

ALIBABA CLOUD

# 阿里云

云数据库 Redis 版  
快速入门

文档版本：20201124

 阿里云

## 法律声明

阿里云提醒您在使用或阅读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。
<b>粗体</b>	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	16
4.步骤3：连接实例	20
4.1. DMS登录云数据库	20
4.2. Redis客户端连接	23
4.3. redis-cli连接	38
4.4. 公网连接	40
4.5. Sentinel兼容模式连接	41
5.使用须知	45
5.1. Redis管理控制台	45
5.2. 使用限制	46
5.3. 命令查询导航	47
5.4. Redis proxy特性说明	48

# 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、持久内存型、容量存储型、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，请确保您的账户余额大于等于100元。

### 优惠活动

首购折扣：活动期间，首次购买阿里云数据库Redis版可享受低至2折优惠，详情请参见[云数据库优惠活动](#)。

### 费用

详情请参见[收费项与价格](#)。


### 选型参考资料

创建Redis实例前，您需要结合产品性能、价格、业务场景、工作负载等因素，做出性价比与稳定性最优的决策。下文将围绕以上因素，着重介绍实例类型、引擎版本、架构、存储介质，为您的选型提供相关参考。

- 实例类型
  - Redis社区版：兼容开源Redis的高性能数据缓存服务，适用于标准的、无特殊业务需求的Redis使用场景。
  - **Redis企业版**：在Redis社区版的基础上开发的强化版Redis服务，从访问延时、持久化需求、整体成本这三个核心维度考量，基于DRAM、NVM和**ESSD云盘**等存储介质，推出了多种不同形态的产品，为您提供更强的性能、更多的数据结构和更灵活的存储方式，满足不同场景下的业务需求。

 **说明** 更多相关资料，请参见[Redis企业版与社区版特性对比](#)、[Redis社区版命令支持](#)和[Redis企业版命令限制](#)。

- 引擎版本：推荐使用更新的引擎版本以支持更多的特性，详情请参见[Redis 6.0新特性说明](#)、[Redis 5.0新特性说明](#)和[Redis 4.0新功能介绍](#)。

 **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。

- 架构：云数据库Redis支持下述架构，您需要根据业务需求选择：
  - 集群架构可轻松突破Redis自身单线程瓶颈，满足大容量、高性能的业务需求。
  - 标准架构采用主从架构，提供高性能的缓存服务和数据高可靠。
  - 读写分离架构提供高可用、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。

更多详情，请参见[架构信息查询导航](#)。

- 存储介质：数据读写和持久化的介质较大程度上影响实例的性能，通常来说，性能排序为内存>Intel傲腾™持久化内存>ESSD云盘>本地盘。

实例的创建方式和对应的约束请参见下表。

 **说明** 关于本地盘实例和云盘实例的区别，请参见[本地盘和云盘实例对比](#)。

创建方法	支持的实例类型	支持的引擎版本	支持的架构	存储介质	
				数据读写介质	持久化介质
创建Redis本地盘实例	Redis社区版 <sup>®</sup>	2.8 4.0 5.0	集群架构 标准架构 读写分离架构	内存	本地盘
	企业版（性能增强型）	5.0	集群架构 标准架构 读写分离架构	内存	本地盘
	企业版（混合存储型）	4.0	集群架构 标准架构	内存和本地盘	本地盘
创建Redis云盘实例	Redis社区版	5.0 6.0	集群架构 标准架构	内存	ESSD云盘
	企业版（性能增强型）	5.0	标准架构	内存	ESSD云盘
创建持久内存型或容量存储型实例	企业版（持久内存型）	1.0（自研，兼容Redis 5.0）	标准架构	Intel傲腾™持久化内存	
	企业版（容量存储型）	1.0（自研，兼容Redis 5.0）	标准架构	ESSD云盘	
创建数据库专属集群	Redis社区版	5.0	标准架构	内存	ESSD云盘
	企业版（性能增强型）	5.0	标准架构	内存	ESSD云盘

## 创建Redis本地盘实例

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，选择商品类型。
  - **包年包月（本地盘）**：在新建本地盘版实例时支付费用。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（本地盘）**：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
基本配置	地域和可用区	实例所在的地理位置。购买后无法更换地域。 <ul style="list-style-type: none"> <li>◦ 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。</li> <li>◦ 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。</li> </ul>
	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。 <p> <b>说明</b> 选择多可用区可实现实例的同城容灾。</p>
	网络类型	<ul style="list-style-type: none"> <li>◦ <b>专有网络（推荐）</b>：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。</li> <li>◦ <b>经典网络</b>：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。</li> </ul> <p> <b>注意</b></p> <ul style="list-style-type: none"> <li>◦ 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。</li> <li>◦ 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。</li> <li>◦ 经典网络中的Redis实例可以<b>切换到专有网络</b>，专有网络中的实例无法切换到经典网络。</li> </ul>
	专有网络	选择实例的专有网络。如果没有专有网络，请参见 <a href="#">创建专有网络</a> 。



