹配的文档

要查找所有具有特定数组值的文档，请将 `find()` 操作与您感兴趣的数组名称和值结合使用。

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

上述操作将返回所有三个文档，因为它们都有一个名为 `Colors` 的数组，并且此数组中的某个位置具有“Red”值。如果您指定值“White”，则查询将仅返回“Poster Paint”。

使用运算符检索文档

以下查询返回“Inventory.OnHand”值小于 50 的所有文档。

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

有关支持的查询运算符的列表，请参阅 [查询和投影运算符 \(p. 97\)](#)。

使用 Amazon DocumentDB 查询地理空间数据

本节介绍了如何使用 Amazon DocumentDB 查询地理空间数据。阅读本节后，您将能够回答如何在 Amazon DocumentDB 中存储、查询和索引地理空间数据。

主题

- [概览 \(p. 1\)](#)
- [索引和存储地理空间数据 \(p. 506\)](#)
- [查询地理空间数据 \(p. 507\)](#)
- [限制 \(p. 509\)](#)

概览

地理空间的常见用例包括从数据中进行邻近分析。例如，“查找西雅图 50 英里范围内的所有机场”，或“从给定地点找到最近的餐厅”。Amazon DocumentDB 使用[GeoJSON 规范](#)表示地理空间数据。GeoJSON 是坐标空间中形状的 JSON 格式化的开源规范。GeoJSON 坐标可捕获经度和纬度，表示类似地球的球体上的位置。

索引和存储地理空间数据

Amazon DocumentDB 使用“Point” GeoJSON 类型来存储地理空间数据。每个 GeoJSON 文档（或子文档）通常由两个字段组成：

- 类型-表示的形状，通知 Amazon DocumentDB 如何解释“坐标”字段。目前，Amazon DocumentDB 只支持积分
- 坐标— 表示为数组中对象的纬度和经度对 — [经度、纬度]

Amazon DocumentDB 还使用 2dsphere 索引来索引地理空间数据。Amazon DocumentDB 支持索引点。Amazon DocumentDB 支持使用 2dsphere 索引进行邻近查询。

让我们考虑一个您正在构建食品配送服务应用程序的情况。您想在 Amazon DocumentDB 中存储各种餐厅的纬度和经度对。为此，首先我们建议您在地理空间字段上创建一个包含纬度和经度对的索引。

```
use restaurantsdb
db.usarestaurants.createIndex({location:"2dsphere"})
```

此命令的输出如下所示：

```
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

创建索引后，您可以开始将数据插入 Amazon DocumentDB 集合中。

```
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
    ]
  }
});
```

查询地理空间数据

Amazon DocumentDB 支持对地理空间数据进行邻近、包含和交叉点查询。邻近查询的一个很好的例子是查找小于一定距离且距离另一个点（城市）大于一段距离的所有点（所有机场）。包含查询的一个很好的例子是查找位于指定区域/多边形（纽约州）中的所有点（所有机场）。交叉点查询的一个很好的例子是查找与点（城市）相交的多边形（州）。您可以使用以下地理空间运营商从数据中获取见解。

- **\$nearSphere**-\$nearSphere 是一个查找运算符，支持从距离 GeoJSON 点最近到最远的点查找点。
- **\$geoNear**-\$geoNear 是一种聚合运算符，支持计算距 GeoJSON 点的距离（以米为单位）。
- **\$minDistance**-\$minDistance 是与结合使用的查找运算符 \$nearSphere 要么 \$geoNear 以筛选至少与中心点指定的最小距离的文档。
- **\$maxDistance**-\$maxDistance 是与结合使用的查找运算符 \$nearSphere 要么 \$geoNear 以筛选最多距离中心点指定最大距离的文档。
- **\$geoWithin**-\$geoWithin 是一个查找运算符，支持查找具有完全存在于指定形状（如多边形）内的地理空间数据的文档。
- **\$geoIntersects**-\$geoIntersects 是一个查找运算符，支持查找其地理空间数据与指定 GeoJSON 对象交叉的文档。

Note

\$geoNear 和 \$nearSphere 需要在邻近查询中使用的 GeoJSON 字段上的 2dsphere 索引。

示例 1

在此示例中，您将学习如何查找按距地址（点）最近的距离排序的所有餐厅（积分）。

要执行这样的查询，您可以使用\$geoNear来计算一组点与另一个点的距离。您还可以添加distanceMultiplier以测量以公里为单位的距离。

```
db.usarestaurants.aggregate([
  {
    "$geoNear":{
      "near":{
        "type":"Point",
        "coordinates":[
          -122.3516,
          47.6156
        ]
      },
      "spherical":true,
      "distanceField":"DistanceKilometers",
      "distanceMultiplier":0.001
    }
  }
])
```

上面的命令将返回按距离指定点的距离（最接近）排序的餐厅。此命令的输出应类似如下所示

```
{ "_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 }
```

示例 2

在此示例中，您将学习如何在距离特定地址（点）2 公里范围内找到所有餐厅（积分）。要执行这样的查询，您可以使用\$nearSphere最小值\$minDistance最大值\$maxDistance来自 GeoJSON 点

```
db.usarestaurants.find({
  "location":{
    "$nearSphere":{
      "$geometry":{
        "type":"Point",
        "coordinates":[
          -122.3516,
          47.6156
        ]
      },
      "$minDistance":1,
      "$maxDistance":2000
    }
  }
},
{
  "name":1
})
```

以上命令将返回距指定点最多 2 公里的餐厅。此命令的输出应类似如下所示

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "name" : "Noodle House" }
```

限制

Amazon DocumentDB 不支持查询或索引多边形、LineString、MultiPOLYGON、MultiLineString 和 GeometryCollection。

查询计划

如何查看查询计划的 executionStats ?

在确定查询的执行速度低于预期速度的原因时，了解查询计划的 executionStats 会很有用。executionStats 提供从特定阶段返回的文档数量 (nReturned)、在每个阶段花费的执行时间 (executionTimeMillisEstimate) 以及生成查询计划所需的时间长度 (planningTimeMillis)。您可以确定查询中最耗时的阶段，以帮助您根据 executionStats 的输出集中精力完成优化工作，如以下查询示例所示。executionStats 参数当前不支持 update 和 delete 命令。

Note

Amazon DocumentDB 在利用分布式、容错、自修复的存储系统的专用数据库引擎上模拟 MongoDB 3.6 API。因此，查询计划和的输出 explain() Amazon DocumentDB 和 MongoDB 之间可能有所不同。希望控制其查询计划的客户可以使用 \$hint 运算符强制选择首选索引。

在 explain() 命令下运行要改进的查询，如下所示。

```
db.runCommand({explain: {query document}}).  
explain("executionStats").executionStats;
```

以下是操作示例。

```
db.fish.find({}).limit(2).explain("executionStats");
```

此操作的输出将类似于下文。

```
{  
  "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
    }
  }
}

```

如果您只想看到上面查询的 `executionStats`，您可以使用以下命令。对于较小的集合，如果性能增益微乎其微，Amazon DocumentDB 查询处理器可以选择不使用索引。

```
db.fish.find({}).limit(2).explain("executionStats").executionStats;
```

查询计划缓存

为了优化性能并缩短规划持续时间，Amazon DocumentDB 在内部缓存查询计划。这样就可以使用缓存的计划直接执行形状相同的查询。

但是，此缓存有时可能会导致同一查询的随机延迟；例如，运行通常需要一秒钟的查询偶尔需要十秒钟。这是因为随着时间的推移，读取器实例缓存了各种形状的查询，从而消耗内存。如果您遇到这种随机缓慢，则无需执行任何操作即可释放内存——系统将为您管理内存使用情况，一旦内存达到一定阈值，它将自动释放。

EXplain 结果

如果您想返回有关查询计划的信息，Amazon DocumentDB 支持详细模式 `queryPlanner`。这些区域有：`explain` 结果以类似于以下格式返回优化程序选择的选定查询计划：

```

{
  "queryPlanner" : {
    "plannerVersion" : <int>,
    "namespace" : <string>,
    "winningPlan" : {
      "stage" : <STAGE1>,
      ...
      "inputStage" : {
        "stage" : <STAGE2>,
        ...
        "inputStage" : {
          ...
        }
      }
    }
  }
}

```

```
}
```

以下各节将定义常见explain结果。

主题

- [扫描和筛选阶段 \(p. 511\)](#)
- [索引交叉 \(p. 512\)](#)
- [索引 \(p. 512\)](#)
- [多指数交口/联盟 \(p. 512\)](#)
- [复合索引 \(p. 513\)](#)
- [排序阶段 \(p. 513\)](#)
- [阶段 \(p. 513\)](#)

扫描和筛选阶段

优化程序可以选择下列扫描之一：

COLSCAN

这个阶段是顺序收藏扫描。

```
{
  "stage" : "COLLSCAN"
}
```

IXSCAN

此阶段将扫描索引键。优化程序可能会在此阶段内检索文档，这可能会导致稍后追加 FETCH 阶段。

```
db.foo.find({"a": 1})
{
  "stage" : "IXSCAN",
  "direction" : "forward",
  "indexName" : <idx_name>
}
```

FETCH

如果优化程序在 IXSCAN 以外的阶段中检索文档，则结果将包括 FETCH 阶段。例如，上面的 IXSCAN 查询可能会导致 FETCH 和 IXSCAN 阶段的组合：

```
db.foo.find({"a": 1})
{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXSCAN",
    "indexName" : <idx_name>
  }
}
```

IXONLYSAN 仅扫描索引键。创建复合索引不会避免 FETCH。

索引交叉

IXAND

如果 Amazon DocumentDB 可以利用索引交叉点，则可能包括一个具有 IXSCAN InputStage 数组的 IXAND 舞台。例如，我们可能会看到如下输出：

```
{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXAND",
    "inputStages" : [
      {
        "stage" : "IXSCAN",
        "indexName" : "a_1"
      },
      {
        "stage" : "IXSCAN",
        "indexName" : "b_1"
      }
    ]
  }
}
```

索引

IXOR

与索引交叉点类似，Amazon DocumentDB 可能包括IXOR阶段inputStages数组\$or运算符。

```
db.foo.find({"$or": [{"a": {"$gt": 2}}, {"b": {"$lt": 2}}]})
```

对于上述查询，解释输出可能如下所示：

```
{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXOR",
    "inputStages" : [
      {
        "stage" : "IXSCAN",
        "indexName" : "a_1"
      },
      {
        "stage" : "IXSCAN",
        "indexName" : "b_1"
      }
    ]
  }
}
```

多指数交口/联盟

Amazon DocumentDB 可以将多个索引交集或联合阶段组合在一起，然后获取结果。例如：

```
{
```

```
{
  "stage" : "FETCH",
  "inputStage" : {
    "stage" : "IXOR",
    "inputStages" : [
      {
        "stage" : "IXSCAN",
        ...
      },
      {
        "stage" : "IXAND",
        "inputStages" : [
          {
            "stage" : "IXSCAN",
            ...
          },
          {
            "stage" : "IXSCAN",
            ...
          }
        ]
      }
    ]
  }
}
```

索引交集或联合阶段的使用不受索引类型（稀疏、复合等）的影响。

复合索引

Amazon DocumentDB 复合索引索引的使用不受限于索引字段的开头子集；它可以使用带有后缀部分的索引，但可能不是很有效。

例如，的复合索引{ a: 1, b: -1 }可以支持以下所有三个查询：

```
db.orders.find( { a: 1 } )
db.orders.find( { b: 1 } )
db.orders.find( { a: 1, b: 1 } )
```

排序阶段

如果请求的排序键上有索引，DocumentDB 可以使用索引来获取订单。在这种情况下，结果将不会包含SORT舞台，而是IXSCAN阶段。如果优化程序倾向于普通的排序，它将包括这样的阶段：

```
{
  "stage" : "SORT",
  "sortPattern" : {
    "a" : 1,
    "b" : -1
  }
}
```

阶段

Amazon DocumentDB 支持两种不同的群组策略：

- SORT_AGGREGATE: On disk sort aggregate.

- `HASH_AGGREGATE`: In memory hash aggregate.

排查 Amazon DocumentDB

以下部分提供了有关如何排查您在使用 Amazon DocumentDB 时（兼容 MongoDB）时可能遇到的问题的信息。

主题

- [连接问题 \(p. 515\)](#)
- [索引创建 \(p. 518\)](#)
- [性能和资源利用率 \(p. 519\)](#)

连接问题

连接时遇到问题？以下内容介绍一些常见场景以及如何解决这些问题

无法 Connect 到 Amazon DocumentDB 终端节点

在尝试连接到 Amazon DocumentDB 时，以下错误消息是您可能收到的最常见的错误消息之一。

```
connecting to: mongodb://docdb-2018-11-08-21-47-27.cluster-ccuszb3pn5e.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-ccuszb3pn5e.us-east-1.docdb.amazonaws.com:27017,
connection attempt failed :
connect@src/mongo/shell/mongo.js:237:13
@(connect):1:6
exception: connect failed
```

此错误消息通常意味着您的客户端（此示例中为 mongo shell）无法访问 Amazon DocumentDB 终端节点。以下几个原因可能会导致出现此情况：

主题

- [从公有终端节点连接 \(p. 515\)](#)
- [跨区域连接 \(p. 516\)](#)
- [从不同的 Amazon VPC 连接 \(p. 516\)](#)
- [安全组阻止入站连接 \(p. 516\)](#)

从公有终端节点连接

您正在尝试直接从笔记本电脑或本地开发计算机连接到 Amazon DocumentDB 集群。

尝试直接从公有终端节点（例如笔记本电脑或本地开发计算机）连接到 Amazon DocumentDB 集群将失败。Amazon DocumentDB 仅面向 Virtual Private Cloud (VPC)，并且当前不支持公有终端节点。因此，您无法从笔记本电脑或 VPC 外部的本地开发环境直接连接到 Amazon DocumentDB 集群。

要从 Amazon VPC 外部连接到 Amazon DocumentDB 集群，您可使用 SSH 隧道。有关更多信息，请参阅[从 Amazon VPC 外部连接到 Amazon DocumentDB 集群 \(p. 457\)](#)。此外，如果您的开发环境位于不同的 Amazon VPC 中，您还可以使用 VPC 对等，并从同一区域或不同区域中的另一个 Amazon DocumentDB 集群。

跨区域连接

您正在尝试连接到另一个区域中的 Amazon DocumentDB 集群。

如果您尝试从集群区域以外的区域中的 Amazon EC2 实例（例如，尝试从美国西部（俄勒冈）区域 (us-east-1) 连接到集群，例如，尝试从美国西部（俄勒冈）区域 (us-east-1) 连接到集群将失败。

要验证您的 Amazon DocumentDB 集群的区域，请运行以下命令。区域位于终端节点中。

```
aws docdb describe-db-clusters \  
  --db-cluster-identifier sample-cluster \  
  --query 'DBClusters[*].Endpoint'
```

此操作的输出将类似于下文。

```
[  
  "sample-cluster.node.us-east-1.docdb.amazonaws.com"  
]
```

要验证您的 EC2 实例的区域，请运行以下命令。

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].Placement.AvailabilityZone'
```

此操作的输出将类似于下文。

```
[  
  [  
    "us-east-1a"  
  ]  
]
```

从不同的 Amazon VPC 连接

您正在尝试从 VPC（不同于您的集群部署到的 Amazon VPC）连接到 Amazon DocumentDB 集群。

如果您的 Amazon DocumentDB 集群和 Amazon EC2 实例位于同一 Amazon Web Services 区域，但不在同一 Amazon VPC 中，否则除非在两个 Amazon VPC 之间启用了 VPC 对等，否则无法直接连接到 Amazon DocumentDB 集群。

要验证您的 Amazon DocumentDB 实例的 Amazon VPC，请运行以下命令。

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-instance \  
  --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

要验证您的 Amazon EC2 实例的 Amazon VPC，请运行以下命令。

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].VpcId'
```

安全组阻止入站连接

您正在尝试连接到 Amazon DocumentDB 集群，并且该集群的安全组不允许集群端口（默认端口：27017）上的入站连接。

假设您的 Amazon DocumentDB 集群和 Amazon EC2 实例位于同一区域和 Amazon VPC 中，并且使用同一 Amazon VPC 安全组。如果您无法连接到 Amazon DocumentDB 集群，这可能是由于您的集群的安全组（即防火墙）不允许您为 Amazon DocumentDB 集群选择的端口（默认端口为 27017）上的入站连接。

要验证您的 Amazon DocumentDB 集群的端口，请运行以下命令。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[DBClusterIdentifier,Port]'
```

要获取您的集群的 Amazon DocumentDB 安全组，请运行以下命令。

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[VpcSecurityGroups[*],VpcSecurityGroupId]'
```

要检查您的安全组的入站规则，请参阅 Amazon EC2 文档中的以下主题：

- 为您的 [Linux 实例授权入站流量](#)
- 为您的 [Windows 实例授权入站流量](#)

测试与 Amazon DocumentDB 实例的连接

您可以使用常见的 Linux 或 Windows 工具测试与集群的连接。

从 Linux 或 Unix 终端测试此连接，方法为输入以下命令（将 `cluster-endpoint` 替换为终端节点并将 `port` 替换为实例的端口）。

```
nc -zv cluster-endpoint port
```

以下是示例操作和返回值的示例：

```
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!
```

连接到无效终端节点

在连接到 Amazon DocumentDB 集群并且使用的是无效的集群终端节点时，将出现与以下内容下类似的错误。

```
mongo --ssl \
  --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 \
  --sslCAFile rds-combined-ca-cn-bundle.pem \
  --username <user-name> \
  --password <password>
```

该输出类似于以下示例：

```
MongoDB shell version v3.6
connecting to: mongod://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
```

要获取集群的有效终端节点，请运行以下命令：

```
aws docdb describe-db-clusters \
  --db-cluster-identifier sample-cluster \
  --query 'DBClusters[*].[Endpoint,Port]'
```

要获取实例的有效终端节点，请运行以下命令：

```
aws docdb describe-db-instances \
  --db-instance-identifier sample-instance \
  --query 'DBInstances[*].[Endpoint.Address,Endpoint.Port]'
```

有关更多信息，请参阅[了解 Amazon DocumentDB 终端节点 \(p. 343\)](#)。

索引创建

以下主题说明了索引或后台索引构建失败时的操作。

主题

- [索引构建失败 \(p. 518\)](#)
- [后台索引构建延迟问题和失败 \(p. 518\)](#)

索引构建失败

Amazon DocumentDB 在索引创建过程中，将使用实例上的本地存储。您可以使用 FreeLocalStorageCloudWatch 指标 (CloudWatch -> Metrics -> DocDB -> Instance Metrics)。如果索引构建用尽所有本地磁盘并且失败，您将收到错误。在将数据迁移到 Amazon DocumentDB 时，我们建议您先创建索引，然后再插入数据。有关迁移策略和创建索引的更多信息，请参阅[迁 Amazon DocumentDB \(p. 112\)](#)在 Amazon DocumentDB 文档和博客中：[使用离线方法从 MongoDB 迁移到 Amazon DocumentDB](#)。

在现有集群上创建索引时，如果索引构建花费的时间超过预期或失败，我们建议您扩展实例来创建索引，创建好索引后，再缩减回来。Amazon DocumentDB 使您能够在几分钟内快速扩展实例大小，使用 Amazon Web Services Management Console 或者 Amazon CLI。有关更多信息，请参阅[管理实例类 \(p. 283\)](#)。使用每秒实例定价时，您只需为使用 1 秒以上的资源付费。

后台索引构建延迟问题和失败

在启动索引构建之前启动的对主实例的所有查询完成执行之前，才会启动 Amazon DocumentDB 中的后台索引构建。如果有长时间运行的查询，则后台索引构建将阻塞直到查询完成，因此可能需要比预期的更长的时间才能完成。即使收藏集为空也是如此。

前景索引构建不会显示相同的阻塞行为。相反，前景索引构建会在索引构建完成之前对集合进行独占锁定。因此，为了在空集合上创建索引并避免阻止任何长时间运行的查询，我们建议使用前景索引构建。

Note

在任何给定时间，Amazon DocumentDB 仅允许在一个集合中构建一个后台索引。如果后台索引构建期间在相同集合上出现 DDL (数据定义语言) 操作 (如 `createIndex()` 或 `dropIndex()`)，后台索引构建将失败。

性能和资源利用率

本部分提供了 Amazon DocumentDB 部署中的常见诊断问题和解决方案。提供的示例使用 mongo shell，并限制在单个实例范围内。要查找实例终端节点，请参阅[了解 Amazon DocumentDB 终端节点 \(p. 343\)](#)。

主题

- [如何查找并终止长时间运行或受阻的查询？ \(p. 519\)](#)
- [如何查看查询计划和优化查询？ \(p. 520\)](#)
- [如何列出实例上正在运行的所有操作？ \(p. 521\)](#)
- [如何知道查询何时进行？ \(p. 523\)](#)
- [我该如何确定为什么系统突然运行缓慢？ \(p. 524\)](#)
- [如何确定一个或多个集群实例上 CPU 使用率高的原因？ \(p. 525\)](#)
- [如何确定实例上的打开游标？ \(p. 526\)](#)
- [如何确定当前的 Amazon DocumentDB 引擎版本？ \(p. 526\)](#)
- [如何识别未使用的索引？ \(p. 526\)](#)
- [如何识别缺失的索引？ \(p. 527\)](#)
- [有用查询的摘要 \(p. 528\)](#)

如何查找并终止长时间运行或受阻的查询？

用户查询可能因查询计划不够理想而运行缓慢，或者由于资源争用而受阻。

要查找因查询计划不够理想而速度缓慢的长时间运行的查询，或者由于资源争用而受阻的查询，请使用 `currentOp` 命令。可以筛选该命令以帮助缩小要终止的相关查询的列表。长时间运行的查询必须拥有关联的 `opid`，才能够终止查询。

以下查询使用 `currentOp` 命令列出受阻或运行时间超过 10 秒的所有查询。

```
db.adminCommand({
  aggregate: 1,
  pipeline: [
    {$currentOp: {}},
    {$match:
      {$or: [
        {secs_running: {$gt: 10}},
        {WaitState: {$exists: true}}]}}}
    {$project: {_id:0, opid: 1, secs_running: 1}}],
  cursor: {}
});
```

接下来，您可以缩小查询，以查找运行时间超过 10 秒的查询的 `opid` 并终止它。

查找并终止运行时间超过 10 秒的查询

1. 查找查询的 `opid`。

```
db.adminCommand({
  aggregate: 1,
  pipeline: [
    {$currentOp: {}},
    {$match:
      {$or:
        [{secs_running: {$gt: 10}},
        {WaitState: {$exists: true}}]}}}
    {$project: {_id:0, opid: 1, secs_running: 1}}],
  cursor: {}
});
```



```
});
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "waitedMS" : NumberLong(0),
  "cursor" : {
    "firstBatch" : [
      {
        "opid" : 24646,
        "secs_running" : 12
      }
    ],
    "id" : NumberLong(0),
    "ns" : "admin.$cmd"
  },
  "ok" : 1
}
```

2. 使用 killOp 操作终止查询。

```
db.adminCommand({killOp: 1, op: 24646});
```

如何查看查询计划和优化查询？

如果查询运行缓慢，可能是因为查询执行需要对集合进行完全扫描以选择相关的文档。有时，可通过创建合适的索引提高查询的运行速度。要检测此情形并确定要在其中创建索引的字段，请使用explain命令。

Note

Amazon DocumentDB 在利用分布式、容错、自修复的存储系统的专用数据库引擎上模拟 MongoDB 3.6 API。因此，查询计划和的输出explain()Amazon DocumentDB 和 MongoDB 之间可能有所不同。希望控制其查询计划的客户可以使用 \$hint 运算符强制选择首选索引。

在 explain 命令下运行要改进的查询，如下所示。

```
db.runCommand({explain: {<query document>}})
```

以下是操作示例。

```
db.runCommand({explain:{
  aggregate: "sample-document",
  pipeline: [{ $match: {x: { $eq: 1 }}}],
  cursor: {batchSize: 1}}
});
```

