阿里云

云上自动化运维（CloudOps）
白皮书

阿里云弹性计算10+位技术专家共同撰写
CARES模型，五大维度看CloudOps成熟度
帮助企业落地最佳实践，发挥云的最大潜力

扫码加入

ECS 粉丝群

阿里云开发者“藏经阁”

海量电子书免费下载

目录

版本说明

版本	描述	更新日期
V1.0	2021 云上架构与运维峰会前夕	2021-10

前言

背景

DevOps 已经成为了近年来运维的主要趋势之一，越来越多的企业在拥抱和实践 DevOps 文化，通过 DevOps 理念的践行/实践，企业提高了研发效率，缩短了业务从研发到上线的周期，从而实现了真正意义上的技术变革。

同时越来越多的企业也都逐步开始借助云计算来帮助企业实现 IT 云化。云计算海量的弹性优势、丰富的标准化云产品、自动化工具、自服务的服务模式不仅能帮助企业加速数字化转型，而且还增强了企业业务发展的敏捷性，使得企业在市场竞争中保持先进性。

DevOps 本质是为了协同公司内多个不同团队快速朝着同一个业务目标前进，而衍生出来的一系列流程和自动化工具，强调的就是组织和业务的敏捷性。而云计算本身就是为了服务于按需取用随取随用的业务场景，标准统一的云产品使用方式，自助取用的服务模式，充分体现了云计算的敏捷性。

DevOps 与云浑然天成，但我们也发现有很多企业即便将 DevOps 搬到云上，却并没有充分利用云的优势。

为了更好地发挥 DevOps 和云的双重敏捷特质，业界需要有一套更为成熟和体系化的理念。因此，我们提出一种新思路——CloudOps（云上自动化运维）。

CloudOps 是传统 IT 运维和 DevOps 的延展，通过云原生架构实现运维的再进化，充分帮助企业降低 IT 运维成本、提升交付速度和系统灵活敏捷度、增强系统可靠性，构建更加安全可信开放的业务平台。

要强调，CloudOps 不等于单纯的 Cloud + DevOps 或者 DevOps on Cloud，而是需要将 DevOps 和云有机结合，才能收获更大价值。

基于阿里云服务几千家企业客户的经验，我们撰写了这本《云上自动化运维白皮书》（简称《CloudOps 白皮书》），并在其中提出了 CloudOps 成熟度模型，建议从五个方面建设和评估 CloudOps 能力，期许提供云上自动化运维的最佳实践和参考，帮助客户更加充分地利用云带来的优势，进一步提高业务交付质量。

这套模型可以简称为 CARES 模型，即成本管理（Cost）、自动化能力（Automation）、可靠性（Reliabilty）、弹性（Elasticity）和安全与合规（Security）五个方面。

CloudOps 成熟度模型是对企业研发效能的评估，也是对云厂商产品能力和自服务能力的评估，评估云厂商是否有提供足够全面的工具和能力，让客户便捷地实现这些功能。

DevOps 理念的兴起

为了提升软件的研发效率和交付质量，越来越多的企业已经开始使用 DevOps，同时将 DevOps 作为一种企业文化来践行。

DevOps 理念囊括团队文化、组织协同和研发运维多方面：DevOps 文化能够促进团队的共同协作，更有效地管理基础架构的稳定性，更快、更好的执行应用程序；通过持续交付和持续构建来提升企业的数字化转型；通过消除研发、运维之间的利益差异和差距，专注于端到端的能力交付和系统建设，让软件交付的全生命周期中的开发、部署、维护和扩展等各个步骤更加有效率，降低故障次数和故障时长，充分体现了以产品和效率为中心来进行软件开发和交付。

DevOps 模型定义了几个成功的关键分组，这些对于应用成功和提升效率非常有帮助。

- 敏捷开发的过程管理：实现高效协同，定义人与人之间的协同，业务和技术之间的协同，组织和团队的治理以及需求管理等多个要素和因子。
- 持续交付：通过定义更好的 CI/CD 工具来完成灵活变更和持续交付部署，更好地构建环境以及提升可视化能力。
- 技术运营提升：可以快速构建所需要的基础设施和资源保证，对于监控预警、问题发现、容量管理、变更管理和成本管理等，提供体系化的支撑。

CloudOps 的优势：云与 DevOps 的双重价值

DevOps 给应用软件开发带来了极大的便利性。由于云服务有着“软件定义一切”和弹性敏捷等特点，跟 DevOps 的理念非常一致，这两者带来的优点是非常类似的。

采用 DevOps 理念，有以下四个方面优点和目的：

- 降低整体成本支出
  - 通过协同不同的团队来降低人员冗余。
  - 通过持续的自动化投入，例如 Infrastructure as Code（IaC）、Ops as Code 等，来降低人工成本。
- 提升整体的交付速度
  - 通过敏捷形态和自动化构建极大的提升应用交付速度。
- 提升灵活性
  - 更快的交付速度意味着可以快速的调整和提升效率。
- 增强系统的可靠性
  - 通过标准化、工具化建设以及自动化实施可以避免人为错误。
  - 组织的自助管理，降低不必要的错误沟通以及问题排查的 MTTR（平均修复时间，Mean time to repair）。

使用云服务管理的也有类似的优点：

- 降低整体成本支出
  - 企业可以减少对硬件的采购和基础资源运维的相关人员，以及不用考虑各种物理资源的差异化。
  - 云计算让企业按需消费资源，并且不用为业务峰值提前购置大量固定资产，只需要迅速根据业务变化来创建资源。
  - 通过合适的选型和资源应用形态搭配选择合适的付费和资源类型。
- 提升整体的交付速度
  - 通过弹性能力，企业可以在几分钟到几个小时内快速响应资源创建或者释放，而不需要长达数天甚至数周的流程。
- 提升灵活性
  - 能够快速的适应市场需求或者各种运营活动的极致需求。
- 增强系统的可靠性
  - 云服务天然提供了高可用的系统设计，例如多可用区、备份恢复、热迁移等手段，降低物理资源故障带来的损失。
  - 通过云服务厂商提供的工具和服务化能力，可以大大降低系统构建成本，同时通过专业的问题排查和诊断服务，可以大幅降低部分问题和故障的排查难度。
  - 能够创建更具弹性、安全性和标准化的系统。

综上，我们能看到云和 DevOps 之间有着许多相似的优点，将 Cloud 和 DevOps 有机结合，将能发挥更大的价值，包括上述提到的降成本、提升交付速度、更灵活、提升系统可靠性等。

云和 DevOps 是浑然天成的，结合了云计算的 DevOps，不仅仅可以提升效率和优化 TCO，同时它更加可以它还使我们的系统能够在不断变化的复杂环境中更可靠地工作。近几年来，我们欣喜地看到越来越多的企业和个人开发者将自己的测试环境和生产环境迁移上云。

云上运维的特点与挑战

然而，将传统的 DevOps 直接搬到云上，并不能充分利用云的优势，获得 1+1 等于或者大于 2 收益，因为相比于传统的 DevOps 的运维模式，云上自动化运维的模式和思维仍然有着不小差异。这也是部分企业上云之后，建立一套云原生自动化运维体系的挑战。

- 操作对象的差别
  - 传统运维，直接操作的是物理的计算、网络、存储的硬件。
  - 云端运维，大多通过软件暴露接口或则 OpenAPI 来进行操作经过抽象的资源。
- 资产和资源的区别
  - 传统运维模式操作的都是企业的资产，需要充分压榨提升单机的利用率和使用率，并需要提前很久规划资源。
  - 云端运维天然有就弹性的属性，除了提升单机利用率，还可以 On-demand 地获取资源和释放，充分的利用 OpenAPI 和应用分组来管控资源。
- 统一化规模化差异
  - 传统运维一般操作的规模相对较小，管理的机房相对明确和有限。
  - 云端运维可以快速的通过资源的弹性能力轻松的管理数百台甚至更大规模跨多个机房的服务器。
- 安全可审计
  - 云端操作高频切很多是自动化的任务，操作来源和对象相对复杂，对操作审计和操作来源和报警的时效性要求比较高。
  - 云端提供的服务可以将服务通过一条命令直接暴露在公网之中，需要更多的设计和思考安全和网络规划能力来降低系统风险。
  - 高频的可编程自动化运维需要有比较好的审计和问题追踪能力，避免越权和不容易被追踪的问题。

总结

相信基于以上论述，读者们更了解了我们为什么认为云端的 DevOps，需要有一套更为成熟和体系化的理念，才能帮助企业在云时代更好地发挥云和 DevOps 的优势。我们提出的新思路——CloudOps（云上自动化运维），就是在此背景下诞生。

阿里云弹性计算团队内部，联合十多位专家，共同编撰了一套 CloudOps 成熟度模型。如前所述，这套模型的几个维度我们可以称为 CARES，即成本管理（Cost）、自动化能力（Automation）、可靠性（Reliabilty）、弹性（Elasticity）和安全与合规（Security）五个方面，来评估企业的 CloudOps 成熟度。每个维度如何理解，我们将在下面章节展开。

CloudOps 的主要衡量维度和定义

DevOps 的趋势

参考 Gartner 2021 Top10 Technology Trends Impacting DevOps 和 Puppet State of DevOps Report 2021，我们观测到了 DevOps 实践有下面的趋势形态:

- 越来越多的企业在公有云中使用 DevOps，但是绝大部分企业都认为自己没有发挥和使用 DevOps 的核心能力。
  - 65%的企业已经在公共云中 DevOps。
  - 只有 20%的企业认为自己充分用到了 DevOps 的全部能力。
  - 自动化已经成为 DevOps 实践中最高优先级的任务，通过结合云的优势和自动化能力，可以进一步推进 DevOps 的能力演进。
- 微服务架构的实施带来巨大的便利，也带来了新的挑战。
  - 服务拆分导致应用激增，统一简单的可观测性是个巨大的挑战。
  - 更多的应用拆分和并行的任务开发模式可能会导致更多的故障点。
  - 应用之间的依赖关系对于单个应用的可靠性和可用性有了更高的要求。
- 分布式应用的复杂性非常高。
  - 网络延迟，容错，消息序列化，不可靠的网络和底层资源，异步性，版本控制。
  - 可测试性和异步调用让链路复杂。
  - 更高更难的 DevOps 要求。
- 自助服务（Self-Service） 已经是企业的一个迫切的需求。
  - 预测到 2025 年，75%的大型企业将建立自助服务基础设施平台，以帮助快速进行产品创新，而 2020 年这一比例为 15%。
  - AI 和 ML 将推动 DevOps 快速进化。

CloudOps 的定义与主要衡量指标

CloudOps 是传统 IT 运维和 DevOps 的延展，通过云原生架构实现运维的再进化，充分帮助企业降低 IT 运维成本、提升交付速度和系统灵活敏捷度、增强系统可靠性，构建更加安全可信开放的业务平台。

DevOps 已经在组织文化、产品、流程和工具有比较详细的定义，即通过敏捷组织和高效的持续集成持续发布，实现业务高质量的快速交付。

因此，本文将不会讨论 DevOps 关于研发支持体系、需求管理、任务管理、代码管理等这内容；而是更多地从公共云上如何进行自动化运维和自助服务的角度，基于我们服务阿里云百万客户的经验，着重梳理了衡量 CloudOps 成熟度的五大维度：

自动化能力

云计算核心就是自动化的运维能力，通过软件定义计算、存储、网络，来实现高级的可编程能力，从而避免人工配置的错误，充分实现可定制的自动化能力。而公有云的服务模式要求云厂商提供的云产品和云服务都必须是统一标准的，即所有云产品和云服务都可以通过 OpenAPI 进行调用，从而实现完全自动化的能力。

弹性能力

云计算另外一个巨大技术红利就是弹性能力，针对计算、网络、存储、安全等基础资源，充分的发挥资源池化和分时复用的价值，通过弹性能力帮助客户应对业务的高峰，充分降低社会成本和企业运营的 IT 成本，提升资源的利用率，可以极速实现资源到应用的水平或者垂直升级，通过秒级到分钟级扩缩容能力，完成计算力的创建和释放。

高可用能力

云计算天生就是为提升可靠性和可用性而设计的：通过大规模数据中心、多数据中心技术，实现数据中心同城灾备，通过对硬件层的虚拟化，来降低和规避物理硬件故障对客户的影响，通过成熟高可用的服务来降低系统的复杂性。为了进一步提升应用的可观测性和问题的排查能力，云平台还会提供比较多的自助服务来做问题的排查和解决。