类别	参数	说明
	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 <a href="#">创建交换机</a> 。
资源组	资源组	选择实例所属的资源组。
实例	版本类型	<ul style="list-style-type: none"> <li>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。</li> <li>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<a href="#">企业版简介</a>。</li> </ul>
	系列类型	<ul style="list-style-type: none"> <li>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<a href="#">性能增强型</a>。</li> <li>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<a href="#">混合存储型</a>。</li> </ul> <p> 说明 版本类型选择为企业版（Tair）时，该参数才会出现且需要设置。</p>
	版本号	Redis的引擎版本。
	架构类型	<ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> <p>详细说明请参见<a href="#">架构信息查询导航</a>。</p>
	分片数	Redis集群实例的分片数，数据将分布在该集群的各个分片上。  <p> 说明 架构类型选择为集群版时，才支持该参数。</p>
节点类型	<ul style="list-style-type: none"> <li>架构类型选择为集群版或标准版时，可选择为下述节点类型： <ul style="list-style-type: none"> <li>双副本：一主一从共两个节点，双机热备架构，数据持久化保存。</li> <li>单副本：仅一个主节点，不能保障数据可用性和服务连续性。</li> </ul> </li> <li>架构类型选择为读写分离版时，可根据只读节点的数量选择节点类型。</li> </ul>	

类别	参数	说明
	实例规格	<p>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<a href="#">规格查询导航</a>。</p> <p><b>说明</b> 实例创建后会自动生成数据库元信息，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。</p>
	密码设置	<ul style="list-style-type: none"> <li>○ <b>稍后设置</b>：在实例创建完成后设置密码，设置方法请参见<a href="#">修改密码</a>。</li> <li>○ <b>立即设置</b>：填入实例的密码。 <ul style="list-style-type: none"> <li>■ 密码长度为8~32位。</li> <li>■ 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。</li> <li>■ 支持的特殊字符为 <code>!@#%&amp;*()*_+-=</code>。</li> </ul> </li> </ul>
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 <b>包年包月（本地盘）</b> 时，您还需要设置购买时长、到期是否自动续费。

- 单击**立即购买**。
- 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。

## 创建Redis云盘实例




- 登录[Redis管理控制台](#)。
- 在页面右上角，单击**创建实例**。
- 在跳转到的购买页面，选择商品类型为**包年包月（云盘版）**。

商品类型	<input type="radio"/> 包年包月(本地盘)	<input type="radio"/> 按量计费(本地盘)	<input checked="" type="radio"/> 包年包月(云盘版)	<input type="radio"/> 数据库专属集群	<input type="radio"/> Tair包年包月
------	---------------------------------	---------------------------------	--	-------------------------------	--------------------------------

**说明** 目前云盘版仅支持包年包月的计费方式，即在新建云盘版实例时支付费用。

- 选择实例配置，参数说明如下表所示。

参数	说明
地域	<p>实例所在的地理位置。购买后无法更换地域。</p> <ul style="list-style-type: none"> <li>○ 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。</li> <li>○ 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。</li> <li>○</li> </ul>

参数	说明
主可用区	<p>即实例所属的可用区，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。</p> <p> <b>说明</b> 选择多可用区可实现实例的同城容灾。</p>
网络类型	<p>固定为<b>专有网络</b>：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。</p> <p> <b>注意</b> 请确保Redis实例与需要互连的ECS实例或RDS实例的网络类型都是专有网络，且处于同一VPC中，否则它们无法通过内网互通。</p>
专有网络	选择实例的专有网络。如果没有专有网络，请参见 <a href="#">创建专有网络</a> 。
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 <a href="#">创建交换机</a> 。
版本类型	<ul style="list-style-type: none"> <li>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。</li> <li>企业版（Tair）性能增强系列：即Redis企业版的性能增强型实例，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<a href="#">性能增强型</a>。</li> </ul>
版本号	<p>Redis的引擎版本。</p> <p> <b>说明</b> 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>
架构类型	<ul style="list-style-type: none"> <li>不启用集群：采用主从（master-replica）双副本架构，详情请参见<a href="#">标准版-双副本</a>。</li> <li>启用集群：采用分片集群架构，详情请参见<a href="#">集群版-双副本</a>。</li> </ul>
分片规格	<p>即实例的分片规格，标准版实例仅包含一个分片。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<a href="#">云盘版实例规格</a>。</p> <p> <b>说明</b> 实例创建后会自动生成数据库元信息，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。</p>
规格类型	固定为 <b>共享型</b> ，多租户共享型资源，提供高性价比。
副本数	固定为 <b>2副本</b> ，一主一从共两个节点，双机热备保障可用性。

参数	说明
分片数量	Redis集群实例的分片数。  <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> <span style="font-size: 1.2em; color: #007bff;">?</span> 说明 当架构类型选择为集群版时，支持该参数。 </div>
存储类型	固定为ESSD，即阿里云ESSD（Enhanced SSD）云盘，详情请参见 <a href="#">ESSD云盘</a> 。
磁盘容量	云盘的容量，仅供系统内部使用，不支持调整。
密码设置	<ul style="list-style-type: none"> <li>○ 稍后设置：在实例创建完成后设置密码，设置方法请参见<a href="#">修改密码</a>。</li> <li>○ 立即设置：填入实例的密码。 <ul style="list-style-type: none"> <li>■ 密码长度为8~32位。</li> <li>■ 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。</li> <li>■ 支持的特殊字符为 <code>!@#%&amp;^*()_+ -=</code>。</li> </ul> </li> <li>○</li> </ul>
实例名称	设置实例的名称，便于后续业务识别。
购买时长	设置实例的包年包月时长。

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。

## 创建持久内存型或容量存储型实例

- **持久内存型实例**：数据在持久内存中存取，提供命令级强持久化能力，适用于对性能要求较高，同时对数据一致性有要求的场景。
- **容量存储型实例**：数据在ESSD云盘中存取，提供命令级强持久化能力，大容量，适用于对性能要求不高，但是对成本有控制要求的场景。

