阿里云
专有云Enterprise版

安全白皮书

产品版本：V3.0.0

文档版本：20171101

为了无法计算的价值 | 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

表 1: 格式约定

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。
	用于警示信息、补充说明等，是用户必须了解的内容。	 **注意:** 导出的数据中包含敏感信息，请妥善保存。
	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 云服务器ECS

1.1 产品介绍

云服务器（Elastic Compute Service，简称ECS）是一种简单高效、处理能力可弹性伸缩的计算服务，帮助您快速构建更稳定、安全的应用，提升运维效率，降低IT成本，使您更专注于核心业务创新。

云服务器（ECS）从实例安全、存储安全、网络安全三个方面保障服务的安全可靠。

1.2 实例安全

云服务器（ECS）在多个层面上保护用户的实例，包括：

- 物理服务器上的虚拟化平台
- ECS实例上的操作系统（Guest OS）
- 防火墙

多重安全措施相互作用，共同保护用户的ECS实例（包括实例本身和实例中的数据），确保用户的ECS实例不会通过未授权的方式被访问。

1.2.1 宿主机的操作系统安全

宿主机的操作系统是阿里云根据云特点定制的、重新增减并进行编译的加固操作系统。同时，对安全策略和安全访问上做了大量的深度加固定制。

1.2.2 实例的操作系统（Guest OS）安全

用户拥有对ECS实例的操作系统的完全控制权，阿里云没有任何权限访问用户的实例及实例上的操作系统。

同时，阿里云强烈建议用户采用安全的方式对ECS实例上的操作系统进行访问和操作。比如使用SSH公钥和私钥对，并妥善保存私钥（至少要求使用复杂密码，可在创建实例时设置）；采用更安全的SSHv2方式远程登录；采用 sudo指令的方式做提权等。

1.2.3 镜像安全

阿里云基础镜像集成了所有已知的高危漏洞补丁，最大限度防止ECS实例上线后即处于高风险状态。阿里云使用数据校验算法和单向散列算法确保镜像完整性，防止被恶意篡改。

在发现新的高危安全漏洞后，用户可以迅速更新基础镜像。同时，用户可以完全自主地对ECS实例上的操作系统进行升级或漏洞修复。

强烈建议在不影响用户业务部署的情况下，使用阿里云的基础镜像作为上云的第一步。

1.2.4 防止IP/MAC/ARP欺骗

在传统网络环境里，IP/MAC/ARP欺骗一直是网络面临的严峻考验。通过IP/MAC/ARP欺骗，黑客可以扰乱网络环境，窃听网络机密。

阿里云云平台通过宿主机上的网络底层技术机制，彻底解决了这一问题。在宿主机数据链路层隔离由ECS实例向外发起的异常协议访问，阻断ECS实例ARP/MAC欺骗，并在宿主机网络层防止ECS实例IP欺骗。

1.3 存储安全

1.3.1 Chunk

ECS用户对虚拟磁盘的读写最终都会被映射为对阿里云数据存储平台上的文件的读写。阿里云提供一个扁平的线性存储空间，并在内部对线性地址进行切片，一个分片称为一个Chunk。对于每一个Chunk，阿里云都会复制出三个副本，并将这些副本按照一定的策略存放在集群中的不同节点上，保证用户数据的可靠。

1.3.2 三份副本的原理

在阿里云数据存储系统中，有三类角色，分别称为Master、Chunk Server和Client。ECS用户的每一个写操作经过层层转换，最终会交由Client来执行，执行过程如下：

1. Client计算出这个写操作对应的Chunk。
2. Client向Master查询该Chunk的三份副本的存放位置。
3. Client根据Master返回的结果，向对应的三个Chunk Server发出写请求。
4. 如果三份副本都写成功，Client向用户返回成功；反之，Client向用户返回失败。

Master的分布策略会综合考虑集群中所有Chunk Server的磁盘使用情况、在不同交换机机架下的分布情况、电源供电情况、及机器负载情况，尽量保证一个Chunk的三个副本分布在不同机架下的不同 Chunk Server 上，从而有效防止由于一个Chunk Server或一个机架的故障导致的数据不可用。

1.3.3 数据保护机制

当有数据节点损坏，或者某个数据节点上的部分硬盘发生故障时，集群中部分Chunk的有效副本数会小于三。一旦发生这种情况，Master就会启动复制机制，在Chunk Server之间复制数据，保证集群中所有Chunk的有效副本数达到三份。

综上所述，对云盘上的数据而言，所有用户层面的操作都会同步到底层三份副本上，无论是新增、修改还是删除数据。通过这种机制，保障用户数据的可靠性和一致性。

另外，在用户进行删除操作后，释放的存储空间由分布式文件系统回收，禁止任何用户访问，并在被再次使用前进行内容擦除（包括云盘的每一块上的内容），最大限度保证用户的数据安全性。

1.4 网络安全

1.4.1 实例的安全隔离

实例的安全隔离包括：

CPU隔离

阿里云ECS支持Xen和KVM两种Hypervisor，基于硬件虚拟化技术VT-x，Hypervisor运行在vmx root模式，而ECS实例运行在vmx non-root模式。通过硬件机制进行隔离，有效地防止了ECS实例访问特权资源，同时也实现了 ECS 实例之间的有效隔离。

内存隔离

在虚拟化层，Hypervisor隔离内存。ECS实例运行时，使用硬件辅助的扩展页表（Extended Page Tables，简称EPT）技术，确保ECS实例之间无法互访对方内存。

ECS实例释放后，它所有的内存会被Hypervisor清零，防止ECS实例关闭后释放的物理内存页内容被其他ECS实例访问到。

存储隔离

在虚拟化层，Hypervisor采用分离设备驱动模型实现I/O虚拟化，ECS实例不能直接访问物理磁盘，所有I/O操作都会被Hypervisor截获处理。Hypervisor保证ECS实例只能访问被分配到的虚拟磁盘空间，从而实现不同ECS实例磁盘空间的安全隔离。

网络隔离

ECS采用虚拟交换机（Virtual Switch）。发往某个ECS实例的报文只会送到这个ECS实例的虚拟网卡所对应的虚拟交换机端口，其他ECS实例不可能接收或嗅探这个报文。

运行在混合（Promiscuous）模式下的虚拟实例也不可能接收或嗅探到去往其他虚拟实例的流量。即使把网络接口设置为混合模式，Hypervisor也不会传送任何到其他目的地址的流量给其他虚拟实例。

同时，阿里云还采用专有网络VPC和安全组防火墙进行网络隔离。

通过以上隔离措施，即使同一个用户拥有的运行在同一台物理服务器上的两个虚拟实例之间也不能嗅探到对方流量。

除此之外，阿里云建议您采用更加安全的方式进行数据加密后存储到ECS实例的虚拟磁盘上，例如采用加密的文件系统等方法。

1.4.2 安全组

安全组是阿里云提供的分布式虚拟化防火墙，具备状态检测包过滤功能，是ECS实例网络安全防护的另一层保障。安全组独立于ECS实例上操作系统内部的防火墙，是在ECS实例外部提供的另一种防护手段。安全组允许设置到单IP单端口粒度的出入方向的策略，可用于安全域隔离控制等。