安全和合规能力

云上的安全涉及到多方面，包括底层技术设施和应用层的。这里我们主要讨论跟底层资源相关的。

首先第一个便是网络安全。区别于传统的 IDC，云计算为了对租户进行隔离，一般会构建私有网络或者专有网络，通常我们称为 VPC（Virtual Private Network）。VPC 相较传统网络有更好的灵活性、易用性和安全性，并且暴露了更多的能力来提升网络扩展性。它允许用户按需规划、定义自己的网段划分和路由规则，将传统的路由器交换机抽象成软件，并暴露给最终用户使用。VPC 良好的扩展性，让用户能够构建简单可信的网络配置，实现企业级复杂的网络环境。对于 VPC 的规则设置和配置，都将大大影响网络安全性。

另外，DevOps 中操作审计和追踪是非常重要的能力，在 CloudOps 中亦然，云计算平台一般也会提供相应的为您提供面向资源和操作的配置历史追踪、配置合规审计等能力，帮助客户轻松实现基础设施的自主监管，确保持续性合规。

成本和资源量化管理

云提供了大规模的资源创建和变配策略，也提供了多种多样的付费和计费手段以及方便灵活的变配方法，如何选择合适的资源规格和付费方式是非常重要的；由于其方便灵活的特性，往往会有类似停机不收计算类资源费用，以及折扣非常低的抢占式实例，特别是按需创建资源和关停不需要的计费资源，需要我们有良好的成本和资源量化管理习惯和能力。

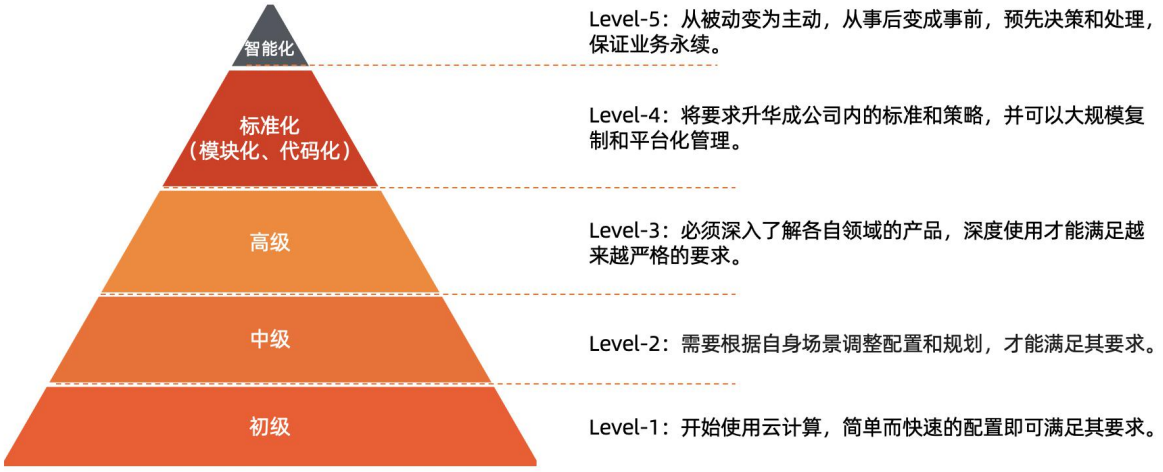
CloudOps 成熟度模型整体和等级说明

递进的成熟度模型

云上运维是一个从简单到复杂、从成长到成熟的过程管理，以降低成本提高效率为核心目标。在现实中，根据使用者的上云状态、使用规模等，其云上运维的思路都不尽相同，但其规律确是有迹可循。

创业公司从第一天开始就在云上部署其生产环境服务客户，而对于已经存在 IT 投入的公司来说，则需要花费更长的时间逐步上云。无论哪种场景，其运维需求都会持续存在，随着业务不断发展，运维也日益复杂，因此有效地规划和制定运维策略和方法就非常重要，该模型致力于提供一个已被大规模验证的最佳实践供各位参考。

成熟度模型等级说明



成熟度的5个等级

在后续章节中，我们会针对 CloudOps 成熟度模型的五大维度进行拆解，并对每个维度进行分级，不同等级说明如下：

初级

代表了该类别的初步使用状态，企业刚开始考虑到这些特性（自动化、可靠性、安全和合规等），在实践中探索相关的云产品，在使用中以默认的使用方式为主，或接受其推荐的配置模式，简单开启相关的功能，这样即可满足当时的需求，所以定义为初级。

该等级往往可以对应到相关云产品的最基础的能力，默认配置、快速配置为主。

中级

随着业务的不断发展，云上规模的逐步扩大，企业对于这些特性（自动化、可靠性、安全和合规等）的需求不断提升，初级的使用方式已经不能够满足需求，默认的配置不再符合具体的场景，此时需要深入了解各个相关云产品的特性，然后做出更多的、符合自身场景的配置和规划。

该等级往往可以对应到相关云产品的中阶能力，需要根据场景调整配置和规划。

高级

企业云上的资源已经具备了一定的规模，对于这些特性（自动化、可靠性、安全和合规等）的需求更加严苛，提高效能和降低成本甚至成为核心诉求之一。此时对于这些特性（自动化、可靠性、安全和合规等）将会一一拆解并进行调研和分析，如何可以最大化地利用云上已有的能力成为其实施的关键，对于各项指标的要求不断接近行业领先水平。

该等级往往可以对应到相关云产品的高级能力，甚至包括很多 beta 能力，以便满足自身严苛的高需求。

标准化

该阶段的特点是要求企业在常规运维操作上具备极佳的可复制能力，其各项流程和指标都成为公司内的一个基本标准，并且已技术的方式将其落地，可以从一个模块复制到其他模块，新建的模块自动获得以上的高标准能力。公司内部对于这些特性（自动化、可靠性、安全和合规等）已经形成了共识和策略，并持续审视和演进。

该等级通常要求一个中心化的标准和策略落地平台，以规范其作业流程和配置，并持续迭代。

智能化

从分离的 Dev 和 Ops 到一体化的 DevOps，再从 DevOps 到更适合云上场景的 CloudOps——Cloud 所特有的规模效应将会使 CloudOps 更快实现智能化运维 AIOps，实现更多的主动化运维（SmartOps，注：AI 和 ML 以前的智能化运维简称为 SmartOps，即主动和自动化相结合的运维方式），以期更加智能地处理出现的故障，甚至在真正的故障出现之前预先处理故障以避免故障的发生。

智能化要求的不仅仅是自动，而是有了部分学习、预测的能力。相同的智能化同样会出现于其他特性中（弹性能力、成本控制等），提供更加准确的预测能力有利于进一步降低成本，而更加智能的成本预测也会帮助财务更早地做出有支撑的决策等。

该等级通常从使用具备 AI 能力的各项云产品开始，然后逐步地从局部的、单个的场景开始，逐步演进成系统化的、全面的智能化，最终实现真正的智能化。

自动化能力

自动化能力是 DevOps 也是云最核心的能力，通过充分的自动化可以更好地结合人员、流程和技术，得以向客户持续交付价值，大幅提升企业的敏捷性和执行效率。

云服务相比 IDC，最大特点是通过提供大量的 OpenAPI 将底层的计算、网络、存储和应用等能力暴露出来，以更简单的形态提供服务，让客户通过编程来定义和完成高阶服务能力。

以阿里云为例，OpenAPI 相当于飞天操作系统的内核 API，通过使用和编排 OpenAPI，可以构建基于云操作系统的高阶能力，从而形成资源管控到应用交付的全生命周期自动化，通过将任务脚本化提升执行效率是自动化的常见做法。

自动化能力分类

通过 OpenAPI 编排能力，可以方便地组合出多种形态的自动化。为了更好的管理资源，企业的系统要能够与云平台高度自动化地集成。当前常见的云自动化可以分为下面三类:

- Infrastructure as Code: 基础设施即代码的理念，希望通过构建可编程的基础设施，来实现资源的高效自动化配置。典型的产品如: Terraform、阿里云的资源编排（ Resource Orchestration Service）、AWS CloudFormation。
- Pipeline as Code: 通过脚本和引擎让原本独立运行于单个或者多个节点的任务连接起来，实现单个任务难以完成的复杂发布流程。
- Policy as Code: 通过自动化代码来管理权限管控或者安全策略，提升自动化能力。

Infrastructure as Code（IaC）

通过 Infrastructure as Code 的工具和产品，可以快速完成云资源的自动化部署服务，一键部署系统所依赖的云资源。该种方式有下面的多种优点，充分降低成本和提升成功率。

- 可以快速地重复部署，如部署测试、预发和生产环境，尤其适合需要在多地域部署的情况。
- 可以减少环境之间的偏差，有助于将部署过程和结果尽量标准化，减少因为环境偏差引入的系统问题。
- 一键部署，极大地提高了部署效率，即可以更快地发布系统和应用。
- 一键销毁，在测试完成或蓝绿部署时，一键销毁所有资源，减少资源浪费，彻底净除环境，以便于下次部署。
- 复用已有的最常见的网站、ML 架构、最佳实践，方便您尽快地进行测试、验证或建立产品雏形。
- 操作都集成访问控制和操作审计，确保基础设施的安全。

虽然 IaC 有非常多的优点，但是从头直接撰写代码来配置和构建系统来实现 Infrastructure as Code，有很高的门槛。目前市场上部分产品也提供了方便的策略来实现自动化生成部署脚本，主要包括下面两种策略来相对简化 as Code 的过程。

- 从图形化的拖拽界面来生成资源描述和依赖关系的定义。
- 通过选择一些已有的资源并根据资源的关联关系来逆向生成配置文件。

- 快速通过一些已有的模板来修改克隆。

Pipeline as Code

随着运维操作越来越复杂，在日常的发布和构建过程中，我们定义了非常多的复杂任务，如何将这些任务通过串行或者并行地方式来高效执行，并达到终态或者回滚是非常关键的；可以通过解构将复杂的任务分解为相对独立的 job 或者 pipeline，交给工作流引擎进行编排执行。

- 通过将执行任务的上下文清晰梳理和可视化依赖关系，能简化之前复杂事务带来的影响。
- 让各个 JOB 单元原子化，可以高效的完成单元任务并降低单个任务的复杂度。
- 通过任务抽象进行功能的维护和扩展。

针对 CICD 和运维的流水线定义，有非常多的产品和工具，典型的产品如：Pipeline as Code With Jenkins 、GitOps、阿里云运维编排（Operation Orchestration Service）、AWS System Manager Automation、Azure Automation 等一系列产品。

Policy as Code

云上的权限模型通常有两类，根据人来定义操作，和根据资源来设定权限，前者的比例是绝大多数的。云上的访问控制都是基于访问策略（Policy）进行的，然后将 Policy 和用户（User）、角色（Role）绑定起来使用。因此云上的权限定义即编写对应的 Policy，通常的 Policy 都是由操作和资源两部分组合而成的一个配置文件，JSON 格式是最常见的类型。相对于其他的 As Code 来说，Policy as Code 的方式是最自然的。

在用户、角色之外，云上有少数资源是可以直接定义权限控制的，如阿里云 OSS、AWS S3，可以直接定义一个资源被谁访问，甚至是可以定义为对互联网公开，允许所有人控制，这类访问是将 Policy 直接和资源绑定的。同样的，Policy 的具体内容也很容易记录在 JSON。

As Code vs Real Code

As Code 的创新也在一直持续，开始出现了一些 Real Code 的产品，如 Pulumi、AWS CDK、阿里云 CDK，这类产品的出现是为了解决编写 JSON 文件的弊端，尤其是 JSON 文件无法进行调试，导致任何错误都需要在运行态才能发现，因此使用真正意义上的 Code 就可以很好地弥补这个缺陷，方式和理念都和 As Code 是一致的。对于更喜欢 Code 的人来说，Real Code 类的产品是个不错的选择。

Self-Service

按照 NIST 对云计算的定义，第一条要求就是 on-demand self-service，用户能够自助获取所需计算资源或服务，而不必和服务供应商交互。

除了上面提到的自动化部署、自动化交付、自动化编排，广义的自助服务也包括下面的部分，借助于前文提到的开放 API 体系和事件体系暴露更多的服务能力。

- 准确实时的事件：一般来说，云厂商会通过 API 或者消息队列，将不同的事件类型、事件名称、事件含义、事件状态和事件等级等信息，推送给用户。用户可以根据事件的形态和类别，进行主动或者自动的做处理和响应，例如实现预测机器可能会发生宕机，通过发布系统主动运维触发的实例重启（Reboot）类型事件来规避不可预期的风险。除了风险类事件，也可以将创建启停资源等事件实时推送，方便做事件的自动化处理和诊断。
- 监控预警：基础的云监控服务可用于获取阿里云资源的相关指标，探测互联网服务可用性，以及针对指标设置警报。包括事件体系自动化完成系统的扩缩容。
- 自助问题诊断和修复：通过暴露 API 和自助诊断的能力，客户可以借助平台提供的能力检测云资源是否有问题并修复问题。通过自动化工具，用户可以进行问题诊断，自助工具会告知用户问题的根因，进而一键修复问题，解决问题时间缩短至分钟级。