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，选择商品类型为**Tair包年包月**。

商品类型

包年包月(本地盘)

按量计费(本地盘)

包年包月(云盘版)

数据库专属集群

Tair包年包月

? 说明 目前仅支持包年包月的计费方式，即在新建实例时支付费用。

4. 选择实例配置，参数说明如下表所示。

参数	说明
----	----

参数	说明
地域	<p>实例所在的地理位置，购买后无法更换地域。</p> <p> <b>说明</b> 目前仅支持华东1（杭州）、华东2（上海）和华北3（张家口）地域。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li></li> </ul>
主可用区	<p>即实例所属的可用区，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。</p> <p> <b>说明</b> 选择多可用区可实现实例的同城容灾。</p>
网络类型	<p>固定为<b>专有网络</b>：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。</p> <p> <b>注意</b> 请确保Redis实例与需要互连的ECS实例或RDS实例的网络类型都是专有网络，且处于同一VPC中，否则它们无法通过内网互通。</p>
专有网络	选择实例的专有网络。如果没有专有网络，请参见 <a href="#">创建专有网络</a> 。
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 <a href="#">创建交换机</a> 。
系列类型	<ul style="list-style-type: none"> <li><b>Tair 持久内存型</b>：数据在持久内存中存取，提供命令级强持久化能力，适用于对性能要求较高，同时对数据一致性有要求的场景，详情请参见<a href="#">持久内存型</a>。</li> <li><b>Tair 容量存储型</b>：数据在ESSD云盘中存取，提供命令级强持久化能力，大容量，适用于对性能要求不高，但是对成本有控制要求的场景，详情请参见<a href="#">容量存储型</a>。</li> </ul>
架构类型	<b>不启用集群</b> ：采用主从（master-replica）双副本架构，详情请参见 <a href="#">标准版-双副本</a> 。
实例规格	<p>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<a href="#">规格查询导航</a>。</p> <p> <b>说明</b> 实例创建后会自动生成数据库元信息，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。</p>
副本数量	固定为2，即一主一从共两个节点，保障可用性。
存储类型	固定为 <b>ESSD PL1</b> ，即性能级别为PL1的阿里云ESSD（Enhanced SSD）云盘，详情请参见 <a href="#">ESSD云盘</a> 。

参数	说明
存储空间	<p>选择系列类型为Tair 容量存储型时，可根据业务需求选择存储空间。</p> <p><b>说明</b> 持久内存型实例不支持调整该参数，仅将ESSD云盘作为系统运行使用（例如日志、数据备份等），不作为数据存取的介质。</p>
密码设置	<ul style="list-style-type: none"> <li>○ 稍后设置：在实例创建完成后设置密码，设置方法请参见<a href="#">修改密码</a>。</li> <li>○ 立即设置：填入实例的密码。 <ul style="list-style-type: none"> <li>■ 密码长度为8~32位。</li> <li>■ 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。</li> <li>■ 支持的特殊字符为 <code>!@#%&amp;*()_+~=</code>。</li> </ul> </li> <li>○</li> </ul>
实例名称	设置实例的名称，便于后续业务识别。
购买时长	设置实例的包年包月时长。

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。

## 创建数据库专属集群

云数据库专属集群MyBase是由多台主机（底层服务器，如ECS I2服务器、神龙服务器）组成的集群，相对于全托管数据库，可以满足您更多的需求。创建方法请参见[创建集群](#)。

## 视频演示

下方视频以创建Redis本地盘实例为例，演示创建过程，视频时长约4分钟。

## 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 <a href="#">Redis管理控制台</a> ，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 <a href="#">订单管理</a> 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 <a href="#">提交工单</a> 处理。

## 相关API


API	说明
<a href="#">CreateInstance</a>	创建一个Redis实例。

## 3.步骤2：设置白名单

为了Redis数据库的安全稳定，在开始使用Redis实例前，您需要设置Redis白名单分组或者在白名单中添加ECS安全组，将客户端IP地址或者IP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。


 **说明** 若实例小版本不满足条件，在设置白名单时会出现提示信息，此时将小版本升级到最新即可，详细步骤请参见[升级小版本](#)。

- 如需通过添加ECS安全组来设置Redis白名单，则Redis实例的引擎版本须为4.0或以上。

### 注意事项

通过DMS登录云Redis实例时，系统会自动生成一个名为ali\_dms\_group的白名单分组，请勿在该分组中添加其它IP地址，以防因DMS产品变动导致其中的其它IP丢失。

### 通过白名单分组设置Redis白名单

 **说明** 您可以同时通过白名单分组和ECS安全组设置Redis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。

- 登录[Redis管理控制台](#)。
- 在界面左上方阿里云图标的右侧选择实例所在的地域。
- 在实例列表页，单击目标实例ID或者其右侧操作栏的**管理**。
- 在实例信息页，单击左侧导航栏中的**白名单设置**。
- 在白名单设置页，创建新的白名单分组或修改已有分组：
  - 如果需要自定义白名单分组名称，则创建新的白名单分组：
    - 单击右侧的**添加白名单分组**。
    - 设置**分组名称**。