安全组是一个逻辑上的分组，这个分组是由同一个地域（Region）内具有相同安全保护需求并相互信任的实例组成。通过安全组可设置单台或多台ECS实例的网络访问控制，是重要的网络安全隔离手段，用于在云端划分网络安全域。

每个ECS实例至少属于一个安全组。同一安全组内的实例之间网络互通，不同安全组的实例之间默认内网不通。通过设置，可以授权某个源安全组或某个源网段，访问或者不能访问目的安全组中的实例；也可以授权允许或者不允许某个目标安全组或者目标网段被目的安全组中的实例访问。

阿里云强烈建议用户使用专有网络VPC、安全组和ECS实例中操作系统内含的防火墙共同作用的策略，以最大限度保障网络安全。

2 对象存储OSS

2.1 什么是 OSS

对象存储服务（Object Storage Service，简称 OSS）提供海量、安全、低成本、高可靠的云存储服务。它可以理解为一个即开即用，无限大空间的存储集群。相比传统自建服务器存储，OSS 在可靠性、安全性、成本和数据处理能力方面都有着突出的优势。使用 OSS，您可以通过网络随时存储和调用包括文本、图片、音频和视频等在内的各种非结构化数据文件。

OSS 将数据文件以对象/文件（object）的形式上传到存储空间（bucket）中。 您可以进行以下操作：

- 创建一个或者多个存储空间
- 每个存储空间中添加一个或多个文件
- 通过获取已上传文件的地址进行文件的分享和下载
- 通过修改存储空间或文件的属性或元信息来设置相应的访问权限
- 通过云控制台执行基本和高级 OSS 任务
- 通过开发工具包 SDK 或直接在应用程序中进行 RESTful API 调用执行基本和高级 OSS 任务。

2.2 身份验证

阿里云用户可以在云控制台里自行创建 Access Key。Access Key 由 AccessKey ID 和 AccessKey Secret 组成，其中 ID 是公开的，用于标识用户身份，Secret 是秘密的，用于用户身份的鉴别。

当用户向 OSS 发送请求时，需要首先将发送的请求按照 OSS 指定的格式生成签名字符串，然后使用 AccessKey Secret 对签名字符串进行加密（基于 HMAC 算法）产生验证码。验证码带时间戳，以防止重放攻击。OSS 收到请求以后，通过 AccessKey ID 找到对应的 AccessKey Secret，以同样的方法提取签名字符串和验证码，如果计算出来的验证码和提供的一致即认为该请求是有效的；否则，OSS 将拒绝处理这次请求，并返回 HTTP 403 错误。

2.3 访问控制

对 OSS 的资源访问分为拥有者访问和第三方用户访问。拥有者是指 bucket 的拥有者，第三方用户是指访问 bucket 资源的其他用户。访问分为匿名访问和带签名访问。对于 OSS 来说，如果请求中没有携带任何和身份相关的信息即为匿名访问。带签名访问是指按照 OSS API 文档中规定的在请求头部或者在请求 URL 中携带签名的相关信息。

OSS 提供 bucket 和 object 的权限访问控制。

Bucket 有三种访问权限：public-read-write，public-read 和 private。

- public-read-write：任何人（包括匿名访问）都可以对该 bucket 中的 object 进行 PUT、Get 和 Delete 操作。
- public-read：只有该 bucket 的创建者可以对该 bucket 内的 object 进行写操作（包括 Put 和 Delete Object）；任何人（包括匿名访问）可以对该 bucket 中的 object 进行读操作（Get Object）。
- private：只有该 bucket 的创建者可以对该 bucket 内的 object 进行读写操作（包括 Put、Delete 和 Get Object）；其他人无法访问该 bucket 内的 object。

用户新创建一个 bucket 时，如果不指定 bucket 权限，OSS 会自动为该 bucket 设置 private 权限。

Object 有四种访问权限：public-read-write，public-read，private 和 default。

- public-read-write：所有用户拥有此 object 的读写权限。
- public-read：非此 object 的 Owner 拥有此 object 的读权限，只有此 object 的 Owner 拥有此 object 的读写权限。
- private：此 object 的 Owner 拥有该 object 的读写权限，其他的用户对此 object 没有读、写权限。
- default：object 遵循 bucket 的访问权限。

用户上传 object 时，如果不指定 object 权限，OSS 会为 object 设置为 default 权限。

2.4 RAM和STS

OSS 已经接入 RAM/STS 鉴权。

RAM (Resource Access Management) 是阿里云提供的资源访问控制服务。通过 RAM，主账号可以创建出子账号，子账号从属于主账号，所有资源都属于主账号，主账号可以将所属资源的访问权限授予给子账号。

STS（Security Token Service）是阿里云提供的临时访问凭证服务，提供短期访问权限管理。STS 可以生成一个短期访问凭证给用户使用，凭证的访问权限及有效期限由用户定义，访问凭证过期后会自动失效。

2.5 高可用性

OSS 服务可用性高达 99.9%。

在一个 Region 内，OSS 数据采用三副本存储，可靠性达到 99.99999999%。

2.6 租户隔离

OSS 将用户数据切片，按照一定规则，离散存储在分布式文件系统中，并且用户数据和数据索引分离存储。OSS 用户认证采用 Access Key 对称密钥认证技术，对于用户的每个 HTTP 请求都验证签名。在用户验证通过后，重组用户离散存储的数据，从而实现多租户间的数据存储隔离。

2.7 服务器端加密

OSS 支持在服务器端对用户上传的数据进行加密（Server-Side Encryption）。当用户上传数据时，OSS 对收到的用户数据加密，然后再将加密得到的数据永久保存下来。用户下载数据时，OSS 自动对保存的加密数据解密后把原始数据返回给用户，并在返回的 HTTP 请求 Header 中声明该数据进行了服务器端加密。换句话说，下载一个进行服务器端加密编码的Object和下载一个普通的Object没有多少区别，因为OSS会为用户管理整个编解码过程。

OSS的服务器端加密编码是Object的一个属性。用户创建 Object 时，只需要在 Put Object 的请求中携带 x-oss-server-side-encryption 的 HTTP Header，并指定其值为 AES256，即可以实现该 Object 的服务器端加密存储。

2.8 客户端加密

客户端加密是指用户数据在发送给远端服务器之前就完成加密，而加密所用的密钥明文只保留在用户本地，从而可以保证用户数据安全，即使数据泄漏别人也无法解密得到原始数据。

2.9 数据传输安全

OSS 支持用户使用安全套接层协议访问 OSS。用户通过访问 OSS 持有证书的域名，保证数据信道安全，避免中间人攻击。

2.10 数据传输完整性

数据在客户端和服务器之间传输时有可能会出错。OSS现在支持对各种方式上传的Object返回其CRC64值，客户端可以和本地计算的CRC64值做对比，从而完成数据完整性的验证。

2.11 访问日志记录

OSS 提供自动保存访问日志记录（logging）功能，用户开启 Bucket 的日志保存功能后，OSS自动将访问这个Bucket的请求日志，以小时为单位，按照固定的命名规则，生成一个Object写入用户指

定的目标Bucket（ Target Bucket），作为审计或者特定行为分析使用。请求日志中包含请求时间、来源 IP、请求对象、返回码、处理时长等内容。

2.12 跨资源共享

跨域访问，或者说JavaScript的跨域访问问题，是浏览器出于安全考虑而设置的一个限制，即同源策略。当来自于A网站的页面中的JavaScript代码希望访问B网站的时候，浏览器会拒绝该访问，因为A、B两个网站是属于不同的域。

