

文档数据库服务

## 常见问题

文档版本 01  
发布日期 2025-09-10



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 目录

|                                                 |           |
|-------------------------------------------------|-----------|
| <b>1 产品咨询</b>                                   | <b>1</b>  |
| 1.1 DDS 和社区版 MongoDB 有什么关系                      | 1         |
| 1.2 DDS 4.2 及其以上版本切 RocksDB 存储引擎相关问题 Q&A        | 1         |
| 1.3 使用文档数据库服务要注意什么                              | 3         |
| 1.4 什么是文档数据库实例可用性                               | 4         |
| 1.5 我的实例是否会受其他用户实例影响                            | 4         |
| 1.6 DDS 是否支持多可用区部署                              | 4         |
| 1.7 实例创建成功后是否支持更换 VPC                           | 4         |
| 1.8 实例创建成功后是否支持更换区域                             | 4         |
| 1.9 什么是 Hidden 节点                               | 4         |
| 1.10 什么是逻辑会话 ( Logical Sessions )               | 4         |
| <b>2 数据库版本</b>                                  | <b>7</b>  |
| 2.1 DDS 是否支持版本升级                                | 7         |
| <b>3 资源冻结/释放/删除/退订</b>                          | <b>8</b>  |
| <b>4 资源及磁盘管理</b>                                | <b>10</b> |
| 4.1 哪些数据会占用文档数据库实例空间                            | 10        |
| 4.2 占用文档数据库服务磁盘空间的日志及文件有哪些                      | 10        |
| 4.3 为什么界面上查看的磁盘使用空间比实际的使用量小                     | 10        |
| 4.4 数据删了磁盘空间但没有减少                               | 11        |
| 4.5 购买的副本集实例是 4 核 8GB 的，为什么驻留内存只有 4GB 左右而不是 8GB | 11        |
| <b>5 扩容及规格变更</b>                                | <b>12</b> |
| 5.1 DDS 是否支持不同类型实例变更                            | 12        |
| 5.2 DDS 是否支持添加节点                                | 12        |
| 5.3 DDS 在分片节点扩容期间服务是否可用                         | 12        |
| <b>6 数据库性能</b>                                  | <b>13</b> |
| 6.1 集群和副本集在什么情况会触发主备切换                          | 13        |
| 6.2 磁盘使用率高问题排查                                  | 14        |
| 6.3 副本集中主备同步存在多长时间的延迟                           | 15        |
| 6.4 副本集主备节点之间的数据同步方式是什么                         | 16        |
| 6.5 分片内存超过 90%产生告警，如何消减                         | 16        |
| 6.6 DDS 集群持续写入数据后查询报错怎么处理                       | 16        |

|                                   |           |
|-----------------------------------|-----------|
| <b>7 数据库权限</b>                    | <b>17</b> |
| 7.1 实例创建时指定的 root 账号拥有什么权限？       | 17        |
| 7.2 默认权限机制                        | 17        |
| 7.3 角色管理                          | 17        |
| 7.3.1 预置角色                        | 18        |
| 7.3.2 自定义角色                       | 20        |
| 7.3.3 创建并管理角色                     | 20        |
| 7.4 用户管理                          | 22        |
| 7.4.1 创建用户                        | 23        |
| 7.4.2 更新用户                        | 26        |
| 7.4.3 删除用户                        | 28        |
| <b>8 创建与删除</b>                    | <b>30</b> |
| 8.1 创建 DDS 实例时，如何选择规格和节点          | 30        |
| 8.2 实例创建成功后，为什么在控制台看不到            | 30        |
| 8.3 可以通过创建模板的方式创建实例吗              | 31        |
| 8.4 为什么我的数据会丢失或被删除                | 31        |
| 8.5 如果删除我的云账号是否会删除备份              | 31        |
| 8.6 删除和退订实例的区别是什么                 | 31        |
| <b>9 数据库连接</b>                    | <b>32</b> |
| 9.1 DDS 实例连接失败怎么办                 | 32        |
| 9.2 实例的连接数满导致实例连接失败，如何处理？         | 39        |
| 9.3 如何查询及限制连接数                    | 41        |
| 9.4 ECS 和 DDS 署在不同的 VPC，网络不通怎么办   | 43        |
| 9.5 应用程序是否需要支持自动重连数据库             | 43        |
| 9.6 如何创建并登录弹性云服务器                 | 43        |
| <b>10 客户端安装</b>                   | <b>44</b> |
| 10.1 如何安装 MongoDB 客户端             | 44        |
| 10.2 如何安装 Robo 3T 工具              | 45        |
| <b>11 数据库基本使用</b>                 | <b>49</b> |
| 11.1 如何查看当前副本集实例的主备节点             | 49        |
| 11.2 如何查询每个表的数据量 mongodb 存储空间分析？  | 50        |
| 11.3 文档数据库中创建失败的索引可以删除吗           | 50        |
| 11.4 DDS 是否支持 TTL 索引清理过期数据        | 50        |
| 11.5 如何使用 MapReduce 命令            | 50        |
| 11.6 DDS 是否支持 \$round 函数          | 51        |
| 11.7 如何管理均衡器 Balancer             | 52        |
| <b>12 数据库迁移</b>                   | <b>54</b> |
| 12.1 DDS 实例是否支持跨 region 迁移        | 54        |
| 12.2 不同账号下的 DDS 数据库如何进行迁移？        | 54        |
| 12.3 高版本 MongoDB 迁移至低版本时，兼容度如何评估？ | 54        |

|                                          |           |
|------------------------------------------|-----------|
| <b>13 数据库存储</b>                          | <b>55</b> |
| 13.1 DDS 使用的存储配置是什么                      | 55        |
| 13.2 数据超过了文档数据库实例的最大存储容量怎么办              | 55        |
| 13.3 什么是 DDS 实例的只读状态?                    | 55        |
| <b>14 数据库参数</b>                          | <b>56</b> |
| 14.1 DDS 支持修改时区吗                         | 56        |
| 14.2 用户需要关注实例的哪些参数                       | 56        |
| 14.3 MapReduce 提示不支持 JavaScript          | 56        |
| 14.4 DDS 是否支持设置 majority 级别的 readConcern | 56        |
| <b>15 备份与恢复</b>                          | <b>58</b> |
| 15.1 DDS 实例删除后, 手动备份是否继续收费               | 58        |
| 15.2 如何将文档数据库备份到弹性云服务器上                  | 58        |
| 15.3 文档数据库服务能够保存多长时间的备份                  | 58        |
| 15.4 如何找回 DDS 丢失的备份数据                    | 58        |
| 15.5 备份空间里面包含哪些文件                        | 58        |
| 15.6 DDS 的备份是如何收费的                       | 59        |
| <b>16 网络安全</b>                           | <b>60</b> |
| 16.1 文档数据库服务有哪些安全保障措施                    | 60        |
| 16.2 为什么在虚拟私有云中使用文档数据库                   | 60        |
| 16.3 如何确保在虚拟私有云中运行的文档数据库的安全              | 60        |
| 16.4 DDS 是否支持选择创建 IPV6 子网                | 60        |
| 16.5 将根证书导入 Windows/Linux 操作系统           | 61        |
| <b>17 监控和告警</b>                          | <b>62</b> |
| 17.1 用户平时需要关注实例的哪些性能指标                   | 62        |
| 17.2 配置 DDS 告警规则时, 找不到常规监控项怎么办?          | 62        |
| 17.3 DDS 的监控指标采集粒度是多少                    | 63        |

# 1 产品咨询

## 1.1 DDS 和社区版 MongoDB 有什么关系

DDS即文档数据库服务，完全兼容MongoDB社区版3.4/4.0/4.2，部分兼容社区版4.4。兼容性详情请参见[版本兼容性](#)。

DDS支持绝大部分的MongoDB命令操作，任何兼容MongoDB的客户端都可以与DDS建立连接进行数据存储及相应操作。

您可以通过[文档数据库服务与自建数据库的对比优势](#)来了解更多DDS的优势。

更多DDS的详细信息请参见[什么是文档数据库服务](#)。

## 1.2 DDS 4.2 及其以上版本切 RocksDB 存储引擎相关问题 Q&A

### RocksDB 和 WiredTiger 两者有什么区别？

- 写机制的区别
  - WiredTiger存储引擎的数据结构为B+ tree，落盘数据中一个键（key）对应一个版本号（version），对某一个键（key）进行插入、更新或删除操作都是直接操作对应的数据节点。
  - Rocksdb存储引擎的数据结构为LSM-tree，写数据就像写日志，以追加写的方式进行。落盘数据除了一个键（key），值（value）之外，还有对应的版本号（version）。同一个键（key）在磁盘上有多个版本（version）的数据。业务层的插入、更新、删除操作，在Rocksdb侧都是写操作。这些多版本数据积压到一定阈值后，会触发后台自动归并，对多版本（version）的数据进行合并，这个行为就是RocksDB的Compaction。
- 数据组织的区别
  - WiredTiger存储引擎中一个集合（collection）的数据或者一个索引的数据对应一个磁盘文件，集合是多大，对应的文件就是多大。
  - RocksDB存储引擎下则没有集合（collection）维度，而是64M一个文件分布在磁盘上。

## RocksDB 优势在哪里？

**表1-1**从开源生态、存储空间、可定制化、内存高压场景、频繁更新删除场景、多表场景六个方面描述了RocksDB对比WiredTiger的优势。

表 1-1 RocksDB 与 WiredTiger 对比

| 存储引擎     | RocksDB                                                                              | WiredTiger                                                                         |
|----------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 开源生态     | RocksDB相比WiredTiger有更好的开源生态(github stars)，一些知名数据库比如TiDB/CRDB/YUGADB都采用了RocksDB为存储引擎。 | 开源社区生态不如RocksDB。                                                                   |
| 存储空间     | RocksDB数据写入过程中，部分业务场景会临时保存多个版本的数据，多版本数据会在后续做异步归并为最终版本，因此存储空间可能会存在临时上涨。               | WiredTiger落盘只保留最新版本的数据，存储空间消耗相对少。                                                  |
| 可定制化     | RocksDB性能调优参数较多，支持根据业务的读写能力要求做定制化的参数修改。                                              | WiredTiger性能调优参数较少。                                                                |
| 内存高压场景   | RocksDB读写性能稳定，不会随着内存压力变化。                                                            | WiredTiger需要频繁地刷脏页，进行内存与磁盘数据置换，引起读写性能波动比较大。                                        |
| 频繁更新删除场景 | RocksDB在积攒了一定量的新写数据后，会自动异步触发后台compact线程对同一数据的多版本进行归并聚合，释放多余的磁盘空间，因此基本上不会出现磁盘碎片化问题。   | WiredTiger在删除数据进行数据多版本归并聚合后，会空余出磁盘空间碎片，并且不会将这部分磁盘空间返回给操作系统，而是留待后续写入使用，磁盘碎片化问题较为明显。 |
| 多表场景     | RocksDB支持10K以上的表及索引数量。                                                               | WiredTiger在表及索引数量达到1K以上时，读写性能会受到影响。                                                |

## 为什么华为云 DDS 4.2 及其以上版本存储引擎使用 RocksDB？

基于**表1-1**的描述，对比WiredTiger，RocksDB有多个方面的优势。华为云DDS基于社区更友好的RocksDB存储引擎，通过自研演进版本，完成对MongoDB 4.2、4.4、5.0等多个版本的API兼容。

## 华为云 DDS 4.2 及其以上版本将存储引擎切换为 RocksDB 会对业务产生影响吗？

- 功能方面：存储引擎是数据库的一个组件，负责管理数据在内存和磁盘中的存储方式。RocksDB和WiredTiger虽然是不同的存储引擎，数据库服务会封装存储引擎，极大兼容社区接口使用方式，开发者使用层面无使用差异感知，业务无差异影响。
- 性能方面：当前RocksDB存储引擎已经经过长期的社区演进，结合DDS团队对其定制化的开发，性能方面二者切换无明显差异，部分查询场景下RocksDB存储引擎表现更优，存储或计算资源使用方面或存在少量差异。

## 使用 RocksDB 存储引擎在典型业务场景下有哪些表现差异与优化建议？

**表1-2**描述了RocksDB相较于WiredTiger在一些典型业务场景下的表现差异与优化建议。

**表 1-2** 典型业务场景下 RocksDB 的表现差异与优化建议

| 业务场景  | RocksDB的表现差异                                                  | 优化建议                                   | 示例                                                          |
|-------|---------------------------------------------------------------|----------------------------------------|-------------------------------------------------------------|
| 多文档扫描 | RocksDB在文档扫描（docsExamined）多的业务场景下对CPU、IO等资源消耗相较于WiredTiger更多。 | 建议通过添加适当的索引进行优化，提高扫描效率，降低CPU、IO等资源的消耗。 | 例如，在coll表A字段没有添加索引的情况下以该字段为条件进行查询：<br>db.coll.find({A: 1})。 |

## 使用 RocksDB 存储引擎的华为云 DDS 在执行大量增删改操作时，发现监控中出现磁盘来回波动、CPU 毛刺的现象，这是为什么？

- RocksDB存储引擎下，所有的写入均为追加写，不管是更新还是删除，都不会直接修改原数据，而是追加写在尾端，因此同key可能会在磁盘上存在不同版本的值。追加写过程中，同一条数据的多版本同时存在，磁盘容量会上升。
- 不同版本的值最终会在新写入的数据量到达一定量后触发后台归并，即RocksDB的Compaction动作。归并数据动作是异步操作，当一轮运行结束后会将本轮涉及到的多版本数据进行归并合并，归并后会生成新的单版本数据文件，删除旧的多版本数据文件，在生成新文件和删除老文件这段过渡期，新老文件会同时存在，即磁盘此时会短暂上涨，归并数据结束后，磁盘容量会即时下降。
- 归并数据操作本身会占用少量CPU资源，因此可能会出现轻微的CPU毛刺现象。

如果发现磁盘和CPU占用持续过高，您可以[提交工单](#)申请，完成工单提交。

## DDS 4.2 及其以上版本是否支持定制化切换存储引擎？

暂无支持计划，如有变动请关注最新服务公告。

## 1.3 使用文档数据库服务要注意什么

1. 故障切换  
文档数据库实例采用多路由+多分片+副本集的部署形态，当路由主机出现故障时，可实时动态切换。副本集包含多个副本，当主节点发生故障时，系统会在30秒之内切换到备节点。
2. 实例的弹性云服务器，对用户都不可见，这意味着，只允许用户应用程序访问数据库对应的IP地址和端口。
3. 文档数据库服务使用的对象存储服务上的文件（备份文件、慢日志文件、审计日志文件、错误日志文件），对用户不可见，它们只对后台管理系统可见。
4. 申请文档数据库服务后，用户不需要进行数据库的基础运维（比如高可用、安全补丁等），但还需要重点关注以下事情：

- a. 文档数据库实例的CPU、IOPS、空间是否足够，如果不够需要优化或者扩容。
- b. 文档数据库实例是否存在性能问题，是否有大量的慢查询，查询语句是否需要优化，是否有多余的索引或者索引缺失等。