此操作的输出将类似于下文 (JSON 格式)。

```
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "db.test",
    "winningPlan" : {
      "stage" : "COLLSCAN"
    }
  },
  "serverInfo" : {
    "host" : "...",
```

```
    "port" : ...,
    "version" : "...",
  },
  "ok" : 1
}
```

上述输出表明，\$match 阶段要求扫描整个集合并检查每个文档中的字段 "x" 是否等于 1。如果集合中有很多文档，集合扫描将非常慢，因此整体查询性能非常低。因此，explain 命令输出中的 "COLLSCAN" 的存在表明，可以通过创建合适的索引来提高查询性能。

在本例中，查询检查所有文档中的字段 "x" 是否等于 1。因此，在字段 "x" 上创建索引，可使查询避免对集合进行完全扫描，并可使用索引更快地返回相关文档。

在字段 "x" 上创建索引后，explain 输出如下所示。

```
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "db.test",
    "winningPlan" : {
      "stage" : "IXSCAN",
      "indexName" : "x_1",
      "direction" : "forward"
    },
  },
  "serverInfo" : {
    "host" : "...",
    "port" : ...,
    "version" : "...",
  },
  "ok" : 1
}
```

因此，在 "x" 字段上创建索引后，\$match 阶段即可使用索引扫描来减少必须对其评估 "x = 1" 谓词的文档的数量。

对于较小的集合，如果性能增益微乎其微，Amazon DocumentDB 查询处理器可以选择不使用索引。

如何列出实例上正在运行的所有操作？

作为用户或主用户，您经常需要列出实例上当前正在运行的所有操作，以进行诊断和故障排除。（有关管理用户的信息，请参阅[Amazon DocumentDB 用户 \(p. 154\)](#)。）

使用mongo shell，您可以使用以下查询列出 Amazon DocumentDB 实例上正在运行的所有操作。

```
db.adminCommand({currentOp: 1, $all: 1});
```

此查询将返回当前在实例上运行的所有用户查询和内部系统任务的完整列表。

此操作的输出将类似于下文（JSON 格式）。

```
{
  "inprog" : [
    {
      "desc" : "INTERNAL"
    },
    {
      "desc" : "TTLMonitor",
      "active" : false
    },
  ],
}
```

```
{
  "desc" : "GARBAGE_COLLECTION"
},
{
  "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" : {
    ...
  }
}
}],
"ok" : 1
}
```

的有效值如下所示："desc"字段：

- **INTERNAL**— 内部系统任务，如游标清理或过时用户清理任务。
- **TTLMonitor**— 生存时间 (TTL) 监控线程。其运行状态在 "active" 字段中反映。
- **GARBAGE_COLLECTION**— 内部垃圾收集器线程。系统中最多可以同时运行三个垃圾收集器线程。
- **CONN**— 用户查询。
- **CURSOR**— 该操作是一个空闲游标，等待用户调用 "getMore" 命令以获得下一批结果。在这种状态下，游标正在消耗内存，但不消耗任何计算。

上述输出还列出了在系统中运行的所有用户查询。每个用户查询都在数据库和集合的上下文中运行，而这两者的并集称为命名空间。每个用户查询的命名空间都可在 "ns" 字段中获得。

有时您需要列出在特定命名空间中运行的所有用户查询。所以之前的输出必须在 "ns" 字段中返回的子位置类型。下面是一个实现要筛选的输出的示例查询。此查询列出当前在数据库中运行的所有用户查询。"db" 和收集 "test" (也就是说，"db.test" 命名空间)。

```
db.adminCommand({aggregate: 1,
  pipeline: [{ $currentOp: {allUsers: true, idleConnections: true}},
    { $match: {ns: { $eq: "db.test" }}}],
  cursor: {}
});
```

作为系统主用户，您可以查看所有用户的查询以及所有内部系统任务。所有其他用户只能查看其各自的查询。

如果查询和内部系统任务的总数超过默认批处理游标大小，则 mongoshell 会自动生成一个迭代器对象 'it' 以查看其余结果。继续执行 'it' 命令，直到所有结果都用完。

如何知道查询何时进行？

由于查询计划不够理想而运行缓慢，或者由于资源争用而受阻。调试此类查询是一个多步骤过程，可能需要多次执行相同的步骤。

调试的第一步是列出长时间运行或受阻的所有查询。以下查询列出了运行时间超过 10 秒或正在等待资源的所有用户查询。

```
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: {}
});
```

定期重复上述查询以确定查询列表是否更改并确定长时间运行或受阻的查询。

如果感兴趣的查询的输出文档有WaitState字段中，表示资源争用是查询运行缓慢或受阻的原因。资源争用可能由 I/O、内部系统任务或其他用户查询导致。

此操作的输出将类似于下文（JSON 格式）。

```
{
  "waitedMS" : NumberLong(0),
  "cursor" : {
    "firstBatch" : [
      {
        "opid" : 201,
        "command" : {
          "aggregate" : ...
        },
        "secs_running" : 208,
        "WaitState" : "IO"
      }
    ],
    "id" : NumberLong(0),
    "ns" : "admin.$cmd"
  },
  "ok" : 1
}
```

如果跨不同集合的很多查询在同一实例上并行运行，或者实例大小对于运行查询的数据集而言过小，则 I/O 可能是瓶颈所在。如果查询为只读查询，则可以通过将每个集合的查询分隔在不同副本中，以缓解前一种情况。对于跨不同集合的并发更新，或者当实例大小对数据集而言过小时，您可以通过扩展实例缓解措施。

如果资源争用是由于其他用户查询引起的，则"blockedOn"输出文档中的字段将包含"opid"影响此查询的查询。使用"opid"遵循的链"WaitState"和"blockedOn"所有查询的字段以查找位于链头的查询。

如果链头的任务是一个内部任务，则在这种情况下的唯一缓解措施将是终止查询并在稍后重新运行它。

下面是示例输出，其中的查找查询在由另一个任务拥有的集合锁上受阻。

```
{
  "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
}
```

如果"WaitState"有价值"Latch"、"SystemLock"、"BufferLock"、"BackgroundActivity"，或者"Other"，资源争用的根源是内部系统任务。如果这种情况持续了很长时间，则唯一的缓解措施将是终止查询并在稍后重新运行它。

我该如何确定为什么系统突然运行缓慢？

以下是系统速度缓慢的一些常见原因：

- 并发查询之间的资源争用过多
- 随着时间的推移，活动的并发查询的数量
- 内部系统任务，例如"GARBAGE_COLLECTION"

要监控系统的长期使用情况，请定期运行以下 "currentOp" 查询并将结果输出到外部存储。此查询会计算系统中每个命名空间中的查询和操作数。然后，您可以分析系统使用情况的结果以了解系统上的负载并采取适当措施。

```
db.adminCommand({aggregate: 1,
                    pipeline: [{ $currentOp: {allUsers: true, idleConnections: true}},
                               { $group: { _id: { desc: "$desc", ns: "$ns", WaitState:
"$WaitState"}, count: { $sum: 1 } } } ]},
                    cursor: {}
                });
```

此查询将返回在每个命名空间中运行的所有查询、所有内部系统任务和每个命名空间的等待状态（如果有）的数量的聚合。

此操作的输出将类似于下文（JSON 格式）。

```
{
  "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
}
```

在前面的输出中，命名空间中的两个用户查询"db.test"在集合锁上被阻止：命名空间中的一个查询"admin.\$cmd"，还有一个内部"TTLMonitor"任务。

如果输出表明有许多查询处于受阻等待状态，请参阅[如何查找并终止长时间运行或受阻的查询？](#) (p. 519)

如何确定一个或多个集群实例上 CPU 使用率高的原因？

以下部分可能会帮助您确定实例 CPU 使用率高的原因。您的结果可能因工作负载而异。

- 要确定实例突然运行缓慢的原因，请参阅[我该如何确定为什么系统突然运行缓慢？](#) (p. 524)
- 要确定并终止特定实例上长时间运行的查询，请参阅[如何查找并终止长时间运行或受阻的查询？](#) (p. 519)
- 要了解查询是否取得进展，请参阅[如何知道查询何时进行？](#) (p. 523)
- 要确定查询长时间运行的原因，请参阅[如何查看查询计划和优化查询？](#) (p. 520)
- 要跟踪长时间运行的查询，请参阅[分析 Amazon DocumentDB 操作](#) (p. 388)。

根据实例 CPU 使用率较高的原因，执行以下一项或多项操作可能会有所帮助。

- 如果主实例具有较高的 CPU 利用率，但副本实例没有，请考虑通过客户端读取首选项设置在副本之间分配读取流量（例如 `secondaryPreferred`）。有关更多信息，请参阅[作为副本集连接到 Amazon DocumentDB \(p. 454\)](#)。

通过允许主实例处理更多写入流量，使用副本进行读取可以更好地利用集群的资源。副本的读取具有最终一致性。

- 如果 CPU 使用率高是由于写入工作负载造成的，则将集群实例的大小更改为更大的实例类型会增加可用于为工作负载提供服务的 CPU 内核数。有关更多信息，请参阅[实例 \(p. 2\)](#) 和 [实例类规格 \(p. 286\)](#)。
- 如果所有群集实例都表现出较高的 CPU 利用率，并且工作负载使用副本进行读取，则向群集添加更多副本将增加可用于读取流量的资源。有关更多信息，请参阅[将 Amazon DocumentDB 实例添加到集群 \(p. 288\)](#)。

如何确定实例上的打开游标？

当连接到 Amazon DocumentDB 实例后，您可以使用命令 `db.runCommand("listCursors")` 以列出该实例上的打开光标。根据实例类型，在给定的 Amazon DocumentDB 实例上，任何给定时间最多可打开 4,560 个活动游标。通常建议关闭不再使用的游标，因为游标占用实例上的资源并具有上限。请参阅[Amazon DocumentDB 配额和限制 \(p. 496\)](#) 对于特定限制。

```
db.runCommand("listCursors")
```

如何确定当前的 Amazon DocumentDB 引擎版本？

要确定当前 Amazon DocumentDB 引擎版本，请运行以下命令。

```
db.runCommand({getEngineVersion: 1})
```

此操作的输出将类似于下文（JSON 格式）。

```
{ "engineVersion" : "2.x.x", "ok" : 1 }
```

Note

Amazon DocumentDB 3.6 的引擎版本为 1.x.x，Amazon DocumentDB 4.0 的引擎版本为 2.x.x。

如何识别未使用的索引？

最佳做法是定期识别和删除未使用的索引，以提高性能和降低成本，因为它消除了用于维护索引的不必要的计算、存储和 I/O。要确定给定集合的索引，请运行以下命令：

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

要确定是否已使用索引，请运行以下命令。命令的输出将描述以下内容：

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

- **ops**— 使用索引的操作数。如果您的工作负载运行了足够长的时间并且您确信工作负载处于稳定状态，那么 ops 值为零表示根本没有使用该索引。
- **since**— 自 Amazon DocumentDB 开始收集索引使用情况统计信息以来的时间，这通常是自上次数据库重启或维护操作以来的值。

要确定集合的总索引大小，请运行以下命令：

```
db.collection.stats()
```

要删除未使用的索引，请运行以下命令：

```
db.collection.dropIndex("indexName")
```

如何识别缺失的索引？

您可以使用[用于记录慢查询的 Amazon DocumentDB 分析器](#)。在慢查询日志中反复出现的查询可能表示需要额外的索引才能提高该查询的性能。

您可以通过查找具有至少执行一个或多个阶段的长时间运行查询来识别有用索引的机会COLLSCAN阶段，这意味着他们查询阶段必须读取集合中的每个文档才能对查询作出响应。

以下示例显示了对在大型集合上运行的出租车集合的查询。

```
db.rides.count({"fare.totalAmount":{$gt:10.0}})
```

为了执行此示例，查询必须执行集合扫描（即读取集合中的每个文档），因为fare.totalAmount字段中返回的子位置类型。此查询的 Amazon DocumentDB 分析器的输出将类似于以下内容：

```
{
  ...
  "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"
}
```

要加快本示例中的查询速度，您需要在其中创建索引。fare.totalAmount，如下所示。

```
db.rides.createIndex( {"fare.totalAmount": 1}, {background: true} )
```

Note

在前台创建的索引（意味着如果{background:true}创建索引时未提供选项）采用独占写锁定，这会阻止应用程序在索引构建完成之前向集合写入数据。在生产集群上创建索引时，请注意这种潜在影响。创建索引时，我们建议设置{background:true}。

一般来说，您希望在基数较高的字段上创建索引（例如，大量唯一值）。在基数较低的字段上创建索引可能会导致未使用的大索引。在创建查询计划时，Amazon DocumentDB 查询优化器会考虑收集的总体大小和索引的选择性。有时候你会看到查询处理器选择COLLSCAN即使在索引存在时也是如此。当查询处理器估计利用索引不会比扫描整个集合产生性能优势时，就会发生这种情况。如果要强制查询处理器利用特定索引，则可以使用hint()运算符，如下所示。


```
db.collection.find().hint("indexName")
```

有用查询的摘要

以下查询对于监控 Amazon DocumentDB 中的性能和资源利用率非常有用。

- 使用以下查询列出所有活动。

```
db.adminCommand({currentOp: 1, $all: 1});
```

- 以下代码列出了所有长时间运行或受阻的查询。

```
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: {}
});
```

- 以下代码终止查询。

```
db.adminCommand({killOp: 1, op: <opid of running or blocked query>});
```

- 使用以下代码可获取系统状态的聚合视图。

```
db.adminCommand({aggregate: 1,
  pipeline: [{currentOp: {allUsers: true, idleConnections: true}},
    {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
      "$WaitState"}, count: {$sum: 1}}}],
  cursor: {}
});
```

Amazon DocumentDB 集群、实例和资源管理 API 参考

本节介绍可通过 HTTP 访问的 Amazon DocumentDB (具有 MongoDB 兼容性) 的集群、实例和资源管理操作，Amazon Command Line Interface(Amazon CLI)，或者 Amazon 开发工具包 您可以使用这些 API 来创建、删除和修改集群和实例。

Important

这些 API 仅用于管理集群、实例和相关资源。有关如何连接到正在运行的 Amazon DocumentDB 集群的信息，请参阅 [入门指南 \(p. 28\)](#)。

主题

- [操作 \(p. 529\)](#)
- [数据类型 \(p. 658\)](#)
- [常见错误 \(p. 710\)](#)
- [常见参数 \(p. 712\)](#)

操作

支持以下操作：

- [AddSourceIdentifierToSubscription \(p. 531\)](#)
- [AddTagsToResource \(p. 533\)](#)
- [ApplyPendingMaintenanceAction \(p. 535\)](#)
- [CopyDBClusterParameterGroup \(p. 537\)](#)
- [CopyDBClusterSnapshot \(p. 539\)](#)
- [CreateDBCluster \(p. 542\)](#)
- [CreateDBClusterParameterGroup \(p. 548\)](#)
- [CreateDBClusterSnapshot \(p. 550\)](#)
- [CreateDBInstance \(p. 552\)](#)
- [CreateDBSubnetGroup \(p. 557\)](#)
- [CreateEventSubscription \(p. 559\)](#)
- [CreateGlobalCluster \(p. 562\)](#)
- [DeleteDBCluster \(p. 565\)](#)
- [DeleteDBClusterParameterGroup \(p. 567\)](#)
- [DeleteDBClusterSnapshot \(p. 568\)](#)
- [DeleteDBInstance \(p. 570\)](#)
- [DeleteDBSubnetGroup \(p. 572\)](#)
- [DeleteEventSubscription \(p. 574\)](#)
- [DeleteGlobalCluster \(p. 576\)](#)
- [DescribeCertificates \(p. 578\)](#)
- [DescribeDBClusterParameterGroups \(p. 580\)](#)

- [DescribeDBClusterParameters](#) (p. 582)
- [DescribeDBClusters](#) (p. 584)
- [DescribeDBClusterSnapshotAttributes](#) (p. 586)
- [DescribeDBClusterSnapshots](#) (p. 588)
- [DescribeDBEngineVersions](#) (p. 591)
- [DescribeDBInstances](#) (p. 594)
- [DescribeDBSubnetGroups](#) (p. 596)
- [DescribeEngineDefaultClusterParameters](#) (p. 598)
- [DescribeEventCategories](#) (p. 600)
- [DescribeEvents](#) (p. 601)
- [DescribeEventSubscriptions](#) (p. 604)
- [DescribeGlobalClusters](#) (p. 606)
- [DescribeOrderableDBInstanceOptions](#) (p. 608)
- [DescribePendingMaintenanceActions](#) (p. 610)
- [FailoverDBCluster](#) (p. 612)
- [ListTagsForResource](#) (p. 614)
- [ModifyDBCluster](#) (p. 616)
- [ModifyDBClusterParameterGroup](#) (p. 621)
- [ModifyDBClusterSnapshotAttribute](#) (p. 623)
- [ModifyDBInstance](#) (p. 625)
- [ModifyDBSubnetGroup](#) (p. 629)
- [ModifyEventSubscription](#) (p. 631)
- [ModifyGlobalCluster](#) (p. 633)
- [RebootDBInstance](#) (p. 635)
- [RemoveFromGlobalCluster](#) (p. 637)
- [RemoveSourceIdentifierFromSubscription](#) (p. 639)
- [RemoveTagsForResource](#) (p. 641)
- [ResetDBClusterParameterGroup](#) (p. 643)
- [RestoreDBClusterFromSnapshot](#) (p. 645)
- [RestoreDBClusterToPointInTime](#) (p. 650)
- [StartDBCluster](#) (p. 655)
- [StopDBCluster](#) (p. 657)

AddSourceIdentifierToSubscription

将源标识符添加到现有事件通知订阅。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

SourceIdentifier

要添加的事件源的标识符：

- 如果源类型是实例，则DBInstanceIdentifier必须提供。
- 如果源类型是安全组，则DBSecurityGroupName必须提供。
- 如果源类型是参数组，则DBParameterGroupName必须提供。
- 如果源类型是快照，则DBSnapshotIdentifier必须提供。

类型: 字符串

必需 是

SubscriptionName

要将源标识符添加到的 Amazon DocumentDB 事件通知订阅的名称。

类型: 字符串

必需 是

响应元素

服务返回以下元素。

EventSubscription

有关您订阅的活动的详细信息。

类型：[EventSubscription \(p. 690\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

SourceNotFound

找不到请求的源。

HTTP 状态代码：404

SubscriptionNotFound

订阅名称不存在。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

AddTagsToResource

向 Amazon DocumentDB 资源中添加元数据标签。您可以在成本分配报告中使用这些标签来跟踪与 Amazon DocumentDB 资源关联的成本或Condition在语句中Amazon Identity and Access ManagementAmazon DocumentDB 的 (IAM) 政策。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

ResourceName

向其中添加标签的 Amazon DocumentDB 资源。此值是 Amazon 资源名称。

类型: 字符串

必填项：是

标签 s.Tag.N

要分配给 Amazon DocumentDB 资源的标签。

类型: 数组Tag (p. 708)对象

必填项：是

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码：404

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码：404

DBSnapshotNotFound

DBSnapshotIdentifier不引用现有快照。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonV2 SDK for Java](#)
- [适用于 JavaScript 的Amazon开发工具包](#)

- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ApplyPendingMaintenanceAction

将待处理的维护操作应用于资源（例如，应用于 Amazon DocumentDB 实例）。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

ApplyAction

应用于此资源的待处理的维护操作。

有效值：system-update，db-upgrade

类型: 字符串

: 必需 是

OptInType

用于指定加入请求类型或撤消加入请求的值。不能撤消 immediate 类型的加入请求。

有效值：

- immediate - 立即应用维护操作。
- next-maintenance - 在资源的下一个维护时段内应用维护操作。
- undo-opt-in - 取消任何现有的 next-maintenance 加入请求。

类型: 字符串

: 必需 是

ResourceIdentifier

待处理的维护操作应用于的资源的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 是

响应元素

服务返回的以下元素。

ResourcePendingMaintenanceActions

表示的输出[ApplyPendingMaintenanceAction \(p. 535\)](#)。

类型：[ResourcePendingMaintenanceActions \(p. 706\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

ResourceNotFoundFault

找不到指定的资源 ID。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CopyDBClusterParameterGroup

复制指定的集群参数组。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

SourceDBClusterParameterGroupIdentifier

源集群参数组的标识符或 Amazon 资源名称 (ARN)。

约束：

- 必须指定有效的集群参数组。
- 如果源集群参数组位于同一个 Amazon Web Services 区域作为副本，请指定有效的参数组标识符；例如，`my-db-cluster-param-group`，或有效的 ARN。
- 如果源参数组位于不同 Amazon Web Services 区域比副本指定有效的集群参数组 ARN；例如，`arn:aws:rds:us-east-1:123456789012:sample-cluster:sample-parameter-group`。

类型: 字符串

: 必需 是

TargetDBClusterParameterGroupDescription

复制的集群参数组的描述。

类型: 字符串

: 必需 是

TargetDBClusterParameterGroupIdentifier

复制的集群参数组的标识符。

约束：

- 不能为 null 或空。
- 必须包含 1 到 255 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：`my-cluster-param-group1`

类型: 字符串

: 必需 是

标签 s.Tag.N

要分配给参数组的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

: 必需 否

响应元素

服务返回以下元素。

DBClusterParameterGroup

有关集群参数组的详细信息。

类型：[DBClusterParameterGroup](#) (p. 669) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误](#) (p. 710)。

DBParameterGroupAlreadyExists

已存在具有相同名称的参数组。

HTTP 状态代码：400

DBParameterGroupNotFound

DBParameterGroupName不引用现有参数组。

HTTP 状态代码：404

DBParameterGroupQuotaExceeded

此请求将导致您超出允许的参数组数目。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CopyDBClusterSnapshot

复制集群的快照。

要从共享的手动集群快照复制集群快照，`SourceDBClusterSnapshotIdentifier`必须是共享集群快照的 Amazon 资源名称 (ARN)。无论加密与否，都只能在相同时复制共享的数据库集群快照。Amazon Web Services 区域。

要在正在复制时取消操作，请删除由标识的目标集群快照。`TargetDBClusterSnapshotIdentifier`当该集群快照位于复制状态。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

`SourceDBClusterSnapshotIdentifier`

要复制的集群快照的标识符。此参数不区分大小写。

约束：

- 必须将有效的系统快照指定为可用 状态。
- 如果源快照在相同Amazon Web Services 区域作为副本，请指定有效的快照标识符。
- 如果源快照位于不同Amazon Web Services 区域指定有效的集群快照 ARN。

示例：`my-cluster-snapshot1`

类型: 字符串

必需 是

`TargetDBClusterSnapshotIdentifier`

要从源集群快照创建的新集群快照标识符。此参数不区分大小写。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：`my-cluster-snapshot2`

类型: 字符串

必需 是

`CopyTags`

设置为`true`将源集群快照的所有标签复制到目标集群快照，否则`false`。默认为 `false`。

类型: Boolean

必需 否

`KmsKeyId`

这些区域有：Amazon KMS加密集群快照的密钥 ID。Amazon KMS 密钥 ID 是 Amazon 资源名称 (ARN)、Amazon KMS 密钥标识符或 Amazon KMS 加密密钥的 Amazon KMS 密钥别名。

如果你从你的Amazon Web Services 账户，您可以为指定一个值`KmsKeyId`用新的加密副本Amazon KMS加密密钥。如果您没有为指定值`KmsKeyId`，则使用相同的加密集群快照副本。Amazon KMSkey 作为源集群快照。

如果您复制从另一个共享的加密集群快照。Amazon Web Services 账户，则必须为指定一个值。KmsKeyId.

