阿里云开发者社区
ALIBABA CLOUD DEVELOPER COMMUNITY

阿里云弹性计算

玩转ECS
从入门到精通

进阶篇

ECS，不只云服务器

云服务器选型指南一本通
云上 ECS九大实践一站搞定
自动化运维与云上架构必读

藏经阁

阿里云开发者电子书系列

扫码查看“玩转 ECS”详情页
解锁更多课程视频

关注百晓生，笑谈云计算

阿里云开发者“藏经阁”
海量免费电子书下载

目录

第一章 入门与选型 4

1.1 如何选择 ECS 云服务器实例规格 5

第二章 自动化运维 25

2.1 服务器自动化迁移中心 SMC 最佳实践及新特性介绍 26

2.2 云上自动化部署和运维的正确姿势 40

2.3 ECS 云助手，实现云上运维自动化 51

2.4 ECS 自助服务之智能诊断和自动化修复 74

2.5 ECS 数据保护-数据备份新特性与最佳实践 88

第三章 架构优化 95

3.1 云上高弹性、低成本解决方案 96

3.2 基于弹性计算网络能力提升容器密度最佳实践 111

3.3 ECS 安全组最佳实践及新特性介绍 117

3.4 Region 化部署和跨可用区容灾介绍 127

第一章 入门与选型

1.1 如何选择 ECS 云服务器实例规格

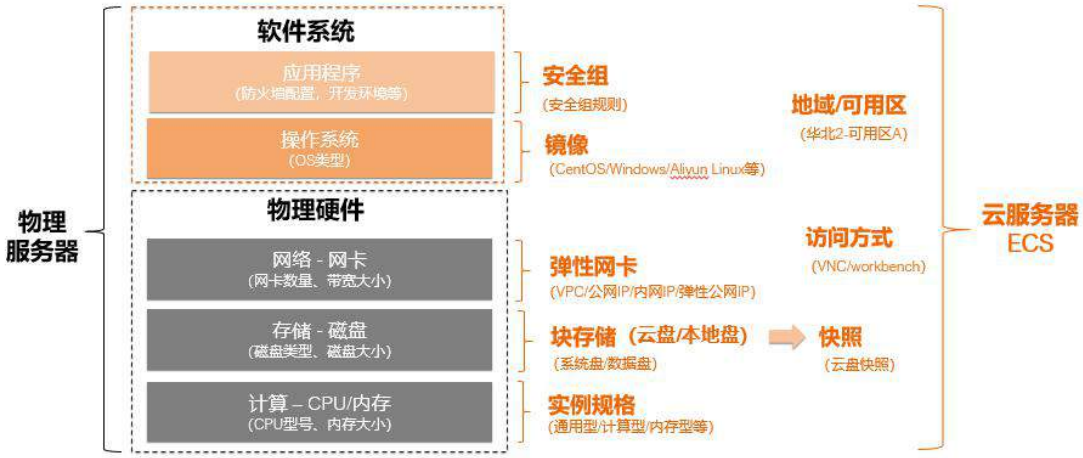
摘要：本文主要大概分为 4 个部分，第一部分介绍云服务器 ECS 的基本概念；第二部分对 ECS 的实例规格族进行一个详细的解读；第三部分给大家实战讲解如何去做 ECS 实例的选型；最后一部分简单介绍一下如何去省钱省力的来使用 ECS。

演讲嘉宾简介：马小婷，阿里云智能产品专家，负责 ECS 服务能力的产品规划与设计工作，包括实例场景化选型工具，实例诊断与修复，ECS 事件系统，ECS 优化推荐，弹性伸缩与弹性供应等，致力于完善 ECS 全生命周期的场景支撑，为用户提供完整高效的服务能力与自助智能的服务体验。

第一部分：云服务器 ECS 基础概念

云服务器的基础概念

第一部分会给大家介绍云服务器的一些基本概念。



在开始前，大家可以回想一下，我们自己购买笔记本电脑的时候中会考虑哪些因素？我自己会先选择品牌，一般情况下在确定了品牌之后，接下来就会考虑硬件配置，主要是物理硬件的配置和软件的配置。

硬件配置上，我首先会考虑**计算性能**，像 CPU 和内存的大小、CPU 的型号等；第二就是**存储**，笔记本电脑的磁盘有多大；第三部分就是**网络能力**，比如网卡有几个，对于玩游戏的同学来说，显卡配置也很重要。除了硬件配置外，我也会考虑电脑的**操作系统**是什么样的，比如 Mac OS， windows 或 ubuntu 等。而拿到电脑之后，我们首先会做一些基础应用软件的安装和配置，包括防火墙等保证我们整个应用环境的安全性。

这是我在现实生活中去购买一台物理电脑的流程，其实这些概念在云上也是适用的，比如说我们在选择一些物理硬件的参数的时候，选 CPU 和内存，对应在云上的话，就是选择 ECS 实例的 CPU 和内存大小以及 CPU 的型号。

存储这一块，磁盘在云上对应的概念就是块存储，在云上块存储其实是包含两个概念，一个概念是云盘，一个概念是本地盘。有一个跟我们现实生活中不太一样的点，是云上的块存储，我们在购买的过程中需要指定用作系统盘还是用作数据盘的。而现实生活中买了一个电脑里面是有一块磁盘，然后我们自己会把磁盘分成系统盘还是数据盘，但在云上的系统盘和数据盘是需要分开购买的，这是一点点区别。

在**网络**这一块其实也是类似的，云上提供弹性网卡，让用户通过访问云服务器就能够联通到网上。

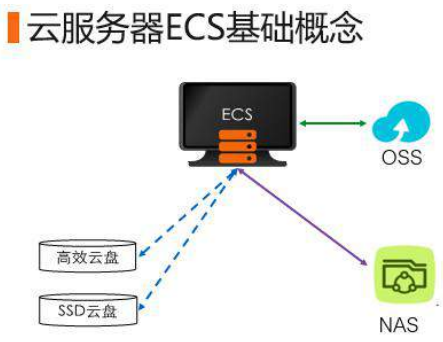
除了这些物理硬件以外，要让一个云服务器真正的跑起来，跟现实生活一样，我们也需要去安装一个操作系统，这个操作系统在云上的概念就是**镜像**，阿里云提供多种不同的镜像版本供大家选择。

除此以外，云服务器还会有一些特殊的概念，比如安全组，本质上是通过一些规则来限定访问的流量，即被哪些应用可以访问。

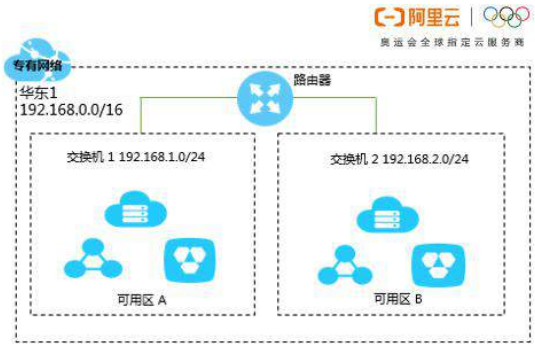
我们在买一个电脑之后，这个物理机在手上，你想要什么时候使用就可以什么时候使用。在云上买完一个云服务器之后，因为这个服务器是在云端或者说在远端，我们访问云服务器的方式就跟我们平时打开一个电脑不太一样，我们需要通过阿里云的控制台或者通过远程连接的工具来登录到我们的云服务器上去。

还有一个小概念是云上的容灾备份能力，就是**快照**。现实生活中，如果我们的电脑磁盘出现了故障，数据出现了损坏就无能为力了，或者只能够找专业的人把数据能够找回来，但是不能够保证说所有的数据都能找回来。云上有快照这样一个概念，它的意思是说对云盘的某一个时间点的数据拍一张照，本质上就是会把磁盘上所有的数据记录下来，如果出现了问题，我们就可以通过快照，快速的回滚到某一个时间点的数据，这样能够保证在业务出现了问题的情况下，快速做灾备的恢复。

整体介绍完云服务器的基本概念之后，接下详细介绍一下云服务器的存储和网络的概念。



- 块存储
  - SSD云盘/高效云盘/ESSD云盘
- 文件存储NAS
  - 可供同地域多台ECS共同使用
- 对象存储OSS
  - 可直接外链供下载使用



- VPC（专有网络）
  - 客户的云上私有网络，用于构建逻辑上彻底隔离的网络环境；每个VPC都由一个路由器、一个以上私有网段和一个以上交换机组成。
- VSwitch（交换机）
  - 是组成专有网络的基础网络设备，用来连接不同的云资源实例。用户可以在每个可用区内创建一个或多个交换机来划分子网。

云上的三种存储方式

第一种是前面已经介绍的块存储的模式，用户创建了一个块存储之后，可以把块存储挂载到实例上，就跟自己使用笔记本电脑过程中，电脑自带的磁盘不够用了，去买移动硬盘来插上来类似。块存储有三种类型，包括普通的高效云盘，还有 SSD 云盘，以及超高性能超低延迟的 ESSD 云盘。

第二种存储方式是文件存储，每一个块存储只能够挂载到一个云服务器上，而每个文件存储可以被多台 ECS 使用。

第三种存储形态是对象存储形态 OSS，这个就类似于百度云盘，使用这种存储的方式，更多的通过一个链接来做文件的读取。

云上的网络

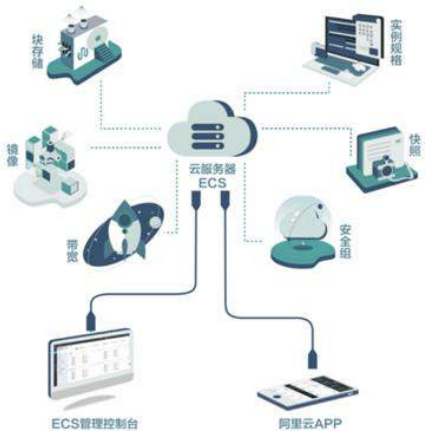
网络部分主要是两个概念，专有网络 VPC 和交换机。

第一个是专有网络 VPC，专有网络是在云上为用户划分一个私有网络，用户通过创建 VPC 可以创建逻辑上彻底隔离的一个网络环境，每一个 VPC 都是由一个路由器以及一个以上的交换机组成的。用户一旦创建了一个 VPC 专有网络，阿里云会自动为用户创建一个对应的路由器，来完成 VPC 下所有网络的转发。同一个 VPC 下的实例之间的内网是互通的，即在同一个 VPC 下实例之间可以通过内网 IP 地址来互相访问。

第二个概念是交换机，前面已经介绍了，一个 VPC 至少有一个路由器。交换机是专有网络的基础网络设备，用来连接不同的实例资源，我们可以通过交换机，在每一个可用区创建多个交换机来划分子网，然后多个交换机之间是可以通过路由器来实现连接和转发。以上是存储和网络的一些基础的概念。

云服务器 ECS 的使用流程

下面我们介绍一下使用 ECS 的流程。



名词	基础概念
ECS实例	等同于一台虚拟机，包含vCPU、内存、磁盘、网络、操作系统等软硬件的计算组件。
实例规格族	代表实例适用的不同业务场景。不同实例规格组的vCPU和内存配比、底层物理硬件等都不同，比如g6/g5代表通用型，vCPU:内存=1:4；c6/c5代表计算型，vCPU:内存=1:2；r6/r5代表内存型，vCPU:内存=1:8 等
实例规格	代表了实例的大小，即vCPU的大小，比如2xlarge代表4 vCPU
镜像	指ECS运行时所使用的操作系统及其他软件配置，阿里云提供多种镜像来源，包括官方提供的公共镜像，镜像市场、共享镜像和用户自定义镜像。
块存储	主要分为云盘和本地盘两种，主要用来做ECS实例的系统盘和数据盘，用户可以像使用物理硬盘一样格式化并建立文件系统来使用块存储。
快照	是一种便捷高效的数据容灾手段，能对块存储在某个时间点的数据进行备份，常用于数据备份、制作自定义镜像等，会额外收费。
安全组	是一种虚拟防火墙，具备状态检测和数据包过滤能力，用于在云上划分安全域；安全组规则可以控制安全组内一台或多台ECS实例的入流量和出流量。
IP地址	IP地址是访问ECS实例的主要方式，VPC下的ECS实例会有私有IP地址和公网IP地址2种，同一VPC下的ECS可以通过私有IP互相通信
地域和可用区	地域指的是ECS实例所在的物理数据中心，每个地域完全独立，同一地域下每个可用区完全隔离，但可用区之间使用低延迟链路相连。

一个 ECS 的实例，我们可以把它理解成一台虚拟机，它包含内存、磁盘、网络和操作系统等软硬件。而一个 ECS 服务器实例是多大的规格，底层的物理硬件是什么样子的，是由对应的实例规格和实例规格族来决定的。实例规格族代表了实例适用的业务场景，它决定了 CPU 和内存配比，以及底层的物理硬件是什么样子的。实例规格代表的是实例的大小，比如说 CPU 的数量是多少。

在确定了实例规格之后，我们还需要去选择对应的存储，因为只有 CPU 和内存的话，数据是没有办法存放的，所以就会有一个块存储。块存储有两种，一种是云盘，一种是本地盘。云盘其实是云上的一种三副本的存储形态，能够给用户提供高可用的能力。云盘主要用来做系统盘和数据盘，只需要像物理盘一样把它格式化就可以使用了，而本地盘可能更多的主要是用来做数据盘。

选择完了计算存储，我们接下来就要看对应的操作系统，云上的操作系统指的是镜像，目前阿里云提供是多种镜像的来源，包括官方提供的这种公共镜像、第三方市场提供的镜像、用户自定义镜像，还允许不同的用户之间共享镜像。

网络方面阿里云会有一个网络带宽，用户可以直接指定。

我们把实例的计算、存储、网络以及操作系统等参数制定好之后，就可以创建一个跟我们物理的笔记本电脑一样的云服务器。

创建完之后，我们通过阿里云的控制台，或者是通过阿里云的 APP，可以直接连接和访问已购买的云服务器。

第二部分：ECS 实例规格族介绍

第二部分我会给大家介绍一下 ECS 实例的规格族是怎么命名的，大家可能在这一块会有比较多的疑问。目前阿里云提供几百种实例规格，所以在选择的过程中会眼花缭乱，其实只要理解了 ECS 的实例规格族的命名方式，和它的信息布局，我们就能够很好的选型了。

实例的架构类型、规格分类与详细信息

在阿里云控制台的购买页面上可以看到，实例规格族的选择上分成三大模块：架构、分类、具体信息。最上面就是我们的实例规格架构的类型，有三种架构类型，分别是通用的 X86 的架构、异构计算（像 GPU 或者是 FPGA、NPU 等）、阿里云自研的神龙裸金属架构。

在每种架构下面会有实例规格的分类，从上图可以看到在 X86 的这种计算型态下，分成了 7 大类实例规格，不同实例规格代表了不同的硬件配置，选择任何一个实例规格的分类之后，我们可以看到对应实例规格的详细信息，这些信息主要分为四部分：

- 第一个就是实例规格族的详细信息，包括对应的规格族和实例规格的代称，这里可以通过点击小问号，能够看到实例规格族的一些详细的描述。
- 第二部分是 CPU 和内存大小的信息，这里是大家在选型的过程中会比较关注的。
- 第三部分是实例的网络能力信息，包括实例内网的带宽和收发包的能力。
- 第四部分是 CPU 的处理型号的信息，包括处理器的主频和睿频这两部分信息。

企业级实例 VS 入门级实例

在控制台的购买页面上可以看到，ECS 的实例规格族特别多，单纯从 CPU 和内存是无法判断它们的区别，所以我们需要从宏观上来看。阿里云 ECS 的实例规格整体是分成两大类，一类是企业级实例，一类是入门级实例。

企业级实例是阿里云在 2016 年 9 月份才推出的，其特点是 vCPU 是独享的，也就意味着我们创建一个企业级实例的时候，实例 vCPU 与我们底层物理的 CPU 是绑定了的，底层的物理 CPU 就不可能再分配给其他的实例了，所以企业级的实例不会出现资源的争抢，因此能保证性能稳定，并且企业级实例提供了非常严格的 SLA 性能保证。

而入门级实例就是 vCPU 跟底层的物理的 CPU 是不绑定的，意味着可能每个 vCPU 是随机分配到底层的空闲的一个物理 CPU 上，如果同一个物理的物理服务器上有多个共享入门级实例的话，不同的实例就会出现资源的争抢，导致 CPU 的性能不稳定。

因为入门级实例存在性能不稳定的特性，所以阿里云现在仅仅提供一种入门级实例，就是在 X86 架构中的共享型实例， 而 X86 架构中的其他实例规格，以及异构架构和神龙架构中的所有实例，都是属于企业级实例。

由于企业级实例性能稳定，并且有严格的 SLA 的保证，所以它比较适合于对业务稳定性有比较高的要求的场景。入门级实例由于不能够保证性能稳定性，所以价格相对便宜，比较适合于一些对性能没有严格要求，或者在某些时段下才会有性能突发要求的场景，比如有些轻负载的应用或者是微服务。

共享型实例

在介绍完 ECS 实例大的分类之后，我们来看一下共享型实例的具体信息。

标准共享型实例（s6/xn4/n4）

- 采用非绑定CPU调度模式,不同实例vCPU会争抢物理CPU资源，并导致高负载时计算性能波动不稳定
- 有可用性SLA保证，但无性能SLA保证

突发性能实例（t6/t5）

- 基准性能是实例可以持续稳定地提供的CPU性能。
- 可持续获得CPU积分，在性能无法满足负载要求时，可以通过消耗更多CPU积分无缝提高计算性能，不会影响部署在实例上的环境和应用

我们前面讲到了只有 X86 架构下的共享型实例才是入门级实例。这类实例比前面实例在四要素以外多出一个参数，即“平均基准的 CPU 计算性能”，基准性能即实例能够持续提供的 CPU 性能。

共享型实例也就是入门级实例，分成两大类，第一类是属于标准的共享型实例， CPU 是不绑定的，只提供基准 CPU 性能，所以当出现资源的争抢，是否能超出基准性能是没有保障。

另外一种特殊的共享型实例，名为突发性能型的共享实例，它主要就是照顾到某些应用在绝大多数的时候 CPU 的使用率可能都不高，负载都不高，但是在某些时候可能会有临时的突发的高性能要求，所以阿里云会提供突发性能的参数，所以您在购买共享型实例的时候，能够通过突发性性能来获得高于平均基准 CPU 性能的能力。

突发性能型的共享实例，如果应用实际用量低于了平均的基准性能，会获得对应的 CPU 的积分，如果在某些场景下性能要求突然提升之后，比如实例对应的 CPU 的使用率超过了 20%，会消耗之前累积的 CPU 的积分，去提升计算性能，让计算性能不会受到影响，这个是突发性能的共享型实例独有的特性。

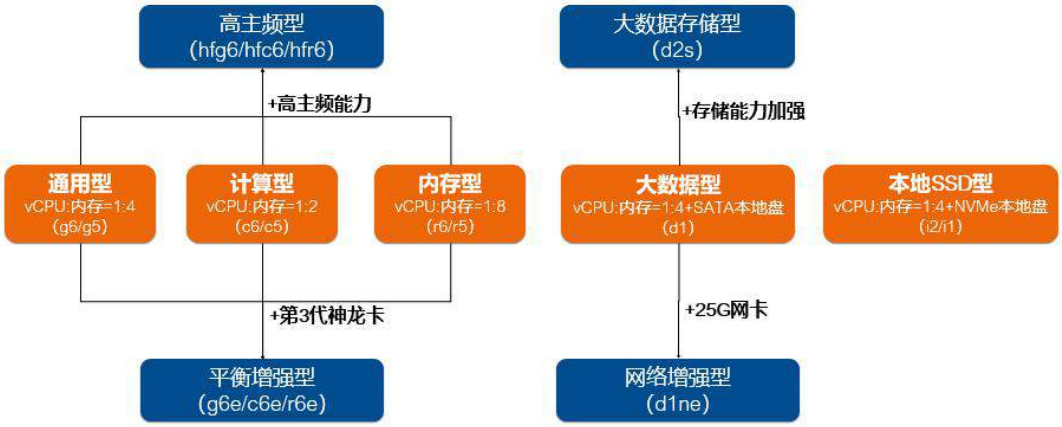
两个特殊的实例规格

除了共享型的入门级实例以外，阿里云还有两个实例规格比较特殊，就是大数据型和本地 SSD。

这两种实例规格会附带一个本地存储，大数据型实例的本地存储是 HDD 盘，本地 SSD 新增的本地存储是具有非常高 I/O 吞吐，并且有低延迟的本地 SSD 盘，具体的信息大家可以在阿里云控制台查看。

企业级实例规格家谱

下面介绍企业级实例规格的家谱，方便我们快速了解各个实例家族的“亲属”关系。企业级实例规格族分成三大块，第一大块是 X86 计算，除了共享型以外，包括通用、计算、内存、高主频、本地 SSD 和大数据型都属于我们的企业级实例，企业级实例每年都在不停地迭代，所以会分成不同的代系，我们在后面会详细介绍不同的代际之间的区别。异构计算里面所有的 GPU 和 FPGA 都是属于企业级的实例，裸金属和高性能计算也是一样的。



首先，我们来介绍 X86 的实例规格的命名方式，分成了 5 种：

第一种实例规格是通用型，顾名思义就是什么场景都能够用，所以这种型号的代称是 g 系列，它的 vCPU 和内存的一个配比是 1:4。

第二种实例规格是计算型，顾名思义就是在某些场景下对 CPU 算力的要求会更高一点，所以它的 vCPU 和内存的配比是 1:2，然后简称为 c 系列。

第三种类型是内存型，提供更多的内存能力，所以它的 CPU 和内存的配比是 1:8，也简称为 r 系列，r 是 RAM 的简称

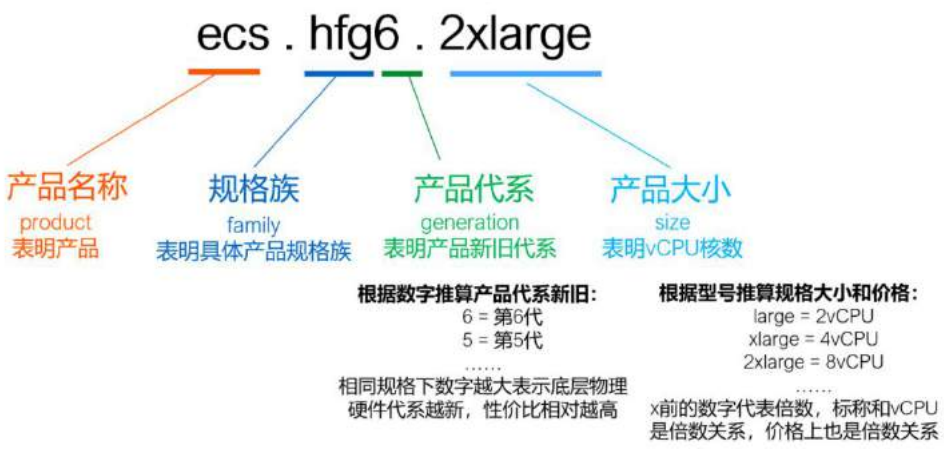
第四种和第五种分别是大数据型和本地 SSD 型，这两种的 CPU 和内存的配比都是 1:4，只是它们配的本地盘的类型是不一样的，导致它们的技能和适合的场景也是不一样的。所以大数据型的简称是 d，本地 SSD 型简称是 i。

在这 5 个基础的实例规格上面，我们会去做一些额外的能力提升，比如说在通用型、计算型和内存型这三种类型下，增加了一些高主频的能力，正常的 CPU 的主频应该是 2.5G 赫兹，但是我们有一些可以是做到 3.2G 赫兹，这种加上高主频的能力就变成了高主频型，会在前面去加上一个 hf 这样的一个标识。

随着技术的演进，神龙架构的神龙卡也是在不断地迭代和改善，搭载了第三代的神龙卡可以整体提升通用型、计算型和内存型这三种实例规格的性能，所以就会出现一个平衡增强型。对于大数据型的话，做了计算和存储的分离，形成了大数据存储型，简称为 d2，而 d2s 是在大数据的基础上，做了一些网络能力的增强，就变成了一个网络增强型。

实例规格的命名方式和规律

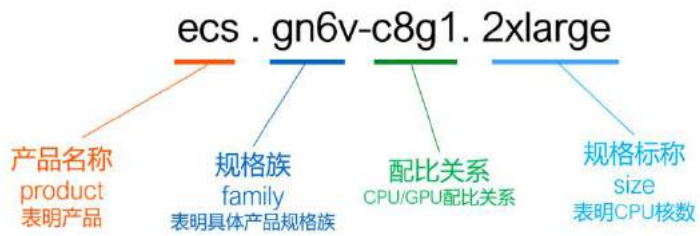
大家通过下图能够看到阿里云实例规格的命名方式和规律。



普通的 X86 实例规格名称是分成了三段，第一部分表示的是产品名称，ECS 是阿里云的产品；第二部分表示了实例的规格和代系，前面已经讲过 hfg 表示是在通用型的基础上增加了高主频的能力，然后 6 代表的是什么？其实它代表的是我们产品的代系，可以根据产品的代系推算对应的产品的一个新旧，比如说 6 代表第 6 代，5 代表的是第 5 代，这个数字越大代表它是更新的一个代系，它底层的物理硬件也会越新，它的性价比相对而言也会越高。

最后一部分是实例的规格，表示的是实例的 vCPU 的核数，large 代表 2 个 vCPU，xlarge 代表 4 个 vCPU，2xlarge 代表的是 8 个 vCPU，以此类推。

