

云数据库 MongoDB 版

集群版快速入门

集群版快速入门

开始使用MongoDB

快速入门流程图

若您初次使用阿里云MongoDB，请先了解阿里云MongoDB使用须知以及阿里云MongoDB管理控制台。

通常，从新购实例到可以开始使用实例，您需要完成如下操作：



文档目的

快速入门旨在介绍如何创建MongoDB实例、进行基本设置以及连接实例数据库，使用户能够了解从购买MongoDB实例到开始使用实例的流程。

目标读者

首次购买MongoDB实例的用户。

新建实例后需要对其进行基本设置的用户。

想要了解如何连接MongoDB实例的用户。

使用须知

用户可以很轻松地将自建数据库MongoDB迁移至云数据库MongoDB版，但是MongoDB版本本身也有一些使用约束需要注意。

操作	约束
搭建数据库复制	- 系统自动搭建了三副本的副本集。 - 其中对用户提供了两个节点（Primary和Secondary），另外一个备份节点隐藏对用户不可见。 - 用户暂时无法自行搭建Secondary节点。
重启数据库	必须通过控制台进行重启实例的操作。
数据迁入	- 推荐您使用DTS导入数据。 - 您也可以使用自带命令行工具导入数据
存储引擎：WiredTiger	- 分片集群版当前支持MongoDB 3.2和MongoDB 3.4两个版本。 - 不支持修改存储引擎。
搭建集群	- 用户可以在售卖页面创建一个集群实例，选择mongos的配置和个数，shard节点的配置和个数（每个shard节点由三节点副本集组成）和configserver。 - 集群的维护工作由阿里云MongoDB运维团队完成。

关于MongoDB控制台

MongoDB管理控制台是用于管理MongoDB实例的Web应用程序，您可以在MongoDB管理控制台上进行实例创建、实例管理、IP白名单设置、密码设置、网络设置等操作。

MongoDB管理控制台是阿里云管理控制台的一部分，关于控制台的通用设置和基本操作请参见使用阿里云管理控制台。

前提条件

使用阿里云账号登录MongoDB管理控制台。若没有阿里云账号，请单击注册。

控制台简介

控制台首页

对于MongoDB所有集群版实例而言，控制台首页的界面信息都是相同的。

登录MongoDB管理控制台，进入**实例列表**页面，如下图所示（仅为示例，请以实际界面为准）。



参考说明：

序号	名称	说明
1	集群版实例列表	MongoDB控制台的首页，显示同一账户中某个地区下的的所有实例信息。
2	地域	单击某一个地域名称，该地域下的所有实例就会显示在实例列表中。
3	刷新	刷新实例信息页面
4	新建实例	创建实例入口
5	实例ID	单击实例的ID可以进入该实例详情页面。
6	查看	单击可以进入该实例详情页面。
7	释放	单击可释放该实例。

MongoDB实例控制台

登录MongoDB管理控制台，在实例列表中单击实例ID或者实例旁边的**查看**，即可进入MongoDB实例的管理详情页面，详情如下表所示：

控制台页面名称	区块名称	描述	常用操作链接
界面上方操作区		可进行备份实例、重启实例操作	-备份实例 -重启实例
基本信息	基本信息	可以查看实例的基本信息，如实例ID、地域、网络类型、规格、磁盘空间。	
	账号管理	查看实例账号，可进行	重置密码

		重置密码操作	
	配置信息	可查看数据库版本，实例创建时间和实例的网络类型。	
	实用信息	显示连接MongoDB云数据库的常用命令语法。	
	当前资源状况	显示实例的Mongos节点和Shard节点个数	- 添加Mongos - 添加Shard
	Mongos或者Shard	- 单击Mongos可以变配、登录、重启以及查看实例的Mongos节点。 - 单击Shard可以变配、登录、重启以及查看实例的Shard节点。 - 重启节点时，可能会导致数据库的读写操作失败，所以不建议用户在重启数据库，或者是重启节点的时候进行数据库的增删改查等操作。	- 变更配置 - 登录数据库 - 重启节点 - 查看Mongos节点或者Shard节点的运行监控信息。
备份与恢复	备份列表	可查看选定时间的数据备份列表并进行下载以及按时间点新建实例	- 下载备份数据 - 按时间点新建实例
	备份设置	按照选定的时间点进行周期性的自动备份。	设置备份周期
监控信息	资源监控	根据选定的数据指标和查询时间查看Mongos节点和Shard节点的监控信息	
安全控制	安全控制	可进行IP白名单设置	IP白名单设置