在实际应用中，经常会有跨域访问的需求，比如用户的网站www.a.com，后端使用了OSS。在网页中提供了使用JavaScript实现的上传功能，但是在该页面中，只能向www.a.com发送请求，向其他网站发送的请求都会被浏览器拒绝。这样就导致用户上传的数据必须从www.a.com中转。如果设置了跨域访问的话，用户就可以直接上传到OSS而无需从www.a.com中转。

OSS 支持 CORS 协议，可以支持用户配置跨域访问权限。用户可以设置 Bucket 允许的跨域请求来源。Bucket 默认不开启 CORS 功能，所有跨域请求都不允许。

2.13 防盗链

OSS是按使用收费的服务，为了防止用户在OSS上的数据被其他人盗链，OSS支持基于HTTP header中表头字段referer的防盗链方法。用户可以通过OSS管理控制台或者API的方式对一个Bucket设置referer字段的白名单和是否允许referer字段为空的请求访问。例如，对于一个名为oss-example的Bucket，设置其referer白名单为**http://www.aliyun.com/**。则所有referer为**http://www.aliyun.com/**的请求才能访问oss-example这个Bucket中的Object。

3 云数据库RDS版

3.1 产品概述

阿里云关系型数据库（Relational Database Service，RDS）是一种稳定可靠、可弹性伸缩的在线数据库服务。基于阿里云分布式文件系统和高性能存储，RDS 支持 MySQL数据库引擎，并且提供了容灾、备份、恢复、监控、迁移等方面的全套解决方案 。

云数据库RDS提供了多样化的安全加固功能来保障用户数据的安全，其中包括但不限于：

- **网络**：IP 白名单、VPC 网络
- **存储**：自动备份
- **容灾**：同城容灾（多可用区实例）、异地容灾（容灾实例）

3.2 租户隔离

RDS通过虚拟化技术进行租户隔离，每个租户拥有自己独立的数据库权限。同时阿里云对运行数据库的服务器进行了安全加固，例如禁止用户通过数据库读写操作系统文件，确保用户无法接触其他用户的数据。

3.3 高可用性

高可用版RDS实例拥有两个数据库节点进行主从热备，主节点发生故障可以迅速切换至备节点，月服务可用性承诺为99.95%。

用户可以随时发起数据库的备份，RDS能够根据备份策略将数据库恢复至任意时刻，提高了数据可回溯性。

3.4 访问控制

- **数据库账号**

  当用户创建实例后，RDS并不会为用户创建任何初始的数据库账户。用户可以通过控制台或者Open API来创建普通数据库账户，并设置数据库级别的读写权限。如果用户需要更细粒度的权限控制，比如表/视图/字段级别的权限，也可以通过控制台或者Open API先创建超级数据库账户，并使用数据库客户端和超级数据库账户来创建普通数据库账户。超级数据库账户可以为普通数据库账户设置表级别的读写权限。

- **IP白名单**

默认情况下，RDS实例被设置为不允许任何IP访问，即127.0.0.1。用户可以通过控制台的数据安全性模块或者Open API 来添加IP白名单规则。IP 白名单的更新无需重启RDS实例，因此不会影响用户的使用。IP白名单可以设置多个分组，每个分组可配置1000个IP或IP段。

3.5 网络隔离

- **VPC网络**

  除了IP白名单外，RDS还支持用户使用VPC来获取更高程度的网络访问控制。VPC是用户在公共云里设定的私有网络环境，通过底层网络协议严格地将用户的网络包隔离，在网络2层完成访问控制；用户可以通过VPN或者专线，将自建IDC的服务器资源接入阿里云，并使用VPC自定义的RDS IP段来解决IP资源冲突的问题，实现自有服务器和阿里云ECS同时访问RDS的目的。

  使用VPC和IP白名单将极大程度提升RDS实例的安全性。

- **Internet**

  部署在VPC中的RDS实例默认只能被同一个VPC中的ECS实例访问。如果有需要也可以通过申请公网IP的方式接受来自公网的访问（不推荐），包括但不限于：

  - 来自ECS EIP的访问。
  - 来自用户自建IDC公网出口的访问。

  IP白名单对RDS实例的所有连接方式生效，建议在申请公网IP前先设置相应白名单规则。

3.6 SQL审计

RDS提供查看SQL明细功能，用户可定期审计SQL，及时发现问题。RDS Proxy记录所有发往RDS的SQL语句，内容包括连接IP、访问的数据库名称、执行语句的账号、SQL语句、执行时长、返回记录数、执行时间点等信息。

3.7 备份恢复

为了保证数据完整可靠，数据库需要常规的自动备份来保证数据的可恢复性。RDS提供两种备份功能，分别为数据备份和日志备份。

3.8 软件升级

RDS 为用户提供数据库软件的新版本。在绝大多数情况下版本升级都是非强制性的。只有用户主动重启了RDS实例的时候，RDS才会将被重启实例的数据库版本升级到新的兼容版本。在极少数情况下（如致命的重大Bug、安全漏洞），RDS会在实例的可运维时间内发起数据库版本的强制升级。

需要注意的是，强制升级的影响仅仅是几次数据库连接闪断，在应用程序正确配置了数据库连接池的情况下不会对应用程序造成明显的影响。用户可以通过控制台或者Open API来修改可运维时间，以避免RDS在业务高峰期发生了强制升级。

3.9 RAM和STS支持

用户通过云账户创建的RDS实例，都是该账户自己拥有的资源。默认情况下，云账户对自己的资源拥有完整的操作权限。

RDS已支持RAM服务。使用阿里云的RAM服务，用户可以将云账户下RDS资源的访问及管理权限授予RAM中子用户。RDS同时支持STS服务，通过临时访问凭证提供短期访问权限管理。

3.10 最佳实践

- **网络设置**

  RDS支持公网的访问和私网的访问，如果仅在阿里云内部使用，建议设置为私网访问。如果再VPC内访问，建议选择VPC实例。

- **IP白名单设置**

  通过RDS控制台，设置RDS连接IP白名单为对应的ECS私网IP或者云服务的私网IP。

4 云数据库Redis版

4.1 产品介绍

云数据库Redis版是兼容Redis协议标准的、提供持久化的内存数据库服务，基于高可靠双机热备架构，满足高读写性能场景及容量需弹性变配的业务需求。

云数据库 Redis版采用主从（Replication）模式搭建。主节点提供日常服务访问，备节点提供 HA高可用，当主节点发生故障，系统会自动在30秒切换至备节点，保证业务平稳运行。

