

ALIBABA CLOUD

阿里云

云数据库 MongoDB 版
快速入门

文档版本：20220705

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.入门概述	06
2.使用限制	08
3.单节点快速入门	16
3.1. 使用限制	16
3.2. 创建单节点实例	17
3.3. （可选）重置密码	22
3.4. 设置白名单	23
3.5. （可选）申请公网连接地址	24
3.6. 连接实例	25
3.6.1. 通过DMS连接MongoDB单节点实例	25
3.6.2. 通过Mongo Shell连接MongoDB单节点实例	27
3.6.3. 通过程序代码连接MongoDB单节点实例	30
4.副本集快速入门	40
4.1. 使用限制	40
4.2. 创建副本集实例	42
4.3. （可选）重置密码	49
4.4. 设置白名单	49
4.5. （可选）申请公网连接地址	50
4.6. 连接实例	52
4.6.1. 通过DMS连接MongoDB副本集实例	52
4.6.2. 通过Mongo Shell连接云数据库MongoDB副本集实例	54
4.6.3. 通过程序代码连接MongoDB副本集实例	61
5.分片集群快速入门	70
5.1. 使用限制	70
5.2. 创建分片集群实例	72
5.3. （可选）重置密码	80

5.4. 设置白名单	80
5.5. (可选) 申请公网连接地址	81
5.6. 连接实例	83
5.6.1. 通过DMS连接MongoDB分片集群实例	83
5.6.2. 通过Mongo Shell连接MongoDB分片集群实例	86
5.6.3. 通过程序代码连接MongoDB分片集群实例	88
5.6.4. 通过程序代码连接DynamoDB协议兼容版实例	96
6.Serverless快速入门	102
6.1. 使用限制	102
6.2. 创建Serverless实例	103
6.3. (可选) 重置密码	106
6.4. 设置白名单	106
6.5. (可选) 申请公网连接地址	107
6.6. 连接实例	108
6.6.1. 通过Mongo Shell连接云数据库MongoDB Serverless实例	108
6.6.2. 通过程序代码连接Serverless实例	110

1. 入门概述

本文介绍云数据库MongoDB实例从创建到连接使用的整体流程，指引您快速掌握创建云数据库MongoDB实例、配置实例信息及连接实例的方法。

部署架构

云数据库MongoDB具有可灵活部署的架构，能够满足不同的业务场景。具体如下：

- 单节点架构

适用于开发、测试、学习培训及其他非企业核心数据存储的场景。您可以根据各类场景的差异适配对应的规格配置，为企业降低更多的成本支出。详情请参见[单节点架构](#)。

- 副本集架构

适用于读多写少或有临时活动的突发业务场景。副本集实例提供一个可供读写访问的Primary节点（主节点），一个、三个或五个提供高可用的Secondary节点（从节点），一个隐藏的Hidden节点（隐藏节点），0~5个可选的ReadOnly节点（只读节点）。您可以根据业务需要，按需增删Secondary节点和ReadOnly节点，更好地实现读取性能扩展节点。详情请参见[副本集架构](#)。

- 分片集群架构

适用于高并发读写的场景。分片集群实例是基于多个副本集（每个副本集使用三节点主从模式+0~5个只读节点）组成，提供Mongos、Shard、ConfigServer三个组件，您可以自由地选择Mongos和Shard节点的个数和配置，无限扩展性能及存储空间，组建不同能力的分片集群实例。详情请参见[分片集群架构](#)。

- Serverless架构

适用于中小型客户群体。Serverless形态提供代理、网络资源、命名空间、存储空间的垂直资源隔离能力，提供计算资源按需计费能力，具有资源用量低、简单易用、弹性灵活、价格低廉等优点，完美解决了MongoDB使用门槛高的问题，帮助中小客户轻松上云。详情请参见[Serverless架构](#)。

操作流程

从新购实例到可以使用实例，您需要完成以下操作：



1. 创建实例。不同类型实例的创建方法如下：

- [创建单节点实例](#)
- [创建副本集实例](#)
- [创建分片集群实例](#)
- [创建Serverless实例](#)

2. （可选）如果创建实例过程中未设置密码或忘记密码，您可以设置或重置密码设置密码，不同类型实例密码的设置方法请参见：

- [重置密码（单节点实例）](#)
- [重置密码（副本集实例）](#)
- [重置密码（分片集群实例）](#)
- [重置密码（Serverless实例）](#)

3. 设置白名单。将需要连接实例的外部设备添加至实例的白名单，以允许外部设备能够访问该实例。不同类型实例白名单的设置方法请参见：
 - [设置白名单（单节点实例）](#)
 - [设置白名单（副本集实例）](#)
 - [设置白名单（分片集群实例）](#)
 - [设置白名单（Serverless实例）](#)
4. （可选）如果您需要使用公网地址连接实例，请您先申请公网连接地址。不同类型实例公网连接地址的申请方法请参见：
 - [申请公网连接地址（单节点实例）](#)
 - [申请公网连接地址（副本集实例）](#)
 - [申请公网连接地址（分片集群实例）](#)
 - [申请公网连接地址（Serverless实例）](#)
5. 连接实例。不同实例类型的连接方法如下：

实例类型	参见文档
单节点	◦ （推荐）通过DMS连接MongoDB单节点实例 ◦ 通过Mongo Shell连接MongoDB单节点实例 ◦ 通过程序代码连接MongoDB单节点实例
副本集	◦ （推荐）通过DMS连接MongoDB副本集实例 ◦ 通过Mongo Shell连接MongoDB副本集实例 ◦ 通过程序代码连接MongoDB副本集实例
分片集群	◦ （推荐）通过DMS连接MongoDB分片集群实例 ◦ 通过Mongo Shell连接MongoDB分片集群实例 ◦ 通过程序代码连接MongoDB分片集群实例
Serverless	◦ 通过Mongo Shell连接Serverless实例 ◦ 通过程序代码连接MongoDB实例

2.使用限制

在使用云数据库MongoDB前，您需要了解不同类型云数据库MongoDB实例本身存在的约束信息，使您初步了解不同类型云数据库MongoDB实例的使用限制，以帮助您选择更适合业务场景的实例。

单节点实例

功能	约束
实例部署	云数据库MongoDB不支持在以下地域的可用区创建单节点实例。 • 华东1（杭州）：杭州可用区J和K。 • 华东2（上海）：上海可用区L和D。 • 华东5（南京-本地地域）：南京-本地地域可用区A。 • 华北1（青岛）：青岛可用区B。 • 华北2（北京）：北京可用区E、L和K。 • 华北3（张家口）：张家口可用区A、B和C。 • 华北5（呼和浩特）：呼和浩特可用区A和B。 • 华北6（乌兰察布）：乌兰察布可用区A、B和C。 • 华南1（深圳）：深圳可用区A、D、C、（C+D+E）和F。 • 华南2（河源）：河源可用区A和B。 • 华南3（广州）：广州可用区A。 • 西南1（成都）：成都可用区A和B。 • 中国（香港）：香港可用区（B+C+D）、B、C和D。 • 澳大利亚（悉尼）：悉尼可用区A和B。 • 马来西亚（吉隆坡）：吉隆坡可用区A和B。 • 印尼（雅加达）：雅加达可用区A、B和C。 • 菲律宾（马尼拉）：马尼拉可用区A。 • 泰国（曼谷）：曼谷可用区A。 • 日本（东京）：东京可用区A和B。 • 韩国（首尔）：首尔可用区A。 • 印度（孟买）：孟买可用区A。 • 阿联酋（迪拜）：迪拜可用区A。 • 美国（硅谷）：硅谷可用区A和B。 • 美国（弗吉尼亚）：弗吉尼亚可用区A和B。 • 英国（伦敦）：伦敦可用区A。 • 德国（法兰克福）：法兰克福可用区A、B和C。
数据库版本	云数据库MongoDB仅3.4和4.0版本支持创建单节点实例。
数据库连接	不支持开启或关闭私网访问加密功能。

功能	约束
数据备份	仅支持快照备份的方式备份数据。  说明 快照备份可以保留某时间点的磁盘数据状态。
数据恢复	- 仅支持按备份点将备份数据回复至新建实例。如何恢复，请参见按备份点将备份数据恢复至新建实例。 - 不支持按时间点恢复备份数据。
服务可用性	不支持服务可用性功能，包括主备切换、角色切换和迁移可用区。
数据安全性	- 仅支持专有网络。如何创建专有网络，请参见创建和管理专有网络。 - 仅支持设置白名单。如何设置，请参见设置白名单。 - 不支持审计日志、SSL和TDE。
日志管理	不支持日志管理功能，包括慢日志、错误日志和运行日志。
CloudDBA（性能诊断与优化）	- 仅支持性能趋势、实时性能和实例会话。 - 不支持空间分析和慢日志。
数据迁移与同步	- 云数据库MongoDB支持使用以下方式迁移单节点实例。 - 自建数据库迁移至云数据库MongoDB 将ECS上单节点或副本集架构的自建MongoDB迁移上云 使用DTS迁移单节点架构的自建MongoDB数据库上云 使用MongoDB工具将自建数据库迁移至单节点实例 - 第三方云数据库迁移至云数据库MongoDB 使用DTS将华为云文档数据库迁移至阿里云 - 云数据库MongoDB实例间迁移 从MongoDB单节点实例迁移至副本集或分片集群实例 跨阿里云账号迁移MongoDB实例 迁移MongoDB实例至其他地域 - 单节点实例不支持增量数据迁移与同步。

副本集实例

功能	约束
----	----

功能	约束
实例部署	- 云数据库MongoDB不支持在以下地域的可用区创建3.4、4.0和4.2版本的实例。 - 华东5（南京-本地地域）：南京-本地地域可用区A。 - 中国（香港）：香港可用区C。 - 泰国（曼谷）：曼谷可用区A。 - 韩国（首尔）：首尔可用区A。 - 云数据库MongoDB不支持在以下地域的可用区创建4.4和5.0版本的实例。 - 华东1（杭州）：杭州可用区J和K。 - 华东2（上海）：上海可用区L和D。 - 华北1（青岛）：青岛可用区B。 - 华北2（北京）：北京可用区E、L和K。 - 华北3（张家口）：张家口可用区B。 - 华北5（呼和浩特）：呼和浩特可用区A。 - 华北6（乌兰察布）：乌兰察布可用区A、B和C。 - 华南1（深圳）：深圳可用区A、D、C、（C+D+E）和F。 - 华南3（广州）：广州可用区A。 - 中国（香港）：香港可用区（B+C+D）。 - 澳大利亚（悉尼）：悉尼可用区A和B。 - 马来西亚（吉隆坡）：吉隆坡可用区A。 - 菲律宾（马尼拉）：马尼拉可用区A。 - 日本（东京）：东京可用区A和B。 - 印度（孟买）：孟买可用区A。 - 阿联酋（迪拜）：迪拜可用区A。 - 美国（硅谷）：硅谷可用区A。 - 英国（伦敦）：伦敦可用区A。 - 德国（法兰克福）：法兰克福可用区C。
搭建副本集	- 系统自动搭建的副本集架构中，包括三个节点。 - Primary节点（主节点）：负责执行和响应数据读写请求。 - Secondary节点（从节点）：通过操作日志（oplog）同步主节点的数据，可在主节点故障时通过选举成为新的主节点，保障高可用。 - Hidden节点（隐藏节点，对用户不可见）：通过操作日志（oplog）同步主节点的数据，可在从节点故障时接替该故障节点成为新的从节点，也可在只读节点故障时接替该故障节点成为新的只读节点，保障高可用。  说明 副本集架构的更多信息，请参见副本集架构。 - 实例运行过程中，您可以根据业务需求变更实例规格。如何变更，请参见变更副本集实例配置。

功能	约束
数据库小版本	如果您的数据库小版本过期或者不在维护列表内，当执行实例版本升级、数据迁移、变更实例配置或数据恢复等操作时，为保证提供更出色的性能和稳定性，系统会默认将您的数据库小版本升级至最新版。
数据库连接	云数据库MongoDB副本集实例的4.4和5.0版本不支持将经典网络切换为专有网络。
数据备份	- 云数据库MongoDB副本集实例的3.4、4.0和4.2版本支持物理备份和逻辑备份。  说明 当实例的数据库版本为云数据库MongoDB3.4时，实例中集合加索引的数量需控制在1万以内，否则可能造成物理备份失败。如果您的业务中集合加索引的数量可能会超过此限制，建议在创建实例时选择数据库版本为云数据库MongoDB4.0或4.2，也可以升级数据库版本至云数据库MongoDB4.04.2，升级方法请参见升级数据库版本。 - 云数据库MongoDB副本集实例的4.4和5.0版本支持快照备份。  说明 快照备份可以保留某时间点的磁盘数据状态。
数据恢复	仅三节点副本集实例支持恢复备份数据至当前实例，具体请参见 恢复备份数据至当前实例 。
数据安全性	- 支持设置白名单，设置方法请参见设置白名单。 - 支持设置SSL加密，设置方法请参见设置SSL加密。 - 仅云数据库MongoDB副本集实例的4.2及以下版本支持设置透明加密数据TDE，设置方法请参见设置透明数据加密TDE。 - 仅云数据库MongoDB副本集实例的4.2及以下版本副本集实例支持审计日志功能。  说明 云数据库MongoDB副本集实例的4.4和5.0版本不支持审计日志和TDE。
日志管理	云数据库MongoDB副本集实例的4.4和5.0版本不支持日志管理功能，包括慢日志、错误日志和运行日志。
CloudDBA（性能诊断与优化）	云数据库MongoDB副本集实例的4.4和5.0版本不支持慢日志。

功能	约束
数据迁移与同步	- 云数据库MongoDB支持使用以下方法迁移副本集实例数据。 - 自建数据库迁移至云数据库MongoDB - 将ECS上单节点或副本集架构的自建MongoDB迁移上云 - 使用DTS迁移副本集架构的自建MongoDB数据库上云 - 使用MongoDB工具将自建数据库迁移至副本集实例 - 第三方云数据库迁移至云数据库MongoDB - 使用NimoShake将Amazon DynamoDB迁移至阿里云 - 使用MongoDB工具将MongoDB Atlas迁移至阿里云 - 使用DTS将MongoDB Atlas数据库迁移至阿里云 - 使用DTS将华为云文档数据库迁移至阿里云 - 使用DTS将腾讯云MongoDB增量迁移至阿里云 - 使用DTS将腾讯云MongoDB副本集实例全量迁移至阿里云 - 使用MongoDB工具将腾讯云MongoDB迁移至阿里云 - 云数据库MongoDB实例间迁移 - 从MongoDB副本集实例迁移至分片集群实例 - 跨阿里云账号迁移MongoDB实例 - 迁移MongoDB实例至其他地域 - 云数据库MongoDB迁移至自建MongoDB数据库 - 逻辑备份恢复至自建数据库 - 将MongoDB物理备份文件恢复至自建数据库 - 云数据库MongoDB支持使用以下方法同步副本集实例数据。 - 云数据库MongoDB版（副本集架构）同步至云数据库MongoDB版（副本集架构或分片集群架构） - 使用MongoShake实现MongoDB副本集间的单向同步 - 使用MongoShake实现MongoDB实例间的延迟同步

分片集群实例

功能	约束
----	----

功能	约束
实例部署	- 云数据库MongoDB不支持在以下地域的可用区创建3.4、4.0和4.2版本的实例。 - 华东5（南京-本地地域）：南京-本地地域可用区A。 - 中国（香港）：香港可用区C。 - 泰国（曼谷）：曼谷可用区A。 - 韩国（首尔）：首尔可用区A。 - 云数据库MongoDB不支持在以下地域的可用区创建4.4和5.0版本的实例。 - 华东1（杭州）：杭州可用区J和K。 - 华东2（上海）：上海可用区L和D。 - 华北1（青岛）：青岛可用区B。 - 华北2（北京）：北京可用区E、L和K。 - 华北3（张家口）：张家口可用区B。 - 华北5（呼和浩特）：呼和浩特可用区A。 - 华北6（乌兰察布）：乌兰察布可用区A、B和C。 - 华南1（深圳）：深圳可用区A、D、C、（C+D+E）和F。 - 华南3（广州）：广州可用区A。 - 中国（香港）：香港可用区（B+C+D）。 - 澳大利亚（悉尼）：悉尼可用区A和B。 - 马来西亚（吉隆坡）：吉隆坡可用区A。 - 菲律宾（马尼拉）：马尼拉可用区A。 - 日本（东京）：东京可用区A和B。 - 印度（孟买）：孟买可用区A。 - 阿联酋（迪拜）：迪拜可用区A。 - 美国（硅谷）：硅谷可用区A。 - 英国（伦敦）：伦敦可用区A。 - 德国（法兰克福）：法兰克福可用区C。
搭建分片集群	- 在创建分片集群实例时，您可以选择Mongos和Shard节点的配置和节点数量。 - 实例运行过程中，您也可以添加或释放Mongos和Shard节点，具体请参见变更分片集群实例配置概览。  说明 - 当分片集群实例中Mongos节点数量为3个及以上时，支持释放，但须至少保留2个Mongos节点。 - 当分片集群实例中Shard节点数量为3个及以上时，支持释放，但须至少保留2个Shard节点。 - 释放Shard节点前，您需要确保剩余Shard节点完全可以分担释放节点的数据，否则，执行释放操作后，实例状态将会一直为删除节点中，无法对实例进行重置密码、角色切换、变更节点连接地址、修改参数等操作。

功能	约束
数据库小版本	如果您的数据库小版本过期或者不在维护列表内，当执行实例版本升级、数据迁移、变更实例配置或数据恢复等操作时，为保证提供更出色的性能和稳定性，系统会默认将您的数据库小版本升级至最新版。
数据读写	分片集群实例中的admin数据库仅支持用户读取，不支持写入。
数据备份	分片集群实例支持物理备份和逻辑备份。
数据恢复	仅支持按时间点创建实例的方式恢复数据，具体请参见 按时间点将备份数据恢复至新建实例 。
数据安全性	- 支持设置白名单，设置方法请参见设置白名单。 - 支持设置透明加密数据TDE，设置方法请参见设置透明数据加密TDE。 - 支持审计日志功能。 - 不支持设置SSL加密。  说明 DynamoDB协议兼容版分片集群实例仅支持专有网络，专有网络的创建方法请参见创建和管理专有网络。
CloudDBA（性能诊断与优化）	DynamoDB协议兼容版分片集群实例不支持CloudDBA功能，包括性能趋势、实时性能、实例会话、空间分析和慢日志。
数据迁移与同步	- 云数据库MongoDB支持使用以下方法迁移分片集群实例数据。 - 自建数据库迁移至云数据库MongoDB - 将ECS上分片集群架构的自建MongoDB迁移上云 - 使用DTS迁移分片集群架构的自建MongoDB数据库上云 - 使用MongoDB工具将自建数据库迁移至分片集群实例 - 第三方云数据库迁移至云数据库MongoDB - 使用NimoShake将Amazon DynamoDB迁移至阿里云 - 使用MongoDB工具将MongoDB Atlas迁移至阿里云 - 使用DTS将MongoDB Atlas数据库迁移至阿里云 - 云数据库MongoDB实例间迁移 - 迁移MongoDB实例至其他地域 - 云数据库MongoDB支持使用以下方法同步分片集群实例数据。 - 云数据库MongoDB版（分片集群架构）间的双向同步
回收站	分片集群实例不支持回收站功能。

Serverless实例

功能	约束
搭建Serverless	- 实例运行过程中，您可以根据业务需要变更实例规格，具体请参见变更Serverless实例配置。 - 不支持释放实例。
修改实例参数	Serverless实例不支持修改实例参数。
数据读写	Serverless实例中的admin数据库仅支持用户读取，不支持写入。
数据库连接	- 不支持开启或关闭私网免密访问功能。 - 不支持将经典网络切换为专有网络。
数据备份与恢复	仅支持通过云数据库MongoDB工具（mongodump和mongorestore）备份与恢复数据库，具体请参见 使用MongoDB工具备份与恢复MongoDB Serverless版实例 。
数据安全性	- 仅支持设置白名单，设置方法请参见设置白名单。 - 不支持设置SSL加密和透明加密数据TDE。 - 不支持审计日志功能。
日志管理	Serverless实例不支持日志管理功能，包括慢日志、错误日志和运行日志。
CloudDBA（性能诊断与优化）	Serverless实例不支持CloudDBA功能，包括性能趋势、实时性能、实例会话、空间分析和慢日志。
数据迁移与同步	云数据库MongoDB支持使用以下方法迁移Serverless实例数据。 使用MongoShake将Serverless数据迁移至副本集或分片集群实例 使用MongoDB工具备份与恢复MongoDB Serverless版实例
回收站	Serverless实例不支持回收站功能。
重启实例	Serverless实例不支持重启。

3. 单节点快速入门

3.1. 使用限制

在使用云数据库MongoDB单节点实例前，您需要了解单节点实例本身存在的约束信息，使您初步了解单节点实例的使用限制，帮助您进一步确认单节点实例是否能够满足您的业务场景。

功能	约束
实例部署	云数据库MongoDB不支持在以下地域的可用区创建单节点实例。 • 华东1（杭州）：杭州可用区J和K。 • 华东2（上海）：上海可用区L和D。 • 华东5（南京-本地地域）：南京-本地地域可用区A。 • 华北1（青岛）：青岛可用区B。 • 华北2（北京）：北京可用区E、L和K。 • 华北3（张家口）：张家口可用区A、B和C。 • 华北5（呼和浩特）：呼和浩特可用区A和B。 • 华北6（乌兰察布）：乌兰察布可用区A、B和C。 • 华南1（深圳）：深圳可用区A、D、C、（C+D+E）和F。 • 华南2（河源）：河源可用区A和B。 • 华南3（广州）：广州可用区A。 • 西南1（成都）：成都可用区A和B。 • 中国（香港）：香港可用区（B+C+D）、B、C和D。 • 澳大利亚（悉尼）：悉尼可用区A和B。 • 马来西亚（吉隆坡）：吉隆坡可用区A和B。 • 印尼（雅加达）：雅加达可用区A、B和C。 • 菲律宾（马尼拉）：马尼拉可用区A。 • 泰国（曼谷）：曼谷可用区A。 • 日本（东京）：东京可用区A和B。 • 韩国（首尔）：首尔可用区A。 • 印度（孟买）：孟买可用区A。 • 阿联酋（迪拜）：迪拜可用区A。 • 美国（硅谷）：硅谷可用区A和B。 • 美国（弗吉尼亚）：弗吉尼亚可用区A和B。 • 英国（伦敦）：伦敦可用区A。 • 德国（法兰克福）：法兰克福可用区A、B和C。
数据库版本	云数据库MongoDB仅3.4和4.0版本支持创建单节点实例。
数据库连接	不支持开启或关闭私网访问加密功能。