创建实例

您可以通过阿里云MongoDB管理控制台创建MongoDB实例。关于实例计费说明，请参见收费项目及价格说明。本文将介绍在MongoDB管理控制台上创建实例的步骤。

前提条件

已注册阿里云账号。

目前集群版只支持创建按量付费的实例，请确保您的账户余额大于等于100元。

操作步骤

登录MongoDB管理控制台。

在**集群实例列表**页面，单击**新建实例**，进入创建实例页面。

在（**集群版本**）**按量付费**下选择实例配置。关于计费方式的选择，请参见收费项目及价格说明。

选择实例配置，参数说明如下表所示。

参数配置区	参数项	说明	注意事项
基本配置	- 地域 - 可用区 - 数据库版本	选择实例所在的地域、可用区和数据库版本。目前MongoDB支持的数据库版本是MongoDB3.2。	不同地域内的产品内网不互通，且购买后不能更换地域，请谨慎选择。
网络类型	- 经典网络 - 专有网络	MongoDB支持经典网络和专有网络（Virtual Private Cloud，简称VPC）。专有网络需要事先创建，或者您也可以在创建实例后再更改网络类型，具体请参见网络设置。 - 经典网络：经典网络中的云服务在网络上不进行隔离，只能依靠云服务自身的安全组或白名单策略来阻挡非法访问。 - 专有网络（Virtual Private Cloud，简称VPC）：专有网络帮助您在阿里云上构建出一个隔离的网络环境，您可以自定义专有网络里面的路由表、IP地址范围和网关。此外，您还可以通过专线或者VPN的	

		方式将自建机房与阿里云专有网络内的云资源组合成一个虚拟机房，实现应用平滑上云。	
配置Mongos	- 单个Mongos节点规格 - Mongos节点数量	Mongos节点的规格以及节点数，不同规格对应不同的最大连接数和IOPS（即读和写分别可以达到的最大值，混合读写最高可以达到指标的2倍）不同。	- 规格（例如4核8G）是指单个Mongos节点的规格。 - 每个实例最多可开通32个Mongos
配置Shard	- 规格 - 存储空间 - 数量	Shard节点的规格、存储空间大小以及节点数。不同规格对应不同的最大连接数和IOPS（即读和写分别可以达到的最大值，混合读写最高可以达到指标的2倍）不同。	- 规格（例如4核8G）和存储空间（例如500G）是指单个Shard节点的规格和存储空间。 - 每个实例最多可开通32个Shard
配置ConfigServers	- 规格 - 存储空间	配置ConfigServers的规格和存储空间。	ConfigServer为三副本高可用架构规格和存储空间不可变。
密码设置	- 设置密码 - 账号密码 - 确认密码	设置实例的密码。	购买实例时必须先设置密码，如果要跟改密码请参见重置密码

单击**立即购买**，进入订单确认页面。

提示：若您需要购买多个不同配置的实例，可以单击加入清单将要购买的实例逐个加入购买清单中，然后在购买清单中单击批量购买，如下图所示。

云数据库MongoDB版
确认订单
商品信息
地域: 华北 1
可用区: 华北 1 可用区 B
网络类型: 经典网络
数据库类型: MongoDB
数据库版本: 兼容3.2的阿里云企业版
Mongos: 4核8G
数量: 2个
Shard: 4核8G
数量: 2个
存储空间: 500GB
configserver: 1核2G
存储空间: 20GB