将加密集群快照复制到另一个Amazon Web Services 区域，设置KmsKeyId到Amazon KMS要用于对目标区域中集群快照副本进行加密的密钥 ID。Amazon KMS加密密钥特定于Amazon Web Services 区域它们是在其中创建的，您无法使用一个加密密钥。Amazon Web Services 区域在另一个Amazon Web Services 区域。

如果您复制未加密的集群快照并为KmsKeyId参数时，将返回一个错误。

类型: 字符串

必需 否

PreSignedUrl

包含签名版本 4 签名请求的 URLCopyDBClusterSnapshot中的 API 操作Amazon Web Services 区域其中包含要复制的源集群快照。您必须使用PreSignedUrl从另一个群集快照复制时的参数Amazon Web Services 区域。

如果您正在使用AmazonSDK 工具或Amazon CLI，您可以指定SourceRegion (或者--source-region(对于)Amazon CLI) 而不是指定PreSignedUrl手动。指定SourceRegion自动生成一个预签名 URL，它是可在源中执行的操作的有效请求。Amazon Web Services 区域。

预签名 URL 必须是对CopyDBClusterSnapshot可以在源代码中执行的 API 操作Amazon Web Services 区域其中包含要复制的集群快照。预签名 URL 请求必须包含以下参数值：

- SourceRegion-包含要复制的快照的区域的 ID。
- SourceDBClusterSnapshotIdentifier-要复制的加密集群快照的标识符。源标识符必须采用 Amazon 资源名称 (ARN) 格式。Amazon Web Services 区域。例如，如果您从 us-east-1 复制加密集群快照Amazon Web Services 区域，然后您的SourceDBClusterSnapshotIdentifier看起来像以下内容：arn:aws:rds:us-east-1:12345678012:sample-cluster:sample-cluster-snapshot.
- TargetDBClusterSnapshotIdentifier-要创建的新集群快照的标识符。该参数不区分大小写。

类型: 字符串

必需 否

标签 s.Tag.N

要分配给集群快照的标签。

类型: 数组[Tag \(p. 708\)](#)对象

必需 否

响应元素

服务返回以下元素。

DBClusterSnapshot

有关集群快照的详细信息。

类型：[DBClusterSnapshot \(p. 671\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterSnapshotAlreadyExistsFault

您已拥有具有给定标识符的集群快照。

HTTP 状态代码：400

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不引用现有集群快照。

HTTP 状态代码：404

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

KMSKeyNotAccessibleFault

访问Amazon KMS键。

HTTP 状态代码：400

SnapshotQuotaExceeded

请求将导致超出允许的快照数目。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CreateDBCluster

创建新的 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

集群标识符。该参数作为一个小写字符串存储。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：`my-cluster`

类型：字符串

：必需 是

Engine

要用于此集群的数据库引擎的名称。

有效值：`docdb`

类型：字符串

：必需 是

可用性区域。可用性区域。N

可在其中创建集群中的实例的 Amazon EC2 可用区的列表。

类型：字符串数组

：必需 否

BackupRetentionPeriod

自动备份的保留天数。您必须指定最小值 1。

默认值：1

约束：

- 必须为介于 1 和 35 之间的值。

类型：整数

：必需 否

DBClusterParameterGroupName

要与此集群关联的集群参数组的名称。

类型：字符串

：必需 否

DBSubnetGroupName

要与此集群关联的子网组。

约束：必须与现有 DBSubnetGroup 的名称匹配。不能是默认值。

示例：mySubnetgroup

类型：字符串

：必需 否

DeletionProtection

指定是否可以删除此集群。如果 DeletionProtection 如果启用，则无法删除集群，除非它被修改并且 DeletionProtection 已禁用。DeletionProtection 防止意外删除集群。

类型：Boolean

：必需 否

启用 CloudWatch LogSports.Member。

需要启用以导出到 Amazon CloudWatch Logs 的日志类型的列表。您可以启用审核日志或分析器日志。有关更多信息，请参阅 [审核 Amazon DocumentDB 事件](#) 和 [分析 Amazon DocumentDB 操作](#)。

类型：字符串数组

：必需 否

EngineVersion

要使用的数据库引擎的版本号。这些区域有：--engine-version 将默认为最新的主引擎版本。对于生产工作负载，我们建议使用预期的主引擎版本明确声明此参数。

类型：字符串

：必需 否

GlobalClusterIdentifier

新全局集群的集群标识符。

类型：字符串

长度约束：最小长度为 1。长度上限为 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

：必需 否

KmsKeyId

这些区域有：Amazon KMS 已加密集群的密钥标识符。

这些区域有：Amazon KMS 密钥标识符是用于 Amazon KMS 加密密钥。如果你使用相同的创建集群 Amazon Web Services 账户那拥有 Amazon KMS 用来加密新集群的加密密钥，可以使用 Amazon KMS 的密钥别名而不是 ARN Amazon KMS 加密密钥。

未在 KmsKeyId 中指定加密密钥时：

- 如果 StorageEncrypted 参数为 true，则 Amazon DocumentDB 将使用您的默认加密密钥。

Amazon KMS 创建默认加密密钥 Amazon Web Services 账户。您的 Amazon Web Services 账户每个密钥有一个不同的默认加密密钥 Amazon Web Services 区域。

类型: 字符串

: 必需 否

MasterUsername

集群的主用户名称。

约束 :

- 必须为 1 到 63 个字母或数字。
- 第一个字符必须是字母。
- 不能是所选数据库引擎的保留字。

类型: 字符串

: 必需 否

MasterUserPassword

主数据库用户的密码。此密码可以包含除正斜杠 (/)、双引号 (") 或 @ 符号之外的任何可打印的 ASCII 字符。

约束 : 必须包含 8 到 100 个字符。

类型: 字符串

: 必需 否

Port

群集中的实例接受连接的端口号。

类型: 整数

: 必需 否

PreferredBackupWindow

使用 BackupRetentionPeriod 参数启用了自动备份时, 自动执行备份的日常时间范围。

默认值是从每个时间段中随机选择的 30 分钟时间。Amazon Web Services 区域。

约束 :

- 必须采用 hh24:mi-hh24:mi 格式。
- 必须采用通用协调时间 (UTC)。
- 不得与首选维护时段冲突。
- 必须至少为 30 分钟。

类型: 字符串

: 必需 否

PreferredMaintenanceWindow

可进行系统维护的每周时间范围 (采用通用协调时间 (UTC)) 。

格式 : ddd:hh24:mi-ddd:hh24:mi

默认值是从每个时间段中随机选择的 30 分钟时间。Amazon Web Services 区域, 可以是一周中的任意一天。

有效值 : Mon、Ted、Wed、Tu、Fri、Sat、Sun

约束：窗口至少为 30 分钟。

类型: 字符串

: 必需 否

PreSignedUrl

当前不支持。

类型: 字符串

: 必需 否

StorageEncrypted

指定集群是否已加密。

类型: Boolean

: 必需 否

标签 s.Tag.N

要分配给集群的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

: 必需 否

vpc 安全组 ID.vpc 安全组 ID.N

要与此集群关联的 EC2 VPC 安全组的列表。

类型: 字符串数组

: 必需 否

响应元素

服务返回的以下元素。

DBCluster

有关集群的详细信息。

类型 : [DBCluster \(p. 663\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterAlreadyExistsFault

您已拥有具有给定标识符的集群。

HTTP 状态代码：400

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码：404

DBClusterParameterGroupNotFound

DBClusterParameterGroupName不引用现有集群参数组。

HTTP 状态代码：404

DBClusterQuotaExceededFault

无法创建集群，因为您已达到允许的最大集群配额。

HTTP 状态代码：403

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码：404

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一个可用区，否则子网组中的子网应至少包含两个可用区。

HTTP 状态代码：400

DBSubnetGroupNotFoundFault

DBSubnetGroupName不引用现有子网组。

HTTP 状态代码：404

GlobalClusterNotFoundFault

这些区域有：GlobalClusterIdentifier不引用现有全局集群。

HTTP 状态代码：404

InsufficientStorageClusterCapacity

当前操作没有足够的可用存储空间。通过更新子网组来使用具有更多可用存储空间的不同可用区，可以解决此错误。

HTTP 状态代码：400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

InvalidDBSubnetGroupStateFault

无法删除子网组，因为它正在使用中。

HTTP 状态代码：400

InvalidGlobalClusterStateFault

当群集处于此状态时，无法执行请求的操作。

HTTP 状态代码：400

InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于常见虚拟私有云 (VPC) 中。

HTTP 状态代码：400

InvalidVPCNetworkStateFault

由于所做的更改，子网组在创建后并不包含所有可用区。

HTTP 状态代码：400

KMSKeyNotAccessibleFault

访问时出错Amazon KMS键。

HTTP 状态代码：400

StorageQuotaExceeded

此请求将导致您超出所有实例中允许的可用存储量。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CreateDBClusterParameterGroup

创建新的集群参数组。

集群参数组中的参数应用到集群中的所有实例。

最初创建集群参数组时，使用的是集群中实例所用数据库引擎的默认参数。在 Amazon DocumentDB 中，您不能直接修改 default.docdb3.6 集群参数组。如果您的 Amazon DocumentDB 集群正在使用默认群集参数组并且您想修改其中的值，则必须首先 [创建新的参数组](#) 要么 [复制现有参数组](#)，修改它，然后将修改后的参数组应用于您的集群。要使新数据库集群参数组和关联设置生效，您必须重新启动集群中的实例，而不进行故障转移。有关更多信息，请参阅 [修改 Amazon DocumentDB 集群参数组](#)。

请求参数

有关所有操作通用的参数的信息，请参阅 [常见参数 \(p. 712\)](#)。

DBClusterParameterGroupName

集群参数组的名称。

约束：

- 不能与现有 DBClusterParameterGroup 的名称匹配。

Note

此值以一个小写字符串存储。

类型: 字符串

: 必需 是

DBParameterGroupFamily

集群参数组系列名称。

类型: 字符串

: 必需 是

Description

集群参数组的描述。

类型: 字符串

: 必需 是

标签 s.Tag.N

要分配给集群参数组的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

: 必需 否

响应元素

服务返回以下元素。

DBClusterParameterGroup

有关集群参数组的详细信息。

类型：[DBClusterParameterGroup](#) (p. 669) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误](#) (p. 710)。

DBParameterGroupAlreadyExists

已存在具有相同名称的参数组。

HTTP 状态代码：400

DBParameterGroupQuotaExceeded

此请求将导致您超出允许的参数组数目。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CreateDBClusterSnapshot

创建集群的快照。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

要为其创建快照的集群的标识符。此参数不区分大小写。

约束：

- 必须匹配现有标识符DBCluster。

示例：my-cluster

类型：字符串

必填项：是

DBClusterSnapshotIdentifier

集群快照的标识符。该参数作为一个小写字符串存储。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：my-cluster-snapshot1

类型：字符串

必填项：是

标签 s.Tag.N

要分配给集群快照的标签。

类型：数组[Tag \(p. 708\)](#)对象

必填项：否

响应元素

服务返回以下元素。

DBClusterSnapshot

有关集群快照的详细信息。

类型：[DBClusterSnapshot \(p. 671\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不引用现有集群。

HTTP 状态代码：404

DBClusterSnapshotAlreadyExistsFault

您已拥有具有给定标识符的集群快照。

HTTP 状态代码：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

SnapshotQuotaExceeded

请求将导致您超出允许的快照数目。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CreateDBInstance

创建新实例。

请求参数

有关所有操作的通用参数的信息，请参见[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

实例所属的集群的标识符。

类型: 字符串

必填 是

DBInstanceClass

实例的计算和内存容量；例如，`db.r5.large`。

类型: 字符串

必填 是

DBInstanceIdentifier

实例标识符。该参数作为一个小写字符串存储。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：`mydbinstance`

类型: 字符串

必填 是

Engine

要用于此实例的数据库引擎的名称。

有效值：`docdb`

类型: 字符串

必填 是

AutoMinorVersionUpgrade

此参数不适用于 Amazon DocumentDB。无论设置的值如何，Amazon DocumentDB 都不会执行次要版本升级。

默认值：`false`

类型: 布尔值

必填 否

AvailabilityZone

在其中创建了实例的 Amazon EC2 可用区。

默认值：终端节点中系统随机选择的可用区Amazon Web Services 区域。

示例：us-east-1d

类型: 字符串

必填 否

CopyTagsToSnapshot

指示是否将标签从数据库实例复制到数据库实例快照的值。默认情况下，不复制标签。

类型: 布尔值

必填 否

EnablePerformanceInsights

指示是否为数据库实例启用 Performance Insights。有关更多信息，请参阅 [使用亚马逊Performance Insights](#)。

类型: 布尔值

必填 否

PerformanceInsightsKMSKeyId

用于加密 Performance Insights 数据的 Amazon KMS 密钥标识符。

这些区域有：Amazon KMS密钥标识符是密钥 ARN、密钥 ID、别名 ARN 或者 KMS 密钥的别名。

如果您未指定值 PerformanceInsights自杀KeyId然后 Amazon DocumentDB 将使用您的默认 KMS 密钥。您的Amazon Web Services 科技账户拥有默认。您的Amazon Web Services 账户在每个Amazon Web Services 云科技区域都有不同的默认 KMS 密钥。

类型: 字符串

必填 否

PreferredMaintenanceWindow

每周可以进行系统维护的时间范围（采用通用协调时间 (UTC)）。

格式：ddd:hh24:mi-ddd:hh24:mi

默认值为每个 Amazon Web Services 区域 8 小时的时间段中随机选择的 30 分钟时段（随机选取周中的某天进行）。

有效值：Mon、Fri、Fri、Fri、Fri、Fri、Th

约束：至少为 30 分钟的时段。

类型: 字符串

必填 否

PromotionTier

该值指定在现有主实例发生故障后将 Amazon DocumentDB 副本提升为主实例的顺序。

默认值：1

有效值：0-15

类型: 整数

必填 否

Tags.Tag.N

要分配给实例的标签。您可以为每个实例分配最多 10 个标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

必填 否

响应元素

以下元素由服务返回。

DBInstance

有关实例的详细信息。

类型 : [DBInstance \(p. 678\)](#) 对象

错误

有关所有操作的常见错误的信息，请参见 [常见错误 \(p. 710\)](#)。

AuthorizationNotFound

指定的 CIDR IP 或 Amazon EC2 安全组未获得指定安全组的授权。

Amazon DocumentDB B 也可能无权使用 IAM 代表您执行必要的操作。

HTTP 状态状态状态状态状态 : 404

DBClusterNotFoundFault

DBClusterIdentifier 不引用现有集需。

HTTP 状态状态状态状态状态 : 404

DBInstanceAlreadyExists

您已经有一个具有给定标识符的实例。

HTTP 状态状态状态状态状态 : 400

DBParameterGroupNotFound

DBParameterGroupName 不引用现有参数组。

HTTP 状态状态状态状态状态 : 404

DBSecurityGroupNotFound

DBSecurityGroupName 不引用现有安全组。

HTTP 状态状态状态状态状态 : 404

DBSubnetGroupDoesNotCoverEnoughAZs

子网组中的子网应至少覆盖两个可用区，除非只有一个可用区。

HTTP 状态状态状态状态状态 : 400

DBSubnetGroupNotFoundFault

DBSubnetGroupName 不引用现有子网组。

HTTP 状态状态状态状态状态 : 404
InstanceQuotaExceeded

该请求将导致您超过允许的实例数量。

HTTP 状态状态状态状态状态 : 400
InsufficientDBInstanceCapacity

指定的实例类在指定可用区中不可用。

HTTP 状态状态状态状态状态 : 400
InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态状态状态状态状态 : 400
InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于公共虚拟私有云 (VPC) 中。

HTTP 状态状态状态状态状态 : 400
InvalidVPCNetworkStateFault

由于进行了更改，子网组在创建后并未覆盖所有可用区。

HTTP 状态状态状态状态状态 : 400
KMSKeyNotAccessibleFault

访问时出错Amazon KMS钥匙。

HTTP 状态状态状态状态状态 : 400
StorageQuotaExceeded

该请求将导致您超过所有实例的允许可用存储量。

HTTP 状态状态状态状态状态 : 400
StorageTypeNotSupported

存储指定的StorageType无法与数据库实例相关需。

HTTP 状态状态状态状态状态 : 400

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon 命令行界面](#)
- [适用于 .NET 的 Amazon SDK](#)
- [适用于 C++ 的 Amazon SDK](#)
- [适用于 Go 的 Amazon SDK](#)
- [适用于 Java V2 的 Amazon SDK](#)
- [Amazon 开发工具包状态为 JavaScript](#)
- [适用于 PHP V3 的 Amazon SDK](#)
- [适用于 Python 的 Amazon SDK](#)
- [适用于 Ruby V3 的 Amazon SDK](#)

CreateDBSubnetGroup

创建新子网组。子网组必须至少包含给定的Amazon Web Services 区域。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBSubnetGroupDescription

子网组的描述。

类型: 字符串

必填项: 是

DBSubnetGroupName

子网组的名称。此值以一个小写字符串存储。

约束: 必须包含不超过 255 个字母、数字、句点、下划线、空格或连字符。不能是默认值。

示例: mySubnetgroup

类型: 字符串

必填项: 是

subnetids.subnetid 标识符。

子网组的 Amazon EC2 子网 ID。

类型: 字符串数组

必填项: 是

标签 s.Tag.N

要分配给子网组的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

必填项: 否

响应元素

服务返回以下元素。

DBSubnetGroup

有关子网组的详细信息。

类型: [DBSubnetGroup \(p. 683\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBSubnetGroupAlreadyExists

DBSubnetGroupName 已由现有子网组使用。

HTTP 状态代码：400

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一个可用区，否则子网组中的子网应至少包含两个可用区。

HTTP 状态代码：400

DBSubnetGroupQuotaExceeded

请求将导致您超出允许的子网组数目。

HTTP 状态代码：400

DBSubnetQuotaExceededFault

请求将导致您超出子网组中允许的子网数目。

HTTP 状态代码：400

InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于常见虚拟私有云 (VPC) 中。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CreateEventSubscription

创建 Amazon DocumentDB 事件通知订阅。此操作需要一个通过使用 Amazon DocumentDB 控制台、Amazon SNS 控制台或 Amazon SNS API 创建的主题的 Amazon 资源名称 (ARN)。要通过 Amazon SNS 获取 ARN，您必须在 Amazon SNS 中创建主题并订阅该主题。ARN 显示在 Amazon SNS 控制台中。

您可以指定源的类型 (`SourceType`) 您希望收通知的。您还可以提供 Amazon DocumentDB 来源列表 (`SourceIds`) 触发事件，您可以提供事件类别列表 (`EventCategories`) 对于您希望收通知的事件。例如，您可以指定 `SourceType = db-instance`、`SourceIds = mydbinstance1`，`mydbinstance2` 和 `EventCategories = Availability, Backup`。

如果您同时指定 `SourceType` 和 `SourceIds` (例如 `SourceType = db-instance` 和 `SourceIdentifier = myDBInstance1`)，您将收到所有 `db-instance` 指定源的事件。如果你指定 `SourceType` 但是不要指定 `SourceIdentifier`，则会收到您所有 Amazon DocumentDB 源的该源类型的事件通知。如果未指定 `SourceType` 或者 `SourceIdentifier`，系统会收到从属于您的客户账户的所有 Amazon DocumentDB 源中生成的事件通知。

请求参数

有关所有操作通用的参数的信息，请参阅 [常见参数 \(p. 712\)](#)。

SnsTopicArn

为事件通知创建的 SNS 主题的 Amazon 资源名称 (ARN)。在您创建主题并订阅到该主题时，Amazon SNS 会创建 ARN。

类型: 字符串

: 必需 是

SubscriptionName

订阅的名称。

约束：名称必须少于 255 个字符。

类型: 字符串

: 必需 是

Enabled

布尔值；设置为 `true` 要激活订阅，请设置为 `false` 来创建订阅但不激活该订阅。

类型: Boolean

: 必需 否

活动类别。EVENTCATEGORY.N

对于是一个事件类别列表 `SourceType` 您要订阅。

类型: 字符串数组

: 必需 否

sourceids.sourceid.Sourceidn.

为其返回事件的事件源的标识符列表。如果未指定，则响应中包含所有源。标识符必须以字母开头，并且只能包含 ASCII 字母、数字和连字符，不能以连字符结尾，也不能包含两个连续的连字符。

约束：

- 如果SourceIds已提供，SourceType还必须提供。
- 如果源类型是实例，则DBInstanceIdentifier必须提供。
- 如果源类型是安全组，则DBSecurityGroupName必须提供。
- 如果源类型是参数组，则DBParameterGroupName必须提供。
- 如果源类型是快照，则DBSnapshotIdentifier必须提供。

类型: 字符串数组

: 必需 否

SourceType

生成事件的源的类型。例如，如果希望收到关于实例生成的事件的通知，则应将此参数设置为。db-instance。如果未指定该值，则将返回所有事件。

有效值：db-instance，db-cluster，db-parameter-group，db-security-group，db-cluster-snapshot

类型: 字符串

: 必需 否

标签 s.Tag.N

要分配给事件订阅的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

: 必需 否

响应元素

服务返回以下元素。

EventSubscription

您订阅的事件的详细信息。

类型：[EventSubscription \(p. 690\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

EventSubscriptionQuotaExceeded

您已达到事件订阅的最大数量。

HTTP 状态代码：400

SNSInvalidTopic

Amazon SNS 回应说，指定主题存在问题。

HTTP 状态代码：400

SNSNoAuthorization

您没有向 SNS 发布的 Amazon 资源名称 (ARN) 的权限。

HTTP 状态代码：400

`SNSTopicArnNotFound`

SNS 主题的 Amazon 资源名称 (ARN) 不存在。

HTTP 状态代码：404

`SourceNotFound`

找不到请求的源。

HTTP 状态代码：404

`SubscriptionAlreadyExist`

提供的订阅名称已存在。

HTTP 状态代码：400

`SubscriptionCategoryNotFound`

提供的类别不存在。

HTTP 状态代码：404

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CreateGlobalCluster

创建一个可以跨多个多个的 Amazon DocumentDB 全局集群 Amazon Web Services 区域。全局群集包含一个具有读写功能的主群集和最新的只读辅助群集。全局群集使用基于存储的快速复制，延迟时间少于一秒钟的区域，使用专用基础架构，而不会影响工作负载的性能。

您可以创建最初为空的全局集群，然后向其中添加主集群和辅助集群。或者，您可以在创建操作期间指定现有集群，此集群将成为全局集群的主集群。

Note

此操作仅适用于 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

GlobalClusterIdentifier

新全局集群的集群标识符。

类型: 字符串

长度约束：最小长度为 1。长度上限为 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必需 是

DatabaseName

数据库的名称，最多 64 个字母和数字字符。如果您不提供名称，Amazon DocumentDB 就不会在您正在创建的全局集群中创建数据库。

类型: 字符串

必需 否

DeletionProtection

新全局集群的删除保护设置。在启用删除保护时，无法删除全局集群。

类型: Boolean

必需 否

Engine

要用于此集群的数据库引擎的名称。

类型: 字符串

必需 否

EngineVersion

全局集群的引擎版本。

类型: 字符串

必需 否

SourceDBClusterIdentifier

要用作全局集群的主集群的 Amazon 资源名称 (ARN)。此参数为可选项。

类型: 字符串

必需 否

StorageEncrypted

新全局集群的存储加密设置。

类型: Boolean

必需 否

响应元素

服务返回以下元素。

GlobalCluster

表示 Amazon DocumentDB 全局集群的数据类型。

类型 : [GlobalCluster \(p. 693\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不引用现有集群。

HTTP 状态代码 : 404

GlobalClusterAlreadyExistsFault

这些区域有 : `GlobalClusterIdentifier`已存在。选择新的全局集群标识符 (唯一名称) 以创建新的全局集群。

HTTP 状态代码 : 400

GlobalClusterQuotaExceededFault

此账户的全局集群数已达到允许的最大数量。

HTTP 状态代码 : 400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码 : 400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)

- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteDBCluster

删除之前预配置的集群。在您删除集群时，会删除该集群的所有自动备份，且无法恢复。不会删除指定集群的手动数据库集群快照。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