## 1.4 什么是文档数据库实例可用性

文档数据库实例可用性的计算公式：

实例可用性=（1-故障时间/服务总时间）×100%

## 1.5 我的实例是否会受其他用户实例影响

文档数据库实例不会受其他用户实例影响，因为每个用户的实例与其他用户的实例是独立的，并且有资源隔离，互不影响。

## 1.6 DDS 是否支持多可用区部署

集群和副本集实例支持购买3AZ的实例。

- 集群实例：  
详情请参见[创建多可用区集群实例](#)。
- 副本集实例：  
详情请参见[创建多可用区副本集实例](#)。

## 1.7 实例创建成功后是否支持更换 VPC

DDS实例创建完成后，暂不支持通过控制台直接切换VPC，建议您在创建实例时谨慎选择所属VPC。

## 1.8 实例创建成功后是否支持更换区域

DDS实例创建完成后，暂不支持通过控制台直接更换区域。

## 1.9 什么是 Hidden 节点

DDS中的Hidden节点（即隐藏节点），主要用于集群和副本集实例备份数据时使用。

Hidden节点和其他备份点除了“名字不同”和“不参与选主”之外，其他完全一样。

## 1.10 什么是逻辑会话（ Logical Sessions ）

### 概述

在dds中，服务端会话（server sessions）或者逻辑会话（logical sessions）是一种机制，用于在服务端维护和管理客户端的操作状态。应用程序使用客户端会话与服务端会话进行交互。

## 会话的生命周期

### 1. 会话的创建

#### - 会话的显式创建

会话在客户端请求时由服务端创建。会话通常由驱动程序通过API创建。例如，使用mongo shell创建session：

```
// Start a session.  
session = db.getMongo().startSession( { readPreference: { mode: "primary" } } );  
  
创建的会话将在服务端维护一个独特的会话ID，用于识别和管理会话。  
  
session { "id" : UUID("1c6480a3-8e8d-4726-9f1b-6bf2f6b2b606") }
```

#### - 会话的隐式创建

隐式会话是指在执行某些操作时，驱动程序自动创建会话而无需显式地调用startSession方法。隐式会话主要用于简化开发者的代码，使操作更加便捷。当客户执行数据库操作而没有显式地提供会话参数时，驱动程序会自动创建一个会话。例如，在插入文档或查询数据时，驱动程序会在后台创建一个隐式会话，并在操作完成后自动结束会话。

#### 注意

服务端可以缓存的最大会话数默认为1000000，当会话数大于该值时，将无法显式或者隐式创建会话。

#### 会话达到上限时的行为

- 拒绝新的会话：如果会话资源达到上限，DDS将无法创建新的会话。这意味着新的操作请求将会被拒绝或阻塞，直到现有的会话释放资源。
- 资源争用：大量的会话可能会导致资源争用，例如内存和CPU利用率增加。这可能会影响数据库的性能和响应时间。
- 事务失败：如果在会话达到上限时尝试启动新事务，可能会导致事务失败。客户端会收到错误消息，指示会话资源已耗尽。

#### - 会话的使用

会话在使用过程中可以进行多种数据库操作，包括读写操作和事务操作。所有在会话上下文中的操作都会绑定到该会话。例如：

```
coll1 = session.getDatabase("mydb1").foo;  
coll2 = session.getDatabase("mydb2").bar;  
coll1.insertOne( { abc: 1 } );  
coll2.insertOne( { xyz: 999 } );
```

#### - 会话的终止

会话可以显式终止，也可以在超时后自动终止。显式终止通过调用endSession方法：

```
session.endSession();
```

服务器会自动清理终止会话所占用的资源。

### 2. 会话的管理

#### - 事务管理

会话是事务管理的核心。在事务开始时，服务器会在会话上下文中记录事务操作，确保这些操作要么全部提交，要么全部回滚。例如：

```
db.getSiblingDB("mydb1").foo.insertOne(  
  {abc: 0},
```

```
{ writeConcern: { w: "majority", wtimeout: 2000 } }
)
db.getSiblingDB("mydb2").bar.insertOne(
{xyz: 0},
{ writeConcern: { w: "majority", wtimeout: 2000 } }
)
// Start a session.
session = db.getMongo().startSession( { readPreference: { mode: "primary" } } );
coll1 = session.getDatabase("mydb1").foo;
coll2 = session.getDatabase("mydb2").bar;
// Start a transaction
session.startTransaction( { readConcern: { level: "local" }, writeConcern: { w:
"majority" } } );
// Operations inside the transaction
try {
coll1.insertOne( { abc: 1 } );
coll2.insertOne( { xyz: 999 } );
} catch (error) {
// Abort transaction on error
session.abortTransaction();
throw error;
}
// Commit the transaction using write concern set at transaction start
session.commitTransaction();
session.endSession();
```

#### - 会话超时

会话有默认的超时时间，默认超时时间为30分钟。服务器会在超时后自动终止未活动的会话，以释放资源。

## 逻辑会话的实现细节

### 1. 会话ID

每个会话都有一个唯一的会话ID，该ID在会话的整个生命周期中保持不变。会话ID用于跟踪和管理会话内的操作。

### 2. 会话资源管理

服务端根据会话的活动情况动态分配资源。未使用的会话元数据会被自动清理，会话元数据清理的writeConcern是majority，这意味着如果majority个数的备机都有时延，则无法清理会话残留的元数据，即使会话结束，服务端的会话元数据也会累积，并且占用会话数量。对于长时间运行的会话，服务端会定期检查其状态，确保不会占用过多资源。

## 使用最佳实践

1. 指定w=majority：建议客户在连接DDS实例时指定w=majority，防止主备时延大导致隐式会话累积残留，此外，如果没有以w=majority写入数据，则发生主备切换时，未同步到备机的数据有丢失风险。
2. 显式结束会话：尽量在操作完成后显式调用endSession，以释放资源。
3. 事务内操作尽量少：在事务内执行的操作应尽量少且简洁，以减少锁定和资源占用。

# 2 数据库版本

---

## 2.1 DDS 是否支持版本升级

### 大版本升级

目前DDS不支持直接通过控制台进行版本升级。如您需要可以通过数据复制服务进行迁移。详情请参见[大版本升级](#)。

### 小版本升级

文档数据库服务DDS支持手动升级内核小版本，内核小版本的升级涉及性能提升、新功能或问题修复等。

具体升级操作，请参见“[补丁升级](#)”内容。

# 3 资源冻结/释放/删除/退订

## DDS 资源为什么被释放了？

客户在华为云购买产品后，如果没有及时地进行续费或充值，将进入宽限期。如宽限期满仍未续费或充值，将进入保留期。在保留期内资源将停止服务。保留期满仍未续费或充值，存储在云服务中的数据将被删除、云服务资源将被释放。请参见[资源停止服务或逾期释放说明](#)。

## DDS 资源为什么被冻结了？

资源冻结的类型有多种，最常见类型为欠费冻结。

## 实例被冻结了，还可以备份数据吗？

不支持，如果是欠费冻结，需要您先续费解冻DDS实例后才能备份数据。

## 怎样将资源解冻？

欠费冻结：用户可通过续费或充值来解冻资源，恢复DDS正常使用。欠费冻结的DDS允许续费、释放或删除；已经到期的包周期DDS不能发起退订，未到期的包周期DDS可以退订。

## 冻结、解冻、释放资源时对业务的影响

- 资源冻结时：
  - 资源将被限制访问和使用，会导致您的业务中断。例如DDS被冻结时，会使得用户无法再连接至数据库。
  - 包周期资源被冻结后，将被限制进行变更操作。
  - 当实例的磁盘空间满且处于异常状态时，自动备份功能不会被触发。
  - 资源被冻结后，可以手动进行退订/删除。
- 资源解冻时：资源将被解除限制，用户可以连接至数据库。
- 资源释放时：资源将被释放，实例将被删除，删除前将依据用户策略决定是否执行[回收站备份](#)。

## 怎样续费？

包年/包月方式购买的DDS到期后，请在管理控制台续费管理页面进行续费操作。详细操作请参考[续费管理](#)。

## 资源被释放了能否恢复？/退订错了可以找回吗？

实例被删除，如果有回收站备份，可以通过[回收站备份](#)来恢复实例，反之则无法找回数据。

退订资源前请一定要仔细确认资源信息。如果退订错了建议重新购买使用。

## 怎样删除 DDS 实例？

- 按需实例，请参见[删除按需实例](#)。
- 包周期实例，请参见[退订包周期实例](#)。

# 4 资源及磁盘管理

## 4.1 哪些数据会占用文档数据库实例空间

以下内容占用文档数据库实例空间：

- 您申请的磁盘空间会有必要的文件系统开销，这些开销包括索引节点和保留块，以及数据库运行必需的空间。
- 除备份数据以外的用户正常的数据库。
- 文档数据库实例正常运行所需的数据，比如系统数据库、数据库回滚日志、索引等。
- 文档数据库服务产生的日志输出文件，这些文件会保证文档数据库实例正常稳定地运行。比如oplog文件，默认占磁盘空间的10%，oplog是轮转日志，所以oplog的留存时间主要由oplog产生速率决定，如需修改，可以通过控制台界面修改参数值oplogSizePercent，具体步骤请参见[修改DDS实例参数](#)。

## 4.2 占用文档数据库服务磁盘空间的日志及文件有哪些

占用磁盘空间的日志及文件如下表：

| 数据库类型   | 文件类型                                                         |
|---------|--------------------------------------------------------------|
| 文档数据库服务 | 日志文件：mongod.log、mongos.log、configsvr.log、mongodb_error.log等。 |
|         | 数据文件：数据库内容文件，索引文件。                                           |
|         | 其他文件：文档数据库服务一些临时文件、yaml配置文件等。                                |

## 4.3 为什么界面上查看的磁盘使用空间比实际的使用量小

存储在DDS磁盘上的数据，是将原始数据经过压缩后存放的，因此，界面显示的磁盘使用空间要比实际的使用量小。

## 4.4 数据删了磁盘空间但没有减少

无论写入、更新或删除（包括索引插入和删除）等操作，在后台实际上都会转成写入。实例在使用过程中，删除数据并不会回收磁盘空间，这些未被回收的磁盘空间被称为磁盘碎片。磁盘碎片会被数据库继续使用，当有新数据插入时，会重复利用这些碎片，而不需要新申请磁盘空间。根据底层存储引擎的不同（RocksDB和WiredTiger），又会有不同的具体表现。

RocksDB在删除数据后，将删除操作直接转化为追加写，在积攒了一定数据量的冗余数据后，会自动触发后台compact线程对同一数据的多版本进行归并聚合，从而释放多余的磁盘空间，因此建议等待系统自动回收。当磁盘空间使用率较高，接近[只读状态](#)阈值时，您可以[提交工单](#)申请，完成工单提交。

WiredTiger在删除数据，进行数据多版本归并聚合后，也会空余出磁盘空间碎片，但wiredTiger并不会将这部分磁盘空间返还给操作系统，而是打上标记预留给当前collection后续的写入，此collection后续的写入会优先使用这部分预留的磁盘空间，如果需要释放这部分磁盘空间，可以尝试通过执行compact命令进行释放（注：此命令会阻塞正常业务，因此默认情况下是禁用状态）。

## 4.5 购买的副本集实例是 4 核 8GB 的，为什么驻留内存只有 4GB 左右而不是 8GB

驻留内存如果占了100%，系统会发生卡顿，业务无法正常进行。

# 5 扩容及规格变更

## 5.1 DDS 是否支持不同类型实例变更

DDS支持集群、副本集和单节点类型的实例，不同类型实例之间不能互相变更，可以通过数据复制服务（DRS）实现数据迁移。

例如：您想将副本集类型的实例变更为集群类型的实例，可以先购买集群实例，通过数据复制服务将已有副本集实例的数据迁移至集群实例。

通过数据复制服务（DRS）进行数据迁移，详细操作请参见[通过数据复制服务迁移数据](#)。

## 5.2 DDS 是否支持添加节点

DDS的集群和副本集实例均支持添加节点，单节点由于架构比较特殊，不支持添加节点。

- 集群实例

集群实例由dds mongos（路由）、Config（配置）和Shard（分片）节点组成。其中，Shard节点和dds mongos节点支持添加节点，Config节点不支持添加节点。

具体操作请参见[添加集群实例的节点](#)。

- 副本集实例

副本集实例由主节点、备节点和隐藏节点组成。系统默认提供的是三节点的实例，您可以根据需要新增节点至五节点或七节点副本集，新增的节点均为Secondary节点。

具体操作请参见[变更副本集实例的节点](#)。

## 5.3 DDS 在分片节点扩容期间服务是否可用

可用，文档数据库服务的扩容操作是在实例中增加新的分片，原实例中的分片不变，不影响服务的可用性。

# 6 数据库性能

## 6.1 集群和副本集在什么情况下会触发主备切换

### 什么情况下会触发主备切换

文档数据库服务支持两种高可用架构：分片集群（Sharding）和副本集（Replica Set）。

- 集群架构提供dds mongos、shard、config三类节点。其中，shard和config为三节点副本集架构，当路由主机出现故障时，可实时动态切换。
- 副本集架构提供Primary、Secondary、Hidden三个节点。其中，Primary和Secondary节点对外提供连接IP。副本集的主节点不是固定的，当出现副本集配置改变、主节点宕机、人为主备切换等状况，会导致主备节点切换，此时副本集可能会选举出新主节点，原来的主节点则降为备节点。
- 内存使用率过高，即实例的负载过高时，会发生主备切换。

更多信息，请参见[集群架构](#)和[副本集架构](#)。

### 主备切换的影响

- 当主节点发生故障时，系统会在30秒之内切换到备节点。
- 如果您的应用通过Primary节点的地址连接，当发生主备切换而更换了主节点，对业务的读写操作造成影响。

### 业务部署建议

请确认您的应用具备自动重连机制，连接闪断后可自动重连，避免在发生主备切换后，从客户端写入数据报错。

- 推荐您使用URL连接地址友好连接集群和副本集实例，当某个节点出现故障时，不会因为主备切换而影响业务的读写操作。详情请参见[连接集群实例](#)和[如何连接副本集实例以实现读写分离和高可用](#)。
- 实例负载过高时，建议您进行扩容操作。以集群为例，详情请参见[变更集群实例的CPU和内存规格](#)。

## 6.2 磁盘使用率高问题排查

### 操作场景

使用文档数据库服务时，如果实例的磁盘使用率过高或完全被使用，将会导致实例不可用。

本章节帮助您分析解决DDS实例磁盘使用率高的问题。

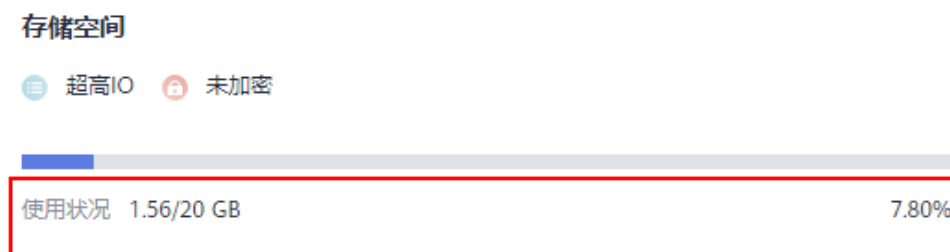
### 查看磁盘使用情况

DDS目前提供如下两种方法用于查看实例磁盘的使用情况。

1. 通过DDS控制台查看磁盘使用情况。

您可以登录DDS控制台，单击目标实例名称，进入“基本信息”页面，在“存储空间”区域可以查看到当前实例的磁盘使用情况。

图 6-1 查看磁盘使用情况



2. 通过查看监控指标（磁盘利用率和磁盘使用量）判断实例磁盘的使用情况。  
如何查看监控指标请参见[查看监控指标](#)。

图 6-2 查看磁盘使用情况



## 解决方案

1. 对于集群实例，可能是因为选择和设置的分片不合理导致数据分布不均衡，从而引起磁盘空间使用率高的情况。

此时，可以对数据库集合进行合适的分片，具体操作请参见[设置数据分片以充分利用分片性能](#)。

2. 随着业务数据的增加，原来申请的数据库磁盘容量不足。此时建议扩容磁盘空间，确保磁盘空间足够。

- 集群实例，请参见[扩容集群实例的磁盘空间](#)。
- 副本集实例，请参见[扩容副本集实例的磁盘空间](#)。
- 单节点实例，请参见[扩容单节点实例的磁盘空间](#)。