阅读云数据库MongoDB版服务条款后，根据后续提示完成支付流程。

- 按量付费实例：单击**去开通**，实例创建完成。

设置白名单

为了数据库的安全稳定，在创建实例后MongoDB会自动将IP为127.0.0.1的IP地址加入白名单中，所以创建实例后应该将需要访问数据库的IP地址或者IP段加入白名单。MongoDB允许您添加1000个IP地址。

注意：在启用目标实例前，请先修改白名单。

添加IP白名单

登录MongoDB管理控制台，找到目标实例。

单击实例ID或者[查看](#)进入**基本信息**页面。

在**基本信息**页面单击**数据安全性**打开**数据安全性**页面，如下图所示：



可选择**手动修改**或者**加载ECS内网IP添加**两种方式设置IP白名单。

单击**手动修改**，手动输入IP/IP段，单击**确定**。如下图所示：



单击**加载ECS内网IP添加**，系统将显示同账号下的ECS内网IP，您可以选择ECS内网中的IP并添加到白名单中，单击**确定**。如下图所示：



注意：

IP地址请以逗号隔开，不可重复，最多1000个。支持格式如：
：0.0.0.0/0，10.23.12.24（IP），10.23.12.24/24（CIDR模式，无类域间路由，/24表示了地址中前缀的长度，范围[1,32]）。

0.0.0.0/0和空代表不设IP访问的限制，数据库将会有高安全风险。建议仅将您的WEB服务器外网IP/IP段设为可访问权限。

后续操作

正确使用白名单可以让MongoDB得到高级别的访问安全保护，建议用户定期维护白名单。

后续操作中，您可以单击**手动修改**或者**加载ECS内网IP**修改白名单。

设置密码

您可以通过MongoDB控制台重新设置实例的密码。

操作步骤

登录MongoDB管理控制台。

单击实例ID或者[查看](#)进入**基本信息**页面。

在**账号管理**栏中，单击**重置密码**，如下图所示。



基本信息	
实例ID： dds-bp192cd98f67c9f4	名称： dds-bp192cd98f67...
地域： 华东 1 可用区 B	存储引擎： WiredTiger

帐号管理	
帐号名： root 权限为admin数据库下的root权限	

在**重置密码**窗口，输入新密码，单击**确认**，如下图所示。



重置密码

账号： root
权限为admin数据库下的root权限

* 新密码：
密码，由字母、数字或下划线组成，长度为6~32位

* 确认新密码：
密码，由字母、数字或下划线组成，长度为6~32位

确定 取消

分片集群连接说明

MongoDB分片集群（Sharded Cluster）通过将数据分散存储到多个分片（Shard）上，来实现高可扩展性。实现分片集群时，MongoDB引入Config Server来存储集群的元数据，引入mongos作为应用访问的入口，mongos从Config Server读取路由信息，并将请求路由到后端对应的Shard上。

云数据库MongoDB版为每个mongos都单独提供了连接地址，同时支持用Connection String的方式连接集群，实现负载均衡及高可用。

- 用户访问mongos跟访问单个mongod类似。
- 所有mongos是对等关系，用户访问分片集群可通过任意一个或多个mongos。
- mongos本身是无状态的，可任意增删。当前一个集群最大支持扩展是32个mongos，最少需保有2个mongos以实现高可用。
- 访问分片集群时，最好将应用负载均匀分散到多个mongos上。

获取连接地址

登录MongoDB管理控制台。

单击实例ID或者[查看](#)进入实例**基本信息**页面，单击右侧**数据库连接**，打开界面如下图所示。

内网连接

网络类型：经典网络

Mongos地址：
MongosID：s-bp1465a513aad2b4 规格：1核2 GB 连接地址：s-bp1465a513aad2b4.mongodb.rds.aliyuncs.com:3717
MongosID：s-bp16e41eab331e14 规格：1核2 GB 连接地址：s-bp16e41eab331e14.mongodb.rds.aliyuncs.com:3717

使用Mongo Shell连接实例：
`mongo --host s-bp1465a513aad2b4.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase admin -u root -p`
`mongo --host s-bp16e41eab331e14.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase admin -u root -p`