云数据库Redis版从多个角度提供方案来保障服务安全可靠，包括但不限于：

- 访问控制：数据库账号密码、IP 白名单
- 高可用架构：双机高可用架构

4.2 产品安全和可靠性方案

4.2.1 访问控制

数据库账号

访问Redis必须通过强制密码认证，是访问Redis的凭证。云数据库Redis版针对短连接等模式做了性能优化，开启密码认证并不会影响Redis的实例性能。

IP 白名单

云数据库Redis版提供了IP白名单来实现网络安全访问控制，支持为每个云数据库Redis版实例单独设置IP白名单。

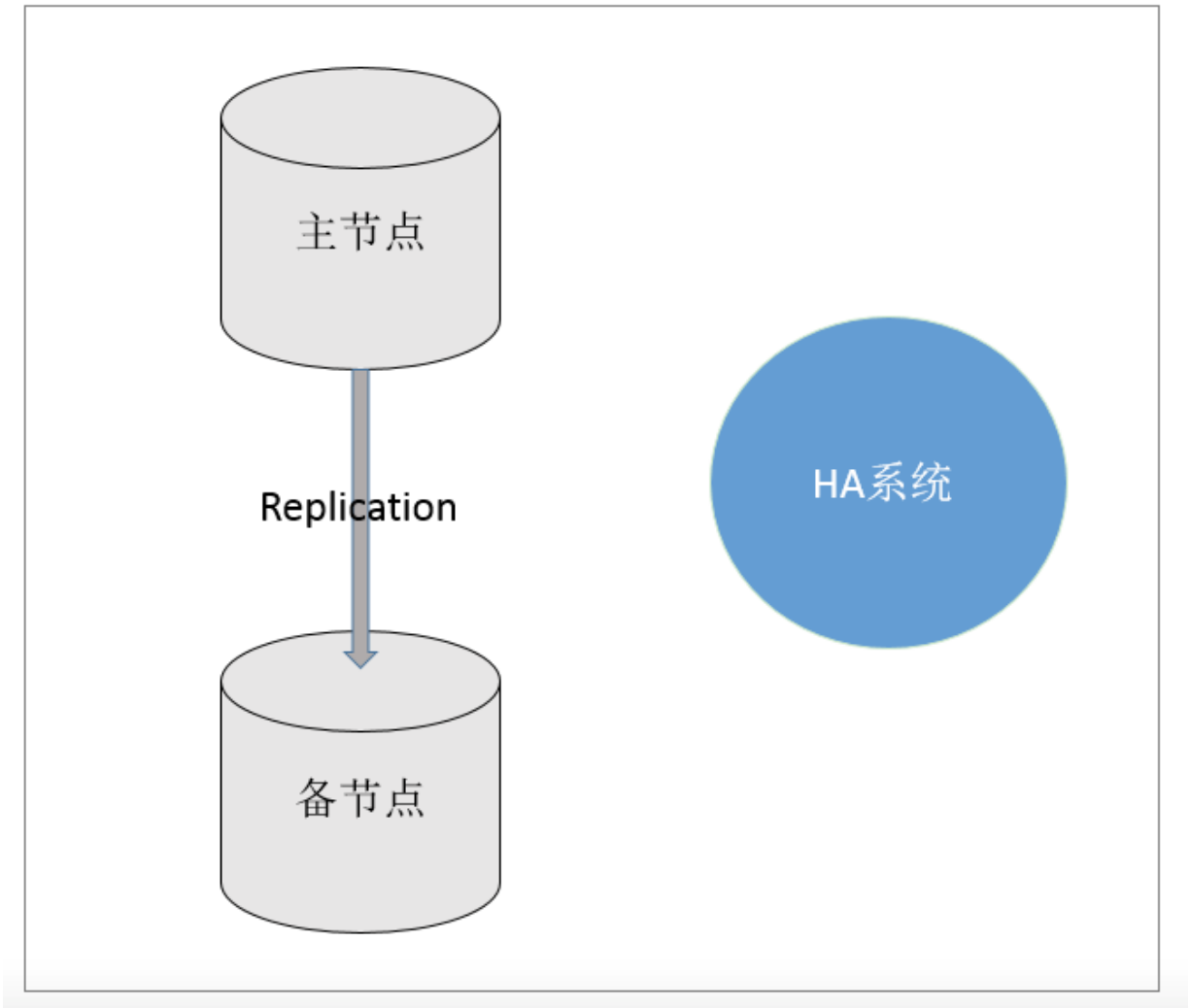
默认情况下，云数据库Redis版实例被设置为允许任何IP访问。您可以通过控制台的**实例信息** > **修改白名单**页面来添加IP白名单规则。IP白名单的更新无需重启实例，不影响使用。同时，IP白名单支持设置IP地址或IP段。

4.2.2 高可用架构

云数据库 Redis采用主从（Replication）模式搭建。

主节点提供日常服务访问，备节点提供 HA 高可用，当主节点发生故障，系统会自动在30秒切换至备节点，保证业务平稳运行。

图 1: Redis主从（Replication）架构



可靠性

- 服务可靠

  采用双机主备架构，主备节点位于不同物理机。主节点对外提供访问，您可通过 Redis 命令行和通用客户端进行数据的增删改查操作。当主节点出现故障，自研的 HA 系统会自动进行主备切换，保证业务平稳运行。

- 数据可靠

  默认开启数据持久化功能，数据全部落盘。支持数据备份功能，您可以针对备份集回滚实例或者克隆实例，有效的解决数据误操作等问题。

兼容性

云数据库 Redis 版标准版在 Redis 2.8基础上进行开发，100%兼容 Redis 协议命令。自建的 Redis 数据库可以平滑迁移至 Redis 标准版。并且提供数据传输工具（DTS）可以进行增量的 Redis 迁移，保证业务平稳过渡。

阿里云自研

- 故障探测切换系统（HA）

  阿里云 Redis 服务封装 HA 切换系统，时时探测主节点的异常情况，可以有效解决磁盘 IO 故障、CPU 故障等问题导致的服务异常，及时进行主备切换从而保证服务高可用。

- 主备复制机制

  阿里云针对 Redis 主从复制机制进行了定制修改，采用增量日志格式进行复制传输。当主备复制中断后，对系统性能及稳定性影响极低，有效避免 Redis 原生复制的弊端。

4.2.3 软件升级

- 云数据库Redis定期提供数据库软件的新版本。
- 版本升级是非强制性的，只有您主动要求，才会升级到指定版本。
- 当云数据库Redis团队评估您的版本存在重大安全隐患时，会主动通知业务安排时间进行升级。云数据库Redis团队将会全程支持进行升级过程。
- 云数据库Redis升级过程通常在五分钟以内完成，升级期间可能有数次数据库连接闪断并且存在1分钟的实例只读。在应用程序正确配置了数据库连接重连（或连接池）的情况下，不会对应用程序造成明显的影响。

4.2.4 服务授权

在没有您授权的情况下，阿里云的售后团队和云数据库Redis开发团队只能查看Redis实例资源、资费和性能相关的信息（例如，云数据库Redis实例的购买时间和到期时间，云数据库Redis的CPU、内存、存储空间的使用情况等。）

只有在获得您授权后，阿里云的售后团队和云数据库Redis开发团队才能在您指定时间查看和修改云数据库Redis实例的配置信息（例如，云数据库Redis实例的IP白名单、审计日志等。）

在任何情况下，阿里云的售后团队和云数据库Redis开发团队不会主动变更云数据库Redis实例的链接信息（包含连接地址和数据库账号。）

5 企业级分布式应用服务EDAS

5.1 产品介绍

企业级分布式应用服务（Enterprise Distributed Application Service，简称EDAS）是企业级互联网架构解决方案的核心产品，充分利用阿里云现有资源管理和服务体系，引入中间件成熟的整套分布式计算框架（包括分布式服务化框架、服务治理、运维管控、链路追踪和稳定性组件等），以应用为中心，帮助企业级客户轻松构建并托管分布式应用服务体系。