功能	约束
数据备份	仅支持快照备份的方式备份数据。  说明 快照备份可以保留某时间点的磁盘数据状态。
数据恢复	- 仅支持按备份点将备份数据回复至新建实例。如何恢复，请参见按备份点将备份数据恢复至新建实例。 - 不支持按时间点恢复备份数据。
服务可用性	不支持服务可用性功能，包括主备切换、角色切换和迁移可用区。
数据安全性	- 仅支持专有网络。如何创建专有网络，请参见创建和管理专有网络。 - 仅支持设置白名单。如何设置，请参见设置白名单。 - 不支持审计日志、SSL和TDE。
日志管理	不支持日志管理功能，包括慢日志、错误日志和运行日志。
CloudDBA（性能诊断与优化）	- 仅支持性能趋势、实时性能和实例会话。 - 不支持空间分析和慢日志。
数据迁移与同步	- 云数据库MongoDB支持使用以下方式迁移单节点实例。 - 自建数据库迁移至云数据库MongoDB 将ECS上单节点或副本集架构的自建MongoDB迁移上云 使用DTS迁移单节点架构的自建MongoDB数据库上云 使用MongoDB工具将自建数据库迁移至单节点实例 - 第三方云数据库迁移至云数据库MongoDB 使用DTS将华为云文档数据库迁移至阿里云 - 云数据库MongoDB实例间迁移 从MongoDB单节点实例迁移至副本集或分片集群实例 跨阿里云账号迁移MongoDB实例 迁移MongoDB实例至其他地域 - 单节点实例不支持增量数据迁移与同步。
连接实例	支持如下方法连接单节点实例。 通过DMS连接MongoDB单节点实例 通过Mongo Shell连接MongoDB单节点实例 通过程序代码连接MongoDB单节点实例

3.2. 创建单节点实例

云数据库MongoDB的单节点实例是阿里云为用户提供的一种高性价比实例，适用于存储企业非核心数据的场景，例如开发、测试、学习、培训等。本文介绍如何创建单节点实例。

前提条件

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

注意事项

- 单节点实例的故障恢复时间较长，无SLA保障。
- 如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS实例满足以下条件，以保证网络的连通性。ECS实例信息的查看方法请参见[查看实例信息](#)。
 - 所在地域相同，并建议所在可用区相同（以减少网络延时）。
 - 网络类型相同。

说明

- 专有网络具有更高的安全性，建议使用专有网络。
- 如果网络类型为专有网络，需确保使用相同的专有网络ID。
- 如果您需要使用专有网络，但ECS实例的网络类型为经典网络，您可以将ECS实例网络类型转换为专有网络，具体请参见[ECS实例从经典网络迁移到专有网络](#)。

使用限制

- 云数据库MongoDB仅支持在以下地域的可用区创建单节点实例。
 - 华东1（杭州）：杭州可用区G、H和I。
 - 华东2（上海）：上海可用区B和G。
 - 华北1（青岛）：青岛可用区C。
 - 华北2（北京）：北京可用区F和H。
 - 华南1（深圳）：深圳可用区E。
 - 新加坡：新加坡可用区A、B和C。
- 云数据库MongoDB仅3.4和4.0版本支持创建单节点实例。
- 单节点实例不支持增量数据迁移与同步、按时间点创建实例恢复数据功能。

费用说明

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

- 包年包月：预付费，在创建实例时就需要支付费用。
- 按量付费：后付费，每小时根据实例配置生成1个收费订单，并扣除相应的账户余额。

 说明 关于收费信息，请参见[收费项目及价格说明](#)。

如果您是首次购买云数据库MongoDB实例，您可以参与数据库首购折扣价活动，活动信息请参见[云数据库优惠活动](#)。

操作步骤

在执行完下列步骤之后，云数据库MongoDB会自动为您配置好数据库，您无需手动进行安装部署。

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击副本集实例列表。

3. 在副本集实例列表页面，单击创建实例。
4. 在商品类型区域，请根据业务需求，选择副本集（包年包月）或副本集（按量付费）。
5. 设置以下参数，完成单节点实例的参数配置。

参数	参数说明
地域	选择实例所属地域。支持创建单节点实例的地域信息，请参见使用限制。  说明 - 实例创建成功后将无法更换地域，请您谨慎选择。 - 如果应用部署在ECS实例，请您选择与ECS实例相同的地域，否则无法直接通过内网通信。
可用区	选择实例所属可用区。支持创建单节点实例的可用区信息，请参见使用限制。  说明 如果应用部署在ECS实例，请您选择与ECS实例相同的可用区，以减少网络延时。
数据库版本	请根据业务需求，选择对应数据库版本。 ◦ MongoDB 4.0 ◦ MongoDB 3.4
主备节点数	选择单节点。
只读节点数	固定为无只读节点。
存储引擎	固定为WiredTiger。
存储类型	固定为ESSD PL1云盘。
资源组	选择管理实例的资源组。您可以选择加入已经创建的资源组，也可以创建一个资源组加入，创建方法请参见创建资源组。  说明 通过设置资源组，您可以对单个云账号下多个地域、多种云资源进行集中的分组管理。
网络类型	固定为专有网络。  说明 - 专有网络VPC（Virtual Private Cloud）是一种隔离的网络环境，安全性和性能均高于传统的经典网络。 - 如果应用部署在ECS实例，您需要确保ECS实例的网络类型为专有网络，以保证网络的连通性。

参数	参数说明
专有网络	选择实例所属专有网络。如果您没有可以使用的专有网络，您可以在专有网络控制台创建专有网络，创建方法请参见 创建和管理专有网络 。
虚拟交换机	选择实例所属交换机。如果您没有可以使用的交换机，您可以在专有网络控制台创建交换机，创建方法请参见 创建和管理交换机 。
规格	选择实例的CPU和内存规格，关于单节点实例支持的规格信息，请参见 实例规格表 。
存储空间	选择实例的存储空间的大小。  说明 存储空间包含您的数据文件、系统文件、日志文件空间。
用户名	固定为root。  说明 admin数据库的root账号。
密码设置	设置root账号的密码。您可以选择以下任意一种方法： - 立即设置：立即设置密码。 - 创建后设置：在实例运行期间设置密码，设置方法请参见（可选）重置密码。
密码	如果您选择立即设置密码，请按照如下规则设置密码： - 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !@#%\$%^&*()_+== - 密码长度为8~32位。
购买时长	如果您创建的是包年包月实例，请您选择购买时长。包月可选择1~9个月，包年可选择1~5年。  说明 如果您希望实例到期后自动续费，勾选到期自动续费，并确保实例在即将到期前您的账户中有足够的余额。
购买数量	选择相同配置实例的数量，可以设置1~10的整数。

6. 根据实例的付费类型，选择对应操作购买实例。

付费类型	操作步骤
------	------

付费类型	操作步骤
包年包月	i. 配置完成后，单击立即购买。 ii. 在确认订单页面，查看配置详情，阅读并勾选服务协议。 iii. 单击去支付，并在支付页面，根据提示完成支付流程。  说明 您也可以单击加入购物车，与其他产品一起进行结算。结算方法请参见购物车结算。
按量付费	i. 配置完成后，单击立即购买。 ii. 在确认订单页面，查看配置详情，阅读并勾选服务协议。 iii. 单击立即开通，系统将在下一小时自动扣取对应金额。

7. 查看实例是否创建成功。

- 支付完成后，单击**管理控制台**，进入云数据库MongoDB控制台。
- 在页面左上角，选择主机所在的资源组和地域。
- 查看创建的实例是否在副本集实例列表中。
 - 是，说明实例已创建，当实例状态为**运行中**时，说明实例创建成功。
 - 否，请您等待10~15分钟后刷新页面重新查看，如果副本集列表中仍然没有，请先排查是否是以下原因导致，如果不是，请联系阿里云技术支持。

导致实例列表中找不到实例的可能原因及解决办法如下。

可能原因	解决方法
地域选择错误	在云数据库MongoDB控制台左上角，选择实例所在的资源组和地域。
实例列表选择错误	在云数据库MongoDB控制台左侧导航栏，选择 副本集实例列表 。
资源不足	新创建的实例可能因为资源不足而退款。请您在订单管理中查看退款，具体请参见订单管理。 在您确认退款后，您可以选择以下任意一种方法处理。 ■ 更换可用区尝试再次购买。 ■ 提交工单处理，具体请参见提交工单。

购物车结算

如果您购买的是包年包月实例，在实例配置完成后，您也可以单击**加入购物车**，与其他产品一起进行结算。结算方法如下。

- 单击页面右侧的**购物车**。
- 在**购物车**面板，勾选您需要购买的产品，并单击**去购物车结算**。
- 在**购物车**页面，确认您需要购买的产品选择是否正确。

 **说明** 您也可以重新选择云数据库MongoDB实例的购买时长和购买数量。

- 是：单击**立即购买**。
 - 否：勾选正确的购买产品，单击**立即购买**。
4. 在**确认订单**页面，阅读并勾选服务协议，并单击**确认购买**。
5. 在**支付**页面，根据提示完成支付流程。

相关API

接口	说明
CreateDBInstance	• 创建云数据库MongoDB单节点实例或副本集实例。 • 克隆云数据库MongoDB单节点实例或副本集实例。
DescribeInstanceAutoRenewalAttribute	查询云数据库MongoDB实例是否为自动付费。
DescribeDBInstanceAttribute	查询云数据库MongoDB实例的详情。

后续步骤

- 实例创建完成后，您还需要执行如下操作。
- 设置实例的白名单，以允许外部设备访问该实例，设置方法请参见[设置白名单](#)。
 - （可选）如果您在创建实例时没有立即设置密码，请您设置密码，设置方法请参见[（可选）重置密码](#)。
 - （可选）如果您需要通过公网连接实例，请您先申请公网连接地址，申请方法请参见[（可选）申请公网连接地址](#)。

3.3. （可选）重置密码

如果您忘记账号密码、修改旧密码，或创建实例时未设置密码，您可以通过云数据库MongoDB提供的重置密码功能重新设置密码。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击**副本集实例列表**。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的**管理**。
5. 在目标实例页面的左侧导航栏，单击**账号管理**。
6. 单击操作列的**重置密码**。
7. 在**重置密码**面板，设置以下参数。

参数	参数说明
账号	固定为root。

参数	参数说明
新密码	按照以下规则设置账号名对应的新密码。 - 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !#\$%^&*()_+= - 密码长度为8~32位。
确认新密码	再次输入新密码。

8. 单击**确定**。

3.4. 设置白名单

为保障MongoDB数据库的安全稳定，系统默认任何设备均无法访问云数据库MongoDB实例（默认IP地址为127.0.0.1），所以在使用云数据库MongoDB实例前，您需要为实例设置白名单，以允许外部设备访问该实例。正确使用白名单可以使云数据库MongoDB实例得到更高级别的安全保护，建议您定期维护白名单。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击**副本集实例列表**。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行**操作列**的**管理**。
5. 在目标实例页面的左侧导航栏，单击**数据安全性 > 白名单设置**。
6. 在**添加白名单分组**区域，选择以下任意一种方法为实例设置白名单。
 - o 手动添加

 说明

- 支持如下格式的IP地址或IP地址段。
 - 单个IP地址，例如10.23.12.24。
 - 0.0.0.0/0
 - **CIDR模式**，即无类域间路由。例如10.23.12.24/24，其中/24表示了地址中前缀的长度，范围[1, 32]。
- 多个IP地址或IP地址段之间以英文逗号（,）分隔。
- 设置为0.0.0.0/0或空表示对访问的IP地址来源不作任何限制，此时云数据库MongoDB数据库将面临高安全风险，请谨慎使用。

- a. 单击目标**分组操作列**的，并选择**手动修改**。
 - b. 在**手动修改**面板的**允许访问IP名单**输入框中输入IP地址或IP地址段。
 - c. 单击**确定**。
- o 加载ECS内网IP添加
 - a. 单击目标**分组操作列**的，并选择**加载ECS内网IP添加**。

- b. 在加载ECS内网IP添加面板的允许访问IP名单列表中，选择需要添加的ECS内网IP地址。
- c. 单击。
- d. 单击确定。

执行结果

添加白名单后，实例的基本信息页面和数据库连接页面将显示实例的专有网络连接地址。

相关任务

- [添加白名单分组](#)
- [添加安全组](#)

3.5. （可选）申请公网连接地址

MongoDB单节点实例支持申请公网连接地址，适用于通过公网连接实例的场景。

背景信息

MongoDB单节点实例支持的网络类型为专有网络和公网，关于连接地址的详细说明请参见下表。

地址类型	说明
专有网络连接地址	• 专有网络是一种隔离的网络环境，安全性和性能均高于传统的经典网络。 • MongoDB实例默认提供专有网络连接地址，通过专有网络连接MongoDB实例可以获取更高的安全性和连接性能。
公网连接地址	• 由于通过公网连接实例存在一定的安全风险，MongoDB实例默认未提供公网连接地址。 • 如果您要使用阿里云以外的设备（例如本地设备）连接MongoDB实例，您可以手动申请公网连接地址。

操作步骤

 **说明** 为保障数据安全性，如您不再需要公网连接地址，请及时[释放公网连接地址](#)。

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击**副本集实例列表**。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的**管理**。
5. 在目标实例页面的左侧导航栏，单击**数据库连接**。
6. 单击**公网连接**区域框右侧的**申请公网地址**。
7. 在弹出的**申请公网地址**对话框中，单击**确定**。

 **说明** 如果您需要通过申请的公网地址来连接实例，您还需要将连接实例的终端所在的公网IP地址加入到实例的白名单中，详情请参见[设置白名单](#)。

执行结果

申请操作完成后，您可以在以下位置查看公网连接地址。关于连接地址的信息，请参见[副本集实例连接说明](#)。

- 基本信息页面的连接信息区域。
- 数据库连接页面的公网连接区域。

更多信息

[如何通过公网连接MongoDB实例](#)

3.6. 连接实例

3.6.1. 通过DMS连接MongoDB单节点实例

数据管理服务DMS（Data Management Service）是一种集数据管理、结构管理、用户授权、安全审计、数据趋势、数据追踪、BI图表、性能优化和服务器管理于一体的可视化、图形化数据管理服务。通过DMS获取MongoDB数据库列表后，您可以更加方便快捷地远程访问、在线管理您的MongoDB数据库。

准备工作

将DMS服务器IP地址（100.104.0.0/16）加入至MongoDB实例的白名单中，详情请参见[设置白名单](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击[副本集实例列表](#)。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的管理。
5. 在目标实例页面右上方，单击[登录数据库](#)。
6. 在[登录实例](#)对话框中，填写相应信息。

登录实例

* 数据库类型

MongoDB

* 实例地区

华东1（杭州）

连接串地址

s-bp-...mongodb.rds.aliyuncs.com:3717

数据库名称

root

* 数据库账号

请输入数据库账户

* 数据库密码

.....

☐ 记住密码

测试连接

登录

取消

参数	说明
数据库类型	目标数据库的类型，系统默认为当前需要连接的数据库类型。
实例地区	目标实例所在的地域，系统默认为当前实例所在地域。
连接串地址	目标实例的连接串地址，系统默认为当前实例的连接串地址。
数据库名称	填入鉴权数据库名，即数据库账号所属的数据库。 说明 如果数据库账号填写的是root，那么对应的数据库名即为admin。 不建议在生产环境中直接使用root账号连接数据库。您可以根据业务需求创建用户并分配权限，详情请参见使用DMS管理MongoDB实例的账号。
数据库账号	目标数据库的账号，初始账号为 root 。
数据库密码	填入该数据库账号对应的密码。 说明 如果忘记root账号的密码，您可以重置密码。

7. （可选）单击**测试连接**，查看是否连接成功。
- 如果弹出**成功**对话框，说明连接成功，请单击**确认**。

如果弹出其他信息提示对话框，仔细阅读提示信息，单击**确认**，并根据提示信息修改配置直至连接成
- > 文档版本：20220705
- 26

功。

8. 单击登录。

 **说明** 如果您希望浏览器记住该账号的密码，可以先选中记住密码，然后单击登录。

常见的连接场景

- [如何通过公网连接MongoDB实例](#)
- [不同网络类型的ECS实例与MongoDB实例如何连接](#)
- [ECS实例与MongoDB实例地域不同时如何连接](#)
- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

相关问题

- [排查Mongo Shell登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查MongoDB CPU使用率高的问题](#)
- [如何查询及限制连接数](#)

3.6.2. 通过Mongo Shell连接MongoDB单节点实例

Mongo Shell是MongoDB自带的数据库管理工具，您可以将Mongo Shell工具下载并安装至客户端（本地服务器或ECS服务器），用于连接云数据库MongoDB单节点实例。本文介绍通过Mongo Shell连接云数据库MongoDB单节点实例的方法。

前提条件

已创建云数据库MongoDB单节点实例。

背景信息

云数据库MongoDB单节点实例提供的默认数据库admin，用于管理默认数据库账号root的信息，建议使用admin之外的数据库实现业务需求（即不对admin库进行任何操作）。

准备工作

- 下载并安装与云数据库MongoDB实例数据库版本相同的MongoDB至客户端（本地服务器或云服务器ECS实例），安装方法请参见[Install MongoDB](#)。

如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS实例满足以下条件，以保证网络的连通性。ECS实例信息的查看方法请参见[查看实例信息](#)。

- 所在地域相同，并建议所在可用区相同（以减少网络延时）。
- 网络类型为**专有网络**，需确保使用相同的专有网络ID。

 **说明** 如果ECS实例的网络类型为经典网络，您可以将ECS实例网络类型转换为专有网络，具体请参见[ECS实例从经典网络迁移到专有网络](#)。

- 将客户端IP地址添加至云数据库MongoDB实例的白名单中，添加方法请参见[设置白名单](#)。

 **说明** 如果您需要使用公网地址连接云数据库MongoDB实例，您可以手动申请公网地址，申请方法请参见 [（可选）申请公网连接地址](#)。

操作步骤

- 1. 登录[MongoDB管理控制台](#)。
- 2. 在左侧导航栏，单击**副本集实例列表**。
- 3. 在页面左上角，选择实例所在的资源组和地域。
- 4. 单击目标实例ID或目标实例所在行操作列的**管理**。
- 5. 在目标实例页面的左侧导航栏，单击**数据库连接**。
- 6. 查看并获取云数据库MongoDB单节点实例的连接地址。

私网连接 - 专有网络 

修改连接地址

节点	地址
Primary	dds- 
ConnectionStringURI	mongodb:// 

公网连接

释放公网地址

修改连接地址

节点	地址
Primary	dds- 
ConnectionStringURI	mongodb:// 

项目	说明
地址类型	◦ 私网连接：专有网络是一种隔离的网络环境，安全性和性能均高于传统的经典网络。云数据库MongoDB实例默认提供专有网络连接地址，通过专有网络连接云数据库MongoDB单节点实例可以获取更高的安全性和性能。 ◦ 公网连接：由于通过公网连接实例存在一定的安全风险，云数据库MongoDB实例默认未提供公网连接地址。如果您要使用公网地址连接云数据库MongoDB实例，您可以手动申请公网连接地址，申请方法请参见 （可选）申请公网连接地址。
Primary	格式： <host>:<port>  说明 ◦ <host>：云数据库MongoDB单节点实例的地址。 ◦ <port>：云数据库MongoDB单节点实例的端口号。 示例： dds-bp1ea17b41abecf43****.mongodb.rds.aliyuncs.com:3717

项目	说明
ConnectionString URI	格式： <pre>mongodb://<username>:<password>@<host>:<port>/<database></pre>  说明 ◦ <code>mongodb://</code>：前缀，代表这是一个ConnectionStringURI连接地址。 ◦ <code><username></code>：云数据库MongoDB单节点实例的账号名，默认为root。 ◦ <code><password></code>：云数据库MongoDB单节点实例账号名对应的密码。 ◦ <code><host></code>：云数据库MongoDB单节点实例的地址。 ◦ <code><port></code>：云数据库MongoDB单节点实例的端口号。 ◦ <code><database></code>：云数据库MongoDB单节点实例的鉴权数据库名，即数据库账号所属的数据库。默认为admin。如果您需要使用其他数据库，您可以先在该数据库中创建账号，然后使用该账户。创建账号的方法请参见db.createUser()。 示例： <pre>mongodb://root:****@dds-bplea17b41abecf43****.mongodb.rds.aliyuncs.com:3717/admin</pre>

7. 执行如下命令，在安装有MongoDB的客户端（本地服务器或ECS服务器）上连接云数据库MongoDB单节点实例。

```
mongo --host <host>:<port> -u <username> -p --authenticationDatabase <database>
```

 说明

- `<host>`：云数据库MongoDB单节点实例的地址。
- `<port>`：云数据库MongoDB单节点实例的端口号。
- `<username>`：云数据库MongoDB单节点实例的账号名，初始账号为root。不建议直接使用root账号连接数据库。您可以根据业务需求创建用户并分配权限，详情请参见[MongoDB数据库账号权限管理](#)。
- `<database>`：鉴权数据库名，即数据库账号所属的数据库。当数据库账号为root时，对应的数据库为admin。如果您希望指定其他数据库，请先在该数据库中使用[db.createUser\(\)](#)命令创建账号，然后再使用该账号进行连接。

示例：

```
mongo --host dds-bplea17b41abecf43****.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

8. 回显如下信息，输入数据库账号对应的密码并按Enter。

```
Enter password:<password>
```

? 说明

- `<password>`：云数据库MongoDB单节点实例账号名对应的密码。输入密码时，密码不可见。
- 如果忘记数据库账号对应的密码，您可以重新设置密码，设置方法请参见 [（可选）重置密码](#)。

回显如下类似信息，说明连接成功。

? 说明 使用不同版本MongoDB中的Mongo Shell连接实例后回显信息不同，请以实际情况为准。