要删除的集群的集群标识符。该参数不区分大小写。

约束：

- 必须匹配现有DBClusterIdentifier。

类型: 字符串

：必填项：是

FinalDBSnapshotIdentifier

在时间创建的新集群快照的集群快照标识符SkipFinalSnapshot设置为false。

Note

指定此参数并且设置SkipFinalShapshot对于是参数true导致错误。

约束：

- 必须为 1 到 255 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

类型: 字符串

：必填项：否

SkipFinalSnapshot

确定删除集群之前是否创建最终集群快照。如果true，则不创建集群快照。如果false，则删除数据库集群之前创建集群快照。

Note

如果SkipFinalSnapshot是false，则必须指定FinalDBSnapshotIdentifier参数。

默认值: false

类型: Boolean

：必填项：否

响应元素

服务返回以下元素。

DBCluster

有关集群的详细信息。

类型：[DBCluster \(p. 663\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不引用现有集群。

HTTP 状态代码：404

DBClusterSnapshotAlreadyExistsFault

您已拥有具有给定标识符的集群快照。

HTTP 状态代码：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

SnapshotQuotaExceeded

请求将导致您超出允许的快照数目。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteDBClusterParameterGroup

删除指定的集群参数组。要添加的集群参数组不能与任何集群关联。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterParameterGroupName

集群参数组的名称。

约束：

- 必须是现有集群参数组的名称。
- 您无法删除默认的集群参数组。
- 不能与任何集群关联。

类型: 字符串

必需：是

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBParameterGroupNotFound

DBParameterGroupName不引用现有参数组。

HTTP 状态代码：404

InvalidDBParameterGroupState

参数组正在使用中，或者处于无效状态。如果您尝试删除参数组，则在参数组处于此状态时无法将其删除。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteDBClusterSnapshot

删除集群快照。如果正在复制快照，则复制操作将会终止。

Note

集群快照必须位于available待删除的状态。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterSnapshotIdentifier

要删除的集群快照的标识符。

约束：必须是中的现有集群快照的名称available状态。

类型: 字符串

必需 是

响应元素

服务返回以下元素。

DBClusterSnapshot

有关集群快照的详细信息。

类型： [DBClusterSnapshot \(p. 671\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不引用现有集群快照。

HTTP 状态代码：404

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)

- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteDBInstance

删除之前预配置的实例。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBInstanceIdentifier

要删除的实例的实例标识符。该参数不区分大小写。

约束：

- 必须与现有实例的名称匹配。

类型: 字符串

必需：是

响应元素

服务返回以下元素。

DBInstance

有关实例的详细信息。

类型：[DBInstance \(p. 678\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码：404

DBSnapshotAlreadyExists

DBSnapshotIdentifier现有快照已在使用。

HTTP 状态代码：400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

SnapshotQuotaExceeded

此请求将导致您超出允许的快照数目。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteDBSubnetGroup

删除子网组。

Note

指定的数据库子网组不得与任何数据库实例关联。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBSubnetGroupName

要删除的数据库子网组的名称。

Note

您不能删除默认子网组。

约束：

必须与现有 DBSubnetGroup 的名称匹配。不能是默认值。

示例：mySubnetgroup

类型：字符串

必需 是

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBSubnetGroupNotFoundFault

DBSubnetGroupName不引用现有子网组。

HTTP 状态代码：404

InvalidDBSubnetGroupStateFault

无法删除子网组，因为它正在使用中。

HTTP 状态代码：400

InvalidDBSubnetStateFault

子网不在可用状态。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)

- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteEventSubscription

删除 Amazon DocumentDB 事件通知订阅。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

SubscriptionName

您希望删除的 Amazon DocumentDB 事件通知订阅的名称。

类型: 字符串

必填项: 是

响应元素

服务返回以下元素。

EventSubscription

您订阅的事件的详细信息。

类型: [EventSubscription \(p. 690\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

InvalidEventSubscriptionState

其他人可能正在修改订阅。请等待几秒钟，然后重试。

HTTP 状态代码: 400

SubscriptionNotFound

订阅名称不存在。

HTTP 状态代码: 404

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)

- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DeleteGlobalCluster

删除全局集群。在尝试删除全局集群之前，必须已分离或删除主集群和辅助集群。

Note

此操作仅适用于 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

GlobalClusterIdentifier

被删除的全局集群的集群标识符。

类型: 字符串

长度约束：最小长度为 1。长度上限为 255。

模式：[A-Za-z][0-9A-Za-z-:._]*

必需：是

响应元素

服务返回以下元素。

GlobalCluster

表示 Amazon DocumentDB 全局集群的数据类型。

类型：[GlobalCluster \(p. 693\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

GlobalClusterNotFoundFault

这些区域有：GlobalClusterIdentifier不引用现有全局集群。

HTTP 状态代码：404

InvalidGlobalClusterStateFault

当集群处于此状态时，无法执行请求的操作。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeCertificates

返回 Amazon DocumentDB 为此提供的证书颁发机构 (CA) 证书列表Amazon Web Services 账户。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

CertificateIdentifier

用户提供的证书标识符。如果指定了此参数，则只返回指定证书的信息。如果省略此参数，则最多列表MaxRecords证书被退回。此参数不区分大小写。

约束

- 必须匹配现有CertificateIdentifier。

类型: 字符串

必填项: 否

过滤器。过滤器。

当前不支持此参数。

类型: 数组[Filter \(p. 692\)](#)对象

必填项: 否

Marker

由之前的 DescribeCertificates 请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 MaxRecords 指定的值。

类型: 字符串

必填项: 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录数超过了指定的MaxRecords 值，则在响应中包含称为标记的分页记号，以便检索剩余的结果。

默认值: 100

约束:

- 最低: 20
- 最大值: 100

类型: 整数

必填项: 否

响应元素

服务返回以下元素。

证书。证书。

此证书列表Amazon Web Services 账户。

类型: 数组[Certificate \(p. 660\)](#)对象

Marker

如果检索的记录数大于，则提供可选的分页令牌MaxRecords。如果指定了此参数，则标记指定列表中的下一条记录。包括的价值Marker在下一个电话中DescribeCertificates导致下一页证书。

类型: 字符串

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

CertificateNotFound

CertificateIdentifier不引用现有证书。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBClusterParameterGroups

返回 `DBClusterParameterGroup` 描述的列表。如果 `DBClusterParameterGroupName` 参数，则列表中只包含指定集群参数组的描述。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

`DBClusterParameterGroupName`

要返回其详细信息的特定集群参数组的名称。

约束：

- 如果提供，必须与现有的名称匹配 `DBClusterParameterGroup`。

类型: 字符串

：必需 否

过滤器。过滤器。

当前不支持此参数。

类型: 数组 [Filter \(p. 692\)](#) 对象

：必需 否

`Marker`

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

：必需 否

`MaxRecords`

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

：必需 否

响应元素

服务返回以下元素。

`dbCluster 参数组。dbcluster参数组.n`

集群参数组的列表。

类型: 数组 [DBClusterParameterGroup \(p. 669\)](#) 对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBParameterGroupNotFound

`DBParameterGroupName`不引用现有参数组。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBClusterParameters

返回特定集群参数组的详细参数列表。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterParameterGroupName

要返回其参数详细信息的特定集群参数组的名称。

约束：

- 如果提供，必须与现有的名称匹配DBClusterParameterGroup。

类型: 字符串

：必填项：是

过滤器。过滤器。

当前不支持此参数。

类型: 数组[Filter \(p. 692\)](#)对象

：必填项：否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为MaxRecords 指定的值。

类型: 字符串

：必填项：否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定MaxRecords值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

：必填项：否

Source

指示仅返回特定源的参数的值。参数源可以是 engine、service 或 customer。

类型: 字符串

：必填项：否

响应元素

服务返回以下元素。

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

参数。参数。

提供集群参数组的参数列表。

类型: 数组 [Parameter \(p. 698\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅 [常见错误 \(p. 710\)](#)。

DBParameterGroupNotFound

`DBParameterGroupName` 不引用现有参数组。

HTTP 状态代码：404

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBClusters

返回有关预配置的 Amazon DocumentDB 集群的信息。此 API 操作支持分页。对于某些管理功能（如集群和实例周期管理），Amazon DocumentDB 利用与 Amazon RDS 和 Amazon Neptune 共享的操作技术。使用 `filterName=engine,Values=docdb` 筛选参数以仅返回 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

用户提供的集群标识符。如果指定了此参数，则只返回特定集群的信息。该参数不区分大小写。

约束：

- 如果提供，则必须匹配现有 `DBClusterIdentifier`。

类型: 字符串

：必填项：否

过滤器。过滤器。

筛选条件指定要描述的一个或多个集群。

支持的筛选条件：

- `db-cluster-id` 接受集群标识符和集群 Amazon 资源名称 (ARN)。结果列表中仅包含由这些 ARN 确定的集群的相关信息。

类型: 数组 [Filter \(p. 692\)](#) 对象

：必填项：否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

：必填项：否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

：必填项：否

响应元素

服务返回以下元素。

dbclusters.dbCluster。

集群列表。

类型: 数组[DBCluster](#) (p. 663)对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误](#) (p. 710)。

DBClusterNotFoundFault

`DBClusterIdentifier`不引用现有集群。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBClusterSnapshotAttributes

返回手动数据库集群快照的集群快照属性名称和值的列表。

当你与其他人共享快照时Amazon Web Services 账户、DescribeDBClusterSnapshotAttributes返回restore属性和的 ID 列表Amazon Web Services 账户有权复制或还原手动群集快照。如果all包含在的值列表中restore属性，则手动集群快照为公有，可以由所有人复制或还原Amazon Web Services 账户。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterSnapshotIdentifier

要描述其属性的集群快照的标识符。

类型: 字符串

必需 是

响应元素

服务返回的以下元素。

DBClusterSnapshotAttributesResult

有关与集群快照关联的属性的详细信息。

类型：[DBClusterSnapshotAttributesResult \(p. 675\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不引用现有集群快照。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBClusterSnapshots

返回有关集群快照的信息。此 API 操作支持分页。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

要为其检索集群快照列表的集群的 ID。此参数不能与 `DBClusterSnapshotIdentifier` 参数。此参数不区分大小写。

约束：

- 如果提供，则必须与现有的标识符匹配 `DBCluster`。

类型: 字符串

：必需 否

DBClusterSnapshotIdentifier

要描述的特定集群快照标识符。此参数不能与 `DBClusterIdentifier` 参数。此值以一个小写字母字符串存储。

约束：

- 如果提供，则必须与现有的标识符匹配 `DBClusterSnapshot`。
- 如果此标识符用于自动快照，则还必须指定 `SnapshotType` 参数。

类型: 字符串

：必需 否

过滤器。过滤器。

当前不支持此参数。

类型: 数组 [Filter \(p. 692\)](#) 对象

：必需 否

IncludePublic

设置为 `true` 如果您提供，则包括可由所有人复制或还原的公有手动集群快照。Amazon Web Services 账户，否则为 `false`。默认为 `false`。

类型: Boolean

：必需 否

IncludeShared

设置为 `true` 包括来自其他人的共享手动集群快照 Amazon Web Services 账户那是这个 Amazon Web Services 账户已获得复制或恢复的权限，否则 `false`。默认为 `false`。

类型: Boolean

：必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定MaxRecords值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

: 必需 否

SnapshotType

要返回的集群快照的类型。可以指定以下值之一：

- automated-返回 Amazon DocumentDB 为您自动创建的所有集群快照Amazon Web Services 账户。
- manual-返回你为你手动创建的所有集群快照Amazon Web Services 账户。
- shared-返回已分享给您的所有手动集群快照Amazon Web Services 账户。
- public-返回已标记为公有的所有集群快照。

如果您没有指定SnapshotType值，则返回自动和手动集群快照。您可以通过在这些结果中包括共享集群快照，方法是IncludeShared对于是参数true。您可以通过在这些结果中包括公有集群快照，方法是IncludePublic对于是参数true。

IncludeShared 和 IncludePublic 参数不适用于 SnapshotType 的值 manual 或 automated。SnapshotType 设置为 shared 时，IncludePublic 参数不适用。SnapshotType 设置为 public 时，IncludeShared 参数不适用。

类型: 字符串

: 必需 否

响应元素

服务返回以下元素。

dbcluster erSnapshots.dbcluster erSnapshot.N

提供集群快照列表。

类型: 数组[DBClusterSnapshot \(p. 671\)](#)对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为MaxRecords 指定的值。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier未引用现有集群快照。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBEngineVersions

返回可用引擎的列表。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBParameterGroupFamily

要返回其详细信息的特定参数组系列的名称。

约束：

- 如果提供，则必须匹配现有DBParameterGroupFamily。

类型: 字符串

：必需 否

DefaultOnly

指示仅返回指定引擎的默认版本还是返回引擎与主要版本的组合。

类型: Boolean

：必需 否

Engine

要返回的数据库引擎。

类型: 字符串

：必需 否

EngineVersion

要返回的数据库引擎版本。

示例：3.6.0

类型: 字符串

：必需 否

过滤器。过滤器。

当前不支持此参数。

类型: 数组[Filter \(p. 692\)](#)对象

：必需 否

ListSupportedCharacterSets

如果指定了此参数且请求的引擎支持 CreateDBInstance 的 CharacterSetName 参数，则响应包括每个引擎版本支持的字符集列表。

类型: Boolean

：必需 否

ListSupportedTimezones

如果指定了此参数且请求的引擎支持 CreateDBInstance 的 TimeZone 参数，则响应包括每个引擎版本支持的时区列表。

类型: Boolean

: 必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

: 必需 否

响应元素

服务返回以下元素。

dbengine 版本。dbGineGine 版本。

有关一个或多个引擎版本的详细信息。

类型: 数组 [DBEngineVersion \(p. 676\)](#) 对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅 [常见错误 \(p. 710\)](#)。

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)

- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBInstances

返回有关预置的 Amazon DocumentDB 实例的信息。此 API 支持分页。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBInstanceIdentifier

用户提供的实例标识符。如果指定了此参数，则只返回特定实例的信息。该参数不区分大小写。

约束：

- 如果提供，必须匹配现有的标识符DBInstance。

类型: 字符串

：必需 否

过滤器。过滤器。

筛选条件指定要描述的一个或多个实例。

支持的筛选条件：

- db-cluster-id-接受集群标识符和集群 Amazon 资源名称 (ARN)。该结果列表仅包括与这些 ARN 所标识集群关联的实例的相关信息。
- db-instance-id-接受实例标识符和实例 ARN。结果列表仅包括由这些 ARN 标识的实例的相关信息。

类型: 数组[Filter \(p. 692\)](#)对象

：必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 MaxRecords 指定的值。

类型: 字符串

：必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定MaxRecords值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

：必需 否

响应元素

服务返回以下元素。

`dbInstances.dbinstance.n`

有关一个或多个实例的详细信息。

类型: 数组 [DBInstance](#) (p. 678) 对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

错误

有关所有操作中常见的错误的信息，请参阅 [常见错误](#) (p. 710)。

`DBInstanceNotFound`

`DBInstanceIdentifier` 不引用现有实例。

HTTP 状态代码：404

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeDBSubnetGroups

返回 DBSubnetGroup 描述的列表。如果 DBSubnetGroupName 列表中只包含指定的描述 DBSubnetGroup。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBSubnetGroupName

要为其返回详细信息的子网组的名称。

类型: 字符串

: 必需 否

过滤器。过滤器。

当前不支持此参数。

类型: 数组 [Filter \(p. 692\)](#) 对象

: 必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值: 100

约束: 最低为 20，最高为 100。

类型: 整数

: 必需 否

响应元素

服务返回以下元素。

`dbsubnetGroups.dbsubnetGroup.n`

有关一个或多个子网组的详细信息。

类型: 数组 [DBSubnetGroup \(p. 683\)](#) 对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBSubnetGroupNotFoundFault

DBSubnetGroupName不引用现有子网组。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeEngineDefaultClusterParameters

返回集群数据库引擎的默认引擎和系统参数信息。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBParameterGroupFamily

要返回其引擎参数信息的集群参数组系列的名称。

类型: 字符串

: 必填项 是

过滤器。过滤器。

当前不支持此参数。

类型: 数组 [Filter \(p. 692\)](#) 对象

: 必填项 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必填项 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值: 100

约束: 最低为 20，最高为 100。

类型: 整数

: 必填项 否

响应元素

服务返回以下元素。

EngineDefaults

包含成功调用 `DescribeEngineDefaultClusterParameters` operation。

类型: [EngineDefaults \(p. 686\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeEventCategories

显示所有事件源类型的类别列表；或如果指定，则显示指定源类型的类别列表。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

过滤器。过滤器。

当前不支持此参数。

类型: 数组[Filter \(p. 692\)](#)对象

: 必需 否

SourceType

生成事件的源的类型。

有效值: db-instance、db-parameter-group、db-security-group

类型: 字符串

: 必需 否

响应元素

服务返回的以下元素。

活动类别 maplist.活动类别map.Map.N

事件类别地图的列表。

类型: 数组[EventCategoriesMap \(p. 689\)](#)对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeEvents

返回过去 14 天内与实例、安全组、快照和数据库参数组相关的事件。对于特定的数据库实例、安全组、快照或参数组，特定于它们的事件可以通过提供名称作为参数来获取。默认情况下，返回过去一小时的事件。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

Duration

从中检索事件的分钟数。

默认值：60

类型：整数

：必需 否

EndTime

要检索事件的时间段的结束，以 ISO 8601 格式指定。

例如：2009-07-08T 18:00 Z

类型：时间戳

：必需 否

活动类别。EVENTCATEGORY.N

触发事件通知订阅的通知的事件类别列表。

类型：字符串数组

：必需 否

过滤器。过滤器。

当前不支持此参数。

类型：数组[Filter \(p. 692\)](#)对象

：必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型：字符串

：必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型：整数

: 必需 否

SourceIdentifier

为其返回事件的事件源的标识符。如果未指定，则响应中包含所有源。

约束：

- 如果SourceIdentifier提供，SourceType还必须提供。
- 如果源类型是DBInstance，aDBInstanceIdentifier必须提供。
- 如果源类型是DBSecurityGroup，aDBSecurityGroupName必须提供。
- 如果源类型是DBParameterGroup，aDBParameterGroupName必须提供。
- 如果源类型是DBSnapshot，aDBSnapshotIdentifier必须提供。
- 不能以连字符结束或包含两个连续连字符。

类型: 字符串

: 必需 否

SourceType

要从中检索事件的事件源。如果未指定值，则返回所有事件。

类型: 字符串

有效值: db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot

: 必需 否

StartTime

要检索事件的时间段的开始，以 ISO 8601 格式指定。

例如：2009-07-08T 18:00 Z

类型: 时间戳

: 必需 否

响应元素

服务返回以下元素。

事件。Event.N

有关一个或多个事件的详细信息。

类型: 数组 [Event \(p. 687\)](#) 对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 MaxRecords 指定的值。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeEventSubscriptions

列出客户账户的所有订阅描述。订阅的描述包

括SubscriptionName、SNSTopicARN、CustomerID、SourceType、SourceID、CreationTime、和Status。

如果你指定SubscriptionName，则列出该订阅的描述。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

过滤器。过滤器。

当前不支持此参数。

类型: 数组[Filter \(p. 692\)](#)对象

必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为MaxRecords 指定的值。

类型: 字符串

必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定MaxRecords值，则在响应中包含分页记号，以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

必需 否

SubscriptionName

您要描述的 Amazon DocumentDB 事件通知订阅的名称。

类型: 字符串

必需 否

响应元素

服务返回以下元素。

事件订阅 slist.EVENT 订阅。

活动订阅列表。

类型: 数组[EventSubscription \(p. 690\)](#)对象

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

SubscriptionNotFound

订阅名称不存在。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeGlobalClusters

返回 Amazon DocumentDB 全局集群的相关信息。此 API 支持分页。

Note

此操作仅适用于 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

过滤器。过滤器。

筛选条件指定要描述的一个或多个全局数据库集群。

支持的筛选条件：`db-cluster-id`接受集群标识符和集群 Amazon 资源名称 (ARN)。结果列表中仅包含由这些 ARN 标识的集群的相关信息。

类型: 数组[Filter \(p. 692\)](#)对象

必需 否

GlobalClusterIdentifier

用户提供的集群标识符。如果指定了此参数，则只返回特定集群的信息。该参数不区分大小写。

类型: 字符串

长度约束：最小长度为 1。长度上限为 255。

模式：`[A-Za-z][0-9A-Za-z-:._]*`

必需 否

Marker

由之前的 `DescribeGlobalClusters` 请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定`MaxRecords`值，则在响应中包含称为标记的分页记号，以便检索剩余的结果。

类型: 整数

必需 否

响应元素

服务返回以下元素。

全局群集。GlobalCluster 成员。

类型: 数组[GlobalCluster \(p. 693\)](#)对象

Marker

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

GlobalClusterNotFoundFault

这些区域有：GlobalClusterIdentifier不引用现有的全局集群。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonJava SDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribeOrderableDBInstanceOptions

返回指定引擎的可订购实例选项的列表。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

Engine

为其检索实例选项的引擎的名称。

类型: 字符串

: 必需 是

DBInstanceClass

实例类筛选值。指定此参数以只显示与指定实例类匹配的可用产品。

类型: 字符串

: 必需 否

EngineVersion

引擎版本筛选值。指定此参数以只显示与指定引擎版本匹配的可用产品。

类型: 字符串

: 必需 否

过滤器。过滤器。

当前不支持此参数。

类型: 数组的[Filter \(p. 692\)](#)对象

: 必需 否

LicenseModel

许可模式筛选值。指定此参数以只显示与指定许可模式匹配的可用产品。

类型: 字符串

: 必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

: 必需 否

Vpc

Virtual Private Cloud (VPC) 筛选值。指定此参数以只显示可用的 VPC 或非 VPC 产品。

类型: Boolean

: 必需 否

响应元素

服务返回以下元素。

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

可订购的数据实例选项。可订购的数据实例选项。

可用于特定可订购实例的选项。

类型: 数组的 [OrderableDBInstanceOption](#) (p. 696)对象

错误

有关所有操作常见错误的信息，请参阅[常见错误](#) (p. 710)。

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DescribePendingMaintenanceActions

返回至少具有一个待处理的维护操作的资源（例如，实例）的列表。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

过滤器。过滤器。

一个筛选条件，用于指定一个或多个资源以返回其待处理的维护操作。

支持的筛选条件：

- `db-cluster-id`-接受集群标识符和集群 Amazon 资源名称 (ARN)。结果列表只包括这些 ARN 标识的集群的待处理维护操作。
- `db-instance-id`-接受实例标识符和实例 ARN。结果列表只包括这些 ARN 标识的数据库实例的待处理维护操作。

类型: 数组[Filter \(p. 692\)](#)对象

: 必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必需 否

MaxRecords

包括在响应中的最大记录数。如果存在的记录多于指定 `MaxRecords` 值，则在响应中包含分页记号（标记），以便检索剩余的结果。

默认值：100

约束：最低为 20，最高为 100。

类型: 整数

: 必需 否

ResourceIdentifier

用于返回其待处理维护操作的资源的 ARN。

类型: 字符串

: 必需 否

响应元素

服务返回以下元素。

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

待定维护操作。资源待定维护操作。N

要应用的维护操作。

类型: 数组 [ResourcePendingMaintenanceActions](#) (p. 706)对象

错误

有关所有操作常见错误的信息，请参阅[常见错误](#) (p. 710).