了解了以上命名规律，就能通过实例规格族的名称推断出来当前这个实例的 CPU 是什么型号、它的是什么样的代系，以及它的 CPU 的数量是多少。

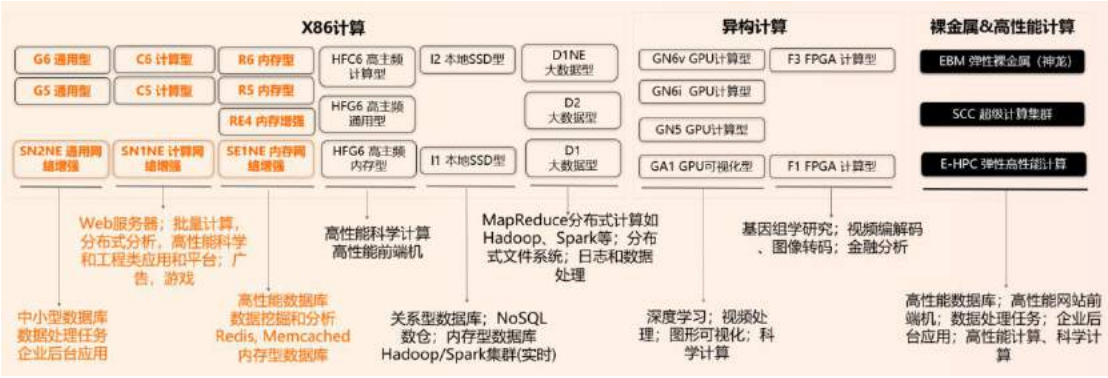


GPU 命名规则也是类似的，只有一个不一样的点，GPU 名称的的中间这一部分会提供 CPU 和 GPU 的的配比关系，因为 GPU 是除了 CPU 以外还会提供一个额外的 GPU 的卡。所以我们也是直接可以通过它的规格族的格式，能够去推断出来它底层的物理的配置。

第三部分：ECS 实例选型实战

第三部分给大家实战讲一下如何做云服务器 ECS 的选型。

简述各种规格实例的适用场景



X86 计算：

- X86 的通用型、计算型和存储型三种实例，CPU 和内存的配比比较一致，所以比较适合做一些中小型的数据库，或者是一些数据处理的任务。
- c 系列的话，主要是计算型，所以比较适合于做一些计算要求比较多的，比如说做一些外部应用，或者做一些批量计算，或者是一些高性能的科学计算类的。
- r 系列的话，因为它的内存比较多，所以比较适合于做一些数据库或者数据分析的应用。
- 高主频实例规格也是比较适合于对 CPU 的主频有比较高要求的高性能科学计算。
- 本地 SSD 类型，更多的适合于做一些关系型数据库或者是 NoSQL 数据库的。
- 而 D 系列的大数据型，可能更适合于做一些大数据集群的一个场景，比如说像这种 Map Reduce 这种。

在异构这一块，分成了两大类：

GPU 比较适合于做深度学习或者是图像视频的可视化的处理；
FPGA 就比较适合于做图像的转码，或者音视频的解码。

裸金属和高性能计算：

更垂直和性能要求更高的一些场景，像一些高性能的数据库或者高性能科学计算场景。

下面我们举几个例子详细介绍一下选型方法。

X86 实例选型推荐

场景	实例推荐	首推实例
Web服务器	计算型	C6/C5/IC5
中间件	通用型	G6/G5
应用服务器	通用型	G6/G5
缓存	内存型	R6/R5
数据库	内存型 本地SSD型	R6/R5+ESSD I2/I1
大数据	大数据	D1NE
AI机器学习	GPU计算型	GN6v/GN5

我们可以把一个 web 应用分成以下几个层次，每个层次做对应的推荐：

- 对于 Apache 和 Nginx 的 web 服务器，，因为它主要做一些计算处理，所以推荐是使用一些计算型的，比如说 c5、c6 这样的；
- 对于像 spring cloud 或者说 MQ 这样的中间件的话，它是属于对于计算和存储的诉求都比较正常的，所以我们是推荐一些通用性的，比如说 g6 这样的实例规格；
- 而应用型因为是属于比较通用的场景，所以 G6 系列就能够满足；
- Redis 和 Memcache 这种缓存应用，对内存的要求是比较高的，所以我们推荐使用内存型的，像 r 系列；
- 对于关系型数据库，我们是可以直接使用内存型，比如说 r 系列配上我们的 SSD 云盘；
- 对于 NoSQL，我们推荐本地 SSD 型的，比如 I 系列；
- 对于大数据的话，类似于 HDFS 或 spark 的这种，我们也有专门的大数据型的，像 d 系列这种的来处理；
- 对于最底层的机器学习的，比如 MXNet 这种训练框架，会有对应的专门的 GPU 计算型。

GPU 实例 选型推荐

GPU 云服务器的场景主要分成两大类，第一大类是人工智能，或者叫机器学习，第二块是图形图像的处理。在机器学习里面也会分成两个场景，一个是训练，一个是推理。所以对于不同细分的垂直领域，我们给了一些规格的推荐，具体可见下图。

下面我们介绍两个相对而言比较复杂的选型场景。

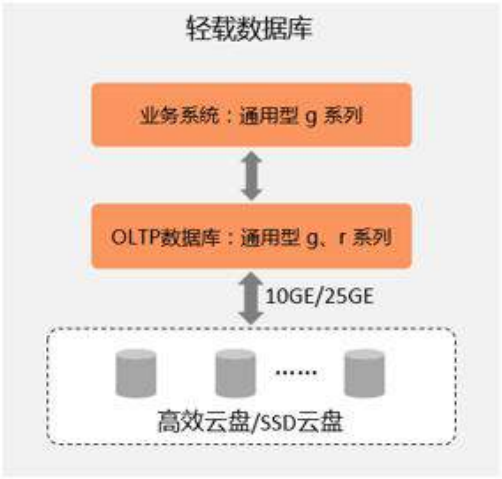
大数据场景实例选型实战

第一个复杂场景是大数据的场景，类似于 Hadoop、Spark 这种大数据集群搭建的时候，如果我们自己手动做搭建，会把过程分成三大块：第一大块就是集群的管理节点的实例规格选型，第二块是集群的计算节点的选型，第三块是集群的数据节点的选型。

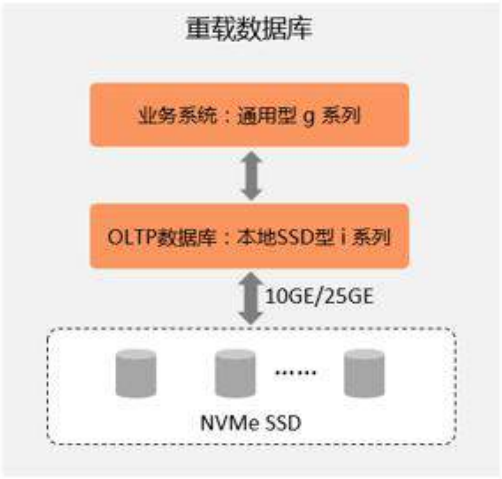
- 管理节点是比较通用的场景，所以直接选择 g 系列就能够很好地处理管理的任务；
- 计算节点更多的是属于比较偏正常的业务负载，所以可以把 g 系列作为主要的选择，搭配 SSD 云盘；
- 数据节点对存储的吞吐和网络的吞吐有比较高的要求，所以推荐使用 d 系列，搭配对应的本地盘，能够完成这种数据的读取量；

所以整体来说，在同样一个大数据的集群里面，不同的任务有不同的特征，所以会选择不同的实例规格。

数据库场景实例选型实战



采用通用型实例规格搭配云盘使用，性价比更高



采用高存储IOPS和低读写延时的本地SSD型实例规格，满足性能要求

第二个复杂场景是关于数据库选型的：

- 对于普通的业务，负载比较轻的数据库，有专门的通用型 g 系列，或者内存型 r 系列搭配高效云盘和 SSD 云盘就能处理，性价比会比较高。因为 g 系列和本地盘或者本地 SSD 比起来，价格还是很有优势的。高效云盘和 SSD 云盘的整体性能，其实也是能够满足日常数据库的场景的。
- 对于业务负载要求非常高的集群，推荐本地 SSD 的 i 系列搭载 NVMe SSD 的云盘，能够实现存储的高 IOPS 和低延时，能够满足重载数据库的性能要求。

X86 第 6 代 vs 第 5 代 实例价格对比

除了性能以外，大家也会关注价格，这里有一个 X86 里面第 6 代和第 5 代的一个价格的对比。

- 计算型C6对比上代C5于北、上、杭、深、青月价高4%，张北 低2%
- 通用型G6对比上代G5全地域月价 低6%-12%
- 内存型R6对比上代R5全地域月价 低2%-10%
- 按量付费全部比上一代 低37%-47%

	第6代实例			华北2		华北3		第5代实例			华北2		华北3		价格对比（6代/5代）			
	规格	vCPU (核)	内存 (GB)	按量	月价	按量	月价		vCPU (核)	内存 (GB)	按量	月价	按量	月价	华北2月价	华北3月价	华北2按量	华北3按量
计算型	ecs.c6.large	2	4	0.39	187	0.27	131	ecs.c5.large	2	4	0.62	179	0.47	134	104%	98%	63%	57%
	ecs.c6.xlarge	4	8	0.78	374	0.55	262	ecs.c5.xlarge	4	8	1.24	358	0.93	269	104%	97%	63%	59%
通用型	ecs.g6.large	2	8	0.5	240	0.35	168	ecs.g5.large	2	8	0.89	255	0.66	191	94%	88%	56%	53%
	ecs.g6.xlarge	4	16	1	480	0.7	336	ecs.g5.xlarge	4	16	1.77	510	1.33	383	94%	88%	56%	53%
内存型	ecs.r6.large	2	16	0.66	318	0.46	220	ecs.r5.large	2	16	1.13	326	0.85	245	98%	90%	58%	54%
	ecs.r6.xlarge	4	32	1.33	636	0.92	440	ecs.r5.xlarge	4	32	2.26	652	1.70	489	98%	90%	59%	54%

可以看到除了计算型的实例在某些区域下，第 6 代实例会比第 5 代 10 实例的价格会略高 4%以外，通用通用型和内存型的包月价格，第 6 代普遍比第 5 代要便宜 2%-12%，所以整体来说的话，第 6 代不仅仅是性能有 20%的提升，而且绝大多数的产品会更便宜。

而按量付费的话，第 6 代的价格比第 5 代的价格会低 37%-47%，这其实是一个非常大的让利的空间。所以在选择按量去购买 ECS 的时候，选择第 6 代会比第 5 代要便宜的要便宜的更多。

选型实战总结

总结选型方法，有三个法则，大家可以记在心里面，在选型的过程中运用。

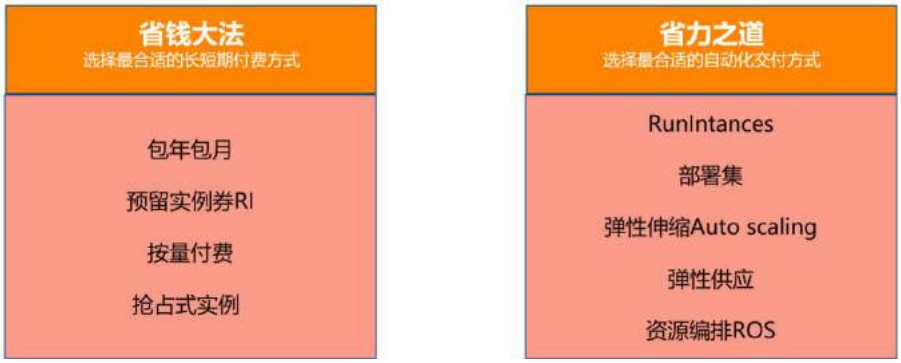
第一个法则是相同大小的企业级的实例比入门级的实例性能更稳定，但是入门级的实例性价比更高，因为企业级的实例它是独占了 vCPU，不存在一个资源的争抢，有性能的保障，但是对于一些个人或者中小网站的应用，如果对性能的诉求并不是那么强的话，选择入门级的实例其实是一个更好的选择。

第二个法则是在相同的实例规格下，新一代的实例规格比老一代的实例规格性价比更高，这是因为新一代的实例规格，做了很多技术的演进和更新换代，能够给公有云用户释放更多的技术红利。

第三个法则是选型时不仅仅要选择合适的实例规格，而且还需要搭载合适的块存储，才能够让云上的应用达到预期的性能。云上会提供 4 种不同的块存储，包括高效云盘、SSD 云盘、ESSD 云盘和本地盘，不同的类型盘的 IOPS 和吞吐是不一样的，所以不仅仅要选合适的实例规格，还要选择合适的块存储，才能够形成合力，达到最佳的性能。

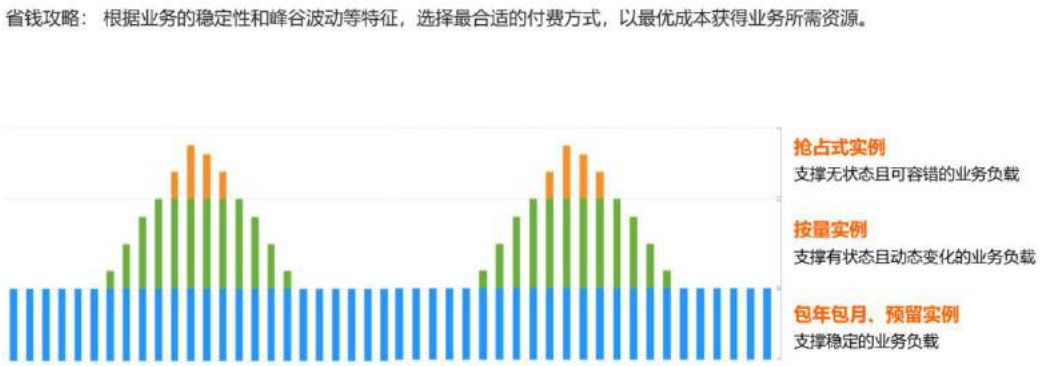
第四部分：ECS 省钱省力之道

在购买云服务器的时候，除了要做实例规格的选型，让选择的实例规格和业务的匹配度更高以外，我们还需要去考虑能不能更便宜，能不能够快速完成资源的交付，所以最后一部分给大家介绍一下 ECS 省钱省力的技巧。



省钱大法

第一个是省钱大法，省钱大法意思是选好了实例规格，还需要选择最适合的付费方式，才能够得到更好优化成本。阿里云目前提供 5 种付费方式，分别是节省计划、包年包月、预留实例券、按量付费、抢占式实例。



如何选择合适的付费方式呢？有一个攻略，就是我们需要根据业务的稳定性和峰谷的波动情况，来选择最适合的付费方式。像节省计划、包年包月、预留实例券就比较适合于稳定的业务负载；有状态并且是动态变化的业务负载的话，可以使用按量付费；而对于完全没有状态，并且具有很高的容灾能力的，可以使用抢占式的实例来交付，因为抢占式实例的价格是可以做到按量付费实例的 10% 的。

省力之道

第二个是省力之道。在云上购买资源的时候，有时候会批量购买，阿里云会提供多种自动化的资源交付模式和工具，能够实现一次配置重复使用，从而提升整个云上部署的速度和效率。

省力攻略：选择合适的自动化工具，可实现一次配置重复使用，提升云上应用部署的速度和效率

比如通过控制台做批量的交付；通过部署集可以完成底层具有容灾能力的算力集群的交付；通过弹性伸缩和弹性供应，能自动化地完成资源的交付；而通过资源编排，可以把多种不同的资源组合交付。

上云选型四步走

总结一下，上云的过程中，我们需要走好四步：

第一步：对自己的业务特征做一些分析，包括对性能的要求，对网络的要求，形成一个基本的判断；

第二步：针对业务特征来选择对应的 ECS 实例规格；

第三步：选择对应的一个付费方式，只有选择最合适的付费方式，才能够实现云上的成本最优；

第四步：选择合适的交付方式，帮我们省时省力地完成资源的交付。

以上就是我今天分享的主要内容，然后如果大家有什么疑问，欢迎大家直接在下面留言跟我互动，谢谢大家的观看。

更多细节欢迎关注电子书《玩转 ECS 从入门到精通（入门篇）》

第二章 自动化运维

2.1 服务器自动化迁移中心 SMC 最佳实践及新特性介绍

摘要：本次分享由阿里云技术专家白辉万（百宝）为大家介绍免费的服务器迁移上云最佳实践方案和新功能特性，包括一键迁云、自动定期同步、一键验证。本次分享内容将帮助企业上云客户越过高高的服务器迁移门槛，快速体验搬站上云新姿势，助力企业数字化转型，适合上云客户、企业 IT 设施运维技术团队、迁云服务商等收看。

演讲嘉宾简介：白辉万（百宝），阿里云技术专家，2017 年加入阿里云，主导服务器迁移中心产品方案开发工作，致力于优化服务器迁云体验；在 Windows/Linux 服务器系统迁移、企业搬站上云等方面拥有丰富的解决方案和实战经验。

本次分享主要围绕以下三个方面：

一、服务器迁移中心 SMC 介绍
二、SMC 最佳实践使用示例
三、SMC 新特性功能介绍

云服务器 ECS（Elastic Computing Service）是每个阿里云用户上云的“第一步”，为了帮助大家更加方便的服务器迁移高门槛，本次分享邀请了阿里云技术专家白辉万（百宝）为大家介绍免费的服务器迁移上云最佳实践方案和新功能特性，帮助大家快速体验搬站上云新姿势。

一、服务器迁移中心 SMC 介绍

阿里云服务器迁移中心 SMC（Server Migration Center），也叫迁云工具，是免费自助式服务器迁移服务，已经于 2017 年 11 月发布上线。SMC 核心功能是帮助企业客户更加方便快捷地将服务器系统数据迁移上云。

1. 企业迁云的几个重要阶段

企业上云的步骤分为以下四步：

第一步是迁移前评估，如进行服务器应用业务的资产分析、制定资金迁移计划、进行迁移测试等。

第二步，实施服务器迁移。

第三步，云端进行服务器业务的验证。

第四步进行业务切换。

注意：如果有其他数据迁移，如海量数据迁移可使用闪电立方，数据库迁移可使用DTS，在此不做展开介绍。

2. 传统迁移方式痛点明显

服务器迁云的过程并非简单工作，传统迁移方式痛点明显：

应用复杂难还原：若企业业务较为老旧，处于无人维护状态，则非常难以重新部署。

周期长影响业务：大量服务器迁移耗时长，容易中断，易导致重复耗时。

人员投入大、效率低：迁移消耗大量人力资源，操作人员技术门槛高，影响正常业务迭代。制作镜像，数据导出导入耗时、效率低。

3. 新一代迁移方式：服务器迁移中心

根据上述痛点问题，阿里云推出了新一代迁移方式：服务器迁移中心，具有以下优势：

高度成熟化，适用各种迁移场景。支持系统盘+数据盘整体迁移，无需重新部署。同时兼容各个服务器系统平台，包括物理机、虚拟机以及各大云平台。基本覆盖所有主流 Windows、Linux、32 位、64 位操作系统版本。

高度自动化，一行命令，无人值守。迁移过程从计算同步到镜像制作到最后迁移结果的验证都可以自动化完成。支持自动定期增量同步的灵活方案，大大减少迁移周期。

高度智能化，自动检测、自适应修复。自动对源服务器进行迁移条件检测并自动提供修复方案，迁移完成后自动完成驱动修复和 cloud-init 安装，保障整体迁移效果，同时无需过多人力干预，最大程度减少人员投入。

二、SMC 最佳实践使用示例

SMC 的特点是轻量、好用、操作门槛非常低。SMC 主要迁移步骤如下，分为源服务器端操作和控制台操作。

产品入口
https://smc.console.aliyun.com/home
帮助文档
https://help.aliyun.com/product/121538.html

产品入口
https://smc.console.aliyun.com/home
帮助文档
https://help.aliyun.com/product/121538.html

1. 迁移简单演示

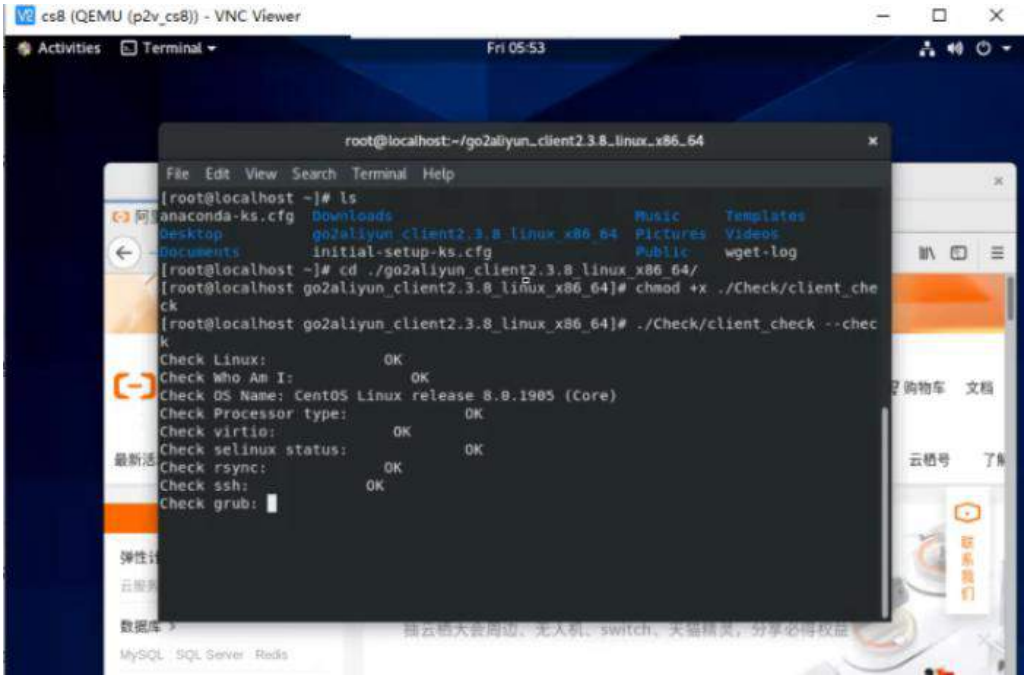
首先登入阿里云云服务器 ECS 控制台，在“部署与弹性”列表点击“服务器迁移中心 SMC”即可进入 SMC 使用页面。

“概览”页面是 SMC 迁移源与迁移任务概况，可直观看到迁移源与迁移任务状态与统计分析情况。

第一步，导入迁移源。下载 SMC 客户端后，找一台线下服务器系统，此处是一台 cs 8 虚拟机，进入工具目录。首先需要检测服务器系统是否符合迁移条件，赋予执行权限后可使用工具包中的检测工具：

```
sudo chmod +x ./Check/client_check
sudo ./Check/client_check --check
```

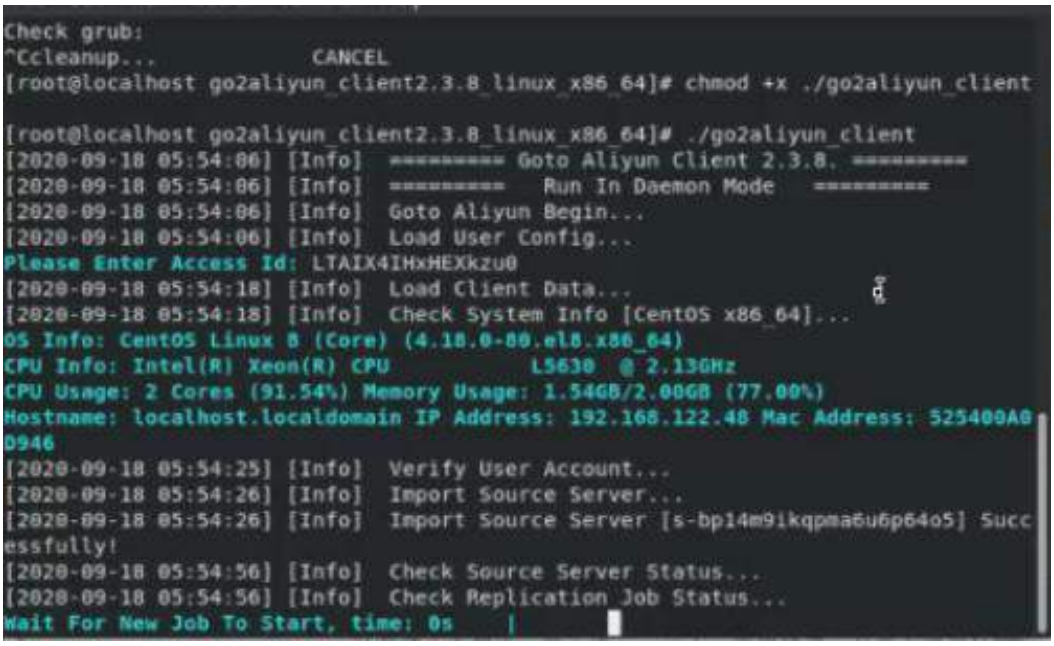
如下图所示，所有条目显示 OK 时，表示该服务器系统满足迁移条件。



接下来运行主程序：

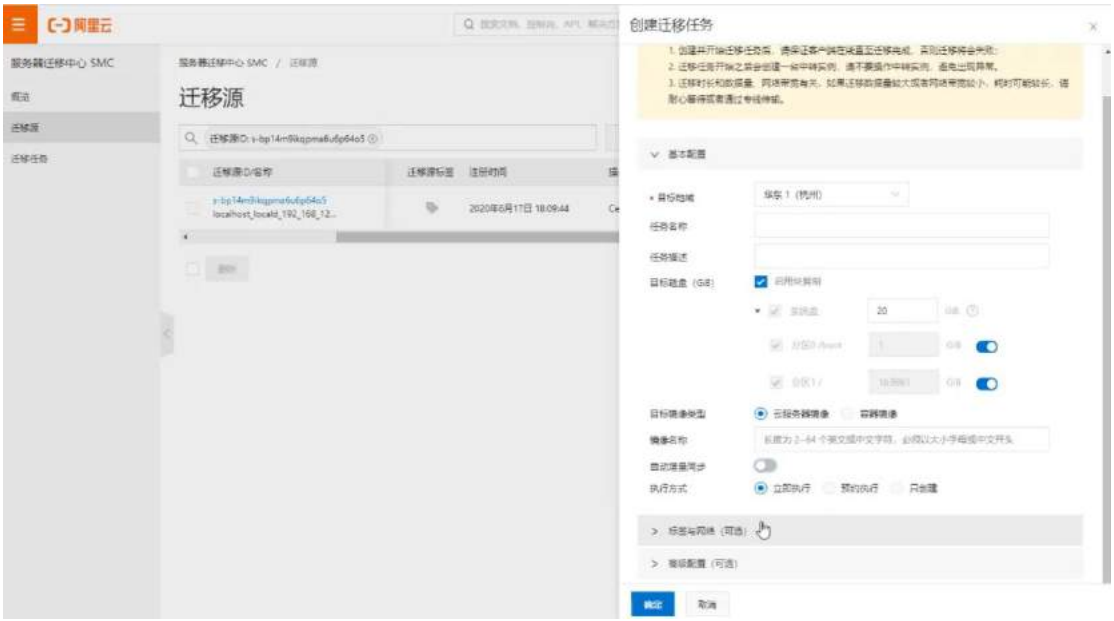
```
sudo chmod +x ./go2aliyun_client
sudo ./go2aliyun_client
```