#### 说明

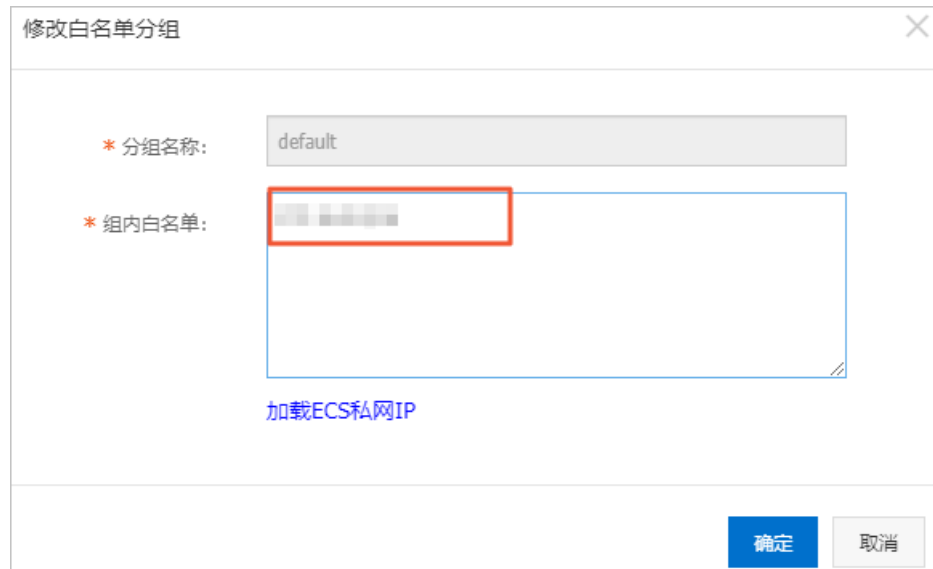
- 分组名称长度为2~32个字符，由小写字母、数字或下划线组成，开头需为小写字母，结尾需为字母或数字。
- 在白名单分组创建成功后，分组名称将不能被修改。

- 如果不需要创建自定义分组，单击目标白名单分组右侧的**修改**。
- 在**添加白名单分组**或**修改白名单分组**对话框中，从以下方式中选择其一继续。
    - 手动修改组内白名单：



a. 在组内白名单区域手动输入需要访问Redis实例的IP地址或者IP段。

手动修改组内白名单



说明

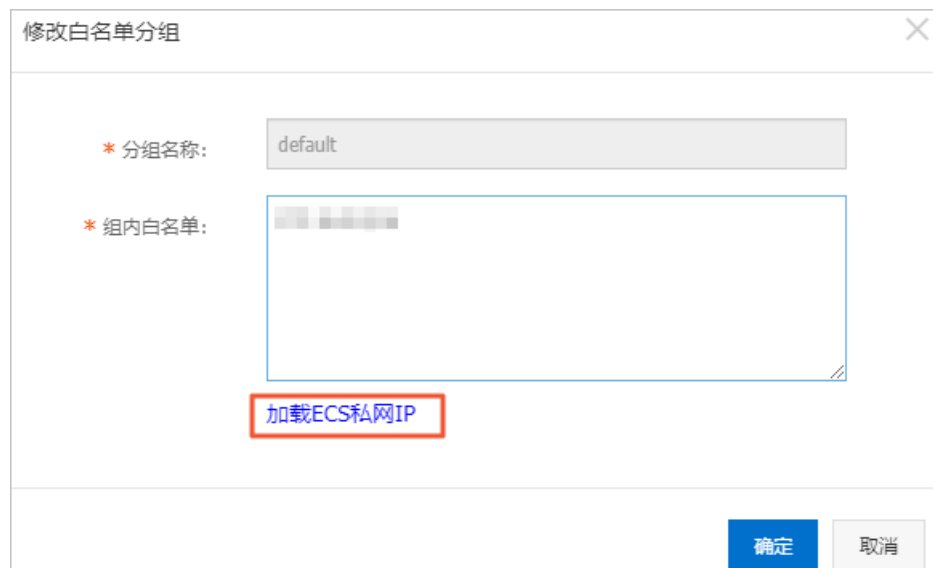
- 设置 0.0.0.0/0 表示允许所有地址访问，可能引发安全风险，不建议如此设置。
- 设置IP段表示该网段内的IP地址都可以访问该Redis实例，如 10.10.10.0/24 。
- 多个IP用英文逗号 (,) 分隔，逗号前后不加空格。
- 每个白名单分组可以设置1000个IP。