ResourceNotFoundFault

找不到指定的资源 ID。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

FailoverDBCluster

强制集群故障转移。

集群的故障转移会将集群中的 Amazon DocumentDB 副本之一（只读实例）提升为主实例（集群写入器）。

如果主实例发生故障，将自动故障转移到 Amazon DocumentDB 副本（如果存在）。当您要模拟主实例的故障以进行测试时，可以强制进行故障转移。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

用于强制进行故障转移的集群标识符。此参数不区分大小写。

约束：

- 必须匹配现有标识符DBCluster。

类型: 字符串

: 必需 否

TargetDBInstanceIdentifier

提升为主实例的实例的名称。

您必须指定集群中 Amazon DocumentDB 副本的实例标识符。例如 mydbcluster-replica1。

类型: 字符串

: 必需 否

响应元素

服务返回以下元素。

DBCluster

有关集群的详细信息。

类型：[DBCluster \(p. 663\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码：404

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ListTagsForResource

列出 Amazon DocumentDB 资源上的所有标签。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

ResourceName

带有要列出的标签的 Amazon DocumentDB 资源。此值是 Amazon 资源名称 (ARN)。

类型: 字符串

：必填项：是
过滤器。过滤器。

当前不支持此参数。

类型: 数组[Filter \(p. 692\)](#)对象

：必填项：否

响应元素

服务返回以下元素。

标签 List.tag.Tag.N

列出一个或多个标签。

类型: 数组[Tag \(p. 708\)](#)对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码：404

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码：404

DBSnapshotNotFound

DBSnapshotIdentifier不引用现有快照。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ModifyDBCluster

修改 Amazon DocumentDB 集群的设置。您可通过在请求中指定这些参数以及新值，更改一个或多个数据库配置参数。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

正在修改的集群的集群标识符。此参数不区分大小写。

约束：

- 必须匹配现有标识符DBCluster。

类型: 字符串

：必需 是

ApplyImmediately

指定是否应尽快异步应用此请求中更改及任何待处理更改的值，无论PreferredMaintenanceWindow设置集群。如果此参数设置为false对集群的更改将在下一维护时段内应用对集群的更改。

这些区域有：ApplyImmediately参数只影响NewDBClusterIdentifier和MasterUserPassword有效值。如果将此参数值设置为false，对NewDBClusterIdentifier和MasterUserPassword值将在下一维护时段内应用。所有其他更改会立即应用，而不管 ApplyImmediately 参数的值如何。

默认值：false

类型: Boolean

：必需 否

BackupRetentionPeriod

自动备份的保留天数。您必须指定最小值 1。

默认值：1

约束：

- 必须为介于 1 和 35 之间的值。

类型: 整数

：必需 否

CloudwatchLogsExportConfiguration

要启用日志类型的配置设置，以便针对特定实例或集群导出到 Amazon CloudWatch Logs。这些区域有：EnableLogTypes和DisableLogTypes数组确定要将哪些日志导出（或不导出）到 CloudWatch Logs。

类型：[CloudwatchLogsExportConfiguration \(p. 662\)](#) 对象

：必需 否

DBClusterParameterGroupName

用于集群的集群参数组的名称。

类型: 字符串

: 必需 否

DeletionProtection

指定是否可以删除此集群。如果DeletionProtection如果启用，则无法删除集群，除非集群被修改并DeletionProtection已禁用。DeletionProtection防止意外删除集群。

类型: Boolean

: 必需 否

EngineVersion

要升级到的数据库引擎的版本号。Amazon DocumentDB 不支持修改引擎版本。

类型: 字符串

: 必需 否

MasterUserPassword

主数据库用户的密码。此密码可以包含除正斜杠 (/)、双引号 (") 或 @ 符号之外的任何可打印的 ASCII 字符。

约束：必须包含 8 到 100 个字符。

类型: 字符串

: 必需 否

NewDBClusterIdentifier

重命名集群时集群的新集群标识符。此值以一个小写字符串存储。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：my-cluster2

类型: 字符串

: 必需 否

Port

集群接受连接的端口号。

约束：值必须是来自的1150到65535。

默认值：与原始集群相同的端口。

类型: 整数

: 必需 否

PreferredBackupWindow

使用 BackupRetentionPeriod 参数启用了自动备份时，自动执行备份的日常时间范围。

默认值是从每个时间段中随机选择的 30 分钟时间。Amazon Web Services 区域。

约束：

- 必须采用 hh24:mi-hh24:mi 格式。
- 必须采用通用协调时间 (UTC)。
- 不得与首选维护时段冲突。
- 必须至少为 30 分钟。

类型: 字符串

：必需 否

PreferredMaintenanceWindow

可进行系统维护的每周时间范围（采用通用协调时间 (UTC)）。

格式：ddd:hh24:mi-ddd:hh24:mi

默认值是从每个时间段中随机选择的 30 分钟时间。Amazon Web Services 区域，它可以是一周中的任意一天。

有效值：Mon、Tue、Tue、Tue、Fi、Sun

约束：至少为 30 分钟的时段

类型: 字符串

：必需 否

vpc 安全组 ID.vpc 安全组 ID.N

集群所属的虚拟私有云 (VPC) 安全组的列表。

类型: 字符串数组

：必需 否

响应元素

服务返回以下元素。

DBCluster

有关集群的详细信息。

类型：[DBCluster \(p. 663\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterAlreadyExistsFault

您已拥有具有给定标识符的集群。

HTTP 状态代码：400

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码：404

DBClusterParameterGroupNotFound

DBClusterParameterGroupName不引用现有集群参数组。

HTTP 状态代码：404

DBSubnetGroupNotFoundFault

DBSubnetGroupName不引用现有子网组。

HTTP 状态代码：404

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

InvalidDBSecurityGroupState

安全组的状态不允许执行删除。

HTTP 状态代码：400

InvalidDBSubnetGroupStateFault

无法删除子网组，因为它正在使用中。

HTTP 状态代码：400

InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于常见虚拟私有云 (VPC) 中。

HTTP 状态代码：400

InvalidVPCNetworkStateFault

由于所做的更改，子网组在创建后并不覆盖所有可用区。

HTTP 状态代码：400

StorageQuotaExceeded

请求将导致超出所有实例中可用的允许存储量。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)

- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ModifyDBClusterParameterGroup

修改集群参数组的参数。要修改多个参数，请提交以下对象的列表：ParameterName、ParameterValue 和 ApplyMethod。在单个请求中，最多可以修改 20 个参数。

Note

对动态参数所做的更改将立即应用。对静态参数所做的更改需要在重启或维护窗口后才生效。

Important

创建集群参数组之后，您应至少等待 5 分钟，再创建使用该集群参数组作为默认参数组的第一个集群。这样，在将参数组用作新集群的默认设置之前，Amazon DocumentDB 可以完成全部创建操作。此步骤对于在为集群创建默认数据库时十分关键的参数（例如，由 character_set_database 参数定义的默认数据库字符集）非常重要。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterParameterGroupName

要修改的集群参数组的名称。

类型: 字符串

: 必需 是

参数。参数。

要修改的集群参数组中的参数列表。

类型: 数组 [Parameter \(p. 698\)](#) 对象

: 必需 是

响应元素

服务返回以下元素。

DBClusterParameterGroupName

集群参数组的名称。

约束：

- 必须为 1 到 255 个字母或数字。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

Note

此值以一个小写字符串存储。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBParameterGroupNotFound

DBParameterGroupName不引用现有参数组。

HTTP 状态代码：404

InvalidDBParameterGroupState

参数组正在使用中，或者处于无效状态。如果您尝试删除参数组，则在参数组处于此状态时无法将其删除。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ModifyDBClusterSnapshotAttribute

向手动集群快照添加属性和值，或者从中删除属性和值。

与其他人共享手动集群快照Amazon Web Services 账户，请指定`restore`作为`AttributeName`，然后使用`ValuesToAdd`参数来添加的 ID 列表Amazon Web Services 账户有权恢复手动群集快照。使用该值`all`要公开手动集群快照，这意味着所有人都可以复制或还原它Amazon Web Services 账户。不要添加`all`对于任何手动集群快照包含您不想向所有人公开的私有信息，则值Amazon Web Services 账户。如果手动集群快照已加密，则可以共享它，但只能通过指定已授权列表来共享。Amazon Web Services 账户ID 的`ValuesToAdd`参数。在这种情况下，您不能使用 `all` 作为该参数的值。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

AttributeName

要修改的集群快照属性的名称。

管理其他的授权Amazon Web Services 账户要复制或还原手动集群快照，请将此值设置为`restore`。

类型: 字符串

: 必需 是

DBClusterSnapshotIdentifier

要修改其属性的集群快照的标识符。

类型: 字符串

: 必需 是

值要添加属性值。N

要添加到所指定属性中的集群快照属性的列表`AttributeName`。

授权其他Amazon Web Services 账户要复制或还原手动集群快照，请将此列表设置为包括一个或多个集群快照Amazon Web Services 账户ID。使手动集群快照可以通过任何方式恢复Amazon Web Services 账户，将其设置为`all`。不要添加`all`对于任何手动集群快照包含您不想向所有人公开的私有信息，则值Amazon Web Services 账户。

类型: 字符串数组

: 必需 否

要删除的值。属性值。N

要从所指定属性中删除的集群快照属性的列表`AttributeName`。

删除对其他人的授权Amazon Web Services 账户要复制或还原手动集群快照，请将此列表设置为包括一个或多个集群快照Amazon Web Services 账户标识符。删除任何人的授权Amazon Web Services 账户要复制或还原群集快照，请将其设置为`all`。如果你指定`all`，一个Amazon Web Services 账户谁的账户ID 被明确添加到`restore`属性仍可以复制或还原手动集群快照。

类型: 字符串数组

: 必需 否

响应元素

服务返回以下元素。

DBClusterSnapshotAttributesResult

与集群快照关联的属性的详细信息。

类型：[DBClusterSnapshotAttributesResult \(p. 675\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`未引用现有集群快照。

HTTP 状态代码：404

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

SharedSnapshotQuotaExceeded

您已超过您可与其共享手动数据库快照的最大账户数。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ModifyDBInstance

修改实例的设置。您可通过在请求中指定这些参数以及新值，更改一个或多个数据库配置参数。

请求参数

有关所有操作的通用参数的信息，请参见[常见参数 \(p. 712\)](#)。

DBInstanceIdentifier

实例标识符。此值以一个小写字符串存储。

约束：

- 必须匹配现有标识符匹配DBInstance。

类型: 字符串

必需 是

ApplyImmediately

指定是否尽快异步应用此请求中的修改和任何待处理的修改，无论如何PreferredMaintenanceWindow为实例设置。

如果将参数设置为false，对实例所做更改在下一维护时段内生效。某些参数更改可能会导致中断，并在下次重新启动时应用。

默认值: false

类型: 布尔值

必需 否

AutoMinorVersionUpgrade

此参数不适用于 Amazon DocumentDB。无论设置的值如何，Amazon DocumentDB 都不会执行次要版本升级。

类型: 布尔值

必需 否

CACertificateIdentifier

指示需要与实例相关联的证书。

类型: 字符串

必需 否

CopyTagsToSnapshot

指示是否将数据库实例的所有标签复制到数据库实例快照的值。默认情况下，不复制标签。

类型: 布尔值

必需 否

DBInstanceClass

实例的新计算和内存容量；例如，db.r5.large。并非所有实例类都可用Amazon Web Services 区域。

如果您修改实例类，则在更改期间会发生中断。更改在下一个维护时段内应用，除非此请求的ApplyImmediately 指定为 true。

默认值：使用现有设置。

类型: 字符串

必需 否

EnablePerformanceInsights

指示是否为数据库实例启用 Performance Insights 的值。有关更多信息，请参阅 [使用亚马逊 Performance Insights](#)。

类型: 布尔值

必需 否

NewDBInstanceIdentifier

重命名实例时，实例的新实例标识符。当您更改实例标识符时，如果您设置了实例标识符，则实例会立即重启Apply Immediately到true。如果设置，则在下一维护时段内生效Apply Immediately到false。此值以一个小写字符串存储。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：mydbinstance

类型: 字符串

必需 否

PerformanceInsightsKMSKeyId

用于加密 Performance Insights 数据的 Amazon KMS 密钥标识符。

这些区域有：Amazon KMS密钥 ARN、密钥 ID、别名 ARN、密钥 ID、密钥 ID、密钥 ID、密钥 ID、密钥 ID、密钥 ID、密钥 ID、密钥 ID、

如果您未指定值 PerformanceInsights自杀KeyId，则亚马逊文档数据库会使用您的默认 KS DocumentDB 使用您的默认 K 您的亚马逊云科技账户有一个默认 KS 云密钥。您的Amazon Web Services 账户在每个Amazon Web Services 云科技区域都有不同的默认 KMS 密钥。

类型: 字符串

必需 否

PreferredMaintenanceWindow

可进行系统维护的每周时间范围（采用 UTC），这可能导致中断。更改此参数不会导致中断，所做更改不会导致中断，所做更改会尽快以异步方式应用。如果有待操作导致重新启动，并且维护时段经过更改，加入了当前时间，更改此参数会导致实例重新启动。如果将此时段移到当前时间，则当前时间与该时段结束之间必须相隔至少 30 分钟以确保应用待处理的更改。

默认值：使用现有设置。

格式：ddd:hh24:mi-ddd:hh24:mi

有效值：Mon、Tue、Wed、Thu、Fri、Wed、Fri、Sat、

约束：必须至少为 30 分钟。

类型: 字符串

必需 否

PromotionTier

该值指定在现有主实例发生故障后将 Amazon DocumentDB 副本提升为主实例的顺序。

默认值：1

有效值：0-15

类型：整数

必需 否

响应元素

以下元素由服务返回。

DBInstance

有关实例的详细信息。

类型：[DBInstance \(p. 678\)](#) 对象

错误

有关所有操作的常见错误的信息，请参见[常见错误 \(p. 710\)](#)。

AuthorizationNotFound

指定的 CIDR IP 或 Amazon EC2 安全组未被授权进入指定的安全组。

Amazon DocumentDB B 也可能无权使用 IAM 代表您执行必要的操作。

HTTP 状态代码：404

CertificateNotFound

CertificateIdentifier不引用现有证书。

HTTP 状态代码：404

DBInstanceAlreadyExists

您已经有一个具有给定标识符的实例。

HTTP 状态代码：400

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码：404

DBParameterGroupNotFound

DBParameterGroupName不引用现有参数组。

HTTP 状态代码：404

DBSecurityGroupNotFound

DBSecurityGroupName不引用现有安全组。

HTTP 状态代码：404

DBUpgradeDependencyFailure

升级失败，因为依赖的资源无法修改。

HTTP 状态代码：400

InsufficientDBInstanceCapacity

指定的实例类在指定可用区中不可用。

HTTP 状态代码：400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

InvalidDBSecurityGroupState

安全组的状态不允许删除。

HTTP 状态代码：400

InvalidVPCNetworkStateFault

由于进行了更改，子网组创建后并未覆盖所有可用区。

HTTP 状态代码：400

StorageQuotaExceeded

该请求将导致您超过所有实例的允许可用存储量。

HTTP 状态代码：400

StorageTypeNotSupported

存储指定的StorageType无法与数据库实例相关联。

HTTP 状态代码：400

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon 命令行界面](#)
- [适用于 .NET 的 Amazon SDK](#)
- [适用于 C++ 的 Amazon SDK](#)
- [适用于 Go 的 Amazon SDK](#)
- [适用于 Java V2 的 Amazon SDK](#)
- [Amazon 开发工具包适用于 JavaScript](#)
- [适用于 PHP V3 的 Amazon SDK](#)
- [适用于 Python 的 Amazon SDK](#)
- [适用于 Ruby V3 的 Amazon SDK](#)

ModifyDBSubnetGroup

修改现有子网组。子网组必须至少包含给定的至少两个可用区的一个子网。Amazon Web Services 区域。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBSubnetGroupName

子网组的名称。此值以一个小写字符串存储。您不能修改默认子网组。

约束：必须与现有 DBSubnetGroup 的名称匹配。不能是默认值。

示例：mySubnetgroup

类型：字符串

：必需 是

subnetids.subnetid 标识符。

子网组的 Amazon EC2 子网 ID。

类型：字符串数组

：必需 是

DBSubnetGroupDescription

子网组的描述。

类型：字符串

：必需 否

响应元素

服务返回以下元素。

DBSubnetGroup

有关子网组的详细信息。

类型：[DBSubnetGroup \(p. 683\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBSubnetGroupDoesNotCoverEnoughAZs

除非只有一个可用区，否则子网组中的子网应至少包含两个可用区。

HTTP 状态代码：400

DBSubnetGroupNotFoundFault

DBSubnetGroupName 不引用现有子网组。

HTTP 状态代码：404

DBSubnetQuotaExceededFault

请求将导致您超出子网组中允许的子网数目。

HTTP 状态代码：400

InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于常见虚拟私有云 (VPC) 中。

HTTP 状态代码：400

SubnetAlreadyInUse

子网已在可用区中使用。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ModifyEventSubscription

修改现有的 Amazon DocumentDB 事件通知订阅。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

SubscriptionName

Amazon DocumentDB 事件通知订阅的名称。

类型: 字符串

: 必填项: 是

Enabled

布尔值；设置为true以激活订阅。

类型: Boolean

: 必填项: 否

活动类别。EVENTCATEGORY.N

事件类别列表SourceTypes您要订阅的。

类型: 字符串数组

: 必填项: 否

SnsTopicArn

为事件通知创建的 SNS 主题的 Amazon 资源名称 (ARN)。在您创建主题并订阅到该主题时，由 Amazon SNS 创建 ARN。

类型: 字符串

: 必填项: 否

SourceType

生成事件的源的类型。例如，如果希望收到关于实例生成的事件的通知，请将该参数设置为db-instance。如果未指定该值，则将返回所有事件。

有效值: db-instance、db-parameter-group、db-security-group

类型: 字符串

: 必填项: 否

响应元素

服务返回以下元素。

EventSubscription

关于您订阅的事件的详细信息。

类型: [EventSubscription \(p. 690\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

EventSubscriptionQuotaExceeded

您已达到事件订阅的最大数量。

HTTP 状态代码：400

SNSInvalidTopic

Amazon SNS 回应说，指定主题存在问题。

HTTP 状态代码：400

SNSNoAuthorization

您没有向 SNS 主题发布 Amazon 资源名称 (ARN) 的权限。

HTTP 状态代码：400

SNSTopicArnNotFound

SNS 主题的 Amazon 资源名称 (ARN) 不存在。

HTTP 状态代码：404

SubscriptionCategoryNotFound

提供的类别不存在。

HTTP 状态代码：404

SubscriptionNotFound

订阅名称不存在。

HTTP 状态代码：404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ModifyGlobalCluster

修改 Amazon DocumentDB 全局集群的设置。您可通过在请求中指定这些参数以及新值，更改一个或多个配置参数（例如：删除保护）或全局集群标识符。

Note

此操作仅适用于 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

GlobalClusterIdentifier

要修改的全局集群的标识符。该参数不区分大小写。

约束：

- 必须与现有全局集群的标识符匹配。

类型: 字符串

长度约束：最小长度为 1。长度上限为 255。

模式：`[A-Za-z][0-9A-Za-z-:._]*`

：必填项：是

DeletionProtection

指示全局集群是否启用了删除保护。在启用删除保护时，无法删除全局集群。

类型: Boolean

：必填项：否

NewGlobalClusterIdentifier

修改全局集群时的全局集群的新标识符。此值以一个小写字符串存储。

- 必须包含 1 到 63 个字母、数字或连字符

第一个字符必须是字母

不能以连字符结束或包含两个连续连字符

示例：`my-cluster2`

类型: 字符串

长度约束：最小长度为 1。长度上限为 255。

模式：`[A-Za-z][0-9A-Za-z-:._]*`

：必填项：否

响应元素

服务返回以下元素。

GlobalCluster

表示 Amazon DocumentDB 全局集群的数据类型。

类型：[GlobalCluster \(p. 693\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

GlobalClusterNotFoundFault

这些区域有：`GlobalClusterIdentifier`不引用现有全局集群。

HTTP 状态代码：404

InvalidGlobalClusterStateFault

当群集处于此状态时，无法执行请求的操作。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

RebootDBInstance

您可能需要重启实例，通常是出于维护目的。例如，如果进行某些更改或更改与实例关联的集群参数组，您必须重启该实例以使更改生效。

重启实例会重新启动数据库引擎服务。重启实例将导致短暂中断，在此期间，实例的状态将设置为rebooting。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBInstanceIdentifier

实例标识符。该参数作为一个小写字符串存储。

约束：

- 必须匹配现有的标识符DBInstance。

类型: 字符串

：必需 是

ForceFailover

何时true，则通过多可用区故障转移进行重启。

约束：您不能指定true如果实例没有为多可用区配置。

类型: Boolean

：必需 否

响应元素

服务返回以下元素。

DBInstance

有关实例的详细信息。

类型：[DBInstance \(p. 678\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码：404

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

RemoveFromGlobalCluster

从全局群集中分离 Amazon DocumentDB 辅助群集。该集群变成具有读写功能的独立群集，而不是只读集群并从不同区域的主群集接收数据。

Note

此操作仅适用于 Amazon DocumentDB 集群。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DbClusterIdentifier

从 Amazon DocumentDB 全局集群分离的集群的 Amazon 资源名称 (ARN)。

类型: 字符串

必填项: 是

GlobalClusterIdentifier

要从 Amazon DocumentDB 全局集群中分离的群集标识符。

类型: 字符串

长度约束: 最小长度为 1。长度上限为 255。

模式: `[A-Za-z][0-9A-Za-z-:._]*`

必填项: 是

响应元素

服务返回以下元素。

GlobalCluster

表示 Amazon DocumentDB 全局集群的数据类型。

类型: [GlobalCluster \(p. 693\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

`DBClusterIdentifier`不引用现有集群。

HTTP 状态代码: 404

GlobalClusterNotFoundFault

这些区域有: `GlobalClusterIdentifier`不引用现有全局集群。

HTTP 状态代码: 404

InvalidGlobalClusterStateFault

当群集处于此状态时，无法执行请求的操作。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

RemoveSourceIdentifierFromSubscription

从现有 Amazon DocumentDB 事件通知订阅中删除源标识符。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

SourceIdentifier

要从订阅中删除的源标识符，例如实例的实例标识符或安全组的名称。

类型: 字符串

: 必需 是

SubscriptionName

从中删除源标识符的 Amazon DocumentDB 事件通知订阅的名称。

类型: 字符串

: 必需 是

响应元素

服务返回以下元素。

EventSubscription

有关您订阅的事件的详细信息。

类型: [EventSubscription \(p. 690\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

SourceNotFound

找不到请求的源。

HTTP 状态代码: 404

SubscriptionNotFound

订阅名称不存在。

HTTP 状态代码: 404

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)

- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

RemoveTagsFromResource

从 Amazon DocumentDB 资源中删除元数据标签。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

ResourceName

从其中删除标签的 Amazon DocumentDB 资源。此值是 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 是

tagkeys.Member。

要删除的标签的标签键 (名称)。

类型: 字符串数组

: 必需 是

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码 : 404

DBInstanceNotFound

DBInstanceIdentifier不引用现有实例。

HTTP 状态代码 : 404

DBSnapshotNotFound

DBSnapshotIdentifier不引用现有的快照。

HTTP 状态代码 : 404

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)

- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ResetDBClusterParameterGroup

将集群参数组的参数修改为默认值。要重置特定参数，请提交以下内容的列表：ParameterName和ApplyMethod。要重置整个集群参数组，请指定DBClusterParameterGroupName和ResetAllParameters参数。

重置整个组时，动态参数立即更新，静态参数设置为。pending-reboot以便在下次数据库实例重启时生效。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterParameterGroupName

要重置的集群参数组的名称。

类型: 字符串

: 必需 是

参数。参数。

集群参数组中要重置为默认值的参数名称列表。如果 ResetAllParameters 参数设置为 true，则无法使用此参数。

类型: 数组 [Parameter \(p. 698\)](#) 对象

: 必需 否

ResetAllParameters

设置为的值 true 将集群参数组中的所有参数重置为其默认值，以及 false 否则为。如果为 Parameters 参数指定了参数名称列表，您无法使用此参数。

类型: Boolean

: 必需 否

响应元素

服务返回以下元素。

DBClusterParameterGroupName

集群参数组的名称。

约束：

- 必须为 1 到 255 个字母或数字。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

Note

此值以一个小写字符串存储。

类型: 字符串

错误

有关所有操作常见错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBParameterGroupNotFound

DBParameterGroupName不引用现有参数组。

HTTP 状态代码：404

InvalidDBParameterGroupState

参数组正在使用中，或者处于无效状态。如果您尝试删除参数组，则在参数组处于此状态时无法将其删除。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的Amazon开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

RestoreDBClusterFromSnapshot

根据快照或集群快照创建新集群。

如果指定快照，则使用默认配置和默认安全组，从源数据库快照创建目标集群。

如果指定集群快照，则使用具有与原始源数据库集群相同配置的源集群恢复点，创建目标集群，不同之处在于新集群在默认安全组中创建。

请求参数

有关所有操作通用的参数的信息，请参阅[常见参数 \(p. 712\)](#)。

DBClusterIdentifier

要从快照或集群快照创建的集群的名称。该参数不区分大小写。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

示例：my-snapshot-id

类型：字符串

：必需 是

Engine

要用于新集群的数据库引擎。

默认值：与源相同。

约束：必须与源的引擎兼容。

类型：字符串

：必需 是

SnapshotIdentifier

要从中进行还原的快照或集群快照的标识符。

您可以使用名称或 Amazon 资源名称 (ARN) 指定集群快照。但是，您只能使用 ARN 指定快照。

约束：

- 必须与现有快照的标识符匹配。

类型：字符串

：必需 是

可用性区域。可用性区域。N

提供可在其中创建还原数据库集群中的实例的 Amazon EC2 可用区的列表。

类型：字符串数组

：必需 否

DBSubnetGroupName

要用于新集群的子网组的名称。

约束：如果提供，则必须与现有的名称匹配DBSubnetGroup。

示例：mySubnetgroup

类型: 字符串

：必需 否

DeletionProtection

指定是否可以删除此集群。如果DeletionProtection如果启用，则无法删除集群，除非集群被修改并DeletionProtection已禁用。DeletionProtection防止意外删除集群。

类型: Boolean

：必需 否

启用 CloudWatch LogSports.Member。

必须启用以导出到 Amazon CloudWatch Logs 的日志类型的列表。

类型: 字符串数组

：必需 否

EngineVersion

要用于新集群的数据库引擎的版本。

类型: 字符串

：必需 否

KmsKeyId

这些区域有：Amazon KMS从数据库快照或集群快照还原加密集群时使用的密钥标识符。

这些区域有：Amazon KMS键标识符 ARN 于Amazon KMS加密密钥。如果要使用相同的集群还原Amazon Web Services 账户那个拥有Amazon KMS用于加密新集群的加密密钥，然后你可以使用Amazon KMS关键别名而不是 ARNAmazon KMS加密密钥。

如果不为 KmsKeyId 参数指定值，则会出现以下情况：

- 如果中的快照或集群快照SnapshotIdentifier将加密，将使用Amazon KMS用于加密快照或集群快照的密钥。
- 如果快照或群集快照SnapshotIdentifier未加密，还原的数据库集群也不会加密。

类型: 字符串

：必需 否

Port

新集群接受连接的端口号。

约束：值必须来自1150到65535。

默认值：与原始集群相同的端口。

类型: 整数

：必需 否

标签 s.Tag.N

要分配给还原的集群的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

: 必需 否

vpc 安全组 ID.vpc 安全组 ID.N

新集群所属的虚拟私有云 (VPC) 安全组的列表。

类型: 字符串数组

: 必需 否

响应元素

服务返回以下元素。

DBCluster

有关集群的详细信息。

类型 : [DBCluster \(p. 663\)](#) 对象

错误

有关所有操作中常见的错误的信息，请参阅[常见错误 \(p. 710\)](#)。

DBClusterAlreadyExistsFault

您已拥有具有给定标识符的集群。

HTTP 状态代码 : 400

DBClusterQuotaExceededFault

无法创建集群，因为您已经达到了允许的最大集群配额。

HTTP 状态代码 : 403

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier 不引用现有集群快照。

HTTP 状态代码 : 404

DBSnapshotNotFound

DBSnapshotIdentifier 不引用现有快照。

HTTP 状态代码 : 404

DBSubnetGroupNotFoundFault

DBSubnetGroupName 不引用现有子网组。

HTTP 状态代码 : 404

DBSubnetGroupNotFoundFault

DBSubnetGroupName 不引用现有子网组。

HTTP 状态代码：404

InsufficientDBClusterCapacityFault

集群没有足够的容量用于当前操作。

HTTP 状态代码：403

InsufficientStorageClusterCapacity

当前操作没有足够的可用存储空间。通过更新子网组来使用具有更多可用存储空间的不同可用区，可以解决此错误。

HTTP 状态代码：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

InvalidDBSnapshotState

快照的状态不允许执行删除。

HTTP 状态代码：400

InvalidRestoreFault

您无法从虚拟私有云 (VPC) 备份还原到非 VPC 数据库实例。

HTTP 状态代码：400

InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于公共虚拟私有云 (VPC) 中。

HTTP 状态代码：400

InvalidVPCNetworkStateFault

由于所做的更改，子网组在创建后并不会覆盖所有可用区。

HTTP 状态代码：400

KMSKeyNotAccessibleFault

访问Amazon KMS键。

HTTP 状态代码：400

StorageQuotaExceeded

请求将导致超出所有实例中可用的允许存储量。

HTTP 状态代码：400

StorageQuotaExceeded

请求将导致超出所有实例中可用的允许存储量。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)

- [适用于 .NET 的 Amazon 开发工具包](#)
- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

RestoreDBClusterToPointInTime

将集群还原到任意时间点。用户可以还原到 LatestRestorableTime 之前最多 BackupRetentionPeriod 天的任意时间点。目标集群是从源集群创建的，其配置与原始集群相同，唯一的区别是新集群是使用默认安全组创建的。

请求参数

有关所有操作的通用参数的信息，请参见 [常见参数 \(p. 712\)](#)。

DBClusterIdentifier

要创建的新集群的名称。

约束：

- 必须包含 1 到 63 个字母、数字或连字符。
- 第一个字符必须是字母。
- 不能以连字符结束或包含两个连续连字符。

类型: 字符串

需 是

SourceDBClusterIdentifier

要从中还原的源集群的标识符。

约束：

- 必须与现有标识符匹配 DBCluster。

类型: 字符串

需 是

DBSubnetGroupName

用于新集群的子网组名称。

约束：如果提供，则与现有的名称匹配 DBSubnetGroup。

示例：mySubnetgroup

类型: 字符串

需 否

DeletionProtection

指定是否可以删除此集群。如果 DeletionProtection 已删除集群，除非集群被修改并 DeletionProtection 已禁用。DeletionProtection 防止意外删除集群。

类型: 布尔值

需 否

EnableCloudwatchLogsExports.member.n

必须启用后才能导出到 Amazon 的日志类型的列表 CloudWatch 日志。

类型: 字符串数组

需 否

KmsKeyId

这些区域有：Amazon KMS从加密集群恢复加密集群时使用的密钥标识符。

Amazon KMS 密钥标识符是 Amazon KMS 加密密钥的 Amazon 资源名称 (ARN)。如果您要恢复具有相同功能的集群Amazon Web Services 账户谁拥有Amazon KMS用于加密新集群的加密密钥，则可以使用Amazon KMS密钥别名而不是 ARNAmazon KMS加密密钥。

您可以恢复到新集群并使用以下命令对新集群进行加密Amazon KMS密钥的密钥Amazon KMS用于加密源集群的密钥。新的数据库集群已加密Amazon KMS密钥由标识KmsKeyId参数。

如果不为 KmsKeyId 参数指定值，则会出现以下情况：

- 如果集群已加密，则使用对还原的集群进行加密Amazon KMS用于加密源群集的密钥。
- 如果集群未加密，则还原的集群未加密。

如果DBClusterIdentifier指未加密的集群，则恢复请求被拒绝。

类型: 字符串

需 否

Port

新集群用于接受连接的端口号。

约束：值必须为来自于1150到65535。

默认值：引擎的默认端口。

类型: 整数

需 否

RestoreToTime

将集群恢复到的日期和时间。

有效值：采用通用协调时间 (UC) 格式的时间 (UC)

约束：

- 必须早于实例的最新可恢复时间。
- 如果未提供 UseLatestRestorableTime 参数，则必须指定。
- 如果 UseLatestRestorableTime 参数为 true，则无法指定。
- 如果 RestoreType 参数为 copy-on-write，则无法指定。

示例：2015-03-07T23:45:00Z

类型: 时间戳

需 否

RestoreType

要执行的还原类型。可以指定以下值之一：

- full-copy – 新数据库集群作为源数据库集群的完整副本还原。
- copy-on-write – 新数据库集群作为源数据库集群的克隆还原。

约束：你无法指定copy-on-write如果源数据库集群的引擎版本早于 1.11。

如果您没有指定 RestoreType 值，则新数据库集群作为源数据库集群的完整副本还原。

类型: 字符串

需 否

Tags.Tag.N

要分配给还原集群的标签。

类型: 数组 [Tag \(p. 708\)](#) 对象

需 否

UseLatestRestorableTime

一个设置为的值 `true` 将集群还原到最近的可还原备份时间，`false` 否则。

默认值 : `false`

约束 : 如果未提供 `RestoreToTime` 参数，则无法指定。

类型: 布尔值

需 否

VpcSecurityGroupIds.VpcSecurityGroupId否

新集群所属的 VPC 安全组的列表。

类型: 字符串数组

需 否

响应元素

以下元素由服务返回。

DBCluster

有关集群的详细信息。

类型 : [DBCluster \(p. 663\)](#) 对象

错误

有关所有操作的常见错误的信息，请参见 [常见错误 \(p. 710\)](#)。

DBClusterAlreadyExistsFault

您已经有一个具有给定标识符的集群。

HTTP 状态代码 : 400

DBClusterNotFoundFault

`DBClusterIdentifier` 不引用现有集群。

HTTP 状态代码 : 404

DBClusterQuotaExceededFault

无法创建集群，因为您已达到允许的最大集群配额。

HTTP 状态代码 : 403

DBClusterSnapshotNotFoundFault

DBClusterSnapshotIdentifier不引用现有集群快照。

HTTP 状态代码：404

DBSubnetGroupNotFoundFault

DBSubnetGroupName不引用现有子网组。

HTTP 状态代码：404

InsufficientDBClusterCapacityFault

集群没有足够的容量来执行当前操作。

HTTP 状态代码：403

InsufficientStorageClusterCapacity

没有足够的存储空间可供当前操作使用。您可以通过更新子网组以使用具有更多可用存储空间的不同可用区来解决此错误。

HTTP 状态代码：400

InvalidDBClusterSnapshotStateFault

提供的值不是有效的集群快照状态。

HTTP 状态代码：400

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码：400

InvalidDBSnapshotState

快照的状态不允许删除。

HTTP 状态代码：400

InvalidRestoreFault

您无法从 Virtual Private Cloud (VPC) 中还原到非 VPC

HTTP 状态代码：400

InvalidSubnet

请求的子网无效，或者请求的多个子网并非全部位于公共虚拟私有云 (VPC) 中。

HTTP 状态代码：400

InvalidVPCNetworkStateFault

由于进行了更改，子网组在创建后并未覆盖所有可用区。

HTTP 状态代码：400

KMSKeyNotAccessibleFault

访问时出错Amazon KMS钥匙。

HTTP 状态代码：400

StorageQuotaExceeded

该请求将导致您超过所有实例的允许可用存储量。

HTTP 状态代码：400

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon 命令行界面](#)
- [适用于 .NET 的 Amazon SDK](#)
- [适用于 C++ 的 Amazon SDK](#)
- [适用于 Go 的 Amazon SDK](#)
- [适用于 Java V2 的 Amazon SDK](#)
- [Amazon 开发工具包 JavaScript](#)
- [适用于 PHP V3 的 Amazon SDK](#)
- [适用于 Python 的 Amazon SDK](#)
- [适用于 Ruby V3 的 Amazon SDK](#)

StartDBCluster

重新启动由指定的已停止集群DBClusterIdentifier. 有关更多信息, 请参阅 [停止和启动 Amazon DocumentDB 集群](#).

请求参数

有关所有操作通用的参数的信息, 请参阅[常见参数 \(p. 712\)](#).

DBClusterIdentifier

要重启的集群的标识符。示例: docdb-2019-05-28-15-24-52

类型: 字符串

必填项: 是

响应元素

服务返回以下元素。

DBCluster

有关集群的详细信息。

类型: [DBCluster \(p. 663\)](#) 对象

错误

有关所有操作中常见的错误的信息, 请参阅[常见错误 \(p. 710\)](#).

DBClusterNotFoundFault

DBClusterIdentifier不引用现有集群。

HTTP 状态代码: 404

InvalidDBClusterStateFault

集群未处于有效状态。

HTTP 状态代码: 400

InvalidDBInstanceState

指定的实例不在可用状态。

HTTP 状态代码: 400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息, 请参阅以下内容:

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)

- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

StopDBCluster

停止由指定的正在运行的集群 `DBClusterIdentifier`。集群必须位于可用状态。有关更多信息，请参阅 [停止和启动 Amazon DocumentDB 集群](#)。

请求参数

有关所有操作通用的参数的信息，请参阅 [常见参数 \(p. 712\)](#)。

`DBClusterIdentifier`

要停止的集群的标识符。示例：docdb-2019-05-28-15-24-52

类型：字符串

必需：是

响应元素

服务返回以下元素。

`DBCluster`

有关集群的详细信息。

类型：[DBCluster \(p. 663\)](#) 对象

错误

有关所有操作常见错误的信息，请参阅 [常见错误 \(p. 710\)](#)。

`DBClusterNotFoundFault`

`DBClusterIdentifier`不引用现有集群。

HTTP 状态代码：404

`InvalidDBClusterStateFault`

集群未处于有效状态。

HTTP 状态代码：400

`InvalidDBInstanceState`

指定的实例不在可用状态。

HTTP 状态代码：400

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [Amazon Command Line Interface](#)
- [适用于 .NET 的Amazon开发工具包](#)
- [适用于 C++ 的Amazon开发工具包](#)

- [适用于 Go 的 Amazon 开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 JavaScript 的 Amazon 开发工具包](#)
- [适用于 PHP V3 的 Amazon 开发工具包](#)
- [适用于 Python 的 Amazon 开发工具包](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

数据类型

支持以下数据类型：

- [AvailabilityZone](#) (p. 659)
- [Certificate](#) (p. 660)
- [CloudwatchLogsExportConfiguration](#) (p. 662)
- [DBCluster](#) (p. 663)
- [DBClusterMember](#) (p. 668)
- [DBClusterParameterGroup](#) (p. 669)
- [DBClusterRole](#) (p. 670)
- [DBClusterSnapshot](#) (p. 671)
- [DBClusterSnapshotAttribute](#) (p. 674)
- [DBClusterSnapshotAttributesResult](#) (p. 675)
- [DBEngineVersion](#) (p. 676)
- [DBInstance](#) (p. 678)
- [DBInstanceStatusInfo](#) (p. 682)
- [DBSubnetGroup](#) (p. 683)
- [Endpoint](#) (p. 685)
- [EngineDefaults](#) (p. 686)
- [Event](#) (p. 687)
- [EventCategoriesMap](#) (p. 689)
- [EventSubscription](#) (p. 690)
- [Filter](#) (p. 692)
- [GlobalCluster](#) (p. 693)
- [GlobalClusterMember](#) (p. 695)
- [OrderableDBInstanceOption](#) (p. 696)
- [Parameter](#) (p. 698)
- [PendingCloudwatchLogsExports](#) (p. 700)
- [PendingMaintenanceAction](#) (p. 701)
- [PendingModifiedValues](#) (p. 703)
- [ResourcePendingMaintenanceActions](#) (p. 706)
- [Subnet](#) (p. 707)
- [Tag](#) (p. 708)
- [UpgradeTarget](#) (p. 709)
- [VpcSecurityGroupMembership](#) (p. 710)

AvailabilityZone

有关可用区的信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Name

可用区的名称。

类型: 字符串

: 必填项 : 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Certificate

证书颁发机构 (CA) 证书 Amazon Web Services 账户。

目录

Note

在下面的列表中，首先介绍了所需的参数。

CertificateArn

证书的 Amazon 资源名称 (ARN)。

示例：`arn:aws:rds:us-east-1::cert:rds-ca-2019`

类型: 字符串

: 必需 否

CertificateIdentifier

识别证书的唯一键。

示例：`rds-ca-2019`

类型: 字符串

: 必需 否

CertificateType

输入证书的类型。

示例：`CA`

类型: 字符串

: 必需 否

Thumbprint

证书的指纹。

类型: 字符串

: 必需 否

ValidFrom

证书有效的开始日期-时间。

示例：`2019-07-31T17:57:09Z`

类型: 时间戳

: 必需 否

ValidTill

证书不再有效的日期-时间。

示例：`2024-07-31T17:57:09Z`

类型: 时间戳

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java 2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

CloudwatchLogsExportConfiguration

要启用日志类型的配置设置，以便针对特定实例或集群导出到 Amazon CloudWatch Logs。

这些区域有：`EnableLogTypes`和`DisableLogTypes`数组确定要将哪些日志导出（或不导出）到 CloudWatch Logs。这些数组中的值取决于正在使用的引擎。

目录

Note

在下面的列表中，首先介绍了所需的参数。

`DisableLogTypes.member.N`

要禁用的日志类型列表。

类型: 字符串数组

必需 否

`EnableLogTypes.member.N`

要启用的日志类型的列表。

类型: 字符串数组

必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBCluster

有关集群的详细信息

目录

Note

在下面的列表中，首先描述了必需的参数。

AssociatedRoles.DBClusterRole.N

提供以下列表Amazon Identity and Access Management与集群关联的 (IAM) 角色。与集群关联的 (IAM) 角色授予集群访问其他集群的权限Amazon代表您提供的服务。

类型: 数组[DBClusterRole](#) (p. 670)对象

必填：否

AvailabilityZones.AvailabilityZone.N

提供可在其上创建集群中实例的 Amazon EC2 可用区的列表。

类型: 字符串数组

必填：否

BackupRetentionPeriod

指定自动快照的保留天数。

类型: 整数

必填：否

CloneGroupId

标识数据库集群与之关联的克隆组。

类型: 字符串

必填：否

ClusterCreateTime

指定集群的创建时间 (采用通用协调时间 (UTC))。

类型: 时间戳

必填：否

DBClusterArn

集群的 Amazon 资源名称 (ARN)。

类型: 字符串

必填：否

DBClusterIdentifier

包含用户提供的集群标识符。此标识符是识别集群的唯一键。

类型: 字符串

必填：否

DBClusterMembers.DBClusterMember.N

提供组成集群的实例列表。

类型: 数组[DBClusterMember \(p. 668\)](#)对象

必填：否

DBClusterParameterGroup

指定集群的集群的集群的集群的集群的集群的集群的集群

类型: 字符串

必填：否

DbClusterResourceId

这些区域有：Amazon Web Services 区域-唯一的不可变集群的唯一标识符。此标识符位于Amazon CloudTrail每当记录条目Amazon KMS集群的密钥被访问。

类型: 字符串

必填：否

DBSubnetGroup

指定与集群关联的子网组的信息，包括名称、描述和子网组中的子网。

类型: 字符串

必填：否

DeletionProtection

指定是否可以删除此集群。如果DeletionProtection已启用，除非集群被修改否则无法删除集群DeletionProtection已禁用。DeletionProtection防止意外删除集群。

类型: 布尔值

必填：否

EarliestRestorableTime

可以将数据库恢复到的最早时间 point-in-time还原。

类型: 时间戳

必填：否

EnabledCloudwatchLogsExports.member.N

此集群配置为导出到 Amazon 的日志类型列表 CloudWatch日志。

类型: 字符串数组

必填：否

Endpoint

指定集群的主实例。

类型: 字符串

必填：否

Engine

提供用于此集群的数据库引擎的数据库引擎的名称。

类型: 字符串

必填: 否

EngineVersion

指示数据库引擎版本。

类型: 字符串

必填: 否

HostedZoneId

指定在您创建托管区域时 Amazon Route 53 分配的 ID。

类型: 字符串

必填: 否

KmsKeyId

如果StorageEncrypted是true, Amazon KMS加密集群的密钥标识符。

类型: 字符串

必填: 否

LatestRestorableTime

指定数据库可以恢复到的最新时间 point-in-time还原。

类型: 时间戳

必填: 否

MasterUsername

包含集群的主用户名。

类型: 字符串

必填: 否

MultiAZ

指定集群是否在多个可用区中具有实例。

类型: 布尔值

必填: 否

PercentProgress

指定操作的进度百分比。

类型: 字符串

必填: 否

Port

指定数据库引擎侦听的端口。

类型: 整数

必填: 否

PreferredBackupWindow

指定在启用自动备份时, 自动执行备份的日常时间范围, 如 BackupRetentionPeriod 所规定。

类型: 字符串

必填: 否

PreferredMaintenanceWindow

指定可进行系统维护的每周时间范围 (采用通用协调时间 (UTC)) 。

类型: 字符串

必填: 否

ReaderEndpoint

集群的读取器端点。集群的读取器终端节点负载平衡集群中可用的 Amazon DocumentDB 副本之间的连接。当客户端请求与读取器终端节点的新连接时, Amazon DocumentDB 将在集群中的 Amazon DocumentDB 副本之间分配连接请求。该功能可帮助平衡跨集群中的多个 Amazon DocumentDB 副本的读取工作负载。

如果发生故障转移, 并且您连接的 Amazon DocumentDB 副本被提升为主实例, 则您的连接将断开。要继续将读取工作负载发送到集群中的其他 Amazon DocumentDB 副本, 您可以重新连接到读取器终端节点。

类型: 字符串

必填: 否

ReadReplicaIdentifiers.ReadReplicaIdentifier.N

包含与此集群关联的辅助群集的一个或多个标识符。

类型: 字符串数组

必填: 否

ReplicationSourceIdentifier

如果该集群是辅助集群, 则包含源集群的标识符。

类型: 字符串

必填: 否

Status

指定此集群的当前状态。

类型: 字符串

必填: 否

StorageEncrypted

指定集群是否已加密。

类型: 布尔值

必填: 否

VpcSecurityGroups.VpcSecurityGroupMembership.N

提供集群所属的Virtual Private Cloud (VPC) 安全组的列表。

类型: 数组[VpcSecurityGroupMembership \(p. 710\)](#)对象

必填：否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon SDK](#)
- [适用于 Go 的 Amazon SDK](#)
- [适用于 Java V2 的 Amazon SDK](#)
- [适用于 Ruby V3 的 Amazon SDK](#)

DBClusterMember

包含有关属于集群的实例的信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBClusterParameterGroupStatus

指定数据库集群此成员的集群参数组状态。

类型: 字符串

必填项: 否

DBInstanceIdentifier

指定集群此成员的实例标识符。

类型: 字符串

必填项: 否

IsClusterWriter

这是一个值 `true` 如果集群成员是集群的主实例，`false` 否则为。

类型: Boolean

必填项: 否

PromotionTier

该值指定在现有主实例发生故障后将 Amazon DocumentDB 副本提升为主实例的顺序。

类型: 整数

必填项: 否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBClusterParameterGroup

有关集群参数组的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBClusterParameterGroupArn

集群参数组的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 否

DBClusterParameterGroupName

提供集群参数组的名称。

类型: 字符串

: 必需 否

DBParameterGroupFamily

提供此集群参数组兼容的参数组系列的名称。

类型: 字符串

: 必需 否

Description

为此集群参数组提供客户指定的描述。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBClusterRole

描述Amazon Identity and Access Management(IAM) 角色，与集群关联。

目录

Note

在下面的列表中，首先介绍了所需的参数。

RoleArn

与数据库集群关联的 IAMRole 的 Amazon 资源名称 (ARN)。

类型: 字符串

必填项：否

Status

描述 IAMRole 与集群之间关联的状态。这些区域有：Status属性返回以下值之一：

- **ACTIVE**-该 iAMRole ARN 已与集群关联，可用于访问其他Amazon服务代表您。
- **PENDING**-该 iAMRole ARN 已与集群关联。
- **INVALID**-iAMrole ARN 与集群关联，但是集群无法承担 iAMrole 来访问其他Amazon服务代表您。