运行后窗口将提示输入阿里云账号 AK 与 SK。输入后继续运行，开始将源服务器系统导入阿里云。

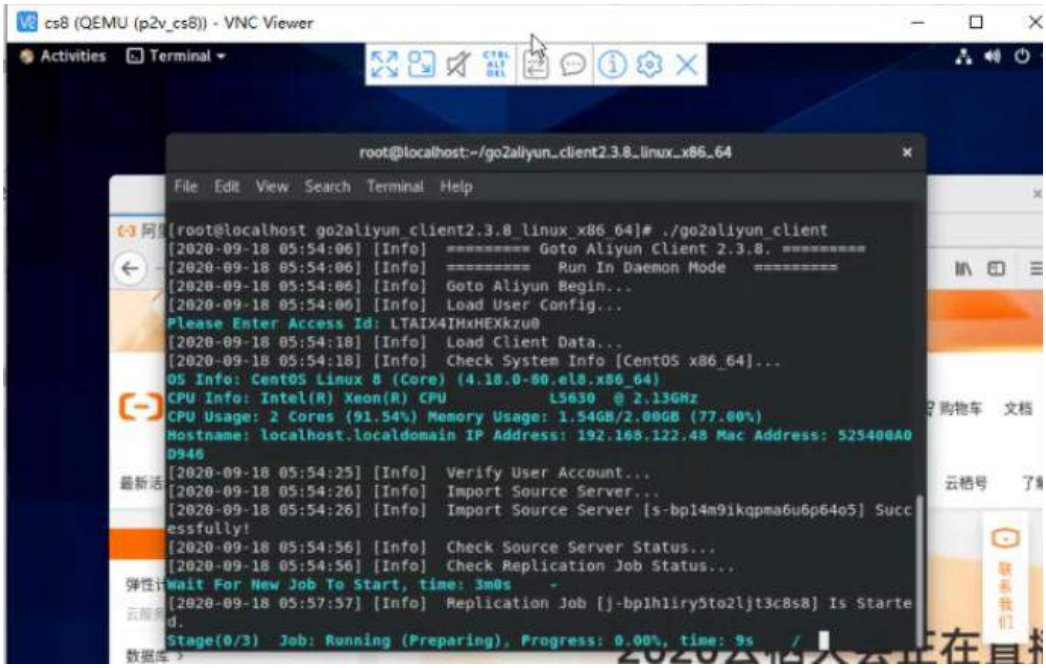


第二步，回到 SMC 控制台，根据迁移源 ID 等找到对应迁移源。如下图所示。注意迁移过程中要始终保证迁移源处于在线状态。在“迁移源详情”可查看迁移源主机名、IP 地址、CPU 等基本信息。

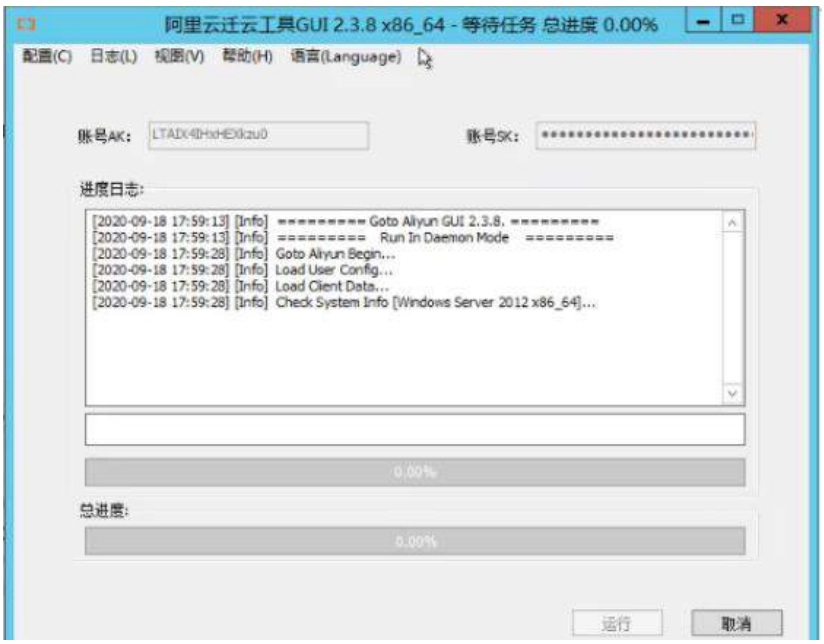
第三步，创建迁移任务。创建迁移任务基本配置需要选择目标区域、适合迁移的磁盘列表、是否开启自动增量同步等等，并可选高级配置。确认信息无误后点击“确定”，则开始迁移任务成功，接下来等待迁移任务完成即可。



迁移时可随时到迁移任务列表观察迁移状态以及进度。若出现迁移失败等情况，可查看日志。SMC 迁移客户端中也会实时显示迁移状态信息。



Windows 系统迁移与 Linux 系统大同小异。Windows 提供了 go2aliyun_client.exe 应用程序，其执行方式也依赖控制台。另外针对 Windows 客户还提供 go2aliyun_gui.exe 可视化界面程序，较符合 Windows 客户操作习惯。输入账号 AK 与 SK，点击运行即可。在此不作其他介绍。



导入迁移任务前下载迁移模板，按规范值填入迁移信息即可批量导入迁移任务。迁移速度依赖于迁移源服务器系统的带宽与数据量。

SMC 迁移支持完整的 Open API 操作，方便具有二次开发需求的用户。

2. 与传统服务器迁移方式对比

SMC 主要操作步骤即下载工具客户端并运行、创建迁移任务邓艾迁移完成两步。对比传统迁移方式，SMC 在操作容易度、迁移速度、系统还原度方面均具有明显优势。

3. 迁云最佳实践建议

根据迁云经验总结，阿里云提出了迁云最佳实践建议，具体为如下表五部分。

4. 企业上云最佳实践

为大家推荐阿里云交付专家团队配合使用 SMC 制作的服务器搬迁实践方案，该方案具有完整的实际操作流程。

服务器搬迁实践方案

出品人：阿瑟（业务支撑团队）

三、SMC 新特性功能介绍——高性能、多方案、自动化

该部分主要介绍近一年来 SMC 发布的新功能特性，聚焦于高性能、多方案、自动化。

1. 快复制迁移功能支持

功能特点：支持 Windows/Linux 前平台快复制迁移功能，提供更高效稳定的迁移效果。

相比文件复制功能，快复制可更有效提升网络使用率，达 90%，同时保证迁移后能与源磁盘分区一致，迁移效果更优。

在快复制推出的基础上，SMC 还支持文件复制、快复制混合迁移，满足更灵活的迁移场景需求。

2. 迁移到目标实例支持

功能特点：满足提前创建目标实例的迁移场景，进一步缩短 cutover 时间。

该功能是基于客户使用场景需求而推出的。一些客户在源服务器系统迁移上云之前就已经购买了目标实例。以前使用 SMC 迁移后只能得到迁移的景象，若带有数据盘，实际迁移操作则更加麻烦。该功能同时可以减少整体迁移时间。

3. 迁移结果自动化验证

功能特点：一键进行迁移结果自动化验证，进一步丰富迁移自服务体验。

迁移完成后，在云端进行业务验证阶段必不可少。SMC 针对该需求提供了一键自动化验证的流程方式。首先将自动使用迁移后的镜像创建临时目标实例。接下来自动安装云助手，然后执行内置系统验证脚本。同时为客户开放了入口，客户可自定义脚本进行业务与系统验证。执行验证完成后可得到验证结果，让客户直观地了解当前系统是否正常。

该功能仍在进行持续完善，将进一步丰富迁移自动化服务的体验。

4. 容器镜像迁移功能新鲜发布

功能特点：支持将服务器系统一键迁移到阿里云 docker 镜像。

2019 年云原生进入商业化爆发增长期，越来越多用户想将自己的应用业务做云原生化改造。

阿里云针对该场景实现并发布了一套将普通服务器系统一键迁移到容器镜像的方案，正正助力企业方便快捷地进行云原生改造、验证。该功能比较新鲜，非常具有前瞻性。

5. 持续增强的企业级迁移方案

阿里云服务器迁移中心自 2017 年推出迁云工具公测版本 1.0 开始，始终为增强企业级迁云方案而努力，并收获了如下成果。

阿里云服务器迁移中心目前为止已经历经 70+迭代版本，服务用户数 3000+，最大单个迁移超 500 服务器，迁移成功率大 97%，并且仍然处于不断改进、持续提升用户体验的状态，更多新特性与核心功能正在计划与实施当中。

本次分享到此结束，有兴趣或有其他需求的用户可持续关注 SMC 服务支持群。

2.2 云上自动化部署和运维的正确姿势

摘要：云上部署和运维相对于传统方法而言更加灵活，只需要编写一次模版就可以随时随地拉起一套环境，做到一键部署。同时支持多环境部署、操作可审计、支持 DevOps 实践。本次分享由阿里云资深技术专家吴君印为大家介绍介绍上云最正确的部署和运维方式，结合阿里的最佳实践，打造快速、安全、可复制、标准化的 DevOps 体验。

演讲嘉宾简介：吴君印，阿里云资深技术专家。负责 ECS 整体服务层面的技术和产品架构工作，并负责阿里云智能内部 OnECS 的技术和产品架构工作，包括产品 ECS 云助手，运维编排 OOS，资源编排 ROS 以及内部 OnECS 产品宙斯，致力于打造以 ECS 为中心的系统管理、自动化和 DevOps 体验。

本次分享主要围绕以下三个方面：

一、云上部署和运维的特点
二、资源编排 ROS
三、运维编排 OOS

今天主要分享的内容是云上自动化部署和运维的正确姿势，下面先来看看云上部署相比于传统的 RDC 部署有哪些不同。

一、云上部署和运维的特点

无论是部署还是运维，在云上都有如下四个特点：

首先，可重复。在云上部署相对于传统 RDC 部署而言更加灵活，只需要编写一次模版就可以随时随地拉起一套环境，做到一键部署。目前有两种类型的环境部署，一种是测试环境、预发环境、生产环境。第二种是在不同地域进行部署，如北京地域、上海地域以及杭州地域。

第二点，多环境保持一致。因为使用的是相同的模版进行部署，所有环境部署出来的结果都一样，这样可以避免人为错误，避免问题排查时的环境影响，环境造成的问题往往是最难排查的问题之一。

第三点，可审计。所有操作均通过 API，所有 API 操作都可以被审计，集成操作审计服务 ActionTrail 即可。

第四点是 DevOps。从环境部署到应用部署都模板化，版本管理使用 Git，可以做代码评审、代码回滚。

资源编排 ROS 和运维编排 OOS

阿里云针对云上部署和运维特点，推出了两个产品，包括资源编排 ROS（Resource Orchestration Service）——解决自动化部署问题，运维编排 OOS（Operation Orchestration Service）——解决自动化运维问题。两款编排产品除了支持 ECS 的实例，还支持其它阿里云的产品，如负载均衡，关系型数据库 RDS，对象存储 OSS 等。

二、资源编排 ROS

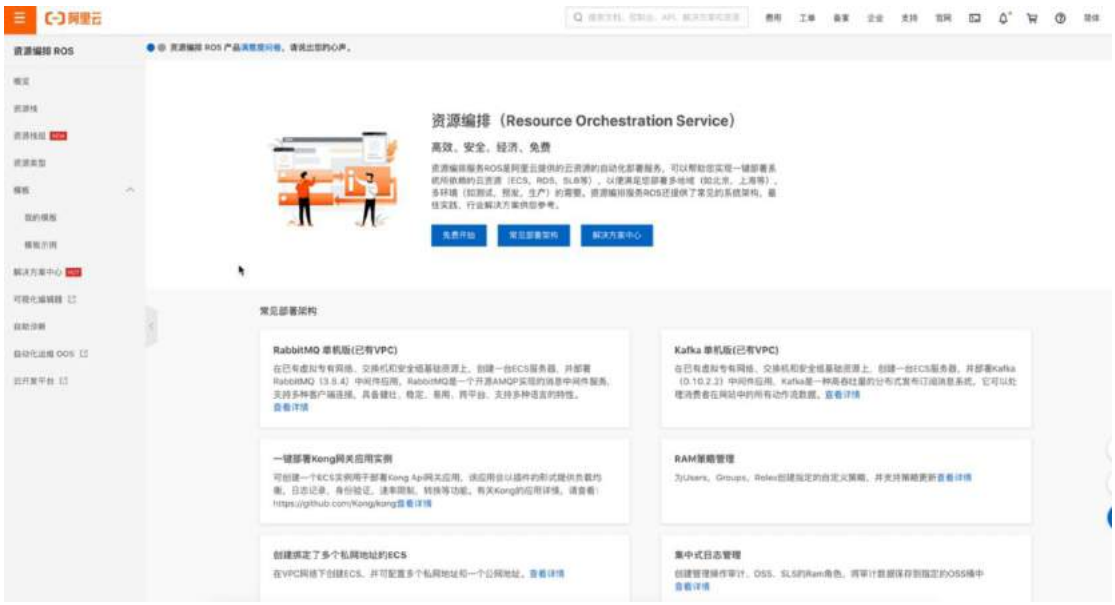
资源编排 ROS 的典型场景

资源编排 ROS 的典型场景主要有四种：

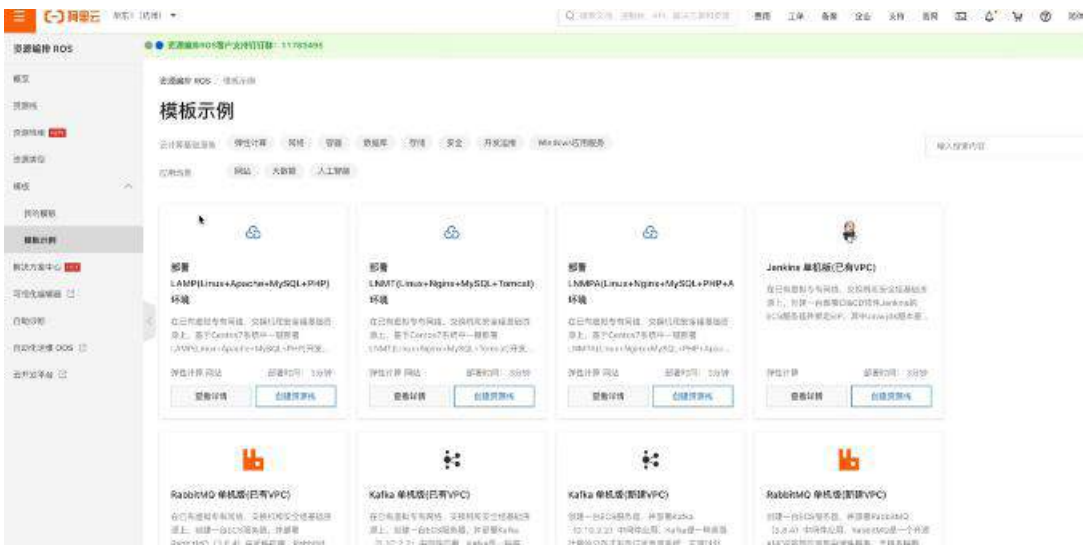
- 第一种是部署模版，资源编排 ROS 是通过模版方式达到可以重复部署的目的，使用模版可以在任何时间任何地点拉起一套环境。
- 第二种是 MSP、ISV 提供自己的部署模版，可以一键开出复杂的业务系统，如 SAP HANA 等系统，将部署时间缩短为几个小时。由于云上的环境都是标准的，只要有测试通过后的模版就可以在不同的环境、不同的账号中重复部署。
- 第三是解决方案中心，阿里云通过自身多年服务客户和双 11 的经验，总结了大量的最佳实践，在解决方案中心中提供了很多高质量的模版，支持开箱即用。
- 第四是 CI/CD 集成，在 DevOps 开发模式下，只有将部署模版放到 CI/CD 中才能打造 DevOps 的开发模式，轻松的做到蓝绿部署，并且支持阿里云云效。

ROS 控制台及操作演示

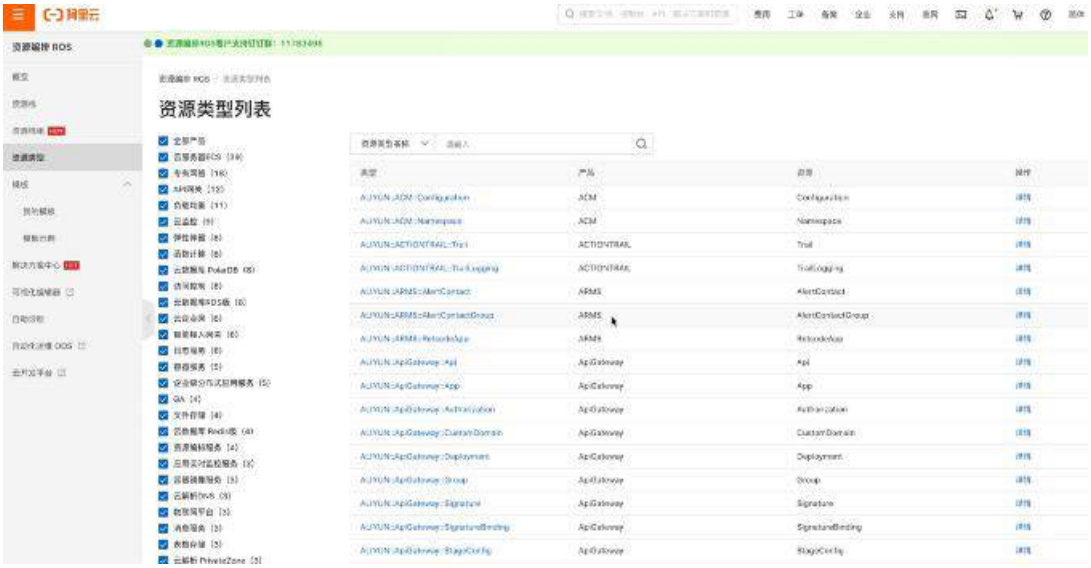
下图是资源编排 ROS 主页 https://rosnext.console.aliyun.com/，上方对 ROS 产品进行了简单的介绍；下方是常见的部署架构作为示例模版。



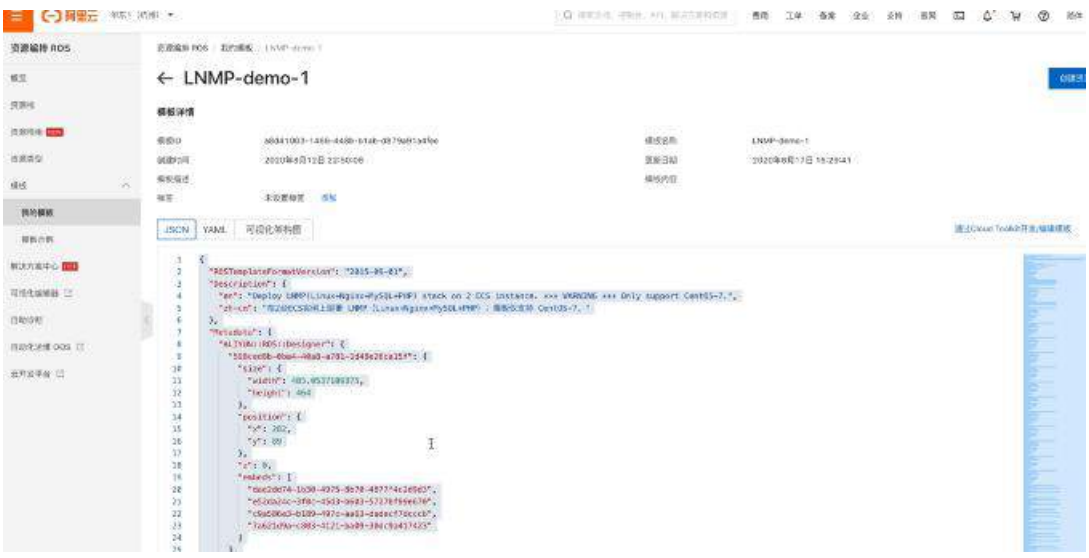
左侧菜单栏中有“我的模版”和“模版示例”，其中我的模版是需要自定义的模版，模版示例中提供了大量常见的部署形式，如 Jenkins、Kafka 等。解决方案中心中是由阿里云解决方案架构师团队、最佳实践团队、各业务方团队和资源编排团队合作共建，将阿里云多年沉淀的最佳实践和针对各种场景的解决方案沉淀为资编排源模版，用户可以使用这些最佳实践模版使得云上部署更加安全高效。



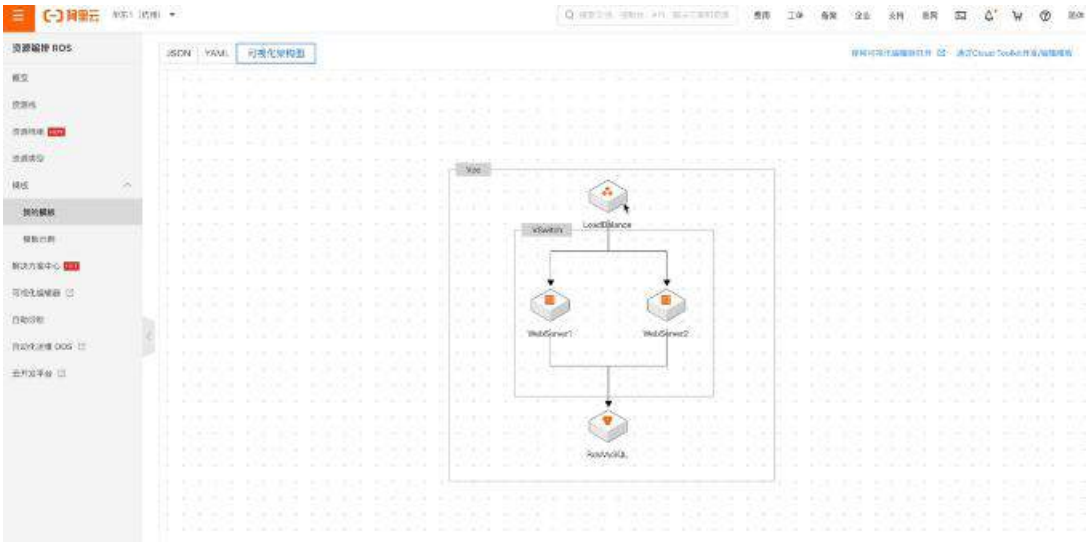
“资源类型”模块中展示了 ROS 支持的阿里云云产品。



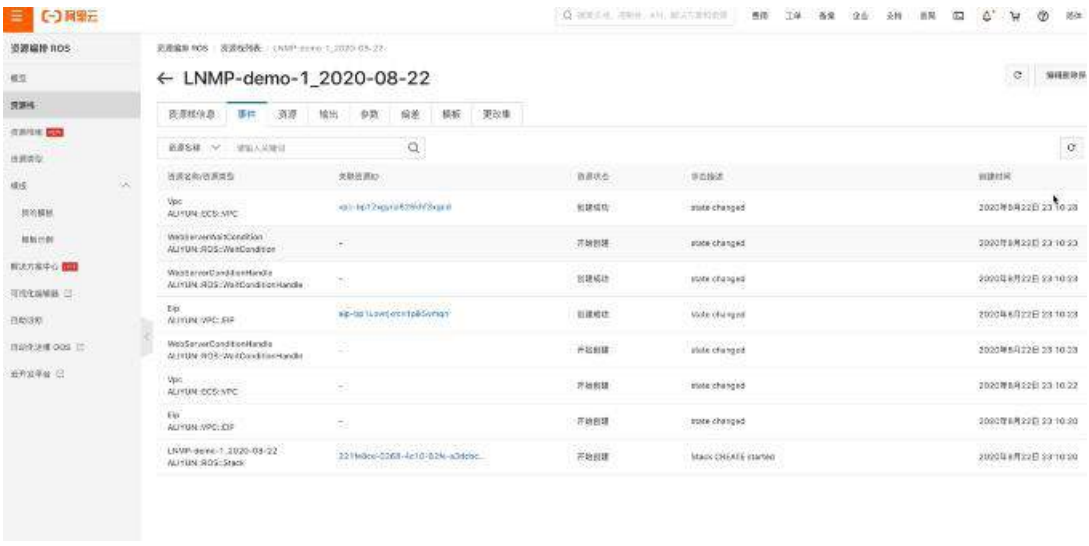
下面以构建我的模版 LNMP-deom-1 作为例子，模版以 JSON 格式表达，也支持 YAML 格式，最重要的是，还提供了可视化的架构图。



可以发现所有的资源都在 VPC 内部，包括关系型数据库 RDS 和 ECS 实例。图中两个 ECS 实例挂在负载均衡 LoadBalance 下面。



接着可以使用此模版创建资源栈，之后通过事件 tab 知道每一步创建步骤。在资源 tab 中看到所有被模版创建的资源，只需要点击资源 ID，就可以打开实例详情页面。在输出 tab 在有显示一个网站链接，可以发现此次网站部署成功。参数 tab 中提供了每次模版的参数。当用户手动修改一些资源，与模版出现不一致时，可以使用检查资源偏差查看不同点。