阿里云自动化运维的主要产品

除了前面提到的云产品 OpenAPI，阿里云也提供了相应的 As Code 的工具类云产品来简化云端运维。我们着重介绍以下 3 个产品，并分享一些我们推荐的使用方向：

资源编排 ROS

资源编排 ROS（Resource Orchestration Service）是最基础的 nfrastructure as Code 云资源管理自动化服务，供用户免费使用。您可以遵循 ROS 定义的模板规范（JSON/YAML）编写资源栈模板，在模板中定义所需的云计算资源（例如：ECS 实例、RDS 数据库实例）、资源间的依赖关系等。ROS 的编排引擎将根据模板自动完成所有资源的创建和配置，实现自动化部署及运维。它有以下的特点:

- 免费服务托管：通过全托管的自动化执行，您可以在模板中定义阿里云资源和配置参数，并说明资源间的依赖关系，然后创建资源栈，从而管理一组资源。
- 支持多账号跨地域部署：同一个 ROS 模板，在多个阿里云账号中跨地域的进行自动化部署。
- 标准化部署：通过将部署环境标准化，减少不同环境的差异，将环境的配置沉淀到模板中，节省部署成本。
- 结果可视化呈现：ROS 通过自动化部署，帮助您通过控制台或 API 清晰查看部署结果， 避免人工逐个检查部署进程。
- 偏差检测：您可以使用偏差检测来识别在 ROS 之外的资源变更，并采取纠正措施，使资源再次与模板定义同步。
- 服务集成：通过集成访问控制（RAM）提供统一的身份认证，您无需单独建立用户认证体系。通过集成操作审计服务（ActionTrail）审查所有的运维操作，包括 ROS 本身。
- 兼容 Terraform 模板：对于已经使用 Terraform 的用户来说，ROS 可以作为托管版本来使用，避免了使用 CLI 的一些问题。

运维编排 OOS

运维编排 OOS 是阿里云提供的免费的 Pipeline as Code 的云上自动化运维服务，能够自动化管理和执行任务。您可以通过模板来定义执行任务、执行顺序、执行输入和输出，然后通过执行模板来完成任务的自动化运行，可以让一份任务轻易的支持多地域和跨地域执行。

- 任务引擎: 可以通过任务编排来执行复杂的运维任务，甚至组合多个场景来完成任务。
- 事件驱动: 一个事件发生时，触发一个运维动作。例如，当某 ECS 实例的 CPU 使用量过高时，为了防止业务中断，可以接收一个云监控的事件，通过这个事件定义任务，例如自动重启。事件驱动场景可以提供主动运维支持，免去中间的人为因素，提高运维效率。
- 定时任务: 支持定义 Cron Job 定时执行所定义的运维动作。
- 批量操作: 批量地执行运维命令，即需要针对多个目标（如 ECS 实例）进行常规操作，以确保业务的正常和平滑运行，并保持业务的健康状态。

RAM & STS

访问控制（RAM）是阿里云提供的管理用户身份与资源访问权限的服务。允许在一个阿里云账号下创建并管理多个身份，并允许给单个身份或一组身份分配不同的权限，从而实现不同用户拥有不同资源访问权限的目的。我们建议在云上避免使用主账号，而通过子账号来进行管控和权限控制。

- 管理 RAM 用户及其密钥：管理每个 RAM 用户及其访问密钥，为用户绑定多因素认证（MFA）设备。
- RAM 用户的访问权限：控制每个 RAM 用户访问资源的权限。
- 集中控制 RAM 用户的资源访问方式：确保 RAM 用户在指定的时间和网络环境下，通过安全信道访问特定的阿里云资源。
- 集中控制云资源：对 RAM 用户创建的实例或数据进行集中控制。当用户离开组织时，实例或数据不会丢失。
- 单点登录管理（SSO）：支持与企业身份提供商（IdP）进行用户 SSO 或角色 SSO。

STS（Security Token Service）是阿里云提供的一种临时访问权限管理服务。通过 STS 获取可以自定义时效和访问权限的临时身份凭证，即安全令牌（STS Token）。有权限的 RAM 用户可以使用其访问密钥调用 AssumeRole 接口，以获取某个 RAM 角色的 STS Token，从而使用 STS Token 访问阿里云资源。

- 通过使用 STS Token，减少长期访问密钥（Accesskey）泄露的风险。
- STS Token 具有时效性，可以自定义有效期，到期后将自动失效，无需定期轮换。
- 可以为 STS Token 绑定自定义权限策略，提供更加灵活和精细的云资源授权。

自助服务套件

阿里云系统事件：主要分为两大类，一类是系统事件，影响实例运行状态的有计划的底层运维事件，或非预期维修事件，一般这类事件是需要主动或者被动做处理的。另一类是通知类事件，这类事件可以由业务方决定是不是监控和消费，通知的事件可以是系统事件（包括运维事件和异常故障）、实例状态变化、挂载和卸载数据盘、创建快照等。通过设置事件通知，您还可以为事件配置消息处理中间件，实现事件驱动的自动化运维。

阿里云自助诊断系统：分为四个主要的模块，一是智能诊断库，它包含了不同云资源、不同操作类型对应的异常信息集合；二是智能方案匹配，主要通过分析异常对应的云资源类型及操作类型，建立二维模型，然后将二维模型输入到智能诊断库，利用最优的查找匹配算法，匹配问题的原因及解决方案；三是诊断展示，主要用来展示诊断状态及查看诊断方案，可以通过 API 和控制台获取结果；四是诊断问题反馈，接收对诊断方案的反馈，通过反馈来完善和调整提升诊断准确度。通过实时提交问题诊断，可以降低问题的反馈沟通成本，以及缩短问题的处理时间，提高了问题处理效率。

自动化能力分级

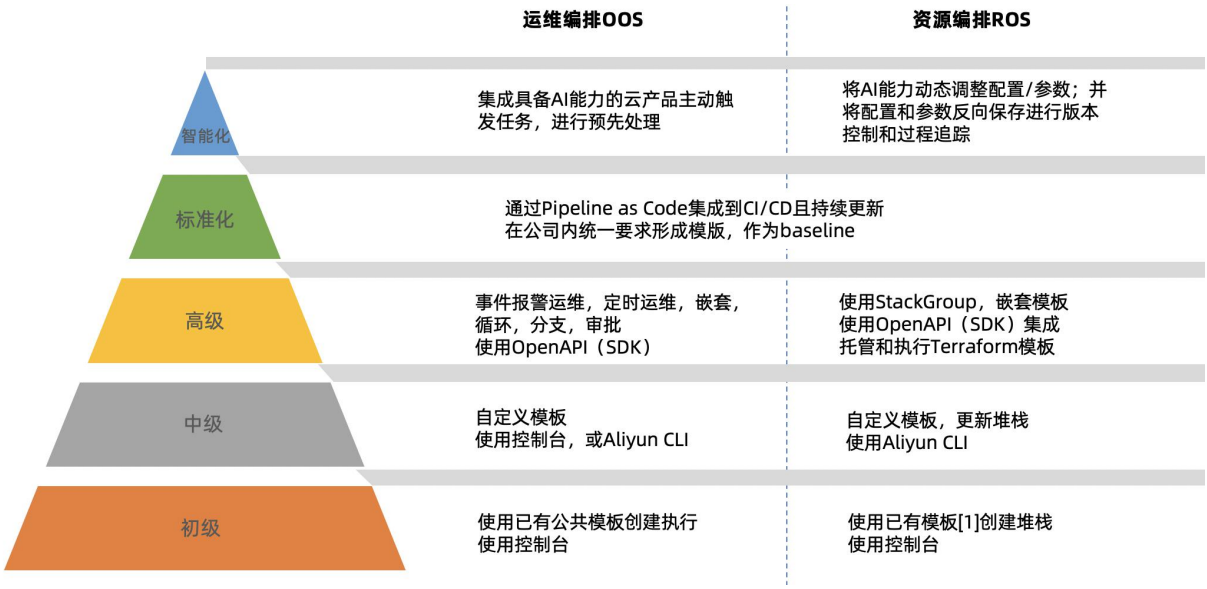
从云产品控制台操作到基于 OpenAPI 开发常用高频任务、到结合自动化编排引擎来完成高阶的运维能力，结合不同的的场景和任务、云端运维的自动化特点和客户用云的形态差异，我们将自动

化的能力分为如下 5 个级别。

级别	自动化运维
1-初级	• 少量的零散的自动化 在网站上手工完成大部分操作
2-中级	• 半手工/半自动化 • 使用可复用的模板来取代重复的构建 • 开始使用 CLI 工具和 OpenAPI 集成
3-高级	• 可以完全自动化地构建一套新环境 • 自动化已经完成体系化建设 • 企业运维人员会使用通过 OpenAPI 进行日常运维工作，同时自行开发运维系统
4-标准化	• 企业的运维系统已具备平台化、模版化和代码化的能力，可根据自身的运维需求定制化开发系统。与此同时，运维人员已具备使用具备模板化的产品来实现运维工作的标准化和自动化 • 可以完全自动化地构建一套新环境和系统 • 新构建的环境都具备相同的配置，允许少数例外 • 环境的更改也通过模块统一有序进行
5-智能化	• 运维自动化程度达 100% • 从代码仓库出发可以从零构建一套新环境和系统（即 GitOps） • 所有的配置都存在代码管理软件中 • 任何环境配置和系统的改变都是从 Code 触发 • 新环境在流水线上自动同步，无须手工添加

自动化能力等级和云产品功能映射

通过下图示意能力等级和云产品功能的映射关系，希望提供进一步的参考，实际使用中可以根据具体的系统设计和业务特点进行合理化调整：



弹性能力

弹性能力是云计算的最重要的能力之一，结合超大规模的资源配置能力，快速实现分钟级的资源需求供给，以满足不同规模场景的弹性需求。

弹性可以满足业务日常所需的资源变化，也给用户带来的更多的好处，比如成本降低、高可用等。在云上使用弹性能力可以提升企业业务的灵活性和稳定性。

垂直弹性

在单体应用、独立应用、有状态应用等场景下，随着业务不断升级和变化，需要快速升级硬件以应对业务变化。比如企业需要进行一些类似大促等运营活动时，不论是计算能力还是网络资源，其需求量都会大于之前的水平。

这种情况下，企业可以对系统进行配置升级，如升级到性能更高的实例规格、提高带宽配置、扩大磁盘大小等，当业务和运营活动结束时，负载整体恢复原有状态，出于成本的考虑，企业可以对云服务器进行配置降级，如降低实例规格、降低带宽值等。

这时可以通过云提供的控制台和 OpenAPI，对云资源的计算、网络、存储资源，进行升降配。垂直伸缩能力通常包括修改资源的规格（例如 vCPU 和内存）、公网带宽配置、数据盘大小、计费方式等。

在业务需求增长时，升级资源规格以保证计算能力；在业务需求下降时，降低实例规格以降低成本。这样的基础上，系统通过定时或者监控指标来监控业务需求的变化，自动触发变配任务，能够多样化满足云上的业务需求。执行变配任务时，系统会自动完成停止目标资源实例、调整实例规格、启动目标资源实例等一系列操作。

水平弹性

在分布式应用、无状态应用、大型应用等场景下，固定数据的云资源已经无法满足业务快速和剧烈的变化，在这样的情况下，要使用云上的水平弹性能力来解决问题，需要依托于云上高效的弹性能力及大规模的资源补给，进行快速扩容。

此时，企业可以借助于云上的弹性伸缩服务，根据业务需求和策略自动调整实例数量。在业务需求增长时，弹性伸缩自动增加实例，来保证计算能力；在业务需求下降时，弹性伸缩自动减少实例，来节约成本。弹性伸缩不仅适合业务量不断波动的应用程序，同时也适合业务量稳定的应用程序。