企业级分布式应用服务（EDAS）从多个角度提供方案来保障服务安全可靠，包括但不限于：

- 权限控制：服务鉴权，应用管控鉴权
- 高可用架构：EDAS Console负载均衡，无中心化服务注册中心

5.2 安全和可靠性方案

5.2.1 权限控制

服务鉴权

在EDAS的远程过程调用协议（Remote Procedure Call Protocol，简称RPC）服务调用过程中，每一次调用都需经过服务鉴权，服务的发布和订阅过程都需要进行服务鉴权。

- 被访问资源：服务名称，服务分组
- 访问对象：用户账号（包括主账号、子账号）

需要鉴权的服务权限类型包括：

- 发布权限
- 订阅权限
- 调用权限

当某用户在 EDAS 创建服务分组（ServiceGroup）时，系统默认为该用户创建与该服务分组（ServiceGroup）相关的服务发布、订阅和调用的权限。当用户为该服务分组（ServiceGroup）创建发布者或者订阅者时，EDAS管控平台会对该服务分组（ServiceGroup）进行鉴权；当用户使用该服务分组（ServiceGroup）进行服务发布和订阅时，EDAS也会对该服务分组（ServiceGroup）进行鉴权；当用户创建EDAS实例后，对于发布在EDAS上的服务调用同样需要鉴权。

鉴权流程使用阿里云AccessKey和SecretKey机制进行签名验证以及资源的权限验证。

应用管控鉴权

在EDAS平台上，明确区分主、子账号。主账号可以通过分配和授权，将指定应用的管控权限授予给指定子账号。

授权范围包括：

- 机器资源
- 负载均衡资源
- 应用资源

5.2.2 高可用架构

EDAS Console负载均衡

EDAS Console实例由多个机器构成，通过负载均衡设备以单一连接串的方式提供服务。当一个EDAS Console机器出现故障时，流量可在短时间内切换到其他EDAS Console机器上。整个切换过程对用户透明，应用代码无需变更，应用进程无需重启。

EDAS无中心化的服务注册中心

服务注册中心由阿里巴巴集团自主研发的软负载产品构成，RPC框架无中心单点。所有的服务调用过程，发布者和消费者均直连，无须经过中间层处理。服务注册中心仅用于服务的发现（即服务的发布和订阅）。

5.2.3 软件升级

- EDAS定期为您提供软件的新版本。
- 版本升级是非强制性的，只有您主动要求，才会升级到指定版本。
- EDAS升级通常在一个工作日内完成，升级期间不会出现任何应用业务异常。

5.2.4 服务授权

License管理

在您通过合法手段购买EDAS产品后，根据合同内容，EDAS项目组会向您发放一个包含产品系列、产品版本、节点规模、使用期限等信息在内的License。

您在第一次使用 EDAS产品时，输入该License后可正常使用。

6 分布式关系型数据库DRDS

6.1 产品介绍

分布式关系型数据库服务（Distributed Relational Database Service，简称DRDS）专注于解决单机关系型数据库扩展性问题，具备轻量(无状态)、灵活、稳定、高效等特性，是阿里巴巴集团自主研发的中间件产品。DRDS高度兼容MySQL协议和语法，支持分库分表、平滑扩容、服务升降配、透明读写分离和分布式事务等特性，具备分布式数据库全生命周期的运维管控能力。

分布式关系型数据库服务（DRDS）从多个角度提供方案来保障服务安全可靠，包括但不限于：

- 访问控制：数据库账号、IP 白名单
- 高可用架构：计算节点高可用、RDS/MySQL高可用

6.2 产品安全和可靠性方案

6.2.1 访问控制

数据库账号

DRDS支持类MySQL的账号和权限体系，具备GRANT、REVOKE、SHOW GRANTS、CREATE USER、DROP USER、SET PASSWORD等相关指令和功能。

创建DRDS数据库时，默认可以指定一个具有所有权限的账号。用此账号可以创建一个或者多个新的账号。

- 权限支持粒度：数据库和表级别（暂不支持全局、列级别）
- 支持相关联的八个基本权限项：
  CREATE、DROP、ALTER、INDEX、INSERT、DELETE、UPDATE、SELECT
- 支持“user@’ host’ ” 用户形式，对host进行访问匹配验证。

> **注意:** 但当业务机器处于专有网络VPC内时，因技术原因无法获取 IP，建议改成“user@’ %’ ” 。

IP 白名单

DRDS提供了IP白名单来实现网络安全访问控制，支持为每个DRDS数据库单独设置IP白名单。

默认情况下，DRDS实例被设置为允许任何IP访问。您可以通过控制台的**DRDS数据库 > 白名单设置**页面来添加IP白名单规则。IP白名单的更新无需重启DRDS实例，不影响使用。同时，IP白名单支持设置IP地址或IP段。

注意: 当业务机器处于专有网络VPC内时，因技术原因无法获取IP，建议去掉IP白名单。

6.2.2 高可用架构

DRDS Server自动切流

DRDS实例由多个DRDS Server（服务进程）构成，通过负载均衡设备以单一连接串的方式提供服务。当一个DRDS Server出现故障时，流量以秒级切换到其他DRDS Server上。整个切换过程对用户透明，应用代码无需变更，应用进程无需重启。

只读实例自动切流

DRDS支持读写分离功能，可以在控制台的 **DRDS数据库 > 读写分离**页面进行配置。将部分读流量分配到备实例上。DRDS将识别出读SQL命令请求，并按照配置的比例下发到主、备RDS/MySQL实例执行，达到读写分离的目的。分配了读流量的备实例称为只读实例。

如果配置了多个只读实例，当其中一个只读实例出现故障（具体表现为无法连接）时，DRDS会自动收回故障实例上的读流量，并按照剩余正常只读实例的读流量比例重新分配执行。

只读实例自动切流过程对用户透明，应用无需重启。当所有只读实例都不可用时，为了防止主实例压力过大，将仍然按比例分配读SQL命令请求到只读实例和主实例上，并对分配到只读实例上的读SQL命令请求快速报错。

说明: 所有写SQL命令请求和事务均自动下发到主实例上执行，与只读实例的可用性无关。

6.2.3 软件升级

- DRDS定期提供数据库软件的新版本。
- 版本升级是非强制性的，只有您主动要求，才会升级到指定版本。
- 当DRDS评估您的版本存在重大安全隐患时，会主动通知业务安排时间进行升级。DRDS团队将会全程支持进行升级过程。
- DRDS升级过程通常在五分钟以内完成，升级期间可能有数次数据库连接闪断。在应用程序正确配置了数据库连接重连（或连接池）的情况下，不会对应用程序造成明显的影响。

6.2.4 服务授权

在没有您授权的情况下，阿里云的售后团队和DRDS开发团队只能查看DRDS实例资源、资费和性能相关的信息（例如，DRDS实例的购买时间和到期时间，DRDS Server的CPU、内存、存储空间的使用情况，SQL 审计日志等。）

只有在获得您授权后，阿里云的售后团队和DRDS开发团队才能在您指定时间查看和修改DRDS实例的配置信息（例如，DRDS实例的IP白名单、DRDS Server的流量权重等。）