```
connecting to: mongodb://dds-bp1ea17b41abecf433330.mongodb.rds.aliyuncs.com:3717/  
MongoDB server version: 3.4.6  
Welcome to the MongoDB shell.
```

? 说明 实例连接成功后，建议您使用admin之外的数据库实现业务需求。

常见的连接场景

- [如何通过公网连接MongoDB实例](#)
- [不同网络类型的ECS实例与MongoDB实例如何连接](#)
- [ECS实例与MongoDB实例地域不同时如何连接](#)
- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

相关问题

- [排查Mongo Shell登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查MongoDB CPU使用率高的问题](#)
- [如何查询及限制连接数](#)

3.6.3. 通过程序代码连接MongoDB单节点实例

云数据库MongoDB版完全兼容MongoDB协议，本文为您介绍使用不同语言的程序代码来连接MongoDB单节点实例。

前提条件

- 获取云数据库MongoDB副本集实例连接地址。具体操作，请参见[副本集实例连接说明](#)。
- 根据您使用的语言下载并安装官方驱动程序。更多信息，请参见[MongoDB Drivers](#)。

Node.js连接示例

Node.js驱动相关信息，请参见[MongoDB Node.js Driver](#)。

1. 在客户端执行如下命令，进行项目初始化。

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

2. 执行如下命令，安装驱动包以及工具包。

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

3. 获取云数据库MongoDB单节点实例连接信息。
4. Node.js Demo Code。

```
const sprintf = require("sprintf-js").sprintf;
const MongoClient = require('mongodb').MongoClient;
const host1 = "dds-*****.mongodb.rds.aliyuncs.com";
const port1 = 3717;
const host2 = "dds-*****.mongodb.rds.aliyuncs.com";
const port2 = 3717;
const username = "root";
const password = "*****";
const replSetName = "mgset-*****";
const demoDb = "test";
const demoColl = "testColl";
// 官方建议使用副本集地址，确保高可用。
const url = sprintf("mongodb://%s:%s@%s:%d,%s:%d/admin?replicaSet=%s",
    username, password, host1, port1, host2, port2, replSetName);
console.info("url:", url);
const client = new MongoClient(url);
// 获取MongoClient。
async function run() {
    try {
        // 连接实例。
        await client.connect();
        // 取得数据库句柄。
        const database = client.db(demoDb);
        // 取得Collection句柄。
        const collection = database.collection(demoColl);
        const demoName = "Node For Demo";
        const doc = { "DEMO": demoName, "MSG": "Hello AliCoudDB For MongoDB" };
        console.info("ready insert document: ", doc);
        // 插入数据。
        const result = await collection.insertOne(doc);
        console.log(
            `A document was inserted with the _id: ${result.insertedId}`,
        );
        // 读取数据。
        const filter = { "DEMO": demoName };
        const findResult = await collection.find(filter);
        await findResult.forEach(console.dir);
    } finally {
        // 关闭连接。
        await client.close();
    }
}
run().catch(console.dir);
```

PHP连接示例

PHP相关信息，请参见[MongoDB PHP Driver](#)。

1. 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"
```


2. 获取云数据库MongoDB单节点实例连接信息。

3. PHP Demo Code。

```
<?php
require 'vendor/autoload.php'; // include Composer goodies
# 实例信息。
$demo_seed1 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_seed2 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_replname = "mgset-*****";
$demo_user = 'root';
$demo_password = '*****';
$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 = $collection->insertOne(['name' => 'ApsaraDB for MongoDB', 'desc' => 'Hello, M
ongodb']);
echo "Inserted with Object ID '{$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](#)文档。
- 下载[jar包](#)。

1. 获取云数据库MongoDB单节点实例连接信息。

2. Java Demo Code。

o Maven配置。

```
<dependencies>
    <dependency>
        <groupId>org.mongodb</groupId>
        <artifactId>mongo-java-driver</artifactId>
        <version>4.1.0</version>
    </dependency>
</dependencies>
```

o 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 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("*****.mongodb.tbc
3.newtest.rdstest.aliyun-inc.com",
        27017);
    public static ServerAddress seed2 = new ServerAddress("*****.mongodb.tbc
3.newtest.rdstest.aliyun-inc.com",
        27017);

    public static String username = "demouser";
    public static String password = "*****";
    public static String ReplSetName = "mgset-*****";
    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(DEFAULT_DB,
            username, 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().requiredRep
licaSetName(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://" + u
sername + ":" + password + "@"
            + seed1 + "," + seed2 + "/" + DEFAULT_DB + "?replicaS
et=" + ReplSetName);
        return new MongoClient(connectionString);
    }
    public static void main(String args[]) {
        MongoClient client = createMongoDBClient();
        // or
        // MongoClient client = createMongoDBClientWithURI();
        try {
            // 取得Collection句柄。

```

```
// 获取Collection实例
MongoDatabase 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("MSG", "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).iterat
or();

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 4.1.0下载地址](#)。
- [PyMongo 4.1.0官方文档](#)。

1. 安装pymongo。

```
pip install pymongo
```

2. 获取云数据库MongoDB单节点实例连接信息。

3. Python Demo Code。

```
import uuid
from pymongo import MongoClient
CONN_ADDR1 = '*****.mongodb.****.*****.rdstest.aliyun-inc.com:27017'
CONN_ADDR2 = '*****.mongodb.****.*****.rdstest.aliyun-inc.com:27017'
REPLICAT_SET = 'mgset-*****'
username = 'demouser'
password = '*****'
#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
```

C#连接示例

C#驱动相关信息，请参见[MongoDB C# Driver](#)。

1. 在客户端执行如下命令，安装如下驱动包。

```
mongocsharpdriver.dll
```

2. 获取云数据库MongoDB单节点实例连接信息。
3. C# Demo Code。

```
using MongoDB.Driver;
using System;
using System.Collections.Generic;
namespace Aliyun
{
    class Program
    {
        static void Main(string[] args)
        {
            //Mongo 实例信息。
            const string host1 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";
            const int port1 = 3717;
            const string host2 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";
            const int port2 = 3717;
            const string replicaSetName = "mgset-300*****";
            const string admin = "admin";
            const string userName = "root";
            const string passwd = "*****";
            try
            {
                Console.WriteLine("开始连接.....");
                MongoClientSettings settings = new MongoClientSettings() {
```

```

MongoClientSettings settings = new MongoClientSettings();
List<MongoServerAddress> servers = new List<MongoServerAddress>();
servers.Add(new MongoServerAddress(host1, port1));
servers.Add(new MongoServerAddress(host2, port2));
settings.Servers = servers;
//设置副本集名称。
settings.ReplicaSetName = replicaSetName;
//设置超时时间为3秒。
settings.ConnectTimeout = new TimeSpan(0, 0, 0, 3, 0);
MongoCredential credentials = MongoCredential.CreateCredential(admin, u
serName, passwd);
settings.Credential = credentials;
MongoClient client = new MongoClient(settings);
var server = client.GetServer();
MongoDatabase database = server.GetDatabase("test");
var collection = database.GetCollection<User>("test_collection");
User user = new User();
user.id = "1";
user.name = "mongo_test";
user.sex = "女";
//插入数据user。
collection.Insert(user);
//获取一条数据。
User result = collection.FindOne();
Console.WriteLine("id:" + result.id + " name:" + result.name + " sex:" +
result.sex);
    Console.WriteLine("连接成功.....");
}
catch (Exception e)
{
    Console.WriteLine("连接异常:" + e.Message);
}
}
}
class User
{
    public string id { set; get; }
    public string name { set; get; }
    public string sex { set; get; }
}
}

```

Go连接示例

Go驱动相关信息，请参见[MongoDB Go Driver](#)。

1. 安装如下驱动包。

```
go get go.mongodb.org/mongo-driver
```

2. 获取云数据库MongoDB单节点实例连接信息。
3. Go Demo Code。

```

package main
import (
    "context"

```

```

"context"
    "fmt"
    "go.mongodb.org/mongo-driver/bson"
    "go.mongodb.org/mongo-driver/mongo"
    "go.mongodb.org/mongo-driver/mongo/options"
    "go.mongodb.org/mongo-driver/mongo/readconcern"
    "log"
)
func main() {
    // Create a Client to a MongoDB server and use Ping to verify that the
    // server is running.
    clientOpts := options.Client().ApplyURI("mongodb://root:****@dds-bp1*****.mongodb
.rds.aliyuncs.com:3717/admin")
    client, err := mongo.Connect(context.TODO(), clientOpts)
    if err != nil {
        fmt.Println("connect failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("connect successful!")
    defer func() {
        if err = client.Disconnect(context.TODO()); err != nil {
            fmt.Println("disconnect failed!")
            log.Fatal(err)
        }
        fmt.Println("disconnect successful!")
    }()
    // Call Ping to verify that the deployment is up and the Client was
    // configured successfully. As mentioned in the Ping documentation, this
    // reduces application resiliency as the server may be temporarily
    // unavailable when Ping is called.
    if err = client.Ping(context.TODO(), nil); err != nil {
        fmt.Println("ping failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("ping successful!")
    // Specify the DefaultReadConcern option so any transactions started through
    // the session will have read concern majority.
    // The DefaultReadPreference and DefaultWriteConcern options aren't
    // specified so they will be inherited from client and be set to primary
    // and majority, respectively.
    opts := options.Session().SetDefaultReadConcern(readconcern.Majority())
    sess, err := client.StartSession(opts)
    if err != nil {
        fmt.Println("start session failed!")
        log.Fatal(err)
        return
    }
    defer func() {
        sess.EndSession(context.TODO())
        fmt.Println("end session!")
    }()
    fmt.Println("start session successful!")
    txnOpts := options.Transaction()

```