客户端使用Connection String URI连接实例（****部分替换为root密码）：
`mongodb://root:****@s-bp1465a513aad2b4.mongodb.rds.aliyuncs.com:3717,s-bp16e41eab331e14.mongodb.rds.aliyuncs.com:3717/admin`

如上图所示，您可以获取各种网络类型下的每个mongos的连接地址，以及为用户直接生成的Connection String URI。

连接说明

云数据库MongoDB分片集群实例每个mongos都提供了连接地址，日常测试或查询时可直接连接某个mongos的地址实现读写操作，但线上应用需通过Connection String的方式连接集群，以实现高可用。

```
客户端使用Connection String URI连接实例（****部分替换为root密码）：  
mongodb://root:****@s-bp1465a513aad2b4.mongodb.rds.aliyuncs.com:3717,s-bp16e41eab331e14.mongodb.rds.aliyuncs.com:3717/admin
```

从控制台获取的connection string将包含以下内容：

```
mongodb://[username:password@]host1[:port1],[host2[:port2],...[,hostN[:portN]]]/[database][?options]
```

- mongodb://：前缀，代表这是一个Connection String。
- username:password@：如果启用了鉴权，需要指定用户密码。
- hostX:portX：多个mongos的地址列表。
- /database：鉴权时，用户帐号所属的数据库。
- ?options：指定额外的连接选项。

所有MongoDB官方Driver都支持通过Connection String来连接，通过上述方式连接分片集群时，客户端会自动将请求分散到多个mongos上，以实现负载均衡。同时，当URI里mongos数量在2个及以上时，或者有mongos故障时，客户端能自动进行failover，将请求都分散到状态正常的mongos上。您可以参阅程序代码连接示例来实现连接操作。

连接实例

程序代码连接

相关链接

[MongoDB Drivers](#)

[Connection String URI Format](#)

说明：本文连接demo仅适用于阿里云提供的MongoDB内网连接地址。

Node.js 连接示例

相关链接

[MongoDB Node.js Driver](#)

项目初始化

```
mkdir node-mongodb-demo
cd node-mongodb-demo
npm init
```

安装驱动包以及工具包

```
npm install mongodb node-uuid sprintf-js --save
```

获取云数据库 MongoDB 连接信息

请查看 MongoDB 连接信息七要素获取方法。

Node.js Demo Code

```
'use strict';

var uuid = require('node-uuid');
var sprintf = require("sprintf-js").sprintf;
var MongoClient = require('mongodb').MongoClient;

var host1 = "demotest-1.mongodb.tbc3.newtest.rdstest.aliyun-inc.com";
var port1 = 27017;

var host2 = "demotest-2.mongodb.tbc3.newtest.rdstest.aliyun-inc.com";
var port2 = 27017;

var username = "demouser";
var password = "123456";

var replSetName = "mgset-1441984991";

var demoDb = "test";
var demoColl = "testColl";

// 官方建议使用的方案
var url = sprintf("mongodb://%s:%d,%s:%d/%s?replicaSet=%s", host1, port1, host2, port2, demoDb, replSetName);
console.info("url:", url);

// 获取mongoClient
mongoClient.connect(url, function(err, db) {
  if(err) {
    console.error("connect err:", err);
    return 1;
  }
});

// 授权. 这里的username基于admin数据库授权
var adminDb = db.admin();
```

```
adminDb.authenticate(username, password, function(err, result) {
  if(err) {
    console.error("authenticate err:", err);
    return 1;
  }

  // 取得Collecton句柄
  var collection = db.collection(demoColl);

  var demoName = "NODE:" + uuid.v1();
  var doc = {"DEMO": demoName, "MESG": "Hello AliCoudDB For MongoDB"};
  console.info("ready insert document: ", doc);

  // 插入数据
  collection.insertOne(doc, function(err, data) {
    if(err) {
      console.error("insert err:", err);
      return 1;
    }
    console.info("insert result:", data["result"]);

    // 读取数据
    var filter = {"DEMO": demoName};
    collection.find(filter).toArray(function(err, items) {
      if(err) {
        console.error("find err:", err);
        return 1;
      }
      console.info("find document: ", items);

      //关闭Client，释放资源
      db.close();
    });
  });
});
```

