阿里云
云数据库 MongoDB 版

用户指南

文档版本：20180912

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
	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。	 **禁止：** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 云上灾备和多活

2 安全

2.1 SSL加密

为提高数据链路的安全性，您可以启用SSL（Secure Sockets Layer）加密，并安装SSL CA证书到您的应用服务。

SSL在传输层对网络连接进行加密，提升通信数据安全性的同时，保证数据的完整性。

关于SSL的设置（查看SSL设置详情、开通SSL、更新SSL、关闭SSL），您可以通过控制台或者API *DescribeDBInstanceSSL*、*ModifyDBInstanceSSL*操作。SSL CA证书只能通过控制台来下载。本章节介绍如何通过控制台查看SSL详情、开通、更新、关闭SSL以及下载SSL CA证书。

说明：

- 目前仅副本集实例支持SSL加密。
- 在开通、更新、下载以及关闭SSL过程中，实例会重启一次，建议您在业务低峰期做以上操作。

开通SSL

1. 登录*MongoDB管理控制台*。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的  > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的**数据安全性** > **SSL** 。
5. 在**SSL**页面，单击未开通旁边的  。
6. 在打开的**重启实例**对话框中，单击**确定**。

说明：

开通SSL加密时，实例会重启一次，建议您在业务低峰期开通SSL。

更新SSL证书有效期

SSL证书的有效期为一年，过了有效期或者在有效期内，您都可以更新SSL证书。

1. 登录*MongoDB管理控制台*。

2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > **SSL** 。
5. 在**SSL**页面，单击更新证书。
6. 在打开的重启实例对话框中，单击确定。

> 说明：
> 更新过程中，实例会重启一次，建议您在业务低峰期更新SSL证书。

下载SSL CA证书

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > **SSL** 。
5. 在**SSL**页面，单击下载证书，将CA证书下载至本地。

关闭SSL

当您不需要SSL时，您可以关闭SSL证书。

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > **SSL** 。
5. 在**SSL**页面，单击已开通旁边的。
6. 在打开的重启实例对话框中，单击确定。

> 说明：
> 关闭SSL加密时，实例会重启一次，建议您在业务低峰期关闭SSL。

相关文档

MongoDB客户端SSL连接示例

2.2 MongoDB客户端SSL连接示例

云数据库MongoDB设置了sslAllowConnectionsWithoutCertificates，使用SSL连接客户端时不需要证书 ，但需要配置Ca验证服务器证书，同时忽略域名检测。

设置SSL加密请参见*SSL加密*。

Node.js SSL连接示例

相关链接：*MongoDB Node.js Driver*

示例代码

将/?ssl = true添加到客户端URI的末尾，sslCA指向ca证书路径，checkServerIndentity设置为false，忽略域名检测。

```
var MongoClient = require('mongodb').MongoClient,
  f = require('util').format,
  fs = require('fs');

// Read the certificate authority
var ca = [fs.readFileSync(__dirname + "/path/to/ca.pem")];

// Connect validating the returned certificates from the server
MongoClient.connect("mongodb://host01:27017,host02:27017,host03:27017
/?replicaSet=myreplset&ssl=true", {
  server: {
      sslValidate:true,
      checkServerIdentity:false,#ignore host name validation
      sslCA:ca
  }
}, function(err, db) {
  db.close();
});
```

PHP SSL连接示例

相关链接：*MongoDB PHP Driver*

示例代码

PHP使用MongoDB\Client::__construct创建client实例。其包含三组参数：$uri、$uriOptions和$driverOptions。

```
function __construct($uri = 'mongodb://127.0.0.1/', array $uriOptions
 = [], array $driverOptions = [])
```

通过$uriOptions设置ssl为true，启用ssl连接。通过$driverOptions设置ca_file指向ca证书路径。allow_invalid_hostname设置为true，忽略域名检测。

```
<?php
$client = new MongoDB\Client(
    'mongodb://host01:27017,host02:27017,host03:27017',
    [   'ssl' => true,
        'replicaSet' => 'myReplicaSet'
    ],
    [
        "ca_file" => "/path/to/ca.pem",
        "allow_invalid_hostname" => true

    ]
);
?>
```

