

ALIBABA CLOUD

Alibaba Cloud

云数据库 Redis 版
快速入门

文档版本：20210122

 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> <i>Instance_ID</i>
[] 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.快速入门概览 ----- 05
- 2.步骤1：创建实例 ----- 06
- 3.步骤2：设置白名单 ----- 10
- 4.步骤3：连接实例 ----- 15
 - 4.1. 通过DMS登录Redis ----- 15
 - 4.2. 通过客户端程序连接Redis ----- 16
 - 4.3. 通过redis-cli连接Redis ----- 33
 - 4.4. 如何通过公网连接Redis实例 ----- 39
 - 4.5. Sentinel兼容模式连接 ----- 40
- 5.使用须知 ----- 43
 - 5.1. Redis管理控制台 ----- 43
 - 5.2. 使用限制 ----- 44
 - 5.3. 命令查询导航 ----- 45
 - 5.4. Redis proxy特性说明 ----- 46

1.快速入门概览

快速入门旨在介绍如何创建Redis实例以及连接实例数据库，使用户能够了解从创建Redis实例到连接并管理实例的流程。

目标读者

- 首次购买Redis实例的用户
- 想要了解如何连接Redis实例的用户

快速入门流程图

若您初次使用云数据库Redis版，请先了解[使用限制](#)以及[Redis管理控制台](#)。

云数据库Redis版完全兼容原生Redis命令，并新增了部分自研命令以提供更优质的服务。各版本命令的支持情况与自研命令的说明请参见[Redis命令](#)。

通常，从新购实例到可以开始使用实例，您需要完成如下操作：

Redis快速入门流程图



2.步骤1：创建实例

云数据库Redis分为社区版和企业版，其中企业版包含多种形态：性能增强型、混合存储型，您可以根据本文的步骤创建适应业务需求的Redis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](#)。
- 若您要创建按量付费的实例，请确保您的账户资金充足。

费用

详情请参见[收费项与价格](#)。

操作步骤

1. 登录[Redis管理控制台](#)。
2. 在页面右上角，单击**创建实例**。
3. 在跳转到的购买页面，选择计费方式。
 - **包年包月**：在新建实例时支付费用。适合长期需求，价格比按量付费更实惠，且购买时长越长，折扣越多。
 - **按量付费**：后付费，按小时扣费。适合短期需求，用完可立即释放实例，节省费用。

 **说明** 按量付费可转为包年包月，包年包月无法转为按量付费。

4. 选择实例配置，参数说明如下表所示。

类别	参数	说明
	地域	实例所在的地理位置。购买后无法更换地域。 <ul style="list-style-type: none"> ◦ 根据目标用户所在的地理位置就近选择地域，提升用户访问速度。 ◦ 确保Redis实例与需要连接的ECS实例创建于同一个地域，否则它们无法通过内网互通，只能通过外网连接，无法发挥最佳性能。
	可用区	可用区是指在同一地域内，电力和网络互相独立的物理区域。同一可用区内ECS实例和Redis实例通过内网连接时，网络延时最小。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> 说明 选择多可用区可实现实例的同城容灾。</p> </div>

基本配置类别	参数	说明
	网络类型	<ul style="list-style-type: none"> ◦ 专有网络（推荐）：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。 ◦ 经典网络：经典网络中的云服务在网络上不进行隔离，只能依靠云服务自身的安全组或白名单策略来阻挡非法访问。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> 注意</p> <ul style="list-style-type: none"> ◦ 请确保Redis实例与需要连接的ECS实例或RDS实例网络类型一致，否则它们无法通过内网互通。 ◦ 如果Redis实例与需要连接的ECS实例或RDS实例的网络类型都是专有网络，请确保各实例在同一VPC中，否则它们无法通过内网互通。 ◦ 经典网络中的Redis实例可以切换到专有网络，专有网络中的实例无法切换到经典网络。 </div>
	专有网络	选择实例的专有网络。如果没有专有网络，请参见 创建专有网络 。
	虚拟交换机	选择专有网络下的虚拟交换机（VSwitch）。如果该专有网络下在当前可用区中没有交换机，请参见 创建交换机 。
	版本类型	<ul style="list-style-type: none"> ◦ 社区版：兼容开源Redis协议标准、提供内存加硬盘的混合存储方式的数据库服务。 ◦ 企业版（Tair）：基于社区版开发的Redis产品，在性能、存储介质、数据结构等方面与社区版形成能力互补，详情请参见企业版简介。
	系列类型	<p>版本类型选择为企业版（Tair）时，您可以选择：</p> <ul style="list-style-type: none"> ◦ 性能增强型：采用多线程模型，性能约为同规格社区版实例的3倍，同时提供多种增强型数据结构模块（modules）简化开发，详情请参见性能增强型。 ◦ 混合存储型：整合了内存和磁盘二者的优势，可大幅度降低用户成本，实现性能与成本的平衡，详情请参见混合存储型。
	版本号	<p>Redis的引擎版本。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> 说明 2.8版本的实例即将停止新购，建议您创建最新版本的Redis实例，以获得更多功能和更高的稳定性。</p> </div>
	架构类型	<ul style="list-style-type: none"> ◦ 集群版：可轻松突破Redis自身单线程瓶颈，满足大容量、高性能的业务需求。 ◦ 标准版：采用主从架构，不仅能提供高性能的缓存服务，还支持数据高可靠。 ◦ 读写分离版：可提供高可用、高性能、高灵活的读写分离服务，解决热点数据集中及高并发读取的业务需求，最大化地节约用户运维成本。 <p>详细说明请参见架构信息查询导航。</p>

实例类别	参数	说明
	分片数	选择Redis集群实例的分片数，数据将分布在该集群的各个分片上。 说明 架构类型选择为集群版时，才支持该参数。
	节点类型	<ul style="list-style-type: none"> 架构类型选择为集群版或标准版时，本参数固定为双副本，即主从双机热备架构，保障可用性。 架构类型选择为读写分离版时，您可以根据只读节点的数量选择节点类型。
	实例规格	选择实例的规格，每种规格都有对应的内存大小、连接数上限、带宽限制等，详情请参见 规格查询导航 。 说明 实例创建后会生成数据库元信息，集群架构的实例每个分片均包含30 MB~50 MB的元信息，整个集群中元信息占用的存储空间为所有分片中元信息占用空间之和。
	密码设置	<ul style="list-style-type: none"> 稍后设置：在实例创建完成后设置密码，设置方法请参见修改或重置密码。 立即设置：填入实例的密码。 <ul style="list-style-type: none"> 密码长度为8~32位。 密码需包含大写字母、小写字母、特殊字符和数字中的至少三种。 支持的特殊字符为 <code>!@#%&^*()_+~=</code>。
购买量	实例名称	设置实例的名称，便于后续业务识别。
	购买数量	选择创建相同配置实例的数量，最大数量为99。
	购买时长	选择付费类型为包年包月时，您还需要设置购买时长、是否自动续费。

5. 单击**立即购买**。

6. 在**确认订单**页面，阅读并选中服务协议，根据提示完成支付流程。
支付成功后等待1~5分钟，即可在[Redis管理控制台](#)看到新购买的Redis实例。

常见问题

● 创建实例需要多长时间？

答：创建实例消耗的时间与实例的分片数成正相关。实例的分片数越多，需要的调配的资源越多，创建需要的时间则越长。例如，创建一个标准版-双副本实例需要2到3分钟，创建一个128分片的集群版-双副本实例需要10~15分钟，创建一个256分片的集群版-双副本实例需要20~40分钟。

说明

- 标准版按照单分片实例计算。
- 读写分离版的每个主节点（读写节点）算作一个分片，每个只读副本算作一个分片。

- 找不到创建的实例？

答：可能为如下原因。

可能原因	解决方法
选择了错误的地域	登录 Redis管理控制台 ，重新选择实例所属的地域。
未刷新或过早刷新控制台	您可以等待一段时间（通常为几分钟）后刷新控制台，再查看Redis实例列表中是否会出现实例。
资源不足	新创建的实例可能因为资源不足而退款，您可以在 订单管理 中查看到退款。 在您确认退款后，您可以更换可用区尝试再次购买，您也可以 提交工单 处理。

- 为什么不能创建单副本Redis实例了？

答：单副本Redis实例已于2019年12月19日下线，详细说明请参见[下线通知](#)。

相关API

API	说明
CreateInstance	创建一个Redis实例。

3.步骤2：设置白名单


为保障Redis数据库的安全稳定，系统默认禁止所有IP地址访问Redis实例。在开始使用Redis实例前，您需要将客户端的IP地址或IP地址段添加到Redis实例的白名单中。正确使用白名单可以让Redis得到高级别的访问安全保护，建议您定期维护白名单。

前提条件

- Redis实例的小版本为最新。关于如何升级，请参见[升级小版本](#)。
- 如需通过添加ECS安全组来设置Redis白名单，则Redis实例的引擎版本需为4.0或以上。