在任何情况下，阿里云的售后团队和DRDS开发团队不会主动变更DRDS实例的链接信息（包含连接地址和数据库账号。）

7 消息队列MQ

7.1 产品介绍

消息队列（Message Queue，简称MQ）是阿里云商用的专业消息中间件，是企业级互联网架构的核心产品。基于高可用分布式集群技术，搭建包括发布订阅、消息轨迹、资源统计、定时（延时）、监控报警等一套完整的消息云服务，帮您实现分布式计算场景中所有异步解耦功能。

MQ由阿里巴巴集团中间件技术部自主研发，是原汁原味的阿里集团中间件技术精华之沉淀，是性价比最高、最可靠的企业级消息中间件产品。

消息队列（MQ）从多个角度提供方案来保障服务安全可靠，包括但不限于：

- 访问控制：资源与账号管理，应用场景可覆盖跨账号资源授权、子账号授权、账户黑名单控制等功能。
- 高可用架构：多节点集群化部署、多副本数据存储、主备复制方式保证服务高可用以及数据高可靠。

7.2 产品安全和可靠性方案

7.2.1 访问控制

鉴权

消息队列的安全访问控制包括以下几个要素：

- 被访问资源：消息主题（Topic）
- 访问对象：用户账号（包括主账号、子账号）

消息队列的权限类型包括：

- 发布权限
- 订阅权限

当用户在消息队列上创建消息主题（Topic）时，系统会默认为该用户创建与该Topic相关的消息发布与消息订阅的权限。当用户为该Topic创建发布者或者订阅者时，消息队列管控平台会对该Topic进行鉴权；当用户使用该Topic进行消息发送和消息订阅时，MQ Broker服务也会对该Topic进行鉴权。当用户创建MQ Broker实例后，对于发布在Broker上的服务调用同样需要鉴权。

鉴权流程使用阿里云AccessKey和SecretKey机制进行签名验证以及资源的权限验证。

账号黑名单

在提供鉴权机制的同时，消息队列提供了“用户黑名单”来实现安全访问控制。

消息队列可以通过设置用户黑名单的方式，控制非法用户（恶意攻击等不合理使用的用户）对MQ进行访问，从而阻止其对消息队列进行恶意的攻击。

授权管理

每个资源有且仅有一个所有者，资源 Owner，且必须是云账号（或者专有云账号）。资源Owner对资源拥有完全控制权限。资源Owner不一定是资源创建者（例如，阿里云RAM子账号被授予创建资源的权限，该RAM子账号创建的资源归属于主账号，该RAM子账号是资源创建者但不是资源Owner。 ）

在未经过资源所有者授权的情况下，其他主账号或者 RAM 子账号是无法对资源进行访问的。资源所有者可以对资源进行授权或者取消授权。

授权方式包括以下两种：

- 消息队列支持在管控平台上为资源Owner提供授权功能，包括跨账号授权和子账号授权。
- 在阿里云访问控制平台上，主账号对子账号进行授权时，可根据不同的授权策略，为子账号赋予不同的权限。

7.2.2 高可用架构

Broker集群部署

为保证服务的可用性，消息队列支持多节点集群化部署。

- 同一个Region内支持跨机房、跨地域部署，提高网络问题的容灾能力。
- 支持集群化部署方式，提高部分节点不可用时的容灾能力。
- 消息重发机制。当某个节点服务不可用时，迅速切换到其他节点，提高集群的服务能力。

Broker主备部署

- 为保证数据的可靠性，消息队列支持一主多备部署方式。主备复制模式包括同步复制和异步复制。
- MQ Broker同时支持主备间自动切换。一旦主机出现不可用时，根据主备切换策略进行切换，迅速恢复服务。

7.2.3 软件升级

升级采取兼容性灰度升级策略，总体上分为MQ客户端升级、MQ Broker升级和MQ Name Server升级。

- 消息队列定期提供客户端的最新版本，每个 Release 版本都有详细说明。客户端升级为非强制性升级，只有您主动要求，才会升级到指定版本。
- MQ Name Server采取灰度发布的策略，采用分批升级的策略，升级过程对用户透明。同时，Name Server升级会保持与客户端以及Broker的兼容。
- MQ Broker升级采取灰度发布的策略，采用分批升级的策略，升级过程对用户透明。同时，Broker升级会保持与客户端的兼容。

8 企业实时监控服务ARMS

8.1 产品介绍

业务实时监控服务（Application Real-Time Monitoring Service,简称ARMS）是一款端到端一体化实时监控解决方案的PaaS级阿里云产品。通过ARMS，您可以基于海量的数据迅速便捷地通过定制化为企业带来秒级的业务监控和响应能力。ARMS产品孵化于阿里巴巴内部业务，经过长时间考验，目前已被广泛用于阿里巴巴内外的商品、物流、风控和各种云产品的各类业务监控场景。

8.2 产品安全和可靠性方案

8.2.1 访问控制

鉴权

当您访问ARMS Console实例时，ARMS将为每个用户创建独特的账户机制保障安全性。

鉴权流程使用阿里云 AccessKey 和 SecretKey 机制。具体流程如下：

1. 用户使用在ARMS控制台上创建的凭证（包含一对AccessKey和SecretKey），并用该凭证订购相应的服务获得通过。
2. 在访问的时候时，使用ARMS SDK对您的任何请求进行以下鉴权操作：
   a. 先进行签名验证，确认消息没有被篡改。
   b. 进行鉴权，检查相应的AccessKey是否有权限调用该服务。

HTTPS

ARMS Console提供将服务开放成为HTTP协议的能力，同时可以在链路上支持SSL，即HTTPS。

说明: 虽然ARMS Console提供了应用到Console之间的连接加密功能，但是SSL需要应用开启服务器端验证才能正常运转。另外，SSL也会带来额外的CPU开销，ARMS Console实例的吞吐量和响应时间都会受到一定程度的影响，具体影响视您的连接次数和数据传输频度而定。

8.2.2 数据隔离

ARMS保障从计算到存储之间用户数据完全隔离。

计算隔离

对于每个用户的任务，ARMS会在JStorm集群中不同的拓扑结构（Topology）进行计算。每个Topology属于且仅属于一个用户，而每个用户视情况可以拥有不同规模不同数量的Topology，以支撑其计算需求。

对于不同用户的任务，如出现异常，例如数据量过大导致内存泄露或者其他潜在程序问题，由于计算隔离，可保障不同用户之间不会受到任何干扰。

存储隔离

对于每个任务的数据集，ARMS后台在列式存储中使用单独的表来存放。其中，每张表都设置有单独的数据生命存放周期（Time to live，简称TTL），以及对应的协处理器（Coprocessor），保证任何用户的数据在升级或销毁时都不会对其他任何用户造成影响。

8.2.3 QoS保障策略

ARMS提供数据计算、存储写入、API 读取等多方面端到端的服务质量（Quality of Service，简称QoS）控制。

数据计算策略

- Topology配置限制策略

  ARMS针对每个Topology上的数据流入模块（Storm Spout）设置了数据计算的最大内存值和最大CPU核数。这样可以保证在数据量突发暴涨的情况下造成的影响有限，不会对整个ARMS计算集群造成影响。