如果原有实例规格的磁盘已是最大的，请先升级规格。

- 集群实例，请参见[变更集群实例的CPU和内存规格](#)。
- 副本集实例，请参见[变更副本集实例的CPU和内存规格](#)。
- 单节点实例，请参见[变更单节点实例的CPU和内存规格](#)。

3. 存在大量过期的数据文件占用磁盘空间，此时需要及时处理过期数据文件。例如：如果整个库不需要了，则可以执行dropDatabase命令进行删除，从而快速释放磁盘空间。

4. 后台数据处理机制原因。

无论写入、更新或删除（包括索引插入和删除）等操作，在后台实际上都会转成写入。实例在使用过程中，删除数据并不会回收磁盘空间，这些未被回收的磁盘空间被称为磁盘碎片。磁盘碎片会被数据库继续使用，当有新数据插入时，会重复利用这些碎片，而不需要新申请磁盘空间。根据底层存储引擎的不同（RocksDB和WiredTiger），又会有不同的具体表现。

RocksDB在删除数据后，将删除操作直接转化为追加写，在积攒了一定数据量的冗余数据后，会自动触发后台compact线程对同一数据的多版本进行归并聚合，从而释放多余的磁盘空间，因此建议等待系统自动回收。当磁盘空间使用率较高，接近[只读状态](#)阈值时，您可以[提交工单](#)申请，完成工单提交。

WiredTiger在删除数据，进行数据多版本归并聚合后，也会空余出磁盘空间碎片，但wiredTiger并不会将这部分磁盘空间返还给操作系统，而是打上标记预留给当前collection后续的写入，此collection后续的写入会优先使用这部分预留的磁盘空间，如果需要释放这部分磁盘空间，可以尝试通过执行compact命令进行释放（注：此命令会阻塞正常业务，因此默认情况下是禁用状态）。

## 6.3 副本集中主备同步存在多长时间的延迟

主节点与备节点的复制延迟是无法利用公式来计算的，它会受到以下因素的影响：

1. 网络通信状况。
2. 主节点的事务压力，即主机每秒执行的事务量。
3. 主节点执行的事务大小，即一个事务要执行多久。
4. 备节点的机器负载状况。

如果主节点在某个时间段压力非常大，每秒执行的事务量非常多，备机通常会出现延迟。

对于文档数据库服务而言，可以在云监控中查看副本集备节点的“主备延时”指标，来观察延迟情况。

## 6.4 副本集主备节点之间的数据同步方式是什么

副本集主备节点之间的数据同步方式是异步方式。

## 6.5 分片内存超过 90%产生告警，如何消减

建议将`enableMajorityReadConcern`参数的值修改为`false`，为了不影响业务，重启该节点。

## 6.6 DDS 集群持续写入数据后查询报错怎么处理

### 问题描述

DDS集群持续写入数据，写入后查询报错。

示例：

```
W SHARDING [Balancer] Failed to enforce tag ranges :: caused by :: ExceededTimeLimit: Unable to obtain shard utilization information for shard01 due to Operation timed out, request was RemoteCommand -- target: 192.168.*.*:8635, db: admin, cmd:{ getShardStatistics: 1, maxTimeMS: 30000 }
```

### 可能原因

由于持续写入数据，导致后台chunk分裂超时。

### 解决方法

该类型报错属于正常的warning报错，对用户的使用不会有影响，稍后重试查询即可。

# 7 数据库权限

## 7.1 实例创建时指定的 root 账号拥有什么权限？

当实例创建完成后，DDS会为用户在admin数据库中创建了一个root账号，该账号拥有MongoDB内置的root权限。MongoDB内置的角色定义请参见[内置角色](#)。

您可以登录到实例后，通过如下命令查看对应账号的权限信息。

```
show users
```

```
show roles
```

## 7.2 默认权限机制

DDS与社区原生的版本相比，对安全进行一系列的增强，以应对越来越严峻的安全挑战。社区原生版本支持不鉴权的方式连接使用数据库的，而DDS采取默认安全策略，连接数据库必须通过鉴权，否则，无法使用数据库。

- 数据库实例创建后，系统会创建默认的管理员用户rwuser，但是需要需要用户指定，并满足密码复杂度要求。
- 通过此用户创建和管理客户自定义的角色和用户。该用户无默认密码，需要客户指定，并且需要满足密码复杂度要求。

### 使用场景

在执行mongodump和mongorestore操作时，如果对全实例进行备份恢复，则会出现权限验证失败。因为rwuser用户对实例上的admin库和config库权限受限。此时，明确指定用户业务相关库表，即可正常操作。

## 7.3 角色管理

DDS通过基于角色的管理来控制用户对数据访问的权限，角色共分为两类：预置角色和自定义角色。

### 7.3.1 预置角色

预置角色是系统自动生成的角色信息，客户端可用的预置角色名称有read，readWrite。

mongodb使用角色来管理数据库的，所以创建一个用户时就需要赋予一个角色。角色除了内置之外，也可以[自定义角色](#)。

表 7-1 常见内置角色

| 角色                   | 权限描述                                                                             | 包含的操作命令                                                                                                                                                                                 |
|----------------------|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| read                 | read角色包含读取所有非系统集合数据和订阅部分系统集合(system.indexes、system.js、system.namespaces)的权限。     | changeStream、collStats、dbHash、dbStats、find、killCursors、listIndexes、listCollections                                                                                                      |
| readWrite            | readWrite角色包含read角色的权限同时增加了对非系统集合数据的修改权限，但只对系统集合system.js有修改权限。                  | collStats、convertToCapped、createCollection、dbHash、dbStats、dropCollection、createIndex、dropIndex、find、insert、killCursors、listIndexes、listCollections、remove、renameCollectionSameDB、update |
| readAnyDatabase      | readAnyDatabase角色包含对除了config和local之外所有数据库的只读权限。同时对于整个集群包含listDatabases命令操作。      | 在MongoDB3.4版本之前，该角色包含对config和local数据库的读取权限。当前版本如果需要对这两个数据库进行读取，则需要在admin数据库授予用户对这两个数据库的read角色。                                                                                          |
| readWriteAnyDatabase | readWriteAnyDatabase角色包含对除了config和local之外所有数据库的读写权限。同时对于整个集群包含listDatabases命令操作。 | 在MongoDB3.4版本之前，该角色包含对config和local数据库的读写权限。当前版本如果需要对这两个数据库进行读写，则需要在admin数据库授予用户对这两个数据库的readWrite角色。                                                                                     |

| 角色                 | 权限描述                                                                                                             | 包含的操作命令                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dbAdmin            | dbAdmin角色包含执行某些管理任务(与schema相关、索引、收集统计信息)的权限,该角色不包含用户和角色管理的权限。                                                    | <ul style="list-style-type: none"><li>对于系统集合(system.indexes、system.namespaces、system.profile)包含命令操作: collStats、dbHash、dbStats、find、killCursors、listIndexes、listCollections、dropCollection and createCollection(仅适用 system.profile)</li><li>对于非系统集合包含命令操作: bypassDocumentValidation、collMod、collStats、compact、convertToCapped、createCollection、createIndex、dbStats、dropCollection、dropDatabase、dropIndex、enableProfiler、reIndex、renameCollectionSameDB、repairDatabase、storageDetails、validate</li></ul> |
| dbAdminAnyDatabase | dbAdminAnyDatabase角色包含类似于dbAdmin角色对于所有数据库管理权限,除了config数据库和local数据库。同时对于整个集群包含listDatabases命令操作。                  | 在MongoDB3.4版本之前,该角色包含对config和local数据库的管理权限。当前版本如果需要对这两个数据库进行管理,则需要在admin数据库授予用户对这两个数据库的dbAdmin角色。                                                                                                                                                                                                                                                                                                                                                                                                      |
| clusterAdmin       | clusterAdmin角色包含MongoDB集群管理最高的操作权限。                                                                              | clusterManager、clusterMonitor和hostManager三个角色的所有权限,并且还拥有dropDatabase操作命令的权限。                                                                                                                                                                                                                                                                                                                                                                                                                           |
| userAdmin          | userAdmin角色包含对当前数据库创建和修改角色和用户的权限。该角色允许向其它任何用户(包括自身)授予任何权限,所以这个角色也提供间接对超级用户(root)的访问权限,如果限定在admin数据库中,也包括集群管理的权限。 | changeCustomData、changePassword、createRole、createUser、dropRole、dropUser、grantRole、revokeRole、setAuthenticationRestriction、viewRole、viewUser                                                                                                                                                                                                                                                                                                                                                            |

| 角色                   | 权限描述                                                                       | 包含的操作命令                                                                                                                                                                                                                                                          |
|----------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| userAdminAnyDatabase | userAdminAnyDatabase角色包含类似于userAdmin角色对于所有数据库的用户管理权限，除了config数据库和local数据库。 | <ul style="list-style-type: none"><li>对于集群包含命令操作：<br/>authSchemaUpgrade、invalidateUserCache、listDatabases</li><li>对于系统集合admin.system.users和admin.system.roles包含命令操作：<br/>collStats、dbHash、dbStats、find、killCursors、planCacheRead、createIndex、dropIndex</li></ul> |

## 7.3.2 自定义角色

用户自定义角色是用户通过命令创建的定制化的角色，只包含CRUD操作的一种或多种，或者内置角色的一种或多种。可以通过命令针对不同的资源、不同的操作进行自定义，除了预置角色无法被修改以外，其他应用方式是相同的。

创建、修改和删除角色

- 要创建角色前，需使用具有权限的用户（可使用rwuser）连接到数据库实例。详情请参见[连接数据库](#)。
- 通过createRole创建自定义角色，可以针对不同db，不同的collection进行权限控制，也可以从其他角色上继承。
- 角色创建完成后如需调整权限，可通过grantPrivilegesToRole，grantRolesToRole，revokeRolesFromRole或revokePrivilegesFromRole命令获取或者收回权限。详情请参见[创建并管理角色](#)。

## 7.3.3 创建并管理角色

### 创建角色

**db.createRole(role, writeConcern)**

- 参数role为必选参数，类型为文档，详情如下：

```
{
  role: "<name>",
  privileges: [
    { resource: { <resource> }, actions: [ "<action>", ... ] },
    ...
  ],
  roles: [
    { role: "<role>", db: "<database>" } | "<role>",
    ...
  ],
  authenticationRestrictions: [
    {
      clientSource: [ "<IP>" | "<CIDR range>", ... ],
      serverAddress: [ "<IP>" | "<CIDR range>", ... ]
    },
    ...
  ]
}
```

各参数及含义如下：

| 字段                         | 类型     | 说明                                                                                                                                                                                                                                                                              |
|----------------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| role                       | string | 角色名称                                                                                                                                                                                                                                                                            |
| privileges                 | 数组     | 必选参数，数组元素为角色具备的权限。<br>该参数设置为空集合意味着该角色无任何权限。                                                                                                                                                                                                                                     |
| resource                   | 文档     | 用于指定库名或者集合名                                                                                                                                                                                                                                                                     |
| actions                    | 数组     | 对应资源可用的操作列表，常用action如下： <ul style="list-style-type: none"><li>• find</li><li>• count</li><li>• getMore</li><li>• listDatabases</li><li>• listCollections</li><li>• listIndexes</li><li>• insert</li><li>• update</li><li>• remove</li></ul> 更多actions请参见 <a href="#">官方文档</a> 。 |
| roles                      | 数组     | 必选参数，数组元素为角色继承的角色名。<br>可以是系统预置的角色read或者readWrite，也可以是用户自定义的角色。                                                                                                                                                                                                                  |
| authenticationRestrictions | 数组     | 可选。用于指定该角色可接入的IP地址或者IP地址段。                                                                                                                                                                                                                                                      |

- writeConcern可选参数用来指定命令的write concern的等级。

## 更新角色

**db.grantPrivilegesToRole(rolename,privileges,writeConcern)**

**db.revokePrivilegesFromRole(rolename,privileges,writeConcern)**

上述命令用户给角色获取或者回收指定的权限。

- 参数rolename为必选参数，用于指定待更新的角色名称。

- 参数privileges为角色待调整的权限。  

```
db.grantPrivilegesToRole(  
  "< rolename >",  
  [  
    { resource: { <resource> }, actions: [ "<action>", ... ] },  
    ...  
  ],  
  { < writeConcern > }  
)
```

表 7-2 privileges 各字段含义

| 字段       | 类型 | 说明                 |
|----------|----|--------------------|
| resource | 文档 | 用于指定库名或者集合名        |
| actions  | 数组 | 参考createRole章节的说明。 |

除了上述命令，还支持updateRole对角色信息整体更新。

**db.updateRole(role, update, writeConcern)**

表 7-3 参数说明

| 字段           | 类型     | 说明                                               |
|--------------|--------|--------------------------------------------------|
| role         | string | 角色名称                                             |
| update       | 数组     | 必选参数，与创建角色命令的privileges参与含义一致。用于整体替换对应角色的全部权限信息。 |
| writeConcern | 文档     | 可选参数用来指定命令的write concern 的等级                     |

删除角色

**db.dropRole(rolename, writeConcern)**

- 参数rolename为必选参数，待删除的角色名称。
- writeConcern可选参数用来指定命令的write concern的等级。

7.4 用户管理

DDS上用户的权限都是基于角色管理，通过给用户赋予不同的角色来进行差异化的权限控制。

为了给文档数据库实例提供管理服务，您在创建数据库实例时，文档数据库服务会自动为实例创建admin、monitor和backup账户。如果试图删掉、重命名、修改这些账户的密码和权限，会导致出错。

对于数据库管理员账户rwuser，以及您所创建的账户，允许修改账户的密码。

### 7.4.1 创建用户

#### 操作须知

- 下面所有操作都对权限要求，默认rwuser用户具备所需权限，如果通过用户自定义用户进行管理，则需要关注是否具备操作权限。
- 使用具备权限的用户（可使用rwuser）连接到数据库实例。
- 通过createUser创建所需的用户，通过设置合适的角色来控制对应用户的权限。其中需要注意的是"passwordDigestor" 参数必须是 "server"，否则命令会执行失败，增加这个约束是为了避免安全隐患。

#### 创建用户

##### db.createUser(user, writeConcern )

- 命令中user为必选参数，类型为文档，包含了要创建用户的身份认证和访问信息。
- writeConcern为可选参数，类型为文档，包含了创建操作的write concern级别。

user文档定义了用户，其格式如下：

```
{
  user: "<name>",
  pwd: "<cleartext password>",
  customData: { <any information> },
  roles: [
    { role: "<role>", db: "<database>" } | "<role>",
    ...
  ],
  authenticationRestrictions: [
    {
      clientSource: [ "<IP>" | "<CIDR range>", ... ],
      serverAddress: [ "<IP>" | "<CIDR range>", ... ]
    },
    ...
  ]
  mechanisms: [ "<SCRAM-SHA-1|SCRAM-SHA-256>", ... ],
  passwordDigestor: "<server|client>"
}
```