PHP 连接示例

相关链接

Mongodb php driver

安装驱动包以及工具包

```
$ pecl install mongodb
$ echo "extension=mongodb.so" >> `php --ini | grep "Loaded Configuration" | sed -e "s|.*:|`
$ composer require "mongodb/mongodb=^1.0.0"
```

获取云数据库MongoDB的连接信息

请查看 [MongoDB 连接信息七要素获取方法](#)。

PHP Demo Code

```
<?php
require 'vendor/autoload.php'; // include Composer goodies

# 实例信息
$demo_seed1 = 'demotest-1.mongodb.test.aliyun-inc.com:3717';
$demo_seed2 = 'demotest-2.mongodb.test.aliyun-inc.com:3717';
$demo_replname = "mgset-1441984463";
$demo_user = 'root';
$demo_password = '123456';
$demo_db = 'admin';

# 根据实例信息构造mongodb connection string
# mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]
$demo_uri = 'mongodb://' . $demo_user . ':' . $demo_password . '@' .
$demo_seed1 . ' . ' . $demo_seed2 . '/' . $demo_db . '?replicaSet=' . $demo_replname;

$client = new MongoClient($demo_uri);
$collection = $client->testDb->testColl;

$result = $collection->insertOne( [ 'name' => 'ApsaraDB for Mongodb', 'desc' => 'Hello, Mongodb' ] );

echo "Inserted with Object ID '{$result->getInsertedId()}'", "\n";

$result = $collection->find( [ 'name' => 'ApsaraDB for Mongodb' ] );

foreach ($result as $entry)
{
    echo $entry->_id, ': ', $entry->name, "\n";
}
?>
```

Java 连接示例

相关链接

官方Quick Start。

Jar包下载。

获取云数据库 MongoDB 连接信息

请查看 [MongoDB 连接信息七要素获取方法](#)。

Java Demo Code

Maven配置

```
<dependencies>
<dependency>
<groupId>org.mongodb</groupId>
<artifactId>mongo-java-driver</artifactId>
<version>3.0.4</version>
</dependency>
</dependencies>
```

Java Code

```
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;

import org.bson.BsonDocument;
import org.bson.BsonString;
import org.bson.Document;

import com.mongodb.MongoClient;
import com.mongodb.MongoClientOptions;
import com.mongodb.MongoClientURI;
import com.mongodb.MongoCredential;
import com.mongodb.ServerAddress;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;

public class Main {

    public static ServerAddress seed1 = new ServerAddress("demotest-1.mongodb.tbc3.newtest.rdstest.aliyun-inc.com",
27017);
    public static ServerAddress seed2 = new ServerAddress("demotest-2.mongodb.tbc3.newtest.rdstest.aliyun-inc.com",
27017);

    public static String username = "demouser";
    public static String password = "123456";
    public static String ReplSetName = "mgset-1441984463";

    public static String DEFAULT_DB = "admin";

    public static String DEMO_DB = "test";
    public static String DEMO_COLL = "testColl";

    public static MongoClient createMongoDBClient() {
        // 构建Seed列表
        List<ServerAddress> seedList = new ArrayList<ServerAddress>();
        seedList.add(seed1);
        seedList.add(seed2);

        // 构建鉴权信息
        List<MongoCredential> credentials = new ArrayList<MongoCredential>();
        credentials.add(MongoCredential.createScramSha1Credential(username,
DEFAULT_DB, password.toCharArray()));
```

```
// 构建操作选项，requiredReplicaSetName属性外的选项根据自己的实际需求配置，默认参数满足大多数场景
MongoClientOptions options = MongoClientOptions.builder()
    .requiredReplicaSetName(ReplSetName).socketTimeout(2000)
    .connectionsPerHost(1).build();
return new MongoClient(seedList, credentials, options);
}

public static MongoClient createMongoDBClientWithURI() {
    //另一种通过URI初始化
    //mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]
    MongoClientURI connectionString = new MongoClientURI("mongodb://" + username + ":" + password + "@" +
        seed1 + "," + seed2 + "/" +
        DEFAULT_DB +
        "?replicaSet=" + ReplSetName);
    return new MongoClient(connectionString);
}

public static void main(String args[]) {
    MongoClient client = createMongoDBClient();
    //or
    //MongoClient client = createMongoDBClientWithURI();
    try {
        // 取得Collecton句柄
        MongoDB database = client.getDatabase(DEMO_DB);
        MongoCollection<Document> collection = database.getCollection(DEMO_COLL);

        // 插入数据
        Document doc = new Document();
        String demoname = "JAVA:" + UUID.randomUUID();
        doc.append("DEMO", demoname);
        doc.append("MESG", "Hello AliCoudDB For MongoDB");
        collection.insertOne(doc);
        System.out.println("insert document: " + doc);
        // 读取数据
        BsonDocument filter = new BsonDocument();
        filter.append("DEMO", new BsonString(demoname));
        MongoCursor<Document> cursor = collection.find(filter).iterator();
        while (cursor.hasNext()) {
            System.out.println("find document: " + cursor.next());
        }
    } finally {
        //关闭Client，释放资源
        client.close();
    }
    return ;
}
}
```