b. 单击确定。

o 加载当前阿里云账号下ECS实例的私网IP：

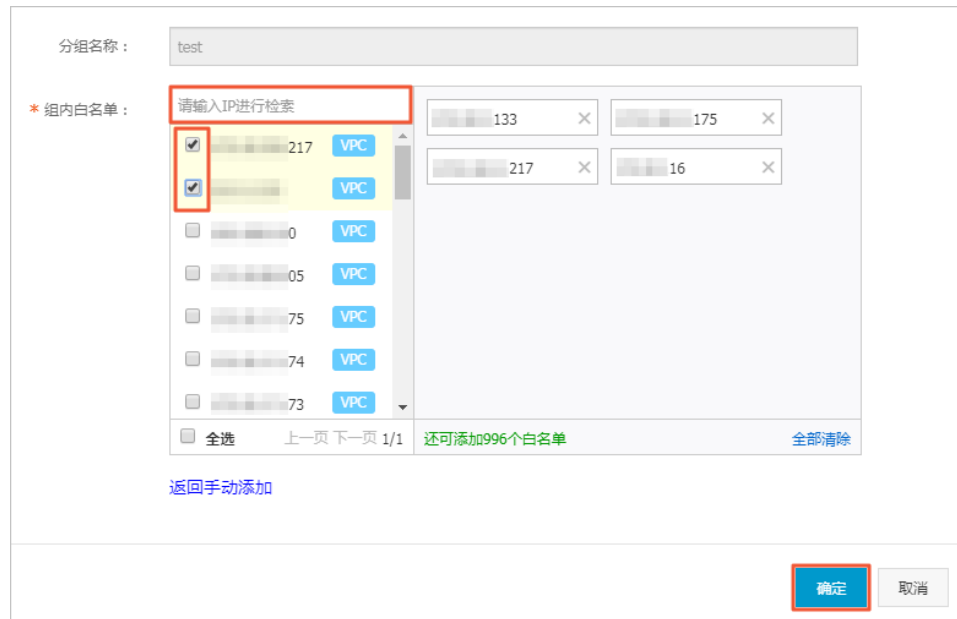
a. 单击加载ECS私网IP。

加载ECS私网IP



b. 选择目标ECS的私网IP。

### 选择ECS私网IP



**说明** 您可以在ECS私网IP列表上方的检索栏中通过ECS实例的名称、ID或IP进行模糊搜索。

c. 单击**确定**。

## 通过ECS安全组设置Redis白名单

ECS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。在Redis白名单中添加ECS安全组后，该安全组中的ECS实例就可以访问Redis实例。更多ECS安全组的信息请参见[安全组概述](#)。

1. 登录[Redis管理控制台](#)。
2. 在界面左上方阿里云图标的右侧选择实例所在的地域。
3. 在实例列表页，单击目标实例ID或者其右侧操作栏的**管理**。
4. 在实例信息页，单击左侧导航栏中的**白名单设置**。
5. 在白名单设置页，单击**添加安全组**。



6. 在添加安全组对话框，完成如下操作：

**说明** 每个Redis实例的白名单中可添加10个ECS安全组。

- i. 选择需要添加的安全组。
- ii. 单击确定。

### 相关API

API	说明
<a href="#">查询实例的IP白名单</a>	调用DescribeSecurityIps查询Redis实例的IP白名单。
<a href="#">修改IP白名单</a>	调用ModifySecurityIps设置Redis实例的IP白名单。

## 4. 步骤3：连接实例


### 4.1. DMS登录云数据库

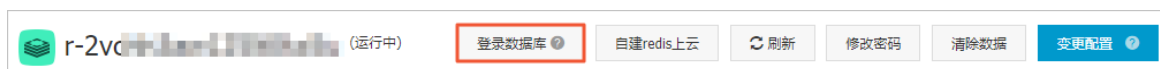
数据管理DMS（Data Management）是一款访问管理云端数据的Web服务，支持Redis、MySQL、SQL Server、PostgreSQL和MongoDB等数据源。DMS提供了数据管理、对象管理、数据流转和实例管理四部分功能。

#### 使用限制

- DMS不支持通过直连地址或外网连接地址登录Redis实例。
- 使用DMS登录Redis企业版性能增强型实例时，暂时无法使用性能增强系列的新增命令。

#### 从Redis管理控制台登录

1. 登录Redis管理控制台。
2. 在界面左上方的菜单栏中选择实例所在的地域。
3. 在实例列表页，单击目标实例ID，或在其右侧操作栏中选择  > 管理。
4. 在实例信息页，单击右上方的登录数据库。



5. 在DMS登录页，输入实例密码。



**说明**

- 如果Redis实例的网络类型为VPC，且开启了**免密访问**，则此处可不输入密码。
- 经典网络及VPC网络的实例都支持DMS。使用DMS登录VPC网络中的实例时，系统需要申请一条特殊通道，所以该类实例在第一次登录时，会有短暂的等待时间。
- 每次登录DMS都会触发SCAN命令扫描Redis数据，因此会被动逐出部分过期数据机制。

6. 单击**登录**。

7. 在**因白名单问题无法登录数据库**对话框中，单击**设置白名单**。

**危险** 该步骤将在Redis实例的**白名单**中新建一个名为ali\_dms\_group的分组，并将DMS客户端的IP添加到其中。请勿在该分组中设置其它IP地址，防止因DMS侧变动导致设置丢失。



**说明** 仅在从未使用DMS登录过该Redis实例时才会出现该对话框。

8. 再次单击**登录**。

### 从DMS管理控制台登录

1. 登录**DMS管理控制台**。
2. 输入目标实例的**连接地址**和密码。



说明

- 如果Redis实例的网络类型为VPC，且开启了**免密访问**，则此处可不输入密码。
- 经典网络及VPC网络的实例都支持DMS。使用DMS登录VPC网络中的实例时，系统需要申请一条特殊通道，所以该类实例在第一次登录时，会有短暂的等待时间。
- 每次登录DMS都会触发SCAN命令扫描Redis数据，因此会被动逐出部分过期数据机制。

3. 根据DMS使用情况完成登录：

- 如果之前已经使用DMS登录过该Redis实例，单击**登录**。
- 如果从未使用DMS登录过该Redis实例，完成以下步骤。
  - a. 在**DMS服务器尚未加入到RDS白名单**对话框，复制DMS IP地址并单击**关闭**。



- b. 将DMS的IP地址**加入白名单**。
- c. 登录**DMS管理控制台**。
- d. 输入目标实例的连接地址和密码。
- e. 单击**登录**。