```

// Open a transaction
result, err := sess.WithTransaction(
    context.TODO(),
    func(sessCtx mongo.SessionContext) (interface{}, error) {
        collection := client.Database("baz").Collection("qux")
        res, err := collection.InsertOne(context.Background(), bson.M{"hello": "world"})
        if err != nil {
            fmt.Println("insert result failed!")
            log.Fatal(err)
            return nil, err
        }
        id := res.InsertedID
        fmt.Println("Id: ", id)
        fmt.Printf("insert result: %v\n", res)
        result := bson.M{}
        filter := bson.D{{"_id", res.InsertedID}}
        if err := collection.FindOne(context.Background(), filter).Decode(&result);
err != nil {
            fmt.Println("find failed!")
            log.Fatal(err)
            return nil, err
        }
        return result, err
    },
    txnOpts)
if err == nil {
    fmt.Printf("result: %v\n", result)
}
}

```

4.副本集快速入门

4.1. 使用限制

在使用云数据库MongoDB副本集实例前，您需要了解副本集实例本身存在的约束信息，使您初步了解副本集实例的使用限制，帮助您进一步确认副本集实例是否能够满足您的业务场景。

功能	MongoDB 4.4和5.0版本约束	MongoDB 4.2及以下版本约束
实例部署	云数据库MongoDB不支持在以下地域的可用区创建4.4和5.0版本的实例。 - 华东1（杭州）：杭州可用区J和K。 - 华东2（上海）：上海可用区L和D。 - 华北1（青岛）：青岛可用区B。 - 华北2（北京）：北京可用区E、L和K。 - 华北3（张家口）：张家口可用区B。 - 华北5（呼和浩特）：呼和浩特可用区A。 - 华北6（乌兰察布）：乌兰察布可用区A、B和C。 - 华南1（深圳）：深圳可用区A、D、C、（C+D+E）和F。 - 华南3（广州）：广州可用区A。 - 中国（香港）：香港可用区（B+C+D）。 - 澳大利亚（悉尼）：悉尼可用区A和B。 - 马来西亚（吉隆坡）：吉隆坡可用区A。 - 菲律宾（马尼拉）：马尼拉可用区A。 - 日本（东京）：东京可用区A和B。 - 印度（孟买）：孟买可用区A。 - 阿联酋（迪拜）：迪拜可用区A。 - 美国（硅谷）：硅谷可用区A。 - 英国（伦敦）：伦敦可用区A。 - 德国（法兰克福）：法兰克福可用区C。	云数据库MongoDB不支持在以下地域的可用区创建4.2及以下版本的实例。 - 华东5（南京-本地地域）：南京-本地地域可用区A。 - 中国（香港）：香港可用区C。 - 泰国（曼谷）：曼谷可用区A。 - 韩国（首尔）：首尔可用区A。
搭建副本集	- 系统自动搭建的副本集架构中，包括三个节点。 - Primary节点（主节点）：负责执行和响应数据读写请求。 - Secondary节点（从节点）：通过操作日志（oplog）同步主节点的数据，可在主节点故障时通过选举成为新的主节点，保障高可用。 - Hidden节点（隐藏节点，对用户不可见）：通过操作日志（oplog）同步主节点的数据，可在从节点故障时接替该故障节点成为新的从节点，也可在只读节点故障时接替该故障节点成为新的只读节点，保障高可用。  说明 副本集架构的更多信息，请参见副本集架构。 - 实例运行过程中，您可以根据业务需求变更实例规格。如何变更，请参见变更副本集实例配置。	

功能	MongoDB 4.4和5.0版本约束	MongoDB 4.2及以下版本约束
数据库小版本	如果您的数据库小版本过期或者不在维护列表内，当执行实例版本升级、数据迁移、变更实例配置或数据恢复等操作时，为保证提供更出色的性能和稳定性，系统会默认将您的数据库小版本升级至最新版。	
数据库连接	不支持切换网络类型。	- 不支持专有网络切换为经典网络。 - 支持经典网络切换为专有网络，切换方法请参见经典网络切换为专有网络。
数据备份	- 不支持物理备份和逻辑备份。 - 仅支持快照备份。  说明 快照备份可以保留某一时点的磁盘数据状态。	- 不支持快照备份。 - 支持物理备份和逻辑备份。  说明 当实例的数据库版本为MongoDB 3.2或MongoDB 3.4时，实例中集合加索引的数量需控制在1万以内，否则可能造成物理备份失败。如果您的业务可能会超过此限制，您可以在创建实例时选择数据库版本为MongoDB 4.0或MongoDB 4.2，也可以升级数据库版本至MongoDB 4.0或MongoDB 4.2，升级方法请参见升级数据库版本。
数据恢复	仅三节点副本集实例支持恢复备份数据至当前实例，具体请参见 恢复备份数据至当前实例 。	
数据安全性	- 不支持设置透明加密数据TDE。 - 不支持审计日志功能。 - 支持设置白名单，设置方法请参见设置白名单。 - 支持设置SSL加密，设置方法请参见设置SSL加密。	- 支持设置白名单，设置方法请参见设置白名单。 - 支持设置SSL加密，设置方法请参见设置SSL加密。 - 支持设置透明加密数据TDE，设置方法请参见设置透明数据加密TDE。 - 支持审计日志功能。
日志管理	不支持日志管理功能，包括慢日志、错误日志和运行日志。	无
CloudDBA（性能诊断与优化）	不支持慢日志。	无

功能	MongoDB 4.4和5.0版本约束	MongoDB 4.2及以下版本约束
数据迁移与同步	- 云数据库MongoDB支持使用以下方法迁移副本集实例数据。 - 自建数据库迁移至云数据库MongoDB - 将ECS上单节点或副本集架构的自建MongoDB迁移上云 - 使用DTS迁移副本集架构的自建MongoDB数据库上云 - 使用MongoDB工具将自建数据库迁移至副本集实例 - 第三方云数据库迁移至云数据库MongoDB - 使用NimoShake将Amazon DynamoDB迁移至阿里云 - 使用MongoDB工具将MongoDB Atlas迁移至阿里云 - 使用DTS将MongoDB Atlas数据库迁移至阿里云 - 使用DTS将华为云文档数据库迁移至阿里云 - 使用DTS将腾讯云MongoDB增量迁移至阿里云 - 使用DTS将腾讯云MongoDB副本集实例全量迁移至阿里云 - 使用MongoDB工具将腾讯云MongoDB迁移至阿里云 - 云数据库MongoDB实例间迁移 - 从MongoDB副本集实例迁移至分片集群实例 - 跨阿里云账号迁移MongoDB实例 - 迁移MongoDB实例至其他地域 - 云数据库MongoDB迁移至自建MongoDB数据库 - 逻辑备份恢复至自建数据库 - 将MongoDB物理备份文件恢复至自建数据库 - 云数据库MongoDB支持使用以下方法同步副本集实例数据。 - 云数据库MongoDB版（副本集架构）同步至云数据库MongoDB版（副本集架构或分片集群架构） - 使用MongoShake实现MongoDB副本集间的单向同步 - 使用MongoShake实现MongoDB实例间的延迟同步	
连接实例	云数据库MongoDB支持使用以下方法连接副本集实例。 - 通过DMS连接MongoDB副本集实例 - 通过Mongo Shell连接云数据库MongoDB副本集实例 - 通过程序代码连接MongoDB副本集实例	

4.2. 创建副本集实例

云数据库MongoDB提供的副本集实例适用于读多写少或有临时活动的突发业务场景，例如阅读类网站、订单查询系统等。本文介绍云数据库MongoDB副本集实例的创建方法。

前提条件

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

注意事项

如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS实例满足以下条件，以保证网络的连通性。ECS实例信息的查看方法请参见[查看实例信息](#)。

- 所在地域相同，并建议所在可用区相同（以减少网络延时）。
- 网络类型相同。

② 说明

- 专有网络具有更高的安全性，建议使用专有网络。
- 如果网络类型为专有网络，需确保使用相同的专有网络ID。
- 如果您需要使用专有网络，但ECS实例的网络类型为经典网络，您可以将ECS实例网络类型转换为专有网络，具体请参见[ECS实例从经典网络迁移到专有网络](#)。

使用限制

- 仅支持在以下地域的可用区创建云数据库MongoDB3.4、4.0和4.2版本的实例。
 - 华东1（杭州）：杭州可用区G、H、I、J和K。
 - 华东2（上海）：上海可用区L、B、D和G。
 - 华北1（青岛）：青岛可用区B和C。
 - 华北2（北京）：北京可用区F、E、H、L和K。
 - 华北3（张家口）：张家口可用区A、B和C。
 - 华北5（呼和浩特）：呼和浩特可用区A和B。
 - 华北6（乌兰察布）：乌兰察布可用区A、B和C。
 - 华南1（深圳）：深圳可用区A、D、C、E、（C+D+E）和F。
 - 华南2（河源）：河源可用区A和B。
 - 华南3（广州）：广州可用区A。
 - 西南1（成都）：成都可用区A和B。
 - 中国（香港）：香港可用区B、D和（B+C+D）。
 - 新加坡：新加坡可用区A、B和C。
 - 澳大利亚（悉尼）：悉尼可用区A和B。
 - 马来西亚（吉隆坡）：吉隆坡可用区A和B。
 - 印尼（雅加达）：雅加达可用区A、B和C。
 - 菲律宾（马尼拉）：马尼拉可用区A。
 - 日本（东京）：东京可用区A和B。
 - 印度（孟买）：孟买可用区A。
 - 阿联酋（迪拜）：迪拜可用区A。
 - 美国（硅谷）：硅谷可用区A和B。

- 美国（弗吉尼亚）：弗吉尼亚可用区A和B。
- 英国（伦敦）：伦敦可用区A。
- 德国（法兰克福）：法兰克福可用区A、B和C。
- 仅支持在以下地域的可用区创建云数据库MongoDB4.4和5.0版本的实例。
 - 华东1（杭州）：杭州可用区G、H和I。
 - 华东2（上海）：上海可用区B和G。
 - 华东5（南京-本地地域）：南京-本地地域可用区A。
 - 华北1（青岛）：青岛可用区C。
 - 华北2（北京）：北京可用区F和H。
 - 华北3（张家口）：张家口可用区A和C。
 - 华北5（呼和浩特）：呼和浩特可用区B。
 - 华南1（深圳）：深圳可用区E。
 - 华南2（河源）：河源可用区A和B。
 - 西南1（成都）：成都可用区A和B。
 - 中国（香港）：香港可用区B、C和D。
 - 新加坡：新加坡可用区A、B和C。
 - 马来西亚（吉隆坡）：吉隆坡可用区B。
 - 印尼（雅加达）：雅加达可用区A、B和C。
 - 泰国（曼谷）：曼谷可用区A。
 - 韩国（首尔）：首尔可用区A。
 - 美国（硅谷）：硅谷可用区B。
 - 美国（弗吉尼亚）：弗吉尼亚可用区A和B。
 - 德国（法兰克福）：法兰克福可用区A和B。

费用说明

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

- 包年包月：预付费，在创建实例时就需要支付费用。
- 按量付费：后付费，每小时根据实例配置生成1个收费订单，并扣除相应的账户余额。

② 说明 关于收费信息，请参见[收费项目及价格说明](#)。

如果您是首次购买云数据库MongoDB实例，您可以参与数据库首购折扣价活动，活动信息请参见[云数据库优惠活动](#)。

操作步骤

在执行完下列步骤之后，云数据库MongoDB会自动为您配置好数据库，您无需手动进行安装部署。

1. 登录[MongoDB管理控制台](#)。
2. 在副本集实例列表页面，单击创建实例。
3. 在商品类型区域，请根据业务需求，选择副本集（包年包月）或副本集（按量付费）。
4. 设置以下参数，完成副本集实例的参数配置。

参数	参数说明
地域	选择实例所属地域。  说明 - 实例创建成功后将无法更换地域，请您谨慎选择。 - 如果应用部署在ECS实例，请您选择与ECS实例相同的地域，否则无法直接通过内网通信。
可用区	选择实例所属可用区。  说明 - 如果应用部署在ECS实例，建议您选择与ECS实例相同的可用区，以减少网络延时。如果是MongoDB 4.2及以下版本的实例，您可以迁移可用区，迁移方法请参见迁移可用区。 - 您可以选择多可用区实现实例的同城容灾，详情请参见创建多可用区副本集实例。
数据库版本	请根据业务需求，选择对应数据库版本。 - MongoDB 5.0 - MongoDB 4.4 - MongoDB 4.2 - MongoDB 4.0 - MongoDB 3.4  说明 您可以在实例运行期间手动升级数据库版本，升级方法请参见升级数据库版本。
主备节点数	根据业务需求，选择副本集实例的节点数量。  说明 MongoDB 5.0和MongoDB 4.4版本副本集实例的主备节点数目前仅支持三节点。
只读节点数	根据业务需求，选择副本集实例的只读节点数量。更多只读节点的信息请参见MongoDB只读节点简介。  说明 MongoDB 5.0和MongoDB 4.4版本副本集实例暂不支持扩展只读节点。
存储引擎	固定为WiredTiger。

参数	参数说明
存储类型	根据业务需要，选择副本集实例的存储类型。 - ESSDPL1云盘。 - SSD本地盘。
资源组	选择管理实例的资源组。您可以选择加入已经创建的资源组，也可以创建一个资源组加入，资源组的创建方法请参见创建资源组。  说明 通过设置资源组，您可以对单个云账号下多个地域、多种云资源进行集中的分组管理。
网络类型	固定为专有网络。  说明 如果应用部署在ECS实例，您需要确保ECS实例的网络类型为专有网络，以保证网络的连通性。
专有网络	选择实例所属专有网络。如果您没有可以使用的专有网络，您可以在专有网络控制台创建专有网络，创建方法请参见创建和管理专有网络。
虚拟交换机	选择实例所属交换机。如果您没有可以使用的交换机，您可以在专有网络控制台创建交换机，创建方法请参见创建和管理交换机。
规格	选择实例的CPU和内存规格，关于副本集支持的规格信息，请参见实例规格表。
存储空间	选择实例的存储空间的大小。  说明 - 该存储空间的大小为每个独立节点的存储空间大小，不是所有节点存储空间大小的总和。 - 用于存放云数据库MongoDB实例的数据文件、系统文件和日志文件。
用户名	固定为root。  说明 admin数据库的root账号。
密码设置	设置root账号的密码。您可以选择以下任意一种方法： 立即设置：立即设置密码。 创建后设置：在实例运行期间设置密码，设置方法请参见 （可选）重置密码。

参数	参数说明
密码	如果您选择立即设置密码，请按照如下规则设置密码： - 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !@#\$%^&*()_+~= - 密码长度为8~32位。
购买时长	如果您创建的是包年包月实例，请您选择购买时长。包月可选择1~9个月，包年可选择1~5年。  说明 如果您希望实例到期后自动续费，勾选到期自动续费，并确保实例在即将到期前您的账户中有足够的余额。
购买数量	选择相同配置实例的数量，可以设置1~10的整数。

5. 根据实例的付费类型，选择对应操作购买实例。

付费类型	操作步骤
包年包月	- 配置完成后，单击立即购买。 - 在确认订单页面，查看配置详情，阅读并勾选服务协议。 - 单击去支付，并在支付页面，根据提示完成支付流程。  说明 您也可以单击加入购物车，与其他产品一起进行结算。结算方法请参见购物车结算。
按量付费	- 配置完成后，单击立即购买。 - 在确认订单页面，查看配置详情，阅读并勾选服务协议。 - 单击立即开通，系统将在下一小时自动扣取对应金额。

6. 查看副本集实例是否创建成功。

- 支付完成后，单击**管理控制台**，进入云数据库MongoDB控制台。
- 在页面左上角，选择实例所在的资源组和地域。
- 在左侧导航栏，选择**副本集实例列表**。

iv. 查看创建的实例是否在副本集实例列表中。

- 是，说明实例已创建，当实例状态为运行中时，说明实例创建成功。
- 否，请您等待10~15分钟后刷新页面重新查看，如果副本集列表中仍然没有，请先排查是否是以下原因导致，如果不是，请联系阿里云技术支持。

导致实例列表找不到实例的可能原因及解决办法如下。

可能原因	解决方法
地域选择错误	在云数据库MongoDB控制台左上角，选择实例所在的资源组和地域。
实例列表选择错误	在云数据库MongoDB控制台左侧导航栏，选择副本集实例列表。
资源不足	新创建的实例可能因为资源不足而退款。请您在订单管理中查看退款，具体请参见订单管理。 在您确认退款后，您可以选择以下任意一种方法处理。 ■ 更换可用区尝试再次购买。 ■ 提交工单处理，具体请参见提交工单。

购物车结算

如果您购买的是包年包月实例，在实例配置完成后，您也可以单击**加入购物车**，与其他产品一起进行结算。结算方法如下：

1. 单击页面右侧的**购物车**。
2. 在**购物车**面板，勾选您需要购买的产品，并单击**去购物车结算**。
3. 在**购物车**页面，确认您需要购买的产品选择是否正确。

 **说明** 您也可以重新选择云数据库MongoDB实例的购买时长和购买数量。

- 是：单击**立即购买**。
 - 否：勾选正确的购买产品，单击**立即购买**。
4. 在**确认订单**页面，阅读并勾选服务协议，并单击**确认购买**。
 5. 在**支付**页面，根据提示完成支付流程。

相关API

接口	说明
CreateDBInstance	• 创建云数据库MongoDB单节点实例或副本集实例。 • 克隆云数据库MongoDB单节点实例或副本集实例。
DescribeInstanceAutoRenewalAttribute	查询云数据库MongoDB实例是否为自动付费。
DescribeDBInstanceAttribute	查询云数据库MongoDB实例的详情。

后续步骤

- 实例创建完成后，您还需执行以下操作：
- （可选）如果您在创建实例时没有立即设置密码，请您设置密码，设置方法请参见（可选）重置密码。
 - 设置实例的白名单，以允许外部设备访问该实例，设置方法请参见设置白名单。
 - （可选）如果您需要通过公网连接实例，请您先申请公网连接地址，申请方法请参见（可选）申请公网连接地址。

4.3. （可选）重置密码

如果您忘记账号密码、修改旧密码，或创建实例时未设置密码，您可以通过云数据库MongoDB提供的重置密码功能重新设置密码。

操作步骤

1. 登录MongoDB管理控制台。
2. 在左侧导航栏，单击副本集实例列表。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的管理。
5. 在目标实例页面的左侧导航栏，单击账号管理。
6. 单击操作列的重置密码。
7. 在重置密码面板，设置以下参数。

参数	参数说明
账号	固定为root。
新密码	按照以下规则设置账号名对应的新密码。 ◦ 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !#\$%^&*()_+= ◦ 密码长度为8~32位。
确认新密码	再次输入新密码。

8. 单击确定。

4.4. 设置白名单

为保障MongoDB数据库的安全稳定，系统默认任何设备均无法访问云数据库MongoDB实例（默认IP地址为127.0.0.1），所以在使用云数据库MongoDB实例前，您需要为实例设置白名单，以允许外部设备访问该实例。正确使用白名单可以使云数据库MongoDB实例得到更高级别的安全保护，建议您定期维护白名单。

操作步骤

1. 登录MongoDB管理控制台。
2. 在左侧导航栏，单击副本集实例列表。
3. 在页面左上角，选择实例所在的资源组和地域。

- 4. 单击目标实例ID或目标实例所在行操作列的管理。
- 5. 在目标实例页面的左侧导航栏，单击数据安全性 > 白名单设置。
- 6. 在添加白名单分组区域，选择以下任意一种方法为实例设置白名单。
 - 手动添加

?

说明

■ 支持如下格式的IP地址或IP地址段。

- 单个IP地址，例如10.23.12.24。
- 0.0.0.0/0
- CIDR模式，即无类域间路由。例如10.23.12.24/24，其中/24表示了地址中前缀的长度，范围[1，32]。

- 多个IP地址或IP地址段之间以英文逗号（,）分隔。
- 设置为0.0.0.0/0或空表示对访问的IP地址来源不作任何限制，此时云数据库MongoDB数据库将面临高安全风险，请谨慎使用。

- a. 单击目标分组操作列的，并选择手动修改。
- b. 在手动修改面板的允许访问IP名单输入框中输入IP地址或IP地址段。
- c. 单击确定。
- 加载ECS内网IP添加
 - a. 单击目标分组操作列的，并选择加载ECS内网IP添加。
 - b. 在加载ECS内网IP添加面板的允许访问IP名单列表中，选择需要添加的ECS内网IP地址。
 - c. 单击。
 - d. 单击确定。

相关任务

- 添加白名单分组
- 添加安全组

4.5. （可选）申请公网连接地址

云数据库MongoDB支持公网连接地址，您可以为实例申请公网连接地址用于使用公网连接云数据库MongoDB实例的场景。本文介绍为云数据库MongoDB副本集实例申请公网连接地址的方法。

背景信息

云数据库MongoDB实例支持的连接地址类型及对应说明如下。

连接地址类型	说明
--------	----

> 文档版本：20220705

50

连接地址类型	说明
专有网络连接地址	• 专有网络是一种隔离的网络环境，安全性和性能均高于传统的经典网络。 • 云数据库MongoDB实例默认提供专有网络连接地址，通过专有网络连接云数据库MongoDB实例可以获取更高的安全性和性能。
经典网络连接地址	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。如果需要切换为更安全的专有网络，请参见经典网络切换为专有网络。  说明 DynamoDB协议的分片集群实例不支持经典网络。
公网连接地址	• 由于通过公网连接实例存在一定的安全风险，云数据库MongoDB实例默认提供专有网络连接地址。 • 如果您要使用阿里云以外的设备（例如本地设备）连接云数据库MongoDB实例，您可以手动申请公网连接地址。

注意事项

- 申请公网连接地址过程中，可能会出现实例重新启动的情况，建议您在业务低峰期进行操作。
- 如果需要通过申请的公网地址来连接实例，您需要将连接实例的终端所在的公网IP地址加入到实例的白名单中，白名单设置方法请参见[设置白名单](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击[副本集实例列表](#)。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的[管理](#)。
5. 在目标实例页面的左侧导航栏，单击[数据库连接](#)。
6. 单击公网连接区域框右侧的[申请公网地址](#)。
7. 在弹出的[申请公网地址](#)对话框中，单击[确定](#)。

执行结果

申请操作完成后，您可以在以下位置查看公网连接地址。关于连接地址的信息，请参见[副本集实例连接说明](#)。

- [基本信息](#)页面的[连接信息](#)区域。
- [数据库连接](#)页面的[公网连接](#)区域。

相关文档

- 通过公网地址连接实例的方法请参见[如何通过公网连接MongoDB实例](#)。
- 为保障数据安全性，如果您不再需要公网连接地址，您可以将公网地址释放，释放方法请参见[释放公网连接地址](#)。
- 通过公网地址连接数据库时，建议使用SSL加密，加密方法请参见[使用Mongo Shell通过SSL加密连接数据库](#)。

4.6. 连接实例

4.6.1. 通过DMS连接MongoDB副本集实例

数据管理服务DMS（Data Management Service）是一种集数据管理、结构管理、用户授权、安全审计、数据趋势、数据追踪、BI图表、性能优化和服务器管理于一体的可视化、图形化数据管理服务。通过DMS获取MongoDB数据库列表后，您可以更加方便快捷地远程访问、在线管理您的MongoDB数据库。

准备工作

根据MongoDB实例的网络类型，将DMS服务器的IP地址加入至MongoDB实例的白名单中，详情请参见[设置白名单](#)。如需连接MongoDB的只读节点，请事先创建只读节点，详情请参见[创建副本集实例](#)或[变更副本集实例配置](#)。

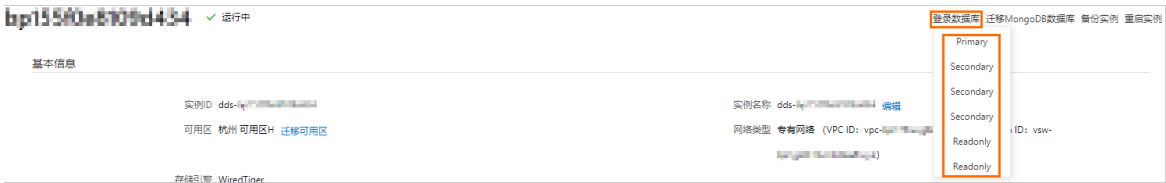
 **说明** 如您已经将DMS服务器的IP地址加入至MongoDB实例的白名单中，可跳过此步骤。

DMS服务器的IP地址

MongoDB实例的网络类型	DMS服务器的IP地址
专有网络	100.104.0.0/16
经典网络	120.55.177.0/24
	121.43.18.0/24
	101.37.74.0/24
	10.153.176.0/24
	10.137.42.0/24
	11.193.54.0/24
	10.152.163.0/24

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击[副本集实例列表](#)。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的[管理](#)。
5. 在目标实例页面右上方，单击[登录数据库](#)。并选择要登录的数据库节点（Primary、Secondary或Readonly），跳转到数据管理DMS页面。



说明

- Primary节点：副本集实例中的主节点，连接该节点可执行数据库的读写操作。
- Secondary节点：副本集实例中的从节点，连接该节点仅能执行数据库的读操作。
- Readonly节点：副本集实例中的只读节点，连接该节点仅能执行数据库的读操作。只读节点的详情请参见[MongoDB只读节点简介](#)。

6. 在登录实例对话框中，填写相应信息。

登录实例

* 数据库类型

MongoDB

* 实例地区

华东1（杭州）

连接串地址

s-bp-...mongodb.rds.aliyuncs.com:3717

数据库名称

root

* 数据库账号

请输入数据库账户

* 数据库密码

.....

☐ 记住密码

测试连接

登录

取消

参数	说明
数据库类型	目标数据库的类型，系统默认为当前需要连接的数据库类型。
实例地区	目标实例所在的地域，系统默认为当前实例所在地域。
连接串地址	目标实例的连接串地址，系统默认为当前实例的连接串地址。
数据库名称	填入鉴权数据库名，即数据库账号所属的数据库。 说明 - 如果数据库账号填写的是root，那么对应的数据库名即为admin。 - 不建议在生产环境中直接使用root账号连接数据库。您可以根据业务需求创建用户并分配权限，详情请参见使用DMS管理MongoDB实例的账号。
数据库账号	目标数据库的账号，初始账号为 root 。

参数	说明
数据库密码	填入该数据库账号对应的密码。  说明 如果忘记root账号的密码，您可以重置密码。

7. 单击登录。

 **说明** 若您希望浏览器记住该账号的密码，可以先勾选记住密码，再单击登录。

常见的连接场景

- [如何通过公网连接MongoDB实例](#)
- [不同网络类型的ECS实例与MongoDB实例如何连接](#)
- [ECS实例与MongoDB实例地域不同时如何连接](#)
- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

相关问题

- [排查 Mongo Shell 登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查 MongoDB CPU使用率高的问题](#)
- [如何查询及限制连接数](#)

4.6.2. 通过Mongo Shell连接云数据库MongoDB副本集实例

Mongo Shell是MongoDB自带的数据库管理工具，您可以将Mongo Shell工具下载并安装至客户端（本地服务器或ECS服务器），用于连接云数据库MongoDB副本集实例。本文介绍通过Mongo Shell连接云数据库MongoDB副本集实例的方法。

前提条件

已创建云数据库MongoDB副本集实例。

背景信息

云数据库MongoDB副本集实例提供的默认数据库admin，用于管理默认数据库账号root的信息，建议使用admin之外的数据库实现业务需求（即不对admin库进行任何操作）。

准备工作

- 下载并安装与云数据库MongoDB实例数据库版本相同的MongoDB至客户端（本地服务器或云服务器ECS实例），安装方法请参见[Install MongoDB](#)。
 - 如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS实例满足以下条件，以保证网络的连通性。ECS实例信息的查看方法请参见[查看实例信息](#)。
 - 所在地域相同，并建议所在可用区相同（以减少网络延时）。

- 网络类型相同。

说明

- 专有网络具有更高的安全性，建议使用专有网络。
- 如果网络类型为专有网络，需确保使用相同的专有网络ID。
- 如果您需要使用专有网络，但ECS实例的网络类型为经典网络，您可以将ECS实例网络类型转换为专有网络，具体请参见[ECS实例从经典网络迁移到专有网络](#)。

- 将客户端IP地址添加至云数据库MongoDB实例的白名单中，添加方法请参见[设置白名单](#)。

说明 如果您需要使用公网地址连接云数据库MongoDB实例，您可以手动申请公网地址，申请方法请参见 [（可选）申请公网连接地址](#)。

- 如果需要连接只读节点，请确保云数据库MongoDB实例中已存在只读节点，如果没有，您可以添加只读节点，添加方法请参见[变更副本集实例配置](#)。

操作步骤

- 登录[MongoDB管理控制台](#)。
- 在左侧导航栏，单击[副本集实例列表](#)。
- 在页面左上角，选择实例所在的资源组和地域。
- 单击目标实例ID或目标实例所在行操作列的管理。
- 在目标实例页面的左侧导航栏，单击[数据库连接](#)。
- 查看并获取云数据库MongoDB副本集实例的连接地址。

私网连接 - 经典网络

节点

地址

Primary

Secondary

Readonly

ReadOnly ConnectionStringURI

ConnectionStringURI

公网连接

节点

地址

Primary

Secondary

Readonly

ReadOnly ConnectionStringURI

ConnectionStringURI

项目	说明
----	----

项目	说明
地址类型	◦ 私网连接： ■ 专有网络：专有网络是一种隔离的网络环境，安全性和性能均高于传统的经典网络。云数据库MongoDB实例默认提供专有网络连接地址，通过专有网络连接云数据库MongoDB单节点实例可以获取更高的安全性和性能。 ■ 经典网络：仅依靠云数据库MongoDB实例的安全组或白名单策略阻挡非法访问。 ◦ 公网连接：由于通过公网连接实例存在一定的安全风险，云数据库MongoDB实例默认未提供公网连接地址。如果您要使用公网地址连接云数据库MongoDB实例，您可以手动申请公网连接地址，申请方法请参见 （可选）申请公网连接地址。
Primary	主节点，连接该节点可执行数据库的读写操作。当发生主备切换时，主节点角色将发生变化，您需要连接新的主节点确保读写操作不受影响。 格式： <pre><host>:<port></pre>  说明 ◦ <code><host></code>：云数据库MongoDB副本集实例主节点的地址。 ◦ <code><port></code>：云数据库MongoDB副本集实例主节点的端口号。 示例： <pre>dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717</pre>
Secondary	从节点，连接该节点仅能执行数据库的读操作。您也可以扩展从节点的数量，扩展方法请参见变更副本集实例配置。 格式： <pre><host>:<port></pre>  说明 ◦ <code><host></code>：云数据库MongoDB副本集实例从节点的地址。 ◦ <code><port></code>：云数据库MongoDB副本集实例从节点的端口号。 示例： <pre>dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717</pre>

项目	说明
ReadOnly	只读节点，连接该节点仅能执行数据库的读操作。您也可以扩展只读节点的数量，扩展方法请参见变更副本集实例配置。 格式： <host>:<port>  说明 <code><host></code>：云数据库MongoDB副本集实例只读节点的地址。 <code><port></code>：云数据库MongoDB副本集实例只读节点的端口号。 示例： dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717

项目	说明
ReadOnlyConnectionString URI	只读节点的高可用连接地址，通过该地址连接可以实现只读节点间的负载均衡及高可用。在拥有多个只读节点的情况下，建议应用程序通过此连接串地址连接只读节点。当实例中的某个只读节点出现故障时，系统无缝切换到下一个只读节点，不会影响应用的读操作。 格式： <pre>mongodb://<username>:<password>@<host1>:<port1>,<host2>:<port2>,...,<hostN>:<portN>/<database>?<option></pre>  说明 ◦ <code>mongodb://</code>：前缀，代表这是一个只读节点的ConnectionStringURI连接地址。 ◦ <code><username></code>：云数据库MongoDB副本集实例的账号名，默认为root。 ◦ <code><password></code>：云数据库MongoDB副本集实例账号名对应的密码。 ◦ <code><host></code>：云数据库MongoDB副本集实例只读节点的地址。 ◦ <code><port></code>：云数据库MongoDB副本集实例只读节点的端口号。 ◦ <code><database></code>：云数据库MongoDB副本集实例的鉴权数据库名，即数据库账号所属的数据库。默认为admin。如果您需要使用其他数据库，您可以先在该数据库中创建账号，然后使用该账户。创建账号的方法请参见<code>db.createUser()</code>。 ◦ <code><option></code>：其他信息。 示例： <pre>mongodb://root:****@dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717/admin?readPreference=secondary&readPreferenceTags=role:readonly&replicaSet=mgset-45816505</pre>

项目	说明
ConnectionStringURI	云数据库MongoDB副本集实例的高可用连接地址，通过该地址连接可以实现负载均衡及高可用。生产环境的应用程序建议通过此连接串地址连接实例。当实例中的某个节点出现故障时，不会因为主备切换而影响应用的读写操作。 格式： <pre>mongodb://<username>:<password>@<host1>:<port1>,<host2>:<port2>,...,<hostN>:<portN>/<database>?<option></pre> 说明 <code>mongodb://</code>：前缀，代表这是一个ConnectionStringURI连接地址。 <code><username></code>：云数据库MongoDB副本集实例的账号名，默认为root。 <code><password></code>：云数据库MongoDB副本集实例账号名对应的密码。 <code><host></code>：云数据库MongoDB副本集实例节点的地址。 <code><port></code>：云数据库MongoDB副本集实例节点的端口号。 <code><database></code>：云数据库MongoDB副本集实例的鉴权数据库名，即数据库账号所属的数据库。默认为admin。如果不指定数据库名，则默认为test。如果您需要使用其他数据库，您可以先在该数据库中创建账号，然后使用该账户。创建账号的方法请参见db.createUser()。 <code><option></code>：其他信息。 示例： <pre>mongodb://root:****@dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717,dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717,dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717/admin?replicaSet=mgset-45816505</pre>

7. 根据业务需要，执行如下对应命令，在安装有MongoDB的客户端（本地服务器或ECS服务器）上连接云数据库MongoDB副本集实例。

- （推荐）高可用连接方式
 - ConnectionStringURI：使用ConnectionStringURI连接数据库，可确保连接的节点始终为Primary节点，不会因为主备切换而影响应用的读写操作。
 - ReadOnly ConnectionStringURI：在拥有多个只读节点的情况下，可使用ReadOnly ConnectionStringURI连接只读节点。当实例中的某个只读节点出现故障时，系统会无缝切换到下一个只读节点，不会影响应用的读操作。

命令格式：