白名单设置方法介绍

设置方法	说明
方法一：通过白名单分组设置白名单	手动添加客户端所属的IP地址到Redis实例的白名单，以允许该客户端访问Redis实例。
方法二：通过ECS安全组设置白名单	安全组 是一种虚拟防火墙，用于控制安全组中的ECS实例的出入流量。如果需要授权多个ECS实例访问Redis实例，您可以通过为Redis实例绑定ECS所属安全组的方式实现快速授权（无需手动填写ECS的IP地址），提升运维的便捷性。

 **说明** 您也可以同时通过上述两种方法设置白名单，白名单分组中的IP地址和安全组中的ECS实例都可以访问该Redis实例。

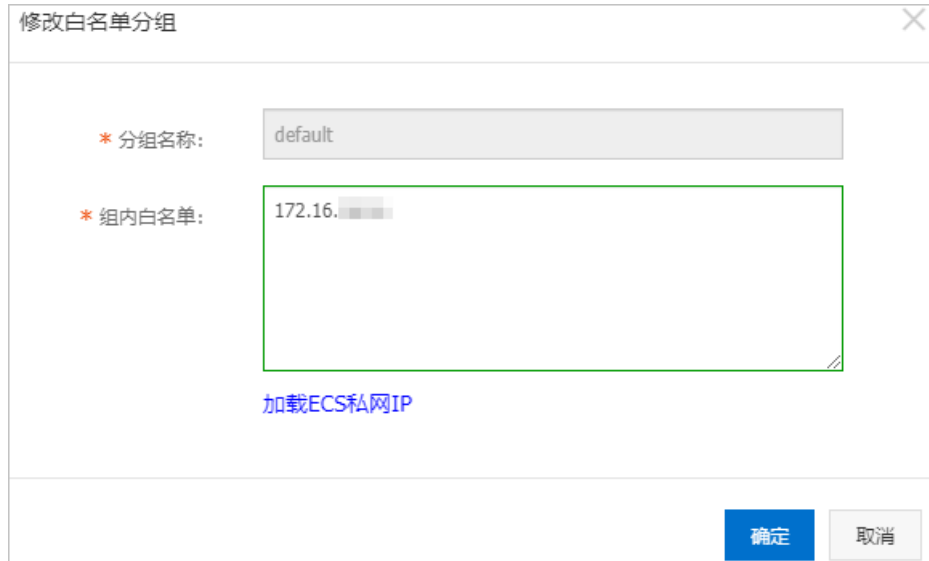
方法一：通过白名单分组设置白名单

1. 登录[Redis管理控制台](#)。
2. 在顶部菜单栏的左上角，选择实例所属的地域。
3. 在实例列表页，单击目标实例ID。
4. 在左侧导航栏，单击**白名单设置**。
5. 找到目标白名单分组，单击**修改**。

 **说明** 您也可以单击**添加白名单分组**创建一个新的分组。分组名称长度为2~32个字符，由小写字母、数字或下划线组成，需以小写字母开头，以小写字母或数字结尾。

6. 在弹出的对话框中，根据要执行的操作，选择下述操作步骤：
 - 手动修改白名单
 - 手动填写IP地址或IP地址段。

手动修改白名单



修改白名单分组

* 分组名称: default

* 组内白名单: 172.16.███

加载ECS私网IP

确定 取消

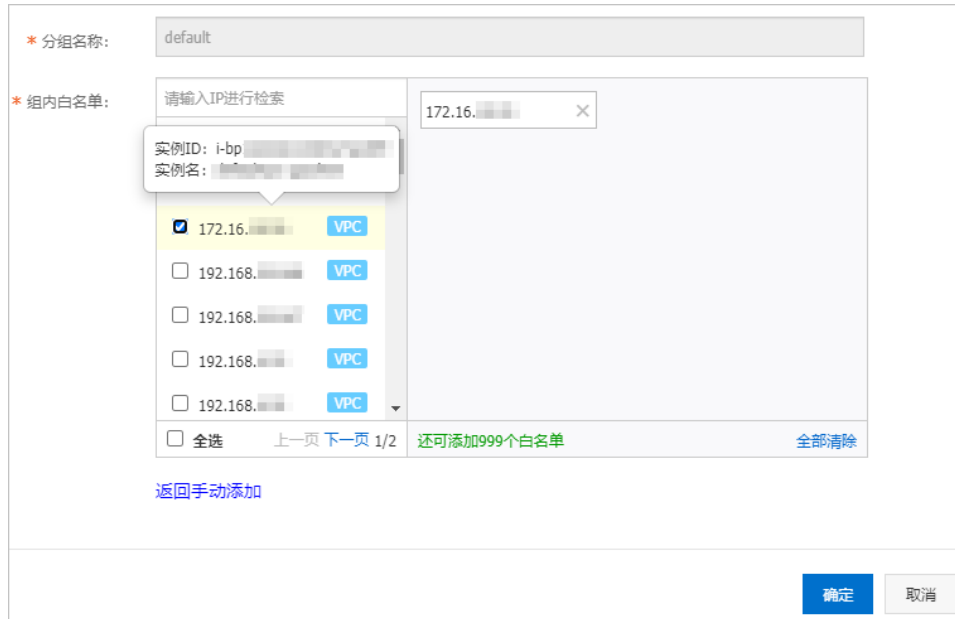
说明

- IP地址以英文逗号分隔，不可重复，最多1000个。支持格式为10.23.12.24（具体IP地址）、10.23.12.24/24（**CIDR模式**，即无类域间路由，/24表示地址中前缀的长度，范围为1~32）。
- 地址中的前缀长度设置为0（例如0.0.0.0/0、127.0.0.1/0）表示允许所有IP地址访问该实例，存在高安全风险，请谨慎设置。

- 加载ECS实例的内网IP地址为白名单
 - a. 单击**加载ECS私网IP**。

b. 根据业务需求选中对应的IP地址。

加载ECS内网IP地址



说明 为便于定位目标IP地址，您可以将鼠标指针悬浮在对应的IP地址上，系统将为您展示该IP地址所属ECS实例的ID和名称。

o 清除白名单分组

当某个白名单分组中的所有IP地址均需要移除且需要保留该分组时，您可以单击清除白名单分组来完成该操作。

7. 单击确定。

方法二：通过ECS安全组设置白名单

将ECS安全组添加至Redis实例后，该安全组中的所有ECS实例既可以通过内网访问Redis实例，也可以通过外网访问Redis实例（Redis实例需已申请公网连接地址，具体操作，请参见[如何通过公网连接Redis实例](#)）。

说明 通过本方法设置白名单时，要求ECS实例和Redis实例的网络类型相同（如果是专有网络，则要求属于同一个VPC ID）。

(可选)

1. 登录Redis管理控制台。
2. 在顶部菜单栏的左上角，选择实例所属的地域。
3. 在实例列表页，单击目标实例ID。
4. 在左侧导航栏，单击白名单设置。
5. 单击添加安全组。
6. 在弹出的对话框中，选择需要添加的安全组。

添加安全组



说明

- 为便于定位目标安全组，您可以将鼠标指针悬浮在安全组ID上，可显示安全组的名称和描述信息；将鼠标指针悬浮在VPC图标上，可显示VPC ID信息。
- 每个Redis实例最多可设置10个安全组。

- 单击**确定**。
- (可选) 当您需要移除所有安全组时，您可以单击**清除安全组**来实现。

常见连接场景

- 如何通过公网连接Redis实例
- ECS实例与Redis实例的网络类型不同时如何连接

相关API

API	说明
DescribeSecurityIps	查询Redis实例的IP白名单。
ModifySecurityIps	设置Redis实例的IP白名单。
DescribeSecurityGroupConfiguration	查询Redis实例白名单中已配置的安全组。
ModifySecurityGroupConfiguration	重新设置Redis实例白名单中的安全组。

常见问题

- Q: Redis实例里有一些自动生成的白名单分组，它们的来源是什么？可以删除吗？
A: 初始情况下，Redis实例的白名单分组仅包含default，随着对实例执行某些操作，白名单分组会逐渐增多，详情请参见下表。

白名单分组名称	来源说明
default	系统默认白名单分组，不可删除。
ali_dms_group	通过DMS登录Redis实例时，授权DMS自动创建的。具体操作，请参见 通过DMS登录Redis 。请勿删除或修改该白名单分组，否则可能导致无法通过DMS登录Redis实例。
hdm_security_ips	使用CloudDBA相关功能时（例如 通过缓存分析快速发现大Key ），授权DAS自动创建的。请勿删除或修改该白名单分组，否则可能导致CloudDBA功能使用异常。

- Q: 为什么通过redis-cli连上后提示 (error) ERR illegal address ?