### 视频演示

您可以通过观看以下视频来了解如何通过DMS登录Redis实例。视频时长2分26秒，DMS登录说明开始于1分19秒处。

## 常见问题

- 怎么在DMS中用自定义账号登录？
- 答：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 `Fakepass666`，则连接Redis时的密码为 `admin:Fakepass666`。
- 连接不上怎么办？
- 答：如果出现连接问题，请参见[连接问题排查与解决](#)。更多的DMS相关信息请参见[DMS文档](#)。

## 4.2. Redis客户端连接


您可以使用多种语言的客户端连接阿里云Redis。

### 连接前提

- 将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](#)。
- 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`。
- 通过直连模式访问Redis集群版实例需要先[开通直连访问](#)，然后[使用直连地址访问Redis实例](#)。

### 注意事项

- 集群架构的Redis实例默认提供的连接地址为proxy（代理）连接地址，连接方式与连接标准架构的Redis实例相同。

 **说明** 如果开通了[直连访问](#)，通过直连地址连接实例时，连接方式与连接开源Redis Cluster相同。

- 如果同一VPC内的实例开启了免密访问功能，则无需提供密码信息，即可连接数据库。
- 如果连接遇到问题，请参见[Redis连接问题排查与解决](#)。

### Redis客户端

由于云数据库Redis提供的数据库服务与原生的数据库服务完全兼容，连接数据库的方式也基本类似。任何兼容Redis协议的客户端都可以访问云数据库Redis版服务，您可以根据自身应用特点选用任何Redis客户端。

Redis的客户端请参见[客户端支持](#)。

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

### Jedis客户端

Jedis客户端访问云数据库Redis版服务，有以下两种方法：

- Jedis单链接（不推荐使用，单次超时无法自动恢复）
- JedisPool连接池连接（推荐使用）

操作步骤如下：

1. 下载并安装Jedis客户端，详细步骤请参见[Jedis使用说明](#)。
2. Jedis单连接示例
  - i. 打开Eclipse客户端，创建一个Project，输入如下代码段：

```
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();
}
}
}
```

- ii. 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云数据库Redis。

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

接下来您就可以通过自己的本地客户端Jedis操作您的云数据库Redis。您也可以通过JedisPool连接池来连接您的云数据库Redis。



### 3. JedisPool连接池示例

- i. 打开Eclipse客户端，创建一个Project，配置pom文件，具体配置如下所示：

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

- ii. 在project中添加如下应用：

```
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;
```

- iii. 如果您的Jedis客户端版本是Jedis-2.7.2，在Project中输入如下代码：

```
JedisPoolConfig config = new JedisPoolConfig();
//最大空闲连接数,应用自己评估, 不要超过ApsaraDB for Redis每个实例最大的连接数
config.setMaxIdle(200);
//最大连接数,应用自己评估, 不要超过ApsaraDB for Redis每个实例最大的连接数
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();
```

- iv. 如果您的Jedis客户端版本是Jedis-2.6、Jedis-2.5, 在Project中输入如下代码:

```
JedisPoolConfig config = new JedisPoolConfig();
//最大空闲连接数,应用自己评估,不要超过ApsaraDB for Redis每个实例最大的连接数
config.setMaxIdle(200);
//最大连接数,应用自己评估,不要超过ApsaraDB for Redis每个实例最大的连接数
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);
    }
}
```

- v. 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云数据库Redis。

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

接下来您就可以通过自己的本地客户端Jedis操作您的云数据库Redis。

## 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](#)。

详细说明请参见[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](#)。

详细说明请参见[GitHub](#)。

## PhpRedis客户端

操作步骤如下所示：

1. 下载并安装PhpRedis客户端，详细步骤请参见[PhpRedis使用说明](#)。
2. 在任何一款可以编辑PHP的编辑器中输入如下代码：

```
<?php
/* 这里替换为连接的实例host和port */
$host = "localhost";
$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，您就可以通过自己的本地客户端PhpRedis访问您的云数据库Redis，详情请参见[官方文档](#)。

## redis-py客户端

操作步骤如下：

1. 下载并安装redis-py客户端，详细步骤请参见[redis-py使用说明](#)。

2. 在任何一款可以编辑Python的编辑器中输入如下代码，即可建立连接通过本地客户端redis-py进行数据库操作。

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
import redis
#这里替换为连接的实例host和port
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')
```

## C/C++客户端

操作步骤如下所示：

1. 下载并编译安装 C 客户端，编译安装代码如下所示：

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

2. 在C/C++编辑器中编写如下代码：

```
#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];
```

```
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 ... */
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。

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

至此完成通过C/C++客户端连接云数据库Redis。

## .net客户端

操作步骤如下所示：

1. 下载并使用.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来连接云数据库Redis。详细的接口用法请参见[接口说明](#)。

```
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 " + key + " value " + 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);
                }
            }
        }
    }
}
```



```
        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

操作步骤如下所示：

1. 下载并安装node-redis。

```
npm install hiredis redis
```

2. 在node-redis客户端中输入如下代码并执行以此连接云数据Redis版。

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


 说明 port为端口，默认为6379；host为连接地址。例如：

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

3. 使用云数据Redis版。

```
// 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 ``
});
client.quit();
```

## C#客户端StackExchange.Redis

 **说明** 为避免StackExchange.Redis通过执行CLUSTER NODES命令，将Redis实例判断集群模式而导致不支持多数据库功能，您需要将cluster\_compat\_enable设置为0，相关操作，请参见[参数设置及说明](#)。

操作步骤如下所示：

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：

```
// 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;
}
```