```
mongo "<连接串地址>"
```

② 说明

- 双引号须为英文双引号（"）。
- <连接串地址>：实例的ConnectionStringURI地址或ReadOnly ConnectionStringURI地址。
ConnectionStringURI或ReadOnly ConnectionStringURI地址中的 **** 需要替换为数据库密码。数据库密码设置请参见[设置密码](#)。

○ 连接一个节点（Primary、Secondary或ReadOnly节点）

```
mongo --host <host>:<port> -u <username> -p --authenticationDatabase <database>
```

② 说明

- <host>：云数据库MongoDB副本集实例Primary、Secondary或ReadOnly节点的地址。
- <port>：云数据库MongoDB副本集实例Primary、Secondary或ReadOnly节点的端口号。
- <username>：云数据库MongoDB副本集实例的账号名，初始账号为root。不建议直接使用root账号连接数据库。您可以根据业务需求创建用户并分配权限，详情请参见[MongoDB数据库账号权限管理](#)。
- <database>：鉴权数据库名，即数据库账号所属的数据库。当数据库账号为root时，对应的数据库为admin。如果不指定数据库名，则默认为test。如果您希望指定其他数据库，请先在该数据库中使用db.createUser()命令创建账号，然后再使用该账号进行连接。

示例：

```
mongo --host dds-bp1366caac83d****.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

在命令行提示 Enter password: 时，输入数据库账号对应的密码并按回车键确认。如果忘记了root账号的密码，您可以重新设置密码，设置方法请参见 [（可选）重置密码](#)。

- ② 说明 输入密码时，密码字符是不可见的。

- ② 说明 实例连接成功后，建议您使用admin之外的数据库实现业务需求。

常见的连接场景

● 如何通过公网连接MongoDB实例

- ② 说明 通过公网地址连接数据库时，建议使用SSL加密连接，详情请参见[使用Mongo Shell通过SSL加密连接数据库](#)。

- 不同网络类型的ECS实例与MongoDB实例如何连接
- ECS实例与MongoDB实例地域不同时如何连接

- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

相关问题

- [排查Mongo Shell 登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查MongoDB CPU使用率高的问题](#)
- [如何查询及限制连接数](#)

4.6.3. 通过程序代码连接MongoDB副本集实例

云数据库MongoDB版完全兼容MongoDB协议，本文为您介绍使用不同语言的程序代码来连接MongoDB副本集实例。

前提条件

- 获取云数据库MongoDB副本集实例连接地址。具体操作，请参见[副本集实例连接说明](#)。
- 根据您使用的语言下载并安装官方驱动程序。更多信息，请参见[MongoDB Drivers](#)。

Node.js连接示例

Node.js驱动相关信息，请参见[MongoDB Node.js Driver](#)。

1. 在客户端执行如下命令，进行项目初始化。

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

2. 执行如下命令，安装驱动包以及工具包。

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

3. 获取云数据库MongoDB副本集实例连接信息。
4. Node.js Demo Code。

```
const sprintf = require("sprintf-js").sprintf;
const MongoClient = require('mongodb').MongoClient;
const host1 = "dds-*****.mongodb.rds.aliyuncs.com";
const port1 = 3717;
const host2 = "dds-*****.mongodb.rds.aliyuncs.com";
const port2 = 3717;
const username = "root";
const password = "*****";
const replSetName = "mgset-*****";
const demoDb = "test";
const demoColl = "testColl";
//官方建议使用副本集地址，确保高可用。
const url = sprintf("mongodb://%s:%s@%s:%d,%s:%d/admin?replicaSet=%s",
    username, password, host1, port1, host2, port2, replSetName);
console.info("url:", url);
const client = new MongoClient(url);
//获取mongoClient。
async function run() {
    try {
        //连接实例。
        await client.connect();
        //取得数据库句柄。
        const database = client.db(demoDb);
        //取得Collection句柄。
        const collection = database.collection(demoColl);
        const demoName = "Node For Demo";
        const doc = { "DEMO": demoName, "MSG": "Hello AliCoudDB For MongoDB" };
        console.info("ready insert document: ", doc);
        //插入数据。
        const result = await collection.insertOne(doc);
        console.log(
            `A document was inserted with the _id: ${result.insertedId}`,
        );
        //读取数据。
        const filter = { "DEMO": demoName };
        const findResult = await collection.find(filter);
        await findResult.forEach(console.dir);
    } finally {
        //关闭连接。
        await client.close();
    }
}
run().catch(console.dir);
```

PHP连接示例

PHP驱动相关信息，请参见[MongoDB PHP Driver](#)。

1. 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"
```

2. 获取云数据库MongoDB副本集实例连接信息。

3. PHP Demo Code。

```
<?php
require 'vendor/autoload.php'; // include Composer goodies
# 实例信息
$demo_seed1 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_seed2 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_replname = "mgset-*****";
$demo_user = 'root';
$demo_password = '*****';
$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 = $collection->insertOne(['name' => 'ApsaraDB for MongoDB', 'desc' => 'Hello, M
ongodb']);
echo "Inserted with Object ID '{$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](#)文档。
- 下载[jar包](#)。

1. 获取云数据库MongoDB副本实例连接信息。

2. Java Demo Code。

i. Maven配置。

```
<dependencies>
    <dependency>
        <groupId>org.mongodb</groupId>
        <artifactId>mongo-java-driver</artifactId>
        <version>4.1.0</version>
    </dependency>
</dependencies>
```

ii. 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("*****.mongodb.t
bc3.newtest.rdstest.aliyun-inc.com",
        27017);

    public static ServerAddress seed2 = new ServerAddress("*****.mongodb.t
bc3.newtest.rdstest.aliyun-inc.com",
        27017);

    public static String username = "demouser";
    public static String password = "*****";
    public static String ReplSetName = "mgset-*****";
    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().requiredR
eplicaSetName(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 + "?replic
aSet=" + ReplSetName);
        return new MongoClient(connectionString);
    }

    public static void main(String args[]) {
        MongoClient client = createMongoDBClient();
        // or
        // MongoClient client = createMongoDBClientWithURI();
        try {

```



```
on(DEMO_COLL);

// 取得Collection句柄。
MongoDatabase database = client.getDatabase(DEMO_DB);
MongoCollection<Document> collection = database.getCollecti

// 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).iter

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 4.1.0](#)下载。
- [PyMongo 4.1.0](#)官方文档。

1. 安装pymongo。

```
pip install pymongo
```

2. 获取云数据库MongoDB副本集实例连接信息。

3. Python Demo Code。

```

import uuid
from pymongo import MongoClient
CONN_ADDR1 = '*****.mongodb.****.*****.rdstest.aliyun-inc.com:27017'
CONN_ADDR2 = '*****.mongodb.****.*****.rdstest.aliyun-inc.com:27017'
REPLICAT_SET = 'mgset-*****'
username = 'demouser'
password = '*****'
#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, MSG="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

```

C#连接示例

C#驱动相关信息，请参见[MongoDB C# Driver](#)。

1. 在客户端执行如下命令，安装如下驱动包。

```
mongocsharpdriver.dll
```

2. 获取云数据库MongoDB副本集实例连接信息。
3. C# Demo Code。

```

using MongoDB.Driver;
using System;
using System.Collections.Generic;
namespace Aliyun
{
    class Program
    {
        static void Main(string[] args)
        {
            //Mongo 实例信息。
            const string host1 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";
            const int port1 = 3717;
            const string host2 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";
            const int port2 = 3717;
            const string replicaSetName = "mgset-300*****";
            const string admin = "admin";
            const string userName = "root";
            const string passwd = "*****";
            try
            {
                Console.WriteLine("开始连接.....");
                MongoClientSettings settings = new MongoClientSettings()

```

```

MongoClientSettings settings = new MongoClientSettings();
List<MongoServerAddress> servers = new List<MongoServerAddress>();
servers.Add(new MongoServerAddress(host1, port1));
servers.Add(new MongoServerAddress(host2, port2));
settings.Servers = servers;
//设置副本集名称。
settings.ReplicaSetName = replicaSetName;
//设置超时时间为3秒。
settings.ConnectTimeout = new TimeSpan(0, 0, 0, 3, 0);
MongoCredential credentials = MongoCredential.CreateCredential(admin, u
serName, passwd);
settings.Credential = credentials;
MongoClient client = new MongoClient(settings);
var server = client.GetServer();
MongoDatabase database = server.GetDatabase("test");
var collection = database.GetCollection<User>("test_collection");
User user = new User();
user.id = "1";
user.name = "mongo_test";
user.sex = "女";
//插入数据user。
collection.Insert(user);
//获取一条数据。
User result = collection.FindOne();
Console.WriteLine("id:" + result.id + " name:" + result.name + " sex:" +
result.sex);
Console.WriteLine("连接成功.....");
}
catch (Exception e)
{
    Console.WriteLine("连接异常:" + e.Message);
}
}
}
class User
{
    public string id { set; get; }
    public string name { set; get; }
    public string sex { set; get; }
}
}

```

Go连接示例

Go驱动相关信息，请参见[MongoDB Go Driver](#)。

1. 安装如下驱动包。

```
go get go.mongodb.org/mongo-driver
```

2. 获取云数据库MongoDB副本集实例连接信息。
3. Go Demo Code。

```

package main
import (
    "context"

```

```

context := context.TODO()
    "fmt"
    "go.mongodb.org/mongo-driver/bson"
    "go.mongodb.org/mongo-driver/mongo"
    "go.mongodb.org/mongo-driver/mongo/options"
    "go.mongodb.org/mongo-driver/mongo/readpref"
    "go.mongodb.org/mongo-driver/mongo/writeconcern"
    "log"
)

func main() {
    // Create a Client to a MongoDB server and use Ping to verify that the
    // server is running.
    clientOpts := options.Client().ApplyURI("mongodb://root:****@dds-bp*****1.mongodb.rds.aliyuncs.com:3717,dds-bp*****2.mongodb.rds.aliyuncs.com:3717/admin?replicaSet=mgset-XXXXX")
    clientOpts.SetReadPreference(readpref.Secondary())
    clientOpts.SetWriteConcern(writeconcern.New(writeconcern.WMajority(), writeconcern.J(true), writeconcern.WTimeout(1000)))
    client, err := mongo.Connect(context.TODO(), clientOpts)
    if err != nil {
        fmt.Println("connect failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("connect successful!")
    defer func() {
        if err = client.Disconnect(context.TODO()); err != nil {
            fmt.Println("disconnect failed!")
            log.Fatal(err)
        }
        fmt.Println("disconnect successful!")
    }()
    // Call Ping to verify that the deployment is up and the Client was
    // configured successfully. As mentioned in the Ping documentation, this
    // reduces application resiliency as the server may be temporarily
    // unavailable when Ping is called.
    if err = client.Ping(context.TODO(), nil); err != nil {
        fmt.Println("ping failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("ping successful!")
    collection := client.Database("baz").Collection("qux")
    res, err := collection.InsertOne(context.Background(), bson.M{"hello": "world"})
    if err != nil {
        fmt.Println("insert result failed!")
        log.Fatal(err)
        return
    }
    id := res.InsertedID
    fmt.Println("Id: ", id)
    fmt.Printf("insert result: %v\n", res)
    result := bson.M{}
    filter := bson.D{{"_id", res.InsertedID}}
    if err := collection.FindOne(context.Background(), filter).Decode(&result); err !=

```

```
    // 如果找不到集合，则返回 nil
    if err := collection.FindOne(context.Background(), bson.M{"_id": "test"}, nil); err != nil {
        if err == mongo.ErrNoDocuments {
            // 没有找到文档
            return nil
        }
        // 其他错误
        log.Fatal(err)
        return
    }
    // 打印结果
    fmt.Printf("result: %v\n", result)
}
```

5.分片集群快速入门

5.1. 使用限制

在使用云数据库MongoDB分片集群实例前，您需要了解分片集群实例本身存在的约束信息，使您初步了解分片集群实例的使用限制，帮助您进一步确认分片集群实例是否能够满足您的业务场景。

功能	DynamoDB协议实例约束	MongoDB协议实例约束
实例部署	- 云数据库MongoDB不支持在以下地域的可用区创建3.4、4.0和4.2版本的实例。 - 华东5（南京-本地地域）：南京-本地地域可用区A。 - 中国（香港）：香港可用区C。 - 泰国（曼谷）：曼谷可用区A。 - 韩国（首尔）：首尔可用区A。 - 云数据库MongoDB不支持在以下地域的可用区创建4.4和5.0版本的实例。 - 华东1（杭州）：杭州可用区J和K。 - 华东2（上海）：上海可用区L和D。 - 华北1（青岛）：青岛可用区B。 - 华北2（北京）：北京可用区E、L和K。 - 华北3（张家口）：张家口可用区B。 - 华北5（呼和浩特）：呼和浩特可用区A。 - 华北6（乌兰察布）：乌兰察布可用区A、B和C。 - 华南1（深圳）：深圳可用区A、D、C、（C+D+E）和F。 - 华南3（广州）：广州可用区A。 - 中国（香港）：香港可用区（B+C+D）。 - 澳大利亚（悉尼）：悉尼可用区A和B。 - 马来西亚（吉隆坡）：吉隆坡可用区A。 - 菲律宾（马尼拉）：马尼拉可用区A。 - 日本（东京）：东京可用区A和B。 - 印度（孟买）：孟买可用区A。 - 阿联酋（迪拜）：迪拜可用区A。 - 美国（硅谷）：硅谷可用区A。 - 英国（伦敦）：伦敦可用区A。 - 德国（法兰克福）：法兰克福可用区C。	
数据库版本	选择分片集群实例的版本与存储引擎存在一定的约束性，具体请参见 版本及存储引擎 。	

功能	DynamoDB协议实例约束	MongoDB协议实例约束
搭建分片集群	- 在创建分片集群实例时，您可以选择Mongos和Shard节点的配置和节点数量。 - 实例运行过程中，您可以变更实例的规格，也可以添加或释放Mongos和Shard节点，具体请参见变更分片集群实例配置概览。  说明 - 当分片集群实例中Mongos节点数量为3个及以上时，支持释放。但须至少保留2个Mongos节点。 - 当分片集群实例中Shard节点数量为3个及以上时，支持释放。但须至少保留2个Shard节点。 - 释放Shard节点前，您需要确保剩余Shard节点完全可以分担释放节点的数据，否则，执行释放操作后，实例状态将会一直为删除节点中，无法对实例进行重置密码、角色切换、变更节点连接地址、修改参数等操作。	
数据库小版本	-	如果您的数据库小版本过期或者不在维护列表内，当执行实例版本升级、数据迁移、变更实例配置或数据恢复等操作时，为保证提供更出色的性能和稳定性，系统会默认将您的数据库小版本升级至最新版。
数据读写	分片集群实例中的admin数据库仅支持用户读取，不支持写入。	
数据备份	仅支持物理备份和逻辑备份。	
数据恢复	仅支持按时间点创建实例的方式恢复数据，具体请参见 按时间点将备份数据恢复至新建实例 。	
数据安全性	- 不支持设置SSL加密。 - 仅支持专有网络，专有网络的创建方法请参见创建和管理专有网络。 - 支持设置白名单，设置方法请参见设置白名单。 - 支持设置透明加密数据TDE，设置方法请参见设置透明数据加密TDE。 - 支持审计日志。	- 不支持设置SSL加密。 - 支持设置白名单，设置方法请参见设置白名单。 - 支持设置透明加密数据TDE，设置方法请参见设置透明数据加密TDE。 - 支持审计日志。
CloudDBA（性能诊断与优化）	不支持CloudDBA功能，包括性能趋势、实时性能、实例会话、空间分析和慢日志。	-

功能	DynamoDB协议实例约束	MongoDB协议实例约束
数据迁移与同步	- 云数据库MongoDB支持使用以下方法迁移分片集群实例数据。 - 自建数据库迁移至云数据库MongoDB - 将ECS上分片集群架构的自建MongoDB迁移上云 - 使用DTS迁移分片集群架构的自建MongoDB数据库上云 - 使用MongoDB工具将自建数据库迁移至分片集群实例 - 第三方云数据库迁移至云数据库MongoDB - 使用NimoShake将Amazon DynamoDB迁移至阿里云 - 使用MongoDB工具将MongoDB Atlas迁移至阿里云 - 使用DTS将MongoDB Atlas数据库迁移至阿里云 - 云数据库MongoDB实例间迁移 - 迁移MongoDB实例至其他地域 - 云数据库MongoDB支持使用以下方法同步分片集群实例数据。 - 云数据库MongoDB版（分片集群架构）间的双向同步	
连接实例	云数据库MongoDB支持使用以下方法连接分片集群实例。 - 通过DMS连接MongoDB分片集群实例 - 通过Mongo Shell连接MongoDB分片集群实例 - 通过程序代码连接DynamoDB协议兼容版实例 - 通过程序代码连接MongoDB分片集群实例	
回收站	不支持回收站功能。	

5.2. 创建分片集群实例

云数据库MongoDB提供的分片集群实例适用于高并发读写的场景。本文介绍云数据库MongoDB分片集群实例的创建方法。

前提条件

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

注意事项

如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS实例满足以下条件，以保证网络的连通性。ECS实例信息的查看方法请参见[查看实例信息](#)。

- 所在地域相同，并建议所在可用区相同（以减少网络延时）。
- 网络类型相同。

说明

- 专有网络具有更高的安全性，建议使用专有网络。
- 如果网络类型为专有网络，需确保使用相同的专有网络ID。
- 如果您需要使用专有网络，但ECS实例的网络类型为经典网络，您可以将ECS实例网络类型转换为专有网络，具体请参见[ECS实例从经典网络迁移到专有网络](#)。

使用限制

- 仅支持在以下地域的可用区创建云数据库MongoDB3.4、4.0和4.2版本的实例。
 - 华东1（杭州）：杭州可用区G、H、I、J和K。
 - 华东2（上海）：上海可用区L、B、D和G。
 - 华北1（青岛）：青岛可用区B和C。
 - 华北2（北京）：北京可用区F、E、H、L和K。
 - 华北3（张家口）：张家口可用区A、B和C。
 - 华北5（呼和浩特）：呼和浩特可用区A和B。
 - 华北6（乌兰察布）：乌兰察布可用区A、B和C。
 - 华南1（深圳）：深圳可用区A、D、C、E、（C+D+E）和F。
 - 华南2（河源）：河源可用区A和B。
 - 华南3（广州）：广州可用区A。
 - 西南1（成都）：成都可用区A和B。
 - 中国（香港）：香港可用区B、D和（B+C+D）。
 - 新加坡：新加坡可用区A、B和C。
 - 澳大利亚（悉尼）：悉尼可用区A和B。
 - 马来西亚（吉隆坡）：吉隆坡可用区A和B。
 - 印尼（雅加达）：雅加达可用区A、B和C。
 - 菲律宾（马尼拉）：马尼拉可用区A。
 - 日本（东京）：东京可用区A和B。
 - 印度（孟买）：孟买可用区A。
 - 阿联酋（迪拜）：迪拜可用区A。
 - 美国（硅谷）：硅谷可用区A和B。
 - 美国（弗吉尼亚）：弗吉尼亚可用区A和B。
 - 英国（伦敦）：伦敦可用区A。
 - 德国（法兰克福）：法兰克福可用区A、B和C。
- 仅支持在以下地域的可用区创建云数据库MongoDB4.4和5.0版本的实例。
 - 华东1（杭州）：杭州可用区G、H和I。
 - 华东2（上海）：上海可用区B和G。
 - 华东5（南京-本地地域）：南京-本地地域可用区A。
 - 华北1（青岛）：青岛可用区C。
 - 华北2（北京）：北京可用区F和H。
 - 华北3（张家口）：张家口可用区A和C。

- 华北5（呼和浩特）：呼和浩特可用区B。
- 华南1（深圳）：深圳可用区E。
- 华南2（河源）：河源可用区A和B。
- 西南1（成都）：成都可用区A和B。
- 中国（香港）：香港可用区B、C和D。
- 新加坡：新加坡可用区A、B和C。
- 马来西亚（吉隆坡）：吉隆坡可用区B。
- 印尼（雅加达）：雅加达可用区A、B和C。
- 泰国（曼谷）：曼谷可用区A。
- 韩国（首尔）：首尔可用区A。
- 美国（硅谷）：硅谷可用区B。
- 美国（弗吉尼亚）：弗吉尼亚可用区A和B。
- 德国（法兰克福）：法兰克福可用区A和B。

费用说明

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

- 包年包月：预付费，在创建实例时就需要支付费用。
- 按量付费：后付费，每小时根据实例配置生成1个收费订单，并扣除相应的账户余额。

 **说明** 关于收费信息，请参见[收费项目及价格说明](#)。

如果您是首次购买云数据库MongoDB实例，您可以参与数据库首购折扣价活动，活动信息请参见[云数据库优惠活动](#)。

操作步骤

在执行完下列步骤之后，云数据库MongoDB会自动为您配置好数据库，您无需手动进行安装部署。

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击分片集群实例列表。
3. 在分片集群实例列表页面，单击创建实例。
4. 在商品类型区域，请根据业务需求，选择分片集群（包年包月）或分片集群（按量付费）。
5. 设置以下参数，完成分片集群实例的参数配置。

参数	参数说明
地域	选择实例所属地域。  说明 ◦ 实例创建成功后将无法更换地域，请您谨慎选择。 ◦ 如果应用部署在ECS实例，请您选择与ECS实例相同的地域，否则无法直接通过内网连接。

参数	参数说明
可用区	选择实例所属可用区。  说明 - 如果应用部署在ECS实例，建议您选择与ECS实例相同的可用区，以减少网络延时。如果选择不相同，您可以迁移可用区，迁移方法请参见迁移可用区。 - 您可以选择多可用区实现实例的同城容灾，详情请参见创建多可用区分片集群实例。
协议类型	选择实例使用的协议。 - MongoDB协议 - DynamoDB协议  说明 关于DynamoDB协议的更多信息，请参见DynamoDB协议兼容版实例支持详情。
数据库版本	请根据业务需求，选择对应数据库版本。 - MongoDB 5.0 - MongoDB 4.4 - MongoDB 4.2 - MongoDB 4.0 - MongoDB 3.4  说明 - DynamoDB协议的分片集群实例仅支持MongoDB 4.0。 - 您可以在实例运行期间手动升级数据库版本，升级方法请参见升级数据库版本。
存储引擎	固定为WiredTiger。
存储类型	根据业务需要，选择分片集群实例的存储类型。 - ESSD云盘：MongoDB 5.0和MongoDB 4.4版本固定为该类型。 - SSD本地盘：MongoDB 4.2、MongoDB 4.0和MongoDB 3.4版本固定为该类型。
资源组	选择管理实例的资源组。您可以选择加入已经创建的资源组，也可以创建一个资源组加入，创建方法请参见创建资源组。  说明 通过设置资源组，您可以对单个云账号下多个地域、多种云资源进行集中的分组管理。