类型: 字符串

必填项：否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBClusterSnapshot

有关集群快照的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

AvailabilityZones.AvailabilityZone.N

提供集群快照中的实例可以还原到的 Amazon EC2 可用区的列表。

类型: 字符串数组

: 必需 否

ClusterCreateTime

指定创建集群的时间，以通用协调时 (UTC) 表示。

类型: 时间戳

: 必需 否

DBClusterIdentifier

指定创建此集群快照的集群的标识符。

类型: 字符串

: 必需 否

DBClusterSnapshotArn

集群快照的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 否

DBClusterSnapshotIdentifier

指定集群快照的标识符。

类型: 字符串

: 必需 否

Engine

指定数据库引擎的名称。

类型: 字符串

: 必需 否

EngineVersion

提供此集群快照的数据库引擎的版本。

类型: 字符串

: 必需 否

KmsKeyId

如果StorageEncrypted是true，Amazon KMS加密集群快照的密钥标识符。

类型: 字符串

: 必需 否

MasterUsername

提供集群快照的主用户名。

类型: 字符串

: 必需 否

PercentProgress

指定估计的已传输数据百分比。

类型: 整数

: 必需 否

Port

指定获取快照时集群侦听的端口。

类型: 整数

: 必需 否

SnapshotCreateTime

提供获取快照的时间，以 UTC 表示。

类型: 时间戳

: 必需 否

SnapshotType

提供集群快照的类型。

类型: 字符串

: 必需 否

SourceDBClusterSnapshotArn

如果集群快照复制自源集群快照，则为源集群快照的 ARN；否则为 null 值。

类型: 字符串

: 必需 否

Status

指定此集群快照的状态。

类型: 字符串

: 必需 否

StorageEncrypted

指定是否加密集群快照。

类型: Boolean

: 必需 否

VpcId

提供与集群快照关联的 Virtual Private Cloud (VPC) ID。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBClusterSnapshotAttribute

包含手动集群快照属性的名称和值。

手动集群快照属性用于授权其他快照属性Amazon Web Services 账户还原手动集群快照。

目录

Note

在下面的列表中，首先介绍了所需的参数。

AttributeName

手动集群快照属性的名称。

名为的属性`restore`是指的清单Amazon Web Services 账户有权复制或还原手动集群快照。

类型: 字符串

必填项：否

AttributeValues.AttributeValue.N

手动集群快照属性的值。

如果`AttributeName`字段设置为`restore`，然后此元素返回的 ID 列表Amazon Web Services 账户有权复制或还原手动集群快照。如果值为`all`在列表中，则手动集群快照为公有，可供任意Amazon Web Services 账户复制或恢复。

类型: 字符串数组

必填项：否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBClusterSnapshotAttributesResult

有关与集群快照关联的属性的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBClusterSnapshotAttributes.DBClusterSnapshotAttribute.N

集群快照的属性和值的列表。

类型: 数组的[DBClusterSnapshotAttribute](#) (p. 674)对象

: 必需 否

DBClusterSnapshotIdentifier

属性应用到的集群快照的标识符。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBEngineVersion

有关引擎版本的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBEngineDescription

数据库引擎的描述。

类型: 字符串

: 必需 否

DBEngineVersionDescription

数据库引擎版本的描述。

类型: 字符串

: 必需 否

DBParameterGroupFamily

要用于数据库引擎的参数组系列的名称。

类型: 字符串

: 必需 否

Engine

数据库引擎的名称。

类型: 字符串

: 必需 否

EngineVersion

数据库引擎的版本号。

类型: 字符串

: 必需 否

ExportableLogTypes.member.N

数据库引擎可用于导出到 Amazon CloudWatch Logs 的日志的类型。

类型: 字符串数组

: 必需 否

SupportsLogExportsToCloudwatchLogs

一个值，指示引擎版本是否支持导出ExportableLogTypes发布到 CloudWatch Logs。

类型: Boolean

: 必需 否

ValidUpgradeTarget.UpgradeTarget.N

此数据库引擎版本可以升级到的引擎版本列表。

类型: 数组[UpgradeTarget \(p. 709\)](#)对象

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBInstance

有关实例的详细信息。

目录

Note

在下面的列表中，首先描述了必需的参数。

AutoMinorVersionUpgrade

不适用。此参数不适用于 Amazon DocumentDB。无论设置的值如何，Amazon DocumentDB 都不会执行次要版本升级。

类型: 布尔值

必需 否

AvailabilityZone

指定实例所在的可用区名称。

类型: 字符串

必需 否

BackupRetentionPeriod

指定自动快照的保留天数。

类型: 整数

必需 否

CACertificateIdentifier

此数据库实例的 CA 证书的标识符。

类型: 字符串

必需 否

CopyTagsToSnapshot

指示是否将标签从数据库实例复制到数据库实例快照的值。默认情况下，不复制标签。

类型: 布尔值

必需 否

DBClusterIdentifier

如果实例是集群的成员，则包含该实例所属集群的名称。

类型: 字符串

必需 否

DBInstanceArn

实例的 Amazon 资源名称 (ARN) 。

类型: 字符串

必需 否

DBInstanceClass

包含实例的计算和内存容量类的名称。

类型: 字符串

必需 否

DBInstanceIdentifier

包含用户提供的数据库标识符。此标识符是识别实例的唯一键。

类型: 字符串

必需 否

DBInstanceStatus

指定此数据库的当前状态。

类型: 字符串

必需 否

DbiResourceId

这些区域有：Amazon Web Services 区域-实例的唯一不可变标识符。此标识符位于Amazon CloudTrail 每当日志条目Amazon KMS访问该实例的密钥。

类型: 字符串

必需 否

DBSubnetGroup

指定与实例关联的子网组的信息，包括名称、描述和子网组中的子网。

类型： [DBSubnetGroup \(p. 683\)](#) 对象

必需 否

EnabledCloudwatchLogsExports.member.N

此实例配置为导出到的日志类型列表 CloudWatch 日志。

类型: 字符串数组

必需 否

Endpoint

指定连接终端节点。

类型： [Endpoint \(p. 685\)](#) 对象

必需 否

Engine

提供用于此实例的数据库引擎的名称。

类型: 字符串

必需 否

EngineVersion

指示数据库引擎版本。

类型: 字符串

必需 否

InstanceCreateTime

提供创建实例的日期和时间。

类型: 时间戳

必需 否

KmsKeyId

如果StorageEncrypted是true , Amazon KMS加密实例的密钥标识符。

类型: 字符串

必需 否

LatestRestorableTime

指定数据库可以恢复到的最新时间 point-in-time还原。

类型: 时间戳

必需 否

PendingModifiedValues

指定对实例的更改处于待处理状态。只有在更改待处理时才包含此元素。特定更改由子元素标识。

类型 : [PendingModifiedValues \(p. 703\)](#) 对象

必需 否

PreferredBackupWindow

指定在启用自动备份时，自动执行备份的日常时间范围，如 BackupRetentionPeriod 所规定。

类型: 字符串

必需 否

PreferredMaintenanceWindow

指定可进行系统维护的每周时间范围（采用通用协调时间 (UTC) ）。

类型: 字符串

必需 否

PromotionTier

该值指定在现有主实例发生故障后将 Amazon DocumentDB 副本提升为主实例的顺序。

类型: 整数

必需 否

PubliclyAccessible

不支持。Amazon DocumentDB B 目前不支持公共终端节点。的值PubliclyAccessible总是false.

类型: 布尔值

必需 否

StatusInfos.DBInstanceStatusInfo.N

只读副本的状态。如果该实例不是只读副本，则为空。

类型: 数组 [DBInstanceStatusInfo \(p. 682\)](#) 对象

必需 否

StorageEncrypted

指定实例是否已加密。

类型: 布尔值

必需 否

VpcSecurityGroups.VpcSecurityGroupMembership.N

提供实例所属的 VPC 安全组元素的列表。

类型: 数组 [VpcSecurityGroupMembership \(p. 710\)](#) 对象

必需 否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon SDK](#)
- [适用于 Go 的 Amazon SDK](#)
- [适用于 Java V2 的 Amazon SDK](#)
- [适用于 Ruby V3 的 Amazon SDK](#)

DBInstanceStatusInfo

提供实例状态信息的列表。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Message

实例出现错误时，错误的详细信息。如果实例未处于出错状态，则此值为空白。

类型: 字符串

: 必需 否

Normal

布尔值是true如果实例运行正常，或者false实例处于错误状态时。

类型: Boolean

: 必需 否

Status

实例的状态。对于StatusType的只读副本，值可以是replicating错误消息，stopped，或者terminated。

类型: 字符串

: 必需 否

StatusType

此值当前是“read replication。”

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

DBSubnetGroup

子网组的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBSubnetGroupArn

数据库子网组的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 否

DBSubnetGroupDescription

提供子网组的描述。

类型: 字符串

: 必需 否

DBSubnetGroupName

子网组的名称。

类型: 字符串

: 必需 否

SubnetGroupStatus

提供子网组的状态。

类型: 字符串

: 必需 否

Subnets.Subnet.N

有关子网组中一个或多个子网的详细信息。

类型: 数组 [Subnet \(p. 707\)](#) 对象

: 必需 否

VpcId

提供子网组的 Virtual Private Cloud (VPC) ID。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)

- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Endpoint

用于访问群集或实例的网络信息。客户端程序必须指定有效的终端节点才能访问这些 Amazon DocumentDB 资源。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Address

指定实例的 DNS 地址。

类型: 字符串

必填项：否

HostedZoneId

指定在您创建托管区域时 Amazon Route 53 分配的 ID。

类型: 字符串

必填项：否

Port

指定数据库引擎侦听的端口。

类型: 整数

必填项：否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

EngineDefaults

包含成功调用的结果 `DescribeEngineDefaultClusterParameters` operation.

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBParameterGroupFamily

要返回其引擎参数信息的集群参数组系列的名称。

类型: 字符串

: 必需 否

Marker

由之前的请求提供的可选分页标记。如果指定此参数，则响应仅包含标记之外的记录，最大数量为 `MaxRecords` 指定的值。

类型: 字符串

: 必需 否

Parameters.Parameter.N

特定群集参数组系列的参数。

类型: 数组 [Parameter \(p. 698\)](#) 对象

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Event

有关活动的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Date

指定事件的日期和时间。

类型: 时间戳

: 必需 否

EventCategories.EventCategory.N

指定事件的类别。

类型: 字符串数组

: 必需 否

Message

提供此事件的文本。

类型: 字符串

: 必需 否

SourceArn

事件的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 否

SourceIdentifier

提供事件的源的标识符。

类型: 字符串

: 必需 否

SourceType

为此事件指定源类型。

类型: 字符串

有效值: db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

EventCategoriesMap

事件来源类型，附有一个或多个事件类别名称。

目录

Note

在下面的列表中，首先介绍了所需的参数。

EventCategories.EventCategory.N

指定源类型的事件类别。

类型: 字符串数组

必需 否

SourceType

返回的类别所属的源类型。

类型: 字符串

必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [Amazon SDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

EventSubscription

关于您订阅的事件的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

CustomerAwsId

这些区域有：Amazon与 Amazon DocumentDB 事件通知订阅关联的客户账户。

类型: 字符串

: 必需 否

CustSubscriptionId

Amazon DocumentDB 事件通知订阅 ID。

类型: 字符串

: 必需 否

Enabled

指示是否启用订阅的布尔值。值为true指示订阅已启用。

类型: Boolean

: 必需 否

EventCategoriesList.EventCategory.N

Amazon DocumentDB 事件通知订阅的事件类别列表。

类型: 字符串数组

: 必需 否

EventSubscriptionArn

事件订阅的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 否

SnsTopicArn

Amazon DocumentDB 事件通知订阅的主题 ARN。

类型: 字符串

: 必需 否

SourceIdsList.SourceId.N

Amazon DocumentDB 事件通知订阅的源 ID 列表。

类型: 字符串数组

: 必需 否

SourceType

Amazon DocumentDB 事件通知订阅的源类型。

类型: 字符串

: 必需 否

Status

Amazon DocumentDB 事件通知订阅的状态。

约束 :

可以是以下任一值 : `creating`、`modifying`、`deleting`、`active`、`no-permission`、`topic-not-exist`

这些区域有 : `no-permission`状态会表明 Amazon DocumentDB 不再有权发布到此 SNS 主题。这些区域有 : `topic-not-exist`状态会表明主题在创建订阅之后删除。

类型: 字符串

: 必需 否

SubscriptionCreationTime

创建 Amazon DocumentDB 事件通知订阅的时间。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Filter

一组命名的过滤器值，用于返回更具体的结果列表。您可以使用筛选器按特定条件（例如 ID）匹配一组资源。

筛选条件中不支持通配符。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Name

筛选器的名称。筛选条件名称区分大小写。

类型: 字符串

: 必填项: 是

Values.Value.N

一个或多个筛选条件值。筛选值区分大小写。

类型: 字符串数组

: 必填项: 是

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

GlobalCluster

表示 Amazon DocumentDB 全局集群的数据类型。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DatabaseName

新全局集群中的默认数据库名称。

类型: 字符串

: 必需 否

DeletionProtection

新全局集群的删除保护设置。

类型: Boolean

: 必需 否

Engine

全局集群使用的 Amazon DocumentDB 数据库引擎。

类型: 字符串

: 必需 否

EngineVersion

指示数据库引擎版本。

类型: 字符串

: 必需 否

GlobalClusterArn

全局集群的 Amazon 资源名称 (ARN)。

类型: 字符串

: 必需 否

GlobalClusterIdentifier

包含用户提供的全局集群标识符。此标识符是识别全局集群的唯一键。

类型: 字符串

长度约束: 最小长度为 1。长度上限为 255。

模式: `[A-Za-z][0-9A-Za-z-:._]*`

: 必需 否

GlobalClusterMembers.GlobalClusterMember.N

全局群集内辅助群集的群集 ID 列表。目前仅限于一件商品。

类型: 数组 [GlobalClusterMember](#) (p. 695)对象

: 必需 否

GlobalClusterResourceId

这些区域有: Amazon Web Services 区域-全局数据库集群唯一的不可变标识符。此标识符可在Amazon CloudTrail每当Amazon KMS访问集群的客户主密钥 (CMK)。

类型: 字符串

: 必需 否

Status

指定此全局集群的当前状态。

类型: 字符串

: 必需 否

StorageEncrypted

全局集群的存储加密设置。

类型: Boolean

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息, 请参阅以下内容:

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

GlobalClusterMember

一种数据结构，其中包含与 Amazon DocumentDB 全局群集关联的任何主群集和辅助群集的信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

DBClusterArn

每个 Amazon DocumentDB 集群的 Amazon 资源名称 (ARN)。

类型: 字符串

必填项：否

IsWriter

指定 Amazon DocumentDB 集群是否是与之关联的 Amazon DocumentDB 全局集群的主集群 (即具有读写功能)。

类型: Boolean

必填项：否

Readers.member.N

与 Aurora 全局集群关联的每个只读辅助集群的 Amazon 资源名称 (ARN)。

类型: 字符串数组

必填项：否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

OrderableDBInstanceOption

可用于实例的选项。

目录

Note

在下面的列表中，首先介绍了所需的参数。

AvailabilityZones.AvailabilityZone.N

实例的可用区域列表。

类型: 数组 [AvailabilityZone \(p. 659\)](#) 对象

: 必需 否

DBInstanceClass

实例的实例类。

类型: 字符串

: 必需 否

Engine

实例的引擎类型。

类型: 字符串

: 必需 否

EngineVersion

实例的引擎版本。

类型: 字符串

: 必需 否

LicenseModel

实例的许可模式。

类型: 字符串

: 必需 否

Vpc

指示实例是否位于 Virtual Private Cloud (VPC) 中。

类型: Boolean

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)

- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Parameter

有关单个参数的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

AllowedValues

指定参数值的有效范围。

类型: 字符串

: 必需 否

ApplyMethod

指示何时应用参数更新。

类型: 字符串

有效值: `immediate` | `pending-reboot`

: 必需 否

ApplyType

指定引擎特定的参数类型。

类型: 字符串

: 必需 否

DataType

指定参数的有效数据类型。

类型: 字符串

: 必需 否

Description

提供参数的说明。

类型: 字符串

: 必需 否

IsModifiable

指示参数可以修改 (`true`) 还是不能修改 (`false`)。一些参数具有安全或操作影响，会阻止更改这些参数。

类型: Boolean

: 必需 否

MinimumEngineVersion

参数可以应用到的最早引擎版本。

类型: 字符串

: 必需 否

ParameterName

指定参数的名称。

类型: 字符串

: 必需 否

ParameterValue

指定参数的值。

类型: 字符串

: 必需 否

Source

指示参数值的源。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

PendingCloudwatchLogsExports

其配置仍处于待处理状态的日志类型的列表。这些日志类型正处于激活或停用过程中。

目录

Note

在下面的列表中，首先介绍了所需的参数。

LogTypesToDisable.member.N

处于启用过程中的日志类型。启用之后，这些日志类型会导出到 Amazon CloudWatch Logs。

类型: 字符串数组

必填项: 否

LogTypesToEnable.member.N

处于停用过程中的日志类型。停用之后，这些日志类型不会导出到 CloudWatch Logs。

类型: 字符串数组

必填项: 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

PendingMaintenanceAction

提供有关资源的待处理维护操作的信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Action

可用于资源的待处理维护操作的类型。

类型: 字符串

: 必需 否

AutoAppliedAfterDate

应用操作时维护时段的日期。维护操作在此日期之后的第一个维护时段期间应用于资源。如果指定了此日期，则忽略任何 `next-maintenance` 加入请求。

类型: 时间戳

: 必需 否

CurrentApplyDate

待处理维护操作应用于资源的生效日期。

类型: 时间戳

: 必需 否

Description

提供有关维护操作的更多详细信息的描述。

类型: 字符串

: 必需 否

ForcedApplyDate

自动应用维护操作时的日期。无论资源的维护时段如何，维护操作都将在此日期应用于资源。如果指定了此日期，则忽略任何 `immediate` 加入请求。

类型: 时间戳

: 必需 否

OptInStatus

指示已为资源接收的加入请求的类型。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

PendingModifiedValues

实例的一个或多个修改过的设置。已请求这些修改后的设置，但尚未应用。

目录

Note

在下面的列表中，首先介绍了所需的参数。

AllocatedStorage

包含新的AllocatedStorage例如将要应用或当前正在应用的大小。

类型: 整数

: 必需 否

BackupRetentionPeriod

指定保留自动备份的挂起天数。

类型: 整数

: 必需 否

CACertificateIdentifier

指定数据库实例的证书颁发机构 (CA) 证书的标识符。

类型: 字符串

: 必需 否

DBInstanceClass

包含新的DBInstanceClass对于是要应用或当前正在应用的实例。

类型: 字符串

: 必需 否

DBInstanceIdentifier

包含新的DBInstanceIdentifier对于是要应用或当前正在应用的实例。

类型: 字符串

: 必需 否

DBSubnetGroupName

实例的新子网组。

类型: 字符串

: 必需 否

EngineVersion

指示数据库引擎版本。

类型: 字符串

: 必需 否

Iops

指定将要应用或当前正在应用的实例的新预配置 IOPS 值。

类型: 整数

: 必需 否

LicenseModel

实例的许可模式。

有效值: `license-included`、`bring-your-own-license`、`general-public-license`

类型: 字符串

: 必需 否

MasterUserPassword

包含实例的主凭证的挂起或当前正在进行的更改。

类型: 字符串

: 必需 否

MultiAZ

指示单可用区实例将更改为多可用区部署。

类型: Boolean

: 必需 否

PendingCloudwatchLogsExports

其配置仍处于待处理状态的日志类型的列表。这些日志类型正处于激活或停用过程中。

类型: [PendingCloudwatchLogsExports \(p. 700\)](#) 对象

: 必需 否

Port

指定实例的挂起端口。

类型: 整数

: 必需 否

StorageType

指定要与实例关联的存储类型。

类型: 字符串

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)

- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

ResourcePendingMaintenanceActions

表示的输出[ApplyPendingMaintenanceAction](#) (p. 535)。

目录

Note

在下面的列表中，首先介绍了所需的参数。

PendingMaintenanceActionDetails.PendingMaintenanceAction.N

一个列表，提供有关资源的待处理维护操作的详细信息。

类型: 数组[PendingMaintenanceAction](#) (p. 701)对象

必填项：否

ResourceIdentifier

挂起的维护操作的资源的 Amazon 资源名称 (ARN)。

类型: 字符串

必填项：否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Subnet

有关子网的详细信息。

目录

Note

在下面的列表中，首先介绍了所需的参数。

SubnetAvailabilityZone

指定子网的可用区。

类型：[AvailabilityZone \(p. 659\)](#) 对象

：必需 否

SubnetIdentifier

指定子网的标识符。

类型: 字符串

：必需 否

SubnetStatus

指定子网的状态。

类型: 字符串

：必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

Tag

分配给包含键值对的 Amazon DocumentDB 资源的元数据。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Key

标签必需的名称。该字符串值的长度可以在 1 到 128 个 Unicode 字符之间，并且不能带有前缀”。aws:“或者”rds:“。该字符串只能包含 Unicode 字母、数字、空格、“_”、“-”、“.”、“/”、“=”、“+”、“-” (Java 正则表达式：`^([\p{L}\p{N} _:/+=-]*)$`)。

类型: 字符串

必填项：否

Value

标签的可选内容。该字符串值的长度可以在 1 到 256 个 Unicode 字符之间，并且不能带有前缀”。aws:“或者”rds:“。该字符串只能包含 Unicode 字母、数字、空格、“_”、“-”、“.”、“/”、“=”、“+”、“-” (Java 正则表达式：`^([\p{L}\p{N} _:/+=-]*)$`)。

类型: 字符串

必填项：否

另请参阅

有关在特定语言的 Amazon 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 Amazon 开发工具包](#)
- [适用于 Go 的 Amazon 开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

UpgradeTarget

实例可以升级到的数据库引擎的版本。

目录

Note

在下面的列表中，首先介绍了所需的参数。

AutoUpgrade

一个值，指示目标版本是否应用于具有AutoMinorVersionUpgrade设置为true。

类型: Boolean

: 必需 否

Description

实例可以升级到的数据库引擎的版本。

类型: 字符串

: 必需 否

Engine

升级目标数据库引擎的名称。

类型: 字符串

: 必需 否

EngineVersion

升级目标数据库引擎的版本号。

类型: 字符串

: 必需 否

IsMajorVersionUpgrade

一个值，用于指示数据库引擎是否已升级到主要版本。

类型: Boolean

: 必需 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

VpcSecurityGroupMembership

用作 Virtual Private Cloud (VPC) 安全组成员资格查询的响应元素。

目录

Note

在下面的列表中，首先介绍了所需的参数。

Status

VPC 安全组的状态。

类型: 字符串

必填项: 否

VpcSecurityGroupId

VPC 安全组的名称。

类型: 字符串

必填项: 否

另请参阅

有关在特定语言的Amazon软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的Amazon开发工具包](#)
- [适用于 Go 的Amazon开发工具包](#)
- [AmazonSDK for Java V2](#)
- [适用于 Ruby V3 的 Amazon 开发工具包](#)

常见错误

本部分列出了所有 API 操作常见的错误。Amazon服务。对于特定于此服务的 API 操作的错误，请参阅该 API 操作的主题。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