弹性伸缩一般应该具备以下能力：

- 丰富灵活的扩缩容模型：伸缩规则用于触发伸缩活动或者智能设置伸缩组边界值，该章节后面会进一步详细介绍常见的扩缩容模型。
- 资源配置信息来源：自动扩容时，系统会根据资源配置信息来源创建资源实例，同时信息来源还要支持启动模板，提高复用性。
- 完善的通知告警机制：自动发送消息至监控告警服务或消息服务，以便及时掌握资源变化动态，进一步实现自动化管理。
- 生命周期管理：在资源扩缩容过程中，可以暂停自动触发的伸缩活动，为客户保留一段自定义操作的时间，直至生命周期挂钩超时结束。
- 监控能力指标矩阵：包含多种指标类型，例如 CPU 使用率、内存占用比、TCP 连接数、出入字节数、Disk 使用率、Java 线程数等，以及自定义指标。

随着客户弹性成熟度的提高，对扩缩容方式也会提出不同的要求，例如在每天中午 12 点开始，业务需求明显增加，需要设置定时任务，在每天 12 点创建 20 台资源以应对业务高峰。目前几种常见的动态扩缩容模型有：

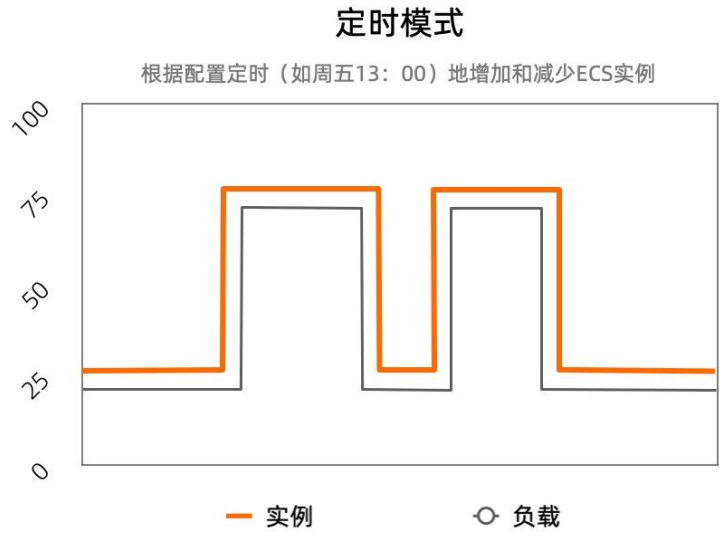
- 定时模式：创建定时任务，在指定时间执行资源扩缩容。
- 指标模式：基于资源的性能指标（如 CPU 利用率、网络流量均值）创建报警任务，当指标数据满足指定的报警条件时，触发报警并执行资源扩缩容。
- 固定数量模式：设置最小/最大期望资源数量，当实例数量低于下限/超过上限时，系统会自动添加/移出资源，使得资源数量等于下限/上限。
- 健康监测模式：定期检查计算资源的运行状态，如果发现一台计算资源未处于运行中状态，则判定为不健康并移出该资源。
- 手动模式：手动进行弹性伸缩，包括手动添加、移出或者删除已有的资源。

弹性能力成熟度模型

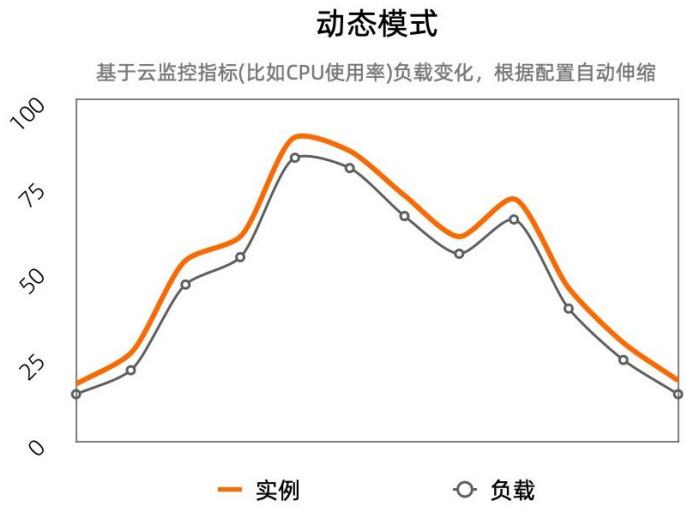
一般来说，用户在上云的不同阶段，弹性需求也有所不同。

在第一阶段，资源容量充足，完全可以满足流量和业务需求，客户可以不考虑弹性能力。如果有流量激增的情况，用户可以利用已有的资源启动模板去快速创建资源容量，实现扩容的目的。

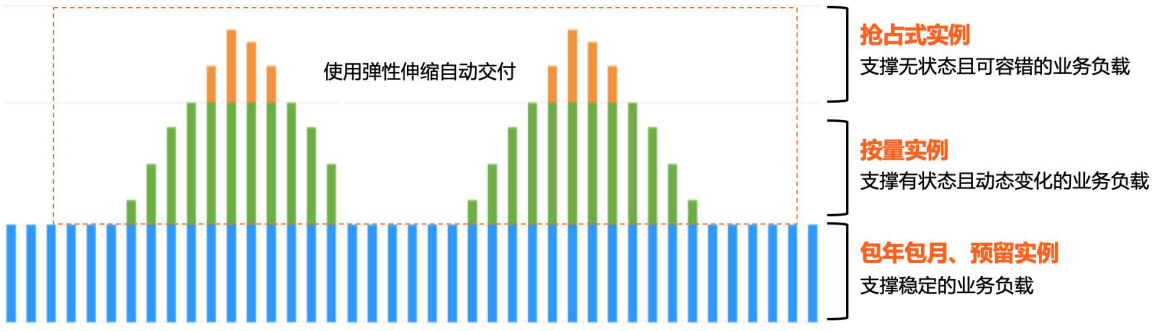
第二阶段，用户有了成本意识，希望减少冗余闲置的资源，于是从垂直和水平维度，通过自动化的手段调整资源容量。例如如下图，在每周五 13 点业务高峰期，自动把 4 核 8G 的服务器配置升级成 8 核 16G，或者自动从 2 台服务器扩容到同样配置的 4 台服务器。



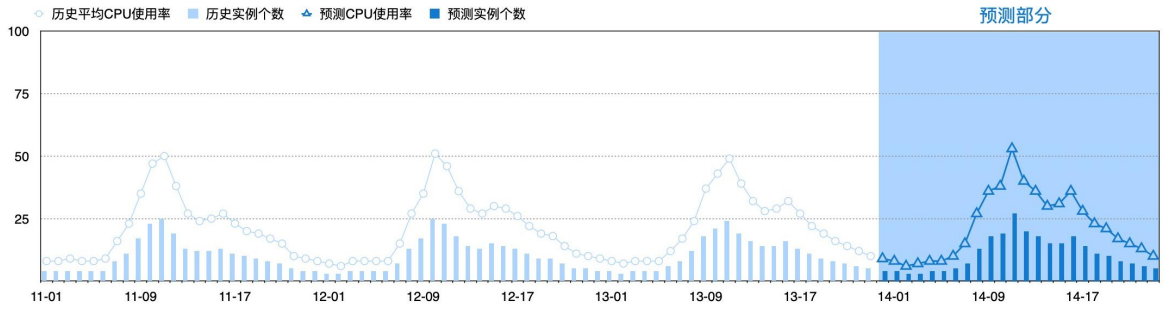
第三阶段，用户可以自动调整系统所有计算、内存、磁盘等资源容量，而且能够精细化控制扩缩容的资源数量，使削峰填谷后的资源使用曲线更加匹配实际的资源需求量。例如当 CPU 利用率在 20%到 40%之间，使用 4 台服务器；当 CPU 利用率在 40%到 60%之间，使用 8 台服务器；当 CPU 利用率在 60%以上，使用 16 台服务器。



第四阶段，对系统的弹性能力建立了一套符合自身业务情况的标准化管理策略，使用多种组合弹性模式来应对流量高峰和成本节省计划。例如对业务基座资源包年包月购买，其他高峰期资源使用弹性伸缩自动购买与释放；根据高峰时间规律，配置定时扩容模式，提前应对峰值；配置目标追踪模式，应对突发流量。



第五阶段，用户开始使用一些智能化手段来进行更精确的扩缩容动作。例如下图所示，基于机器学习，可以通过分析历史监控数据预测未来监控指标值，并支持自动创建定时任务，智能设置资源容量的边界值。



所以，建议的弹性能力成熟度模型如下：

级别	成本和资源量化意识
1-初级	• 容量充足或变化不大，对弹性无强烈需求，无需通过弹性能力调整资源容量
2-中级	• 部分资源实现了自动化变配和扩容缩容，如定时模式，监控触发模式
3-高级	• 整个系统的容量都能够主动、自动地按需调整 • 调整的颗粒度能够被精细化控制
4-标准化	• 标准化应用和服务的自动化容量管理策略 • 每个应用和服务都能够按需随时开启标准化策略 • 基于标准化策略可以调整更适合的自动伸缩规则
5-智能化	• 能够根据历史数据和实时数据叠加，进行更精确，更准确的自动扩缩容

可靠性

云计算提供了从数据中心、硬件、数据、自助服务等多个层次的可靠性构建能力。

对比于传统的 IDC，云计算的超大规模的数据中心，以及多可用区支持，让用户可利用云资源低成本、高扩展、高可靠地快速构建同城容灾、异地容灾等服务（和数据）的高可用方案。

云计算通过虚拟化等技术对客户屏蔽了底层物理硬件，与此同时云厂商通过虚拟化、热迁移等技术，来减少甚至规避物理硬件故障导致的服务受损，进一步提升了用户服务的连续性以及高可用。

为了进一步提升可靠性，云还提供了丰富的可观测性以及自助服务能力。基于此，用户可以构建多层次的可观测性能力，并实现服务故障自动发现、自动诊断、以及自愈能力，同时通过混沌工程提前发现生产环境潜在风险。

构建多地容灾架构

云计算有天然的规模化和可靠性优势。云厂商不仅提供了超大规模的数据中心，同时提供了全球多地域服务，每个地域是完全独立的数据中心，多个地域之间完全独立。而每个地域又有多个可用区，每个可用区之间电力和网络互相独立。

对于可靠性要求较高的应用，通常会做同城多机房部署，避免由于单机房的网络、电力等物理故障导致应用整体不可用。该场景下，在云上，用户可以使用同一个地域的多个可用区来部署，通过多个可用区的互通能力来完成应用间通讯，同时多可用区的物理隔离性极大提升了应用的容灾能力。针对多可用区部署的服务，云服务厂商不仅会在云资源的供给上提供物理隔离的多个可用区支持，同时也会开放 OpenAPI 能力来供用户查询并控制可用区的不同类型云资源，用户可以基于 OpenAPI 服务能力来构建自己的多可用区部署能力。

特大型的重要商业系统，对系统的容灾能力提出了更高的要求，同城多机房解决的是机房维度的单点问题，无法解决某个城市因为天灾人祸导致的城市级故障。该场景下，在云上可以使用多地域部署，另外，多地域间物理距离适当远一些，避免单地域故障导致服务整体不可用，以此来提升应用的极致高可用。对比于传统的 IDC 异地容灾方案，云天然的多地域支持会极大简化用户跨地域运营服务的成本。

对于高阶用户，云服务厂商会提供 GSLB 全球负载均衡以及对应的 CDN 服务来辅助支撑基础设施的高可用，也会提供自动化的伸缩能力，比如 AWS 的 Auto Scaling 产品、阿里云的弹性伸缩 ESS 产品，用户可用通过配置 Auto Scaling 策略来实现服务多可用区、多地域的自动化部署，保障服务基础设施始终处于高可靠性状态。

数据备份和容灾恢复能力

云服务厂商在数据的高可靠性上具备天然的优势，不仅体现在存储的多副本和数据可靠性极高的 SLA 保障上，同时还以服务化的方式向用户暴露了 OpenAPI，用户可利用云厂商提供的快照、镜像等能力，实现数据备份容灾的高可靠性能力建设。

快照能力是云厂商提供的数据备份方面的核心产品能力，用户可使用快照进行系统盘、数据盘的备份，同时支持增量备份模式，帮助客户节约存储成本。快照支持手动备份与自动备份，推荐使用自动备份方式实现自动快照生成、轮转。对于特定业务场景，可以通过手动方式进行指定快照生成与保留时间，也可以设置为永久保留。当系统出现故障，需要将磁盘（系统盘或者数据盘）数据恢复到历史某一时刻，可以使用快照回滚能力，将指定磁盘回滚，通过快照数据的恢复能力，来提升数据的容灾能力。