表 7-4 user 参数说明

| 字段   | 类型     | 说明                                                                          |
|------|--------|-----------------------------------------------------------------------------|
| user | string | 新用户名称。                                                                      |
| pwd  | string | 用户密码，如果您在\$external数据库上运行db.createUser()以创建将凭据存储在 MongoDB外部的用户，则pwd字段不是必需的。 |

| 字段                         | 类型     | 说明                                                            |
|----------------------------|--------|---------------------------------------------------------------|
| customData                 | 文档     | 可选的。任何任意信息，该字段可用于存储管理员希望与此特定用户关联的任何数据。例如，这可以是用户的全名或员工ID。      |
| roles                      | 数组     | 授予用户的角色。可以指定一个空数组[]来创建没有角色的用户。                                |
| authenticationRestrictions | 数组     | 可选的。服务器对创建的用户强制执行的身份验证限制，用于指定该角色可接入的IP地址或者IP地址段。              |
| mechanisms                 | 数组     | 可选的。指定用于创建SCRAM用户凭据的特定SCRAM机制。当前只包含SCRAM-SHA-1和SCRAM-SHA-256。 |
| passwordDigestor           | string | 可选的。指示是在server端还是client端验证密码，默认是server。                       |

## 示例

- 创建用户时不同数据库赋予不同角色

使用db.createUser()在products数据库中创建accountAdmin01用户。

```
use products
db.createUser( { user: "accountAdmin01",
  pwd: "Changeme_123",
  customData: { employeeId: 12345 },
  roles: [ { role: "clusterAdmin", db: "admin" },
    { role: "readAnyDatabase", db: "admin" },
    "readWrite" ] },
  { w: "majority", wtimeout: 5000 } )
```

以上操作赋予用户accountAdmin01以下角色：

- 在admin数据库中角色为clusterAdmin和readAnyDatabase。
- 在products数据库中角色为readWrite。

- 创建用户时同一数据库赋予多个角色

以下操作创建一个在products数据库中角色为readWrite和dbAdmin的用户，用户名为accountUser。

```
use products
db.createUser(
  {
    user: "accountUser",
    pwd: "Changeme_123",
    roles: [ "readWrite", "dbAdmin" ]
```

```
}  
)
```

- **创建用户时不赋予角色**

以下操作在admin数据库中创建一个用户名为reportsUser的用户，但是没有赋予用户角色。

```
use admin  
db.createUser(  
  {  
    user: "reportsUser",  
    pwd: "Chagneme_123",  
    roles: [ ]  
  }  
)
```

- **创建管理员用户并赋予角色**

下面的操作将在admin 数据库中创建名为appAdmin的用户，并赋予该用户对config数据库的读写访问权限，使用户可以更改分片群集的某些设置，如分片均衡器设置。

```
use admin  
db.createUser(  
  {  
    user: "appAdmin",  
    pwd: "Changeme_123",  
    roles:  
      [  
        { role: "readWrite", db: "config" },  
        "clusterAdmin"  
      ]  
  }  
)
```

- **创建有身份验证限制的用户**

以下操作将在管理员数据库中创建一个名为restricted的用户。该用户只能从IP地址192.0.2.0连接到IP地址198.51.100.0时才能进行身份验证。

```
use admin  
db.createUser(  
  {  
    user: "restricted",  
    pwd: "Changeme_123",  
    roles: [ { role: "readWrite", db: "reporting" } ],  
    authenticationRestrictions: [ {  
      clientSource: ["192.0.2.0"],  
      serverAddress: ["198.51.100.0"]  
    } ]  
  }  
)
```

- **仅使用SCRAM-SHA-256证书创建用户**

下面的操作将创建一个只有SCRAM-SHA-256凭据的用户。

```
use reporting  
db.createUser(  
  {  
    user: "reportUser256",  
    pwd: "Changeme_123",  
    roles: [ { role: "readWrite", db: "reporting" } ],  
    mechanisms: [ "SCRAM-SHA-256" ]  
  }  
)
```

如果设置了authenticationMechanisms参数，mechanisms字段只能包含authenticationMechanisms参数中指定的值。

7.4.2 更新用户

db.updateUser(username, update, writeConcern )

- 命令中参数username为要更新的用户名。
- update为文档类型，包含用户替换数据的文档。
- writeConcern为可选参数，更新操作的write concern级别。

```
db.updateUser(  
  "<username>",  
  {  
    customData : { <any information> },  
    roles : [  
      { role: "<role>", db: "<database>" } | "<role>",  
      ...  
    ],  
    pwd: passwordPrompt(), // Or "<cleartext password>"  
    authenticationRestrictions: [  
      {  
        clientSource: ["<IP>" | "<CIDR range>", ...],  
        serverAddress: ["<IP>", | "<CIDR range>", ...]  
      },  
      ...  
    ],  
    mechanisms: [ "<SCRAM-SHA-1|SCRAM-SHA-256>", ... ],  
    passwordDigestor: "<server|client>"  
  },  
  writeConcern: { <write concern> }  
)
```

表 7-5 update 参数说明

| 字段                         | 类型     | 说明                                                              |
|----------------------------|--------|-----------------------------------------------------------------|
| customData                 | 文档     | 可选。任意信息                                                         |
| roles                      | 数组     | 可选。授予用户的角色。roles数组的更新将覆盖以前的数组的值。                                |
| pwd                        | string | 可选。用户密码                                                         |
| authenticationRestrictions | 数组     | 可选。服务器对用户实施的身份验证限制，用于指定该角色可接入的IP地址或者IP地址段。                      |
| mechanisms                 | 数组     | 可选的。指定用于创建SCRAM用户凭据的特定SCRAM机制。当前只包含SCRAM-SHA-1 和 SCRAM-SHA-256。 |

| 字段               | 类型     | 说明                                     |
|------------------|--------|----------------------------------------|
| passwordDigester | string | 可选的。指示是在server端还是client端验证密码，默认是server |

## 示例

- 更新用户信息

products数据库中的用户appClient01，其信息如下：

```
{
  "_id" : "products.appClient01",
  "token" : NumberLong("8424642624807814713"),
  "user" : "appClient01",
  "db" : "products",
  "customData" : {
    "emplID" : "12345",
    "badge" : "9156"
  },
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "products"
    },
    {
      "role" : "read",
      "db" : "inventory"
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
```

下面操作会替换用户的自定义数据和角色数据：

```
use products
db.updateUser( "appClient01",
{
  customData : { employeeId : "0x3039" },
  roles : [
    { role : "read", db : "assets" }
  ]
})
```

products数据库中的用户appClient01，经过更新后信息如下：

```
{
  "_id" : "products.appClient01",
  "token" : NumberLong("8424642624807814713"),
  "user" : "appClient01",
  "db" : "products",
  "customData" : {
    "employeeId" : "0x3039"
  },
  "roles" : [
    {
      "role" : "read",
      "db" : "assets"
    }
  ]
}
```

```
    }  
  ],  
  "mechanisms" : [  
    "SCRAM-SHA-1",  
    "SCRAM-SHA-256"  
  ]  
}
```

- **更新用户以仅使用SCRAM-SHA-256凭证**

reporting数据库中的用户reportUser256，其信息如下：

```
{  
  "_id" : "reporting.reportUser256",  
  "token" : NumberLong("2827251846225877395"),  
  "user" : "reportUser256",  
  "db" : "reporting",  
  "roles" : [ ],  
  "mechanisms" : [  
    "SCRAM-SHA-1",  
    "SCRAM-SHA-256"  
  ]  
}
```

以下操作会将当前同时拥有 SCRAM-SHA-256 和 SCRAM-SHA-1 全权证书的用户更新为只拥有 SCRAM-SHA-256 全权证书的用户。

---

 **注意**

- 如果密码未与mechanisms一起指定，则只能将mechanisms更新为用户当前SCRAM机制的子集。
- 如果密码与mechanisms一起指定，则可以指定任何受支持的SCRAM机制。
- 对于SCRAM-SHA-256，passwordDigestor必须是默认值 "server"。

---

```
db.updateUser(  
  "reportUser256",  
  {  
    mechanisms: [ "SCRAM-SHA-256" ]  
  }  
)
```

reporting数据库中的用户reportUser256，经过更新后信息如下：

```
{  
  "_id" : "reporting.reportUser256",  
  "token" : NumberLong("2827251846225877395"),  
  "user" : "reportUser256",  
  "db" : "reporting",  
  "roles" : [ ],  
  "mechanisms" : [  
    "SCRAM-SHA-256"  
  ]  
}
```

### 7.4.3 删除用户

**db.dropUser(username, writeConcern)**

- username为要从数据库中删除的用户名。
- writeConcern为可选参数，移除操作的writeConcern级别。

## 示例

下面是操作将reportUser1用户从产品数据库中删除。

```
use products
db.dropUser("reportUser1", {w: "majority", wtimeout: 5000})
```

删除成功后执行db.getUser将显示null。

```
replica:PRIMARY> db.getUser("reportUser1")
null
```

# 8 创建与删除

## 8.1 创建 DDS 实例时，如何选择规格和节点

- 实例规格越高，性能越好。DDS支持的实例规格请参见[数据库实例规格](#)。
- 相同实例规格下，节点数越多，性能越好。性能相关信息请参见[性能白皮书](#)。
- 建议根据业务实际的测试数据选择规格，并考虑可靠性冗余，和未来业务增长，适当预留一些资源。

### 说明

- 对于集群实例，根据经验，单核CPU支持16GB数据时性能较好。
- 对于副本集实例，使用[快速购买](#)方式，里面会向您推荐多数用户选择的配置。详情请参见[快速购买副本集实例](#)。

## 8.2 实例创建成功后，为什么在控制台看不到

当您创建完DDS实例后，在管理控制台上找不到，本文提供了常见的几种原因。

- 权限不足  
可能是由于您的账号没有相应的[权限](#)，您可以通过[创建用户并授权使用DDS](#)来配置对应的权限。
- 区域选择不正确  
只有切换到实例所在的区域，才能在实例列表中看到创建的实例。

您可以单击管理控制台左上角的，选择实例所在的区域和项目。

图 8-1 选择区域



### 8.3 可以通过创建模板的方式创建实例吗

在创建文档数据库实例时，不需要使用模板。用户可选择不同实例规格，相当于提前准备好了很多模板，提供给用户创建实例。

### 8.4 为什么我的数据会丢失或被删除

文档数据库服务不会删除和操作用户的任何数据。出现这种情况，请检查是否为误操作，必要时可利用已有备份恢复文件。

目前可采取如下方式：

- 使用文档数据库服务的恢复功能，请参见[备份与恢复](#)。
- 将备份数据导入文档数据库服务，请参见[数据迁移](#)。

### 8.5 如果删除我的云账号是否会删除备份

一旦删除云账号，自动备份和手动备份也随之删除。

### 8.6 删除和退订实例的区别是什么

针对不同计费模式的数据库实例，释放资源的方式不同。

- 对于“按需计费”模式的数据库实例，购买后无订单生成，根据业务需要释放资源时，在“实例管理”页面手动删除即可。
- 对于“包年/包月”模式的数据库实例，购买后会生成订单，您需要退订订单，从而释放数据库实例资源，详细操作请参见[退订包周期实例](#)。

# 9 数据库连接

## 9.1 DDS 实例连接失败怎么办

### 问题分析

主要从九个方面考虑：

#### 1. 排除数据库实例异常

例如：DDS系统故障，实例状态异常，实例或表被锁定。

#### 2. （常见）使用正确的客户端连接方式

- 数据库安装包须从官网下载并正确安装，请参见[如何安装MongoDB客户端](#)。
- 建议使用4.0版本以上的MongoDB客户端连接实例。
- 内网连接需要DDS与ECS实例必须在同一区域、VPC内。
- 公网连接需要购买或使用已有EIP，并对DDS实例绑定该EIP。

#### 3. 使用正确的SSL方式安全连接

界面SSL开关开启和关闭，分别对应不同的连接命令。例如：

- 开关开启：`./mongo --host<DB_HOST>--port<DB_PORT>-u<DB_USER>-p--authenticationDatabaseadmin--ssl --sslCAFile<FILE_PATH> --sslAllowInvalidHostnames`
- 开关关闭：`./mongo --host<DB_HOST>--port<DB_PORT>-u<DB_USER>-p --authenticationDatabase admin`

#### 4. 排除连接命令错误

例如：连接地址错误、端口参数配置错误、用户名和密码错误、SSL方式下命令错误。

#### 5. （常见）排除网络不通

##### 内网访问

- 确认ECS与DDS是否在同一个区域、同一个VPC。  
ECS与DDS在不同VPC的场景，可以通过建立[VPC对等连接](#)实现网络互通。
- 检查安全组规则。
  - 安全组外访问安全组内的DDS集群实例时，需要为安全组[添加相应的入方向规则](#)。

- c. 在ECS上测试是否可以正常连接到DDS实例地址的端口。

公网访问

- a. 检查安全组规则。
- i. 安全组外访问安全组内的DDS集群实例时，需要为安全组[添加相应的入方向规则](#)。
- b. 检查网络ACL规则。
- c. 相同区域主机进行ping测试。

跨网段访问（配置IP映射）

通过内网连接副本集实例时，当客户端和副本集实例部署在不同网段时，且客户端所在的网段不在“192.168.0.0/16”，“172.16.0.0/24”和“10.0.0.0/8”时，需要进行跨网段访问配置，以实现网络互通。

- a. 确保在源端ECS网络与实例节点网络连通，如果网络不通，可以参考[对等连接](#)进行相关配置。
- b. 跨网段访问的操作请详见[跨网段访问配置](#)。

6. [（常见）排除实例的连接数满的情况](#)

实例连接数过多，可能会导致业务侧无法正常连接。

7. [（常见）排除磁盘满的情况](#)

实例磁盘使用率过高会导致实例状态异常，无法正常连接实例。

8. [排除CPU使用率飙升问题](#)

CPU使用率很高或接近100%，会导致数据读写处理缓慢、无法获取连接、出现报错等，从而影响业务正常运行。

9. [连接失败的常见报错](#)

包含连接失败的常见报错，以及相应的解决方法。

排查思路

1. 排除数据库实例异常

检查办法：请在控制台检查数据库实例状态是否为“正常”。

可能原因：DDS系统故障，实例状态异常，实例或表被锁定等。

解决方法

- DDS实例状态异常，您可以[提交工单](#)申请，完成工单提交。
- 系统故障、实例或表被锁定等，请尝试重启功能，详细操作请参考[重启实例](#)。

图 9-1 查看实例状态

| 筛选                                            | 按包规格 | 通过   | 所有项目 | 全部实例                  | 请输入实例名称、实例ID或连接IP                       | Q  | 快速搜索 | 🔄    | 🔍  |
|-----------------------------------------------|------|------|------|-----------------------|-----------------------------------------|----|------|------|----|
| 实例名称/ID                                       | 运行状态 | 运行模式 | 计费模式 | 连接地址                  | 企业项目                                    | 操作 |      |      |    |
| db-3c74<br>@9a52a0f3a54a858c71251b77eb1e8dno2 | 正常   | 集群   | 包年包月 | 2021/04/26 15:55:4... | mongodb://newser****@192.168.0.64:27011 | 默认 | 登录   | 重置密码 | 更多 |

2. 使用正确的客户端连接方式

建议使用4.0版本以上的MongoDB客户端连接实例，安装客户端操作请参见[如何安装MongoDB客户端](#)。

以集群为例：

- a. 内网连接实例的具体操作步骤，请参见[通过内网连接集群实例](#)。

b. 公网连接实例的具体操作步骤，请参见[通过公网连接集群实例](#)。

表 9-1 客户端连接方式

| 连接方式 | 使用场景                                                                                                       |
|------|------------------------------------------------------------------------------------------------------------|
| 内网方式 | 系统默认提供内网IP地址。<br>当应用部署在ECS上，且该ECS与DDS实例处于同一区域，同一VPC时，建议单独使用内网IP连接ECS与DDS实例。                                |
| 公网方式 | 不能通过内网IP地址访问DDS实例时，使用公网访问，建议单独绑定弹性公网IP连接ECS（或公网主机）与DDS实例。<br>用户需要购买弹性公网IP，请参见 <a href="#">弹性公网IP计费说明</a> 。 |