A: 您的redis-cli所属设备的IP地址未添加至白名单中，请确认白名单配置。

- Q: 我的设备所属的IP地址没有添加到白名单中，可以通过telnet测试端口连通性吗？

A: 可以，telnet命令会返回如下提示：

```
Escape character is '^]'.  
Connection closed by foreign host.
```

4. 步骤3：连接实例

4.1. 通过DMS登录Redis

通过数据管理DMS（Data Management），您无需安装客户端即可快速登录Redis数据库，帮助您可视化地管理Redis数据库。

背景信息

DMS支持管理多种关系型数据库和NoSQL数据库，是一种集数据管理、结构管理、用户授权、安全审计、数据趋势、数据追踪于一体的数据管理服务。您可以使用数据管理服务实现易用的数据库管理入口，让数据更安全、管理更高效、数据价值更清晰。

使用限制

在DMS中执行Redis命令存在一些限制，更多信息，请参见[DMS支持的Redis命令](#)。

操作步骤

1. 登录[Redis管理控制台](#)。
2. 在顶部菜单栏的左上角，选择实例所属的地域。
3. 在实例列表页，单击目标实例ID。
4. 在页面右上角，单击[登录数据库](#)。
5. 在跳转到的DMS控制台，设置登录方式。

设置DMS登录方式

登录实例 ✕

* 数据库类型 ▼

* 实例地区 ▼

* 实例ID ▼

* 访问方式 ✓▼

* 数据库账号 ✓

* 数据库密码 ✓

记住密码 ?

访问方式	说明
账号+密码登录 (推荐)	分别填写数据库账号和对应的密码，关于如何创建数据库账号，请参见 创建与管理账号 。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>说明</p> <ul style="list-style-type: none"> Redis实例默认包含一个以实例ID命名的数据库账号（例如r-bp10noxlhcoim2****），您可以通过该数据库账号来登录（密码在您创建实例时已设置）。 如果忘记密码，您可以重置密码。具体操作，请参见修改或重置密码。 </div>
免密登录	如果Redis实例已开启免密访问，选择该方式无需填写密码即可直接登录。具体操作，请参见 开启专有网络免密访问 。
密码登录	使用创建实例时设置的密码登录Redis（即以实例ID命名的数据库账号对应的密码），如果忘记密码，您可以重置密码。具体操作，请参见 修改或重置密码 。

6. 单击登录。

说明 如您没有将DMS服务器的IP地址添加至Redis实例的白名单中，系统将弹出对话框提示，您需要单击设置白名单，系统会为Redis实例创建一个名为ali_dms_group的白名单分组，并将DMS服务器的IP地址加入该分组中。

完成登录示例



7. 完成登录后，即可在SQLConsole页签对应的文本框中输入并执行Redis命令，例如执行DBSIZE命令查询当前库有多少个键（Key）。关于云数据库Redis支持的命令，请参见[Redis命令支持概览](#)；关于所有Redis命令的介绍和使用方法，请参见[Redis官网](#)。

相关文档

- 通过redis-cli连接Redis
- 通过客户端程序连接Redis

4.2. 通过客户端程序连接Redis

云数据库Redis与原生Redis完全兼容，连接数据库的方式也基本相同，您可以根据自身应用特点选用任何兼容Redis协议的客户端程序。本文列举一些常见的客户端程序的代码示例，帮助您快速连接。

前提条件

根据客户端程序的部署位置，完成下述操作：

客户端程序部署位置	需完成的操作
ECS实例（推荐）	<ol style="list-style-type: none"> 1. 确保ECS实例与Redis实例属于同一专有网络（即实例基本信息中的专有网络ID一致）。 <div data-bbox="485 595 1385 846" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>说明</p> <ul style="list-style-type: none"> 如果专有网络不同，您可以更换ECS实例所属的专有网络。具体操作，请参见更换ECS实例的VPC。 如果网络类型不同，例如ECS实例为经典网络，Redis实例为专有网络。如何连接，请参见ECS实例与Redis实例的网络类型不同时如何连接。 </div> 2. 获取ECS实例的内网IP地址。具体操作，请参见查询ECS实例的IP地址。 3. 将ECS实例的内网IP地址添加至Redis实例的白名单中。具体操作，请参见设置IP白名单。
本地	<ol style="list-style-type: none"> 1. Redis实例默认仅提供内网连接地址，通过公网连接时您需要手动申请公网连接地址。具体操作，请参见申请公网连接地址。 2. 在本地客户端所属的设备上执行<code>curl ipinfo.io grep ip</code>获取其公网IP地址，返回示例如下： <div data-bbox="485 1160 1385 1290" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <pre>root@:~# curl ipinfo.io grep ip % Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 100 249 100 249 0 0 1272 0 --:--:-- --:--:-- --:--:-- 1270 "ip": "192.168.1.203", "readme": "https://ipinfo.io/missingauth"</pre> </div> <div data-bbox="485 1330 1385 1424" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>说明 如果本地客户端所属设备为Windows操作系统，您可以访问ipinfo获取公网地址。</p> </div> 3. 将本地客户端的公网IP地址添加至Redis实例的白名单中。具体操作，请参见设置IP白名单。

注意事项

- 如果您的Redis实例为[集群架构](#)或[读写分离架构](#)，实例默认会提供Proxy（代理）节点的连接地址，连接方式与连接标准架构的Redis实例相同。

说明 集群架构的实例通过[直连地址](#)连接时，连接方式与连接开源Redis Cluster相同。

- 如果实例开启了[专有网络免密访问](#)，同一专有网络下的客户端程序无需设置密码即可连接Redis实例。

如何获取连接信息

在使用客户端程序连接Redis实例时，通常您需要获取以下信息并设置在代码中：

需获取的信息	获取方式
实例的连接地址	Redis实例支持多种连接地址，推荐使用专有网络连接，可获得更高的安全性和更低的网络延迟。更多信息，请参见 查看连接地址 。
端口号	端口号默认为6379，您也可以自定义端口号。具体操作，请参见 修改连接端口 。
实例的账号（部分客户端程序无需设置）	Redis实例默认会创建一个以实例ID命名的账号（例如r-bp10noxlhcoim2****），您也可以创建一个新的账号并赋予权限。更多信息，请参见 创建与管理账号 。
账号的密码	<p>根据选取账号的不同，密码的填写格式有一定区别：</p> <ul style="list-style-type: none"> 默认账号（即以实例ID命名的账号）：直接填写密码即可。 新创建的账号：密码格式为 <code><user>:<password></code>。例如自定义账号为 <code>testaccount</code>，密码为 <code>Rp829dlwa</code>，密码需填写为 <code>testaccount:Rp829dlwa</code>。 <p> 说明 如果忘记密码，请参见修改或重置密码。</p>

常见客户端程序

关于Redis支持的客户端列表，请参见[Redis Clients](#)。

- [Jedis客户端](#)
- [Lettuce客户端](#)（不推荐使用）
- [PhpRedis客户端](#)
- [redis-py客户端](#)
- [C或C++客户端](#)
- [.net客户端](#)
- [node-redis客户端](#)
- [C#客户端StackExchange.Redis](#)

Jedis客户端

1. 下载并安装Jedis客户端。具体操作，请参见[Jedis使用说明](#)。
2. 根据业务需求选择连接方式。
 - [JedisPool连接池连接](#)（推荐）


- a. 打开Eclipse客户端，创建一个Project并配置pom文件，具体内容如下：

```
<dependency>
<groupId>redis.clients</groupId>
<artifactId>jedis</artifactId>
<version>2.7.2</version>
<type>jar</type>
<scope>compile</scope>
</dependency>
```

- b. 在Project中输入下述代码添加相关应用。

```
import org.apache.commons.pool2.PooledObject;
import org.apache.commons.pool2.PooledObjectFactory;
import org.apache.commons.pool2.impl.DefaultPooledObject;
import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
import redis.clients.jedis.HostAndPort;
import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;
import redis.clients.jedis.JedisPoolConfig;
```

- c. 根据Jedis客户端版本，在Project中输入下述代码，然后根据注释提示修改代码。

 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

- Jedis 2.7.2版本


```
JedisPoolConfig config = new JedisPoolConfig();
//最大空闲连接数，需自行评估，不超过Redis实例的最大连接数
config.setMaxIdle(200);
//最大连接数，需自行评估，不超过Redis实例的最大连接数
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
String host = "*.aliyuncs.com";
String password = "密码";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
try {
jedis = pool.getResource();
/// ... do stuff here ... for example
jedis.set("foo", "bar");
String foobar = jedis.get("foo");
jedis.zadd("sose", 0, "car");
jedis.zadd("sose", 0, "bike");
Set<String> sose = jedis.zrange("sose", 0, -1);
} finally {
if (jedis != null) {
jedis.close();
}
}
/// ... when closing your application:
pool.destroy();
```

- Jedis 2.6或Jedis 2.5版本

```
JedisPoolConfig config = new JedisPoolConfig();
//最大空闲连接数，需自行评估，不超过Redis实例的最大连接数
config.setMaxIdle(200);
//最大连接数，需自行评估，不超过Redis实例的最大连接数
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
String host = "*.aliyuncs.com";
String password = "密码";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
boolean broken = false;
try {
    jedis = pool.getResource();
    /// ... do stuff here ... for example
    jedis.set("foo", "bar");
    String foobar = jedis.get("foo");
    jedis.zadd("sose", 0, "car");
    jedis.zadd("sose", 0, "bike");
    Set<String> sose = jedis.zrange("sose", 0, -1);
}
catch(Exception e)
{
    broken = true;
} finally {
if (broken) {
    pool.returnBrokenResource(jedis);
} else if (jedis != null) {
    pool.returnResource(jedis);
}
}
```