  如果您需要应对可预估的数据量暴涨情况，则建议在突发情况下增加 Topology配 置，增大相应的任务并发数以及内存数，以解决计算规模问题。

- Topology公共任务保护策略

  对于公共任务，ARMS把多个任务放置在一个Topology进行计算。为了防止一个任务拖垮所有Topology，ARMS设置了流量和维度限制上限，防止一个任务过量计算而拖垮整个集群。

存储写入策略

ARMS针对每个用户对存储的写入做了限制，限制大小跟Topology的数量和大小成正比。

这样可以防止当数据量暴涨时，某个用户将整个存储写挂。列式存储的写入限制不会造成用户的数据丢失。当写入实际速度小于数据实际的产生速度时，计算任务会相应地被阻塞，也就是实时监控数据会有相应的延时产生。

如果您需要解决可预见的列式存储写入瓶颈，只需要增加Topology的数量和大小，以提高存储的写入QoS限制。

API读取策略

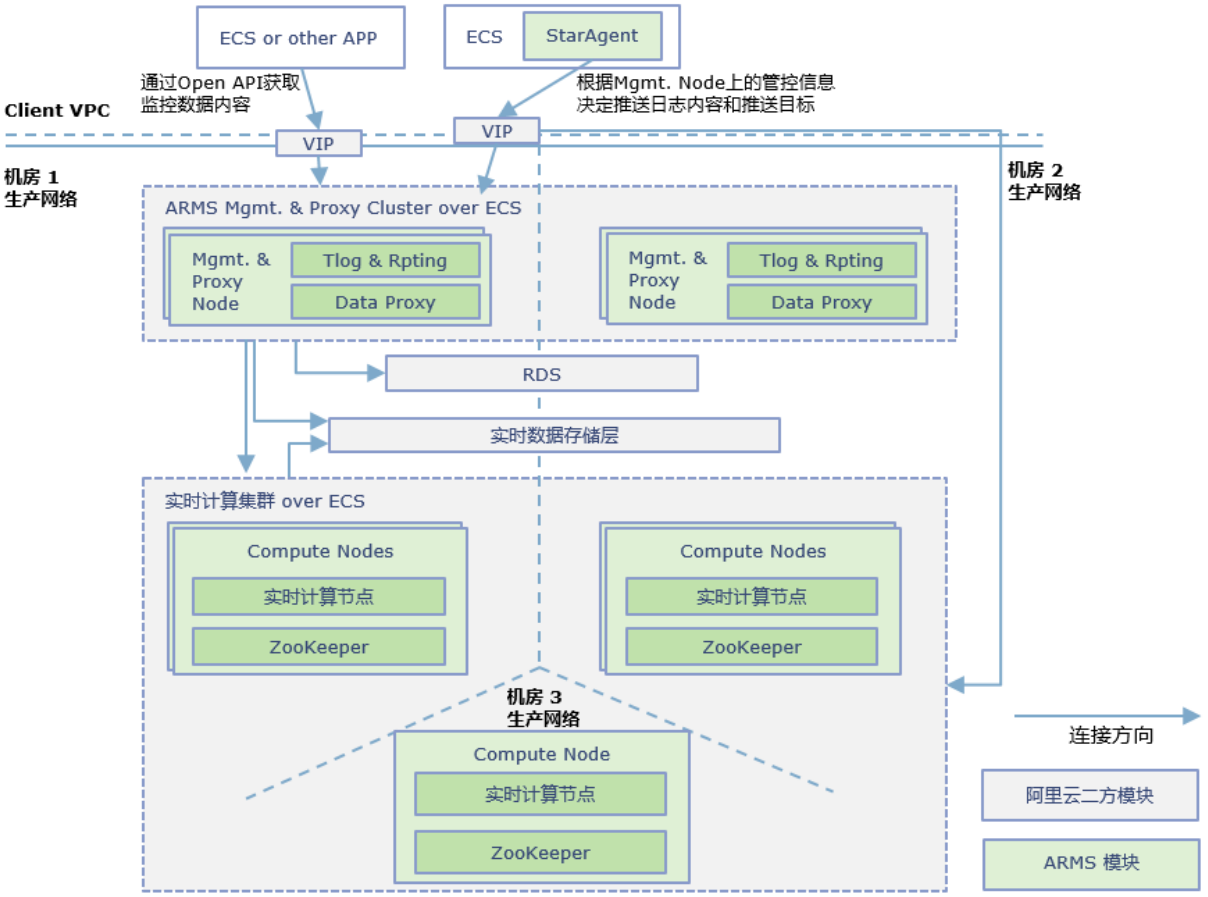
ARMS上的存储数据结果可以通过API被用户或者其他应用读取。为了防止应用将ARMS的API代理节点（ARMS Console）或存储读挂，ARMS 对 API 读取速度做了限制。默认每个用户的API读取速度限制是100。

如果有大量的API读取峰值需求，可通过调整相应QoS策略来解决。

8.2.4 高可用和容灾保障

ARMS的系统架构达到总体无单点，支持双站点双活式容灾。

图 2: ARMS系统架构图



站点内高可用

ARMS系统架构无论在数据输入层、计算层、存储层、还是 API 调用层，都采用高可用架构，且理论上支持无限横向扩张。

- 数据输入层在专有云或公共云上均可对接简单日志服务（SLS）及消息队列服务（MQ），在可用性和性能上均无单点。
- 计算层采用JStorm集群。当计算节点宕机，JStorm主控节点（Nimbus）将根据Zookeeper信息自动在健康的Supervisor上重新启动计算任务。
- 存储层采用列式存储，本身三份备份。即使两台节点同时宕机，数据仍可用。
- API调用代理层均为无状态节点，通过DNS/VIP将负载均衡分散到各代理节点，整个代理层可横向扩展。

站点间容灾策略

ARMS机房发生整体宕机时，理论上会存在分钟级恢复时间目标（RTO）和几乎为零的恢复点目标（RPO）。

- 数据接入层、计算层、数据代理层需要故障切换到容灾站点时，理论上会存在些许RTO。
- 列式存储通过多活部署，集群直接跨站点。灾难时，理论RTO为零。数据通过集群内数据复制功能，理论上能达到近乎为零的RPO 。

8.2.5 软件升级

- 升级采取兼容性灰度升级策略。
- 升级总体上就是ARMS Console升级。列式存储、JStorm集群以及数据接入层几乎无需或很少需要升级。
- ARMS Console升级采取灰度发布的策略，分批进行升级。

9 全局事务服务GTS

9.1 产品介绍

全局事务服务（Global Transaction Service，简称 GTS）是一款高性能、高可靠、接入简单的分布式事务中间件，用于解决分布式环境下的事务一致性问题。

传统的事务主要是指单机数据库的原子性、一致性、隔离性、持久性（Atomicity、Consistency、Isolation、Durability，简称ACID）特性。GTS在支持分布式数据库事务的基础上，将事务的范围拓展到了多种资源，让分布式环境下的多个资源的操作加入事务的范畴，赋予了分布式资源操作ACID特性。