3. 使用正确的SSL方式安全连接

a. （推荐）SSL方式：实例连接管理页面的SSL开关开启，并且上传证书到ECS。

```
./mongo --host<DB_HOST>--port<DB_PORT>-u<DB_USER>-p--  
authenticationDatabaseadmin--ssl --sslCAFile<FILE_PATH> --  
sslAllowInvalidHostnames
```

图 9-2 SSL 开启



b. 普通方式：实例连接管理页面的SSL开关关闭。

```
./mongo --host<DB_HOST>--port<DB_PORT>-u<DB_USER>-p --  
authenticationDatabase admin
```

图 9-3 SSL 关闭



4. 排除连接命令错误

请获取正确的连接地址、端口参数配置、用户名和密码、SSL方式下命令，并重试连接实例。

**SSL内网连接集群示例：**  
`./mongo mongodb://  
rwuser:<password>@<DB_HOST>:<DB_PORT>,<DB_HOST>:<DB_PORT>/test?  
authSource=admin --ssl --sslCAFile <FILE_PATH> --  
sslAllowInvalidHostnames`

- 连接地址

连接信息可在“实例管理”页的“连接地址”列获取。

图 9-4 连接地址

| 实例名称/ID                          | 实例备注 | 实例类型 | 数据库版本   | 存储引擎       | 运行状态 | 计费模式                     | 连接地址                                                               | 操作           |
|----------------------------------|------|------|---------|------------|------|--------------------------|--------------------------------------------------------------------|--------------|
| 005-7c94<br>c38050842e490507a... |      | 副本集  | 社区版 4.0 | WiredTiger | 正常   | 按量计费<br>2022/08/21 09... | mongodb://rwuser:rwuser-password@192.168.22.105:8635,192.168.22... | 登录 查看监控指标 更多 |

- 数据库端口  
目标实例的“连接管理”页面，“内网连接”页签的“数据库端口”。
- 用户名和密码  
root管理员账号及其对应的密码，如若密码不正确或者忘记了密码，可通过[重置管理员密码](#)来修改密码。
- 证书名称  
SSL证书文件名，该文件需放在执行该命令的路径下。

**SSL公网连接集群示例：**`./mongo mongodb://rwuser:<password>@<DB_HOST>:<DB_PORT>/test?authSource=admin --ssl --sslCAFile <FILE_PATH> --sslAllowInvalidHostnames`

- 连接地址  
单击实例名称，在实例“连接管理”页面“公网连接”页签的公网连接地址处获取。
- 数据库端口  
目标实例的“连接管理”页面，“公网连接”页签的“数据库端口”。
- 用户名和密码  
root管理员账号及其对应的密码，如若密码不正确或者忘记了密码，可通过[重置管理员密码](#)来修改密码。
- 证书名称  
SSL证书文件名，该文件需放在执行该命令的路径下。

5. 排除网络不通

内网访问

- 检查ECS与DDS是否在同一个区域、同一个VPC。
  - 不同区域的云服务之间内网互不相通，无法访问实例。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
  - 不同VPC下，可以通过建立[VPC对等连接](#)实现网络互通。
- 检查安全组规则。  
安全组外访问安全组内的DDS实例时，需要为安全组[添加相应的入方向规则](#)。
- 在ECS上测试是否可以正常连接到DDS实例地址的端口。

```
telnet <实例地址> {8635}
```

  - 可以通信，说明网络是正常的。
  - 无法通信，您可以[提交工单](#)申请，完成工单提交。

公网访问

- 检查安全组规则。  
安全组外访问安全组内的DDS实例时，需要为安全组[添加相应的入方向规则](#)。
- 检查网络ACL规则。

- i. 进入虚拟私有云[网络ACL](#)列表。
- ii. 检查EIP绑定的网卡是否在网络ACL关联的子网下。
- iii. 查看网络ACL当前是“开启”状态还是“关闭”状态。

若网络ACL为“开启”状态，需要[添加ICMP放通规则进行流量放通](#)。

注意：网络ACL的默认规则是丢弃所有出入方向的包，关闭“网络ACL”后，其默认规则仍然生效。

- c. 相同区域主机进行ping测试。

如果在原ECS上没有ping通DDS实例绑定的EIP，请在相同区域的另一台ECS上去ping该EIP，如果可以正常ping通，说明虚拟网络正常，您可以[提交工单](#)申请，完成工单提交。

### 跨网段访问（配置IP映射）

通过内网连接副本集实例时，当客户端和副本集实例部署在不同网段时，且客户端所在的网段不在“192.168.0.0/16”，“172.16.0.0/24”和“10.0.0.0/8”时，需要进行跨网段访问配置，以实现网络互通。

- a. 确保在源端ECS网络与实例节点网络连通，如果网络不通，可以参考[对等连接](#)进行相关配置。
- b. 跨网段访问的操作请详见[跨网段访问配置](#)。

### 6. 排除实例的连接数满的情况

- a. 查看实例的连接数是否已达上限，请参见[如何查询及限制连接数](#)。  
若实例连接数已满请参见[实例的连接数满怎么处理](#)。
- b. 云监控服务目前可以监控数据库CPU、内存、磁盘、连接数等指标，并且设置告警策略，出现告警时可以提前识别风险。

### 7. 排除磁盘满的情况

**检查方法：**磁盘空间使用率可通过管理控制台或云监控服务查看。

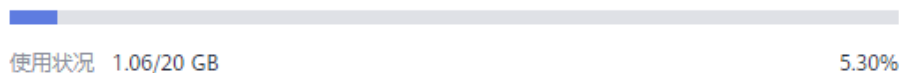
- 通过管理控制台查看

选择目标实例，单击实例名称，进入“基本信息”页面。在“存储空间”模块，查看磁盘使用率。

图 9-5 存储空间

#### 存储空间

超高IO 未加密



- 通过云监控服务查看

选择目标实例，单击“查看监控指标”，跳转到云监控页面，查看目标实例的磁盘使用量指标。

**可能原因：**当实例处于“磁盘空间满”状态时，需扩容至磁盘空间使用率小于80%才可使实例处于可用状态，使数据库恢复正常的写入操作。

**解决方案：**

- 随着业务数据的增加，原来申请的数据库磁盘容量可能会不足，建议用户扩容磁盘空间，确保磁盘空间足够。具体操作请参见[扩容磁盘](#)。
- 请及时处理过期数据文件。
- 云监控服务目前可以监控数据库CPU、内存、磁盘、连接数等指标，并且设置告警策略，出现告警时可以提前识别风险。

**8. 排除CPU使用率过高的情况**

CPU使用率很高或接近100%，会导致数据读写处理缓慢、无法获取连接、出现报错等，从而影响业务正常运行。

**解决方案：**请参见[如何排查DDS实例CPU使用率高的问题](#)。

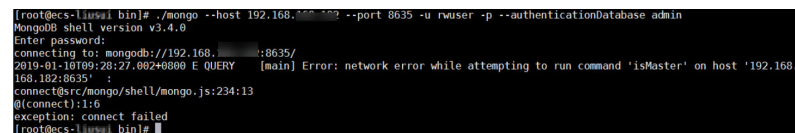
**9. 连接失败的常见报错**

- **连接失败，提示：network error while attempting to run command 'isMaster'**

使用如下命令连接DDS实例时报错。

```
./mongo --host 192.168.168.182 --port 8635 -u rwuser -p XXXXXXXXXXXX --  
authenticationDatabase admin
```

**图 9-6 连接失败**



```
[root@ecs-liauwai bin]# ./mongo --host 192.168.168.182 --port 8635 -u rwuser -p XXXXXXXXXXXX --  
authenticationDatabase admin  
MongoDB shell version v3.4.0  
Enter password:  
connecting to: mongod://192.168.168.182:8635/  
2019-01-18T09:28:27.882+0800 E QUERY [main] Error: network error while attempting to run command 'isMaster' on host '192.168.  
168.182:8635' :  
connect@src/mongo/shell/mongo.js:234:13  
@(connect):1:6  
exception: connect failed  
[root@ecs-liauwai bin]#
```

具体处理方法请参见[连接失败，提示：network error while attempting to run command 'isMaster'](#)。

- **连接失败，提示：No route to host以及connection attempt failed**

使用如下命令连接DDS实例时报错。

```
mongo --host 192.168.1.6 --port 8635 -u rwuser -p XXXXXXXXXXXX --  
authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --  
sslAllowInvalidHostnames
```

报错信息如下：

```
MongoDB shell version v3.4.17  
connecting to: mongod://192.168.1.6:8635/  
2019-09-19T09:38:36.954+0800 W NETWORK [thread1] Failed to connect to 192.168.1.6:8635,  
in(checking socket for error after poll), reason: No route to host  
2019-09-19T09:38:36.954+0800 E QUERY [thread1] Error: couldn't connect to server  
192.168.1.6:8635, connection attempt failed :  
connect@src/mongo/shell/mongo.js:240:13  
@(connect):1:6  
exception: connect failed
```

具体处理方法请参见[连接失败，提示：No route to host以及connection attempt failed](#)。

- **连接失败，提示：No route to host以及connection attempt failed**

使用如下命令连接DDS实例时报错。

```
mongo --host 192.168.168.116 --port 8635 -u rwuser -p XXXXXXXXXXXX --  
authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --  
sslAllowInvalidHostnames
```

报错信息如下：

```
MongoDB shell version v3.4.17
connecting to: mongodb://192.168.168.116:8635/
2019-09-19T09:39:24.306+0800 W NETWORK [thread1] The server certificate does not match
the host name. Hostname: 192.168.168.116 does not match CN: 172.16.2.65
MongoDB server version: 4.0.3
WARNING: shell and server versions do not match
2019-09-19T09:39:24.329+0800 E QUERY [thread1] Error: Authentication failed. :
DB.prototype._authOrThrow@src/mongo/shell/db.js:1461:20
@(auth):6:1
@(auth):1:2
exception: login failed
```

具体处理方法请参见[连接失败，提示：No route to host以及connection attempt failed](#)。

– **连接失败，提示：couldn't connect to server**

使用如下命令连接DDS实例时报错。

```
mongo --host 192.168.64.201 --port 8635 -u rwuser -p xxxxxxxxxx --
authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --
sslAllowInvalidHostnames
```

报错信息如下：

```
MongoDB shell version v3.4.17
connecting to: mongodb://192.168.64.201:8635/
2019-09-19T09:45:48.168+0800 W NETWORK [thread1] Failed to connect to
192.168.64.201:8635 after 5000ms milliseconds, giving up.
2019-09-19T09:45:48.168+0800 E QUERY [thread1] Error: couldn't connect to server
192.168.64.201:8635, connection attempt failed :
connect@src/mongo/shell/mongo.js:240:13
@(connect):1:6
exception: connect failed
```

具体处理方法请参见[连接失败，提示：couldn't connect to server](#)。

– **连接失败，提示：cannot list multiple servers in URL without 'replicaSet' option**

使用如下命令连接DDS副本集实例时报错。

```
./mongo mongodb://
rwuser:xxxxxxxxxx@192.168.168.116:8635,192.168.200.147:8635/test?
authSource=admin&replicaSet=replica
```

报错信息如下：

```
FailedToParse: Cannot list multiple servers in URL without 'replicaSet' option try 'mongo --help'
for more information.
```

使用如下命令连接兼容MongoDB 3.4版本的DDS实例时报错。

```
mongo "mongodb://
rwuser:xxxxxxxxxx@192.168.95.167:8635,192.168.92.43:8635/test?
authSource=admin"
```

报错信息如下：

```
FailedToParse: Cannot list multiple servers in URL without 'replicaSet' option
try 'mongo --help' for more information.
```

具体处理方法请参见[连接失败，提示：cannot list multiple servers in URL without 'replicaSet' option](#)。

– **Java驱动连接实例失败，提示：Timeout while receiving message**

使用Java驱动连接DDS实例时报错。

报错信息如下：

```
org.springframework.data.mongodb.UncategorizedMongoDbException: Timeout while receiving
message; nested exception is com.mongodb.MongoSocketReadTimeoutException: Timeout while
receiving message
```

具体处理方法请参见[Java驱动连接实例失败，提示：Timeout while receiving message](#)。

- 连接失败，提示：**exception: login failed 以及 U\_STRINGPREP\_PROHIBITED\_ERROR**

使用如下命令连接DDS实例时报错。

```
./mongo "mongodb://  
rwuser:xxxxxx@192.168.0.45:8635,192.168.0.96:8635/test?  
authSource=admin&replicaSet=replica"
```

报错信息如下：

```
MongoDB shell version v4.0.3  
connecting to: mongodb://192.168.0.45:8635,192.168.0.96:8635/test?  
authSource=admin&replicaSet=replica  
2021-11-05T05:52:53.717+0000 I NETWORK [js] Starting new replica set monitor for replica/  
192.168.0.45:8635,192.168.0.96:8635  
2021-11-05T05:52:53.718+0000 I NETWORK [ReplicaSetMonitor-TaskExecutor] Successfully  
connected to 192.168.0.45:8635 (1 connections now open to 192.168.0.45:8635 with a 5 second  
timeout)  
2021-11-05T05:52:53.718+0000 I NETWORK [js] Successfully connected to 192.168.0.96:8635 (1  
connections now open to 192.168.0.96:8635 with a 5 second timeout)  
Implicit session: session { "id" : UUID("5945d2a5-8275-4e3c-b06f-632f062a2ead") }  
MongoDB server version: 4.0.3  
2021-11-05T05:52:53.722+0000 I NETWORK [js] Marking host 192.168.0.96:8635 as failed ::  
caused by :: Location50692: can't authenticate against replica set node 192.168.0.96:8635 ::  
caused by :: Error preflighting normalization: U_STRINGPREP_PROHIBITED_ERROR  
2021-11-05T05:52:53.722+0000 I NETWORK [js] Successfully connected to 192.168.0.45:8635 (1  
connections now open to 192.168.0.45:8635 with a 0 second timeout)  
2021-11-05T05:52:53.723+0000 I NETWORK [js] Marking host 192.168.0.45:8635 as failed ::  
caused by :: Location50692: can't authenticate against replica set node 192.168.0.45:8635 ::  
caused by :: Error preflighting normalization: U_STRINGPREP_PROHIBITED_ERROR  
2021-11-05T05:52:53.724+0000 I NETWORK [js] Marking host 192.168.0.96:8635 as failed ::  
caused by :: Location50692: can't authenticate against replica set node 192.168.0.96:8635 ::  
caused by :: Error preflighting normalization: U_STRINGPREP_PROHIBITED_ERROR  
2021-11-05T05:52:53.725+0000 I NETWORK [js] Marking host 192.168.0.45:8635 as failed ::  
caused by :: Location50692: can't authenticate against replica set node 192.168.0.45:8635 ::  
caused by :: Error preflighting normalization: U_STRINGPREP_PROHIBITED_ERROR  
2021-11-05T05:52:53.725+0000 E QUERY [js] Error: can't authenticate against replica set node  
192.168.0.45:8635 :: caused by :: Error preflighting normalization:  
U_STRINGPREP_PROHIBITED_ERROR :  
DB.prototype._authOrThrow@src/mongo/shell/db.js:1685:20  
@(auth):6:1  
@(auth):1:2  
exception: login failed
```

具体处理方法请参见[连接失败，提示：exception: login failed 以及 U\\_STRINGPREP\\_PROHIBITED\\_ERROR](#)。