Python 连接示例

相关链接

[pymongo下载地址](#)

官方文档

安装pymongo

```
pip install pymongo
```

获取云数据库MongoDB连接信息

请查看 MongoDB 连接信息七要素获取方法。

Python Demo Code

```
import uuid
from pymongo import MongoClient

#两地址
CONN_ADDR1 = 'demotest-1.mongodb.tbc3.newtest.rdstest.aliyun-inc.com:27017'
CONN_ADDR2 = 'demotest-2.mongodb.tbc3.newtest.rdstest.aliyun-inc.com:27017'
REPLICAT_SET = 'mgset-1441984463'

username = 'demouser'
password = '123456'

#获取mongoclient
client = MongoClient([CONN_ADDR1, CONN_ADDR2], replicaSet=REPLICAT_SET)

#授权. 这里的user基于admin数据库授权
client.admin.authenticate(username, password)

#使用test数据库的collection:testColl做例子, 插入doc, 然后根据DEMO名查找
demo_name = 'python-' + str(uuid.uuid1())
print 'demo_name:', demo_name
doc = dict(DEMO=demo_name, MESSG="Hello ApsaraDB For MongoDB")
doc_id = client.test.testColl.insert(doc)
print 'doc_id:', doc_id

for d in client.test.testColl.find(dict(DEMO=demo_name)):
    print 'find documents:', d
```

更多程序代码连接示例请参见MongoDB官网。

DMS连接数据库

DMS 是一款访问管理云端数据的Web服务，支持Redis、MySQL、SQL Server、PostgreSQL和

MongoDB等数据源。DMS提供了数据管理、对象管理、数据流转和实例管理四部分功能。您可以通过控制台上的图形化工具（DMS）进行连接。

操作步骤

登录MongoDB管理控制台，找到目标实例。

单击实例ID或者[查看](#)进入**基本信息**页面。

找到如下界面，单击**登录**可以直接跳转到**DMS 数据管理登录**页面。

Mongos	Shard			
ID	描述	规格	域名信息	操作
s-bp1531442cc40764	s-bp1531442cc407...	1核2 GB	s-bp1531442cc40764.mongodb.rds.aliyuncs.com:3717	变更配置 性能监控 登录 重启
s-bp1fa6e5fd074874	s-bp1fa6e5fd0748...	 1核2 GB	s-bp1fa6e5fd074874.mongodb.rds.aliyuncs.com:3717	变更配置 性能监控 登录 重启