- Jedis单连接（不推荐，单次超时无法自动恢复）


打开Eclipse客户端，创建一个Project，输入下述代码，然后根据注释提示修改代码。

 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

```
import redis.clients.jedis.Jedis;
public class jedistest {
public static void main(String[] args) {
try {
String host = "xx.kvstore.aliyuncs.com";//控制台显示访问地址
int port = 6379;
Jedis jedis = new Jedis(host, port);
//鉴权信息
jedis.auth("password");//password
String key = "redis";
String value = "aliyun-redis";
//select db默认为0
jedis.select(1);
//set一个key
jedis.set(key, value);
System.out.println("Set Key " + key + " Value: " + value);
//get 设置进去的key
String getvalue = jedis.get(key);
System.out.println("Get Key " + key + " ReturnValue: " + getvalue);
jedis.quit();
jedis.close();
}
catch (Exception e) {
e.printStackTrace();
}
}
}
```

3. 运行上述Project，在Eclipse的控制台输出如下运行结果则表示您已成功连接至云数据库Redis。

```
Set Key redis Value aliyun-redis
Get Key redis ReturnValue aliyun-redis
```

 **警告** 在使用Jedis的过程中，如果设置了一些不合理的参数或错误使用某些功能可能会引起报错，关于如何排查，请参见[Jedis常见异常汇总](#)。

Lettuce客户端

Lettuce支持完整Redis API的同步和异步通信使用。由于Lettuce客户端在请求多次请求超时后，不再自动重连，当云数据库Redis因故障等因素导致代理或者数据库节点发生切换时，可能出现连接超时导致无法重连。为避免此类风险，推荐您使用[Jedis客户端](#)。

② 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

更多信息，请参见[Lettuce](#)。

TairJedis客户端

TairJedis是阿里云基于Jedis开发的Redis企业版专用客户端，除了Jedis的原有功能，还支持Redis企业版[数据结构模块](#)包含的[命令](#)。

② 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

更多信息，请参见[tairjedis-sdk](#)。

PhpRedis客户端

1. 下载并安装PhpRedis客户端。具体操作，请参见[PhpRedis使用说明](#)。
2. 在PHP编辑器中输入下述代码，然后根据注释提示修改代码。


② 说明 关于如何获取Redis实例的连接地址、账号和密码，请参见[如何获取连接信息](#)。

```
<?php
/* 这里替换为连接的实例连接地址和端口 */
$host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
$port = 6379;
/* 这里替换为实例ID和实例的密码 */
$user = "test_username";
$password = "test_password";
$redis = new Redis();
if ($redis->connect($host, $port) == false) {
    die($redis->getLastError());
}
if ($redis->auth($password) == false) {
    die($redis->getLastError());
}
/* 认证后就可以进行数据库操作，详情请参见https://github.com/phpRedis/phpredis */
if ($redis->set("foo", "bar") == false) {
    die($redis->getLastError());
}
$value = $redis->get("foo");
echo $value;
?>
```

3. 执行上述代码即可完成连接。更多信息，请参见[官方文档](#)。

redis-py客户端

1. 下载并安装redis-py客户端。具体操作，请参见[redis-py使用说明](#)。
2. 在Python编辑器中输入下述代码，然后根据注释提示修改代码。

 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
import redis
#替换host的值为实例的连接地址，替换port的值为端口。
host = 'localhost'
port = 6379
#这里替换为实例password
pwd = 'test_password'
r = redis.StrictRedis(host=host, port=port, password=pwd)
#连接建立后就可以进行数据库操作，详情请参见https://github.com/andymccurdy/redis-py
r.set('foo', 'bar');
print r.get('foo')
```


3. 执行上述代码即可完成连接。

C或C++客户端

1. 执行下述命令，下载并编译安装C客户端。

```
git clone https://github.com/redis/hiredis.git
cd hiredis
make
sudo make install
```

2. 在C或C++编辑器中输入下述代码，然后根据注释提示修改代码。

 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <hiredis.h>
int main(int argc, char **argv) {
    unsigned int j;
    redisContext *c;
    redisReply *reply;
    if (argc < 4) {
        printf("Usage: example xxx.kvstore.aliyuncs.com 6379 instance_id password\n");
        exit(0);
    }
```



```
}
const char *hostname = argv[1];
const int port = atoi(argv[2]);
const char *instance_id = argv[3];
const char *password = argv[4];
struct timeval timeout = { 1, 500000 }; // 1.5 seconds
c = redisConnectWithTimeout(hostname, port, timeout);
if (c == NULL || c->err) {
if (c) {
printf("Connection error: %s\n", c->errstr);
redisFree(c);
} else {
printf("Connection error: can't allocate redis context\n");
}
exit(1);
}
/* AUTH */
reply = redisCommand(c, "AUTH %s", password);
printf("AUTH: %s\n", reply->str);
freeReplyObject(reply);
/* PING server */
reply = redisCommand(c, "PING");
printf("PING: %s\n", reply->str);
freeReplyObject(reply);
/* Set a key */
reply = redisCommand(c, "SET %s %s", "foo", "hello world");
printf("SET: %s\n", reply->str);
freeReplyObject(reply);
/* Set a key using binary safe API */
reply = redisCommand(c, "SET %b %b", "bar", (size_t) 3, "hello", (size_t) 5);
printf("SET (binary API): %s\n", reply->str);
freeReplyObject(reply);
/* Try a GET and two INCR */
reply = redisCommand(c, "GET foo");
printf("GET foo: %s\n", reply->str);
freeReplyObject(reply);
reply = redisCommand(c, "INCR counter");
printf("INCR counter: %lld\n", reply->integer);
freeReplyObject(reply);
/* again ... */
replv = redisCommand(c, "INCR counter");
```

```

reply = redisCommand(c, "INCR counter");
printf("INCR counter: %lld\n", reply->integer);
freeReplyObject(reply);
/* Create a list of numbers, from 0 to 9 */
reply = redisCommand(c, "DEL mylist");
freeReplyObject(reply);
for (j = 0; j < 10; j++) {
    char buf[64];
    snprintf(buf, 64, "%d", j);
    reply = redisCommand(c, "LPUSH mylist element-%s", buf);
    freeReplyObject(reply);
}
/* Let's check what we have inside the list */
reply = redisCommand(c, "LRANGE mylist 0 -1");
if (reply->type == REDIS_REPLY_ARRAY) {
    for (j = 0; j < reply->elements; j++) {
        printf("%u) %s\n", j, reply->element[j]->str);
    }
}
freeReplyObject(reply);
/* Disconnects and frees the context */
redisFree(c);
return 0;
}

```

3. 编译上述代码。

```
gcc -o example -g example.c -I /usr/local/include/hiredis -lhiredis
```

4. 测试运行，完成连接。

```
example xxx.kvstore.aliyuncs.com 6379 instance_id password
```

.net客户端


 **警告** 如果您的Redis实例为**集群架构**或**读写分离架构**，且需要执行切换或选择数据库的操作（即使用多数据库功能），您必须先将`cluster_compat_enable`参数设置为0（即关闭原生Redis Cluster语法兼容），否则将提示报错：`Multiple databases are not supported on this server; cannot switch to database`。具体操作，请参见[参数设置及说明](#)。

1. 执行下述命令，下载.net客户端。

```
git clone https://github.com/ServiceStack/ServiceStack.Redis
```

2. 在.net 客户端中新建.net项目。

3. 添加客户端引用，引用文件在库文件的ServiceStack.Redis/lib/tests中。
4. 在新建的.net项目中输入如下代码，然后根据注释提示修改代码。更多信息，请参见[接口说明](#)。

 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ServiceStack.Redis;
namespace ServiceStack.Redis.Tests
{
    class Program
    {
        public static void RedisClientTest()
        {
            string host = "127.0.0.1";/*访问host地址*/
            string password = "password";/*密码*/
            RedisClient redisClient = new RedisClient(host, 6379, password);
            string key = "test-aliyun";
            string value = "test-aliyun-value";
            redisClient.Set(key, value);
            string listKey = "test-aliyun-list";
            System.Console.WriteLine("set key " + key + " value " + value);
            string getValue = System.Text.Encoding.Default.GetString(redisClient.Get(key));
            System.Console.WriteLine("get key " + getValue);
            System.Console.Read();
        }
        public static void RedisPoolClientTest()
        {
            string[] testReadWriteHosts = new[] {
                "redis://password@127.0.0.1:6379"/*redis://密码@访问地址:端口*/
            };
            RedisConfig.VerifyMasterConnections = false;/*需要设置
            PooledRedisClientManager redisPoolManager = new PooledRedisClientManager(10/*连接池个数*/, 10/*
            连接池超时时间*/, testReadWriteHosts);
            for (int i = 0; i < 100; i++){
                IRedisClient redisClient = redisPoolManager.GetClient();/*获取连接
                RedisNativeClient redisNativeClient = (RedisNativeClient)redisClient;
                redisNativeClient.Client = null;/*ApsaraDB for Redis不支持client setname所以这里需要显示的把client
```