参数	参数说明
网络类型	固定为专有网络。  说明 如果应用部署在ECS实例，您需要确保ECS实例的网络类型为专有网络，以保证网络的连通性。
专有网络	选择实例所属专有网络。如果您没有可以使用的专有网络，您可以在专有网络控制台创建专有网络，创建方法请参见创建和管理专有网络。  说明 - 如果需要更改网络类型，具体请参见切换实例网络类型。 - 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。具体方案请参见经典网络平滑迁移到VPC的混访方案。
虚拟交换机	选择实例所属交换机。如果您没有可以使用的交换机，您可以在专有网络控制台创建交换机，创建方法请参见创建和管理交换机。
Mongos规格	选择实例中Mongos节点的规格，关于Mongos节点支持的规格信息，请参见实例规格表实例规格概述。  说明 实例运行期间，您可以变更Mongos节点的配置和数量。
Mongos数量	选择实例中Mongos节点的数量，取值范围为2~32。  说明 - 每个Mongos节点默认为单节点架构，请选择2个及以上的Mongos节点，以保障实例的高可用性。 - 单个分片集群实例默认最多支持32个Mongos节点，如果业务需要32个以上的Mongos节点，您可以提交工单申请，提交工单的方法请参见提交工单。
Shard规格	选择实例中Shard节点的规格，关于Shard节点支持的规格信息，请参见实例规格表实例规格概述。  说明 实例运行期间，您可以变更Shard节点的配置和数量。

参数	参数说明
Shard存储空间	选择实例存储空间的大小。  说明 - 该存储空间的大小为每个独立节点的存储空间大小，不是所有节点存储空间大小的总和。 - Shard节点的存储空间包含您的数据文件、系统文件、日志文件空间。
Shard只读节点数	选择分片集群实例的只读节点数量，取值范围为0~5。更多只读节点的信息，请参见MongoDB只读节点简介。
Shard数量	选择实例中Shard节点的数量，取值范围为2~32。  说明 - 每个Shard节点默认为三节点副本集架构，请选择2个及以上的Shard节点，确保能够根据业务需求合理设置数据分片，以充分利用Shard节点的存储空间和计算性能，具体请参见设置数据分片以充分利用Shard性能。 - 单个分片集群实例默认最多支持32个Shard节点，如果业务需要32个以上的Shard节点，您可以提交工单申请，提交工单的方法请参见提交工单。
Configserver规格	固定为1核 2 GB。
ConfigServer存储空间	固定为20 GB。
密码设置	设置root账号的密码。您可以选择以下任意一种方法： 立即设置：立即设置密码。 创建后设置：在实例运行期间设置密码，设置方法请参见（可选）重置密码。
密码	如果您选择立即设置密码，请按照如下规则设置密码： - 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成。 !@#\$%^&*()_+ -= 为特殊字符。 - 密码长度为8~32位。

参数	参数说明
购买时长	如果您创建的是包年包月实例，请您选择购买时长。包月可选择1~9个月，包年可选择1~5年。  说明 如果您希望实例到期后自动续费，勾选到期自动续费，并确保实例在即将到期前您的账户中有足够的余额。

6. 根据实例的付费类型，选择对应操作购买实例。

付费类型	操作步骤
包年包月	i. 配置完成后，单击立即购买。 ii. 在确认订单页面，查看配置详情，阅读并勾选服务协议。 iii. 单击去支付，并在支付页面，根据提示完成支付流程。  说明 您也可以单击加入购物车，与其他产品一起进行结算。结算方法请参见购物车结算。
按量付费	i. 配置完成后，单击立即购买。 ii. 在确认订单页面，查看配置详情，阅读并勾选服务协议。 iii. 单击立即开通，系统将在下一小时自动扣取对应金额。

7. 查看分片集群实例是否创建成功。

- i. 支付完成后，单击**管理控制台**，进入云数据库MongoDB控制台。
- ii. 在页面左上角，选择实例所在的资源组和地域。
- iii. 在左侧导航栏，选择**分片集群实例列表**。

- iv. 查看创建的实例是否在分片集群实例列表中。
- 是，说明实例已创建，当实例状态为运行中时，说明实例创建成功。
 - 否，请您等待10~15分钟后刷新页面重新查看，如果分片集群列表中仍然没有，请先排查是否是以下原因导致，如果不是，请联系阿里云技术支持。

导致实例列表找不到实例的可能原因及解决办法如下：

可能原因	解决方法
地域选择错误	在云数据库MongoDB控制台左上角，选择实例所在的资源组和地域。
实例列表选择错误	在云数据库MongoDB控制台左侧导航栏，选择分片集群实例列表。
资源不足	新创建的实例可能因为资源不足而退款。请您在订单管理中查看退款，具体请参见订单管理。 在您确认退款后，您可以选择以下任意一种方法处理： ■ 更换可用区尝试再次购买。 ■ 提交工单处理，具体请参见提交工单。

购物车结算

如果您购买的是包年包月实例，在实例配置完成后，您也可以单击**加入购物车**，与其他产品一起进行结算。结算方法如下：

1. 单击页面右侧的**购物车**。
2. 在**购物车**面板，勾选您需要购买的产品，并单击**去购物车结算**。
3. 在**购物车**页面，确认您需要购买的产品选择是否正确。

 **说明** 您也可以重新选择云数据库MongoDB实例的**购买时长**和**购买数量**。

- 是：单击**立即购买**。
 - 否：勾选正确的购买产品，单击**立即购买**。
4. 在**确认订单**页面，阅读并勾选服务协议，并单击**确认购买**。
 5. 在**支付**页面，根据提示完成支付流程。

相关API

接口	说明
CreateShardingDBInstance	• 创建云数据库MongoDB分片集群实例。 • 克隆云数据库MongoDB分片集群实例。
DescribeInstanceAutoRenewalAttribute	查询云数据库MongoDB实例是否为自动付费。
DescribeDBInstanceAttribute	查询云数据库MongoDB实例的详情。

后续步骤

实例创建完成后，您还需执行以下操作：

- （可选）如果您在创建实例时没有立即设置密码，请您设置密码，设置方法请参见（可选）重置密码。
- 设置实例的白名单，以允许外部设备访问该实例，设置方法请参见设置白名单。
- （可选）如果您需要通过公网连接实例，请您先申请公网连接地址，申请方法请参见（可选）申请公网连接地址。

5.3. （可选）重置密码

如果您忘记账号密码、修改旧密码，或创建实例时未设置密码，您可以通过云数据库MongoDB提供的重置密码功能重新设置密码。

操作步骤

1. 登录MongoDB管理控制台。
2. 在左侧导航栏，单击分片集群实例列表。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的，并选择管理。
5. 在目标实例页面的左侧导航栏，单击账号管理。
6. 单击操作列的重置密码。
7. 在重置密码面板，设置以下参数。

参数	参数说明
账号	固定为root。
新密码	按照以下规则设置账号名对应的新密码。 ◦ 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !#\$%^&*()_+= ◦ 密码长度为8~32位。
确认新密码	再次输入新密码。

8. 单击确定。

5.4. 设置白名单

为保障MongoDB数据库的安全稳定，系统默认任何设备均无法访问云数据库MongoDB实例（默认IP地址为127.0.0.1），所以在使用云数据库MongoDB实例前，您需要为实例设置白名单，以允许外部设备访问该实例。正确使用白名单可以使云数据库MongoDB实例得到更高级别的安全保护，建议您定期维护白名单。

操作步骤

1. 登录MongoDB管理控制台。
2. 在左侧导航栏，单击分片集群实例列表。
3. 在页面左上角，选择实例所在的资源组和地域。

4. 单击目标实例ID或目标实例所在行操作列的，并选择**管理**。
5. 在目标实例页面的左侧导航栏，单击**数据安全性 > 白名单设置**。
6. 在**添加白名单分组**区域，选择以下任意一种方法为实例设置白名单。
 - 手动添加

 说明

- 支持如下格式的IP地址或IP地址段。
 - 单个IP地址，例如10.23.12.24。
 - 0.0.0.0/0
 - **CIDR模式**，即无类域间路由。例如10.23.12.24/24，其中/24表示了地址中前缀的长度，范围[1, 32]。
- 多个IP地址或IP地址段之间以英文逗号（,）分隔。
- 设置为0.0.0.0/0或空表示对访问的IP地址来源不作任何限制，此时云数据库MongoDB数据库将面临高安全风险，请谨慎使用。

- a. 单击目标分组操作列的，并选择**手动修改**。
 - b. 在**手动修改**面板的**允许访问IP名单**输入框中输入IP地址或IP地址段。
 - c. 单击**确定**。
- 加载ECS内网IP添加
 - a. 单击目标分组操作列的，并选择**加载ECS内网IP添加**。
 - b. 在**加载ECS内网IP添加**面板的**允许访问IP名单**列表中，选择需要添加的ECS内网IP地址。
 - c. 单击。
 - d. 单击**确定**。

相关任务

- **添加白名单分组**
- **添加安全组**

5.5. （可选）申请公网连接地址

云数据库MongoDB支持公网连接地址，您可以为实例申请公网连接地址用于使用公网连接云数据库MongoDB实例的场景。本文介绍为云数据库MongoDB分片集群实例申请公网连接地址的方法。

背景信息

云数据库MongoDB实例支持的连接地址类型及对应说明如下。

连接地址类型	说明
--------	----

连接地址类型	说明
专有网络连接地址	• 专有网络是一种隔离的网络环境，安全性和性能均高于传统的经典网络。 • 云数据库MongoDB实例默认提供专有网络连接地址，通过专有网络连接云数据库MongoDB实例可以获取更高的安全性和连接性能。
经典网络连接地址	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。如果需要切换为更安全的专有网络，请参见经典网络切换为专有网络。  说明 DynamoDB协议的分片集群实例不支持经典网络。
公网连接地址	• 由于通过公网连接实例存在一定的安全风险，云数据库MongoDB实例默认提供专有网络连接地址。 • 如果您要使用阿里云以外的设备（例如本地设备）连接云数据库MongoDB实例，您可以手动申请公网连接地址。

注意事项

- 申请公网连接地址过程中，可能会出现实例重新启动的情况，建议您在业务低峰期进行操作。
- 如果需要通过申请的公网地址来连接实例，您需要将连接实例的终端所在的公网IP地址加入到实例的白名单中，白名单设置方法请参见[设置白名单](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击[分片集群实例列表](#)。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的，并选择[管理](#)。
5. 在目标实例页面的左侧导航栏，单击[数据库连接](#)。
6. 单击[公网连接](#)区域框右侧的[申请公网地址](#)。
7. 在[申请公网地址](#)面板，设置以下参数。

参数	选项	说明
	Shard	Shard节点。 如果需要为Shard节点申请公网连接地址，您需要先申请Shard节点的连接地址，申请方法请参见申请Shard或ConfigServer节点连接地址。  说明 在某些特定场景下（例如集群间数据同步），需要通过外网读取Shard节点的Oplog信息，您可以申请相应节点的公网连接地址，以满足业务需求。

参数 节点类型	选项	说明
	CS	ConfigServer节点。 如果需要为ConfigServer节点申请公网连接地址，您需要先申请ConfigServer节点的连接地址，申请方法请参见 申请Shard或ConfigServer节点连接地址 。 ❓ 说明 在某些特定场景下（例如集群间数据同步），需要通过外网读取ConfigServer节点的配置信息，您可以申请相应节点的公网连接地址，以满足业务需求。
	Mongos	Mongos节点。 ❓ 说明 一般业务读写数据只需连接至Mongos节点即可。
节点ID	当前实例节点的ID。	选择需要申请公网地址的节点ID。

8. 单击**确定**。
9. （可选）如果您需要为分片集群实例中的多个节点申请公网连接地址，您可以重复执行上述步骤完成申请。

❓ 说明 连续申请公网连接地址时，需要等待实例状态变为运行中。

执行结果

申请操作完成后，您可以在以下位置查看公网连接地址。关于连接地址的信息，请参见[分片集群实例连接说明](#)。

- 基本信息页面的连接信息区域。
- 数据库连接页面的公网连接区域。

相关文档

- 通过公网地址连接实例的方法请参见[如何通过公网连接MongoDB实例](#)。
- 为保障数据安全性，如您不再需要公网连接地址，您可以将公网地址释放，释放方法请参见[释放公网连接地址](#)。
- 通过公网地址连接数据库时，建议使用SSL加密，加密方法请参见[使用Mongo Shell通过SSL加密连接数据库](#)。

5.6. 连接实例

5.6.1. 通过DMS连接MongoDB分片集群实例

数据管理服务DMS（Data Management Service）是一种集数据管理、结构管理、用户授权、安全审计、数据趋势、数据追踪、BI图表、性能优化和服务器管理于一体的可视化、图形化数据管理服务。通过DMS获取MongoDB数据库列表后，您可以更加方便快捷地远程访问、在线管理您的MongoDB数据库。

准备工作

根据MongoDB实例的网络类型，将DMS服务器的IP地址加入至MongoDB实例的白名单中，详情请参见[设置白名单](#)。

?

说明

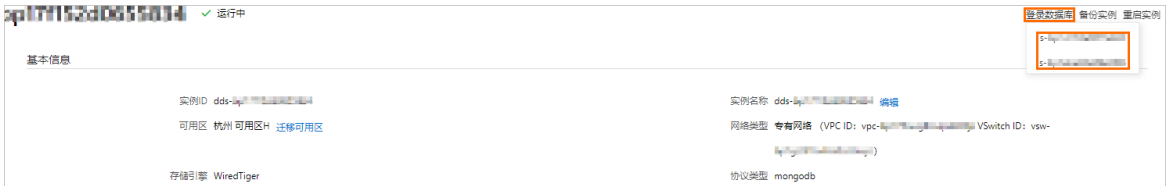
如您已经将DMS服务器的IP地址加入至MongoDB实例的白名单中，可跳过此步骤。

DMS服务器的IP地址

MongoDB实例的网络类型	DMS服务器的IP地址
专有网络	100.104.0.0/16
经典网络	120.55.177.0/24 121.43.18.0/24 101.37.74.0/24 10.153.176.0/24 10.137.42.0/24 11.193.54.0/24

操作步骤

- 1. 登录[MongoDB管理控制台](#)。
- 2. 在左侧导航栏，单击分片集群实例列表。
- 3. 在页面左上角，选择实例所在的资源组和地域。
- 4. 单击目标实例ID或目标实例所在行操作列的，并选择管理。
- 5. 在目标实例页面右上方，单击登录数据库。并选择任意Mongos节点ID，跳转到数据管理DMS页面。



- 6. 在登录实例对话框中，填写相应信息。

登录实例

* 数据库类型

MongoDB

* 实例地区

华东1（杭州）

连接串地址

s-bp-...mongodb.rds.aliyuncs.com:3717

数据库名称

root

* 数据库账号

请输入数据库账户

* 数据库密码

.....

☐ 记住密码

测试连接

登录

取消

参数	说明
数据库类型	目标数据库的类型，系统默认为当前需要连接的数据库类型。
实例地区	目标实例所在的地域，系统默认为当前实例所在地域。
连接串地址	目标实例的连接串地址，系统默认为当前实例的连接串地址。
数据库名称	填入鉴权数据库名，即数据库账号所属的数据库。 说明 如果数据库账号填写的是root，那么对应的数据库名即为admin。 不建议在生产环境中直接使用root账号连接数据库。您可以根据业务需求创建用户并分配权限，详情请参见使用DMS管理MongoDB实例的账号。
数据库账号	目标数据库的账号，初始账号为 root 。
数据库密码	填入该数据库账号对应的密码。 说明 如果忘记root账号的密码，您可以重置密码。

7. 单击左下角的**测试连接**。

 **说明** 当提示“测试成功”时，说明信息配置正确。如果测试失败，请根据报错信息排查并修改，直至测试成功。

8. 单击登录。

 **说明** 如果您希望浏览器记住该账号的密码，可以先勾选记住密码，再单击登录。

常见的连接场景

- [如何通过公网连接MongoDB实例](#)
- [不同网络类型的ECS实例与MongoDB实例如何连接](#)
- [ECS实例与MongoDB实例地域不同时如何连接](#)
- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

相关问题

- [排查 Mongo Shell 登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查 MongoDB CPU使用率高的问题](#)
- [如何查询及限制连接数](#)

5.6.2. 通过Mongo Shell连接MongoDB分片集群实例

Mongo Shell是MongoDB数据库自带的数据库管理工具，您可以在本地或ECS上安装Mongo Shell工具，然后通过Mongo Shell连接MongoDB分片集群实例。

前提条件

- 为保障鉴权成功，请安装与MongoDB实例版本相对应的Mongo Shell版本。安装步骤请参见官方文档[Install MongoDB](#)（请根据您的客户端版本在页面左上角选择版本号）。
- 已将客户端的IP地址加入到MongoDB实例的白名单中，详情请参见[设置白名单](#)。

 **说明** 如需通过公网连接MongoDB实例，需要[申请公网连接地址](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击[分片集群实例列表](#)。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的，并选择[管理](#)。
5. 在目标实例页面的左侧导航栏，单击[数据库连接](#)。获取Mongos节点的连接地址。

私网连接 - 专有网络 ②					开启内网免密访问	切换为经典网络	更多操作
ID	节点类型	节点	地址	操作			
	Mongos	Primary	s-l****.mongodb.rds.aliyuncs.com:3717	释放			
	Mongos	Primary	s-l****.mongodb.rds.aliyuncs.com:3717	释放			
ConnectionStringURI	Mongos	-	mongodb://****:****@****.mongodb.rds.aliyuncs.com:****	释放			
公网连接					申请公网地址	修改连接地址	
ID	节点类型	节点	地址	操作			
	Mongos	Primary	s-bp****.pub.mongodb.rds.aliyuncs.com:3717	释放			
ConnectionStringURI	Mongos	-	mongodb://****:****@****.pub.mongodb.rds.aliyuncs.com:****				

6. 在安装有Mongo Shell的本地服务器或ECS中连接实例。

```
mongo --host <mongos_host> -u <username> -p --authenticationDatabase <database>
```

② 说明

- <mongos_host>: 任一Mongos节点连接地址中的连接地址。
- <username>: MongoDB实例的数据库账号，初始账号为root。不建议在生产环境中直接使用root账号连接数据库。您可以根据业务需求创建用户并分配权限，详情请参见[MongoDB数据库账号权限管理](#)。
- <database>: 鉴权数据库名，即数据库账号所属的数据库。当数据库账号为root时，对应的数据库为admin。如果您希望指定其他数据库，请先在该数据库中使用db.createUser()命令创建账号，然后再使用该账号进行连接。

示例：

```
mongo --host s-bp****.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

7. 在命令行提示 Enter password: 时，输入数据库账号对应的密码并按回车键确认。如果忘记了root账号的密码，您可以重新设置密码，设置方法请参见[重置密码](#)。

② 说明

输入密码时，密码字符是不可见的。

常见的连接场景

- [如何通过公网连接MongoDB实例](#)
- [不同网络类型的ECS实例与MongoDB实例如何连接](#)
- [ECS实例与MongoDB实例地域不同时如何连接](#)
- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

相关问题

- [排查 Mongo Shell 登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查 MongoDB CPU使用率高的问题](#)

- [如何查询及限制连接数](#)

5.6.3. 通过程序代码连接MongoDB分片集群实例

云数据库MongoDB版完全兼容MongoDB协议，本文为您介绍使用不同语言的程序代码来连接分片集群实例。

前提条件

- 获取云数据库MongoDB副本集实例连接地址。具体操作，请参见[分片集群实例连接说明](#)。
- 根据您使用的语言下载并安装官方驱动程序。更多信息，请参见[MongoDB Drivers](#)。

 **说明** 连接分片集群实例时，无需指定下列Demo中的replicaSet相关参数。

Node.js连接示例

Node.js驱动相关信息，请参见[MongoDB Node.js Driver](#)。

1. 在客户端执行如下命令，进行项目初始化。

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

2. 执行如下命令，安装驱动包以及工具包。

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

3. 获取云数据库MongoDB单节点实例连接信息。
4. Node.js Demo Code。


```
const sprintf = require("sprintf-js").sprintf;
const MongoClient = require('mongodb').MongoClient;
const host1 = "dds-*****.mongodb.rds.aliyuncs.com";
const port1 = 3717;
const host2 = "dds-*****.mongodb.rds.aliyuncs.com";
const port2 = 3717;
const username = "root";
const password = "*****";
const replSetName = "mgset-*****";
const demoDb = "test";
const demoColl = "testColl";
// 官方建议使用副本集地址，确保高可用。
const url = sprintf("mongodb://%s:%s@%s:%d,%s:%d/admin?replicaSet=%s",
    username, password, host1, port1, host2, port2, replSetName);
console.info("url:", url);
const client = new MongoClient(url);
// 获取MongoClient。
async function run() {
    try {
        // 连接实例。
        await client.connect();
        // 取得数据库句柄。
        const database = client.db(demoDb);
        // 取得Collection句柄。
        const collection = database.collection(demoColl);
        const demoName = "Node For Demo";
        const doc = { "DEMO": demoName, "MSG": "Hello AliCoudDB For MongoDB" };
        console.info("ready insert document: ", doc);
        // 插入数据。
        const result = await collection.insertOne(doc);
        console.log(
            `A document was inserted with the _id: ${result.insertedId}`,
        );
        // 读取数据。
        const filter = { "DEMO": demoName };
        const findResult = await collection.find(filter);
        await findResult.forEach(console.dir);
    } finally {
        // 关闭连接。
        await client.close();
    }
}
run().catch(console.dir);
```

PHP连接示例

PHP相关信息，请参见[MongoDB PHP Driver](#)。

1. 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"
```