同样，用户可用自定义镜像，将快照的操作系统、数据环境信息完整的包含在镜像中，然后使用自定义镜像创建多台具有相同操作系统和数据环境信息的实例。对于多地容灾架构下，用户实现多地域部署时，可以使用镜像的跨地域复制能力来实现镜像备份的分发，从而实现多地域部署情况下的数据备份。

应用可观测能力

为了帮助用户更快、更直观、更简单发现系统内部问题，云服务厂商提供了完善的工具与服务化能力，用户基于此来构建不同层次的可观测能力，同时利用云厂商提供的自助服务来快速发现云资源、甚至自身业务服务的问题。

为了支持不同层次的用户需求，云厂商通常会提供以下几大类监控服务能力：**云资源监控**、**应用层 APM**、**用户业务层监控**。

- **云资源监控**，应用依赖的底层资源监控，比如资源的 CPU、内存、网络等指标的使用率等。通过基础监控，用户可以自助发现云资源发生的异常，这是可观测性最基础的能力，比如阿里云的云监控（Cloud Monitoring Service）、AWS 的 CloudWatch 都提供了云资源监控能力。云厂商同时会提供云资源的诊断能力，用户可一键发起对云资源的诊断，来自助发现云资源可能存在的问题。更进一步，云厂商会提供运维事件能力，比如 AWS 和阿里云均提供了运维事件能力。用户可以基于云的服务能力，通过 OpenAPI 或者事件订阅等方式感知到云资源的异常事件，并通过自定义操作来实现云资源的自动化运维，提升云资源使用的极致可靠性。关于运维自动化编排能力，AWS 的 system manager 与阿里云的运维编排 OOS 产品提供了上述场景的解决方案。
- **应用层 APM**，基于云资源部署的具体应用场景，包括应用指标性能（Metric）、系统调用链（Tracing）、日志监控（Logging）三个维度，比如应用的 JVM 指标、线程池监控、RPC 服务的成功率、时延、错误率监控、以及应用全链路追踪能力。

云资源监控只能发现云资源的问题，对于部署在云上的大规模服务来说，应用层问题的监控和定位能力是更加复杂和困难的。对此，云厂商会提供应用维度的监控和定位服务，主要提供应用维度的标准监控能力，比如应用运行时、线程池、数据库、中间件、接口调用等；从生态角度，也会提供诸如 Prometheus、Kubernetes 等开源产品的支持；另外，从链路视角，会提供前端监控、APP 监控等服务能力。

除了监控能力，云厂商同时会提供应用链路追踪能力，通过 Trace 能力帮助用户发现链路、主机、数据库等多个维度的问题，来实现用户应用问题的自助诊断。比如使用阿里云 ARMS（Application Real-Time Monitoring Service,），可以通过控制台或者 OpenAPI 轻松构建应用监控、APP 监控、以及运行 Prometheus 实例，同时实现链路追踪能力。

- **用户业务层监控**，应用于服务所提供的具体应用场景，比如电商类应用通常监控订单走势、订单成功率、支付成功率等。业务层监控通常是通过业务埋点方式实现，传统运维典型的解决方案是 ELK。

云服务厂商会通过产品以服务化的方式来提供日志服务，用户使用日志服务做日志、数据的采集与集成，并基于此做 Logging 和 Metering。用户通过自定义应用系统的内容、格式，并通过日志服务收集，并在日志服务中配置自定义细粒度监控大盘，观测自身业务运行情况，同时配置预警体系，建设用户层问题发现与定位能力。

阿里云的 SLS 作为云原生观测分析平台，为 Log/Metric/Trace 等数据提供大规模、低成本、实时平台化服务，用户可以通过 SLS 构建业务大盘并设置监控预警，同时可以基于 SLS 做全链路的日志串联来进行问题分析与故障定位。

在可观测性能力的基础上，云服务厂商同时会提供应用高可用服务，比如阿里云的 AHAS（Application High Availability Service），可以通过流量防护、故障演练、多活容灾、开关预案来实现自动化流量控制、业务降级与预案执行，更进一步通过混沌工程来完成故障巡检、故障注入、以及系统稳态度量。

弹性容错能力

除了在基础设施、数据上的容错能力外，云服务厂商通常也会提供应用服务的容错能力，帮助用户构建具备弹性、容错能力的分布式系统。

- 弹性容错能力，分布式系统核心的两个弹性容错能力是流控与降级，通过流控来保护应用过载，通过降级来容忍业务部分有损换取整体可靠性。传统的流控方式是人工判断干预，高阶的方式是通过监控体系自动发现热点流量或异常流量，自动化选择自适应过载保护或者设置自动降级策略并执行。阿里云的 AHAS 提供了分布式应用调用链路的流量防护、应用间调用降级的解决方案。
- 混沌工程与故障演练，混沌工程（Chaos Engineer）是一种提高分布式系统弹性能力的工程实践，通过主动制造故障，测试系统在各种压力下的行为，在生产环境提前识别潜在的故障，避免故障真实发生。故障演练是遵循混沌工程实验原理的实践之一，其建立了一套标准的演练流程，包含准备阶段、执行阶段、检查阶段和恢复阶段。通过四阶段的流程，覆盖用户从计划到还原的完整演练过程，并通过可视化的方式清晰呈现给用户。阿里云的 AHAS 产品提供了故障演练的完整解决方案，覆盖了主流的分布式解决方案，比如强弱依赖演练、消息演练、容器演练、容灾演练等。

可靠性体系阿里云主要产品

从基础设施可靠性、数据可靠性到应用可观测性、APM、自助诊断、弹性容错能力等服务可靠性，阿里云都提供了完备的产品解决方案。用户可以利用这一系列能力，来提升自身服务的可靠性。

全球化超级数据中心

阿里云基础设施目前已面向全球四大洲，开服运营 25 个公共云地域、80 个可用区，此外还拥有 4 个金融云、政务云专属地域，并且致力于持续的新地域规划和建设。通过全球化的布局、超级规模的数据中心、持续的投入与深入布局来保障阿里云基础设施坚实、可靠。

快照与自定义镜像

从块存储技术角度，阿里云的块存储设备在具备高性能和低时延的优势下，同时提供了极高 SLA 保障了数据的可靠性，其中云盘采用分布式三副本机制，为 ECS 实例提供 99.9999999%的数据

可靠性保证。

从数据备份与容灾恢复角度，阿里云提供了快照 2.0 技术，提供了更高的快照额度、更灵活的自动任务策略，并进一步降低了对业务 I/O 的影响，同时增量快照能力可以以更快的快照制作速度和更小的空间占用，帮助用户提升效率并降低成本。用户可以通过自定义快照策略实现快照自动化备份，以极低的成本完成数据备份，在故障场景，用户可以通过控制台或者 OpenAPI 来手动或着自动化完成快照回滚、数据恢复。同样的原理适用于自定义镜像，用户可以通过镜像的制作、复制、恢复来完成数据备份、中转、恢复。

自助问题排查

阿里云的基础云产品服务比如 ECS、RDS、虚拟网络均提供了云资源侧的自助诊断能力，以 ECS 和 RDS 诊断为例简单介绍。

- ECS 自助问题排查：ECS 自助问题排查提供的实例健康诊断、操作异常诊断、安全组规则检测、以及网络连通性诊断，可以全方位帮助用户诊断实例的操作系统配置、磁盘状态、网络配置、网络状态等配置异常，同时给予修复建议方案，帮助用户及时处理潜在风险。
- 数据库自治服务 DAS（Database Autonomy Service）：DAS 是一种基于机器学习和专家经验实现数据库自感知、自修复、自优化、自运维及自安全的云服务，帮助您消除人工操作引发的服务故障，有效保障数据库服务的稳定、安全及高效。

云监控 CMS

云监控服务可用于收集获取阿里云资源的监控指标或用户自定义的监控指标，探测服务可用性以及针对指标设置警报。使您全面了解阿里云上的资源使用情况、业务的运行状况和健康度，并及时收到异常报警做出反应，保证应用程序顺畅运行。

- 基础监控，云上云下统一的主机监控解决方案及百余款云产品监控。

- 网络监控，基于私网和公网的网络可用性监控。
- 业务监控，过日志监控、自定义监控把业务数据归集到云上进行统一监控和管理。

日志服务 SLS

日志服务（SLS）是云原生观测分析平台，为 Log/Metric/Trace 等数据提供大规模、低成本、实时平台化服务。一站式提供数据采集、加工、分析、告警可视化与投递功能，全面提升研发、运维、运营和安全等场景数字化能力。作为云原生观测分析平台。

- 数据采集，支持 Log/Metric/Trace 统一采集，支持服务器/应用/移动设备/网页/IoT 等数据源接入，支持阿里云产品/开源系统/云间/云下日志数据接入。
- 数据加工，通过灵活语法，在不编写代码情况下支持各种复杂数据提取、解析、富化、分发等需求，支持结构化分析。
- 查询分析，提供关键词、SQL92、AIOps 函数等多种方式，支持面向文本+结构化数据实时查询分析，异常巡检与智能分析。
- 监控告警，具备丰富的可视化组件，可创建所见即所得的交互式分析大盘。同时支持实时可编排的告警功能，可随时随地掌握业务动向。
- 日志审计，多账户下实时自动化、中心化采集云产品日志并进行审计，支持升级所需合规存储、查询及信息汇总报表。
- 投递与消费，与各种实时计算及服务实时对接，并可以实现自定义消费。支持数据投递至存储类服务，支持压缩、自定义 Partition 以及行列等各种存储格式。

应用实时监控服务 ARMS

应用实时监控服务（Application Real-Time Monitoring Service，简称 ARMS）是一款应用性能管理产品，包含前端监控、应用监控和 Prometheus 监控三大子产品，涵盖了浏览器、小程序、APP、分布式应用和容器环境等性能管理，能帮助用户实现全栈式的性能监控和端到端的全链路追踪诊断。

- 实时洞察，即刻提升应用性能。前端、应用至底层机器，应用实时监控服务 ARMS 提供了完整的数据大盘监控，展示请求量、响应时间、FullGC 次数、慢 SQL 和异常次数、应用间调用次数与耗时等重要的关键指标，时刻了解应用程序的运行状况，确保向用户提供优质的使用体验。
- 全面掌握 Web 端性能数据，提供优质体验。应用实时监控服务 ARMS 前端监控专注于 Web 端体验数据监控，从页面打开速度、页面稳定性和外部服务调用成功率这三个方面监测 Web 页面的健康度，帮助您降低页面加载时间、减少 JS 错误，有效提升用户体验。
- Prometheus 监控，云原生时代一站式体验。应用实时监控服务 ARMS 提供 Prometheus 全托管式云服务，无需安装运维，一键开启，开箱即用监控大盘。

链路追踪 XTrace

链路追踪（Tracing Analysis）为分布式应用的开发者提供了完整的调用链路还原、调用请求量统计、链路拓扑、应用依赖分析等工具。能够帮助开发者快速分析和诊断分布式应用架构下的性能瓶颈，提高微服务时代下的开发诊断效率。

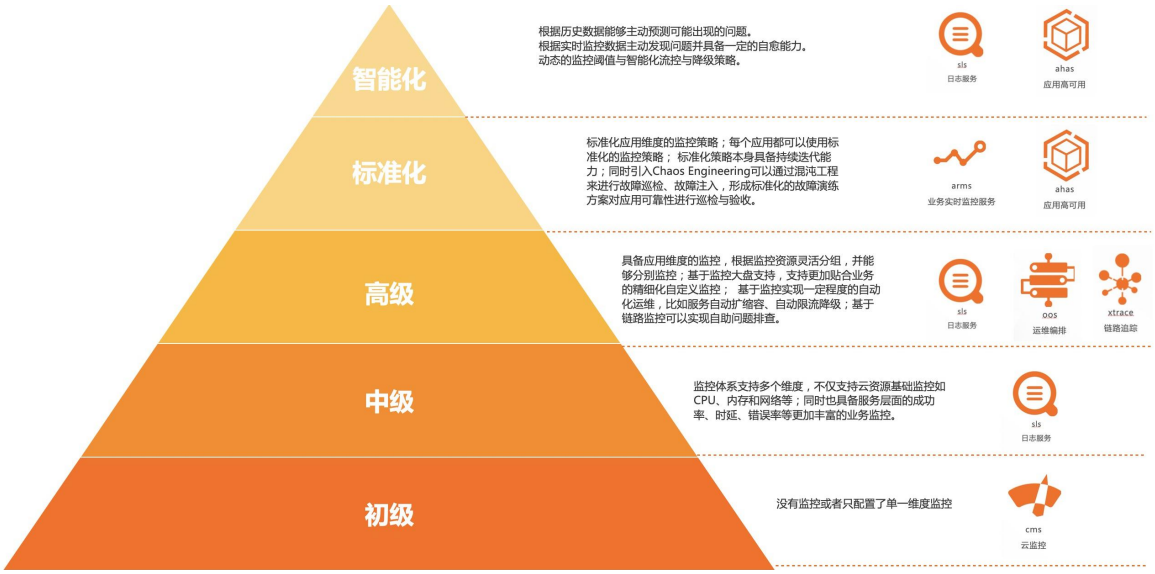
- 分布式调用链查询诊断，同时支持微服务程序 HTTP、Dubbo、HSF 等接口进行追踪与 PaaS 调用，如对数据库、NoSQL、MQ 等调用进行追踪。
- 应用性能实时汇总，可以通过跟踪整个应用程序的用户请求，来实时汇总，组成应用程序的单个服务和资源。
- 分布式拓扑动态发现，可以收集您的所有分布式微服务应用和相关 PaaS 产品的分布式调用信息。