三、运维编排 OOS

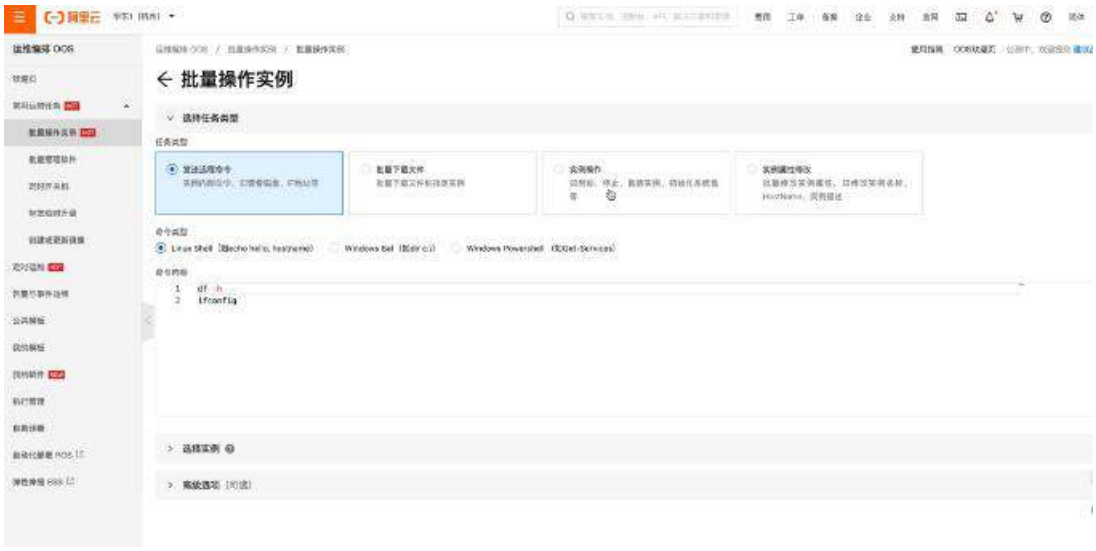
运维编排 OOS 的典型场景

运维编排 OOS 的典型场景同样分为四种：

- 首先是批量操作实例和执行远程命令，例如启动、停止等，相比于其它方式无需密码，无需登录，无需使用跳板机，且无需担心安全问题，运维编排使用了阿里云 RAM 进行控制。
- 第二种场景是定时运维，在固定的时间执行制定的命令，相当于云上 Cron 服务，并且免托管，分布式。
- 第三种场景是支持报警和事件驱动运维，当某个事件发生时自动触发告警任务。
- 第四是提供大量丰富的公共模版，阿里云总结了大量的典型运维场景，并将总结成果开源到了 Github 上，欢迎大家贡献优质模版，共同打造运维社区。

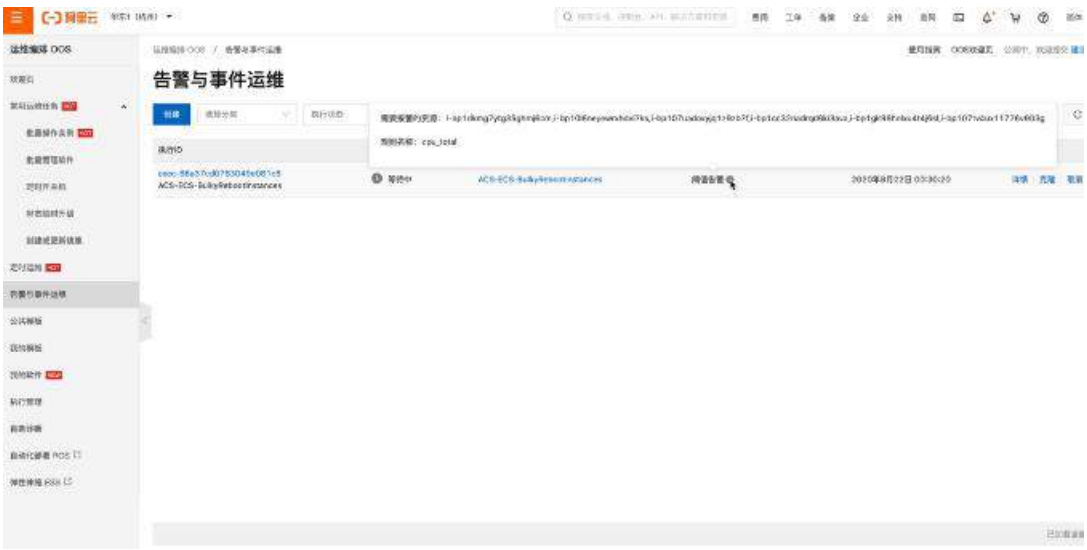
OOS 控制台及操作演示

下图是运维编排 OOS 主页 https://oos.console.aliyun.com/，左侧菜单栏中有批量操作实例模块，任务类型包含发送进程命令、批量下载文件、实例操作、实例属性修改等。批量管理软件模块中可以批量的给实例下载和安装常见的软件，在我的软件模块可以自行部署和卸载。

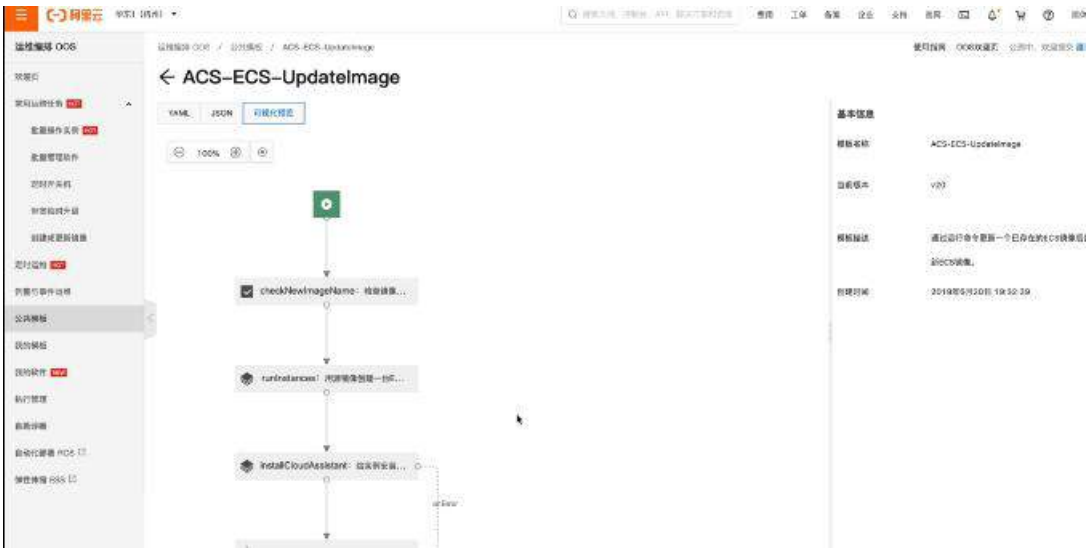


定时开关机模块中可以选择在指定的时间关闭、开启或重启实例。在包年包月的服务器情况下，客户需要在固定的时间升级临时宽带，等高峰过去后再下降，以达到节约成本的目的。在创建更新镜像模块中可以基于已有的实例进行更新，也可以基于已有的镜像创建实例，进一步更新，再创建新的镜像。

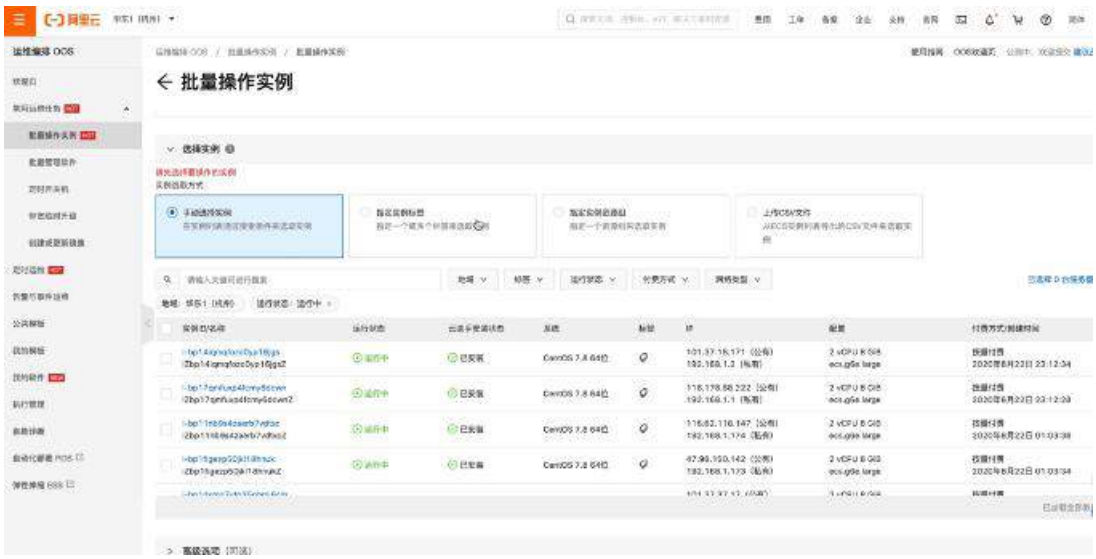
定时运维模块可以在固定的时间和固定的地域执行指定的任务。告警与事件运维模块中若控制台上显示当某个事件发生时自动触发模版，比如 CPU 使用率过高时重启实例操作。



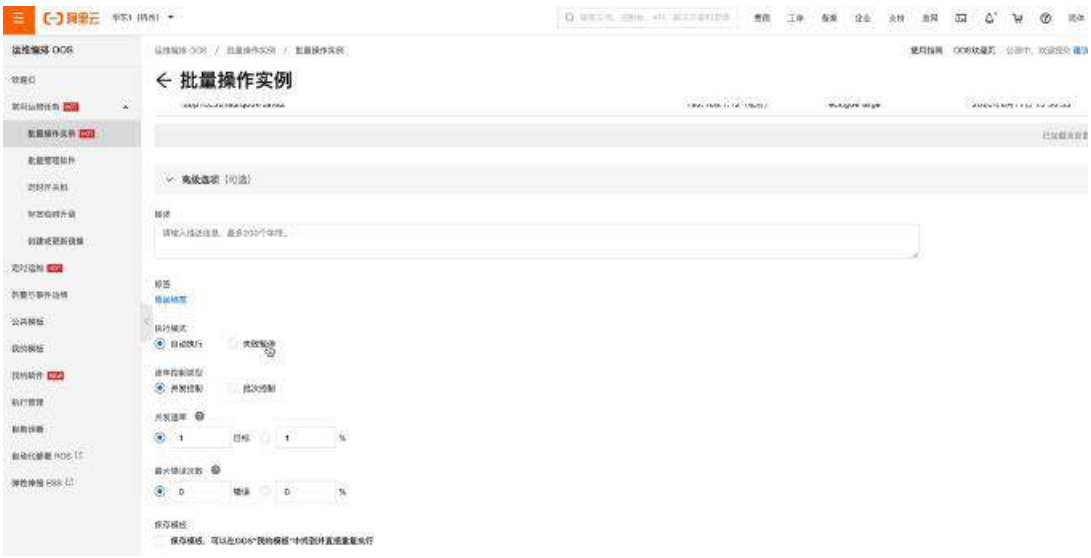
所有的模版都提供了可视化视图，提供了更加直观的展示方式，还提供了 YAML 和 JSON 两种格式的文本，方便用版本管理软件如 Git 进行管理。



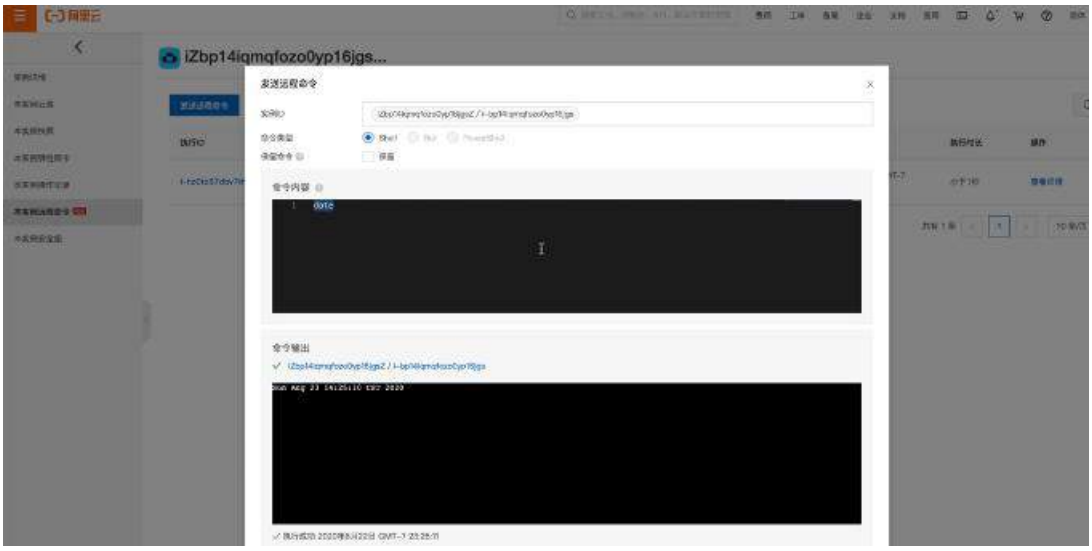
下图展示的是批量操作实例，发送远程命令，命令内容是发送输出命令。之后选择实例，可以手动选择，可以指定实例标题，也可以指定实例资源组，或者上传 csv 文件，从 ECS 实例表中导出 csv 文件来选择实例。



在高级选项部分可以配置执行模式，如出现错误时继续执行还是暂停实例，设置并发速率，允许的最大错误次数等。



此外还有更加快速的执行实例命令方式，在实例列表模块中选择具体的实例，进入实例详情页后会显示本实例远程命令，显示了历史的执行命令，同时可以发送新的远程命令。其次在实例列表中同时选择多台实例，选择更多，发送远程命令，这时多台实例就可以同时执行命令。

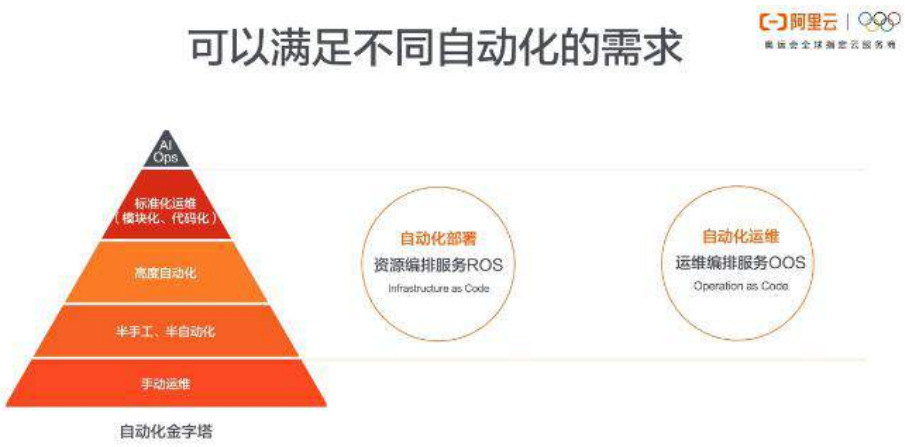


使用 ROS、OOS 的部分阿里云产品

下图中列出了支持 ROS、OOS 的常见阿里云产品，包括 ACS 容器服务、FC 函数计算、SLS 日志服务、SMC 服务迁移中心等等，这与产品本身的部署场景契合。

云产品需要支持多种地域，阿里云有 22 个地域，使用 ROS 和 OSS 可以最大提高部署和运维效率。阿里云对内部系统变更有非常严格的要求，需要提供信息完整的变更单、申请、审批、以及需要为变更过程中可能出现的问题提前准备脚本。因此 OSS 会预先提供变更模版和回滚模版，从而提供自动化运维程度，降低人工错误。

客户对自动化运维有不同的需要，从下图左侧可以分出运维的几个层次，从最底层的手动运维、到半手工，半自动化运维、再到高度自动化运维、标准化运维以及智能运维（AIOps）、大部分客户的需求集中在中间三层，大部分的公司处于半手工，半自动化运维，异或高度自动化的方式，少部分的公司更加激进的走到了更加标准化运维，享受到了更加 DevOps 的方式，阿里云自动化部署 ROS 和自动化运维 OOS 的主打场景可以满足这三个主要层次的自动化需求。



今天的分享到此结束，感兴趣的同学可持续关注云上自动化部署 ROS 和运维 OOS 产品动态。

2.3 ECS 云助手，实现云上运维自动化

摘要：本次内容由阿里云技术专家朱士松（锐奇)为大家介绍《ECS 云助手，实现云上运维自动化》。以往需要操作 ECS 实例内部的系统时，要先通过公网或跳板机或远程桌面登入实例，而后在实例内执行一些文件或命令操作。如果使用 ECS 云助手，不但免除公网环境的限制需求，而且能通过阿里云 OpenAPI 完全自动化地实现文件发送与执行命令，适用于在 ECS 实例内部署与更新应用、监控系统或应用的运行状态、以及批量操作多个实例内部系统的这些场景。本篇内容适合于 ECS 系统运维人员。

演讲嘉宾简介：朱士松（锐奇)，阿里云技术专家，2016 年加入阿里云，先后开发了 ECS 售卖约束系统、阿里云区块链服务，目前负责 ECS 云助手。

本次分享主要围绕以下四个方面：

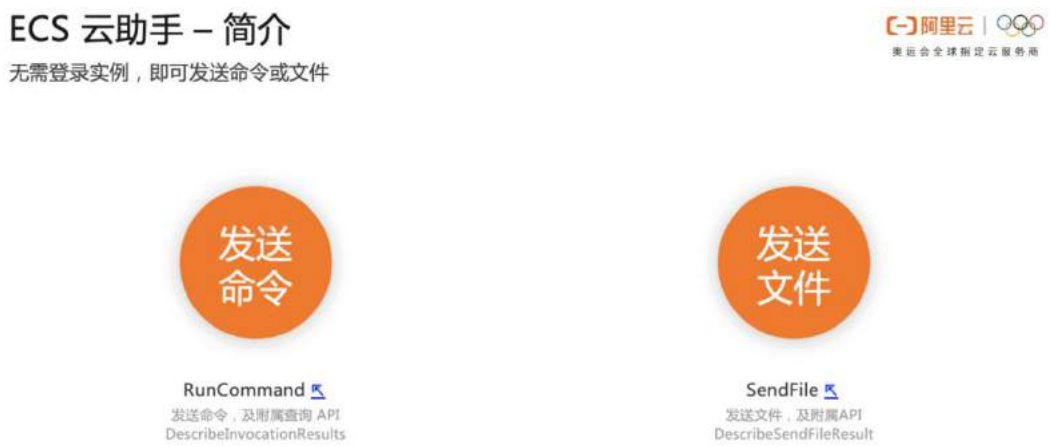
一、云助手-功能简介
二、使用说明与演示
三、远程操作方式比较
四、云助手的适用场景

一、ECS 云助手简介

云助手是阿里云 ECS 提供的一种自动化的远程操作方式，在阿里云官方的系统镜像中几乎都包含有云助手。

云助手的使用方法比较简单，只有两项主要功能：

1. 向指定的实例发送命令，对应 API ecs:RunCommand
2. 向指定的实例发送文件，对应 API ecs:SendFile



通过 API 使用云助手

发送命令（RunCommand）

发送命令的功能，由 ECS:RunCommand API 承载，API 的主要参数如下：

```
aliyun ecs RunCommand \
  --RegionId="cn-shenzhen" \
  --InstanceId.1="i-wz9g75dkmfp0ofsplnlr" \
  --InstanceId.2="i-wz9g75dkmfp0ofsplnls" \
  --Type="RunShellScript" \
  --CommandContent="yum install -y git" \
  --Timeout=60
```

- 参数："RegionId" – 指的是目标 ECS 实例所在的地域
- 参数："InstanceId – 可以指定该地域下的一个或多个 ECS 实例
- 参数："Type" – 指的是脚本类型，目前支持三种：分别是
  - Linux 上支持执行 Shell 脚本，类型值 RunShellScript
  - Windows 上支持的 Batch 与 PowerShell 脚本，类型值 RunBatScript 与 RunPowerShellScript
- 参数："CommandContent" – 指的是脚本内容，比如当前示例通过 yum 安装 git 客户端
- 参数：Timeout – 批的时脚本执行超时时间，默认 60 秒；
- 关于 ecs:RunCommand 的详细 API 文档：

  https://help.aliyun.com/document_detail/141751.html.

- 推荐使用 aliyun 命令行工具 (https://help.aliyun.com/document_detail/110244.html) 执行阿里云 API。

调用了 RunCommand 之后，将会创建一个任务，并返回 InvokeId 值；之后可使用 DescribeInvocationResults 轮论这次任务的执行进度与结果；关于 DescribeInvocationResults 说明，请参见文档：https://help.aliyun.com/document_detail/64845.html

发送文件（SendFile）

发送文件的功能，由 SendFile API 承载，API 的主要参数如下：

```
aliyun ecs SendFile \
  --RegionId="cn-shenzhen" \
  --InstanceId.1="i-wz9g75dkmfp0ofsplnlr" \
  --InstanceId.2="i-wz9g75dkmfp0ofsplnls" \
  --TargetDir="/root/.ssh/" \
  --Name="authorized_keys" \
  --Content="ssh-rsa AAAA...."
```

- 其他参数：RegionId & InstanceId - 指定实例所在地域和实例 ID 列表
- 参数：TargetDir 与 Name - 分别指定文件在实例上的目录名与文件名
- 参数：Content - 指定文件的内容

关于 ecs:SendFile 的详细 API 文档：https://help.aliyun.com/document_detail/184118.html

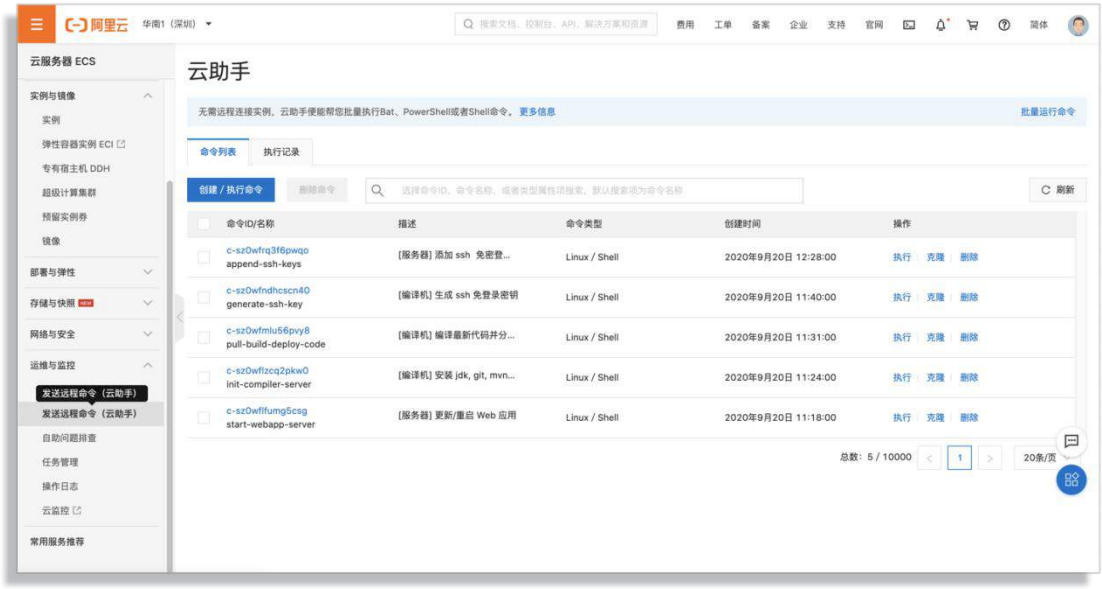
调用了 SendFile 之后，也将会创建一个任务，并返回 InvokeId ；之后可使用 DescribeSendFileResults 轮论这次任务的执行进度与结果；

了解了以上两对 API，也就学会云助手的主要用法，那么就可以在脚本或代码中使用。

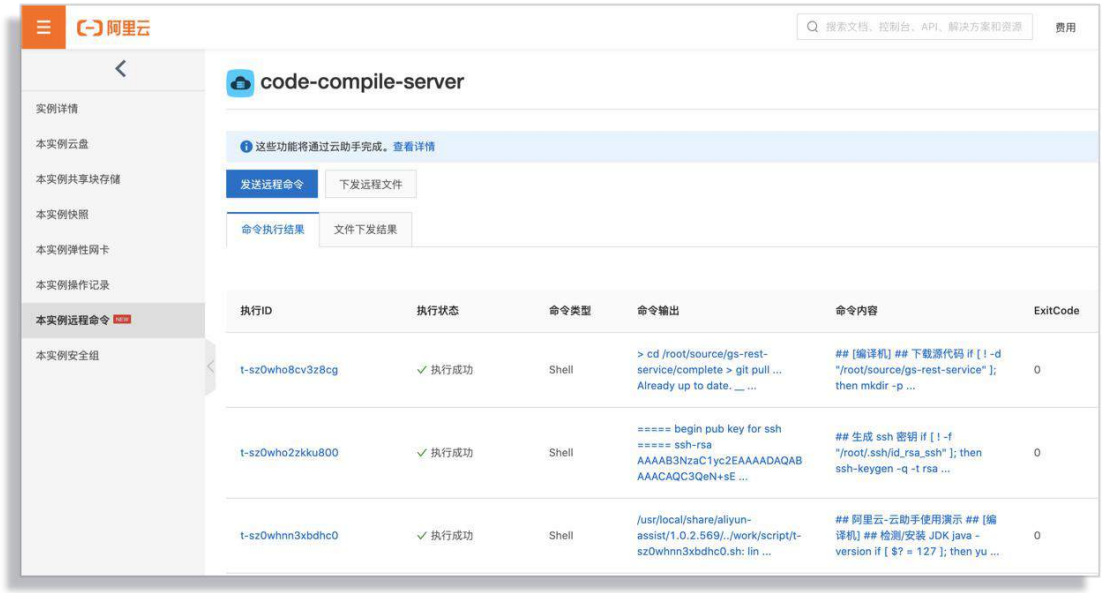
通过控制台使用云助手

如果想通过阿里云控制台使用云助手，可以从这个两个位置找到“云助手”控制台：

一是 Ecs 控制台 https://ecs.console.aliyun.com/ 左侧的“运维与监控” 服务列表这里



二是 Ecs “实例详情”页面的“本实例远程命令”