#### 🔍 说明

ConfigurationOptions包含很多选项，例如keepAlive、connectRetry、name，详情请参见StackExchange.Redis.ConfigurationOptions。

4. 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. 下面给出5种数据结构的demo，它们的API和原生略有不同，分别用String、Hash、List、Set、Sorted Set开头代表5种数据结构。
  - o 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]);
```

- o 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();
```

- o 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();
```

## 4.3. redis-cli连接

您可以使用原生工具redis-cli来连接阿里云Redis。

### redis-cli连接简介

redis-cli是原生Redis自带的命令行工具，可以帮助您通过简单的命令连接Redis实例，进行数据管理。

使用redis-cli，您可以在阿里云ECS实例上的Linux系统中连接云数据库Redis版实例，或者在本地主机上通过公网访问Redis实例。通过阿里云内网访问Redis实例能够提供更高的安全和性能保障，您可以在ECS上使用redis-cli与同一VPC内的Redis实例或者同地域的经典网络Redis实例建立连接；如果您的场景需要在本地主机上从公网访问Redis实例，请先参见[公网连接](#)，申请公网连接地址，再使用本文介绍的连接方法。

### redis-cli安装方法

在Linux环境中安装原生Redis即可使用redis-cli。具体步骤请参见[Redis社区版官网](#)。

### 前提条件

#### 内网连接

- ECS实例与Redis实例的网络类型同为VPC时，二者需在同一地域的同一VPC中。
- ECS实例与Redis实例的网络类型同为经典网络时，二者需在同一地域。
- Redis实例的白名单中加入了ECS实例的内网IP地址。
- ECS中已安装原生Redis。
- 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`。

#### 公网连接

- Redis实例申请了公网连接地址，申请方式参见[公网连接](#)。
- Redis实例的白名单中加入了本地主机的公网IP地址。
- 本地主机的系统为Linux。

- 本地主机已安装原生Redis。
- 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`。


## 注意事项

- 如果通过内网地址访问Redis实例并开启了VPC免密，连接时无需密码验证，开启方式请参见[VPC免密](#)。
- 如果通过公网地址访问Redis实例并开启了VPC免密，连接时仍然需要密码验证。
- 如果连接遇到问题，请参见[Redis连接问题排查与解决](#)。

## 连接方法

在Linux命令行使用如下命令连接Redis实例：

```
redis-cli -h <host> -p <port> -a '<password>'
```

 说明 使用直连地址访问Redis集群实例需要设置 `-c` 选项，示例如下。

```
redis-cli -h <host> -p <port> -a '<password>' -c
```

## 选项说明

选项	说明
-h	Redis实例的连接地址。 <ul style="list-style-type: none"> <li>● 内网连接：系统默认提供内网连接地址，查看方式请参见<a href="#">查看连接地址</a>。</li> <li>● 公网连接：您需要申请公网连接地址才能从公网连接实例，申请方式请参见<a href="#">公网连接</a>。</li> </ul>
-p	Redis实例的服务端口。默认的内网连接和公网连接端口均为6379，可自定义，修改方式参见 <a href="#">修改连接端口</a> 。
-a	Redis实例的连接密码，使用单引号（'）包裹连接密码，例如 <code>'Rp829dlwa'</code> 。 <p> 说明 为提高安全性，您也可以不设置该选项，在执行连接命令后再输入 <code>auth &lt;password&gt;</code> 来验证密码（如下图所示）。</p>
-c	使用直连地址访问云Redis集群版实例时需要设置该参数。


### 连接示例

