

目录

目录	1
购买使用常见问题	2
云数据库Redis适用于哪些场景?	2
如何访问服务?	2
服务兼容Redis哪个版本?	2
与其他公有云服务提供厂商同类产品相比,有什么特点?	2
支持哪种方式的数据持久化?	2
通过什么方式保证服务可用性和数据可靠性?	2
主从切换的时间窗口有多久?会对用户的正常使用造成影响吗?	2
如果从服务器down掉,在恢复之后会保证故障期间主服务器上的数据更新被完整的同步到吗?	2
用户需要对从服务以及数据持久化所需要的磁盘空间单独付费吗?	2
支持对主服务配置多个从服务吗?	2
服务支持主从服务读写分离吗?	2
服务支持在线扩容吗?扩容过程中会对用户访问和用户数据造成影响吗?	2
支持缩容吗?	2
支持最大QPS在什么规模?	2
redis过期键是指什么?监控中显示的过期键个数和淘汰键个数是指什么?	3
只读实例从哪里同步数据,是主节点还是备节点?	3
主从版实例能否升级到集群模式?	3
服务提供什么样的承诺?	3

购买使用常见问题

云数据库Redis适用于哪些场景？

可以充当磁盘数据存储系统，如关系数据库的缓存，提供比磁盘访问更高的数据访问性能；同时，由于支持数据持久化，还可以直接充当键值存储系统。

如何访问服务？

访问服务，用户需要开通金山云云服务器KEC服务，支持内网访问，开通服务之后，通过配置实例安全组的方式配置允许访问服务的云服务器KEC。

服务兼容Redis哪个版本？

以Redis4.0和Redis5.0为基准实现，除个别命令之外，支持全命令集，具体信息请见[使用限制-支持命令](#)。

与其他公有云服务提供厂商同类产品相比，有什么特点？

支持Redis单主从模式，能够提供随容量增大而线性增长的访问性能。支持在线热扩容，平滑升级。

支持哪种方式的数据持久化？

用户可以通过在控制台上配置“appendonly”参数来激活AOF持久化。需要指出的是，如果用户不主动配置“appendonly”参数（默认“no”，不激活），那么用户的数据不会持久化。

通过什么方式保证服务可用性和数据可靠性？

通过跨物理机主从备份以及双重自动failover机制保证服务高可用（HA）。

主从切换的时间窗口有多久？会对用户的正常使用造成影响吗？

从主服务down掉到从服务切换为主的身份正常提供服务，整个过程能够在秒级完成，同时考虑到网络闪断情况，主从切换需要较为谨慎，因此一般不会超过30秒，在这个过程中，用户的访问可能会有闪断的情况，但是一般用户应用都会有重连机制，因此用户一般不会对主从切换能有感知。

如果从服务器down掉，在恢复之后会保证故障期间主服务器上的数据更新被完整的同步到吗？

会，从服务器会保证将故障期间主服务器上的数据更新完整地同步过来。

用户需要对从服务以及数据持久化所需要的磁盘空间单独付费吗？

不需要，服务只按照缓存容量收费，不需要对从服务以及磁盘空间单独付费。

支持对主服务配置多个从服务吗？

支持，最多支持7个只读副本，实现一主多从的服务。

服务支持主从服务读写分离吗？

支持，适用于读写压力较大的场景。

服务支持在线扩容吗？扩容过程中会对用户访问和用户数据造成影响吗？

在线扩容，一键升级，用户无感知。

支持缩容吗？

控制台暂不支持。但可以联系商务，通过运营平台进行操作。

支持最大QPS在什么规模？

主从版：峰值QPS10万（以key<512B，value<1000B的场景为例）。集群实例QPS=分片数*10w（预估值，以具体的测试场景为准）。分片数根据内存系统分配，最小3片。

redis过期键是指什么？监控中显示的过期键个数和淘汰键个数是指什么？

redis过期键是指设置了过期时间的键，redis内部有机制轮询删除到了过期时间的键。控制台监控显示：淘汰键个数，是指运行以来，淘汰的 key 的数量；过期键个数，是指运行以来，过期的 key 的数量；只是统计一下过去有多少键过期。这些键都已经删除了。

只读实例从哪里同步数据，是主节点还是备节点？

采用星型复制的方式：所有只读节点和备节点都是从主同步数据，各节点之间保持独立，同时复制链比较短，复制延迟比较小。

主从版实例能否升级到集群模式？

目前控制台不支持自主，两种模式架构不一致，集群版有proxy负责命令转发，路由。如有需要，请联系售后发工单。

服务提供什么样的承诺？

主从、集群实例提供99.95%服务可用性和99.999%数据可靠性，企业版集群公测期间暂不提供相关承诺。

服务可用性99.95%，计算方法如下：服务可用性=单个实例每个服务周期所有可用时间/（单个实例每个服务周期所有可用时间+单个实例每个服务周期所有不可用时间），统计周期为月，时间单位分钟，如果服务能够在五分钟之内恢复，则不计入不可用时间。

数据可靠性99.999%，在按月统计的周期内，每十万个服务实例最多有一个数据丢失的情况。