Java SSL连接示例

相关链接：*MongoDB Java Driver*

示例代码

将MongoClientOptions设置sslEnabled为True，启用ssl连接。将sslInvalidHostNameAllowed设置为true，忽略域名检测。

```
import com.mongodb.MongoClientURI;
import com.mongodb.MongoClientOptions;
MongoClientOptions options
= MongoClientOptions.builder().sslEnabled(true).sslInvalidHostNameAl
lowed(true).build();
MongoClient client = new MongoClient("mongodb://host01:27017,host02:
27017,host03:27017/?replicaSet=myreplset", options);
```

java设置ca证书，需要使用keytool工具：

```
keytool -importcert -trustcacerts -file <path to certificate authority
 file>
        -keystore <path to trust store> -storepass <password>
```

在程序中设置JVM 系统属性以指向正确的信任库和密钥库。

```
System.setProperty("javax.net.ssl.trustStore","/trust/mongoStore.ts");
```

```
System.setProperty("javax.net.ssl.trustStorePassword","StorePass");
```

Python SSL连接示例

相关链接：*MongoDB Python Driver*

示例代码

设置ssl=True启用ssl连接，ssl_ca_certs参数用来指向ca文件路径，ssl_match_hostnames设置为False，忽略域名检测。

```
import ssl
from pymongo import MongoClient

uri = "mongodb://host01:27017,host02:27017,host03:27017/?replicaSet=
myreplset"
client = MongoClient(uri,
                     ssl=True,
                     ssl_ca_certs='ca.pem',
                     ssl_match_hostnames=False)
```

C SSL连接示例

相关链接：*MongoDB C Driver*

示例代码

将/?ssl = true添加到客户端URI的末尾，C使用*mongoc_ssl_opt_t*来配置ssl选项，ca_file指向ca证书路径。将allow_invalid_hostname设置为false，忽略域名检测。

```
mongoc_client_t *client = NULL;
client = mongoc_client_new (
      "mongodb://host01:27017,host02:27017,host03:27017/?replicaSet=
myreplset&ssl=true");
const mongoc_ssl_opt_t *ssl_default = mongoc_ssl_opt_get_default ();
mongoc_ssl_opt_t ssl_opts = { 0 };

/* optionally copy in a custom trust directory or file; otherwise the
default is used. */
memcpy (&ssl_opts, ssl_default, sizeof ssl_opts);
ssl_opts.ca_file = "/path/to/ca.pem"
ssl_opts.allow_invalid_hostname = false
mongoc_client_set_ssl_opts (client, &ssl_opts);
```

C ++ SSL连接示例

相关链接：*MongoDB C++ Driver*

示例代码

将/?ssl = true添加到客户端URI的末尾。C++通过 *mongocxx::options::ssl* 设置SSL参数，ca_file参数用来指定ca文件路径。

说明：

mongocxx驱动现不支持忽略域名检测。

```
#include <mongocxx/client.hpp>
#include <mongocxx/uri.hpp>
#include <mongocxx/options/client.hpp>
#include <mongocxx/options/ssl.hpp>

mongocxx::options::client client_options;
mongocxx::options::ssl ssl_options;

// If the server certificate is not signed by a well-known CA,
// you can set a custom CA file with the `ca_file` option.
ssl_options.ca_file("/path/to/ca.pem");

client_options.ssl_opts(ssl_options);

auto client = mongocxx::client{
    uri{"mongodb://host01:27017,host02:27017,host03:27017/?replicaSet=
myreplset&ssl=true"}, client_opts};
```

Scala SSL连接示例

相关链接：*MongoDB Scala Driver*

示例代码

Scala驱动程序使用Netty提供的SSL底层支持与MongoDB服务器进行SSL连接。其中，将MongoClientOptions设置sslEnabled为True，启用ssl连接；将sslInvalidHostNameAllowed设置为true，忽略域名检测。

```
import org.mongodb.scala.connection.{NettyStreamFactoryFactory,
SslSettings}

MongoClientSettings.builder()
                   .sslSettings(SslSettings.builder()
                                           .enabled(true)

                                           .invalidHostNameAllowed(
true)
                                           .build())

                   .streamFactoryFactory(NettyStreamFactoryFactory())
                   .build()
val client: MongoClient = MongoClient("mongodb://host01:27017,host02:
27017,host03:27017/?replicaSet=myreplset")
```

scala设置ca证书与java相同，同样需要使用keytool工具。

```
keytool -importcert -trustcacerts -file <path to certificate authority
 file>
```

```
        -keystore <path to trust store> -storepass <password>
```