2. 获取云数据库MongoDB单节点实例连接信息。

3. PHP Demo Code。

```
<?php
require 'vendor/autoload.php'; // include Composer goodies
# 实例信息。
$demo_seed1 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_seed2 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_replname = "mgset-*****";
$demo_user = 'root';
$demo_password = '*****';
$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 = $collection->insertOne(['name' => 'ApsaraDB for MongoDB', 'desc' => 'Hello, M
ongodb']);
echo "Inserted with Object ID '{$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](#)文档。
- 下载[Jar包](#)。

1. 获取云数据库 MongoDB 连接信息。

2. Java Demo Code。

○ Maven配置。

```
<dependencies>
  <dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>4.1.0</version>
  </dependency>
</dependencies>
```

○ Java Code。

```
package mongodb;
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 shardingConnectTest {
    public static ServerAddress seed1 = new ServerAddress("11.238.XX.XX",
        3717);
    public static ServerAddress seed2 = new ServerAddress("11.238.XX.XX",
        3717);
    public static String username = "autotest";
    public static String password = "autotest";
    // public static String ReplSetName = "mgset-*****";
    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_D
B,
        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().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[:por
tN]]][/[database][?options]]
        MongoClientURI connectionString = new MongoClientURI("mongodb://" + username
+ ":" + password + "@"
        + seed1 + "," + seed2 + "/" + DEFAULT_DB);
        return new MongoClient(connectionString);
    }
    public static void main(String args[]) {
        MongoClient client = createMongoDBClientWithURI();
        // or
        // MongoClient client = createMongoDBClientWithURI();
        try {
            // 取得Collection句柄
            MongoDatabase database = client.getDatabase(DEMO_DB);
            MongoCollection<Document> collection = database.getCollection(DEMO_COLL);
            // 插入数据

```

```

// 写入数据
Document doc = new Document();
String demoname = "JAVA:" + UUID.randomUUID();
doc.append("DEMO", demoname);
doc.append("MSG", "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 4.1.0下载地址](#)。
- [PyMongo 4.1.0官方文档](#)。

1. 安装pymongo。

```
pip install pymongo
```

2. 获取云数据库MongoDB单节点实例连接信息。

3. Python Demo Code。

```

import uuid
from pymongo import MongoClient
CONN_ADDR1 = '*****.mongodb.****.****.rdstest.aliyun-inc.com:27017'
CONN_ADDR2 = '*****.mongodb.****.****.rdstest.aliyun-inc.com:27017'
REPLICAT_SET = 'mgset-*****'
username = 'demouser'
password = '*****'
#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, MSG="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

```

C#连接示例

C#驱动相关信息，请参见[MongoDB C# Driver](#)。

1. 在客户端执行如下命令，安装如下驱动包。

```
mongocsharpdriver.dll
```

2. 获取云数据库MongoDB单节点实例连接信息。

3. C# Demo Code。

```
using MongoDB.Driver;
using System;
using System.Collections.Generic;
namespace Aliyun
{
    class Program
    {
        static void Main(string[] args)
        {
            //Mongo 实例信息。
            const string host1 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";

            const int port1 = 3717;
            const string host2 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";

            const int port2 = 3717;
            const string replicaSetName = "mgset-300*****";
            const string admin = "admin";
            const string userName = "root";
            const string passwd = "*****";
            try
            {
                Console.WriteLine("开始连接.....");
                MongoClientSettings settings = new MongoClientSettings();
                List<MongoServerAddress> servers = new List<MongoServerAddress>();
                servers.Add(new MongoServerAddress(host1, port1));
                servers.Add(new MongoServerAddress(host2, port2));
                settings.Servers = servers;
                //设置副本集名称。
                settings.ReplicaSetName = replicaSetName;
                //设置超时时间为3秒。
                settings.ConnectTimeout = new TimeSpan(0, 0, 0, 3, 0);
                MongoCredential credentials = MongoCredential.CreateCredential(admin, u
serName, passwd);
                settings.Credential = credentials;
                MongoClient client = new MongoClient(settings);
                var server = client.GetServer();
                MongoDB database = server.GetDatabase("test");
                var collection = database.GetCollection<User>("test_collection");
                User user = new User();
                user.id = "1";
                user.name = "mongo_test";
                user.sex = "女";
                //插入数据user。
```

```
collection.Insert(user);
//获取一条数据。
User result = collection.FindOne();
Console.WriteLine("id:" + result.id + " name:" + result.name + " sex:" +
result.sex);

    Console.WriteLine("连接成功.....");
}
catch (Exception e)
{
    Console.WriteLine("连接异常:"+e.Message);
}
}
}
class User
{
    public string id { set; get; }
    public string name { set; get; }
    public string sex { set; get; }
}
}
```

Go连接示例

Go驱动相关信息，请参见[MongoDB Go Driver](#)。

1. 安装如下驱动包。

```
go get go.mongodb.org/mongo-driver
```

2. 获取云数据库MongoDB分片集群实例连接信息。
3. Go Demo Code。

```
package main
import (
    "context"
    "fmt"
    "go.mongodb.org/mongo-driver/bson"
    "go.mongodb.org/mongo-driver/mongo"
    "go.mongodb.org/mongo-driver/mongo/options"
    "go.mongodb.org/mongo-driver/mongo/readconcern"
    "go.mongodb.org/mongo-driver/mongo/readpref"
    "go.mongodb.org/mongo-driver/mongo/writeconcern"
    "log"
)
func main() {
    // Create a Client to a MongoDB server and use Ping to verify that the
    // server is running.
    clientOpts := options.Client().ApplyURI("mongodb://root:****@s-bp1****.mongodb.rds.aliyuncs.com:3717,s-bp2****.mongodb.rds.aliyuncs.com:3717/admin")
    clientOpts.SetReadPreference(readpref.PrimaryPreferred())
    clientOpts.SetWriteConcern(writeconcern.New(writeconcern.WMajority(), writeconcern.J(true), writeconcern.WTimeout(1000)))
    client, err := mongo.Connect(context.TODO(), clientOpts)
    if err != nil {
        fmt.Println("connect failed!")
    }
}
```

```

        log.Fatal(err)
        return
    }
    fmt.Println("connect successful!")
    defer func() {
        if err = client.Disconnect(context.TODO()); err != nil {
            fmt.Println("disconnect failed!")
            log.Fatal(err)
        }
        fmt.Println("disconnect successful!")
    }()
    // Call Ping to verify that the deployment is up and the Client was
    // configured successfully. As mentioned in the Ping documentation, this
    // reduces application resiliency as the server may be temporarily
    // unavailable when Ping is called.
    if err = client.Ping(context.TODO(), nil); err != nil {
        fmt.Println("ping failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("ping successful!")
    // Specify the DefaultReadConcern option so any transactions started through
    // the session will have read concern majority.
    // The DefaultReadPreference and DefaultWriteConcern options aren't
    // specified so they will be inherited from client and be set to primary
    // and majority, respectively.
    opts := options.Session().SetDefaultReadConcern(readconcern.Majority())
    sess, err := client.StartSession(opts)
    if err != nil {
        fmt.Println("start session failed!")
        log.Fatal(err)
        return
    }
    defer func() {
        sess.EndSession(context.TODO())
        fmt.Println("end session!")
    }()
    fmt.Println("start session successful!")
    txnOpts := options.Transaction()
    result, err := sess.WithTransaction(
        context.TODO(),
        func(sessCtx mongo.SessionContext) (interface{}, error) {
            collection := client.Database("baz").Collection("qux")
            res, err := collection.InsertMany(context.Background(), []interface{}{bson.
M{"shardkey": 0}, bson.M{"shardkey": 1}, bson.M{"shardkey": 2}})
            if err != nil {
                fmt.Println("insert result failed!")
                log.Fatal(err)
                return nil, err
            }
            id := res.InsertedIDs
            fmt.Println("Ids: ", id)
            fmt.Printf("insert result: %v\n", res)
            result := bson.M{}

```

```
filter := bson.D{{"shardkey", 0}}
if err := collection.FindOne(context.Background(), filter).Decode(&result);
err != nil {
    fmt.Println("find failed!")
    log.Fatal(err)
    return nil, err
}
return result, err
},
txnOpts)
if err == nil {
    fmt.Printf("result: %v\n", result)
}
}
```

5.6.4. 通过程序代码连接DynamoDB协议兼容版实例

本文档通过AWS CLI、Python、Java多个维度，介绍如何连接到DynamoDB协议兼容版实例。

前提条件

已创建DynamoDB协议兼容版实例（协议类型选择DynamoDB协议），创建方法请参见[创建分片集群实例](#)。

准备工作

- 获取DynamoDB协议兼容版实例的连接地址。
 - i. 登录[MongoDB管理控制台](#)。
 - ii. 在页面左上角，选择实例所在的资源组和地域。
 - iii. 在左侧导航栏，单击分片集群实例列表。
 - iv. 找到目标实例，单击实例ID。
 - v. 在左侧导航栏，单击数据库连接。
 - vi. 在私网连接 - 专有网络区域，查看实例的连接地址。

私网连接 - 专有网络		更多操作
连接协议类型	连接地址	
DynamoDB兼容协议	dds-1e17****.mongodb.amazonaws.com:27017	

- （可选）如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS服务器满足以下条件，以保证网络的连通性。
 - 所在地域相同。如果您已经创建了ECS服务器，您可以查看ECS所在地域，查看方法请参见[查看ECS实例所在地域](#)。
 - （可选）建议所属可用区相同，以减少网络延时。如果您已经创建了ECS服务器，您可以查看ECS所在可用区，查看方法请参见[查看ECS实例所在地域](#)。
 - 网络类型相同。如果您已经创建了ECS服务器，您可以查看ECS的网络类型，查看方法请参见[查看ECS实例网络类型](#)。当ECS服务器的网络类型为经典网络时，您可以将经典网络迁移至专有网络，迁移方法请参见[ECS实例从经典网络迁移到专有网络](#)。

AWS CLI连接示例

以Ubuntu 16.04.6 LTS系统为例介绍通过AWS CLI（AWS Command Line Interface）连接DynamoDB协议兼容版实例的方法。关于AWS CLI的详细信息，请参见[AWS Command Line Interface是什么](#)。

1. 安装AWS CLI客户端。

- i. 执行如下命令，获取最新版的AWS CLI并将其重命名为 `awscliv2.zip` 。

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

- ii. 执行如下命令，解压 `awscliv2.zip` 文件。

```
unzip awscliv2.zip
```

 **说明** 如果您尚未安装 `unzip`，请先通过 `apt install unzip` 命令安装，然后再执行上面的解压命令。

- iii. 执行如下命令，安装AWS CLI。

```
sudo ./aws/install
```

当命令行窗口中出现 `You can now run: /usr/local/bin/aws --version` 提示，则表示您已成功安装AWS CLI。此时您可以执行 `/usr/local/bin/aws --version` 命令查看当前AWS CLI的版本号。

2. 执行 `/usr/local/bin/aws configure` 配置AWS CLI，分别输入下列几项参数（每输入一项参数按回车）：

参数	说明	示例
AWS Access Key ID	输入您AWS账号的Access Key ID，如没有可以输入任意字符。	XXXXXXXXXX
AWS Secret Access Key	输入您AWS账号的Secret Access Key，如没有可以输入任意字符。	XXXXXXXXXX
Default region name	输入您AWS的DynamoDB数据库所在的地域，如没有可以根据示例输入。	us-west-2
Default output format	默认的输出格式，此处可以留空。	json

 **说明** 更多AWS CLI配置说明的相关信息，请参见[AWS CLI基本配置](#)。

3. 执行如下命令，连接目标DynamoDB协议兼容版实例并创建一张表。

```
aws dynamodb --endpoint-url <DynamoDB协议兼容版实例的连接串地址> \  
create-table --table-name <需要创建的表名> \  
--attribute-definitions AttributeName=<属性名>,AttributeType=<属性的数据类型> \  
--key-schema AttributeName=<指定主键的属性名>,KeyType=<主键的角色> \  
--provisioned-throughput ReadCapacityUnits=<预设读吞吐量>,WriteCapacityUnits=<预设写吞吐量>
```

参数	说明
<code>--endpoint-url</code>	需要连接的目标DynamoDB协议兼容版实例地址。需要以 <code>HTTP://</code> 开头。

参数	说明
<code>create-table</code>	创建表的命令。 ? 说明 更多信息，请参见create-table。
<code>--table-name</code>	指定需要创建表的名称。
<code>--attribute-definitions</code>	描述表或索引整体架构的属性数组。该选项还需指定如下两个子选项： <code>AttributeName</code>：属性的名称。 <code>AttributeType</code>：属性的数据类型。 ? 说明 更多信息，请参见create-table。
<code>--key-schema</code>	指定表或索引的主键属性。该选项还需指定如下两个子选项： <code>AttributeName</code>：指定通过 <code>--attribute-definitions</code> 创建的某个属性名为主键。 <code>KeyType</code>：指定主键的角色。 ? 说明 更多信息，请参见create-table。
<code>--provisioned-throughput</code>	指定表或索引的预设吞吐量。该选项还需指定如下两个子选项： <code>ReadCapacityUnits</code>：指定预设读吞吐量。 <code>WriteCapacityUnits</code>：指定预设写吞吐量。 ? 说明 更多信息，请参见create-table。

示例：

```
/usr/local/bin/aws dynamodb --endpoint-url http://dds-xxxx.mongodb.rds.aliyuncs.com:3717 #连接到指定的DynamoDB协议兼容版实例地址。 \
  create-table --table-name student #创建一个名为student的表。 \
    --attribute-definitions AttributeName=name,AttributeType=S AttributeName=age,AttributeType=N #表的架构中，有类型为String的name属性和类型为Number的age属性。 \
    --key-schema AttributeName=name,KeyType=HASH AttributeName=age,KeyType=RANGE #指定name属性为分区键，并指定age属性为排序键。 \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 #指定预设的读写吞吐量各为5。
```

当命令行中打印出如下内容，则表示您已成功连接到目标DynamoDB协议兼容版实例并创建了一张表。

```
{
  "TableDescription": {
    "TableName": "student",
    "KeySchema": [
      {
        "AttributeName": "name",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "age",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "TableSizeBytes": 0,
    "ItemCount": 0
  }
}
```

Python连接示例

- 已安装Python 2.6或更高版本。更多信息，请参见[Python官网](#)。
- 已安装Boto3。更多信息，请参见[安装Boto3](#)。

以下代码演示通过Python连接到DynamoDB协议兼容版实例并创建一个名为 `Book` 的表格：

```
import boto3
def create_book_table(dynamodb=None):
    if not dynamodb:
        dynamodb = boto3.resource('dynamodb', endpoint_url="http://dds-xxxx.mongodb.rds.aliyuncs.com:3717")
    table = dynamodb.create_table(
        TableName='Book',
        KeySchema=[
            {
                'AttributeName':'title',
                'KeyType':'HASH'
            },
            {
                'AttributeName':'year',
                'KeyType':'RANGE'
            }
        ],
        AttributeDefinitions=[
            {
                'AttributeName':'title',
                'AttributeType':'S'
            },
            {
                'AttributeName':'year',
                'AttributeType':'N'
            },
        ],
        ProvisionedThroughput={
            'ReadCapacityUnits':5,
            'WriteCapacityUnits':5
        }
    )
    return table

if __name__ == '__main__':
    book_table =create_book_table()
    print("Tablestatus:", book_table.table_status)
```

Java连接示例

安装AWS SDK for Java, 具体请参见[安装AWS SDK for Java](#)。

以下代码演示通过Java连接到DynamoDB协议兼容版实例并创建一个名为 `Book` 的表格：

```

package com.amazonaws.codesamples.gsg;
import java.util.Arrays;
import com.amazonaws.client.builder.AwsClientBuilder;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.services.dynamodbv2.document.DynamoDB;
import com.amazonaws.services.dynamodbv2.document.Table;
import com.amazonaws.services.dynamodbv2.model.AttributeDefinition;
import com.amazonaws.services.dynamodbv2.model.KeySchemaElement;
import com.amazonaws.services.dynamodbv2.model.KeyType;
import com.amazonaws.services.dynamodbv2.model.ProvisionedThroughput;
import com.amazonaws.services.dynamodbv2.model.ScalarAttributeType;
public class MoviesCreateTable {
    public static void main(String[] args) throws Exception {

        AmazonDynamoDB client =AmazonDynamoDBClientBuilder.standard()
            .withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("http://d
ds-xxxx.mongodb.rds.aliyuncs.com:3717",
"us-east-1"))
            .build();
        DynamoDB dynamoDB = new DynamoDB(client);
        String tableName ="Book";
        try {
            System.out.println("Creating table...");
            Table table =dynamoDB.createTable(tableName,
                Arrays.asList(new
KeySchemaElement("title", KeyType.HASH), // 分区键
                    new KeySchemaElement("year", KeyType.RANGE)), // 排序键
                Arrays.asList(new AttributeDefinition("title", ScalarAttributeType.S),
                    new AttributeDefinition("year", ScalarAttributeType.N)),
                new ProvisionedThroughput(5L, 5L));
            table.waitForActive();
            System.out.println("OK. Table status: " + table.getDescription().getTableStatus(
));
        }
        catch (Exception e) {
            System.err.println("Unable to create table: ");
            System.err.println(e.getMessage());
        }
    }
}

```

其他程序代码的连接示例

请参见[开始使用DynamoDB和AWS开发工具包](#)。

6.Serverless快速入门

6.1. 使用限制

在使用云数据库MongoDB Serverless实例前，您需要了解Serverless实例本身存在的约束信息，使您初步了解Serverless实例的使用限制，帮助您进一步确认Serverless实例是否能够满足您的业务场景。

功能	约束
数据库版本	仅支持MongoDB 4.2版本。
搭建Serverless	- 实例运行过程中，您可以根据业务需要变更实例规格，具体请参见变更Serverless实例配置。 - 不支持释放实例。
修改实例参数	不支持修改实例参数。
数据读写	Serverless实例中的admin数据库仅支持用户读取，不支持写入。
账号管理	系统提供默认的数据库账号，但不支持创建新的数据库账号。
数据库连接	- 不支持开启或关闭私网免密访问功能。 - 不支持将经典网络切换为专有网络。
数据备份与恢复	仅支持通过MongoDB工具（mongodump和mongoexport）备份与恢复数据库，具体请参见 使用MongoDB工具备份与恢复MongoDB Serverless版实例 。
服务可用性	不支持服务可用性功能，包括主备切换、角色切换和迁移可用区。
数据安全性	- 不支持审计日志、SSL和TDE。 - 仅支持专有网络，专有网络的创建方法请参见创建和管理专有网络。 - 仅支持设置白名单，设置方法请参见设置白名单。
日志管理	不支持日志管理功能，包括慢日志、错误日志和运行日志。
CloudDBA（性能诊断与优化）	不支持CloudDBA功能，包括 性能趋势 、 实时性能 、 实例会话 、 空间分析 和 慢日志 。
数据迁移与同步	支持使用如下方法迁移Serverless实例数据。 使用MongoShake将Serverless数据迁移至副本集或分片集群实例 使用MongoDB工具备份与恢复MongoDB Serverless版实例
连接实例	- 不支持通过DMS连接Serverless实例。 - 支持如下方法连接Serverless实例。 通过Mongo Shell连接云数据库MongoDB Serverless实例 通过程序代码连接Serverless实例

功能	约束
回收站	不支持回收站功能。
重启实例	不支持重启。

6.2. 创建Serverless实例

云数据库MongoDB提供的Serverless实例适用于中小型客户群体。本文介绍云数据库MongoDB Serverless实例的创建方法。

前提条件

已注册阿里云账号，注册流程请参见[注册阿里云账号](#)。

注意事项

- 如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS服务器满足以下条件，以保证网络的连通性。
- 所在地域相同。如果您已经创建了ECS服务器，您可以查看ECS所在地域，查看方法请参见[查看ECS实例所在地域](#)。
 - （可选）建议所属可用区相同，以减少网络延时。如果您已经创建了ECS服务器，您可以查看ECS所在可用区，查看方法请参见[查看ECS实例所在地域](#)。
 - 网络类型为**专有网络**。如果您已经创建了ECS服务器，您可以查看ECS的网络类型，查看方法请参见[查看ECS实例网络类型](#)，当ECS服务器的网络类型为**经典网络**时，您可以将**经典网络**迁移至**专有网络**，迁移方法请参见[ECS实例从经典网络迁移到专有网络](#)。

费用说明

云数据库MongoDB Serverless实例支持包年包月付费类型，关于收费信息请参见[收费项目及价格说明](#)。

如果您是首次购买云数据库MongoDB实例，您可以参与数据库首购折扣价活动，活动信息请参见[云数据库优惠活动](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击**Serverless实例列表**。
3. 在**Serverless实例列表**页面，单击**创建实例**。
4. 在**商品类型**区域，请根据业务需求，选择**Serverless（预置容量）**。
5. 设置以下参数，完成Serverless实例的参数配置。

参数	参数说明
----	------

参数	参数说明
地域	选择实例所属地域。  说明 - 实例创建成功后将无法更换地域，请您谨慎选择。 - 如果应用部署在ECS实例，请您选择与ECS实例相同的地域，否则无法直接通过内网通信。
可用区	选择实例所属可用区。  说明 - 如果应用部署在ECS实例，建议您选择与ECS实例相同的可用区，以减少网络延时。 - 您可以选择多可用区实现实例的同城容灾。
专有网络	选择实例所属专有网络。如果您没有可以使用的专有网络，您可以在专有网络控制台创建专有网络，创建方法请参见创建和管理专有网络。  说明 如果需要更改网络类型，具体请参见切换实例网络类型。
虚拟交换机	选择实例所属交换机。如果您没有可以使用的交换机，您可以在专有网络控制台创建交换机，创建方法请参见 创建和管理交换机 。
计算规格	选择实例的计算规格。取值范围为100~8000 CU，步长为100 CU。
存储空间	选择实例存储空间的大小。取值范围为1~100 GB，步长为1 GB。
用户密码	设置数据库账号的密码，请按照如下规则设置密码：  注意 Serverless实例创建成功后，系统提供默认的数据库账号，不支持创建新的数据库账号。 - 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !@#\$%^&*()_+~= - 密码长度为8~32位。
实例名称	自定义实例的名称。
购买数量	选择相同配置实例的数量，可以设置1~99的整数。
购买时长	选择购买时长。按天可选择1天、7天和14天；包月可选择1~9个月；包年可选择1~5年。