二、使用实践示例

接下来，咱们通过一个实践，体验一下云助手的使用，实现一项常规的任务：将一个应用的代码自动部署到 ECS 服务器上，并且当代码更新时立即更新服务。

准备的资源

将要使用的资源如下：

1. 应用代码：这里使用示例代码是 spring-boot 的官方示例程序 gs-rest-service 并稍有修改。
   代码库地址：https://github.com/treesong/gs-rest-service
2. 代码编译机：一台 ECS，将安装 JDK + git + maven，负责下载和编译代码并打包；
3. Web 服务器：三台 ECS，负责将安装 JRE，部署代码包，并启动 Web 应用；
4. 负载均衡：一个 SLB，后端挂载这三台 Web 应用服务器，并对外提供服务；

操作的步骤

操作步骤如下：

1）创建虚拟专有网络(VPC)与虚拟交换机

如果您已经拥有虚拟专有网络与虚拟交换机，可以跳过本步骤

转到 VPC 控制台 (https://vpc.console.aliyun.com/vpc/cn-shenzhen/vpcs/new)，创建 VPC 实例及交换机：

新建 VPC 的设置项：

- VPC 名称：vpc-for-assist-demo
  - IPv4 网段：192.168.0.0/16
- 交换机名称：vsw-for-axt-demo-d
  - IPv4 网络：192.168.0.0/29
- 其他选项：可使用默认值，或按需自由设置

注：在实际应用中，您也可以根据网络规划需要，选择使用其他网段

2）创建一台 ECS 实例，用作编译服务器

新建 ECS 的设置项：

- 数量：1 台
- 镜像： CentOS，或其他 Linux 类型
- 网络：专有网络，并选择上一步创建的专有网络(vpc-for-assist-demo)与交换机(vsw-for-axt-demo-d)
- 公网 IP：不需要
- 实例名称：code-compile-server
- 主机名称：code-compile-server
- 其他选项：可使用默认值，或按需自由设置

确认订单，以创建 ECS 实例。

3）另创建两台 ECS 实例，用作应用服务器

在现有的 VPC 实例 vpc-for-assist-demo 下，另创建一个新虚拟交换机实例，设置项：

- 交换机名称：vsw-for-axt-demo-e
  - IPv4 网络：192.168.1.0/29
- 其他选项：可使用默认值，或按需自由设置

在新的虚拟交换机实例（vsw-for-axt-demo-e）下，创建 3 台 ECS 作为应用服务器，设置项：

- 数量：2 台
- 镜像：CentOS，或其他 Linux 类型
- 网络：专有网络，并选择上一步创建的专有网络实例与交互机(vsw-for-axt-demo-e)
- 公网 IP：不需要
- 实例名称：webapp-server-
- 主机名称：webapp-server-
- 有序后辍：是，为 实例名称 和 主机名 添加有序后缀
- 其他选项：可使用默认值，或按需自由设置

确认订单，以创建 ECS 实例。

4）配置 VPC 网络以允许 ECS 实例出公网

因需要从公网上下载源代码等，因此需要允许 ECS 实例 code-compile-server 可访问公网，需要：

- 配置 vpc-for-assist-demo 的 NAT 网关
- 为该 NAT 网关创建 SNAT 条目，为 vsw-for-axt-demo-d 绑定一个公网 IP (略过该步骤的详细过程，如需详细步骤，请参考 VPC 的使用资料)

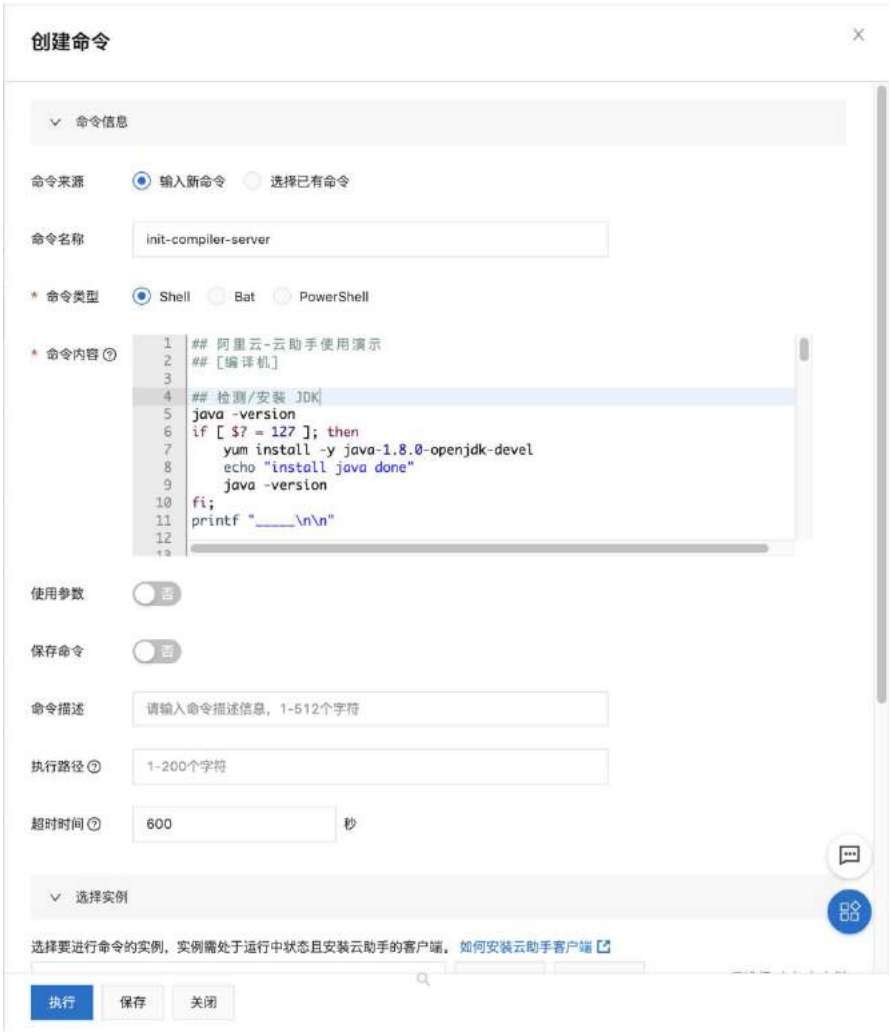
5）编译机的初始化

转到云助手的控制台（https://ecs.console.aliyun.com/#/cloudAssistant/region/cn-shenzhen）。

编译机的初始化

创建以下命令，以初始化编译机(code-compile-server)，作用

- 安装 JDK
- 安装 GIT
- 下载 Maven 并配置 settings.xml
- 生成访问 github.com 的密钥对



- 命令名称：1-init-compiler-server.sh
- 命令类型：Shell
- 超时时间：600(秒)
- 命令内容：(如下，或从 https://github.com/treesong/aliyun-assist-demo 获得)

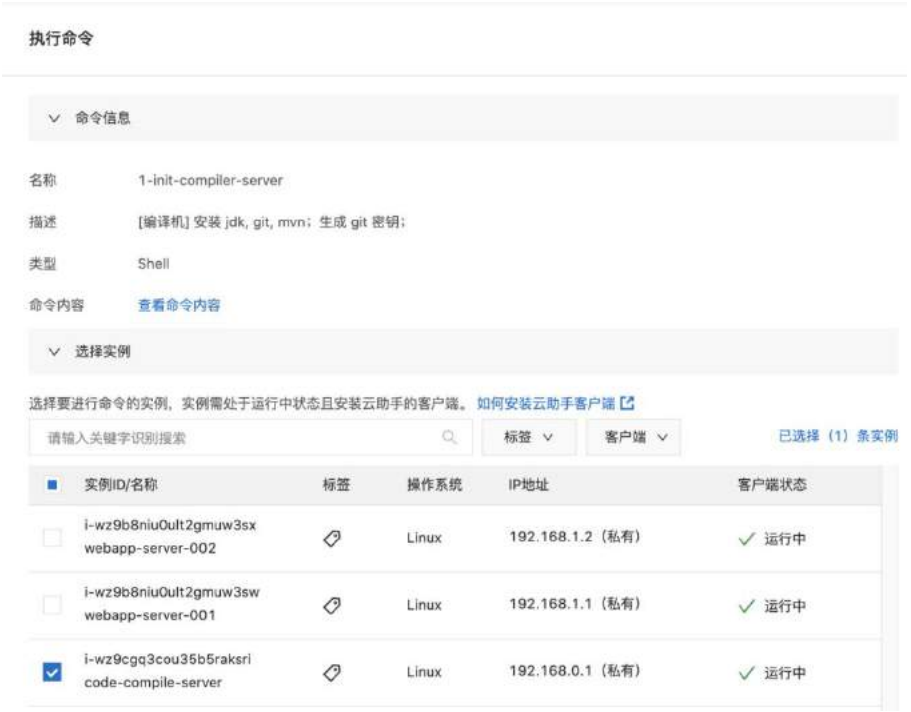
```
## 阿里云-云助手使用演示
## [编译机]

## 检测/安装 JDK
java -version
if [ $? = 127 ]; then
    yum install -y java-1.8.0-openjdk-devel
    echo "install java done"
    java -version
fi;
```

```
    printf "____\n\n"

    ## 检测/安装 GIT
    git --version
    if [ $? = 127 ]; then
        yum install -y git
        echo "install git done"
    fi;
    printf "____\n\n"

    ## 检测/下载 Maven
    cd /root
    if [ ! -d "/root/apache-maven" ]; then
        wget -q
https://mirrors.bfsu.edu.cn/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.zip
        unzip -q -o -d ./ ./apache-maven-3.6.3-bin.zip
        ln -s /root/apache-maven-3.6.3/ /root/apache-maven
    fi;

    export PATH=/root/apache-maven/bin/:$PATH
    mvn --version
    printf "____\n\n"

    ## 更新 maven settings.xml 配置
    echo '
    <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

        <localRepository>/root/.m2/repository</localRepository>

        <mirrors>
            <mirror>
                <id>aliyun</id>
                <name>aliyun Maven</name>
                <mirrorOf>central</mirrorOf>
                <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
            </mirror>
            <mirror>
                <id>CN</id>
                <name>OSChina Central</name>
                <url>http://maven.oschina.net/content/groups/public/</url>
```

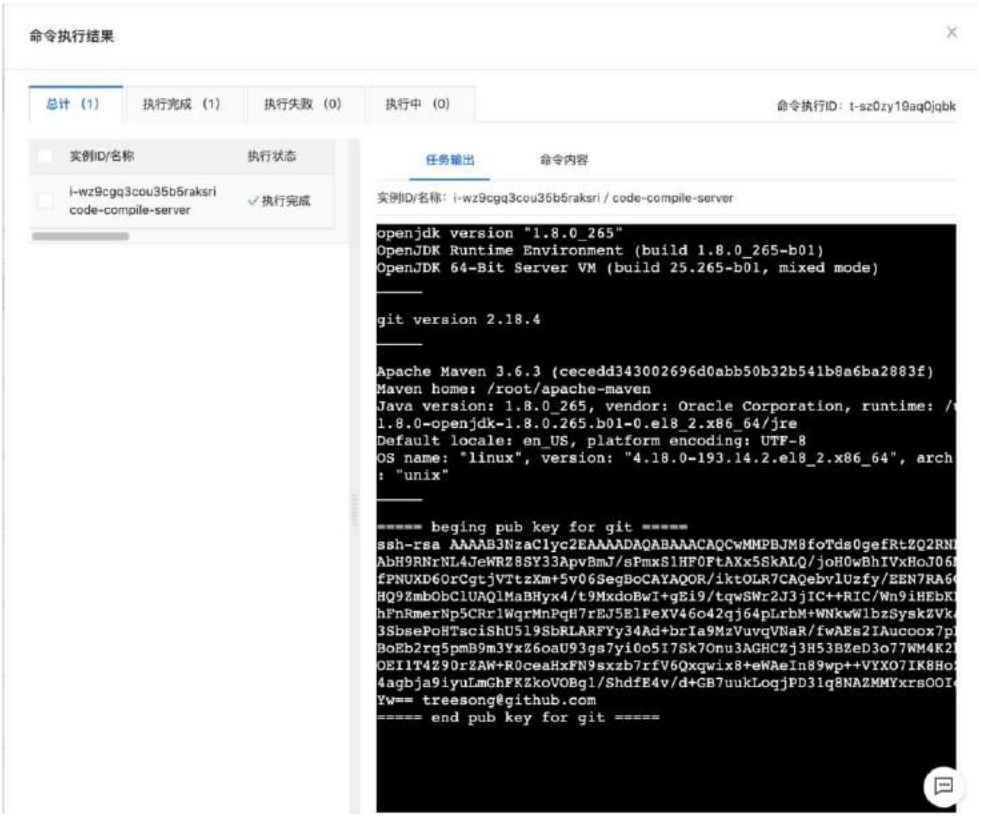
```
                <mirrorOf>central</mirrorOf>
            </mirror>
        </mirrors>

        <profiles></profiles>

</settings>
' > /root/apache-maven/conf/settings.xml


## 生成 git 密钥
if [ ! -f "/root/.ssh/id_rsa_git" ]; then
        ssh-keygen -q -t rsa -b 4096 \
          -C "treesong@github.com" \
          -f /root/.ssh/id_rsa_git \
          -N ""
fi;

## 配置自动选择 git 密钥
echo "host github.com
        HostName github.com
        StrictHostKeyChecking no
        User treesong
        IdentityFile /root/.ssh/id_rsa_git
" > /root/.ssh/config

echo "===== beging pub key for git ====="
cat /root/.ssh/id_rsa_git.pub
echo "===== end pub key for git ====="
```

点击“保存”按钮，保存该条命令。

如上图，已经创建了该条命令，继续“执行”，并选择 ECS 实例 code-compile-server


执行命令
命令信息
名称 1-init-compiler-server
描述 [编译机] 安装 jdk, git, mvn；生成 git 密钥；
类型 Shell
命令内容 查看命令内容
选择实例
选择要进行命令的实例，实例需处于运行中状态且安装云助手的客户端。如何安装云助手客户端
请输入关键字识别搜索
标签
客户端
已选择（1）条实例
实例ID/名称 标签 操作系统 IP地址 客户端状态
i-wz9b8niu0ult2gmuw3sx webapp-server-002 Linux 192.168.1.2（私有） 运行中
i-wz9b8niu0ult2gmuw3sw webapp-server-001 Linux 192.168.1.1（私有） 运行中
i-wz9cgq3cou35b5raksri code-compile-server Linux 192.168.0.1（私有） 运行中

等待执行完成。


命令执行结果
总计（1）
执行完成（1）
执行失败（0）
执行中（0）
命令执行ID：t-sz0zy19aq0jqbk
实例ID/名称
执行状态
i-wz9cgq3cou35b5raksri code-compile-server
执行完成
任务输出
命令内容
实例ID/名称：i-wz9cgq3cou35b5raksri / code-compile-server
openjdk version "1.8.0_265"
OpenJDK Runtime Environment (build 1.8.0_265-b01)
OpenJDK 64-Bit Server VM (build 25.265-b01, mixed mode)
git version 2.18.4
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /root/apache-maven
Java version: 1.8.0_265, vendor: Oracle Corporation, runtime:
1.8.0-openjdk-1.8.0.265.b01-0.el8_2.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.18.0-193.14.2.el8_2.x86_64", arch
: "unix"
===== beging pub key for git =====
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCwMMPBJM8foTds0gefRtZQ2RN
AbH9RNrNL4JeWRZ8SY33ApvBmJ/sPmxSlHF0FtAXx5SkALQ/joH0wBhIVxHoJ06
fPNUXD6OrCgtjVTtzXm+5v06SegBoCAYAQOR/iktOLR7CAQebvlUzfy/EEN7RA6
HQ9ZmbObClUAQlMaBHyx4/t9MxdoBwI+gEi9/tqwSWr2J3jIC++RIC/Wn9iHEbK
hFnRmerNp5CRr1WqrMnPqH7rEJ5ElPeXV46o42qj64pLrbM+WNkwW1bzSyskZVk
3SbsePoHTsciShU519SbRLARFYy34Ad+brIa9MzVuvqVNaR/fwAEs2IAucoox7p
BoEb2rq5pmB9m3YxZ6oaU93gs7yi0o5I7Sk7Onu3AGHC2j3H53BZeD3o77WM4K2
OEI1T4Z90rZAW+R0ceaHxFN9sxzb7rfV6Qxgwix8+eWAeIn89wp++VYXO7IK8Ho
4agbja9iyuLmGhFKZkoVOBg1/ShdfE4v/d+GB7uukLoqjPD31q8NAZMMYxrsOOI
Yw== treesong@github.com
===== end pub key for git =====

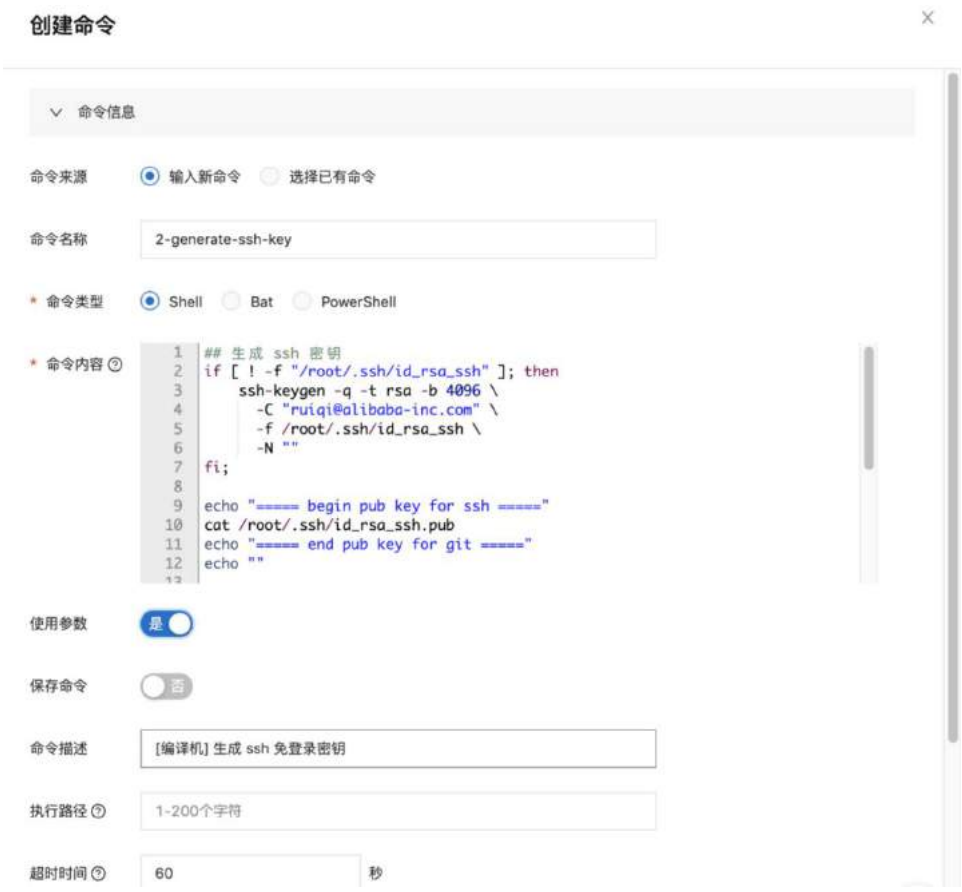
命令执行所生成的 git 公钥，请用来添加在您的 git 帐号中，以允许从主机 code-compile-server 内从 github 上拉取应用代码 https://github.com/treesong/aliyun-assist-demo

生成 ssh 密钥对，以免密编译机分发应用包到应用服务器

注：您也可以将应用包上传到 OOS 中，并在应用服务器内下载应用包

创建以下命令，以初始化编译机(code-compile-server)，作用：

- 生成一个 ssh 密钥对，以用于 scp 应用包到应用服务器



- 命令名称：2-generate-ssh-key.sh
- 命令类型：Shell
- 命令内容：(如下，或从 https://github.com/treesong/aliyun-assist-demo 获得)
- 使用参数：是

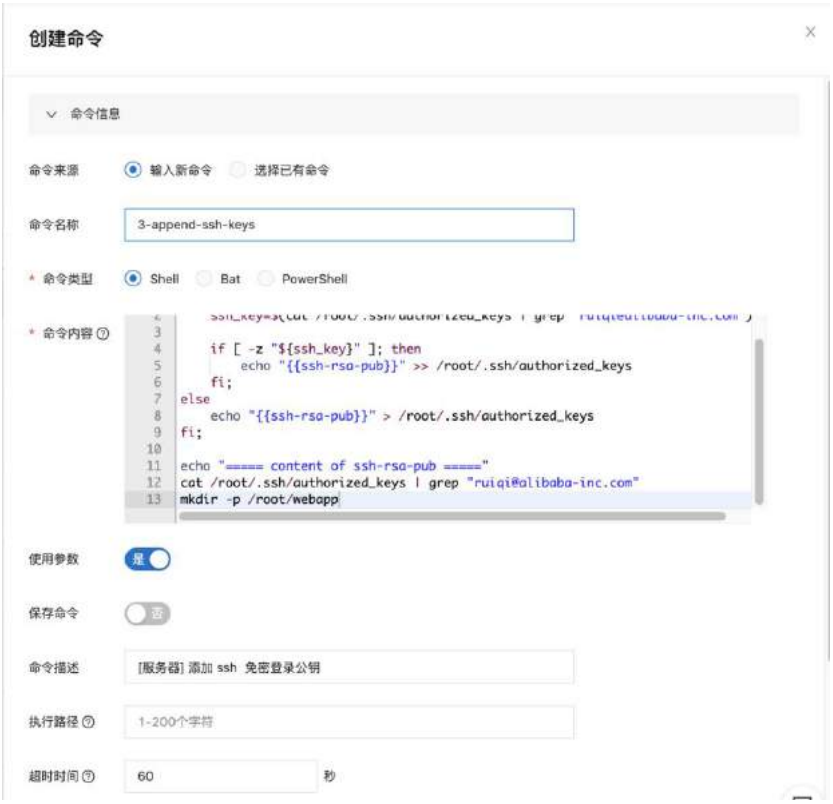
```
## 生成 ssh 密钥
if [ ! -f "/root/.ssh/id_rsa_ssh" ]; then
      ssh-keygen -q -t rsa -b 4096 \
         -C "ruiqi@alibaba-inc.com" \
         -f /root/.ssh/id_rsa_ssh \
         -N ""
fi;

echo "===== begin pub key for ssh ====="
cat /root/.ssh/id_rsa_ssh.pub
echo "===== end pub key for git ====="
echo ""
done;
```

选择实例 code-compile-server 并执行该脚本，执行完成后将显示新生成的 id_rsa_ssh.pub 文件内容。

6） 应用服务器的初始化

添加 ssh 免密登录公钥到应用服务器（webapp-server-*）

- 添加 ssh 免密登录公钥（公钥内容来自上一步生成的 id_rsa_ssh.pub 文件内容）



- 命令名称：3-append-ssh-keys
- 命令类型：shell
- 命令内容：(如下，或从 https://github.com/treesong/aliyun-assist-demo 获得)
- 使用参数：是

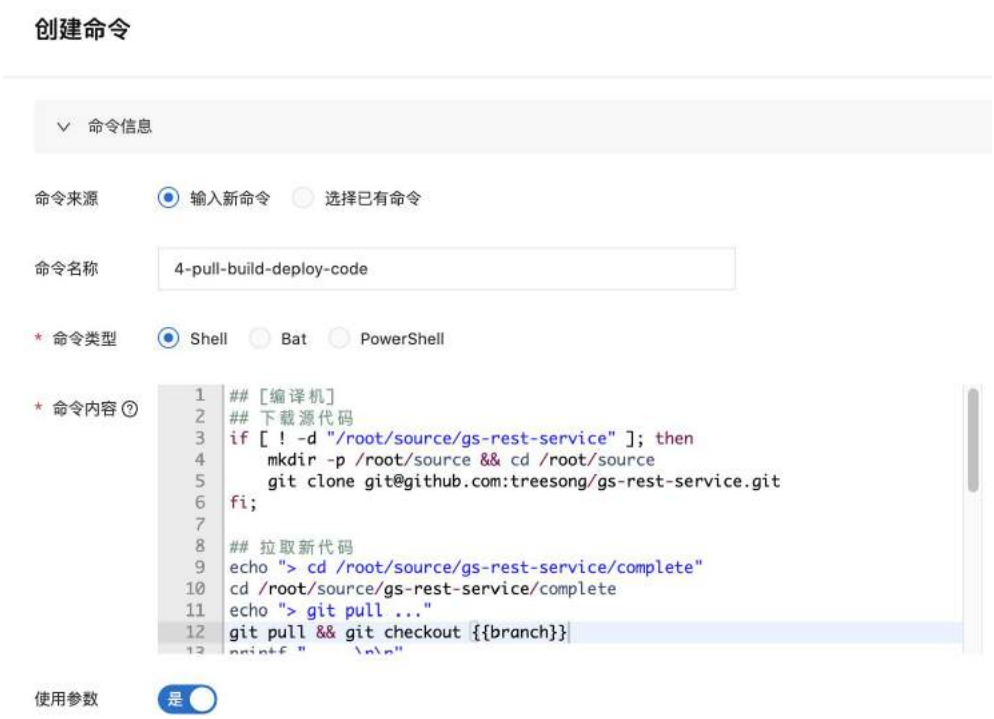
```
if [ -f "/root/.ssh/authorized_keys" ]; then
    ssh_key=$(cat /root/.ssh/authorized_keys | grep "ruiqi@alibaba-inc.com")

    if [ -z "${ssh_key}" ]; then
        echo "{{ssh-rsa-pub}}" >> /root/.ssh/authorized_keys
    fi;
else
    echo "{{ssh-rsa-pub}}" > /root/.ssh/authorized_keys
fi;

echo "===== content of ssh-rsa-pub ====="
cat /root/.ssh/authorized_keys | grep "ruiqi@alibaba-inc.com"
mkdir -p /root/webapp
```