GTS是阿里云商用的企业级产品，产品稳定性及可用性完全按照阿里巴巴内部标准来实施，应用只需要极少的代码改造和配置，即可享受分布式事务带来的便利。

全局事务服务（GTS）提供了多样化的安全加固功能来保障使用的安全，包括但不限于：

- 访问鉴权
- 流量控制
- 数据多份落盘
- 集群容灾

9.2 产品安全和可靠性方案

9.2.1 访问控制

鉴权

当用户创建GTS分组后，在使用GTS服务时需要鉴权。

鉴权过程使用阿里云AccessKey和SecretKey机制，具体流程如下：

1. 获取用户在GTS控制台上创建的分组ID，并获取阿里云给用户派发的一对AccessKey和SecretKey。
2. 在访问GTS服务端的时候，使用GTS SDK对事务调用消息进行相应的签名。
3. 在消息到达GTS服务端后，GTS服务端会进行如下鉴权操作：
   a. 进行签名验证，确认消息没有被篡改。
   b. 进行鉴权，检查相应的AccessKey和分组ID是否有权限调用该事务分组。

9.2.2 流量控制

在提供访问控制机制的同时，GTS提供流量控制来实现事务创建消息的访问控制，控制的量可以通过Diamond配置中心进行配置。根据用户的实际业务使用需求，GTS可以配置流量控制量的大小，保证机器不会因为事务量的突然暴增而出现异常。

9.2.3 数据多份落盘

GTS将事务的中间状态多份落盘存储在多台机器上，经过严格断电测试，严格保证数据一致性。

9.2.4 集群容灾

GTS的服务端是以集群的形式存在，一组GTS服务由三台物理机组成，提供高可用服务。即使突发事件造成集群中某一台机器崩溃，GTS仍然能够提供一半的服务能力。并且，重启崩溃的机器后，这台机器将恢复之前的事务场景，继续进行事务的处理。

9.2.5 软件升级

- 升级采取兼容性升级策略。
- 升级总体上分为GTS client升级和GTS server升级。
- 升级GTS client前一般要先升级GTS server。
- GTS server升级向下兼容，并在内部集成了灰度策略，升级时对用户完全没有影响。
- GTS client升级时，可以采用灰度发布策略，对于GTS client的分批升级不会造成整体的事务不可用。

10 云服务总线CSB

10.1 产品介绍

云服务总线（Cloud Service Bus，简称CSB）应用于专有云、公共云、以及混合云场景，实现跨系统、跨协议的服务互通。主要针对需要进行管理和控制，包括安全授权、流量限制的系统间服务访问和对外开放场景。

越来越多的企业组织需要以API方式把自己的核心业务资产贯通整理并开放给合作伙伴、或者让第三方的应用整合，以发掘业务模式、提高服务水平、拓展合作空间。云服务总线（CSB）面向专有云和专有域，帮助企业在自己的多个系统之间、或者与合作伙伴以及第三方的系统之间实现跨系统跨协议的服务能力互通。各个系统以发布、订购服务API的形式相互开放，并对服务API进行统一管理和组织、围绕API互动，实现企业内部各部门之间、以及企业与合作伙伴或者第三方开发者之间业务能力的融合、重塑、和创新。

云服务总线（CSB）提供了多样化的安全加固功能来保障服务调用的安全，其中包括但不限于：

- 访问控制：鉴权、IP黑白名单、防止重放、HTTPS
- 流量控制：整体流控、服务级别流控
- 容灾：服务注册中心多级存储容灾、Broker 集群容灾、Broker 同城容灾（多可用区实例）

10.2 产品安全和可靠性方案

10.2.1 访问控制

鉴权

当用户创建CSB Broker实例后，对于发布在Broker上的服务调用都需要鉴权。

鉴权过程使用阿里云AccessKey和SecretKey机制，具体流程如下：

1. 需使用在CSB控制台上创建的凭证（包括一对AccessKey和SecretKey)订购相应的服务。
2. 在访问的时候，CSB SDK对服务调用消息进行相应的签名。在消息到达Broker后，CSB Broker会进行如下鉴权操作：
   a. 进行签名验证，确认消息没有被篡改。
   b. 进行鉴权，检查相应的AccessKey 是否有权限调用该服务。

IP黑白名单

在提供鉴权机制的同时，CSB提供了IP黑白名单来实现网络安全访问控制。

默认情况下，CSB Broker实例被设置为允许任何IP访问。您可以通过控制台的来添加IP黑白名单规则。IP黑白名单的更新无需重启CSB Broker实例，不会影响使用。

- IP黑名单：支持将恶意用户的IP或者IP段加入黑名单，以阻止该用户的访问，从而阻止其对Broker进行攻击。
- IP白名单：提供了跳过鉴权控制的机制。

防止重放

CSB提供了防止请求重放的功能。默认该功能关闭，可以根据相应安全要求打开。

防止请求重放提供了如下机制：根据请求时间戳和Broker上的时间进行对比，如果超过设定的阈值，则拒绝超时的请求。

HTTPS

CSB Broker提供将服务开放成为HTTP协议的能力，同时可以在链路上支持SSL，即HTTPS。同时，在Broker之间级联的时候，也支持HTTPS协议。

> **说明:** 虽然CSB Broker提供了应用到Broker之间的连接加密功能，但是SSL需要应用开启服务器端验证才能正常运转。另外，SSL也会带来额外的CPU开销，CSB Broker实例的吞吐量和响应时间都会受到一定程度的影响，具体影响视您的连接次数和数据传输频度而定。

10.2.2 流量控制

流量控制用于防止CSB Broker集群过载，防止出现极端大量请求同时到来的情况下出现的集群雪崩现象。

CSB Broker支持两个级别的流量控制：实例级别和服务级别。

实例级别流控

实例级别流控用于对每一个CSB Broker实例进行流量控制，对Broker单一实例进行过载保护。在当前的流量超过实例级别的最大流量的时候，Broker会拒绝新的请求直到流量低于最大流量。

服务级别流控

服务级别流控用于对后端接入的服务进行保护，防止通过CSB进行调用的服务流量过大，对后端的服务造成较大的影响。

服务级别的流控支持对接入CSB的每一个服务单独的设置流量控制，CSB会把服务级别的流量控制分配到每一个对应的CSB Broker实例对应的服务上。对每一个Broker而言，在当前的流量超过该实例对应服务的最大流量的时候，Broker会拒绝新请求对该服务的访问，直到流量低于最大流量。

10.2.3 容灾

服务注册中心多级容灾

为了保证数据完整可靠，CSB的服务注册中心采用了多级容灾的策略。

- 服务注册信息保存在数据库里，利用数据库提供的备份功能，可以采用每天或者定时备份的策略。
- CSB设计了缓存结构，Broker只和缓存的注册中心进行通信，不直接使用数据库，以防止Broker集群重启对数据库造成冲击。
- CSB Broker自身也设计了本地的内存缓存，在无法连接到缓存注册中心的时候，可以采用本地数据继续提供服务。

Broker集群容灾

CSB Broker采用无状态的设计策略，全部的状态都放在注册中心中，因此CSB Broker具备线性可扩展的特性。整个集群具备当单台机器不可用时，整体服务质量不受到大的影响的能力，并且支持集群灰度升级（需要接入层进行配合。）

10.2.4 软件升级

- 升级采取兼容性灰度升级策略。
- 升级总体上分为注册中心升级和CSB Broker升级。
- 注册中心升级采取灰度发布的策略，采取分批升级策略，不会造成注册中心的整体不可用。
- 注册中心在升级的过程中可能会造成短暂的服务更新延迟，在升级完毕后，可以恢复。
- 注册中心升级保持向下兼容。
- CSB Broker升级同样采取灰度发布的策略，采用分批升级策略。