登录DMS时先跳转至DMS登录页，填写如下相应的信息，方可成功登录MongoDB，如下图所示。

 **RDS数据库登录** [独立单元](#)

☐ 记住密码

[关于DMS \(Data Management Service \)](#)
Copyright © DMS All Rights Reserved (Alibaba 数据管理产品)

用户名：root

密码：开通实例时指定的密码或控制台重置密码

验证数据库：admin

登陆成功后您就可以创建新的数据库以及做其他与MongoDB数据库相关的操作。更多DMS中的MongoDB操作请参考：[DMS for MongoDB](#)。

Mongo shell连接

注意：请使用mongo shell 3.0及以上的版本，才能连接云数据库MongoDB 版，否则无法鉴权成功。

操作步骤

登录MongoDB管理控制台。

单击目标实例ID或者查看，进入基本信息页，如下图所示，获取MongoDB连接域名地址和端口号。

ID	描述	规格	域名信息	操作
s-m5e2be1275374454	s-m5e2be12753744...	4核 8G	s-m5e2be1275374454.mongodb.rds.aliyuncs.com:3717	变配 性能监控 登录 重启
s-m5e2ea2490ca1824	s-m5e2ea2490ca18...	4核 8G	s-m5e2ea2490ca1824.mongodb.rds.aliyuncs.com:3717	变配 性能监控 登录 重启

- 标号为1、2的是两个节点的连接地址和端口号，连接地址和端口号之间用冒号隔开。
- 单击Mongos节点中的任何一个节点登录都可以。

在ECS上使用mongo命令进行连接，命令样例如下：

```
mongo --host dds-xxxx.mongodb.rds.aliyuncs.com:3717 -u root -p 123456 --authenticationDatabase admin
```

mongo shell常见错误

- 连接问题
- 连接数问题
- 负载高问题

公网连接

ECS Linux 篇

目前云数据库MongoDB是需要通过ECS的内网进行连接访问，如果您本地需要通过公网访问云数据库MongoDB，可以在ECS Linux云服务器中安装rinetd进行转发实现。

操作步骤

在云服务器ECS Linux上安装rinetd。

```
wget http://www.boutell.com/rinetd/http/rinetd.tar.gz&&tar -xvf rinetc.tar.gz&&cd rinetc
sed -i 's/65536/65535/g' rinetc.c (修改端口范围)
mkdir /usr/man&&make&&make install
```

打开配置文件。

```
vi /etc/rinetc.conf
```

在配置文件中输入如下内容：

```
0.0.0.0 3717 MongoDB的链接地址 3717
logfile /var/log/rinetc.log
```

执行如下命令启动rinetd。

```
rinetc
```

注意：通过echo rinetc >>/etc/rc.local可以设置为自启动，可以使用kill rinetc结束该进程。

验证测试。

在本地通过mongo shell连接ECS Linux服务器后进行登录验证，比如安装了 rinetc的服务器的IP是1.1.1.1：

```
mongo --host 1.1.1.1:3717 -u root -p 密码 --authenticationDatabase admin
```