```
[root@ ~]# redis-cli -h r-bp1-1.redis.rds.aliyuncs.com -p 6379
r-bp1-1.redis.rds.aliyuncs.com:6379> auth a
OK
r-bp1-1.redis.rds.aliyuncs.com:6379>
```

## 常见问题

Q: 通过redis-cli连接时提示“ERR invalid password”。

A: Redis实例未添加正确的白名单，请添加正确的白名单。

 **说明** 由于实例的小版本过低，提示的内容可能不准确。将实例升级至最新的小版本后，如果遇到未添加正确的白名单，命令行会提示“ERR illegal address”，升级方法请参见[升级小版本](#)。

## 视频演示


您可以通过观看以下视频来了解如何通过redis-cli连接Redis实例，视频时长约2分钟。

## 4.4. 公网连接

云数据库Redis版实例的公网连接地址（Public Endpoint）是一个可以通过因特网直接访问的地址。您可以通过公网连接地址从公网访问云数据库Redis版实例。相对于使用内网连接地址连接Redis实例，使用公网连接地址会提高网络耗时，影响Redis服务的性能。在生产环境中，建议您通过内网地址连接Redis，确保Redis服务的高性能。

### 前提条件

- 请确保Redis实例的白名单中设置了ECS实例或本地主机的公网IP地址。设置方式请参见[步骤2：设置白名单](#)。
- 2.8或5.0版本的Redis实例在开启免密访问后无法申请公网连接地址，如需使用公网连接请先关闭免密访问再申请公网连接地址。免密访问的开关方法请参见[开启专有网络免密访问](#)。

 **说明** 4.0版本的Redis实例可以在开启免密访问后申请公网连接地址，此时使用内网地址访问Redis实例无需密码，使用公网地址访问仍需要密码。

- 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`。

### 适用场景

- 本地访问：通过本地主机连接阿里云Redis实例。
- 跨账号访问：通过ECS连接不同阿里云账号下的Redis实例。
- 跨地域访问：同一阿里云账号下的ECS实例与Redis实例需要互连，但二者属于不同地域。
- 跨VPC访问：同一阿里云账号下的ECS实例与Redis实例需要互连，二者属于同地域但VPC不同。
- 跨网络类型访问：同一阿里云账号下的ECS实例与Redis实例需要互连，二者属于相同地域，但网络类型不同。

### 费用

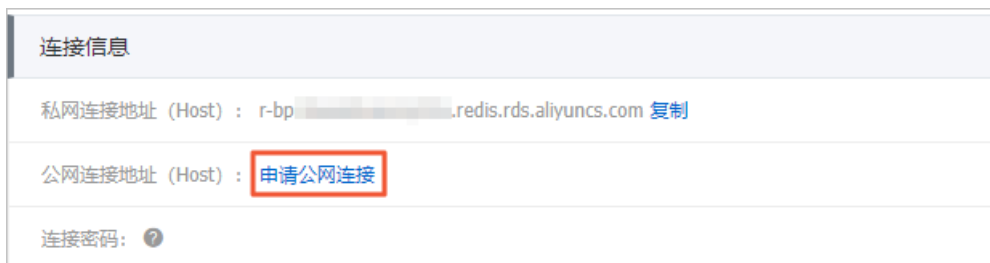
Redis公网连接功能及公网流量暂不收取费用。

### 获取公网连接地址

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. 在连接信息区域框，单击申请公网连接。



连接信息

私网连接地址 (Host) : r-bp1...redis.rds.aliyuncs.com 复制

公网连接地址 (Host) : 申请公网连接

连接密码: ?

5. 在弹出的对话框中，填写自定义的连接地址和端口或保持默认值。



申请外网连接

连接地址: r-bp1...redis.rds.aliyuncs.com  
由字母，数字组成，小写字母开头，8-64个字符

端口: 6379  
端口范围: 1024~65535

将为当前Redis实例申请外网连接，申请后需要外网IP设置白名单可访问。

确定 取消

6. 单击确定。

操作完成后，即可在连接信息区域框中查看到对应的公网连接地址。

? 说明 如果不再需要公网连接地址，单击释放公网连接即可。

## 使用公网地址连接实例

您可以使用redis-cli或各语言的Redis客户端等工具连接Redis实例，连接方式请参见如下文档：

- [redis-cli连接](#)
- [Redis客户端连接](#)

## 公网连接失败的解决方法

- 请确认使用的地址是Redis实例的公网连接地址而不是内网连接地址。
- 请检查Redis白名单中是否添加了客户端所在主机的公网IP地址。
- 内网连接问题请参见[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 &lt;master名称&gt;</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 &lt;master名称&gt;</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。

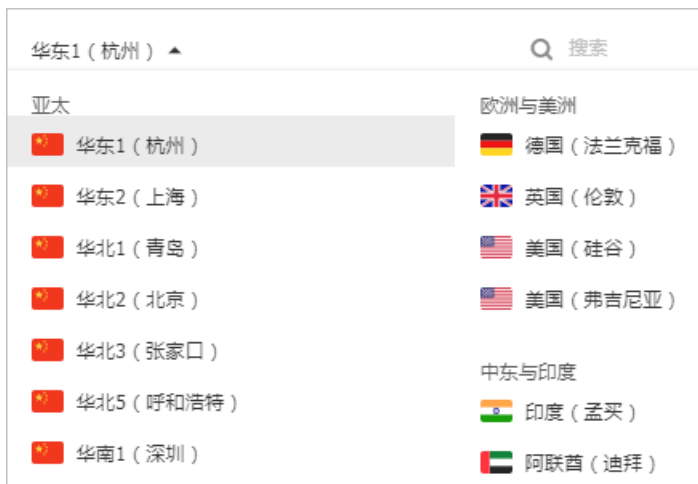
说明 若没有阿里云账号，请先[注册](#)。

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[DMS登录云数据库](#)。

### 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
- 缓存分析

## 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 style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p>说明</p> <ul style="list-style-type: none"> <li>• 所有DB存储的数据总量受限于实例的内存大小；</li> <li>• 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）。</li> </ul> </div>
Redis命令支持	详情请参见 <a href="#">命令查询导航</a> 。

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 <a href="#">报警设置</a> 。 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"> <li>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。</li> <li>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。</li> </ul>
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 <code>AOF_FSYNC_EVERYSEC</code> 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
<a href="#">Redis 2.8命令支持</a>	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。
<a href="#">Redis 4.0命令支持</a>	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。
<a href="#">Redis 5.0命令支持</a>	本章节介绍引擎版本（版本号）为5.0的云数据库Redis实例支持的命令，包括Redis 5.0标准版、Redis 5.0集群版、和Redis 5.0读写分离版。
<a href="#">未开放的命令</a>	本章节介绍云数据库Redis暂未开放的原生Redis命令。
<a href="#">集群架构的命令限制</a>	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。
<a href="#">阿里云自研的Redis命令</a>	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
------	----

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。
性能增强型实例命令支持概览	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： <ul style="list-style-type: none"> <li>• CAS/CAD命令</li> <li>• TairString命令</li> <li>• TairHash命令</li> <li>• TairGIS命令</li> <li>• TairBloom命令</li> <li>• TairDoc命令</li> </ul>

## 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[命令查询导航](#)。

- 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[命令查询导航](#)。

### 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[命令查询导航](#)。

## 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。

 **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。