执行命令 3-append-ssh-keys.sh，选项：

- 参数 ssh-ras-pub：内容填入 2-generate-ssh-key 生成与打印的 id_rsa_ssh.pub 文件内容
- 目标实例：选择全部的 webapp-server-* 实例

7）编译与分发代码

拉取最新的代码，编译打包，并分发到应用服务器



创建命令选项：

- 命令名称：4-pull-build-deploy-code
- 命令类型：Shell
- 命令内容：(如下，或从 https://github.com/treesong/aliyun-assist-demo 获得)
- 使用参数：是

```
## [编译机]
## 下载源代码
if [ ! -d "/root/source/gs-rest-service" ]; then
    mkdir -p /root/source && cd /root/source
    git clone git@github.com:treesong/gs-rest-service.git
fi;

## 拉取新代码
echo "> cd /root/source/gs-rest-service/complete"
cd /root/source/gs-rest-service/complete
echo "> git pull ..."
git pull && git checkout {{branch}}
```

```
    printf "____\n\n"

    ## 编译代码
    export PATH=/root/apache-maven/bin/:$PATH
    echo "> mvn clean package -Dmaven.test.skip=true"
    mvn clean package -Dmaven.test.skip=true
    printf "____\n\n"

    ## 分发代码
    if [ ! -z "{{vm-ip-list}}" ]; then
       for ip in {{vm-ip-list}}; do
          echo "> scp *.jar to $ip ..."
          scp -i /root/.ssh/id_rsa_ssh.pub ./target/rest-service-0.0.1-SNAPSHOT.jar
root@$ip:/root/webapp/
       done;
       echo "copy files done."
    else
       echo "copy files skip."
    fi;
```

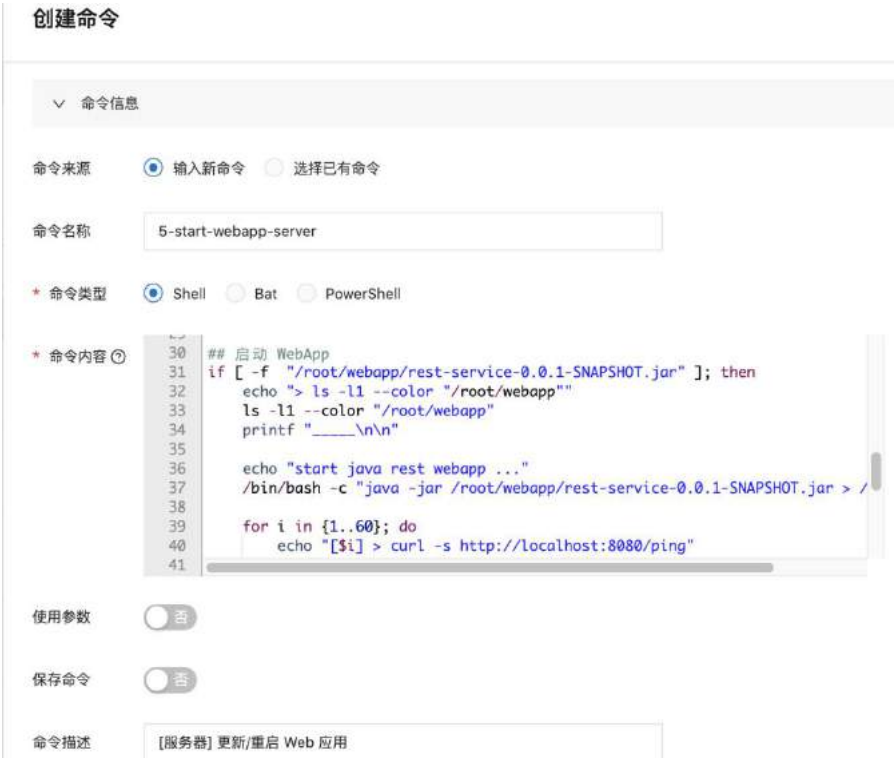
执行命令选项：

- 命令参数：
  - branch：等部署的代码分支
  - vm-ip-list：应用服务器 IP 列表
- 目标实例：代码服务器(code-compile-server)

8）启动/重启 Web 应用

在 webapp-server-* 上启动应用

- 检查与安装 JRE
- 停止 WebApp
- 启动 WebApp

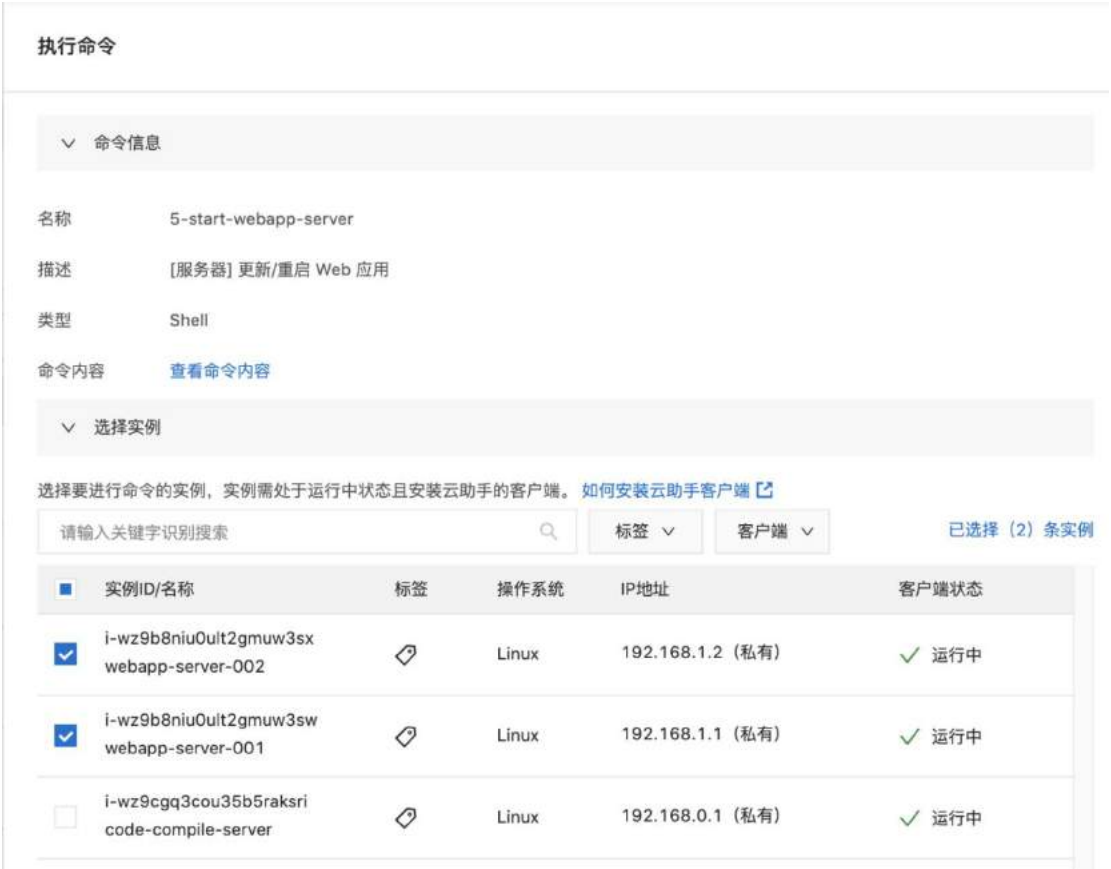


创建命令选项：

- 命令名称：5-start-webapp-server
- 命令类型：Shell
- 命令内容：(如下，或从 https://github.com/treesong/aliyun-assist-demo 获得)

```
## 阿里云-云助手使用演示
## [应用服务器]
## 安装 JRE/JDK

java -version
if [ $? = 127 ]; then
      echo "install jdk ..."
      yum install -y java-1.8.0-openjdk-devel
      echo "install jdk done"
      java -version
fi;
printf "____\n\n"

## 停止 WebApp
pid=$(jps -l | grep jar | cut -d' ' -f 1)
if [[ $pid =~ ^[0-9]+$ ]]; then
    jps -l | grep jar
    echo "stop java process $pid ..."
    kill -9 $pid
fi;

pid=$(jps -l | grep rest | cut -d' ' -f 1)
if [[ $pid =~ ^[0-9]+$ ]]; then
    jps -l | grep rest
    echo "stop java process $pid ..."
    kill -9 $pid
fi;
printf "____\n\n"

## 启动 WebApp
if [ -f   "/root/webapp/rest-service-0.0.1-SNAPSHOT.jar" ]; then
      echo "> ls -l1 --color "/root/webapp""
      ls -l1 --color "/root/webapp"
      printf "____\n\n"

      echo "start java rest webapp ..."
      /bin/bash -c "java -jar /root/webapp/rest-service-0.0.1-SNAPSHOT.jar > /dev/null &"

      for i in {1..60}; do
            echo "[$i] > curl -s http://localhost:8080/ping"
            msg=$(curl -s http://localhost:8080/ping)
            if [ "$msg" = "pong" ]; then
```

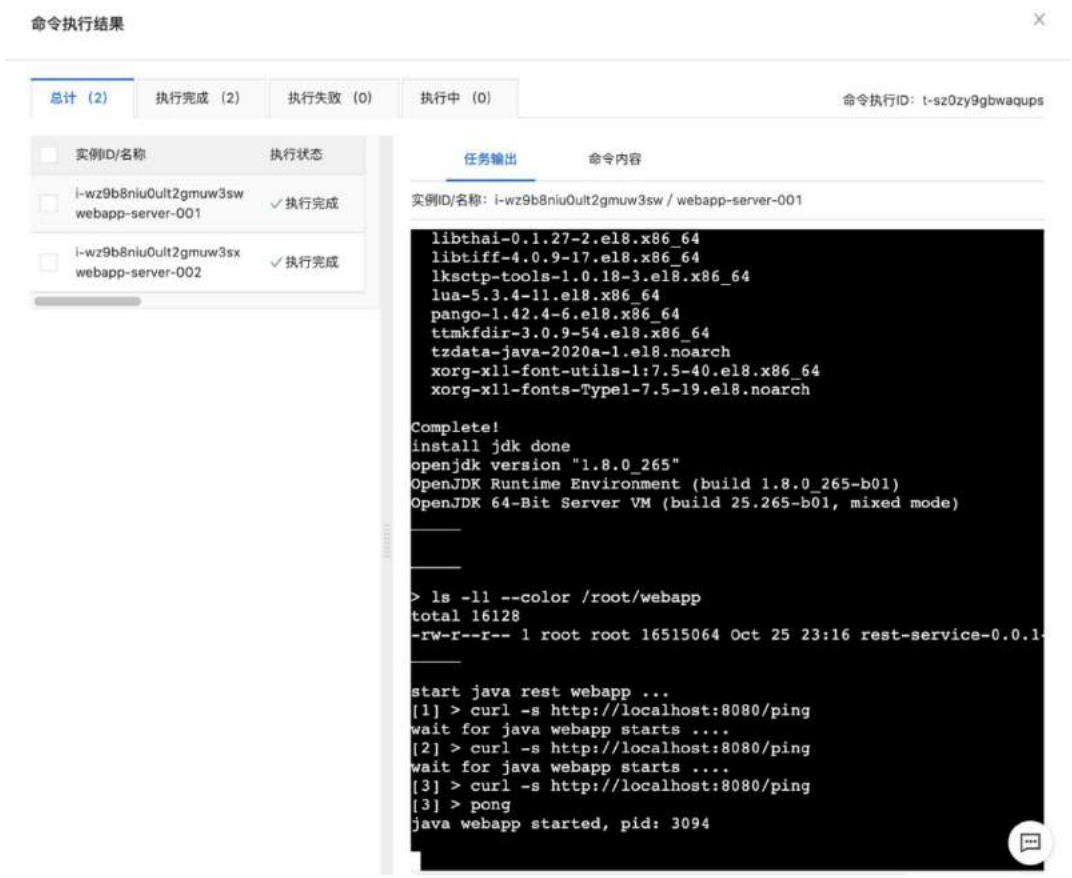
```
            echo "[$i] > $msg"
            pid=$(jps -l | grep rest | cut -d' ' -f 1)
            echo "java webapp started, pid: $pid"
            break;
        fi;

        echo "wait for java webapp starts ...."
        sleep 2
    done;
else
    echo "file not exists: /root/webapp/rest-service-0.0.1-SNAPSHOT.jar"
    exit 127
fi;
```



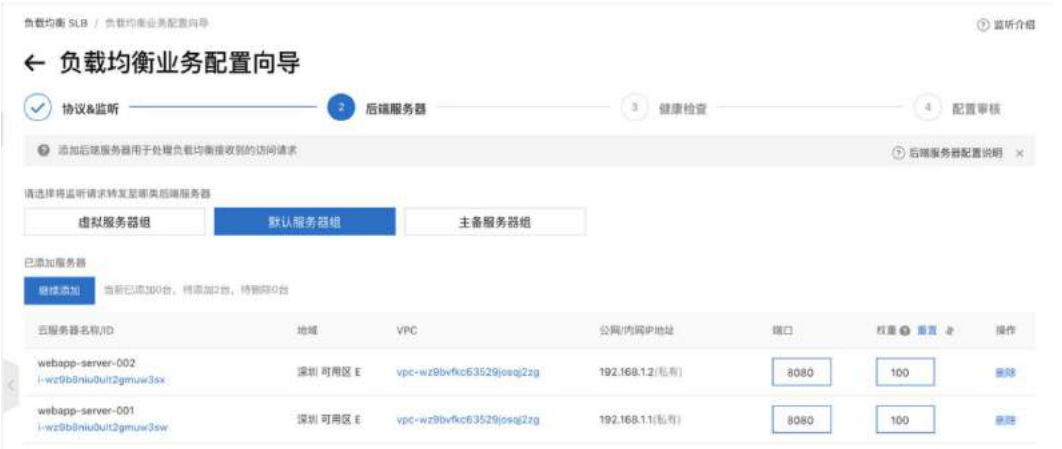
执行命令选项：

- 目标实例：选择全部的 webapp-server-*

执行命令，并等待执行完成



9）挂载应用服务器到 SLB，以供互联网访问



操作小结

通过以上的演示，大家可以看到，全程不需要连接进入实例系统内部；通过使用云助手发送脚本命令，就可以完成实例内部的变更运维操作；

因此，你可以通过阿里云 OpenAPI 自动化的实现以上操作，并且当代码有更新时，自动触发该系列操作，实现自动更新应用。可以访问 https://github.com/treesong/aliyun-assist-demo 获得更多的自动化脚本，或使用阿里云提供的相关应用部署服务。

三、远程操作方式比较

相比于其他常用登入系统的方式，例如 Linux SSH 或 Windows 的 RemteDesktop ，云助手的以下多个方面的个方面优势：

1）免公网流量：

云助手的功能实现，是通过控制系统来完成。所以不需要让实例暴露在公网上，即有利于实例的安全，也节省了公网流量费用。

2）免登入系统：

登入系统需要有系统的帐号用户与密码，或者私钥；而密码与私钥的使用与管理上有许多不便。云助手一切操作使用都基于阿里云帐号 AK，有云上强大的帐号/AK 管理体系。

3）有权限控制：

通过 RAM 角色与权限的管理，所以做到严格的控制子帐号是否执行特定操作的能力，可以事前防止不被授权的操作。

4）有操作审计：

同样，云上的 API 操作都有 ActionTrail 记录，可以被事后审计。

5）支持自动化：

这也是云助手最显著的优势，有了 API 就可以很容易的跟其他系统集成，以实现云上运维的

四、云助手的适用场景

通过上的介绍与演示，咱们可以体会到，借助云助手，能够自动化的处理 ECS 实例创建后的多个使用环节，例如：

- 系统与应用的安装配置
- 服务程序的部署与更新
- 系统的监控与数据采集
- 系统的问题诊断与修复

阿里云也有在这些方面上继续丰富的服务，给大家的工作来带来更多的便利。

结束&感谢

感谢大家的收看，今天的分享就到这里，欢迎持续关注阿里云-玩转 ECS 系列视频/文章。

2.4 ECS 自助服务之智能诊断和自动化修复

摘要：自助服务水平的高低是云厂商的核心竞争力，阿里云经过过去几年的积累，已经有了非常高效的自助服务能力。今天就将这些能力透露给最终用户。本次分享由阿里云高级技术专家滕圣波（云普）为大家介绍 ECS 自助服务之智能诊断与智能修复，解密背后的 AI 与实现细节，剧透 ECS 自助服务的未来。

演讲嘉宾简介：滕圣波（云普），阿里云高级技术专家，2018 年 5 月加入阿里云，作为架构师搭建了 ECS 的事件体系，同时也是阿里云的官方自动化运维平台-运维编排服务的主架构师之一，目前负责 ECS 智能自治服务、云桌面等领域。在加入阿里云之前，是 VMware 中国研发中心终端用户计算部门的架构师，拥有北京邮电大学计算机专业的硕士和学士学位。

本次分享主要围绕以下四个方面：

一、ECS 自助服务概要
二、智能诊断
三、自动化修复
四、自助服务背后的 AI 与数据能力

自助服务水平的高低是云厂商的核心竞争力，阿里云经过过去几年的积累，已经有了非常高效的自助服务能力。今天就将这些能力透露给最终用户。本次分享由阿里云高级技术专家滕圣波（云普）为大家介绍 ECS 自助服务，解析 ECS 自助服务主要包含哪些方面的自助服务，并从诊断和修复两个方面为大家解密自助服务的技术实现细节，最后给大家介绍冰山之下阿里云的 AI 及数据能力，剧透 ECS 自助服务的未来。

一、ECS 自助服务概要

1. 人工客服

人工客服流程

自助客服或者智能客户越来越普遍，其实从线下银行的 ATM 开始，用户就能体会到自助服务带来的便捷与省时。与自助服务相对的是人工客服的服务。在阐述自助服务之前，下面先谈谈与之相对的人工客服服务。

阿里云人工客服流程如下图所示：

首先用户遇到了一个问题，便向阿里云控制台中的智能在线模块的智能机器人诉说自己的诉求，如果智能机器人判断是一个问题，则自动开工单，用户也可以自己在线开工单描述自己的诉求。

所有工单到一线客服端，一线客服会与用户反复的确认具体的诉求，比如是什么商品，订单号是多少，具体什么时间，影响用户的影响面是多少。

这些问题弄清楚之后如果一线客服可以自己解决则直接指导用户解决问题。如果不能，则将问题向上反馈到二线技术支持端。一线客服是阿里云小二，二线技术客服是阿里云自营的技术专家，技术专家与用户沟通与处理疑难杂症。

如果二线技术专家依然解决不了问题，如阿里云本身的服务缺陷，或者用户受限制的特权类应用，则上升到三线工程师或产品专家手中，他们是阿里云研发团队内部最后台的技术人员和产品人员。真正需要修复代码或权限的问题才由三线工程师解决。

整个问题处理链条非常长，涉及到很多部门和人员。而针对大客户会有专门的企业服务钉钉群，相较工单能够得到更及时的响应。

阿里云对外公开的业务不可用工单响应时间小于 40 分钟，这仅仅表示一线客服响应的时间。真正问题解决周期大概是 1 至 24 小时。即使是企业客服钉钉群，依然不能保证分钟级的解决时间。

人工客服主要有几个痛点：

1）首先是需要多次反复的沟通流程。

因为一线客服没有权限查询用户具体的查询或操作记录，所以不得不与用户进行反复的沟通，需要询问用户的操作时间，操作的 request ID，从而在内部工单系统中补充这些信息，方便后面的二线及三线客服排查问题。这就导致沟通成本高，而且用户也未必放心将这些隐私信息交给客服。

2）其次，客服问题处理时间较长。

这是因为但凡需要人解决的问题，就无法很快的处理和解决。人需要读完所有的日志，还需要进行逻辑判断和分析。在问题复杂，数据量大，人工处理时需要时间就会较长。一线客服可以处理的问题或许需要半小时，二线客服处理问题则需要 2-3 小时，如果需要三线客服来处理问题则要以天为单位来计算。

3）第三点，人工客服处理问题是通过内部接口处理的，用户会问客服做了什么操作，解决了问题，但目前并没有把所有操作透露给用户，导致用户质疑操作是否透明。

2. 自助服务

随后，阿里云提出了服务的升级方案，既开始提供自助服务。自助服务的理念是由用户自己借助 AI 的能力检测问题并修复问题。如下图中提供了自助工具，用户可以进行问题诊断，自助工具会告知用户问题的根因，进而用户借助自动修复工具，一键修复问题，解决问题时间缩短至在分钟级。

自助服务水平的高低是云厂商的核心竞争力，阿里云经过过去几年的积累，已经有了非常高效的自助服务能力。今天就将这些能力透露给最终用户。

目前阿里云自助服务功能可以覆盖 80%的 ECS 常见问题，剩余 20%不能覆盖的问题依然可以通过开工单解决。

对于 80%的问题，解决周期从几小时缩短至分钟级，这就意味着了户的故障修复时间大大缩短，提升了用户的体验。

整个自助服务过程中完全不需要人工参与，所有操作记录在用户端可见，保证安全合规，无隐私泄露风险。

诊断工具和修复工具都是通过 AI+数据的方式，借助阿里云海量的工单数据，可以越来越精准地进行问题诊断和修复。

二、智能诊断

1. ECS 常见问题

自助修复工具背后，需要厂商有准确的健康诊断能力，发现故障的存在与产生的原因。

ECS 最常见的问题可以分为四类：实例无法远程访问、实例无法启动和停止、实例性能异常、磁盘扩容未生效等。

实例无法远程访问，包含 SSH，VNC，或者是 RDT。这样的远程无法访问问题造成的原因是千差万别的，如网络不通，实例没有启动，服务异常等等。即使是网络不通背后也有很多原因，如安全组不通，运营商的网络出现故障。因此对故障的诊断并不是简单的 if else 的问题。

2. ECS 诊断能力

阿里云提供了一键开启 ECS 健康诊断能力，为了达到 80%的目标，需要进行全面的体检，从内到外分别是 ECS 服务自身的健康诊断（包括阿里云网络服务，数据化服务，后台硬件服务），磁盘健康诊断（如存储空间，IO 读写速率，磁盘本身的一致性），网络健康诊断（包括网络链路层诊断，网卡丢包，网卡加载等），Guest OS 健康诊断（网络配置，关键文件配置错误，权限错误等等）。

下图展示了目前所支持的 ECS 诊断能力。

首先，从用户场景方面，针对无法远程连接问题将虚拟化异常、物理机异常、资源争抢受限（入门级的实例中，会出现一台机器上存储资源争抢的情况）、服务控制侧异常等现象根因透露给用户。

针对实例无法停止或启动问题，着重诊断磁盘健康服务，所谓磁盘加载异常指的是云盘在 Guest OS 以内加载失败，还有磁盘 IO Hang，磁盘读写受限，扩缩容异常等根因。

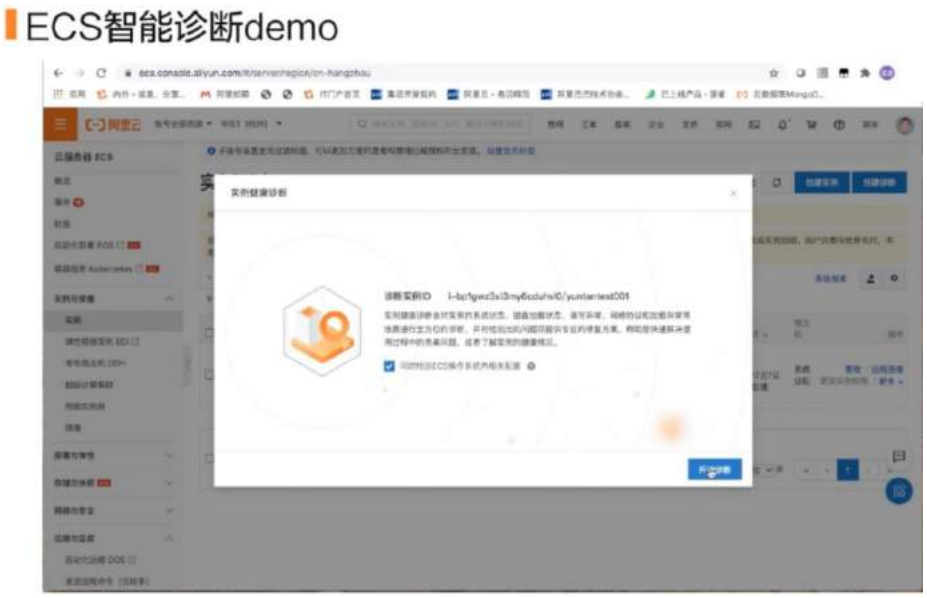
网络问题分为几类不同的表象，最常见的有网络延迟、网络丢包等。网络健康服务会针对网卡加载异常、网络链路异常、网卡丢包、网络会话异常等现象进行排查。