通过上述步骤即可实现：您本地的PC或服务器通过公网连接ECS Linux 3717端口，对云数据库MongoDB进行访问。

注意：您可以通过该方案进行测试使用，因rinetd为开源软件，如在使用过程中存在疑问，您可以参看其官方文档或与rinetd官方进行联系确认。

ECS Windows 篇

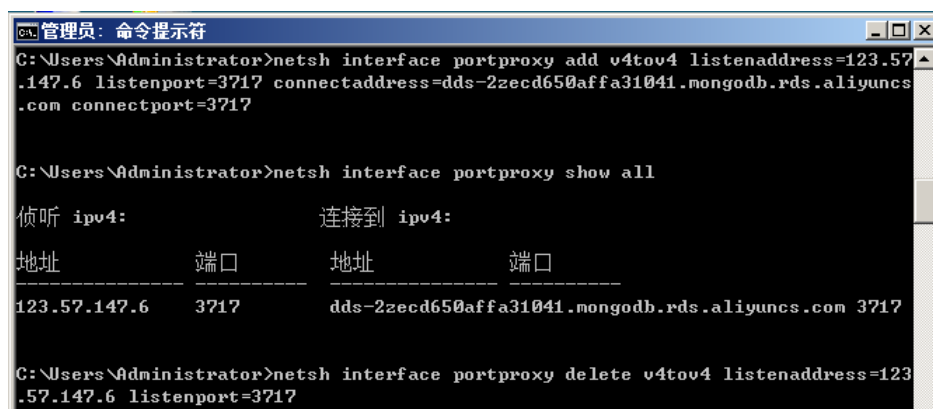
目前云数据库MongoDB是需要通过 ECS 的内网进行连接访问，如果您本地需要通过公网访问云数据库MongoDB，可以在 ECS Windows 云服务器中通过 netsh 进行端口映射实现。

操作步骤

登录 ECS Windows 服务器，在 CMD 中执行以下命令：

```
netsh interface portproxy add v4tov4 listenaddress=ECS服务器的公网IP地址 listenport=3717  
connectaddress=云数据库MongoDB的连接地址 connectport=3717
```

如下图：



注意：

netsh interface portproxy delete v4tov4 listenaddress=ECS公网服务器的公网IP地址

listenport=3717 //可以删除不需要的映射。

netsh interface portproxy show all //可以查看当前服务器中存在的映射。

设置完成后进行验证测试。

在本地 MongoDB shell 连接 ECS Windows 服务器后进行数据写入和查询验证，ECS Windows 服务器的 IP 是1.1.1.1，即 telnet 1.1.1.1 3717。

```
[root@iZ... init.d]# mongo --host 123.57.147.6:3717 -u root -p --authenticationDatabase admin
MongoDB shell version: 3.0.8
connecting to: 123.57.147.6:3717/test
mgset-682723:PRIMARY>
```

通过上述步骤即可实现：您本地的 PC 或服务器通过公网连接 ECS Windows 3717端口，对云数据库 MongoDB 进行访问。

注意：因 portproxy 由微软官方提供，未开源使用，您如果配置使用过程中遇到疑问，可参看 netsh 的 portproxy 使用说明或向微软官方咨询确认。或者您也可以考虑通过其他的方案实现，比如通过 portmap 配置代理映射。

数据导入

使用自带命令行工具迁移数据

MongoDB自带了mongodump和mongoexport的命令行进行数据导入导出操作。

注意：请使用MongoDB3.0以上版本的mongodump和mongoexport工具。

迁移步骤

备份自建数据库数据。

连接自建数据库，选用相应权限的账号，执行 mongodump 将自建数据库内容导出。例如，执行以下语句导出所有数据库，默认生成文件名为dump的备份文件。

```
mongodump --host xxx:27017 --authenticationDatabase admin -u xxx -p xxx
```

详细参考请见[mongodump 命令官方文档](#)。

将备份出的文件导入至云数据库MongoDB版。

根据上一步导出生成的备份文件，执行 `mongorestore` 命令将数据全部导入至云数据库MongoDB。例如，执行以下语句将数据库全部导入。

```
mongorestore --host dds-xxx:3717 --authenticationDatabase admin -u root -p xxx dump
```

数据迁出

迁至本地MongoDB数据库

云数据库MongoDB版提供常规备份任务设置及临时备份的功能。您可以定期或任意时间点进行备份任务，通过下载备份数据将数据迁出。

登录MongoDB管理控制台，找到目标实例。

单击实例ID或者[查看](#)进入**基本信息**页面。

在**基本信息**页面单击**备份恢复**打开**备份与恢复**页面

切换到**备份列表**页签下，选择目标备份集，单击**下载**。

将备份文件自行下载后，在本地搭建MongoDB环境。

执行 `mongorestore` 命令便可以完成数据迁出操作。例如，执行以下命令进行数据导入：

```
cat xx.ar (控制台下载的文件) | mongorestore -h xxx --port xxx -u[root user] -p[root password] --drop --gzip --archive -vvvv --stopOnError
```