在程序中设置JVM 系统属性以指向正确的信任库和密钥库。

```
System.setProperty("javax.net.ssl.trustStore","/trust/mongoStore.ts");
System.setProperty("javax.net.ssl.trustStorePassword","StorePass");
```

Golang SSL连接示例

相关链接：*MongoDB Golang Driver*、*Crypto tls package*

示例代码

Golang驱动程序使用crypto/tls包提供的SSL底层支持与MongoDB服务器进行SSL连接。其中，Config结构用来配置ssl选项 ；RootCAs用来指定ca证书；InsecureSkipVerify设置为true，忽略域名检测。

```
import (
    "crypto/tls"
    "crypto/x509"
    "gopkg.in/mgo.v2
)
rootPEM, err := ioutil.ReadFile("path/to/ca.pem")
roots := x509.NewCertPool()
ok := roots.AppendCertsFromPEM([]byte(rootPEM)
tlsConfig := &tls.Config{
                  RootCAs: roots,
       InsecureSkipVerify: true
}
url := "mongodb://host01:27017,host02:27017,host03:27017/?replicaSet=
myreplset&ssl=true"
dialInfo, err := ParseURL(url)
dialInfo.DialServer = func(addr *ServerAddr) (net.Conn, error) {
    return tls.Dial("tcp", addr.String(), tlsConfig)
}

session, err := DialWithInfo(dialInfo)
if err != nil {
    panic(err)
}
session.Close()
```

2.3 设置白名单

创建MongoDB实例后，需要将允许访问该实例的IP地址或者IP段加入到实例白名单中，以允许外部设备访问该MongoDB实例。本章节介绍如何通过控制台设置白名单。关于如何通过API设置白名单，请参见*ModifySecurityIps*。

系统会为MongoDB实例创建一个默认的default白名单分组，默认的白名单只包含一个默认IP地址127.0.0.1，表示任何设备均无法访问该MongoDB实例。在您设置白名单时，需要先将IP地址127.0.0.1删除，再通过以下两种方法设置白名单。

- 手动修改白名单分组：将要访问实例的IP地址或IP段添加到白名单分组中。
- 加载ECS内网IP：将您当前阿里云账号下与MongoDB实例所属地域相同的ECS实例的IP地址添加到MongoDB实例白名单分组中。

> 说明：
>
> - 若将白名单设置为%或者0.0.0.0/0，表示允许任何IP地址访问该MongoDB实例。该设置将极大降低数据库的安全性，如非必要请勿使用。
> - 系统默认的default白名单分组只能被修改，不能被清空或者删除。
> - 除default分组之外的白名单分组可以删除，但不能被清空。

手动修改白名单分组

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理。
4. 在实例基本信息页面，单击左侧导航栏中的**数据安全性** > **白名单设置**。
5. 单击某个白名单分组名右侧的 > **手动修改**。
6. 在**手动修改白名单**页面中，将允许访问实例的IP地址或者IP段添加到**允许访问IP名单**中。
   - 填写IP段时，如10.10.10.0/24，表示10.10.10.X的IP地址都可以访问该MongoDB实例。
   - 若您需要添加多个IP地址或IP段，请用英文逗号隔开（逗号前后都不能有空格），例如192.168.0.1,172.16.213.9。
7. 单击**确定**。

加载ECS内网IP

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理。
4. 在实例基本信息页面，单击左侧导航栏中的**数据安全性** > **白名单设置**。

5. 单击某个白名单分组名右侧的 > 加载ECS内网IP添加。
6. 在加载ECS内网IP添加页面中，允许访问IP名单下显示您当前阿里云账号下与MongoDB实例所属地域相同的ECS实例的IP地址，选择需要的IP地址，单击将其添加至白名单中。
7. 单击确定。

删除白名单分组

你可以删除除default分组之外的白名单分组。

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > 白名单设置。
5. 单击某个白名单分组名右侧的 > 删除白名单分组，弹出删除白名单分组提示框。
6. 单击确定，删除白名单分组。

2.4 审计日志

MongoDB审计日志记录了您对数据库执行的所有操作。通过审计日志记录，您可以对数据库进行故障分析、行为分析、安全审计等操作，有效帮助您获取数据的执行情况。

前提条件

目前只有副本集实例和分片集群实例支持审计日志，单节点实例暂不支持该功能。

开启和关闭审计日志只能通过控制台操作。详情请参见开启审计日志和关闭审计日志。

查询审计日志可以通过控制台或者API来完成。具体操作，请参见通过控制台查询和下载审计日志，通过*DescribeAuditRecords*查询审计日志。

实例为副本集实例时，您可以通过控制台设置审计的数据库操作类型。详情，请参见审计设置。

集群版实例暂不支持您自行设置审计的数据库操作类型。开启审计日志时，系统自动将admin，slow，query，insert，update，delete数据库操作作为审计项。