应用高可用服务 AHAS

应用高可用服务（Application High Availability Service）专注于提高应用及业务的高可用能力，主要提供流量防护、故障演练、多活容灾、开关预案四大核心能力。用户通过各模块可以快速低成本地在营销活动场景、业务核心场景全面提升业务稳定性和韧性。

- 流量监控与防护，提供包括 QPS、并发线程、响应时间（RT）、异常、CPU/load、网络流量等指标的秒级监控能力。同时，提供应用级别的流量控制、应用间的降级隔离、单机自适应过载保护、热点流量探测与防控、脉冲流量削峰填谷、慢方法/SQL 的自动熔断、分布式流量控制等。
- 网关防护，支持 Nginx/Ingress 网关层流量控制以及 Spring Cloud Gateway、Zuul 等常用 API gateway 的流量防护，从流量入口处拦截骤增流量，防止下游服务被压垮。
- 开关预案，支持代码中配置项的动态管理，根据需求为某个应用开启或关闭部分功能，或设置某个性能指标的阈值。通常用于设置黑白名单、运行时动态调整日志级别、降级业务功能等场景。
- 混沌工程与故障演练，提供一站式架构分析、故障巡检、故障注入、系统稳态度量等功能，帮助用户增强分布式系统的容错性和可恢复性，帮助系统平稳上云。
- 多活容灾，支持分布在多个站点的系统同时对外提供服务，保障故障场景下的业务快速恢复。横向囊括容灾架构的上线、运维、演练、切流、升级到下线的全生命周期。纵向包含业务流量的完整路径，从流量接入，到服务化调用，异步化消息，再到最终数据落库。

可靠性成熟度模型



基于云上问题发现、定位与恢复的能力建设情况，可靠性成熟度划分为以下 5 个等级。

等级	可靠性
1-初级	• 没有监控或者只配置了单一维度的基础监控
2-中级	• 监控体系支持多个维度，不仅支持云资源基础监控如 CPU、内存和网络等 • 同时也具备服务层面的成功率、时延、错误率等更加丰富的业务监控
3-高级	• 具备应用维度的监控，根据监控资源灵活分组，并能够分别监控 • 基于监控大盘支持，支持更加贴合业务的精细化自定义监控 • 基于监控实现一定程度的自动化运维，比如服务自动扩缩容、自动限流降级 • 基于链路监控可以实现自助问题排查
4-标准化	• 标准化应用维度的监控策略，每个应用都可以使用标准化的监控策略 • 标准化策略本身具备持续迭代能力 • 同时引入 Chaos Engineering 可以通过混沌工程来进行故障巡检、故障注入，形成标准化的故障演练方案对应用可靠性进行巡检与验收
5-智能化	• 根据历史数据能够主动预测可能出现的问题 • 根据实时监控数据主动发现问题并具备一定的自愈能力 • 动态的监控阈值与智能化流控与降级策略

安全和合规能力

云计算安全或云安全指通过一系列策略、控制和技术，共同确保数据、基础设施和应用安全，保护云计算环境免受外部和内部网络安全威胁和漏洞的影响。

安全责任共担模型

不同于传统的 IDC，云计算是一种共享技术模型，其安全责任由双方共同承担，这通常被称为**安全责任共担模型**。

宏观上讲，云计算平台负责基础设施（包括跨地域、多可用区部署的数据中心，以及骨干传输网络）和物理设备（包括计算、存储和网络设备）的安全，并负责运行在云操作系统之上的虚拟化层和云产品层的安全。同时，云平台也负责平台侧的身份和访问的控制和管理、监控和运营，从而为客户提供高可用和高安全的云服务。

客户负责以安全的方式配置和使用各种云上产品，并基于这些云产品的安全能力，以安全可控的方式构建自己的云上应用和业务，保障云上数据的安全。

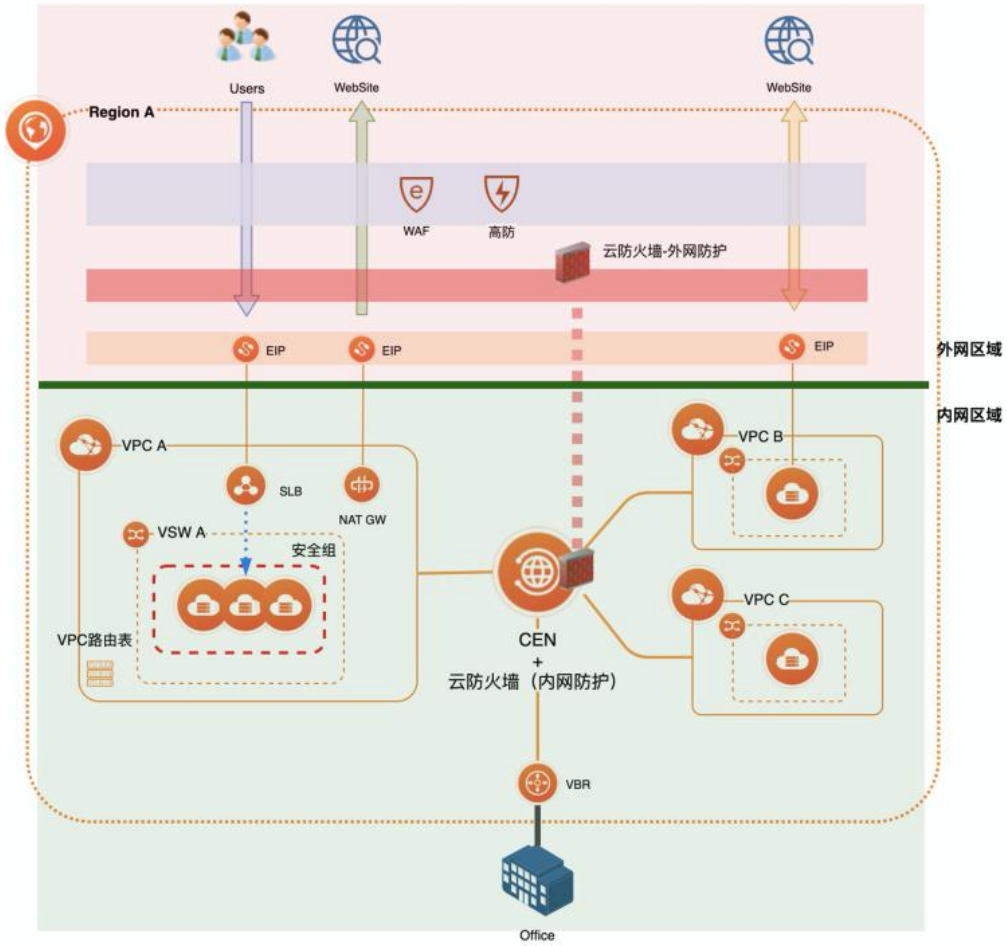
网络安全

出于操作和安全的原因，在云上将网络进行隔离是非常重要的。

云计算利用虚拟网络（Virtual Private Cloud，简称 VPC），来抽象物理网络并创建网络资源池，实现数据链路层的隔离，为每个用户提供一张独立隔离的安全网络环境。不同 VPC 之间内部网络完全隔离，只能通过对外映射的 IP 互连。在 VPC 内部，用户可以自定义 IP 地址范围、网段、路由表和网关等；此外，用户可以通过 VPN 网关、高速通道物理专线、智能接入网关等服务将本地数据中心和云上 VPC 打通，也可以通过云企业网实现全球网络互通，从而形成一个按需定制的

网络环境，实现应用的平滑迁移上云和对数据中心的扩展。

此外，网络是所有云服务的唯一入口，网络攻击是种类最多、危害最大，也是最难防护的风险之一。云计算平台会提供一套成熟的网络安全架构，以应对来自互联网的各种威胁。在阿里云上，可以通过安全组、网络 ACL、路由策略或网络专线来控制虚拟网络的访问权限。除了对内网网络访问的控制之外，还需要配置云防火墙、应用程序防火墙、DDoS 防护等安全措施，针对各种外部网络安全威胁，进行安全防护。



系统审计

操作审计和追踪是安全生命周期的重要组成部分，可以识别潜在安全配置错误、威胁或意外行为，也用于支持质量流程、法律或合规义务，还可以用于威胁识别和响应工作。

云计算平台会提供相应的面向资源和操作的配置历史追踪、配置合规审计等能力，让用户完全专注于审计任务本身，摆脱对某一具体硬件或程序的依赖，通过数据的云存储，实现各类审计信息的数字化，使各种审计资源，包括审计人员、程序和相关的硬件设备，通过云来协同工作，使审计资源得到充分优化利用，以促进信息的交流和共享，轻松实现系统的自主监管和确保持续性合规。

在阿里云上，操作审计（Action Trail）、配置审计（Config）和日志审计服务提供了审计和更改跟踪功能来审计资源和更改配置。

管控通道安全能力

传统的运维通道需要借助 SSH 取得密钥进行管理，并开放相应的网络端口，密钥管理不当以及网络端口暴露都会对云上资源带来很大的安全隐患。原生的阿里云云上自动化运维通道——云助手，可以帮助客户安全、高效的运维云上资源。通过云助手，可以在云服务器 ECS 上实现批量运维、执行命令和发送文件等操作；通过云助手 Session Manager，可以交互式运维 ECS 实例。以上运维操作都无需密码，无需登录，ECS 实例不需要使用公网，也不需要通过跳板机，通过云助手以下安全机制保证运维通道的安全性：

- **权限控制**，云助手通过 RAM 策略，从实例、资源组、标签、源 IP 地址等多个维度控制用户对实例的访问权限。只有具有权限的用户才能通过云助手通道运维 ECS 实例。
- **链路可靠**，全链路采用 Https 协议进行交互，传输过程中对数据进行加密。ECS 实例入方向采用内部安全管控链路，无需用户开放端口，降低被入侵的风险；出方向通过内网进行通信，无需暴露公网即可使用。
- **内容安全**，通过云助手通道传输的命令内容，通过加密及签名校验的方式，确保传输过程中无法被篡改，保证命令内容的安全性。
- **日志审计**，通过云助手通道传输的命令、文件都可以通过 API 进行审计，用户可以查询执行的时间、身份、执行内容、执行结果等信息。同时支持将日志投递到存储（OSS）或日志（SLS）等系统中，提供日志归档、分析能力。

身份和访问管理

身份和访问管理是当今的 IT 面临的最大挑战之一，也深受云计算的影响。云的 IAM（Identity and Access Management，身份认证与访问管理）解决方案实现零信任安全，确保没有人是可信的，并且在每个资源的访问点评估访问请求。这将允许每个应用程序、每个策略和每个访问场景的分布式访问决策，基于云的 IAM 解决方案允许组织使用单点登录、多因素身份验证和访问控制来直接提供对云服务的安全访问。

账户是资源使用的硬边界，我们建议根据功能、业务、合规性要求等来进行账号的分配和隔离。在多用户需要协同操作资源的场景中，建议避免直接共享使用账号，共享账号的密钥等机密信息会大大增加泄露风险，一旦泄露会威胁账号下所有资源的安全。建议使用访问控制创建用户和用户组，并授予各用户和用户组最小权限，可以有效降低风险。