IncompleteSignature

请求签名不符合Amazon标准。

HTTP 状态代码：400

InternalFailure

由于未知错误、异常或故障，请求处理失败。

HTTP 状态代码：500

InvalidAction

所请求的操作无效。验证操作是否已正确键入。

HTTP 状态代码：400

InvalidClientTokenId

在我们的记录中没有所提供的 X.509 证书或 Amazon 访问密钥 ID。

HTTP 状态代码：403

InvalidParameterCombination

不得共用的参数被一起使用。

HTTP 状态代码：400

InvalidParameterValue

为输入参数提供的值无效或超出范围。

HTTP 状态代码：400

InvalidQueryParameter

这些区域有：Amazon 查询字符串格式错误或未遵循 Amazon 标准。

HTTP 状态代码：400

MalformedQueryString

查询字符串包含语法错误。

HTTP 状态代码：404

MissingAction

请求中遗漏了一个操作或必需参数。

HTTP 状态代码：400

MissingAuthenticationToken

请求中必须包含有效的 (已注册的) Amazon 访问密钥 ID 或 X.509 证书。

HTTP 状态代码：403

MissingParameter

未提供用于指定操作的必需参数。

HTTP 状态代码：400

NotAuthorized

您没有执行此操作的权限。

HTTP 状态代码：400

OptInRequired

Amazon 访问密钥 ID 需要订阅服务。

HTTP 状态代码：403

RequestExpired

请求到达服务的时间超过请求上的日期戳或请求到期日期 (如针对预签名 URL) 15 分钟，或者请求上的日期戳离到期还有 15 分钟以上。

HTTP 状态代码：400

ServiceUnavailable

由于服务器发生临时故障而导致请求失败。

HTTP 状态代码：503

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

ValidationError

输入未能满足由Amazon服务。

HTTP 状态代码：400

常见参数

以下列表包含所有操作用于使用查询字符串对签名版本 4 请求进行签名的参数。所有操作特有的参数都列在该操作的主题中。有关签名版本 4 的更多信息，请参阅[签名版本 4 签名流程](#)中的Amazon Web Services 一般参考。

Action

要执行的操作。

类型：字符串

必填项：是

Version

请求所针对的 API 版本，格式为 YYYY-MM-DD 表示。

类型：字符串

必填项：是

X-Amz-Algorithm

用于创建请求签名的哈希算法。

条件：在查询字符串而不是 HTTP 授权标头中包含身份验证信息时，请指定此参数。

类型：字符串

有效值：AWS4-HMAC-SHA256

必填项：条件

X-Amz-Credential

凭证范围值，该值是一个字符串，其中包含您的访问密钥、日期、目标区域、所请求的服务和终止字符串（“aws4_request”）。此值用以下格式表示：access_key/YYYYMMDD/领域/服务/aws4_ 请求。

有关更多信息，请参阅。[任务 2：为签名版本 4 创建要签名的字符串](#)中的Amazon Web Services 一般参考。

条件：在查询字符串而不是 HTTP 授权标头中包含身份验证信息时，请指定此参数。

类型：字符串

必填项：条件

X-Amz-Date

用于创建签名的日期。格式必须为 ISO 8601 基本格式 (YYYYMMDD'THHMMSS'Z')。例如，以下日期时间是有效的 X-Amz Date 值：20120325T120000Z。

条件：X-Amz Date 对所有请求而言是可选的；它可以用于覆盖对请求签名所使用的日期。如果使用 ISO 8601 基本格式指定了日期标头，则无需 X-Amz Date。使用 X-Amz Date 时，它始终会覆盖日期标头的值。有关更多信息，请参阅 [处理签名版本 4 中的日期](#) 中的 Amazon Web Services 一般参考。

类型：字符串

必填项：条件

X-Amz-Security-Token

通过致电获得的临时安全令牌(Amazon Security Token Service Amazon STS)。有关支持临时安全证书的服务列表，Amazon 请转到 [Security Token Service Amazon 与 IAM 结合使用的服务](#) 中的 IAM 用户指南。

条件：如果您使用的临时安全凭证从 Amazon 安全令牌服务，您必须包括安全令牌。

类型：字符串

必填项：条件

X-Amz-Signature

指定从待签字符串和派生的签名密钥计算的十六进制编码签名。

条件：在查询字符串而不是 HTTP 授权标头中包含身份验证信息时，请指定此参数。

类型：字符串

必填项：条件

X-Amz-SignedHeaders

指定作为规范请求一部分包含的所有 HTTP 标头。有关指定签名标头的更多信息，请参阅 [任务 1：创建签名版本 4 的规范请求](#) 中的 Amazon Web Services 一般参考。

条件：在查询字符串而不是 HTTP 授权标头中包含身份验证信息时，请指定此参数。

类型：字符串

必填项：条件

发行说明

按发布日期描述 Amazon DocumentDB (MongoDB 兼容性) 功能、改进和错误修复。随着 2020 年 11 月 11 日 MongoDB 4.0 兼容性的推出，发行说明将包括 Amazon DocumentDB 3.6 和 4.0 引擎版本的更新。

您可以通过运行以下命令来确定当前 Amazon DocumentDB 引擎版本：

```
db.runCommand({getEngineVersion: 1})
```

如果您的集群不在最新版本的引擎上，则可能您有可能会升级引擎的待处理维护。有关更多信息，请参阅 [维护 Amazon DocumentDB \(p. 356\)](#)。

主题

- [2022 年 1 月 21 日 \(p. 714\)](#)
- [2021 年 10 月 25 日 \(p. 714\)](#)
- [2021 年 6 月 24 日 \(p. 715\)](#)
- [2021 年 5 月 4 日 \(p. 715\)](#)
- [2021 年 1 月 15 日 \(p. 716\)](#)
- [2020 年 11 月 9 日 \(p. 716\)](#)
- [2020 年 10 月 30 日 \(p. 718\)](#)
- [2020 年 9 月 22 日 \(p. 718\)](#)
- [2020 年 7 月 10 日 \(p. 719\)](#)
- [2020 年 6 月 30 日 \(p. 719\)](#)

2022 年 1 月 21 日

新功能

Amazon DocumentDB 4.0 (引擎版本 2.0.5706)

- 现在支持 Amazon DocumentDB Graviton2 (r6g.large、r6g.2xlarge、r6g.4xlarge、r6g.8xlarge、r6g.12xlarge、r6g.16xlarge 和 t4g. 中) 实例现在受支持

Amazon DocumentDB 3.6 (引擎版本 1.0.207781) 和 4.0 (引擎版本 2.0.5706)

- 增加了对以下 MongoDB API 的支持
 - \$reduce
 - \$mergeObjects
 - \$geoWithin
 - \$geoIntersects

2021 年 10 月 25 日

新功能

Amazon DocumentDB 3.6 (引擎版本 1.0.207780) 和 4.0 (引擎版本 2.0.5704)

- 增加了对以下 MongoDB API 的支持
 - `$literal`
 - `$map`
 - `$$ROOT`
- 对该项的支持 GeoSpatial 查询功能。请参阅此[博客帖子](#)有关更多详情
- Support 用户定义的角色进行访问控制。请参阅此[博客帖子](#)有关更多详情
- Amazon DocumentDB JDBC 驱动程序，可通过 BI 工具（如 Tableau）和 SQL Workbench 等查询工具启用连接

错误修复和其他更改

Amazon DocumentDB 3.6 (引擎版本 1.0.207780) 和 4.0 (引擎版本 2.0.5704)

- 错误修复 `$natural` 在明确的情况下正确排序 `.sort()` 同时存在 `$natural`
- 修复了更改直播使用的错误 `$redact`
- 错误修复 `$ifNull` 处理空数组
- 当删除当前登录用户或取消该用户对持续活动的权限时，资源消耗过多/服务器崩溃的错误修复
- 错误修复 `listDatabase` 和 `listCollection` 权限检查
- Bug 修复多键元素的重复数据删除逻辑

2021 年 6 月 24 日

新功能

Amazon DocumentDB 3.6 (引擎版本 1.0.207117) 和 4.0 (引擎版本 2.0.3371)

- `r5.8xlarge` and `r5.16xlarge` instances are now supported. Learn more at the blog post [Amazon DocumentDB 现在支持 r5.8xlarge 和 r5.16xlarge 实例](#).
- [全球集群](#) are now supported to provide disaster recovery from region-wide outages and enable low-latency global reads by allowing reads from the nearest Amazon DocumentDB cluster.

2021 年 5 月 4 日

新功能

查看此中的所有新功能[博客帖子](#).

Amazon DocumentDB 3.6 (引擎版本 1.0.207117) 和 4.0 (引擎版本 2.0.3371)

- `renameCollection`
- `$zip`
- `$indexOfArray`
- `$reverseArray`
- `$ATUNE`
- `$$$` support for update

- Index scan for ##

错误修复和其他更改

Amazon DocumentDB 3.6 (引擎版本 1.0.207117) 和 4.0 (引擎版本 2.0.3371)

- Reduced memory usage for \$in queries
- Fixed a memory leak in multikey indexes
- Fixed the explain plan and profiler output for \$out
- Added a timeout for operations from internal monitoring system to improve reliability
- Fixed a defect impacting the query predicates passed to multikey indexes

2021 年 1 月 15 日

新功能

Amazon DocumentDB 4.0 (引擎版本 2.0.722)

- None

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- Ability to use an index with the \$lookup aggregation stage
- find () queries with projections can be served direction from an index (covered query)
- Ability to use ## () with the findAndModify
- Performance optimizations for \$addToSet operator
- Improvements to reduce overall index sizes
- New aggregation operators: \$ifNull, \$replaceRoot, \$setIsSubset, \$setInster##, \$setUnion, and \$setEquals
- Users can also end their own cursors without requiring the killCursor role

2020 年 11 月 9 日

新功能

请参阅中的所有新功能[此博客帖子](#)。

Amazon DocumentDB 4.0 (引擎版本 2.0.722)

- MongoDB 4.0 compatibility
- ACID transactions
- Support for ###client.watch () or mongo.watch () and the database level (db.watch ()) change streams
- Ability to start or resume a change streams using #####
- Extend your change stream retention period to 7 days (previously 24 hours)

- Amazon DMS target for Amazon DocumentDB 4.0
- CloudWatch metrics: `####`, `## OpenMAX`, `#####`, `#####`, and `#####`
- New fields for transactions in `currentOp`, `###Status`, and `profiler`.
- Ability to use an index with the `$lookup` aggregation stage
- `find ( )` queries with projections can be served direction from an index (covered query)
- Ability to use `## ( )` with the `findAndModify`
- Performance optimizations for `$addToSet` operator
- Improvements to reduce overall index sizes.
- New aggregation operators: `$ifNull`, `$replaceRoot`, `$setIsSubset`, `$setInster##`, `$setUnion`, and `$setEquals`
- With the `ListCollection` and `listDatabase` commands, you can now optionally use the `####` and `#####` parameters to allow users to list the collections and databases that they have permission to access without requiring the `listCollections` and `listDatabase` roles, respectively
- Users can also end their own cursors without requiring the `killCursor` role
- Comparing numeric types of subdocuments is now consistent with comparing numeric types of first-level documents. The behavior in Amazon DocumentDB 4.0 is now compatible with MongoDB.

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- None

错误修复和其他更改

Amazon DocumentDB 4.0 (引擎版本 2.0.722)

- `$setOnInsert` no longer allow updates when using the positional operator `$`. The behavior in Amazon DocumentDB 4.0 is now compatible with MongoDB.
- Fixed issue with `$Create Collection` and set `#### ID`
- Projection for nested documents
- Changed default setting for working memory to scale with instance memory size
- Garbage collection improvements
- Lookup with empty key in path, behavior difference with mongo
- Fixed `DateToString` bug in timezone behavior
- Fixed `$push` (aggregation) to respect sort order
- Fixed bug in `$currentOp` with aggregate
- Fixed issue with `readPreference` on secondary
- Fixed issue with validating `$recreateIndex` is the same database as the command was issued
- Fixed inconsistent behavior for `MinKey`, `MaxKey` lookup fails
- Fixed issue with `$size` operator not working with composite array
- Fixed issue with the negation of `$in` with regex
- Fixed issue with `$ DISTINCT` command run against a view
- Fixed issue with aggregations and find commands sorting missing fields differently
- Fixed `$eq` to regular expression not checking type
- Fixed `$currentDate` bug in timestamp ordinal position behavior
- Fixed millisecond granularity for `$currentDate`

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- None

2020 年 10 月 30 日

新功能

查看中的所有新功能[此博客帖子](#)。

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- Added the ability to open a change stream cursor at the cluster level `####.watch ()` or `mongo.watch ()` and the database (`db.watch ()`)
- Ability to increase the change stream retention period to 7 days (previously 24 hours)

错误修复和其他更改

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- Various general case performance improvements
- A targeted security improvement
- Fixed an issue with skip sort on second field of a compound index
- Enable regular index for equality on single field of a multi-key index (not compound)
- Fixed authentication race condition
- Fixed issue that caused an infrequent garbage collection crash
- RBAC security improvement
- Added `#####` metric
- Performance improvements for certain workloads on `r5.24xlarge` instances

2020 年 9 月 22 日

新功能

请参阅中的所有新功能[此博客帖子](#)。

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- `$out` aggregation stage
- Increased the maximum number of connections and cursor per instance by as much as 10x

错误修复和其他更改

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- None

2020 年 7 月 10 日

新功能

查看中的所有新功能[此博客帖子](#).

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- Cross Region Snapshot Copy

错误修复和其他更改

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- None

2020 年 6 月 30 日

新功能

查看中的所有新功能[此博客帖子](#).

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- T3 medium instances

错误修复和其他更改

Amazon DocumentDB 3.6 (引擎版本 1.0.206295)

- Idle memory reclamation for t3 instances
- Authentication improvements
- Improved SASL authentication performance
- Fixed `currentOp` issue when exceeding maximum possible ops
- Fixed `killOps` issue for bulk update and delete
- Improvements to `$sample` performance with `$match`
- Fixed support for `$$` in cond case in redact stage
- Fixed various recurring crash root causes
- Improvements to TTL sweeping to reduce IOs and latency
- Optimized memory utilization for `$unwind`
- Fixed collection stats race condition with drop index
- Fixed race condition during concurrent index build
- Fixed infrequent crash in `hash_search` in index

Amazon DocumentDB 开发人员指南的文档历史记录

- API 版本：2014-10-31
- 最近文档更新时间：2020 年 11 月 11 日

下表介绍了此版本的文档。Amazon DocumentDB 开发人员指南。

update-history-change	update-history-description	update-history-date
全球群集	添加了有关如何使用全局群集的文档。	2021 年 6 月 2 日
事件订阅	添加了活动订阅文档。	2021 年 3 月 26 日
版本 3.6 升级	记录了 3.6 版在基于角色的访问控制、聚合运算符和性能方面的改进。	2021 年 1 月 15 日
MongoDB 4.0 兼容	Amazon DocumentDB 现在与 MongoDB 的 4.0 版本兼容。	2020 年 11 月 9 日
入门指南	使用 Amazon EC2、Robo3T 或 Studio3T 开始使用 Amazon DocumentDB 的新入门指南。	2020 年 8 月 15 日
支持其他可用区	Amazon DocumentDB 添加了对亚太地区 (首尔) (ap-northeast-2) 的支持。	2020 年 7 月 14 日
添加了对跨区域复制快照的支持。	Amazon DocumentDB 增加了对跨复制群集快照的支持。Amazon Web Services 区域。有关更多信息，请参阅 跨区域复制快照 。	2020 年 7 月 10 日
添加了对 T3 实例类的支持。	在支持 Amazon DocumentDB 的所有区域中添加了对 T3 实例类型的支持。有关更多信息，请参阅 不同区域支持的实例类和实例类规格 。	2020 年 6 月 30 日
添加了对的支持Amazon GovCloud (US)。	目前，Amazon DocumentDB 可在 Amazon GovCloud (US) 区域 (us-gov-west-1)。	2020 年 6 月 29 日
添加了 16 个新的 CloudWatch 指标。	Amazon DocumentDB 增加了对 16 个新的 Amazon CloudWatch 指标的支持。有关更多信息，请参阅 使用 CloudWatch 监控 Amazon DocumentDB 。	2020 年 6 月 23 日
添加了对 null 字符和 \$regex 运算符的支持。	Amazon DocumentDB 添加了对字符串中的 null 字符的支持，并能够为 \$regex 使用索引。要查看	2020 年 6 月 22 日

	支持的 MongoDB API 和聚合管道功能，请参阅 与 MongoDB 之间的功能差异 。	
增加了对改进的多键索引功能的支持。	Amazon DocumentDB 增加了对改进的多键索引功能的支持，这些功能包括对大于 2048 个字节的数组编制索引的功能和使用相同数组中的多个键创建复合多键索引的能力。有关更多信息，请参阅 与 MongoDB 之间的功能差异 。	2020 年 4 月 23 日
添加了对 Amazon DocumentDB 的删除保护的支持Amazon CloudFormation堆栈。	Amazon DocumentDB 添加了对在创建 Amazon DocumentDB 时启用删除保护的支持。Amazon CloudFormation堆栈。	2020 年 4 月 20 日
增加了对基于角色的访问控制的支持。	Amazon DocumentDB 添加了对使用内置角色进行访问控制的支持。	2020 年 3 月 26 日
增加了对在加拿大 (中部) (ca-central-1) 中添加了对其他可用区的支持。	现在 Amazon DocumentDB 在具有 R5 类实例和 3 个可用区的加拿大 (中部) 区域 (ca-central-1) 中可用。	2020 年 3 月 26 日
增加了对两个额外的 MongoDB API 的支持	Amazon DocumentDB 添加了对的支持\$dateFromString和executionStatsMongoDB API。	2020 年 3 月 23 日
增加了对五个额外的 MongoDB API 的支持。	Amazon DocumentDB 添加了对的支持\$arrayToObjectId、\$slice、\$mod、和\$rangeMongoDB API。	2020 年 2 月 6 日
增加了对加拿大 (Central) 的支持。	现在，Amazon DocumentDB 在具有 R5 类实例的加拿大 (中部) 区域 (ca-central-1) 中可用。	2019 年 12 月 11 日
增加了对 ChangeStreamLogSize 的支持。	Amazon DocumentDB 添加了对的支持ChangeStreamLogSize对于 Cloudwatch 指标。	2019 年 11 月 22 日
增加了对欧洲 (巴黎) 区域的支持	现在，Amazon DocumentDB 在欧洲 (巴黎) 区域 (eu-west-3) 中配合 R5 类实例使用。	2019 年 10 月 30 日
增加了对亚太区域 (孟买) 区域的支持	Amazon DocumentDB 现在，在具有 R5 类实例的亚太区域 (ap-south-1) 中可用。	2019 年 10 月 17 日
增加了对三个额外的 MongoDB API 的支持	Amazon DocumentDB 添加了对的支持\$updateFields、\$concatArrays、和\$lookupMongoDB API。	2019 年 10 月 16 日

添加了对亚太地区（新加坡）区域的支持	Amazon DocumentDB 现在，在具有 R5 类实例的亚太区域（新加坡）中可用。	2019 年 10 月 14 日
增加了有关更新 TLS 证书的新文档	增加了有关更新 CA 证书以使用新的 CA 证书来创建 TLS 连接的说明。	2019 年 10 月 2 日
增加了对证书的 API 支持	Amazon DocumentDB 是用于实例的新证书数据类型。有关更多信息，请参阅 DBInstance 。	2019 年 10 月 1 日
支持查询分析	Amazon DocumentDB 添加了对集群的实例和数据库上支持的操作进行分析的功能。	2019 年 8 月 19 日
在亚太地区（东京）增加了第三个 AZ	Amazon DocumentDB 为您在亚太地区（东京）中的计算实例添加了第三个可用区 (AZ)。	2019 年 8 月 9 日
支持其他 Mongo API	添加了对其他聚合管道功能的支持，其中包括 <code>\$in</code> 、 <code>\$isoWeek</code> 、 <code>\$isoWeekYear</code> 、 <code>\$isoDayOfWeek</code> ，和 <code>\$dateToString</code> 聚合运算符和 <code>\$addToSet</code> 聚合阶段。Amazon DocumentDB 还添加了对的支持 <code>top()</code> 用于集合级别诊断的命令以及修改 <code>expireAfterSecondsTTL</code> 索引的参数使用 <code>collMod()</code> 命令。	2019 年 7 月 31 日
增加了对欧洲（伦敦）的支持	现在，Amazon DocumentDB 在欧洲（伦敦）中配合 R5 类实例使用。	2019 年 7 月 18 日
添加了代码示例	在 R 和 Ruby 中添加了代码示例，用于以编程方式连接到 Amazon DocumentDB。	2019 年 7 月 17 日
添加了最佳实践	添加了最佳实践，帮助您管理 Amazon DocumentDB 成本。	2019 年 7 月 17 日
支持停止和启动集群	Amazon DocumentDB 增加了对停止和启动集群以控制开发和测试环境成本的支持。	2019 年 7 月 1 日
支持集群删除保护	为了防止您的集群遭到意外删除，Amazon DocumentDB 添加了删除保护功能。有关更多信息，请参阅以下主题： 创建 Amazon DocumentDB 集群 、 修改 Amazon DocumentDB 集群 、 删除 Amazon DocumentDB 集群 ，和 <code>DeletionProtection</code> 在 API 主题中 DBCluster 。	2019 年 7 月 1 日
功能差异更新	向功能差异添加了隐式事务。	2019 年 6 月 12 日

功能差异添加	添加了有关 Amazon DocumentDB 中的存储和索引压缩的注释。	2019 年 6 月 13 日
增加了支持区域	Amazon DocumentDB 现在，在具有 R5 类实例的亚太地区（悉尼）中可用。	2019 年 6 月 5 日
更多区域支持 R5 实例类	为 4 个额外区域添加了 R5 实例类支持：美国东部（俄亥俄）、美国西部（俄勒冈）和欧洲（爱尔兰）。通过此次更改，支持 Amazon DocumentDB 的所有区域都支持 R5 实例。	2019 年 5 月 17 日
支持的其他区域	使用 R5 实例类添加了对 2 个其他区域的支持：亚太区域（东京）（ap-northeast-1）和亚太区域（首尔）（ap-northeast-2）。有关更多信息，请参阅 按区域支持的实例类 和 实例类规格 。	2019 年 5 月 8 日
添加了更多连接代码示例	添加了 Java 和 C# 代码示例，用于连接到 Amazon DocumentDB。	2019 年 4 月 24 日
其他 Mongo API 支持	添加了对七个聚合字符串运算符（\$indexOfBytes、\$indexOfCP、\$strLenBytes、\$strLenCP、\$toLower、和 \$split）、九个日期时间运算符（\$dayOfYear、\$dayOfMonth、\$dayOfWeek、\$year、\$month、\$hour、\$minute 和 \$millisecond）和 \$sample 聚合管道分段的支持。	2019 年 4 月 4 日
添加了连接代码示例	添加了 Python、Node.js 和 Go 的代码示例，用于连接到 Amazon DocumentDB。	2019 年 3 月 21 日
对法兰克福区域和 R5 实例的支持	添加了对欧洲（法兰克福）区域（eu-central-1）与 R5 实例类的支持。有关更多信息，请参阅 按区域支持的实例类 和 实例类规格 。	2019 年 3 月 13 日
聚合管道运算符支持	添加了对新的聚合字符串运算符（\$concat、\$substr、\$substrBytes、\$substrCP、\$strcasecmp）、数组聚合运算符（\$size）、聚合组累加器运算符（\$push）和聚合阶段（\$redact 和 \$indexStats）的支持。我们还添加了对位置数组运算符（\$[] 和 \$[<identifier>]）和 hint() 的支持。	2019 年 2 月 28 日
引擎升级	添加了确定待处理的集群修改和升级集群引擎版本的文档。	2019 年 2 月 15 日
审核事件	添加了对使用 Amazon CloudWatch Logs 审核数据库事件的支持。	2019 年 2 月 12 日

Quick Start (快速入门)	添加了“快速入门”主题，帮助您通过轻松开始使用 Amazon DocumentDBAmazon CloudFormation.	2019 年 1 月 11 日
公开发布版 (p. 720)	这是 Amazon DocumentDB 的最初公开版本 (与 MongoDB 兼容)。本版本包括 开发人员指南 和集成的 资源管理 API 参考 。	2019 年 1 月 9 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。