```
对象置为null
try
{
    string key = "test-aliyun1111";
    string value = "test-aliyun-value1111";
    redisClient.Set(key, value);
    string listKey = "test-aliyun-list";
    redisClient.AddItemToList(listKey, value);
    System.Console.WriteLine("set key " + key + " value " + value);
    string getValue = redisClient.GetValue(key);
    System.Console.WriteLine("get key " + getValue);
    redisClient.Dispose();//
}catch (Exception e)
{
    System.Console.WriteLine(e.Message);
}
}
System.Console.Read();
}
static void Main(string[] args)
{
    //单链接模式
    RedisClientTest();
    //连接池模式
    RedisPoolClientTest();
}
}
```

node-redis客户端

1. 下载并安装node-redis。

```
npm install hiredis redis
```

2. 在node-redis客户端中输入下述代码，然后根据注释提示修改代码。

 说明 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。

```
var redis = require("redis"),
client = redis.createClient(<port>, <"host">, {detect_buffers: true});
client.auth("password", redis.print)
```

参数说明：

- <port>：Redis实例的服务端口，默认为6379。
- <"host">：Redis实例的连接地址。

配置示例：

```
var redis = require("redis"),
    client = redis.createClient(6379, "r-abcdefg.redis.rds.aliyuncs.com", {detect_buffers: true});
client.auth("password", redis.print)
```

3. 执行上述代码完成连接。
4. 使用云数据库Redis。

```
// 写入数据
client.set("key", "OK");
// 获取数据，返回String
client.get("key", function (err, reply) {
  console.log(reply.toString()); // print `OK`
});
// 如果传入一个Buffer，返回也是一个Buffer
client.get(new Buffer("key"), function (err, reply) {
  console.log(reply.toString()); // print `<Buffer 4f 4b>`
});
client.quit();
```

C#客户端StackExchange.Redis

 **警告** 如果您的Redis实例为**集群架构**或**读写分离架构**，且需要执行切换或选择数据库的操作（即使用多数据库功能），您必须先将cluster_compat_enable参数设置为0（即关闭原生Redis Cluster语法兼容），否则将提示报错：`RedisCommandException: Multiple databases are not supported on this server; cannot switch to database: 1`。具体操作，请参见[参数设置及说明](#)。

1. 下载并安装StackExchange.Redis。
2. 添加引用。

```
using StackExchange.Redis;
```

3. 初始化ConnectionMultiplexer。

ConnectionMultiplexer是StackExchange.Redis的核心，它被整个应用程序共享和重用，应该设置为单例，它的初始化如下：

🔍 说明

- 关于如何获取Redis实例的连接地址和密码，请参见[如何获取连接信息](#)。
- ConfigurationOptions包含很多选项，例如keepAlive、connectRetry、name，详情请参见[ConfigurationOptions](#)。

```
// redis config
private static ConfigurationOptions configurationOptions = ConfigurationOptions.Parse("127.0.0.1:6379,password=xxx,connectTimeout=2000");
//the lock for singleton
private static readonly object Locker = new object();
//singleton
private static ConnectionMultiplexer redisConn;
//singleton
public static ConnectionMultiplexer getRedisConn()
{
    if (redisConn == null)
    {
        lock (Locker)
        {
            if (redisConn == null || !redisConn.IsConnected)
            {
                redisConn = ConnectionMultiplexer.Connect(configurationOptions);
            }
        }
    }
    return redisConn;
}
```

4. 由于GetDatabase()返回的对象是轻量级的，每次用的时候从ConnectionMultiplexer对象中获取即可。

```
redisConn = getRedisConn();
var db = redisConn.GetDatabase();
```

5. 通过客户端程序操作Redis数据库。

🔍 说明 下文列出常见数据接口的demo示例，和原生API略有不同。

- String

```
//set get
string strKey = "hello";
string strValue = "world";
bool setResult = db.StringSet(strKey, strValue);
Console.WriteLine("set " + strKey + " " + strValue + ", result is " + setResult);
//incr
string counterKey = "counter";
long counterValue = db.StringIncrement(counterKey);
Console.WriteLine("incr " + counterKey + ", result is " + counterValue);
//expire
db.KeyExpire(strKey, new TimeSpan(0, 0, 5));
Thread.Sleep(5 * 1000);
Console.WriteLine("expire " + strKey + ", after 5 seconds, value is " + db.StringGet(strKey));
//mset mget
KeyValuePair<RedisKey, RedisValue> kv1 = new KeyValuePair<RedisKey, RedisValue>("key1", "value1");
KeyValuePair<RedisKey, RedisValue> kv2 = new KeyValuePair<RedisKey, RedisValue>("key2", "value2");
db.StringSet(new KeyValuePair<RedisKey, RedisValue>[] {kv1,kv2});
RedisValue[] values = db.StringGet(new RedisKey[] {kv1.Key, kv2.Key});
Console.WriteLine("mget " + kv1.Key.ToString() + " " + kv2.Key.ToString() + ", result is " + values[0] + "
&&" + values[1]);
```

- Hash

```
string hashKey = "myhash";
//hset
db.HashSet(hashKey, "f1", "v1");
db.HashSet(hashKey, "f2", "v2");
HashEntry[] values = db.HashGetAll(hashKey);
//hgetall
Console.Write("hgetall " + hashKey + ", result is");
for (int i = 0; i < values.Length; i++)
{
    HashEntry hashEntry = values[i];
    Console.Write(" " + hashEntry.Name.ToString() + " " + hashEntry.Value.ToString());
}
Console.WriteLine();
```

- List

```
//list key
string listKey = "myList";
//rpush
db.ListRightPush(listKey, "a");
db.ListRightPush(listKey, "b");
db.ListRightPush(listKey, "c");
//lrange
RedisValue[] values = db.ListRange(listKey, 0, -1);
Console.WriteLine("lrange " + listKey + " 0 -1, result is ");
for (int i = 0; i < values.Length; i++)
{
    Console.WriteLine(values[i] + " ");
}
Console.WriteLine();
```

- o Set

```
//set key
string setKey = "mySet";
//sadd
db.SetAdd(setKey, "a");
db.SetAdd(setKey, "b");
db.SetAdd(setKey, "c");
//sismember
bool isContains = db.SetContains(setKey, "a");
Console.WriteLine("set " + setKey + " contains a is " + isContains );
```

- o Sorted Set


```

string sortedSetKey = "myZset";
//sadd
db.SortedSetAdd(sortedSetKey, "xiaoming", 85);
db.SortedSetAdd(sortedSetKey, "xiaohong", 100);
db.SortedSetAdd(sortedSetKey, "xiaofei", 62);
db.SortedSetAdd(sortedSetKey, "xiaotang", 73);
//zrevrangebyscore
RedisValue[] names = db.SortedSetRangeByRank(sortedSetKey, 0, 2, Order.Ascending);
Console.WriteLine("zrevrangebyscore " + sortedSetKey + " 0 2, result is ");
for (int i = 0; i < names.Length; i++)
{
    Console.WriteLine(names[i] + " ");
}
Console.WriteLine();

```

相关文档

- [Redis连接问题排查与解决](#)
- [报警设置](#)
- [查看监控数据](#)

4.3. 通过redis-cli连接Redis

redis-cli是原生Redis自带的命令行工具，您可以在ECS实例或本地设备上通过redis-cli连接云数据库Redis，进行数据管理。

连接Redis使用的网络类型

网络类型	说明
专有网络（推荐）	专有网络VPC（Virtual Private Cloud）是您自己独有的云上私有网络，不同的专有网络之间二层逻辑隔离，拥有较高的安全性和性能。redis-cli部署在ECS实例上时，通过专有网络连接至Redis实例，可获得更高的安全性和更低的网络延迟。
公网	<p>公网即互联网，当本地设备需要测试或管理Redis实例时，可在本地设备上部署redis-cli，然后通过公网连接至Redis实例。</p> <p> 说明 通过公网连接不会产生流量费用，但存在一定的安全风险，推荐通过专有网络连接以获取更高的安全性。</p>

redis-cli说明

- 由于redis-cli是原生Redis自带的命令行工具，通常您需要通过在ECS实例或本地设备上安装Redis服务的方式，完成redis-cli的安装。

说明 除redis-cli以外，您可以通过[数据管理DMS \(Data Management\)](#)免客户端快速连接Redis数据库，帮助您可视化^{管理}Redis数据库。具体操作，请参见[通过DMS登录Redis](#)。

- redis-cli的版本可以与Redis实例的大版本不同。

redis-cli连接方法 (Linux系统)

- 安装redis-cli (如已安装，可跳过)。
 - 登录待安装redis-cli的设备，例如[ECS实例](#)或本地设备。
 - 执行下述命令下载Redis源码文件：