10. 如果上述方法均不能解决您的疑问，您可以[提交工单](#)申请，完成工单提交，为您解答。

## 9.2 实例的连接数满导致实例连接失败，如何处理？

数据库连接数表示应用程序可以同时连接数据库的数量，与您应用程序或者网站能够支持的最大用户数没有关系。

- 对于集群实例，一般指客户端同dds mongos之间的连接数。
- 对于副本集实例，一般指客户端同Primary节点和Secondary节点之间的连接数。
- 对于单节点实例，一般指客户端同单节点之间的连接数。

当DDS实例的连接数已满时，新发起的连接请求将无法被响应，从而导致实例连接失败。

## 问题现象

以下为常见的几种报错情况：

- 使用Mongo Shell连接实例时，出现如下提示，表示当前连接池的连接数已满。

图 9-7 提示信息

```
[root@623- ~]# mongo "mongodb://rwuser:~@192.168.0.180:8635,192.168.0.50:8635/test?authSource=admin&replicaSet=replica"
MongoDB shell version v3.4.17
connecting to: mongodb://rwuser:623.Huawei@192.168.0.180:8635,192.168.0.50:8635/test?authSource=admin&replicaSet=replica
2019-10-16T17:43:45.203+0800 I NETWORK [thread1] Starting new replica set monitor for replica/192.168.0.180:8635,192.168.0.50:8635
2019-10-16T17:43:45.205+0800 W NETWORK [ReplicaSetMonitor-TaskExecutor-0] No primary detected for set replica
2019-10-16T17:43:45.205+0800 I NETWORK [ReplicaSetMonitor-TaskExecutor-0] All nodes for set replica are down. This has happened for 1 check
s in a row.
2019-10-16T17:43:45.708+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:45.708+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 2 checks in a row.
2019-10-16T17:43:46.210+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:46.210+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 3 checks in a row.
2019-10-16T17:43:46.712+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:46.712+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 4 checks in a row.
2019-10-16T17:43:47.215+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:47.215+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 5 checks in a row.
2019-10-16T17:43:47.717+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:47.717+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 6 checks in a row.
2019-10-16T17:43:48.218+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:48.218+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 7 checks in a row.
2019-10-16T17:43:48.721+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:48.721+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 8 checks in a row.
2019-10-16T17:43:49.222+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:49.222+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 9 checks in a row.
2019-10-16T17:43:49.724+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:49.724+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 10 checks in a row.
2019-10-16T17:43:50.226+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:50.226+0800 I NETWORK [thread1] All nodes for set replica are down. This has happened for 11 checks in a row.
2019-10-16T17:43:50.727+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:51.230+0800 W NETWORK [thread1] No primary detected for set replica
2019-10-16T17:43:51.731+0800 W NETWORK [thread1] No primary detected for set replica
```

- 使用Python连接实例时，出现以下提示，表示当前连接池的连接数已满。  
pymongo.errors.ServerSelectionTimeoutError: connection closed, connection closed

## 处理思路

- 请及时排查业务侧连接是否有效，优化实例连接，释放不必要的连接。
- 排查“net.maxIncomingConnections”参数的值及实例规格，修改参数值或对数据库进行规格扩容。
- 云监控服务目前可以监控数据库CPU、内存、磁盘、连接数等指标，并且设置告警策略，出现告警时可以提前识别风险。具体请参考[云监控服务用户指南](#)。

## 处理方法

- 优化实例连接，释放不必要的连接。
  - 您可以通过重启实例来临时释放所有的连接。具体请参见[重启实例或节点](#)。
  - 您可以查询节点当前连接数，以及当前连接来源，分析各个终端和DDS实例分别建立了多少连接，并做出相应调整，具体请参见[如何查询及限制连接数](#)。
- 修改参数值或对数据库进行规格扩容。

实例的最大连接数可通过修改参数“net.maxIncomingConnections”的值进行调整。如何修改参数值请参见[编辑参数模板](#)。

  - 如果参数值是“default”，表示最大连接数为缺省值，并且与实例的规格相关，具体请参见[数据库实例规格](#)。
  - 如果调整连接数过大，可能导致服务崩溃，这时只能通过变更实例规格增加承载的连接数的数量，具体请参见[变更实例规格](#)。

### 说明

如果实例参数模板为默认参数模板，由于默认参数模板中不允许修改参数值，您可以创建一个参数模板，并修改对应参数值，修改后将新参数模板关联到该实例。

3. 检查是否有慢查询，建议添加索引进行优化。

## 9.3 如何查询及限制连接数

下面以副本集实例为例，介绍如何查询实例的连接使用情况，以及设置连接池的连接数。

### 查询当前连接数

根据您购买的文档数据库实例规格不同，最大连接数也不同。

#### 说明

最大连接数是指实例中每个节点的最大连接数。

示例：您购买了2核4GB规格的三节点副本集实例，那么该实例的Primary节点和Secondary节点的最大连接数均为1000，Hidden节点由于其架构特殊性，不对外提供服务。

使用Mongo Shell连接副本集主节点后，执行命令`db.serverStatus().connections`查询节点当前连接数。

```
replica:PRIMARY> db.serverStatus().connections
{ "current" : 7, "available" : 398, "totalCreated" : 818364 }
```

您需要关注以下参数及对应的值：

- current：当前已经建立的连接数。
- available：当前可用的连接数。

### 查询当前连接来源

**步骤1** 使用Mongo Shell连接副本集主节点后，切换至admin数据库。

```
replica:PRIMARY> use admin
```

**步骤2** 执行命令`db.runCommand({currentOp: 1, $all: true})`，查询连接来源。

通过分析命令的输出结果，您可以查询每个连接对应的来源IP地址。从而得出各个终端和DDS实例分别建立了多少连接。

图 9-8 输出结果

```
replica:PRIMARY> db.runCommand({currentOp: 1, $all: true})
{
  "inprog" : [
    {
      "desc" : "conn828171",
      "threadId" : "139911043778304",
      "connectionId" : 828171
      "client" : "192.168.1.100:58096",
      "appName" : "MongoDB Shell",
      "clientMetadata" : {
        "application" : {
          "name" : "MongoDB Shell"
        },
        "driver" : {
          "name" : "MongoDB Internal Client",
          "version" : "3.4.17"
        },
        "os" : {
          "type" : "Linux",
          "name" : "CentOS Linux release 7.4.1708 (Core) ",
          "architecture" : "x86_64",
          "version" : "Kernel 3.10.0-693.11.1.el7.x86_64"
        }
      },
      "active" : true,
      "opid" : 3719193,
      "secs_running" : 0,
      "microsecs_running" : NumberLong(30),
      "op" : "command",
      "ns" : "admin.$cmd",
      "query" : {
        "currentOp" : 1,
        "$all" : true
      },
      "numYields" : 0,
      "locks" : {
      },
      "waitingForLock" : false,
      "lockStats" : {
      }
    }
  ]
}
```

----结束

## 如何限制终端的连接数

文档数据库服务支持通过Connection String URI登录数据库。通过Connection String URI登录数据库时，在URI末尾加上“&maxPoolSize=<integer>”，即可设置连接池的连接数。

示例：使用Mongo Shell连接副本集实例时，限制连接池的连接数为10。

```
mongo "mongodb://
rwuser:xxxxxxxxxx@192.168.168.116:8635,192.168.200.147:8635/test?
authSource=admin&replicaSet=replica&maxPoolSize=10"
```

图 9-9 限制连接数

```
root@servercedea23-0d5f-49e6-8a8c-94259dcf09cb ycsb]# mongo "mongodb://rwuser:xxxxxxxxxx@192.168.168.116:8635,192.168.200.147:8635/test?authSource=admin&replicaSet=replica&maxPoolSize=10"
MongoDB shell version v3.4.17
connecting to: mongodb://rwuser:xxxxxxxxxx@192.168.168.116:8635,192.168.200.147:8635/test?authSource=admin&replicaSet=replica&maxPoolSize=10
2019-09-19T11:13:25.634+0800 I NETWORK [thread1] Starting new replica set monitor for replica/192.168.168.116:8635,192.168.200.147:8635
2019-09-19T11:13:25.654+0800 I NETWORK [thread1] Successfully connected to 192.168.85.63:8635 (1 connections now open to 192.168.168.116:8635 with a 5 second timeout)
2019-09-19T11:13:25.655+0800 I NETWORK [ReplicasetMonitor-taskexecutor-0] Successfully connected to 192.168.56.158:8635 (1 connections now open to 192.168.168.116:8635 with a 5 second timeout)
MongoDB server version: 3.4.14
replica:PRIMARY>
```

### 说明

关于不同语言的客户端如何限制连接池的数量，请参考[MongoDB官网](#)各语言客户端的API文档。

## 9.4 ECS 和 DDS 署在不同的 VPC，网络不通怎么办

- 参考[切换ECS虚拟私有云](#)，将ECS的虚拟私有云切换为与DDS相同的虚拟私有云。
- 参考[VPC对等连接](#)，为两个不同的虚拟私有云建立对等连接，实现内网互通。

## 9.5 应用程序是否需要支持自动重连数据库

建议您的应用程序支持自动重连数据库，当数据库重启后，无需人工介入，应用会自动恢复，提供更高的可用性。

同时建议您的应用程序采用长连接方式连接数据库，以降低资源消耗，提高性能。

## 9.6 如何创建并登录弹性云服务器

创建并登录弹性云服务器，请参见《弹性云服务器快速入门》中“[购买弹性云服务器](#)”和“[登录弹性云服务器](#)”。

- 该弹性云服务器用于连接文档数据库实例，需要与待连接的实例处于同一虚拟私有云子网内。
- 创建弹性云服务器时，要选择操作系统，例如Redhat6.6，并为其绑定EIP（Elastic IP，弹性公网IP）。
- 正确配置安全组，使得弹性云服务器可以通过文档数据库服务“基本信息”页面下节点的“内网IP”访问文档数据库实例。

# 10 客户端安装

## 10.1 如何安装 MongoDB 客户端

MongoDB官网提供了针对不同操作系统的客户端安装包，其二进制安装包下载页面链接为：<https://www.mongodb.com/try/download/community>。

### 须知

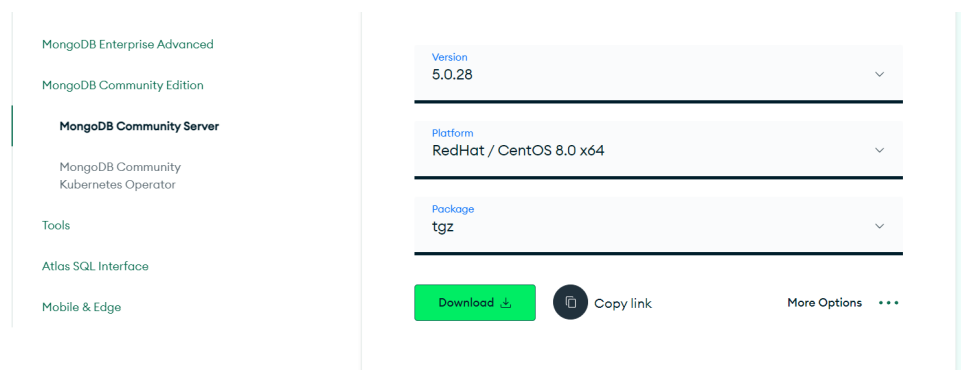
- 本章节以“RedHat/CentOS 8.0 x64”和MongoDB 5.0.28为例，介绍如何获取相应安装包并完成安装。
- 在您实际安装过程中，请根据实际所用操作系统选择与实例版本相匹配的客户端版本。

### 操作步骤

#### 步骤1 获取安装包。

1. 进入官网下载链接地址：<https://www.mongodb.com/try/download/community>。
2. 在“Version”中选择“5.0.28”，在“Platform”中选择“RedHat/CentOS 8.0 x64”，在“Package”中选择“tgz”。如图10-1所示。

图 10-1 MongoDB 官网页面



3. 将安装包上传到弹性云服务器上，请参见如下两种方式：

#### 说明

如何登录弹性云服务器，请参见[如何创建并登录弹性云服务器](#)。

- 单击“Download”下载5.0.28版本的二进制安装包，安装包名称为“mongodb-linux-x86\_64-rhel80-5.0.28.tgz”，将安装包上传至弹性云服务器上。
- 单击“Copy link”获取下载链接地址，登录弹性云服务器，执行`wget copylink`。

#### 说明

`copylink`请替换为实际下载的连接地址。

- 步骤2** 在弹性云服务器上，解压安装包。

```
tar zxvf mongodb-linux-x86_64-rhel80-5.0.28.tgz
```

#### 说明

安装包名字根据实际情况替换。

- 步骤3** 进入安装包的“bin”文件夹下，获取客户端工具。

```
cd mongodb-linux-x86_64-rhel80-5.0.28/bin
```

#### 说明

安装包名字根据实际情况替换。

其中，常用工具包含如下：

- MongoDB客户端mongo。
- 数据导出工具mongoexport。
- 数据导入工具mongoimport。

- 步骤4** 使用客户端工具前，需要对工具赋予执行权限。

- 执行`chmod +x mongo`，赋予连接实例的权限。
- 执行`chmod +x mongoexport`，赋予导出数据数据的权限。
- 执行`chmod +x mongoimport`，赋予导入数据的权限。