阿里云提供身份和访问管理的以下安全功能，在账号级别防范风险。

- **身份认证**，用户可以使用其云账号(即主账号)或其云账号下 RAM 用户的密码登录阿里云控制台并对其云上资源进行操作，或者使用阿里云的 Access Key(AK)通过 API 访问阿里云资源时对用户身份进行认证，也可以通过阿里云 Security Token Service(STS) 为 RAM 用户、阿里云服务、身份提供商等受信实体提供短期访问资源的权限凭证的云服务。颁发令牌时，管理员可以根据需要来定义令牌的权限和自动过期时间(默认为 1 小时过期)。此外，阿里云还支持 MFA 认证、SSO 认证、SSH 密钥对认证方式。
- **访问授权（RAM）**，阿里云为客户提供了多种工具和功能，用来帮助客户在各种情况下授权资源的使用权力。其中，阿里云为客户提供 Resource Access Management(RAM)资源访问控制服务，用于用户身份管理与资源访问控制。RAM 使得一个阿里云账号(主账号)可拥有多个独立的子用户(RAM 用户)，从而避免与其他用户共享云账号密钥，并可以根据最小权限原则为不同用户分配最小的工作权限，从而降低用户的信息安全管理风险。RAM 授权策略可以细化到对某个 API-Action 和 Resource-ID 的细粒度授权，还可以支持多种限制条件(例如，源 IP 地址、安全访问通道 SSL/TLS、访问时间、多因素认证等)。

应用安全

随着应用程序开发实践的不断进步和采用新的流程、模式和技术，应用安全也在以难以置信的速度发展。云计算通过对平台类产品的应用系统制定并实施详细的安全控制措施，为应用程序带来安全优势，来帮助客户减少信息安全的成本和投入。

在应用安全层面，阿里云为客户提供了应用环境安全、应用配置安全和应用自身保护的三个维度的安全功能。

- **应用环境安全**，阿里云漏洞扫描服务可帮助用户自动发现其网站的关联资产，并进行高效精准的自动化漏洞渗透测试和敏感内容监测等，保障上线前线上应用环境的安全性。阿里云也提供为用户的应用环境进行安全加固的服务，在获得客户授权委托的情况下，远程登录到客户的业务系统服务器上，根据用户的资产情况和安全需求，对其外网或内网主机进行全方位的基线加固和组件升级，提前修补系统潜在的各种高危漏洞和安全威胁。
- **应用配置安全**，应用配置管理(Application Configuration Management，简称 ACM)，是一款在分布式架构环境中对应用配置进行集中管理和推送的产品。利用 ACM，用户可以在微服务、DevOps、大数据等场景下极大减轻配置管理的工作量，并增强配置管理的服务能力。为了确保敏感配置(数据源、Token、用户名、密码等)的安全性，降低用户配置的泄露风险，
- ACM 提供了创建加密配置的功能。
- **应用保护**，阿里云提供 Web 应用防火墙(Web Application Firewall，简称 WAF)服务，基于云安全大数据和智能计算能力，通过防御 SQL 注入、XSS 跨站脚本、常见 Web 服务器插件漏洞、木马上传、非授权核心资源访问等 OWASP 常见 web 攻击，过滤海量恶意访问，避免网站资产数据泄露，保障网站应用的安全性与可用性。

安全和合规方面阿里云主要产品

安全组

安全组是阿里云提供的实例级别虚拟化防火墙，具备状态检测和数据包过滤功能，可用于在云端划分各个 ECS 实例(在容器服务中，即各个容器集群)间的安全域。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。使用安全组可设置单台或多台云服务器的网络访问控制，它是重要的网络安全隔离手段，用于在云端划分网络安全域。配置严格的安全组访问权限，是最简单直接的防范网络攻击、屏蔽恶意流量的方式。

网络 ACL

网络 ACL 是专有网络 VPC 中的网络访问控制功能。您可以自定义设置网络 ACL 规则，并将网络 ACL 与交换机绑定，实现对交换机中云服务器 ECS 实例流量的访问控制。网络 ACL 是在 VSW 粒度对进出 VSW 的流量做检测和数据包过滤。

云防火墙

云防火墙在安全组、网络 ACL 功能的基础上提供了补充，为构建网络安全环境提供了更好的“深层防御”。安全组、网络 ACL 提供分布式网络层流量过滤，以限制每个订阅中虚拟网络内资源的访问流量。如果用户需要跨虚拟网络，启用某些应用程序级别的保护时，则需要使用云防火墙服务。

阿里云云防火墙是业界首款公共云环境下的 SaaS 化防火墙，可以统一管理互联网到业务的南北向访问策略，以及业务与业务之间的东西向微隔离策略。通过云防火墙，用户可以对南北向和东西向访问的网络流量进行分析，并支持全网流量(互联网访问流量、安全组间流量等)可视化，并支持对主动外联行为的分析和阻断。云防火墙还集成了入侵检测(IPS)功能和威胁情报能力，并支持入侵检测分析。同时，云防火墙支持网络流量及安全事件日志存储功能，默认保存 6 个月的安全事件日志、网络流量日志及防火墙操作日志，满足网安法和等保 2.0 的相关要求。

Web 应用防火墙 WAF

Web 应用防火墙（WAF），云防火墙为所有端口和协议提供网络别的保护，Web 应用程序防火墙(WAF)是应用程序网关的一项功能，为您的网站或 App 业务提供一站式安全防护。WAF 可以有效识别 Web 业务流量的恶意特征，在对流量进行清洗和过滤后，将正常、安全的流量返回给服务器，避免网站服务器被恶意入侵导致服务器性能异常等问题，保障网站的业务安全和数据安全。

DDoS 防护

阿里云使用自主研发的 DDoS 防护系统保护所有数据中心，支持防护全类型 DDoS 攻击，并通过 AI 智能防护引擎对攻击行为进行精准识别和自动加载防护规则，保证网络的稳定性。同时，阿里云的 DDoS 防护系统支持通过安全报表实时监控风险和防护情况。

操作审计 ActionTrail

操作审计（ActionTrail）为用户提供统一的云资源操作安全日志管理，记录云账号下的用户登录及资源访问操作，包括操作人、操作时间、源 IP 地址、资源对象、操作名称及操作状态。利用 ActionTrail 保存的操作记录，用户可以实现安全分析、入侵检测、资源变更追踪以及合规性审计。为了满足用户的合规性审计需要，用户往往需要获取主账户和其子用户的详细操作记录。ActionTra-il 所记录的操作事件可以满足此类合规性审计需求。

配置审计 Config

配置审计 Config 是面向云上资源的审计服务，为用户提供跨区域的资源清单和 检索能力，记录资源的历史配置快照，形成配置时间线。当资源发生配置变更时，自动触发合规评估，并针对“不合规”配置发出告警。使用户能够实现对于海量云上资源合规性的自主监控，应对企业内部和外部合规的需要。

日志审计服务

在继承现有日志服务所有功能外，阿里云还支持多账户下实时自动化、中心化采集云产品日志并进行审计，以及支持审计所需的存储、查询及信息汇总。日志审计服务覆盖基础（ActionTrail、容器服务 Kubernetes 版）、存储（OSS、NAS）、网络（SLB、API 网关）、数据库（关系型数据库 RDS、云原生分布式数据库 PolarDB-X 1.0、云原生数据库 PolarDB）、安全（WAF、DDoS 防护、云防火墙、云安全中心）等产品，并支持自由对接其他生态产品或自有 SOC 中心。

云助手

云助手是专为云服务器 ECS 打造的原生自动化运维工具，通过免密码、免登录、无需使用跳板机的形式，在 ECS 实例上实现批量运维、执行命令（Shell、Powershell 和 Bat）和发送文件等操作。典型的使用场景包括：安装卸载软件、启动或停止服务、分发配置文件和执行一般的命令（或脚本）等。

访问控制 RAM

访问控制 RAM 使您能够安全地集中管理对阿里云服务和资源的访问。您可以使用 RAM 创建并管理子用户和用户组，并通过权限管控他们对云资源的访问。

云安全中心

云安全中心是一个实时识别、分析、预警安全威胁的统一安全管理系统，通过防勒索、防病毒、防篡改、合规检查等安全能力，实现威胁检测、告警响应、攻击溯源的自动化安全运营闭环，保护您的云上资产和本地服务器安全，并满足监管合规要求。

应用配置管理 ACM

应用配置管理(Application Configuration Management，简称 ACM) 是一款在分布式架构环境中对应用配置进行集中管理和推送的产品。

云安全成熟度模型

等级	描述
初级	• 使用基础的云网络能力构建和维护应用，人工管理安全流程
基础	• 漏洞与合规驱动的安全和审计能力 • 少量使用了云原生自动化运维通道
体系化	• 建立安全流程和账号使用规范，关键系统具备完整审计能力 • 使用云原生自动化运维通道执行日常的运维
标准化	• 标准化安全流程，核心数据可控 • 安全和审计事件可追溯 • 完全通过云原生自动化运维通道、Serverless 服务运维
智能化	• 中心化、自动化、智能化的应用安全体系和审计体系 • 数据驱动，自主优化安全和合规防护的能力

成本和资源量化管理

云服务相比 IDC 的最大特点之一是使用资源而非持有资产。在云上不仅可以快速的创建和释放资源，相比 IDC 也可大大的降低使用成本。云上的成本主要包括资源成本和运维成本，前面已经有章节介绍了自动化运维，下面主要介绍如何控制资源成本。

以云服务器为例，它的资源成本主要由计算、存储、网络三大部分构成。在云上，针对每部分，一般会提供丰富的规格和计费方式。

比如，计算资源有通用型、计算型；存储有云盘、高效云盘。选择合适的规格能有效降低资源成本，如电商网站运营选择计算型实例，相比通用性/内存优化型，同样规格的实例（如：8C16G）成本将增加 40%以上。

计费方式直接决定资源的定价，选择合适的计费方式可以直接节省成本。如相比使用按量计费，选择抢占式实例最高可节省 90%的成本。当然所有规格和计费方式的选择，需根据具体应用场景和业务类型来决策。

云计算中常用的计费模式

云上提供了多种付费模式，分别满足长周期、低成本以及周期高弹性的需求。如包年包月、按量付费、节省计划、预留实例券和抢占式实例等。

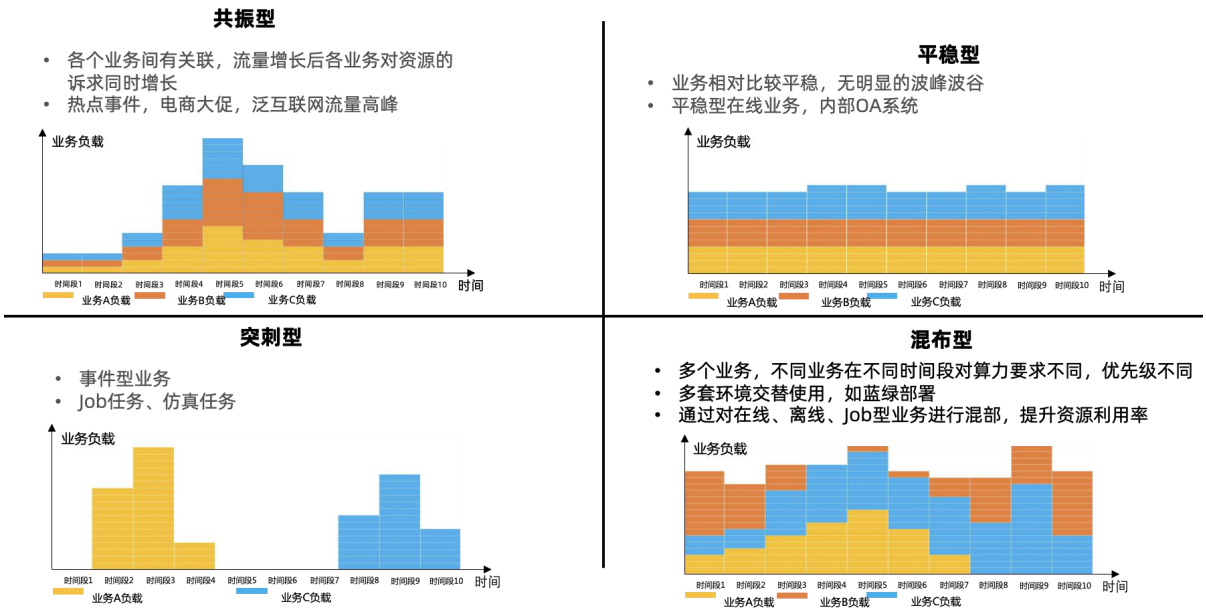
长期稳定的业务，选择使用包年包月的付费模式，可以在预留资源的同时享受更大的价格优惠，但相对丧失了灵活性。对于高弹性业务选择按量付费或抢占实例，使用灵活且按需按秒计费，而预留实例和节省计划则兼顾了资源的灵活性和低成本。