```
wget https://download.redis.io/releases/redis-6.0.9.tar.gz
```

说明 本文以redis-6.0.9版本为例演示操作流程，您也可以安装其他版本。具体操作，请参见[Redis官网](#)。

- 执行下述命令解压Redis源码文件：

```
tar xzf redis-6.0.9.tar.gz
```

- 执行下述命令进入解压后的目录并编译安装Redis源码文件：

```
cd redis-6.0.9&&make
```

说明 编译安装需要一段时间 (通常2分钟~3分钟)。

- 根据redis-cli的安装位置，完成下述操作：

redis-cli安装位置	需完成的操作
ECS实例 (推荐)	<ol style="list-style-type: none"> 确保ECS实例与Redis实例属于同一专有网络 (即实例基本信息中的专有网络ID一致)。 <div data-bbox="544 1451 1385 1675" data-label="Text"> <p>说明</p> <ul style="list-style-type: none"> 如果专有网络不同，您可以更换ECS实例所属的专有网络。具体操作，请参见更换ECS实例的VPC。 如果网络类型不同，例如ECS实例为经典网络，Redis实例为专有网络。如何连接，请参见ECS实例与Redis实例的网络类型不同时如何连接。 </div> 获取ECS实例的内网IP地址。具体操作，请参见查询ECS实例的IP地址。 将ECS实例的内网IP地址添加至Redis实例的白名单中。具体操作，请参见设置IP白名单。

redis-cli安装位置	需完成的操作
本地	<p>i. Redis实例默认仅提供内网连接地址，通过公网连接时您需要手动申请公网连接地址。具体操作，请参见申请公网连接地址。</p> <p>ii. 在本地设备上执行<code>curl ipinfo.io grep ip</code>命令，获取其公网IP地址，返回示例如下：</p> <pre>root@:~# curl ipinfo.io grep ip % Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 100 249 100 249 0 0 1272 0 --:--:-- --:--:-- --:--:-- 1270 "ip": ".203", "readme": "https://ipinfo.io/missingauth"</pre> <p>iii. 将本地设备的公网IP地址添加至Redis实例的白名单中。具体操作，请参见设置IP白名单。</p>

3. 获取连接信息并执行下述命令连接Redis实例：

```
src/redis-cli -h <hostname> -p <port> [-c]
```

参数说明

参数	说明	获取方式
<hostname>	Redis实例的连接地址	<ul style="list-style-type: none"> ECS实例通过专有网络连接：需获取Redis实例的专有网络连接地址。 本地设备通过公网连接：需获取Redis实例的公网连接地址。 具体操作，请参见 查看连接地址 。
<port>	Redis实例的端口号	端口号默认为6379，您也可以自定义端口号。具体操作，请参见 修改连接端口 。
-c	启用集群模式	只有当Redis实例为 集群架构 ，且需要使用 直连地址 连接时，才需要加上-c。

默认地址连接示例（适用于通过默认地址连接的场景，例如[标准架构](#)实例的连接地址或[集群架构](#)实例的代理地址）：

```
src/redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379
```

直连地址连接示例（适用于通过[直连地址](#)连接的场景）：

```
src/redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379 -c
```

4. 执行下述命令完成密码验证：

```
AUTH <password>
```

 **说明** 如果实例已开启[专有网络免密访问](#)，通过专有网络连接Redis实例时无需执行本步骤即可正常执行Redis命令。

<password>：账号的密码，根据选取账号的不同，密码的填写格式有一定区别。如果忘记密码，您可以重置密码。具体操作，请参见[修改或重置密码](#)。

- 默认账号（即以实例ID命名的账号）：直接填写密码即可。
- 新创建的账号：密码格式为 <user>:<password>。例如自定义账号为 `testaccount`，密码为 `Rp829dlwa`，密码需填写为 `testaccount:Rp829dlwa`。

示例：

```
AUTH testaccount:Rp829dlwa
```

验证成功后，返回如下结果：

```
OK
```

说明 如果出现报错，请参见[常见连接报错](#)。

5. 完成验证后，即可执行Redis命令，例如执行DBSIZE命令查询当前库有多少个键（Key）。关于云数据库Redis支持的命令，请参见[Redis命令支持概览](#)；关于所有Redis命令的介绍和使用方法，请参见[Redis官网](#)。

redis-cli连接方法（Windows系统）

说明 目前仅支持64位的Windows系统。

1. 安装redis-cli（如已安装，可跳过）。
 - i. 登录待安装redis-cli的设备，例如[ECS实例](#)或本地设备。
 - ii. 下载[Redis-x64-3.2.100.zip](#)文件。

说明 本文以redis-3.2版本（最新版本）为例演示操作流程，您也可以安装其他版本。更多信息，请参见[Microsoft Archive-Redis](#)。

- iii. 将 `Redis-x64-3.2.100.zip`解压至待安装的目录，本案例将其解压至 `D:\Redis-x64-3.2.100` 目录中。

2. 根据redis-cli的安装位置，完成下述操作：

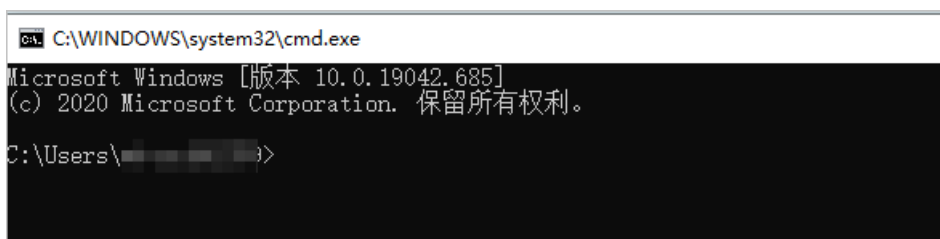
redis-cli安装位置	需完成的操作
ECS实例 （推荐）	<ol style="list-style-type: none"> i. 确保ECS实例与Redis实例属于同一专有网络（即实例基本信息中的专有网络ID一致）。 <p> 说明 如果专有网络不同，您需要更换ECS实例所属的专有网络。具体操作，请参见更换ECS实例的VPC。</p> ii. 获取ECS实例的内网IP地址。具体操作，请参见查询ECS实例的IP地址。 iii. 将ECS实例的内网IP地址添加至Redis实例的白名单中。具体操作，请参见设置IP白名单。

redis-cli安装位置	需完成的操作
本地	<ul style="list-style-type: none"> i. Redis实例默认仅提供内网连接地址，通过公网连接时您需要手动申请公网连接地址。具体操作，请参见申请公网连接地址。 ii. 在本地设备上访问ipinfo获取其公网地址。 iii. 将本地设备的公网IP地址添加至Redis实例的白名单中。具体操作，请参见设置IP白名单。

3. 在安装了redis-cli的ECS实例或本地设备上，使用Windows键+R组合键打开运行对话框，然后输入cmd并单击确定。

系统将为您打开Windows命令行窗口。

Windows命令行窗口



4. 执行下述命令，进入redis-cli所属的目录。

```
cd /d <path>
```

<path>: Redis-x64-3.2.100.zip解压后的完整路径，本案例为D:\Redis-x64-3.2.100。

示例：

```
cd /d D:\Redis-x64-3.2.100
```

5. 获取连接信息并执行下述命令连接Redis实例：

```
redis-cli -h <hostname> -p <port> [-c]
```

参数说明

参数	说明	获取方式
<hostname> >	Redis实例的连接地址	<ul style="list-style-type: none"> ◦ ECS实例通过专有网络连接：需获取Redis实例的专有网络连接地址。 ◦ 本地设备通过公网连接：需获取Redis实例的公网连接地址。 具体操作，请参见 查看连接地址 。
<port>	Redis实例的端口号	端口号默认为6379，您也可以自定义端口号。具体操作，请参见 修改连接端口 。
-c	启用集群模式	只有当Redis实例为 集群架构 ，且需要使用 直连地址 连接时，才需要加上-c。

参数	说明	获取方式

默认地址连接示例（适用于通过默认地址连接的场景，例如[标准架构](#)实例的连接地址[集群架构](#)实例的代理地址）：

```
redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379
```

直连地址连接示例（适用于通过[直连地址](#)连接的场景）：

```
redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379 -c
```

6. 执行下述命令完成密码验证：

```
AUTH <password>
```

 **说明** 如果实例已开启[专有网络免密访问](#)，通过专有网络连接Redis实例时无需执行本步骤即可正常执行Redis命令。

<password>：账号的密码，根据选取账号的不同，密码的填写格式有一定区别。如果忘记密码，您可以重置密码。具体操作，请参见[修改或重置密码](#)。


- 默认账号（即以实例ID命名的账号）：直接填写密码即可。
- 新创建的账号：密码格式为 <user>:<password>。例如自定义账号为 `testaccount`，密码为 `Rp829dlwa`，密码需填写为 `testaccount:Rp829dlwa`。

示例：