- 步骤5** 客户端安装成功后，如需连接实例，请参见[连接集群实例](#)、[连接副本集实例](#)和[连接单节点实例](#)。

----结束

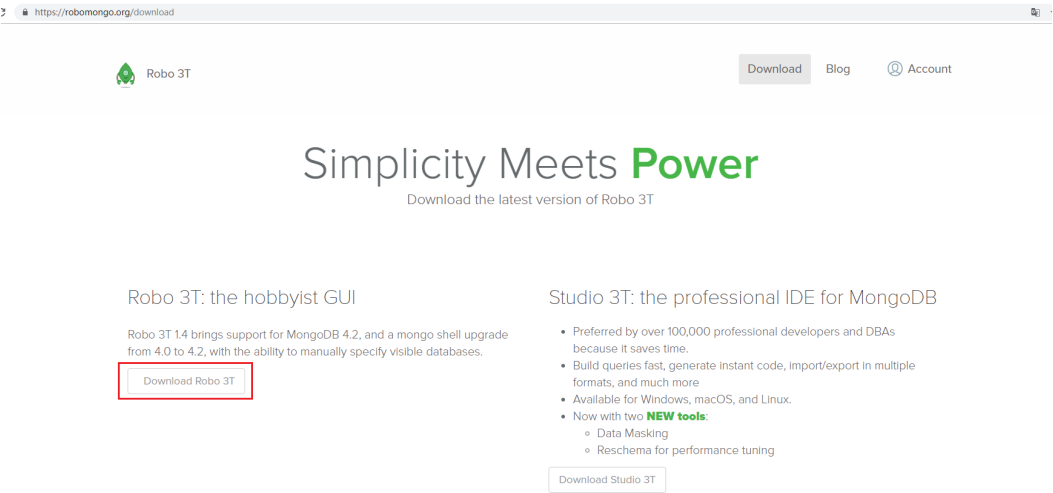
## 10.2 如何安装 Robo 3T 工具

本章节介绍如何获取Robo 3T工具的安装包并完成安装。

### 操作步骤

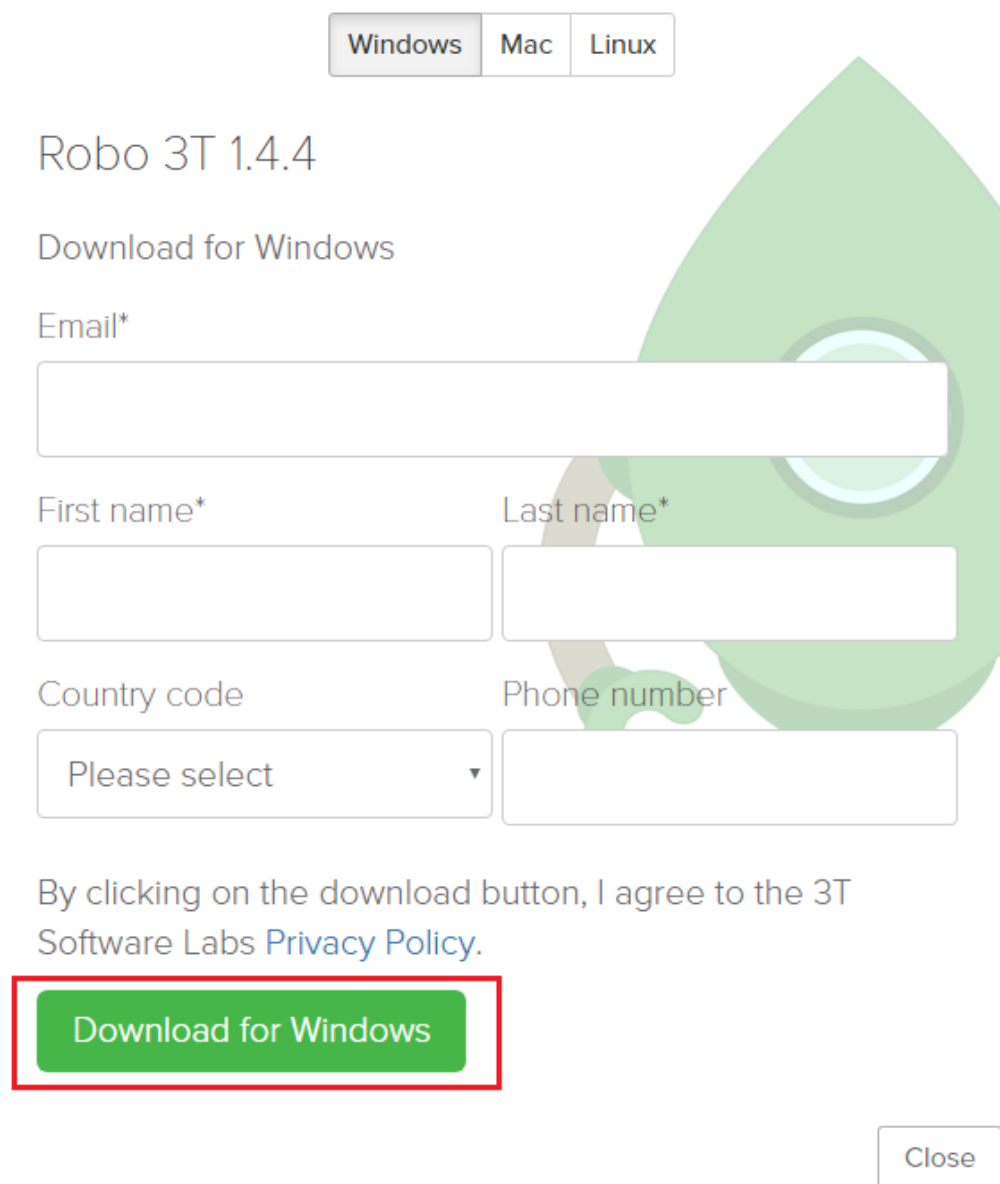
- 步骤1** 打开Robo 3T下载地址：<https://robomongo.org/download>，单击“Download Robo 3T”。

图 10-2 下载页面



**步骤2** 在弹出框中填写相关信息后，单击“Download for Windows”下载“robo3t-1.4.4-windows-x86\_64-e6ac9ec5.zip”。

图 10-3 下载工具

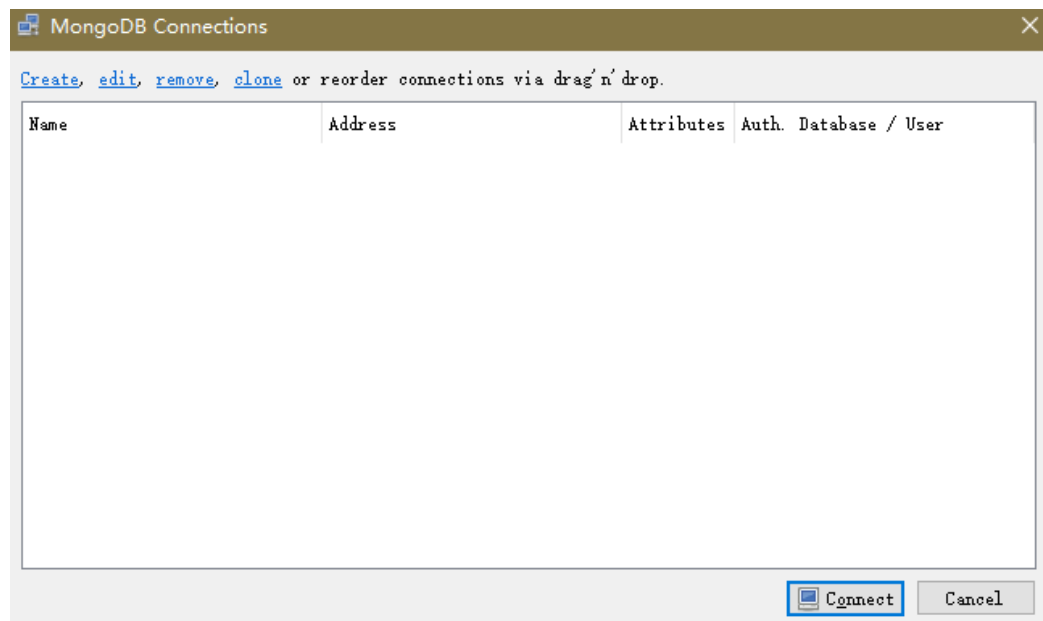


The image shows a web form for downloading Robo 3T 1.4.4. At the top, there are three tabs: 'Windows' (selected), 'Mac', and 'Linux'. Below the tabs, the text 'Robo 3T 1.4.4' is displayed. Underneath, it says 'Download for Windows'. The form includes several input fields: 'Email\*', 'First name\*', 'Last name\*', 'Country code' (a dropdown menu with 'Please select' as the current value), and 'Phone number'. A green button labeled 'Download for Windows' is highlighted with a red rectangle. Below the button, there is a text line: 'By clicking on the download button, I agree to the 3T Software Labs [Privacy Policy](#).' At the bottom right, there is a 'Close' button.

**步骤3** 解压**步骤2**下载的压缩包文件，双击解压目录下的“robo3t.exe”文件，开始执行安装。

**步骤4** 安装完成后，打开工具，界面展示如**图10-4**所示。

图 10-4 工具主界面



**步骤5** 工具安装成功后，如需连接实例，请参见《文档数据库服务快速入门》中各实例类型下通过公网连接实例的内容。

----结束

# 11 数据库基本使用

## 11.1 如何查看当前副本集实例的主备节点

您可以在DDS控制台查看当前实例的节点信息，具体操作如下：

1. 登录文档数据库服务控制台。
2. 在“实例管理”页面，单击目标实例名称。

图 11-1 实例管理



3. 在左侧导航树，单击“连接管理”，右侧界面可以查看节点信息。

图 11-2 连接管理



## 11.2 如何查询每个表的数据量 mongodb 存储空间分析?

通过如下脚本可以进行存储空间分析。

```
db.adminCommand({ listDatabases: 1 }).databases.forEach(function (_db) {
  print("\nDatabase: " + _db.name);
  // 获取数据库中所有集合的信息
  var collections = db.getSiblingDB(_db.name).getCollectionInfos();
  // 创建一个数组来存储集合名称和大小
  var collectionSizes = [];
  // 遍历所有集合
  collections.forEach(function (collection) {
    // 获取集合名称
    var name = collection.name;
    // 获取集合中的文档数量
    var size = db.getSiblingDB(_db.name).getCollection(name).stats().size;
    // 将集合名称和大小添加到数组中
    collectionSizes.push({ name: name, size: size });
  });
  // 按size排序
  collectionSizes.sort(function (a, b) {
    return a.size - b.size;
  });
  // 输出排序后的结果
  collectionSizes.forEach(function (item) {
    print("\t" + item.name + "\t\t" + item.size + " (bytes)");
  });
});
```

## 11.3 文档数据库中创建失败的索引可以删除吗

使用语句`db.<collection>.stats()`，查询文档数据库中有哪些索引，然后对创建失败的索引进行删除。

### 说明

collection请根据需要进行填写。

## 11.4 DDS 是否支持 TTL 索引清理过期数据

DDS支持TTL索引清理过期数据。详情请参考[官方指南](#)。

## 11.5 如何使用 MapReduce 命令

### 命令含义

对大数据集执行map-reduce操作。

### 如何启用 MapReduce 命令

MapReduce命令由DDS参数模板参数“security.javascriptEnabled”控制，默认值为“false”，表示**mapreduce**和**group**命令将无法使用。如果需要使用MapReduce命令，将参数值修改为“true”，修改后需要重启实例才生效。

- 对于集群实例，需要变更全部shard节点关联参数模板中对应参数值，再重启实例生效。
- 对于副本集或单节点实例，直接变更实例关联参数模板中对应参数值，再重启实例生效。

如何变更参数值请参见[编辑参数模板](#)。

## 使用 MapReduce 命令失败时常见报错处理

报错信息：cannot run map reduce without the js engine或map is not defined

图 11-3 案例一

```
replica:PRIMARY> use 1016;
switched to db 1016
replica:PRIMARY> db.lmktable.insert({"post_text": "testllllllllllllllllllllllllllllll", "user_name": "mark", "status": "active"});
{"lmktable_temp": {"nInserted": 1}}
replica:PRIMARY> db.lmktable.mapReduce(function(){emit(this.user_name,1)}),function(key, values) {return Array.sum(values)},{"query":{"status":"active"},"out":"lmktable_temp"});
2019-10-16T03:35:31.277+0000 E QUERY [js] Error: map reduce failed:{"
  "ok" : 0,
  "errmsg" : "Cannot run map reduce without the js engine",
  "code" : 16149,
  "codeName" : "Location16149"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DBCollection.prototype.mapReduce@src/mongo/shell/collection.js:1145:1
@shell:1:1
```

图 11-4 案例二

```
replica:PRIMARY>
replica:PRIMARY> db.mythings.find( { age: { $lt: 25 } } )
replica:PRIMARY> db.runCommand(
... {
... mapreduce: 'mythings'
... ,map: map
... ,reduce: reduce
... ,out: 'a'
... ,keepTemp: false
... }
... )
2019-10-16T03:38:30.420+0000 E QUERY [js] ReferenceError: map is not defined :
@shell:4:2
```

可能原因：MapReduce命令被限制不能使用。

处理方法：将实例参数模板中参数“security.javascriptEnabled”的值修改为“true”，并重启实例生效，即可使用该命令。

### 📖 说明

如果实例参数模板为默认参数模板，由于默认参数模板中不允许修改参数值，您可以创建一个参数模板，并修改对应参数值，修改后将新参数模板关联到该实例，具体请参见[应用参数模板](#)。

## 11.6 DDS 是否支持 \$round 函数

DDS支持\$round函数。详情请参见[聚合函数兼容性列表](#)。

## 11.7 如何管理均衡器 Balancer

### 使用场景

因为Balancer在进行块迁移时会占用节点的系统资源，可能导致集群的读写性能出现一定程度的下降，因此如果您已开启数据分片并且对实例性能有较高要求或者业务流量有比较明显的高低峰差异时，为避免影响业务：

- 建议在业务低峰期时**开启Balancer**。
- 建议在业务高峰期时**关闭Balancer**。
- 建议**设置Balancer的活动窗口**，让其在业务低峰期工作。

### 使用限制

- 只有集群实例支持Balancer功能管理操作。
- Balancer功能默认是开启状态。

### 开启 Balancer

**步骤1** 通过Mongo Shell连接集群实例。

**步骤2** 在dds mongos节点命令行中，切换至config数据库。

```
use config
```

**步骤3** 执行如下命令，开启Balancer功能。

```
sh.setBalancerState(true)
```

----结束

### 关闭 Balancer

**步骤1** 通过Mongo Shell连接集群实例。

**步骤2** 在dds mongos节点命令行中，切换至config数据库。

```
use config
```

**步骤3** 执行如下命令，查看Balancer运行状态。

```
while( sh.isBalancerRunning() ) {  
    print("waiting...");  
    sleep(500);  
}
```

- 如返回值为空，则表示Balancer没有处于执行任务的状态，此时可执行下一步的操作，关闭Balancer。
- 如返回值为waiting，则表示Balancer正在执行块迁移，此时不能执行关闭Balancer的命令，否则可能引起数据不一致。

**步骤4** 如执行**步骤3**的命令后返回的值为空，可执行关闭Balancer命令。

```
sh.stopBalancer()
```

----结束

## 设置 Balancer 的活动窗口

为避免迁移给业务带来影响，您可以通过该操作使Balancer在规定时间内工作。

### 说明

处于关闭状态的Balancer无法设置活动窗口。所以设置之前，请先[开启Balancer](#)。

**步骤1** 通过Mongo Shell连接集群实例。

**步骤2** 在dds mongos节点命令行中，切换至config数据库。

```
use config
```

**步骤3** 执行如下命令，设置Balancer的活动窗口。

```
db.adminCommand({"setBalancerActiveWindow": 1, start : "<start-time>", stop : "<stop-time>"})
```

### 说明

- <start-time>：开始时间，时间格式为HH:MM（UTC时间），HH取值范围为00 - 23，MM取值范围为00 - 59。
- <stop-time>：结束时间，时间格式为HH:MM（UTC时间），HH取值范围为00 - 23，MM取值范围为00 - 59。
- 使用rwuser执行setBalancerActiveWindow调整balancer时间窗口，如出现“CommandNotFound”报错，其原因是小版本低。发生此情况请升级实例小版本后尝试，或您可以[提交工单](#)申请，完成工单提交。

执行sh.status()命令，查看Balancer的活动窗口。如图11-5所示，活动窗口被设置为22:00 - 24:00。

图 11-5 查看 Balancer 的活动窗口

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("64478ef7d201d17f7beb47c7")
  }
  shards:
    { "_id" : "shard_1", "host" : "shard_1/172.16.29.203:8637,172.16.61.17:8637", "state" : 1 }
    { "_id" : "shard_2", "host" : "shard_2/172.16.27.2:8637,172.16.47.171:8637", "state" : 1 }
  active mongoses:
    "4.0.3" : 2
  autosplit:
    Currently enabled: yes
  balancer:
    Currently enabled: no
    Currently running: no
    Balancer active window is set between 22:00 and 24:00 server local time
```

----结束

## 去除 Balancer 的活动窗口

如需Balancer始终处于运行状态，请按照如下命令设置：