开启审计日志

1. 登录*MongoDB*管理控制台。

2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > 审计日志。
5. 单击开启审计，弹出开启日志提示框。

   说明：

   开启审计日志时，CloudDBA索引推荐功能将同步开启。关于CloudDBA索引推荐，请参见推荐索引。
6. 单击确定，开启审计日志。

   开启审计日志后，您可以做以下操作。

查询和下载审计日志

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > 审计日志。
5. 在导出文件下，您可以查询、导出以及下载审计日志。
   - 查询：您可以输入数据库名字（DB）、数据库的登录账户名（User）、集合中的任何一个词或者记录（Keyword），选择或者输入起始时间、截止时间来按条件查询审计日志。如表 *2-1:* 审计日志文件列表所示。

     表 2-1: 审计日志文件列表

参数	说明
数据库名	若在查询时指定数据库的名字，则显示实例中指定数据库的审计日志。 若查询时没有指定数据库名，则显示实例中所有数据库的审计日志。

参数	说明
账号名	若在查询时指定了登录数据库的账户名，则显示实例中指定账户的数据库的审计日志。 若查询时没有指定登录数据库的账户名，则显示实例中所有数据库的审计日志。
客户端IP	若查询时指定了登录数据库的客户端IP，则显示实例中指定登录客户端IP的数据库的审计日志。 若查询时没有指定登录数据库的客户端IP，则显示实例中所有数据库的审计日志。
执行语句	若查询时指定了Keyword，则显示实例中包含Keyword执行语句的数据库审计日志。 若查询时没有指定Keyword，则显示实例中所有数据库的审计日志。
消耗时间（微秒）	数据库语句的执行时间。
返回记录数	数据库语句执行后返回的记录数。
线程ID	-
执行时间	语句的执行时间。

- 导出文件：导出审计日志文件。

  说明：

  如果满足过滤条件的语句总量超过100万条，则只会导出100万条。 导出语句的速度为900行 / 秒，100万条语句的导出时间预估为20分钟。

- 文件列表：查看导出的审计日志文件列表，如表 2-2: 导出审计日志文件列表所示。

表 2-2: 导出审计日志文件列表

参数	说明
文件ID	系统自动生成的审计日志文件ID。
文件状态	审计日志文件有两种文件状态。 • 未开始：系统还未开始或者正在导出审计日志文件。 • 归档完成：成功导出审计日志文件。 说明：

参数	说明
	只有归档完成的文件才能被下载。
审计日志起始时间	审计日志的起始时间。
审计日志结束时间	审计日志的结束时间。
下载地址	单击下载地址，将审计日志下载至本地。
日志文件大小	审计日志文件的大小。

关闭审计日志

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > 审计日志。
5. 单击关闭审计，弹出关闭日志提示框。

   说明：

   关闭审计日志后，对日志的采集将会关闭，也无法对后继的数据库操作进行审计，且之前保存的审计日志也将清除。
6. 单击确定，关闭审计日志。

审计设置

MongoDB副本集实例在开启审计日志后，支持您自行设置审计哪些数据库操作。

1. 登录*MongoDB*管理控制台。
2. 在控制台左上方选择地域。
3. 单击目标实例ID。

   您也可以单击目标实例右侧的 > 管理 。
4. 在实例基本信息页面，单击左侧导航栏中的数据安全性 > 审计日志。
5. 单击审计设置。
6. 在审计设置对话框中，设置审计的数据库操作类型。

您可以选择以下数据库操作。

- admin：运维操作；
- slow：慢查询；
- query：查询；
- insert：插入；
- update：更新；
- delete：删除；
- command：协议命令。例如，aggregate聚合方法等。

说明：

- 在2018年7月份之前开启审计日志的实例，审计日志中默认审计操作类型有admin，slow，insert，update，delete，command。没有设置query查询操作，如有需要，可通过审计设置功能设置。
- 2018年7月份之后开启审计日志的实例，审计日志中默认审计操作类型有admin，slow，query，insert，update，delete，command。

7. 单击**确定**，完成设置。

3 云上灾备和多活