```
AUTH testaccount:Rp829dlwa
```


验证成功后，返回如下结果：

```
OK
```

 **说明** 如果出现报错，请参见[常见连接报错](#)。

7. 完成验证后，即可执行Redis命令，例如执行[DBSIZE](#)命令查询当前库有多少个键（Key）。关于云数据库Redis支持的命令，请参见[Redis命令支持概览](#)；关于所有Redis命令的介绍和使用方法，请参见[Redis官网](#)。

常见连接报错

 **说明** 如果实例的小版本过低，报错信息可能不准确，建议将实例升级至最新的小版本。具体操作，请参见[升级小版本](#)。

报错信息	原因及解决方法
(error) ERR illegal address	未设置正确的白名单，具体操作，请参见 redis-cli连接方法（Linux系统） 或 redis-cli连接方法（Windows系统） 中的步骤2。
(error) ERR client ip is not in whitelist	
(error) ERR invalid password	<p>密码错误，请确认使用正确的密码和密码格式。根据选取账号的不同，密码格式有一定区别。</p> <ul style="list-style-type: none"> 默认账号（即以实例ID命名的账号）：直接填写密码即可。 新创建的账号：密码格式为 <code><user>:<password></code>。例如自定义账号为 <code>testaccount</code>，密码为 <code>Rp829dlwa</code>，密码需填写为 <code>testaccount:Rp829dlwa</code>。 <p>? 说明 如果忘记密码，您可以重置密码。具体操作，请参见修改或重置密码。</p>

4.4. 如何通过公网连接Redis实例

当本地设备需要测试或管理Redis实例时，您可以为Redis实例申请公网地址，然后在本地设备上通过公网连接至Redis实例。

注意事项

通过公网连接不会产生流量费用，但是会增加网络延迟，影响Redis服务的性能，且存在一定的安全风险。推荐通过[专有网络连接](#)，可获得更低的网络延迟和更高的安全性。

操作步骤

1. 根据本地设备的操作系统，选择下述方法获取本地设备对应的公网地址：

o Linux操作系统

在本地设备上执行`curl ipinfo.io |grep ip`命令，获取其公网IP地址，返回示例如下：

```
root@:~# curl ipinfo.io |grep ip
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  249  100  249    0     0  1272    0  --:--:-- --:--:-- --:--:-- 1270
"ip": "192.168.1.203",
"readme": "https://ipinfo.io/missingauth"
```

o Windows操作系统

在本地设备上访问[ipinfo](#)，获取其公网地址。

2. 将本地设备的公网IP地址添加至Redis实例的白名单中。具体操作，请参见[设置IP白名单](#)。

3. 为Redis实例申请公网连接地址，具体操作，请参见[申请公网连接地址](#)。

申请到的公网地址与端口



- 通过获取到的公网连接地址，使用redis-cli或Redis客户端程序连接Redis实例。
 - 通过redis-cli连接Redis
 - 通过客户端程序连接Redis

相关文档

- Redis连接问题排查与解决

4.5. Sentinel兼容模式连接

云数据库Redis提供Sentinel（哨兵）兼容模式，开启后客户端可以像连接原生Redis Sentinel一样连接Redis实例。

前提条件

- Redis实例的架构为标准版。
- Redis实例的引擎版本为4.0（社区版）或5.0。
- 已开启Sentinel兼容模式，详情请参见[开启Sentinel兼容](#)。
- 已将客户端的IP地址（ECS实例的内网IP地址或本地主机的外网IP地址）加入[Redis白名单](#)。

Redis Sentinel简介

Redis Sentinel为开源Redis提供主从实例监控、故障告警、自动故障切换等服务，很多使用本地自建Redis数据库并且对可靠性要求较高的业务场景都用到了Sentinel。为了给这类场景中的Redis数据库迁移上云提供方便，阿里云开发了Sentinel兼容模式。


? **说明** 阿里云云数据库Redis版使用自研的**高可用服务HA**组件，无需Sentinel。

开启Sentinel兼容模式后，您可以使用如下的Sentinel相关命令：

命令	说明
SENTINEL sentinels	查询指定master的Sentinel实例列表以及这些Sentinel实例的状态。使用方式： <pre>SENTINEL sentinels <master名称></pre>
SENTINEL get-master-addr-by-name	查询指定master的IP地址和端口号。使用方式： <pre>SENTINEL get-master-addr-by-name <master名称></pre>

? **说明** 以上命令不支持2.8版本的云数据库Redis实例。

Sentinel免密连接示例

 **说明** 以Sentinel模式免密连接Redis实例时，Redis实例需为专有网络且需要开启免密访问，详情请参见[切换为专有网络](#)和[开启专有网络免密访问](#)。

本示例以spring-data-redis为例，开源代码配置如下：

```
@Bean
public JedisConnectionFactory connectionFactory() {
    RedisSentinelConfiguration sentinelConfig = new RedisSentinelConfiguration()
        .master("original-master-name")
        .sentinel(original-sentinel-1-host, original-sentinel-1-port)
        .sentinel(original-sentinel-2-host, original-sentinel-2-port);
    JedisPoolConfig poolConfig = new JedisPoolConfig();
    ...
    JedisConnectionFactory connectionFactory = new JedisConnectionFactory(sentinelConfig, poolConfig);
    return connectionFactory;
}
```

参数说明：

- master-name：可自定义一个名称。
- sentinel-host：设置为Redis实例的专有网络连接地址。
- sentinel-port：设置为Redis实例的端口号，默认端口号为6379。

阿里云Redis Sentinel兼容模式连接代码配置示例：

```
@Bean
public JedisConnectionFactory connectionFactory() {
    RedisSentinelConfiguration sentinelConfig = new RedisSentinelConfiguration()
        .master("any-name")
        .sentinel("r-*****.redis.rds.aliyuncs.com", 6379);
    JedisPoolConfig poolConfig = new JedisPoolConfig();
    ...
    JedisConnectionFactory connectionFactory = new JedisConnectionFactory(sentinelConfig, poolConfig);
    return connectionFactory;
}
```


Sentinel密码连接示例

本示例以Jedis为例，开源代码配置如下：

```
String masterName = "original-master-name";
Set<String> sentinels = new HashSet<>();
sentinels.add("original-sentinel-1-host:original-sentinel-1-port");
sentinels.add("original-sentinel-2-host:original-sentinel-2-port");
GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
String dbPassword = "original-db-password";
String sentinelPassword = "original-sentinel-password";
JedisSentinelPool jedisSentinelPool =
    new JedisSentinelPool(masterName, sentinels, poolConfig,
        2000, 2000, dbPassword,
        0, null, 2000, 2000,
        sentinelPassword, null);
```

参数说明：

- masterName：可自定义一个名称。
- sentinels.add：设置为Redis实例的专有网络连接地址和端口号，格式为 `r-*****.redis.rds.aliyuncs.com:6379`。
- dbPassword/sentinelPassword：设置为Redis实例的密码。如果忘记密码，您可以通过控制台重置密码，详情请参见[修改或重置密码](#)。

 说明 如果使用自定义账号连接Redis，连接密码格式需为 `<user>:<password>`。例如，自定义账号的用户名为admin，密码为Rp829dlwa，则连接Redis时的密码为 `admin:Rp829dlwa`。

阿里云Redis Sentinel兼容模式连接代码配置示例：