ECS 诊断能力不仅覆盖底层网络，还会对 Guest OS 以内网络进行健康诊断。

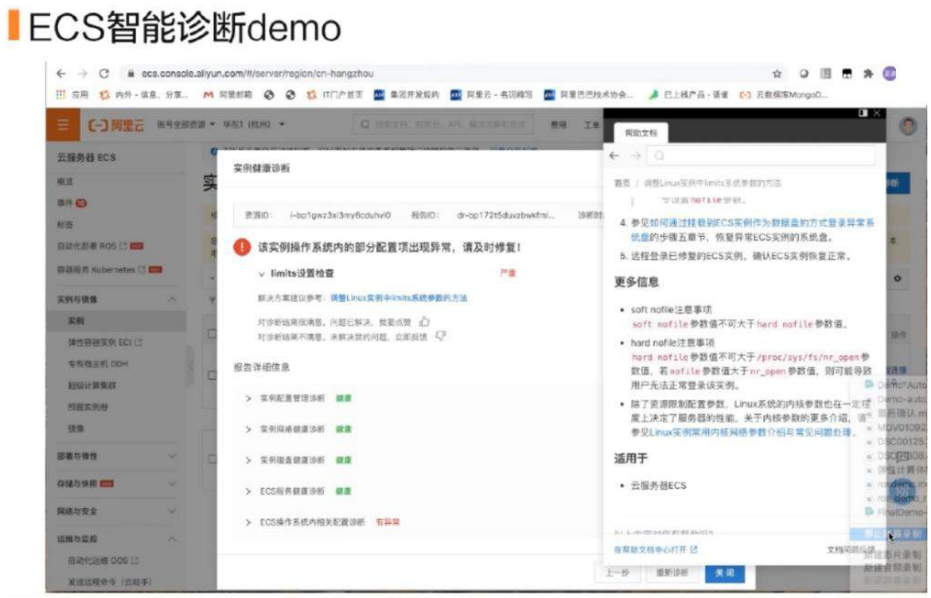
针对 Guest OS 问题，首先检查所有进程，检查 CPU 使用率，网络配置项，关键系统文件权限，文件系统配置等问题。从而判断 Guest OS 是否有可能出现问题，以及修复问题。

3. ECS 智能诊断 demo

那用户怎么样可以使用这个自助智能诊断服务？下面是一个简单的 ECS 智能诊断的 demo，右键菜单“更多”中有“实例健康状态”，勾选“同时检测 ECS 系统内相关配置”，就可以进行包含 Guest OS 的更全面的检查。如果不勾选则只会对服务侧进行检查。因为 Guest OS 的检测需要用户授权才能执行。可以发现一共进行了 54 项检查，用户可以继续查看针对报告和详细细节。最后会请求用户反馈。



如果检查不通过，则如下图中一样可以排查出是哪些项有问题。下图显示是 Guest OS 中 Linux 系统参数配置异常。下方给出了详细文档帮助用户进行问题修复。

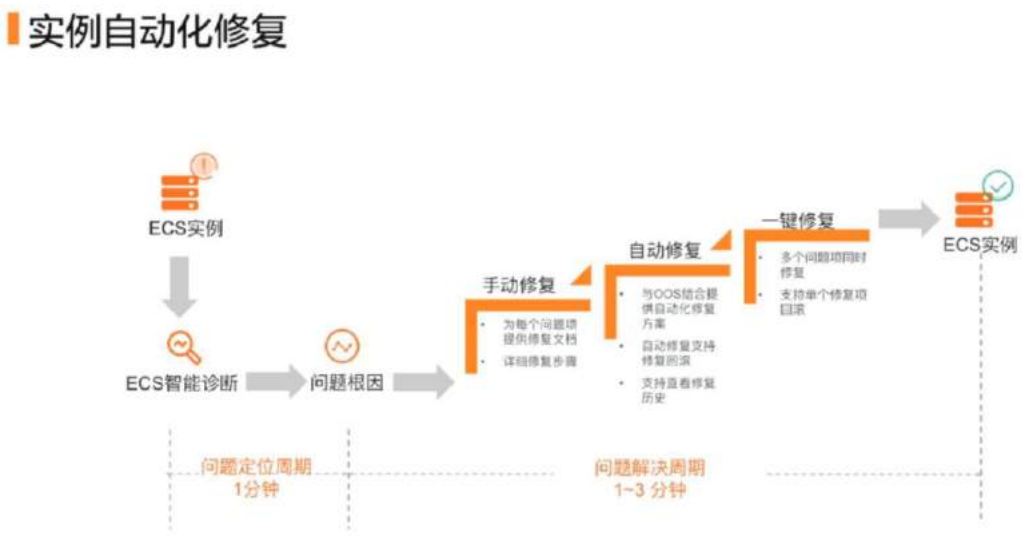


三、自动化修复

1. 实例自动化修复

诊断本身只是第一步，当诊断出来根因之后需要进行修复。目前 ECS 自助服务提供的是文档和链接，指引大家进行修复，由此可以更加保护用户隐私。

阿里云目前正在做自动化修复功能。实例自动化修复逻辑如下图，问题定位周期是 1 分钟，即问题诊断过程，找到根因之后用户可以手动修复，此时提供修复文档和详细修复步骤；还可以选择自动修复，即与 OOS（阿里云运维编排系统）结合提供自动化修复方案，为修复场景提供一系列的公共模版。



公共模版指的是阿里云对公有云的最佳实践。在具体的修复场景中再次进行检查，判断问题根因，再集合用户配置进行问题修复。阿里云也在控制台中提供一键修复能力，支持多个问题同时修复。而由于修复本身是一个高危操作，因此还支持单个修复项的回滚。阿里云即提供 Guest OS 内部的修复能力，还提供基于快照的整体修复能力。在修复之前对整个 ECS 实例做备份，修复之后重新诊断问题是否修复成功，要求用户确认。如果用户确认修复不成功，则进行回滚，恢复到实例之前的状态。秒级快照能力为一键修复提供了强有力的支持。

2. ECS 修复能力

对修复能力而言，而是着重对应诊断能力。自助诊断服务判断出问题根因，针对具体的根因，提供不同的修复能力。

下图展示了针对诊断能力提供的修复能力一览表。

ECS修复能力一览表

比如，针对 ECS 系统服务或磁盘修复，首先进行重启，再进行重新部署。此时可能丢掉本地化实例原始数据；再进行自动故障上报，故障比较多时进行故障隔离，帮助客户进行迁移操作。

针对网络问题，修复系统会进行安全组规则调整；同时做故障网络设备隔离，如果故障是由底层的网络设备引起的，修复方案就是使用正常的设备提供服务。

当发现 Guest OS 以内的网络配置不正确时，修复系统会自动校正配置使得网络通畅。

ECS 系统服务修复方案中包括，推荐用户进行实例规格升级、磁盘规格升级、关键系统问题权限授予、或者手动开启若干个关键系统进程（ssh）支持远程连接、还有磁盘文件挂载变更、网络参数变更等。

这些能力还会随着诊断能力不断的扩充，未来希望 95%的工单都可以自动诊断，以及 80%的工单可以自动修复，剩余的是人工诊断和修复。

3. 修复能力透明合规性

修复能力本身是一个风险操作，因此其透明合规性非常重要。

阿里云通过运维编排服务 OOS 提供自动化引擎，云助手命令提供 Guest OS 内的执行能力。

OSS 和 Guest OS 都是用户侧的工具，使用了用户侧的 RAM 权限进行所有操作。这样使得一切修复逻辑可见，管理员可以在用户侧看到所有操作步骤，包括 OOS 公共模版命令和云助手公共命令。阿里云目前已经在 Github 上开源了云助手所有代码。

其次，一切操作可回滚，通过镜像和快照实现整机的数据备份。首先是进行操作系统内的数据备份，在无法回滚时进行整机的数据备份。并且一切权限可控，阿里云所有的操作都是通过 RAM 角色，而 RAM 角色是由管理员自己配置，随时修改或禁用 RAM 角色的 RAM 功能。

最后，一切修复操作都可以审计和追溯。自助修复功能很快会与大家见面，感兴趣的用户可以先行体验自助诊断功能。

修复能力的透明合规

1，运维编排服务OOS提供自动化引擎，云助手命令提供GuestOS内的执行能力。
2，一切修复逻辑可见：OOS公共模板和云助手公共命令，代码开源
3，一切修复操作可回滚：镜像、快照，数据备份
4，一切权限可控：阿里云RAM角色控制。
5，一切记录可审计: 阿里云操作审计ActionTrail

四、诊断数据背后的 AI 和数据能力

1. AI 算法

上面提到的 AI 修复，自动诊断以及优化推荐都只是冰山之上的用户体验，在冰上之下是 AI 算法和数据中台的支持。

AI 算法中最重要的是根因分析和特征分类。

- 根因分析是指，在日志数据和 Guest OS 中发现很多可能的问题原因，但究竟哪个是真正的 root cause 则需要 AI 做分析。人分析时会看时间，发生的顺序，调用链路，AI 也是同样的逻辑。
- 特征分类是针对用户的操作和异常进行分类，将用户的操作、配置、异常分配到具体的根因上。
- 态势感知是对风险的预测。
- 预测和推荐其中的预测是非常重要的，很多诊断需要在用户没有感知时就提供异常诊断，将风险扼杀在发生前。
- 用户画像是针对用户本身的属性进展诊断，不同的用户往往有不同的操作记录，不同的异常问题，以及不同的行为，这都需要不同的诊断，因此用户画像和行为分析可以辅助自助诊断。
- 决策树或专家经验也是重要的诊断方式。

支持 AI 算法的是数据中台，无论是数据的清洗还是打标都离不开数据中台的建设。

2. 数据中台

数据中台涉及数据采集、数据清洗、数据分析和数据模型。
数据采集中分为三类数据，包括实时数据、准实时数据、离线数据：

- 用户当前的健康数据、网络数据都属于实时数据。
- 用户当前的操作记录、监控数据属于准实时数据。
- 离线数据是指过去每一天的数据的快照，离线数据是可以支持构建用户画像，行为分析的数据。

同时从采集数据源角度可以分为物理机数据、虚拟化数据（虚拟化库，如阿里云神龙）、网络数据（网络组件）、控制面数据（用户所有操作记录）、Guest OS 内数据（云监控及云助手采集数据）。

所有数据采集完成后是非常杂乱的，需要进行进一步处理。首先将所有数据变成监控项，产生告警、metrics、日志。同时提供查询分析能力，即提供给 AI 还提供给网络平台。事件通知是通过数据产生的数据推送和订阅，如 AI 中台对某一列数据感兴趣，则可以进行订阅，特定事件出现时推送给订阅对象。

3. AI 举例

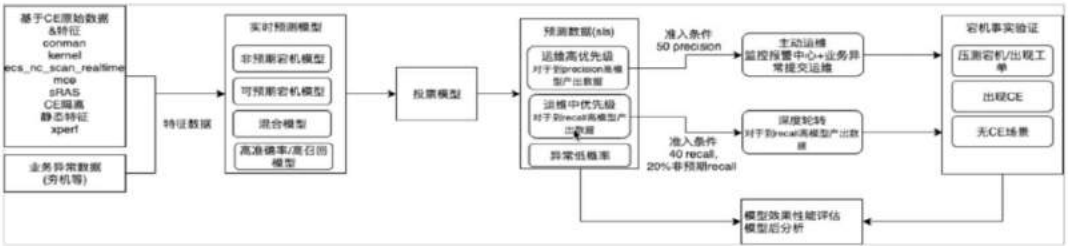
实时内存异常感知

下面举一个例子，即实时内存异常感知。实际上，数据和算法处理过程中会遇到大量的类似的例子。实时内存异常感知指的是当内存出现可能预期的错误时，会影响到虚拟机的稳定性，因此需要第一时间识别到内存的错误并进行内存的替换。

下图展示了针对此类实时内存异常感知问题所对应的 AI 算法模型运作流程。

AI举例-实时内存异常感知

- 准确率 :70%以上
- 实时预测链路延时控制在100s以内



首先，采集原始数据，包括 CE（更正的错误）原始数据、特征等；

接下来，进行数据处理，特征数据进入到实时预测模型中，进行非预测宕机模型、可预测宕机模型、混合模型、高准确率、高召回模型；

下一步进入投票模型，投票到各种各样的优先级的 sls 预测数据中，当 precision 大于 50%时进入主动运维监控报警中心，产生告警；

告警生成后，进行宕机事实验证，如果出现问题了表明算法正确，如果没有出现问题则回到算法中进行更正。

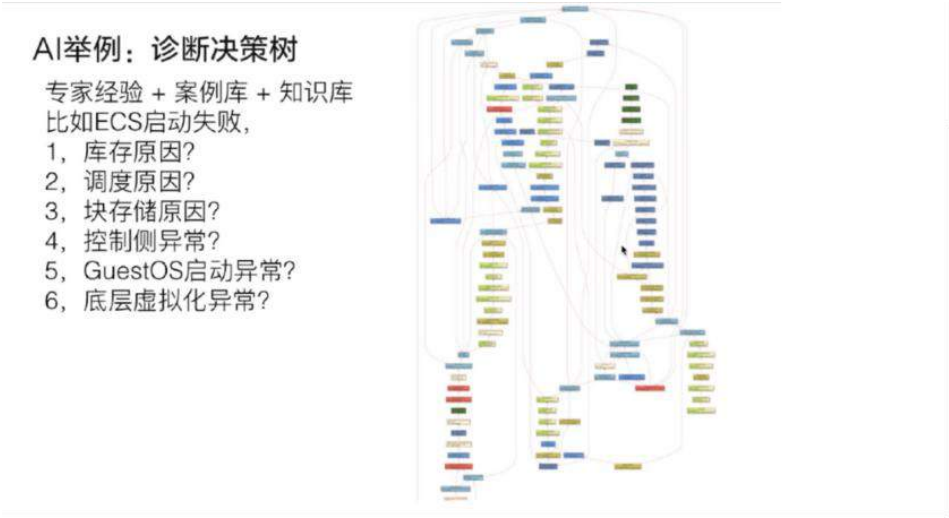
诊断决策树

此外，再给大家介绍一个例子：诊断决策树，这个例子很容易理解。

诊断决策树有三个关键要素，首先是专家经验，其次是案例库，还有知识库。

大量的工单经过一线、二线及三线人工客服形成了专家经验；案例库是阿里云内部的；知识库是提供给用户用的。

专家经验是基于案例库和知识库抽象出来的各种逻辑规则，比如 ECS 启动失败原因可能是库存原因、调度原因、块存储、控制侧异常、Guest OS 启动异常、底层虚拟化异常等。专家决策和决策树会依次排查可能的原因，下图中每个方块都是一个案例，决策树中专家经验和案例库是固定的，但如果某个链路中的案例很多，会先走这条链路，也就是说决策树中的案例库先后顺序和权值是 AI 自动调整的。



总结

自助服务是云厂商的核心能力，自助诊断和自动修复是自助服务的核心功能。当大家遇到 ECS 问题时，请先尝试自助诊断服务，而不是直接开工单，这样可以更快速的解决问题，节省时间。最后，ECS 自助服务团队求贤若渴，欢迎大家加入！有需要的同学可联系本次演讲嘉宾滕圣波（云普）。今天的分享到此结束，欢迎大家持续关注阿里云 ECS 更多服务能力的更新。

2.5 ECS 数据保护–数据备份新特性与最佳实践

摘要：本文中，阿里云智能弹性计算专家余初武（悟元）将结合阿里云近期推出的数据备份新特性（快照极速备份、一致性快照组）来介绍云上环境如何做数据备份的最佳实践；适合需要构建云上架构的工程师，架构师和云上实施从业人员收看。

演讲嘉宾简介：余初武(悟元)，阿里云技术专家，2011 年加入阿里巴巴，一直从事服务端研发工作；2015 年加入阿里云 ECS 团队，在 ECS 管控、云盘、快照等多个管控领域有丰富的研发经验。

本次分享主要围绕以下三个方面：

一、快照极速可用特性
二、一致性快照组
三、总结与思考

数据是企业重要资产，作为存储数据的介质，IT 设施发生问题是不可避免的，人为误操作或者程序 bug 导致数据丢失的情况也偶有发生。因此，每一个企业都应该做好数据备份，保证数据的安全与业务的可用。

为了提升备份与同步的性能，阿里云推出了快照极速可用特性和一致性快照组，为了客户提供更高性能的数据备份功能。

本文将重点介绍如何利用这两个新特性，以及阿里云提供的各种运维部署工具，便捷地完成云上自建数据库的数据备份的两个最佳实践。

第一个是自建数据库的磁盘扩容场景，适用于大多数的企业；第二个则是使用了多磁盘自建数据库的场景，则更多见于大型复杂的业务。

一、极速可用特性——秒级，非一致性数据备份

阿里云 ECS 的极速可用特性主要包括四个方面，分别是快照秒级可用、云盘回滚性能 0 损失、ESSD 增值特性以及全地域支持等。阿里云 ECS 极速可用特性的典型使用场景包括快速搭建研发测试环境；业务关键配置的变更保护，实现秒级备份相关磁盘数据；云盘极速回滚，并实现回滚的磁盘性能无损耗。

基于极速可用的特性，用户仅需要几秒钟的时间就可以复制出一个新磁盘。

这一过程也非常简单，首先创建一个带极速可用特性的快照，关键参数的设置如下图所示，主要包括 InstantAccess 和 InstantAccessRetentionDays，前者设置为 True 就可以设置成为极速可用的快照，后者则是极速可用特性的保留天数，可以让这特性到期之后就会自动被关闭。当快照创建完成（极速可用特性开启的情况下，不需要等快照进度完成）之后，就能够快速创建磁盘并立即挂载使用。

案例：自建数据库，磁盘空间不够，怎么办?

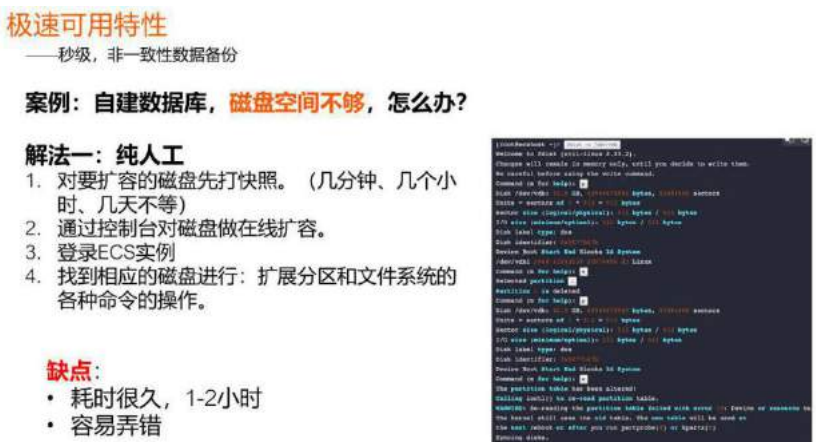
在这样的情况下，最直接能想到的解决方案是纯人工方式。

首先，对于需要扩容的磁盘打好一个普通快照，这个过程一般都比较慢，往往需要几分钟、几小时以及几天不等的时间。

其次，需要人工登录到控制台对磁盘进行在线扩容。

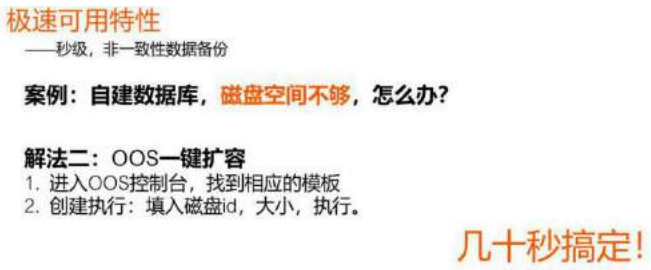
再次，要登录到实例内部找到相应的磁盘进行扩展分区以及文件系统等各种命令的操作，而这些命令往往是非常复杂的，也是非常容易出错的。

这一方案的缺点十分明显，那就是耗时很久，平均需要 1 到 2 小时，而且很容易出错。



而目前阿里云推荐的最佳解决方案是将上述过程全部通过编码实现自动化，做成 OOS（运维编排，Operation Orchestration Service）的模板，通过 OOS 模块实现一键扩容，完成上述方案的全部过程。

这种方案的使用方式就非常简单了，用户可以直接进入到 OOS 控制台，找到相应的模板并创建一个相应的执行即可，整个过程只需要几十秒就可以完成，而且可以进一步优化至十几秒。



接下来对于刚才提到的 OOS 一键扩容的关键技术内幕进行讲解。

其实在该方案背后主要包括三个关键技术：分别是快照的极速可用特性、通过云助手执行扩展分区的命令以及磁盘的序列号。

这里值得注意的是通过云助手执行扩展分区命令时，我们无法知道具体扩展的是哪一块磁盘，因此才需要磁盘的序列号。

磁盘序列号这一特性目前在公有云 ECS 上也已经上线了，用户通过 DescribeDisks 就可以返回磁盘序列号 SerialNumber，之后通过云助手将磁盘序列号传递给 GuestOS 内部的脚本，而 GuestOS 内部的脚本则可以通过 udevadm info 这串命令获取任意一块磁盘已挂载设备的序列号，这个序列号与 DescribeDisks 返回的序列号是完全一致的，而且从磁盘诞生之后，序列号就不会再发生任何改变，因此可以作为磁盘在 GuestOS 内部的唯一标识，并且与 OpenABI 的接口实现唯一关联。

这样才能帮助我们准确无误地找到需要扩展的磁盘去执行相应的命令。同时，因为 ECS 具有快照极速可用特性，秒级地打出了一个数据备份，一旦发生任何意外，还可以通过快照实现秒级回滚，基本可以做到万无一失地实现自动扩容过程。

二、一致性快照组——崩溃一致性数据备份

介绍完快照极速可用特性，我们继续跟大家分享下一致性快照组。

一致性快照组的主要特点主要包括四点：即多云盘 IO 写入一致性、ESSD 云盘增值特性、实例级别保护以及功能免费。

适用的场景主要有三种：

第一种场景，企业上云的时候可以实现实例级别整机的保护和备份；

第二种场景，自建数据库特别是跨多云盘自建数据的模式下，一定要使用一致性快照来备份；

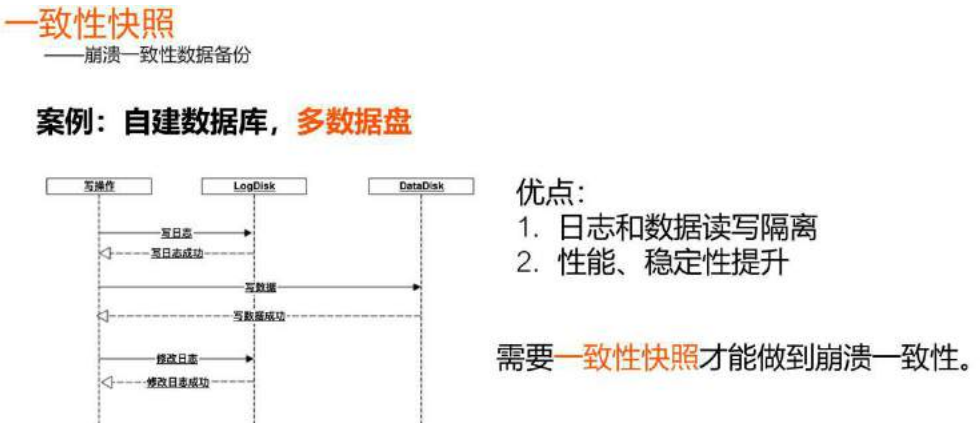
第三种场景是 SAP HANA 的整机一致性保护，也需要用到一致性快照。

案例：使用多数据盘自建数据库

这里要介绍的案例是自建数据库时使用了多数据盘，这样的做法主要是为了将数据库常见的日志和数据拆分到独立的云盘上去，使得整个数据库的性能和稳定性都能够得到较大的提升，同时实现日志和数据的读写隔离。

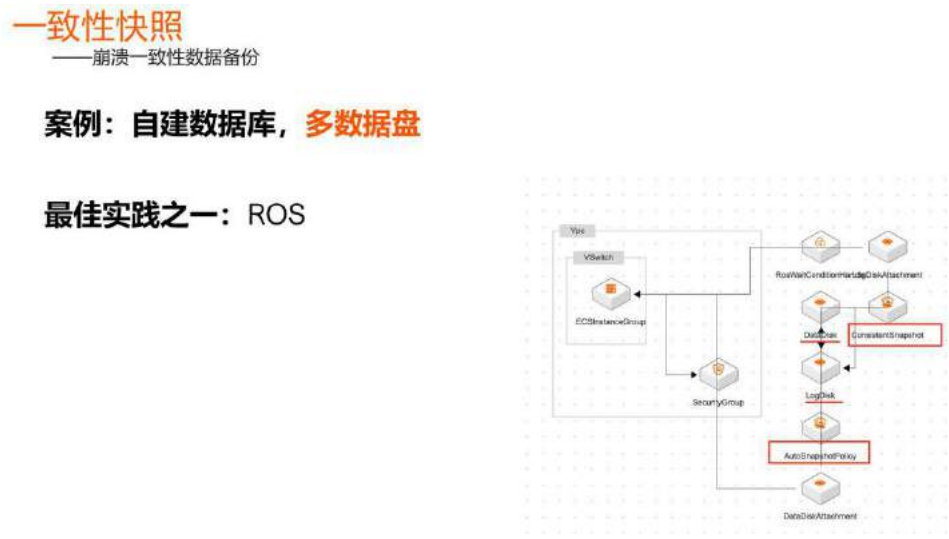
在这种情况下，一旦需要拆分就会遇到两块磁盘上数据存在强关联关系的问题。熟悉数据库的同学都知道，任何一次写操作都会先进行日志操作，日志写完之后再去修改真正的数据，当数据写入完成之后再回来修改日志，比如像 MySQL 的 Redo 日志等。

可见，日志和数据存在强依赖逻辑关系，此时如果打普通快照，那么存在任何一点点时差都会导致写入数据在两块磁盘上存在不一致的问题，此时就必须要用到一致性快照，实现崩溃一致性的数据备份。



对于这样的案例场景，阿里云也提供了最佳实践，也就是使用 ROS（资源编排，Resource Orchestration Service）。

该方案的基本过程就是将上述理念通过 ROS 实现模板化，通过 ROS 创建完全一样的数据库系统。方案关键点在于创建两块独立的数据盘，一块放数据，另外一块放日志，同时对于两块数据盘赋予系统的数据库快照策略 Auto Snapshot Policy，并定期地对于两块盘进行数据备份，并且使用一致性快照进行备份，避免出现废弃数据的问题。