对比项	包年包月	按量付费	抢占式实例	预留实例券	节省计划
使用方式	所有操作与购买的某台实例绑定	所有操作与购买的某台实例绑定	所有操作与购买的某台实例绑定	资源和账单解耦，搭配按量付费实例灵活使用	资源和账单解耦，搭配按量付费实例灵活使用
付费方式	预付费，一次性付清	后付费，按秒计费，按小时出账	后付费，按秒计费，按小时出账	可选全预付、部分预付或 0 预付	可选全预付、部分预付或 0 预付
价格特点	相比按量付费有较大优惠	相比其它方式价格最高	价格随市场供需变化而浮动，可低至按量付费的 1 折	相比按量付费有一定优惠，价格与包年包月相近	相比按量付费有一定优惠，灵活性极佳，但价格比预留实例券略高
应用场景	适用于固定的 7*24 服务，例如 Web 服务、数据库等	应对爆发业务，例如临时扩展、测试、科学计算等	应对爆发业务，例如临时扩展、测试、科学计算等	抵扣按量付费实例账单，可用于固定的 Web 服务、数据库等	抵扣按量付费实例账单，可用于固定的 Web 服务、数据库等

根据业务类型选择合适的计费方式降低成本

根据实际的业务需求和应用场景，为不同类型应用的资源选择合适计费方式，能更好地优化资源使用成本。

阿里云提供的可参考计费方式组合如下：



业务类型	共振型	平稳型	突刺型	混部型
业务特征	各个业务间有关联，流量增长后各业务对资源的诉求同时增长	业务相对比较平稳，无明显的波峰波谷	各个业务之间关联不大，具有突发性	多个业务，不同业务在不同时间段对算力的要求不同，优先级不同
场景举例	热点事件、电商大促、泛互联网流量高峰	平稳型在线业务，如内部 OA 系统	事件型业务、Job 任务、仿真任务	多套环境交替使用（如蓝绿部署），在线、离线、Job 任务型混合部署
推荐计费方式	按量付费+节省计划（或预留实例券）	包年包月 按量付费+节省计划（或预留实例券）	按量付费 高频突刺部分可适当搭配节省计划（或预留实例券）	按量付费+节省计划（或预留实例券）

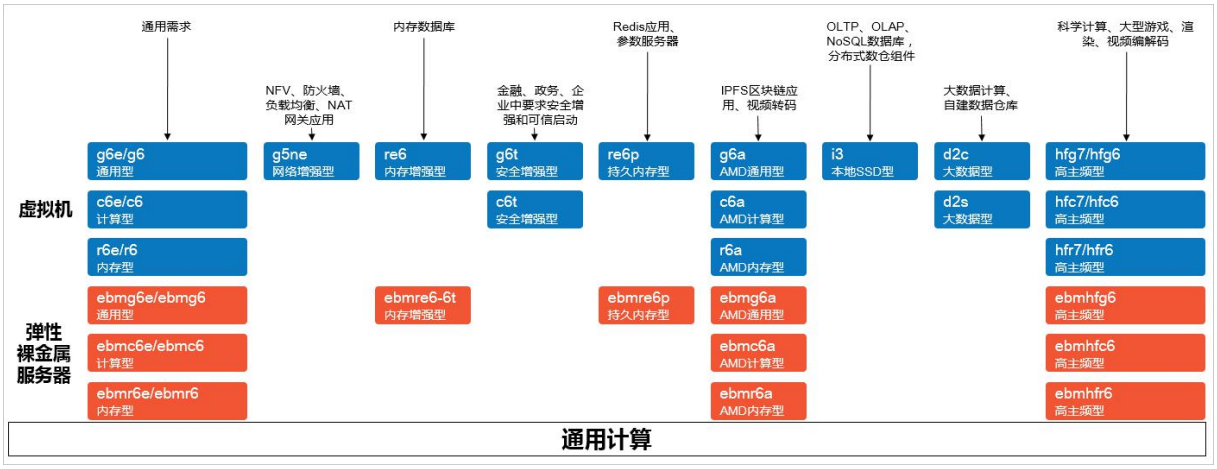
根据应用场景选择合适的规格降低成本

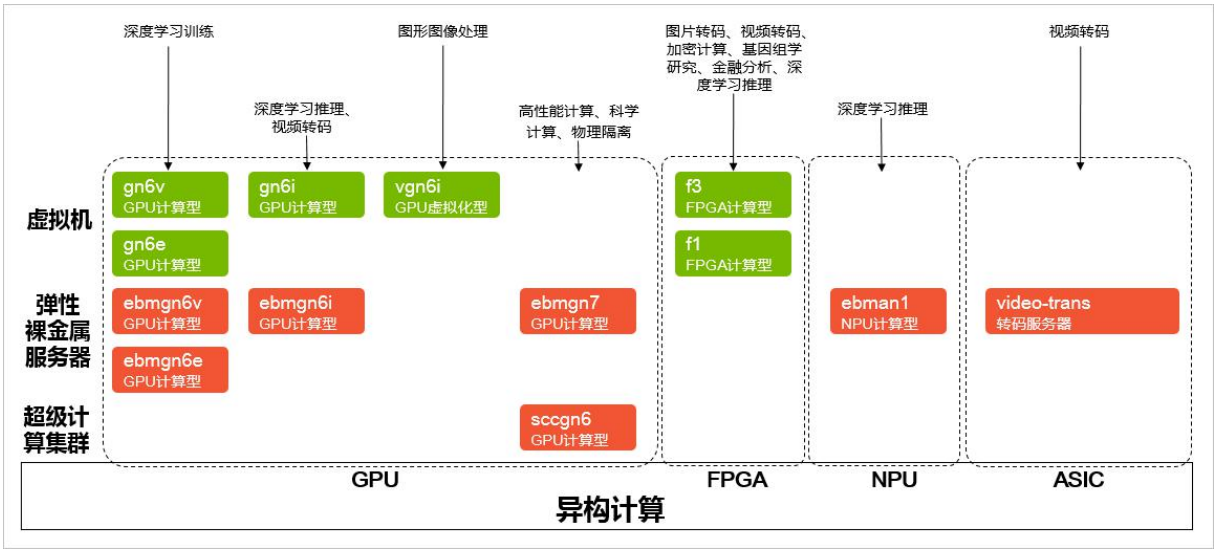
云服务器根据不同的使用场景推出了不同的规格，比如常见的规格有：通用型、计算型、内存型、大数据型、GPU 型、裸金属服务器、突发性能实例等，用户可根据自身实际的业务场景来进行选择。

如阿里云 ECS 云服务器实例规格可以分为以下多种：

- x86 计算：通用型、计算型、内存型、大数据型、本地 SSD、高主频、增强型及共享型等；
- 异构计算：GPU 计算型、GPU 虚拟化型、ASIC 型、FPGA 计算型、NPU 计算型；
- 弹性裸金属服务器（神龙）：CPU 型、GPU 型、AMD 等；
- 超级计算集群：CPU 型、GPU 型；
- ARM 计算：通用型、计算型、内存型。

可参考实际应用场景选择对应的实例规格：





通过提升资源利用率来降低成本

除选择合适规格和计费方式外，提升资源利用率也是控制成本的有效方法：在业务需要时创建资源，在业务空闲时停止或释放闲置资源；在业务增长时升级资源规格，在业务需求下降时降低规格；时升级换代旧机型使得同等价格能得到更优的性能。

阿里云提供如下的能力帮助您更改资源配置或计费方式：

- 节省停机模式：应用于按量付费的实例。在不需要使用资源时，可以以节省停机模式停止该实例，此时实例计算部分会停止收费，起到降低相关费用、节约成本的作用。
- 灵活变配能力：可以通过修改实例规格（vCPU 和内存）、公网带宽、磁盘大小以及各配件计费方式，来实时调整资源降低成本。
  - 实例规格升降配
  - 修改实例付费类型
  - 修改公网带宽
  - 修改公网带宽计费方式
  - 修改数据盘大小和类型
  - 修改数据盘计费方式

成本和资源量化意识成熟度模型

级别	成本和资源量化意识
1-初级	• 整体资源量较小，付费模式单一 • 没有太多弹性需求
2-中级	• 有相对数量的资源，关注成本上限 • 付费模式较多样化 • 部分弹性需求、资源按需创建或扩容
3-高级	• 有相对数量的资源，开始精细化管理 • 定制化的成本分析能力 • 精确化控制每个应用和服务的成本
4-标准化	• 多维度的成本分析模型 • 标准化应用的成本治理策略 • 模块化成本控制能力，增强研发敏捷性
5-智能化	• 能够预测下一个月份/季度的费用 • 根据历史数据或用户需求给出合适的成本计划

CloudOps 成熟度模型全景图

等级	自动化 Automation	可靠性 Reliabilty	安全和合规 Security	弹性容量管理 Elasticity	成本管理 Cost
L1-初级	• 在网站上手工完成大部分操作 • 使用公共的组件来完成日常运维任务	• 没有监控或者配置了单一维度的基础监控	• 使用基础的云网络能力构建和维护应用，人工管理安全流程	• 容量满足需求，或容量充足，手动调整资源容量	• 整体资源量较小，付费模式单一 • 没有太多弹性需求
L2-中级	• 定义模板或者 Pipeline 任务来取代重复构建 • 开始使用 CLI 工具或者 OpenAPI 来做任务管理	• 监控体系支持多个维度，包括资源和服务维度	• 漏洞与合规驱动的安全和审计能力 • 少量使用了云原生自动化运维通道	• 部分资源根据设置的策略实现了自动变配和扩容缩容	• 有相对数量的资源，关注成本上限 • 付费模式较多样化 • 部分弹性需求、资源按需创建或扩容

等级	自动化 Automation	可靠性 Reliabilty	安全和合规 Security	弹性容量管理 Elasticity	成本管理 Cost
L3-高级	• 完全自动化地构建一套新环境 • 自动化已经完成体系化建设 • 使用通过 OpenAPI 进行日常运维工作，同时自行开发运维系统	• 具备应用维度的监控 • 具备精细化自定义监控和配置能力 • 基于监控实现一定程度的自动化运维 • 基于链路监控实现自助问题排查	• 建立安全流程和账号使用规范，关键系统具备完整审计能力 • 使用云原生自动化运维通道执行日常的运维	• 自动按需调整个系统容量 • 调整的颗粒度能够被精细化控制	• 有相对数量的资源，开始精细化管理 • 定制化的成本分析能力 • 精确化控制每个应用和服务的成本
L4-标准化	• 使用模板实现运维的标准化、自动化 • 完全自动化地构建新环境和系统 • 对网络和安全策略变更也 Code 化 • 能够使用自动化工具自助完成排障	• 监控具备标准化管理能力 • 引入混沌工程保障可靠性	• 标准化安全流程，核心数据可控 • 安全和审计事件可追溯 • 完全通过云原生自动化运维通道、Serverless 服务运维	• 用标准化策略自动管理容量及对应规则 • 每个应用和服务都能按需随时开启标准化策略	• 多维度的成本分析模型 • 标准化应用的成本治理策略 • 模块化成本控制能力，增强研发敏捷性

等级	自动化 Automation	可靠性 Reliabilty	安全和合规 Security	弹性容量管理 Elasticity	成本管理 Cost
L5-智能化	• 从代码仓库出发从零构建新环境和系统（即 GitOps） • 所有的配置都存在代码管理软件中 • 环境配置和系统改变均由代码触发 • 新环境在流水线上完全自动同步	• 根据历史数据主动预测故障 • 具备一定的故障自愈能力 • 智能化流控与降级策略	• 中心化、自动化、智能化的应用安全体系和审计体系 • 数据驱动，自主优化安全和合规防护的能力	• 能够根据历史数据和实时数据叠加，进行更精确，更准确的自动扩缩容	• 能够预测下一个月份/季度的费用 • 根据历史数据或用户需求给出合适的成本计划

参考文档

- The State of DevOps in 2021: A Report Roundup
- Top 5 DevOps trends for 2022
- 2021 STATE OF DEVOPS REPORT
- Gartner 2021 Top10 Technology Trends Impacting DevOps
- Puppet State of DevOps Report 2021

扫码进群交流