```
db.adminCommand({ 'unsetBalancerActiveWindow' : 1 })
```

# 12 数据库迁移

## 12.1 DDS 实例是否支持跨 region 迁移

您可以使用数据复制服务（Data Replication Service，简称DRS）实现云上跨Region的数据库之间的迁移。

具体迁移操作，请参见“[通过数据复制服务迁移数据](#)”内容。

## 12.2 不同账号下的 DDS 数据库如何进行迁移？

用户需要将A账号的DDS-A实例迁移到B账号的DDS-B，需要通过数据复制服务DRS来实现，详情请参见[DRS支持跨账号云数据库迁移吗](#)。

## 12.3 高版本 MongoDB 迁移至低版本时，兼容度如何评估？

文档数据库作为一种数据库软件类云服务与传统软件一样，高版本兼容低版本的功能，少数情况出现功能断代时官方资料会给予专门的总结和公布。当从高版本迁移至低版本时意味着高版本出现的新功能在低版本中均不支持，所以需要您侧重评估在业务中是否使用了这类新功能。

评估方式通常为预估建议和演练建议：

- 预估建议：通过查询社区文档版本发布说明，可以查看新版本发布时的新增特性，进行逐一分析判断是否在业务代码中使用该特性；
- 演练建议：通过[华为云数据复制服务](#)或者社区提供的[通过mongodump和mongorestore工具](#)迁移数据至低版本，进行业务功能模拟测试，通过观察业务功能实际表现验证迁移至低版本是否可行；

如果您的业务较为复杂，您可以[提交工单](#)申请，完成工单提交，寻求专业的评估支撑。

# 13 数据库存储

## 13.1 DDS 使用的存储配置是什么

文档数据库服务存储采用云硬盘，具体情况请参考《[云硬盘用户指南](#)》。

文档数据库服务的备份数据存储采用对象存储服务，不占用用户购买的数据库空间。  
关于文档数据库实例存储的硬件配置，请参见《[对象存储服务用户指南](#)》。

## 13.2 数据超过了文档数据库实例的最大存储容量怎么办

占用存储空间的文件请参考[哪些数据会占用文档数据库实例空间](#)。

如果您的应用程序所需的存储容量超过最大分配量，可以通过[扩容存储](#)来增加存储容量。

若是集群实例，还可以增加shard节点，具体操作请参见[添加节点](#)。

## 13.3 什么是 DDS 实例的只读状态？

为保护DDS实例的正常运行，在存储空间即将被写满的时候，数据库会被设置成磁盘满只读状态，该状态下只支持数据的读取，不支持写入和更新，可以通过[扩容存储](#)空间来解除该状态，重新获得写入和更新的能力。

具体规则如下：

- 用户购买的存储空间超过600GB，当剩余的可用存储空间为18GB时，实例状态将被设置为只读。
- 用户购买的存储空间小于等于600GB，当实际使用量到总存储空间的97%时，实例状态将被设置为只读。
- 用户购买的存储空间超过600GB，当剩余的可用存储空间大于90GB时，只读状态会自动解除。
- 用户购买的存储空间小于等于600GB，当实际使用量小于总存储空间的85%时，只读状态会自动解除。

此时建议您清理多余的资源或进行存储空间扩容。

# 14 数据库参数

## 14.1 DDS 支持修改时区吗

服务器时间为UTC时间，不支持修改服务器时间，建议您从应用上控制数据的插入时间。

## 14.2 用户需要关注实例的哪些参数

参数相关说明如下：

- 各参数的详细说明请参见[MongoDB官网](#)。
- “net.maxIncomingConnections” 依赖于实例的规格，实例规格不同对应其默认值也不同，因此，此参数在用户未设置前显示为“default”。
- “disableJavaScriptUIT” 和 “security.javascriptEnabled” 配合使用，设置统计函数功能。
  - “disableJavaScriptUIT”，默认值为“true”，表示禁用JavaScriptUIT编译技术。
  - “security.javascriptEnabled”，默认值为“false”，表示不允许mongod上执行javascript脚本，mapreduce、group等命令将无法使用。

## 14.3 MapReduce 提示不支持 JavaScript

为了安全考虑，DDS默认限制了JavaScript脚本的执行。但是如果您需要执行JavaScript，可以通过DDS控制台将“**security.javascriptEnabled**”参数值改为“true”即可。具体操作方法请参见[编辑参数模板](#)。

## 14.4 DDS 是否支持设置 majority 级别的 readConcern

MongoDB可以通过writeConcern来定制写策略，通过readConcern来定制读策略。

当指定readConcern级别为majority时，能保证用户读到的数据已经写入到大多数节点，而这样的数据不会发生回滚，避免了脏读的问题。

DDS目前不支持设置majority级别的readConcern。

如果有该需求的场景，可以将WriteConcern设置为majority，表示将数据写入到大多数节点，这样就保证了大多数节点的数据一致。然后通过读取单个节点数据就能保证用户读到的数据已经写入到大多数节点，而这样的数据不会发生回滚，避免了脏读的问题。

# 15 备份与恢复

## 15.1 DDS 实例删除后，手动备份是否继续收费

文档数据库实例删除后，手动备份默认为您保留，并按照对象存储服务的计费标准收取相应的费用。

## 15.2 如何将文档数据库备份到弹性云服务器上

您可以通过[mongoexport的方式](#)将数据库备份到弹性云服务器上，但不建议将弹性云服务器作为数据库备份空间使用。

强烈推荐使用文档数据库实例的[备份功能](#)，将备份数据存放到专业的对象存储服务上，以获得更高的数据可靠性和服务保障。

## 15.3 文档数据库服务能够保存多长时间的备份

文档数据库服务自动备份可保留天数默认7天，用户后期可进行修改，可设置范围为1~732天。手动备份没有时间限制，且用户可根据需要删除该备份文件。

## 15.4 如何找回 DDS 丢失的备份数据

如果未开通云审计服务，则无法查询对应的操作记录。

建议您开通云审计服务，可以在云审计服务页面查找对应的备份删除记录。详情请参见[查看追踪事件](#)。

## 15.5 备份空间里面包含哪些文件

备份空间包含自动备份策略产生的自动备份文件，手动备份文件和增量日志备份文件。

## 15.6 DDS 的备份是如何收费的

文档数据库服务所有的全量备份和增量备份都存储在对象存储服务上，不占用用户购买实例的存储空间，DDS提供了和实例磁盘大小相同的部分免费存储空间，用于存放您的备份数据。

自动备份的生命周期和实例的生命周期相同。如果实例删除，那么自动备份也会同时被删除，如果您有手动备份，手动备份不会自动删除，具体请参见[删除手动备份](#)。

举个例子，假如用户购买实例时，存储空间选择了200GB，那么所有的备份空间合计，只有超过200GB才开始计费，前200GB是免费的。对于超过免费容量的存储空间，目前仅支持按需计费，用多少收取多少费用，具体收费请参见[计费说明](#)。

---

### 须知

免费的存储空间是在收取了数据盘的存储空间费用后赠送的，数据盘存储空间冻结以后不再收费，因此不再享受备份赠送空间。

用户实例冻结后，将没有免费的存储空间，会导致实例原有备份收费。

- 如果选择解冻实例，将恢复免费的存储空间。
  - 如果直接删除冻结的实例，原有的自动备份将会同时被删除，原有的手动备份需要手动删除，删除后备份空间不会继续收费。
-

# 16 网络安全

## 16.1 文档数据库服务有哪些安全保障措施

文档数据库服务可设置所属虚拟私有云，从而确保实例与其它业务实现网络安全隔离。另外，通过统一身份认证服务，可以实现对文档数据库服务资源的访问权限控制。

## 16.2 为什么在虚拟私有云中使用文档数据库

虚拟私有云允许您在私有、隔离的网络创建虚拟网络环境，从中可以对私有IP地址范围、子网、路由表和网络网关等方面进行控制。借助虚拟私有云，您可以定义虚拟网络拓扑和网络配置，使其非常类似于您在自己的数据中心所操作的传统IP网络。

在下面的情况下您可能想在虚拟私有云中使用文档数据库服务：

您希望运行面向公众的Web应用程序，同时保留公众不可访问的后端服务器。您可以在一个虚拟私有云内同时创建一个文档数据库实例和弹性云服务器实例，同时为弹性云服务器实例分配公网IP，将Web服务器部署在弹性云服务器实例。

## 16.3 如何确保在虚拟私有云中运行的文档数据库的安全

虚拟私有云安全组可用来帮助确保虚拟私有云内文档数据库实例的安全。此外，通过网络访问控制列表（ACL），可以允许或拒绝进入和退出各个子网的网络流量。内部安全基础设施（包括网络防火墙、入侵检测和防护系统）可以监视通过IPsec虚拟专用网络连接进入或退出虚拟私有云的所有网络流量。

## 16.4 DDS 是否支持选择创建 IPV6 子网

目前已支持选择IPV6网段的子网，该功能当前处于公测阶段，如需使用，请[提交工单](#)申请开通。

## 16.5 将根证书导入 Windows/Linux 操作系统

### 导入 Windows 操作系统

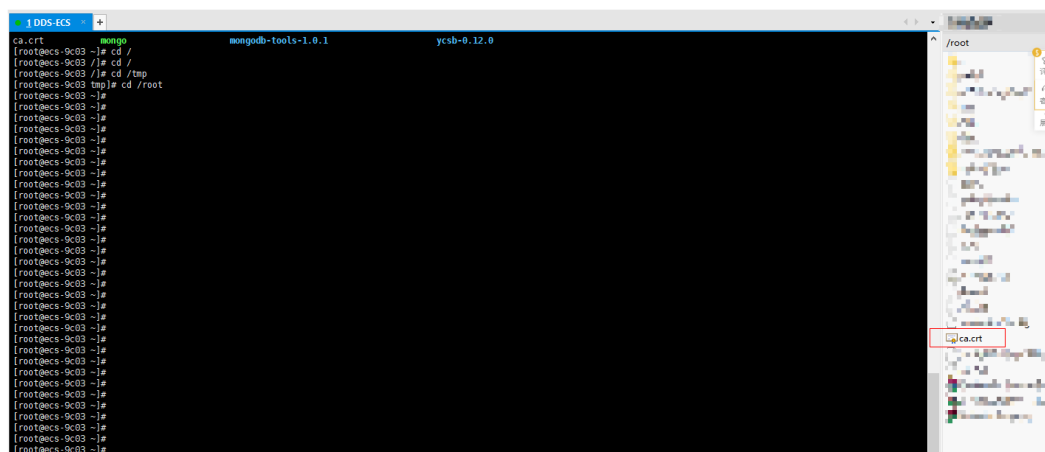
1. 单击“开始”，运行框输入“MMC”，回车。
2. 在MMC控制台菜单栏中单击“文件”，选择“添加/删除管理单元”。
3. 在“添加或删除管理单元”对话框，选择“可用的管理单元”区域的“证书”。单击“添加”添加证书。
4. 在“证书管理单元”对话框，选择“计算机账户”，单击“下一步”。
5. 在“选择计算机”对话框，单击“完成”。
6. 在“添加或删除管理单元”对话框，单击“确定”。
7. 在MMC控制台，双击“证书”。
8. 右键单击“受信任的根证书颁发机构”，选择“所有任务”，单击“导入”。
9. 单击“下一步”。
10. 单击“浏览”。
11. 找到下载的根证书ca.crt文件，单击“打开”，然后在向导中单击“下一步”。
12. 单击“下一步”。
13. 单击“完成”。
14. 单击“确定”，完成根证书导入。

### 导入 Linux 操作系统

您可以使用任何终端连接工具（如WinSCP、PuTTY等工具）将证书上传至Linux系统任一目录下。

示例：

图 16-1 导入证书



# 17 监控和告警

## 17.1 用户平时需要关注实例的哪些性能指标

用户平时需要关注的监控指标有：CPU利用率、内存利用率、磁盘空间利用率。

更多监控指标信息，请参见[文档数据库服务支持的监控指标](#)。

可以根据实际应用场景配置告警提示，当收到告警，可采取相应措施消除告警。

**配置示例：**

- 如果在某段时间内（如5min），CPU的利用率出现多次（如3次）不低于某特定值（如90%）的情况，则发出相应云监控告警。
- 如果在某段时间内（如5min），内存的利用率出现多次（如4次）不低于某特定值（如90%）的情况，则发出相应云监控告警。
- 如果在某段时间内（如5min），磁盘的使用率出现多次（如5次）不低于某特定值（如85%）的情况，则发出相应云监控告警。

 **说明**

云监控告警详细配置方法，详见《云监控用户指南》中“告警规则管理”。

如果CPU、内存使用率长期较高，且无法扩容磁盘容量，可以选择变更CPU和内存规格。

**采取措施：**

收到与磁盘空间利用率有关的告警：

- 调查磁盘空间消耗，查看是否可以从实例中删除数据或是将数据存档到其他系统以释放空间。
- 通过磁盘空间扩容增大磁盘空间，请参见[扩容存储](#)。

## 17.2 配置 DDS 告警规则时，找不到常规监控项怎么办？

**问题现象**

配置DDS告警规则时，发现没有一些常规的监控指标，例如：CPU利用率、内存利用率、磁盘空间利用率等指标。

## 问题分析

在云监控服务界面配置告警规则时，设置的监控维度和该监控指标支持的维度不一致，导致查看不到对应的监控指标。

例如：CPU使用率的监控对象为文档数据库集群实例下的dds mongos节点、文档数据库实例下的主节点和文档数据库实例下的备节点，其维度是节点级别的，但是在云监控界面配置告警规则时选择的维度是文档数据库实例，二者的监控维度不一致，所以就查看不到CPU使用率的监控指标。

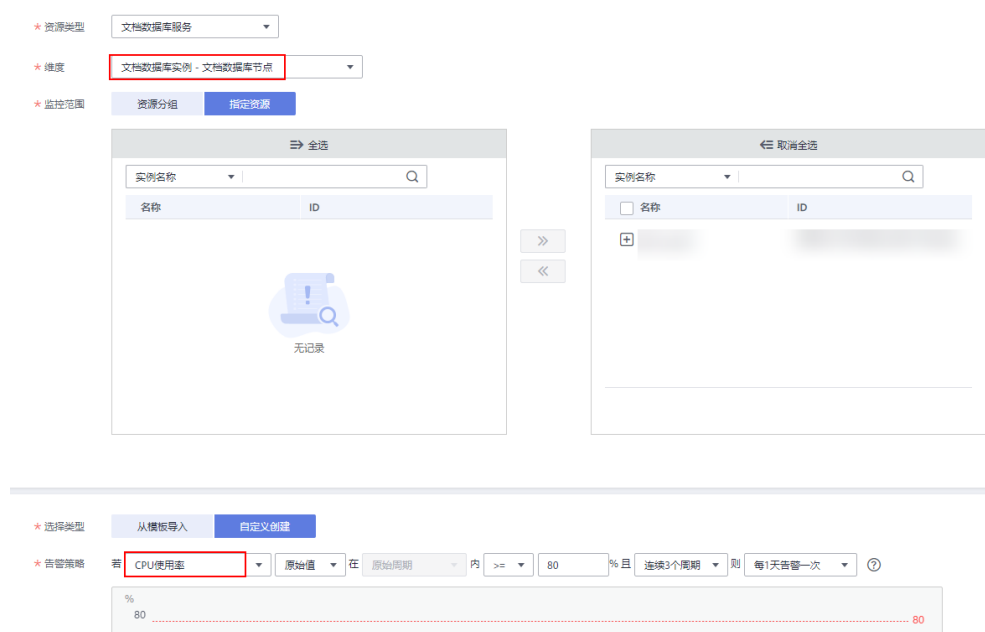
## 解决方案

DDS支持实例级别和节点级别的指标监控，需要根据监控指标支持的维度在告警规则页面设置对应的维度。不同的指标他的维度是不同，详情请参见[文档数据库支持的监控指标](#)。

例如：CPU使用率支持的监控对象类型为：文档数据库集群实例下的dds mongos节点、文档数据库实例下的主节点和文档数据库实例下的备节点。

那么在云监控服务界面配置告警规则时，维度需要选择“文档数据库实例-文档数据库节点”，就可以在“告警策略”中查看到CPU使用率了。

图 17-1 配置监控维度



## 17.3 DDS 的监控指标采集粒度是多少

DDS默认支持的监控周期为1分钟，为了提高监控指标的瞬时精确值，您也可以将监控周期设置为5秒。

具体操作请参见[设置秒级监控](#)。