6. 购买实例。

- i. 配置完成后，单击**立即购买**。
- ii. 在**确认订单**页面，查看配置详情，阅读并勾选服务协议。
- iii. 单击**去支付**，并在支付页面，根据提示完成支付流程。

7. 查看Serverless实例是否创建成功。

- i. 支付完成后，单击**管理控制台**，进入云数据库MongoDB控制台。
- ii. 在页面左上角，选择实例所在的资源组和地域。
- iii. 在左侧导航栏，选择**Serverless实例列表**。
- iv. 查看创建的实例是否在Serverless实例列表中。
 - 是，说明实例已创建，当实例状态为**运行中**时，说明实例创建成功。
 - 否，请您等待10~15分钟后刷新页面重新查看，如果Serverless列表中仍然没有，请先排查是否是以下原因导致，如果不是，请联系阿里云技术支持。

可能导致实例列表中找不到实例的可能原因及解决办法。

可能原因	解决方法
地域选择错误	在云数据库MongoDB控制台左上角，选择实例所在的资源组和地域。
实例列表选择错误	在云数据库MongoDB控制台左侧导航栏，选择 Serverless实例列表 。
资源不足	新创建的实例可能因为资源不足而退款。请您在订单管理中查看退款，具体请参见订单管理。 在您确认退款后，您可以选择以下任何一种方法处理： ■ 更换可用区尝试再次购买。 ■ 提交工单处理，具体请参见提交工单。

相关API

接口	说明
CreateServerlessDBInstance	创建云数据库MongoDB Serverless版实例。
DescribeInstanceAutoRenewalAttribute	查询云数据库MongoDB实例是否为自动付费。
DescribeDBInstanceAttribute	查询云数据库MongoDB实例的详情。

后续步骤

实例创建完成后，您还需要执行以下操作：

- （可选）如果您在创建实例时没有立即设置密码，请您设置密码，设置方法请参见（可选）[重置密码](#)。
- 设置实例的白名单，以允许外部设备访问该实例，设置方法请参见[设置白名单](#)。
- （可选）如果您需要通过公网连接实例，请您先申请公网连接地址，申请方法请参见（可选）[申请公网连接地址](#)。

6.3. （可选）重置密码

如果您忘记账号密码、修改旧密码，或创建实例时未设置密码，您可以通过云数据库MongoDB提供的重置密码功能重新设置密码。

操作步骤

- 1. 登录MongoDB管理控制台。
- 2. 在左侧导航栏，单击Serverless实例列表。
- 3. 在页面左上角，选择实例所在的资源组和地域。
- 4. 单击目标实例ID或目标实例所在行操作列的, 并选择管理。
- 5. 在基本信息页面的连接信息区域，单击账号名右侧的重置密码。
- 6. 在重置密码面板，设置以下参数。

参数	参数说明
账号	固定为需要重置密码的账号名。
新密码	按照以下规则设置账号名对应的新密码。 - 密码由大写字母、小写字母、数字、特殊字符中的至少三种组成，特殊字符为： !#\$%^&*()_+= - 密码长度为8~32位。
确认新密码	再次输入新密码。

- 7. 单击确定。

6.4. 设置白名单

为保障MongoDB数据库的安全稳定，系统默认任何设备均无法访问云数据库MongoDB实例（默认IP地址为127.0.0.1），所以在使用云数据库MongoDB实例前，您需要为实例设置白名单，以允许外部设备访问该实例。正确使用白名单可以使云数据库MongoDB实例得到更高级别的安全保护，建议您定期维护白名单。

操作步骤

- 1. 登录MongoDB管理控制台。
- 2. 在左侧导航栏，单击Serverless实例列表。
- 3. 在页面左上角，选择实例所在的资源组和地域。
- 4. 单击目标实例ID或目标实例所在行操作列的, 并选择管理。
- 5. 在目标实例页面的左侧导航栏，单击安全设置。
- 6. 在添加白名单分组区域，选择以下任何一种方法为实例设置白名单。
 - 手动添加

说明

- 支持如下格式的IP地址或IP地址段。
 - 单个IP地址，例如10.23.12.24。
 - 0.0.0.0/0
 - CIDR模式，即无类域间路由。例如10.23.12.24/24，其中/24表示了地址中前缀的长度，范围[1，32]。
- 多个IP地址或IP地址段之间以英文逗号（,）分隔。
- 设置为0.0.0.0/0或空表示对访问的IP地址来源不作任何限制，此时云数据库MongoDB数据库将面临高安全风险，请谨慎使用。

- a. 单击目标分组操作列的，并选择手动修改。
- b. 在手动修改面板的允许访问IP名单输入框中输入IP地址或IP地址段。
- c. 单击确定。
- o 加载ECS内网IP添加
 - a. 单击目标分组操作列的，并选择加载ECS内网IP添加。
 - b. 在加载ECS内网IP添加面板的允许访问IP名单列表中，选择需要添加的ECS内网IP地址。
 - c. 单击.
 - d. 单击确定。

相关任务

- 添加白名单分组
- 添加安全组

6.5. （可选）申请公网连接地址

云数据库MongoDB支持公网连接地址，您可以为实例申请公网连接地址用于使用公网连接云数据库MongoDB实例的场景。本文介绍为云数据库MongoDB Serverless实例申请公网连接地址的方法。

背景信息

云数据库MongoDB Serverless实例支持的连接地址类型及对应说明如下。

地址类型	说明
专有网络连接地址	● 专有网络是一种隔离的网络环境，安全性和性能均高于传统的经典网络。 ● 云数据库MongoDB Serverless实例默认提供专有网络连接地址，通过专有网络连接云数据库MongoDB Serverless实例可以获取更高的安全性和连接性能。
公网连接地址	● 由于通过公网连接实例存在一定的安全风险，云数据库MongoDB Serverless实例默认提供专有网络连接地址。 ● 如果您要使用阿里云以外的设备（例如本地设备）连接云数据库MongoDB Serverless实例，您可以手动申请公网连接地址。

注意事项

- 申请公网连接地址过程中，可能会出现实例重新启动的情况，建议您在业务低峰期进行操作。
- 如果需要通过申请的公网地址来连接实例，您需要将连接实例的终端所在的公网IP地址加入到实例的白名单中，白名单设置方法请参见[设置白名单](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击[Serverless实例列表](#)。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的，并选择[管理](#)。
5. 在目标实例页面的左侧导航栏，单击[数据库连接](#)。
6. 单击公网连接区域右侧的[申请公网地址](#)。
7. 在弹出的[申请公网地址](#)对话框中，单击[确定](#)。

执行结果

申请操作完成后，您可以在以下位置查看公网连接地址。关于连接地址的信息，请参见[Serverless实例连接说明](#)。

- [基本信息](#)页面的[连接信息](#)区域。
- [数据库连接](#)页面的[公网连接](#)区域。

相关文档

- 通过公网地址连接实例的方法请参见[如何通过公网连接MongoDB实例](#)。
- 为保障数据安全性，如您不再需要公网连接地址，您可以将公网地址释放，释放方法请参见[释放公网连接地址](#)。

6.6. 连接实例

6.6.1. 通过Mongo Shell连接云数据库MongoDB

Serverless实例

Mongo Shell是MongoDB自带的数据库管理工具，您可以将Mongo Shell工具下载并安装至客户端（本地服务器或ECS服务器），用于连接云数据库MongoDB Serverless实例。本文介绍通过Mongo Shell连接云数据库MongoDB Serverless实例的方法。

前提条件

已创建云数据库MongoDB Serverless实例。

注意事项

云数据库MongoDB Serverless实例提供的默认数据库admin，用于管理admin数据库账号（即创建云数据库MongoDB Serverless实例生成的账号）的信息，不支持进行任何操作。您在使用云数据库MongoDB Serverless实例时，需要先创建admin之外的数据库，然后对该数据库进行操作。

准备工作

- 下载并安装与云数据库MongoDB实例数据库版本相同的MongoDB至客户端，安装方法请参见[Install MongoDB](#)。

如果应用部署在ECS实例，您需要确保云数据库MongoDB实例和ECS实例满足以下条件，以保证网络的连通性。ECS实例信息的查看方法请参见[查看实例信息](#)。

- 所在地域相同，并建议所在可用区相同（以减少网络延时）。
- 网络类型为**专有网络**，需确保使用相同的专有网络ID。

 **说明** 如果ECS实例的网络类型为经典网络，您可以将ECS实例网络类型转换为专有网络，具体请参见[ECS实例从经典网络迁移到专有网络](#)。

- 将客户端IP地址添加至云数据库MongoDB实例的白名单中，添加方法请参见[设置白名单](#)。

 **说明** 如果您需要使用公网地址连接云数据库MongoDB实例，您可以手动申请公网地址，申请方法请参见 [（可选）申请公网连接地址](#)。

操作步骤

1. 登录[MongoDB管理控制台](#)。
2. 在左侧导航栏，单击**Serverless实例列表**。
3. 在页面左上角，选择实例所在的资源组和地域。
4. 单击目标实例ID或目标实例所在行操作列的，并选择**管理**。
5. 在目标实例页面的左侧导航栏，单击**数据库连接**。
6. 查看并获取云数据库MongoDB Serverless实例的连接地址。
7. 执行如下命令，在安装有MongoDB的客户端（本地服务器或ECS实例）上连接云数据库MongoDB Serverless实例。

```
mongo --host <host>:<port> -u <username> -p --authenticationDatabase <database>
```

说明

- `<host>`：云数据库MongoDB Serverless实例的地址。
- `<port>`：云数据库MongoDB Serverless实例的端口号。
- `<username>`：云数据库MongoDB Serverless实例的账号名。
- `<database>`：云数据库MongoDB Serverless实例的鉴权数据库名，即数据库账号所属的数据库。默认为admin。如果您需要使用其他数据库，您可以先在该数据库中创建账号，然后使用该账户。创建账号的方法请参见[db.createUser\(\)](#)。

示例：

```
mongo --host dds-bplmjur6cuum****.mongodb.rds.aliyuncs.com:3717 -u user5495**** -p --authenticationDatabase admin
```

8. 回显如下信息，输入数据库账号对应的密码并按Enter。

```
Enter password:<password>
```

❓ 说明

- `<password>`：云数据库MongoDB Serverless实例账号名对应的密码。输入密码时，密码不可见。
- 如果忘记数据库账号对应的密码，您可以重新设置密码，设置方法请参见（可选）重置密码。

回显如下类似信息，说明连接成功。

❓ 说明 使用不同版本MongoDB中的Mongo Shell连接实例后回显信息不同，请以实际情况为准。

```
connecting to: mongodb://dds-bp1mjur6cuum****.mongodb.rds.aliyuncs.com:3717/?compressor
s=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("26c55fd4-5215-4e8a-b07e-2647146c****") }
MongoDB server version: 4.2.10
```

9. 创建admin之外数据库实现业务需求。创建方法，请参见[db.createUser\(\)](#)。

其他连接场景

- [如何通过公网连接MongoDB实例](#)
- [不同网络类型的ECS实例与MongoDB实例如何连接](#)
- [ECS实例与MongoDB实例地域不同时如何连接](#)
- [ECS实例与MongoDB实例不在同一阿里云账号时如何连接](#)

常见问题

- [排查Mongo Shell登录问题](#)
- [排查因连接数耗尽导致的数据库连接问题](#)
- [排查MongoDB CPU使用率高的问题](#)
- [如何查询及限制连接数](#)

6.6.2. 通过程序代码连接Serverless实例

云数据库MongoDB版完全兼容MongoDB协议，本文介绍用不同语言的程序代码来连接数据库的相关示例。

前提条件

- 获取Serverless实例连接地址。具体操作，请参见[Serverless实例连接说明](#)。
- 根据您使用的语言下载并安装官方驱动程序。更多信息，请参见[MongoDB Drivers](#)。

❓ 说明 连接Serverless实例时，无需指定下列Demo中的replicaSet相关参数。

Node.js连接示例

Node.js驱动相关信息，请参见[MongoDB Node.js Driver](#)。

1. 在客户端执行如下命令，进行项目初始化。

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

2. 执行如下命令，安装驱动包以及工具包。

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

3. 获取云数据库MongoDB Serverless实例连接信息。
4. Node.js Demo Code。

```
const sprintf = require("sprintf-js").sprintf;
const MongoClient = require('mongodb').MongoClient;
const host1 = "dds-*****.mongodb.rds.aliyuncs.com";
const port1 = 3717;
const host2 = "dds-*****.mongodb.rds.aliyuncs.com";
const port2 = 3717;
const username = "root";
const password = "*****";
const replSetName = "mgset-*****";
const demoDb = "test";
const demoColl = "testColl";
//官方建议使用副本集地址，确保高可。
const url = sprintf("mongodb://%s:%s@%s:%d,%s:%d/admin?replicaSet=%s",
    username, password, host1, port1, host2, port2, replSetName);
console.info("url:", url);
const client = new MongoClient(url);
//获取mongoClient
async function run() {
    try {
        //连接实例。
        await client.connect();
        //取得数据库句柄。
        const database = client.db(demoDb);
        //取得Collection句柄。
        const collection = database.collection(demoColl);
        const demoName = "Node For Demo";
        const doc = { "DEMO": demoName, "MESG": "Hello AliCoudDB For MongoDB" };
        console.info("ready insert document: ", doc);
        //插入数据。
        const result = await collection.insertOne(doc);
        console.log(
            `A document was inserted with the _id: ${result.insertedId}`,
        );
        //读取数据。
        const filter = { "DEMO": demoName };
        const findResult = await collection.find(filter);
        await findResult.forEach(console.dir);
    } finally {
        //关闭连接。
        await client.close();
    }
}
run().catch(console.dir);
```

PHP连接示例

PHP驱动相关信息，请参见[MongoDB PHP Driver](#)。

1. 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|.*:|s*||" `
composer require "mongodb/mongodb:^1.0.0"
```


2. 获取云数据库MongoDB Serverless实例连接信息。

3. PHP Demo Code。

```
<?php
require 'vendor/autoload.php'; // include Composer goodies
# 实例信息。
$demo_seed1 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_seed2 = '*****.mongodb.test.aliyun-inc.com:3717';
$demo_replname = "mgset-*****";
$demo_user = 'root';
$demo_password = '*****';
$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 = $collection->insertOne(['name' => 'ApsaraDB for MongoDB', 'desc' => 'Hello, M
ongodb']);
echo "Inserted with Object ID '{$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](#)文档。
- 下载[jar包](#)。

1. 获取云数据库MongoDB Serverless实例连接信息。

2. Java Demo Code。

o Maven配置。

```
<dependencies>
    <dependency>
        <groupId>org.mongodb</groupId>
        <artifactId>mongo-java-driver</artifactId>
        <version>4.1.0</version>
    </dependency>
</dependencies>
```

o 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 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("*****.mongodb.tbc
3.newtest.rdstest.aliyun-inc.com",
        27017);
    public static ServerAddress seed2 = new ServerAddress("*****.mongodb.tbc
3.newtest.rdstest.aliyun-inc.com",
        27017);

    public static String username = "demouser";
    public static String password = "*****";
    public static String ReplSetName = "mgset-*****";
    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, D
EFAULT_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().requiredRep
licaSetName(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],...[,ho
stN[:portN]]][/[database][?options]]
        MongoClientURI connectionString = new MongoClientURI("mongodb://" + u
sername + ":" + password + "@"
            + seed1 + "," + seed2 + "/" + DEFAULT_DB + "?replicaS
et=" + ReplSetName);
        return new MongoClient(connectionString);
    }
    public static void main(String args[]) {
        MongoClient client = createMongoDBClient();
        // or
        // MongoClient client = createMongoDBClientWithURI();
        try {
            // 取得Collection句柄。

```

```
// 获取Collection实例
MongoDatabase 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("MSG", "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).iterate
or();

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](#)。

1. 安装pymongo。

```
pip install pymongo
```

2. 获取云数据库MongoDB Serverless实例连接信息。

3. Python Demo Code。

```
import uuid
from pymongo import MongoClient
CONN_ADDR1 = '*****.mongodb.****.*****.rdstest.aliyun-inc.com:27017'
CONN_ADDR2 = '*****.mongodb.****.*****.rdstest.aliyun-inc.com:27017'
REPLICAT_SET = 'mgset-*****'
username = 'demouser'
password = '*****'
#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, MSG="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
```

C#连接示例

C#驱动相关信息，请参见[MongoDB C# Driver](#)。

1. 在客户端执行如下命令，安装如下驱动包。

```
mongocsharpdriver.dll
```

2. 获取云数据库 MongoDB 连接信息。
3. C# Demo Code。

```
using MongoDB.Driver;
using System;
using System.Collections.Generic;
namespace Aliyun
{
    class Program
    {
        static void Main(string[] args)
        {
            //Mongo 实例信息
            const string host1 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";
            const int port1 = 3717;
            const string host2 = "dds-t4n*****.mongodb.singapore.rds.aliyuncs.
com";
            const int port2 = 3717;
            const string replicaSetName = "mgset-300*****";
            const string admin = "admin";
            const string userName = "root";
            const string passwd = "*****";
            try
            {
                Console.WriteLine("开始连接.....");
                MongoClientSettings settings = new MongoClientSettings() {
```

```

MongoClientSettings settings = new MongoClientSettings();
List<MongoServerAddress> servers = new List<MongoServerAddress>();
servers.Add(new MongoServerAddress(host1, port1));
servers.Add(new MongoServerAddress(host2, port2));
settings.Servers = servers;
//设置副本集名称
settings.ReplicaSetName = replicaSetName;
//设置超时时间为3秒
settings.ConnectTimeout = new TimeSpan(0, 0, 0, 3, 0);
MongoCredential credentials = MongoCredential.CreateCredential(admin, u
serName, passwd);
settings.Credential = credentials;
MongoClient client = new MongoClient(settings);
var server = client.GetServer();
MongoDatabase database = server.GetDatabase("test");
var collection = database.GetCollection<User>("test_collection");
User user = new User();
user.id = "1";
user.name = "mongo_test";
user.sex = "女";
//插入数据user
collection.Insert(user);
//获取一条数据
User result = collection.FindOne();
Console.WriteLine("id:" + result.id + " name:" + result.name + " sex:" +
result.sex);
Console.WriteLine("连接成功.....");
}
catch (Exception e)
{
    Console.WriteLine("连接异常:" + e.Message);
}
}
}
class User
{
    public string id { set; get; }
    public string name { set; get; }
    public string sex { set; get; }
}
}

```

Go 连接示例

相关链接：[MongoDB Go Driver](#)。

1. 安装如下驱动包。

```
go get go.mongodb.org/mongo-driver
```

2. 获取云数据库MongoDB Serverless实例连接信息。
3. Go Demo Code。

```

package main
import (
    "context"

```

```

"context"
"fmt"
"go.mongodb.org/mongo-driver/bson"
"go.mongodb.org/mongo-driver/mongo"
"go.mongodb.org/mongo-driver/mongo/options"
"go.mongodb.org/mongo-driver/mongo/readconcern"
"go.mongodb.org/mongo-driver/mongo/readpref"
"go.mongodb.org/mongo-driver/mongo/writeconcern"
"log"
)

func main() {
    // Create a Client to a MongoDB server and use Ping to verify that the
    // server is running.
    clientOpts := options.Client().ApplyURI("mongodb://root:****@s-bp1*****.mongodb.r
ds.aliyuncs.com:3717,s-bp2*****.mongodb.rds.aliyuncs.com:3717/admin")
    clientOpts.SetReadPreference(readpref.PrimaryPreferred())
    clientOpts.SetWriteConcern(writeconcern.New(writeconcern.WMajority(), writeconcern.
J(true), writeconcern.WTimeout(1000)))
    client, err := mongo.Connect(context.TODO(), clientOpts)
    if err != nil {
        fmt.Println("connect failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("connect successful!")
    defer func() {
        if err = client.Disconnect(context.TODO()); err != nil {
            fmt.Println("disconnect failed!")
            log.Fatal(err)
        }
        fmt.Println("disconnect successful!")
    }()
    // Call Ping to verify that the deployment is up and the Client was
    // configured successfully. As mentioned in the Ping documentation, this
    // reduces application resiliency as the server may be temporarily
    // unavailable when Ping is called.
    if err = client.Ping(context.TODO(), nil); err != nil {
        fmt.Println("ping failed!")
        log.Fatal(err)
        return
    }
    fmt.Println("ping successful!")
    // Specify the DefaultReadConcern option so any transactions started through
    // the session will have read concern majority.
    // The DefaultReadPreference and DefaultWriteConcern options aren't
    // specified so they will be inherited from client and be set to primary
    // and majority, respectively.
    opts := options.Session().SetDefaultReadConcern(readconcern.Majority())
    sess, err := client.StartSession(opts)
    if err != nil {
        fmt.Println("start session failed!")
        log.Fatal(err)
        return
    }
    defer func() {

```

```

    sess.EndSession(context.TODO())
    fmt.Println("end session!")
}()
fmt.Println("start session successful!")
txnOpts := options.Transaction()
result, err := sess.WithTransaction(
    context.TODO(),
    func(sessCtx mongo.SessionContext) (interface{}, error) {
        collection := client.Database("baz").Collection("qux")
        res, err := collection.InsertMany(context.Background(), []interface{}{bson.
M{"shardkey": 0}, bson.M{"shardkey": 1}, bson.M{"shardkey": 2}})
        if err != nil {
            fmt.Println("insert result failed!")
            log.Fatal(err)
            return nil, err
        }
        id := res.InsertedIDs
        fmt.Println("Ids: ", id)
        fmt.Printf("insert result: %v\n", res)
        result := bson.M{}
        filter := bson.D{{"shardkey", 0}}
        if err := collection.FindOne(context.Background(), filter).Decode(&result);
err != nil {
            fmt.Println("find failed!")
            log.Fatal(err)
            return nil, err
        }
        return result, err
    },
    txnOpts)
if err == nil {
    fmt.Printf("result: %v\n", result)
}
}

```