```
String masterName = "any-name";
Set<String> sentinels = new HashSet<>();
sentinels.add("r-*****.redis.rds.aliyuncs.com:6379");
GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
String dbPassword = "admin:Rp829dlwa";
String sentinelPassword = "admin:Rp829dlwa";
JedisSentinelPool jedisSentinelPool =
    new JedisSentinelPool(masterName, sentinels, poolConfig,
        2000, 2000, dbPassword,
        0, null, 2000, 2000,
        sentinelPassword, null);
```

5.使用须知

5.1. Redis管理控制台

Redis管理控制台是用于管理Redis实例的Web应用程序，管理Redis实例所需的基础操作都可以在控制台上完成。您可以通过本文档了解控制台的使用方法。

Redis管理控制台是阿里云管理控制台的一部分，关于控制台的通用设置和基本操作请参见[使用阿里云管理控制台](#)。本文将介绍Redis控制台的通用界面，若有差异，请以控制台实际界面为准。

控制台首页

登录[Redis管理控制台](#)即进入控制台首页。对于Redis所有类型的实例而言，控制台首页的界面信息都是相同的。

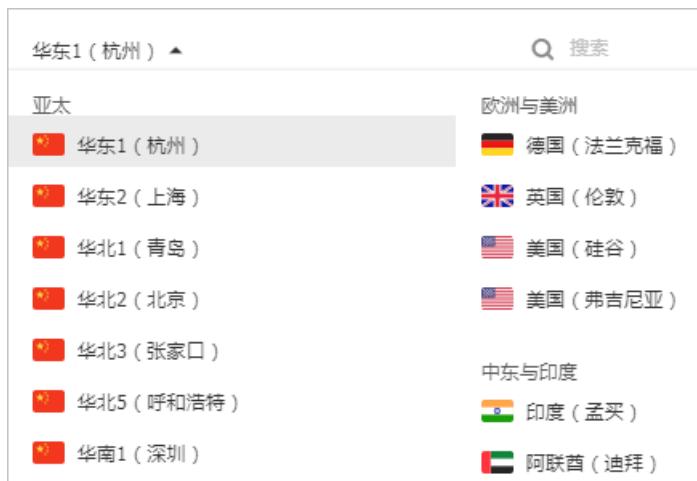
 说明 若没有阿里云账号，请先[注册](#)。

Redis管理控制台首页



控制台首页主要功能区介绍如下。

- 区域1为地域选择菜单，单击或光标悬浮于该区域即展示区域列表。单击目标地域即可切换到该地域的实例列表。



- 区域2为首页左侧导航栏。登录Redis管理控制台后默认进入实例列表页。
 - 实例列表中展示本地域的实例资源列表。详情请参见下文对区域3的说明。
- 区域3为实例列表页。实例列表页面中会展示实例ID、状态、已用内存及配额、可用区、版本、实例规格、创建时间、付费方式、网络类型等信息。

 说明

已用内存及配额是由底层系统根据采集信息进行离线汇总得到的结果，存在10分钟左右的延迟，与当前时间的实际值可能存在差别。

如果需要查看实时信息，建议登录DMS进行查看，详细步骤请参见[通过DMS登录Redis](#)。

控制台常用功能导航

- [性能监控](#)
- [报警设置](#)
- [白名单设置](#)
- [参数设置](#)
- [账号管理](#)
- [备份与恢复](#)
- [日志管理](#)

5.2. 使用限制

您可以在本文中查看云数据库Redis版各数据类型以及部分功能特性的使用限制。

项目	说明
List数据类型	没有List个数限制，单个元素最大值为512 MB，推荐list的元素个数小于8192，value最大长度不超过1 MB。
Set数据类型	没有set个数限制，单个元素最大值为512 MB，推荐set的元素个数小于8192，value最大长度不超过 1 MB。
Sorted set数据类型	没有sorted set个数限制，单个元素最大值为512 MB，推荐sorted set的元素个数小8192，value最大长度不超过1 MB。
Hash数据类型	没有field个数限制，单个元素最大值为512 MB，推荐元素个数小于8192，value最大长度不超过1 MB。
DB数限制	<p>每个实例支持256个DB。</p> <div data-bbox="555 1518 1385 1729" style="background-color: #e1f5fe; padding: 10px;"> <p> 说明</p> <ul style="list-style-type: none"> • 所有DB存储的数据总量受限于实例的内存大小； • 单个DB占用内存按照使用情况自动分配，上限为实例内存（例如DB0占用全部内存而其它DB无数据）。 </div>
Redis命令支持	详情请参见 Redis命令支持概览 。

项目	说明
监控报警	云数据库Redis版未提供容量告警，需要用户到云监控中进行配置。配置方法请参见 报警设置 。 建议设置好以下监控的报警：实例故障、实例主备切换、已使用连接百分比、操作失败数、已用容量百分比、写入带宽使用率、读取带宽使用率。
数据过期删除策略	<ul style="list-style-type: none"> 主动过期，系统后台会周期性的检测，发现已过期的key时，会将其删除。 被动过期，当用户访问某个key时，如果该key已经过期，则将其删除。
空闲连接回收机制	服务端不主动回收Redis空闲连接，由用户管理。
数据持久化策略	采用 AOF_FSYNC_EVERYSEC 方式，每秒fsync。

5.3. 命令查询导航

云数据库Redis存在多个版本、系列和架构，各种类型的实例对于Redis命令的支持度有所不同。根据本章节的导航信息，您可以快速找到云数据库Redis各版本支持的命令和限制使用的命令。

通用命令说明

 **说明** 通用命令说明同时适用于社区版和企业版。

文档标题	简介
Redis 2.8命令支持	本章节介绍引擎版本（版本号）为2.8的云数据库Redis实例支持的命令，包括Redis 2.8标准版、Redis 2.8集群版、和Redis 2.8读写分离版。2.8版本的云Redis实例兼容原生Redis 3.0版本的命令。
Redis 4.0命令支持	本章节介绍引擎版本（版本号）为4.0的云数据库Redis实例支持的命令，包括Redis 4.0标准版、Redis 4.0集群版、和Redis 4.0读写分离版。
未开放的命令	本章节介绍云数据库Redis暂未开放的原生Redis命令。
集群架构实例的命令限制	由于部署架构的不同，云数据库Redis集群版相比标准版在原生Redis命令的支持上有一定的区别，本章节介绍云Redis集群版中的命令使用限制。
阿里云自研的Redis命令	在兼容原生Redis命令之外，云数据库Redis还支持多个自研Redis命令，可以在云Redis集群版或读写分离版中使用，帮助您更方便地管理实例。


企业版命令说明

文档标题	简介
Redis企业版命令限制	Redis企业版兼容大多数的原生Redis命令，为保障服务性能，部分命令的使用受到限制。

文档标题	简介
企业版（性能增强型）支持的新命令	<p>Redis企业版（性能增强型）除支持云Redis社区版的命令以外，还支持下述新的命令：</p> <ul style="list-style-type: none"> • CAS和CAD命令 • TairString命令 • TairHash命令 • TairGIS命令 • TairBloom命令 • TairDoc命令

5.4. Redis proxy特性说明

在云数据库Redis集群版和读写分离版的架构中，代理服务器（proxy）承担着路由转发、负载均衡与故障转移等职责。本节介绍proxy的路由转发规则以及proxy处理只读节点异常和特殊命令方法。


 **说明** 由于proxy的演进，集群中proxy的绝对个数并不完全代表proxy处理能力，阿里云会保证集群规格中proxy的配比符合规格说明的要求。

Redis集群版中proxy的路由转发规则

- 基础转发规则
 - 对于需要操作单个key的命令，proxy会根据key所属的slot将请求发送给储存着该slot的数据分片。
 - 对于需要操作两个或以上key的命令，如果这些key是储存在不同分片中的，proxy会将命令拆分成多个命令分别发送给对应的分片。该类命令的详细信息请参见[Redis命令支持概览](#)。

- 发布订阅命令转发规则

proxy会将PUBLISH、SUBSCRIBE等发布订阅命令发送到集群中第一个数据分片的master节点。

 **警告** 请合理使用发布订阅命令，防止给第一个数据分片的master节点带来过大的性能压力。

- 指定了idx参数的命令的转发规则

在使用阿里云自研的集群命令，例如IINFO、ISCAN、IMONITOR、IMEMORY时，如果通过idx参数指定了分片ID，proxy会将这些命令发送到指定的数据分片。该类命令的详细信息请参见[Redis命令支持概览](#)。

Redis读写分离版中proxy的路由转发规则

- 基础转发规则
 - proxy将写命令直接转发到master节点。
 - proxy将读命令根据权重发送到master节点和只读节点，所有节点的权重相同。例如，如果实例有5个只读节点，proxy会将所有读命令平均发送到包括master节点和只读节点在内的6个节点，每个节点的比重都是1/6。

 **说明** SLOWLOG和DBSIZE也属于读命令。

- 发送到master节点

proxy会将以下命令发送到master节点：

- 事务命令MULTI和EXEC。
 - Lua脚本命令EVAL和EVALSHA。
 - SCAN和INFO。
 - 发布订阅相关命令，包括PUBLISH、SUBSCRIBE等。
- 发送到第一个只读节点

proxy会将HSCAN、SSCAN、ZSCAN命令发送到第一个只读节点。在没有正常运行的只读节点时，这些命令会被发送到master节点。

- 发送到指定只读节点

proxy会根据读写分离版专属命令RIINFO、RIMONITOR、RIMEMORY中idx参数和ro_slave_idx参数的设置，将命令发送到相应的只读节点。其中idx用于在集群版读写分离实例中指定数据分片，ro_slave_idx用于在任意读写分离实例中指定只读节点。读写分离版专属命令的详细说明请参见[Redis命令支持概览](#)。

proxy下线只读节点

proxy实时感知每个只读节点的状态，在以下情况中，proxy将暂时下线相应的只读节点。

- 探测到只读节点异常时，proxy会自动降低该节点的服务权重。如果该只读节点连续连接失败超过一定次数，proxy会停止该节点的服务，待异常被修复后重新启用该节点。
- 探测到只读节点处于全量同步过程中时，proxy会暂时下线该节点，直到同步完成。

proxy对特殊命令的处理方式

一般情况下，proxy与后端数据分片会创建长连接处理用户请求。但是，如果请求中包含以下命令，proxy会根据命令的处理需求，在相应的数据分片创建额外的连接来处理后续请求。

- 阻塞类命令：BRPOP、BRPOPLPUSH、BLPOP、BZPOPMAX、BZPOPMIN。
- 事务类命令：MULTI、EXEC、WATCH。
- MONITOR类命令：MONITOR、IMONITOR、RIMONITOR。
- 订阅命令：SUBSCRIBE、UNSUBSCRIBE、PSUBSCRIBE、PUNSUBSCRIBE。

② 说明

- 每个数据分片的连接数上限为10000，请合理使用上述命令，避免连接数超限。
- 同时在多个db进行SCAN操作也会创建大量新连接，请注意避免连接数超限。