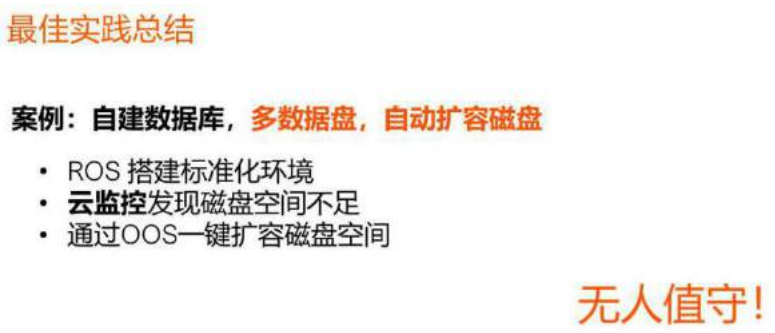
三、总结与思考

以上的案例都是基于自建数据库的，这两个案例的关键点在于多数据盘和自动扩容磁盘。

将上述两个案例进行串联才能够看到真实的使用场景，也就是先用 ROS 固化上述提到的最佳实践过程，也就是实现多数据盘，即日志盘和数据盘的隔离，并且使用自动快照的策略定期地打一致性快照。

与此同时，配上云监控就能够在磁盘空间不足的时候，及时报警，此时在通过 OOS 一键扩容实现磁盘的自动扩容。

当然，这个过程还可以更进一步优化，在 OOS 控制台配置相应的云监控项目，当收到监控项报警之后自动触发 OOS 运行和扩容的模板进行一键扩容，真正地实现自动扩容，也就是所谓“无人值守”。



对于本次介绍的新特性进行总结，本次主要介绍了极速可用和一致性快照两个新特性，这两个新特性很快就会上线供大家使用。

对于极速可用特性而言，建议结合 OOS、云助手来磁盘或者其他场景的自动运维实践。如果要对云盘进行操作或者自动化运维则需要使用磁盘序列号在 GuestOS 内部唯一地标识一块磁盘，这样才能做到准确无误。一致性快照则是在 MySQL 这种多盘场景下才会使用，主要用来实现奔溃一致性备份。

本次分享到此结束，感兴趣的同学可持续关注和学习云上环境数据保护最佳实践。

第三章 架构优化

3.1 云上高弹性、低成本解决方案

摘要：本次内容由阿里云智能高级技术专家贾少天为大家介绍弹性伸缩和弹性供应的产品力和功能概要，帮助您快速了解弹性的能力和应用方式，方便快速在业务场景中使用弹性来解决日常业务变化对于资源的弹性需求，如果您已经在使用按量、spot 实例或者对成本优化有需求，可以着重进行了解。

演讲嘉宾简介：贾少天，阿里云高级技术专家，2010 年加入阿里云，目前专注于 ECS 弹性体系的建设，通过弹性伸缩和弹性供应组多种方式来满足用户弹性同时降成本的需求，帮助用户在高可用和低成本间找到更合适的解决方案。

本次分享主要围绕以下三个方面：

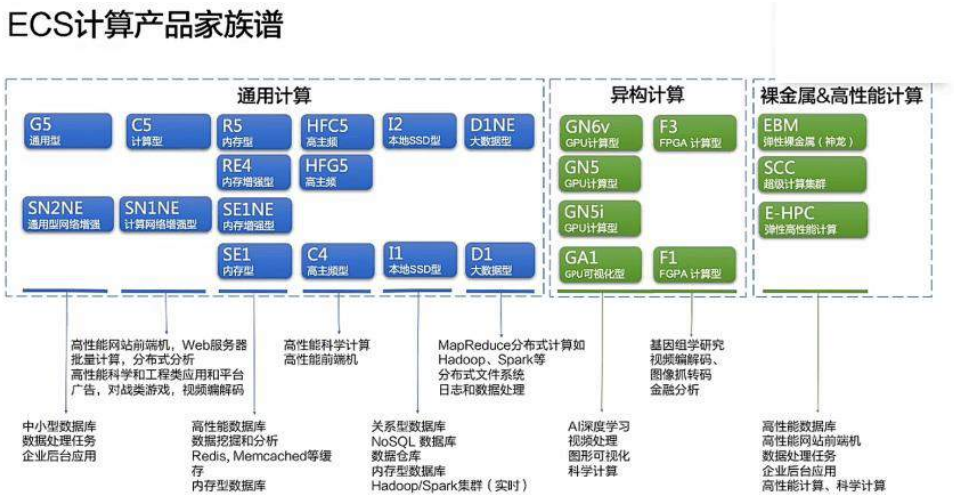
一、ECS 概况
二、弹性伸缩
三、弹性供应

本次分享主要帮助用户快速了解云上弹性能力和应用方式，同时通过弹性伸缩和弹性供应组多种方式实现成本优化。

一、ECS 概况

1. ECS 计算产品家族谱

ECS 产品是云服务器产品，具有各种各样实例规格，不同的实例规格具有不同是适用场景，用户可通过自身使用的业务场景或是运行内所需性能等方面的考量选择适合自己的实例规格。针对不同的场景，我们所推荐的实例规格不同，如下图所示，对比自身业务领域的情况选择合适的实例规格。



2. ECS 付费方式

阿里推出后付费和预付费两种付费类型。

后付费按照用户使用情况付费，按量付费是秒级计费方式，使用一秒钟或一分钟则收取一秒钟或一分钟的费用。抢占式实例是根据价格的意向来使用 ECS，根据市场价格变化或是需求发展变化系统会随时进行资源释放，但可以保障用户最低一个小时的使用，且用户必须保证业务产品是无状态业务产品，这样可以更大程度上降低业务成本。

预付费模式可按月、按年、按周的方式购买实例规格，如果用户预期两年内使用阿里云服务，则可选择按年方式购买，在年付费方式的基础上还具有各种各样的折扣。如果用户的业务场景预期以周为单位，例如已知下一周有大的负载情况或做一期业务场景活动推广等，则用户可按周的方式购买，同时也可按月方式购买，用户可根据自身的不同业务场景自定义选择购买方式。

另外在此基础上阿里还提供了预留实例，预留实例与包年包月相近，可按照年、月、周的方式购买，但该方式不根据资源实例规格长短定，用户可选择自身范围的实例规格在一定周期内使用，是资源与折扣的解耦。

2020 年，阿里云还推出了最新的付费方式——节省计划，节省计划是一种折扣权益计划，承诺在一定期限（1 年或 3 年）内每小时使用固定数量的资源，即可拥有较低的按量付费折扣。节省计划需要按小时承诺消费金额。相比起包年包月和预留实例券的方式，节省计划的灵活性更高，可让用户灵活变更实例的规格与地域。

ECS付费选择

- 付费方式的灵活使用是获得业务敏捷性的基础，也是降低IT成本的最核心手段，阿里云ECS提供极多样的资源付费方式：

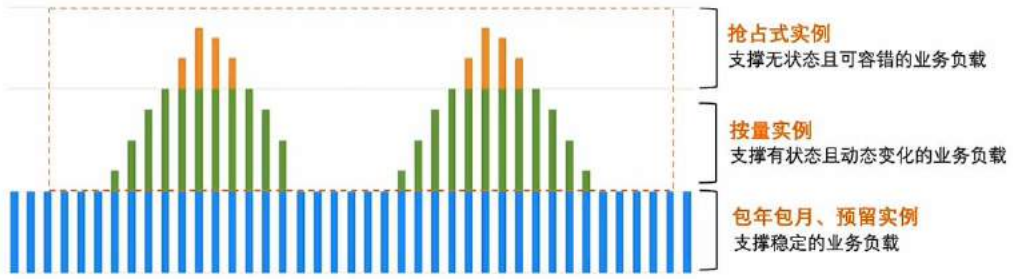
	后付费		预付费			
功能项	抢占式实例	按量付费	按月（1~N）	按年（1~5）	按周	预留实例
付款方式	后付费		预付费，最短1个月	预付费，最短1年	预付费，最短1周	1或3年 全首付/部分首付/0首付
价格	按量价的1折起	秒级按量付费	折合小时价低	折合小时价最低，最低三折	灵活性及成本均衡	与包年月相近
释放	系统/用户释放	用户释放	到期释放		到期释放	客户主动释放
适用场景	应对爆发业务临时扩展，压测，科学计算，批量计算等，转码等场景	应对爆发业务临时扩展，临时测试	固定的7x24服务，如web服务、数据库		短期计算业务，业务选型测试	DevOPS场景 资源与折扣解耦

用户可能会在这些实例规格和付费类型的选择方面存在各种各样的问题，不了解在什么情况下选择最合适的实例规格和付费类型，所以阿里会推荐一种面向业务的最佳实践。

购买包年包月资源或预留资源可作为基础业务负载的保证，例如购买二十台资源作为日常基础请求量的负载保证，在此基础上购买按量付费类型来支撑每天负载发生的变化，只要能快速扩容或是负载快速缩容，就可以使用比包年包月更低成本的资源。另外用户具有无状态的业务创建，可做到高容错，可在此基础上使用 Spot 扛最上面的高峰值。

希望通过包年包月、按量、抢占式三种付费类型的结合，让用户整体的成本达到最低的同时拥有高质量的服务器。

低成本的最佳实践



多种付费类型组合最低成本完成业务支撑

3. ECS 资源弹性交付方式

阿里云提供四种 ECS 资源弹性交付方式，Create Instance 是最早期提供的单实例支付方式，每次可交付一个实例。随着业务的发展在此基础上提供了 Run Instances 接口，单次可批量交付 100+实例，但该支付方式限制单可用区和单实例规格。随后阿里提供弹性伸缩和弹性供应两个产品，来帮助用户解决更加复杂的业务需求。

弹性伸缩是自动化的交付工具，可以一次性批量交付（2000/50000）实例。

如果伸缩组配置的普通安全组，那伸缩组的最大实例数是 2000，如果有更多实例的需求的话就需要配置企业安全组，同时一次配置后可重复使用。

另外，该产品支持定时和触发等功能，例如 CPU 达到 70%时触发扩容，少于 30%时触发缩容，或是用户对于自身业务场景非常清楚扩容的时机则可进行主动触发，预测功能可通过过去使用情况自动预测未来的发展变化，自动进行扩容缩容。

在此基础上该产品支持多可用区和多实例规格，方便用户在不同的可用区做高可用甚至提供更多的实例规格使成功率更高。另外还提供了成本优化模式。

弹性供应在原有基础上提供了交付资源至交付计算力的环节，同时也是 ECS 原生大规划交付方式。原先用户交付 1000 个实例，现在可以交付 1000vCPU 或是其他纬度的业务场景，变成一种灵活度很高的产品形态。同产品伸缩相同批量交付实例规格达（2000/50000），以及配置的重复使用。同时也支持按量+Spot+RI 的组合。

4. 云上弹性面临的挑战

用户在云上弹性方面面临各种各样的挑战：

- 例如某创业公司在 app 爆红后频繁无法访问，是由于业务突然大量涌入后业务增添的负担支撑不住这么大的压力；
- 或是在线视频网站每天早八点或中午十二点都会迎来业务的高峰期，而对于如何应对高峰期则可能需要定时的扩容缩容；
- 或艺人爆出八卦引发社交用户的热议，突然变为之前频率流量的十倍左右的负载，如何快速解决该类问题的扩容来满足业务场景；
- 或是双十一大促销需要临时扩容时企业如何准备；
- 以及业务低谷期也保有全量资源，闲置成本非常高时包年包月成本如何解决。

在云上很多时候会认为不可同时做到高可用和低成本，如果做到低成本可能顾不到高可用，在后面讲师会分享某些场景上可以同步做到高可用和低成本。

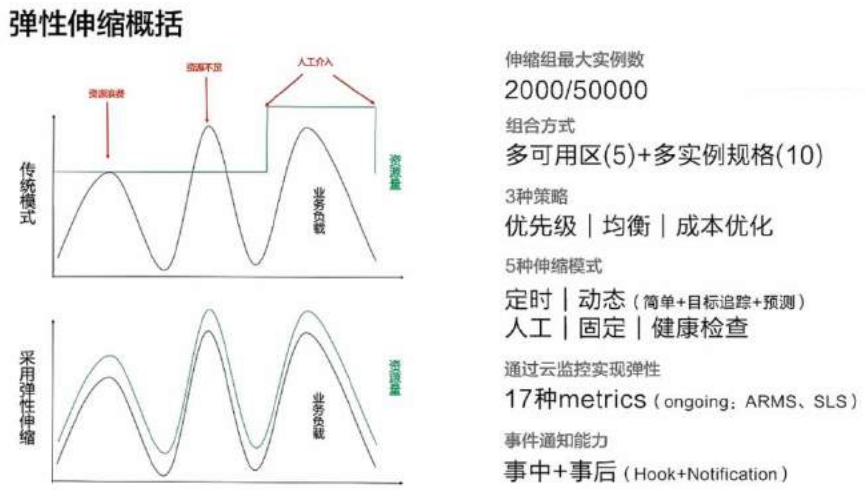
二、弹性伸缩

1. 弹性伸缩概述

弹性应用的产品应该满足的业务场景有几种。首先是传统模式，会保有一部分资源，但该资源刚好可以扛住平时的高峰期，在低峰期就会形成资源浪费，或是在高分期扛不住，只

能扛住业务的平均值时，那么在高峰期时会无法支持业务导致崩溃或不可用，又或是到一定程度需要扩容时要进行人工扩容，但是需要收容时还需人工自行判断。所以采用弹性伸缩就会自动根据整体业务负载的情况自动做扩容缩容，保证用户用最少的资源负载业务整体的变化。

从下图右边中可看到关于弹性伸缩整体提供的能力：



伸缩组最大实例数：以组的方式进行管理，最大实例数值为 2000/50000。

组合方式：支持五种可用区和十种实例规格提高了整体资源交付的成功率。

三种策略：提供优先级策略、均衡策略及成本优化策略。优化策略是指用户可设置一批资源，该策略按照每种资源创建的成功率自动判断每种资源当前交付使用最快最高成功率的方式。均衡策略是指用户设置多个可用区后，在多个可用区下进行整体均衡的分配，这样可保证一旦某一个可用区下发现整体不可用时业务依旧可以继续有效运行。成本优化策略结合 Spot+RI 场景可帮助用户在稳定的同时做到低成本。

五种伸缩模式：定时、动态、人工、固定和健康检查模式。动态模式分为三种：简单、目标追踪和预测。简单伸缩策略可以定义为 CPU 达到 70%时增加五台，CPU 小于 20%时减少三台，用户可根据自身经验判断当天需要扩缩的数量。目标追踪可定义为整体业务需要保持 CPU 达到 60%，当业务高峰期 CPU 达到 80%时这时系统会自动帮用户换算具体需要扩容多少资源才能满足业务不受损，当 CPU 降到 40%或 30%时系统会判断当前整体减少多少资源来保证业务最低运行。预测策略根据过去使用情况来预测未来的使用情况。

通过云监控实现弹性：支持十七种云监控指标进行触发包括 ASMA、SLS 等。

提供事件通知能力：事中+事后整体的事件订阅，可保证业务起到全局控制。

2. 弹性伸缩核心概念

弹性伸缩的核心概念主要分为四种，如下图所示。

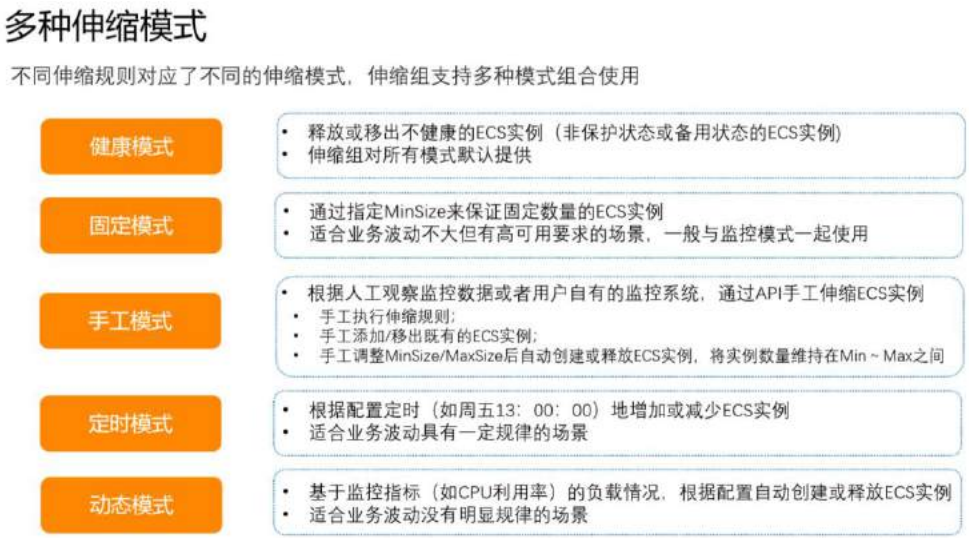
在伸缩组上设置弹性伸缩配置，例如实例规格种类，整体启动脚本的样式，使用登录方式以及镜像等，设置完可随时使用和修改。

在此基础上设置伸缩规则和通知，伸缩规则是指具体增加多少台减少多少台，或者按照目标追踪的方式设置 CPU 保持在 70%，另外在每次成功或失败时可以订阅最终的结果到云监控或 MNS 消息队列中进行编程，来应对每次成功或失败的结果，里面会包含成功时扩容多少资源，资源 ID 是什么样的，或者对接到云监控系统或者钉钉，可以知道导致失败的原因并快速做出预警进行干预。

伸缩任务具有定时任务，支持一年和临时提醒功能，在一年期限到达前会及时进行提醒，报警任务涉及到 CPU 或 Memory 等，另外伸缩任务还可进行自动或手动触发。

3. 多种伸缩模式

阿里云弹性计算提供不同的伸缩的规则，对应了不同的伸缩模式，包括：



健康模式：通过健康检查发现一些不健康 ECS 实例时会进行自动清理，并创建新实例来替换不健康实例。

固定模式：可设置最大值最小值实例期望数量来保证业务正常运行。

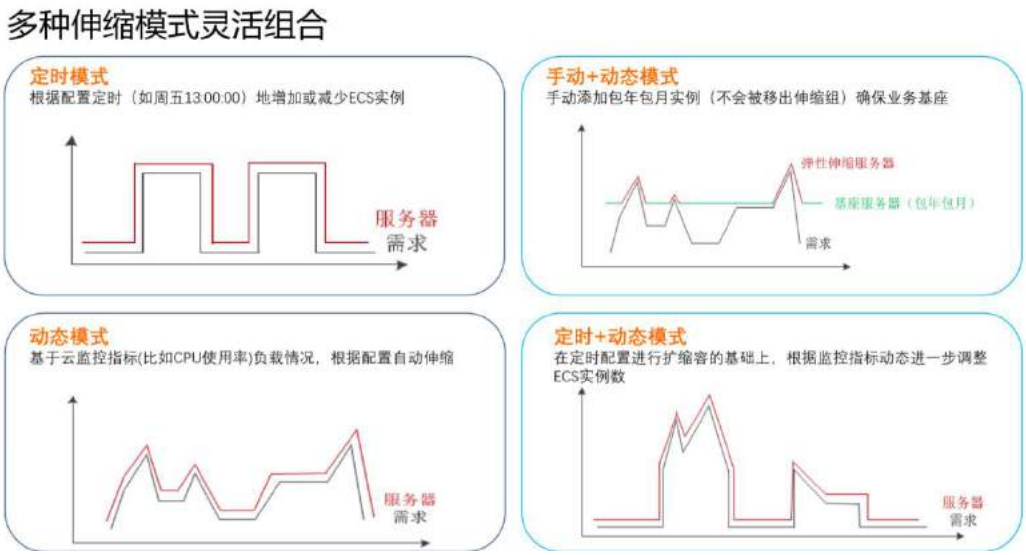
手工模式：用户如果对自己业务场景有非常详细的了解或自有监控系统时，可通过 API 方式执行手工模式。

定时模式：例如在线视频网站每天早八点或中午十二点会迎来业务的高峰期，可以在此期间进行定时扩容，度过高峰期的两三个小时后再缩容。

动态模式：通过负载变化进行扩容缩容。

4. 多种伸缩模式灵活组合

图片中显示在不同场景下不同模式的表现。



定时模式是在一个时间点扩容运行几个小时后再缩容，整体线条是直上直下的。

动态模式是服务器的变化走向是与整体负载环境的变化走向一致，趋向于负载的变化。

手动+动态模式购买包年包月资源作为底座来扛日常业务，高峰期的负载环境变化通过按年的方式扩容缩容来保证，这样整体稳定性更高，因为已有 60%或 70%的负载情况被 cover，剩下的 30%负载情况通过动态的方式 cover。

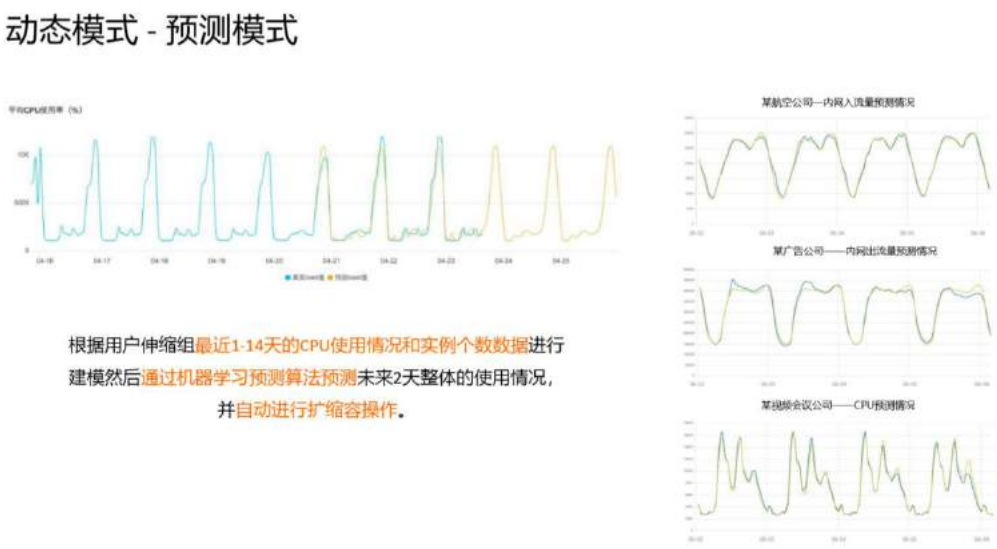
定时+动态模式说的则是例如资源扩容完后可能会发生数量变化，如有新电影或电视剧上映时具有高热度，此时原先扩容的值会不够用，则需要通过监控指标的变化在此基础上进一步扩容。

动态模式–预测模式

预测模式通过过去 1–14 天的 CPU 使用情况或实例个数的变化进行建模，通过机器学习预测算法预测未来 2 天整体的使用情况，并自动进行扩容缩容操作。

此场景适合非常规则的周期性场景，每天的业务在固定的时间点上进行固定变化，图片中展示的场景就很规律，每天都在固定的时间点周围进行扩容和缩容，这样所达到的预测结果会更加准确，所以推荐规则性的场景使用预测模式效果会更好。

该模式好处在于不用感知 CPU 的设置值，它会自动进行运算并自动判断多少实例可在此基础上满足当前的业务需要，是因为历史上实际的整体情况已经被纳入到所考虑的范围内。如果用户某一天做活动或推广的业务期间突然涌入大量流量时，也可叠加目标追踪模式，预测模式+目标追踪模式叠加后在发生变化时可自动感知和扩容，并在此基础上再进行扩容或缩容，所以频率性在遇到特殊的抖动或是特殊的变化时是有相关的规避方式的。



5. 成本优化方案

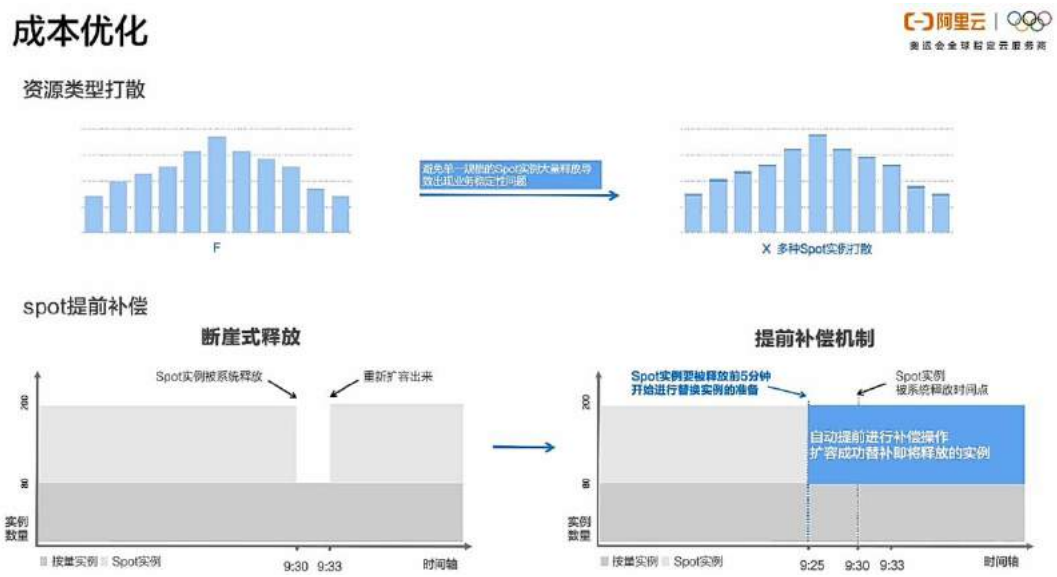
前面讲解了解到阿里云 ECS 的多种付费方式以及弹性伸缩的多种策略之后，用户可以利用这两种能力来实现成本优化。接下来会讲解多种成本优化方案，用户可根据自身情况自行选择。

在 ECS 付费方式中提到了包年包月、按量付费和抢占式 Spot 等不同付费类型，这些付费类型可组合成多种成本优化方案。用户可根据上图中显示的六种方案选择成本最低稳定性更高的使用方式。

然而在方案选择上阿里平台则非常推荐 A（预付费/RI+按量+Spot）方式，该方式将稳定性和低成本做到相对较好的结合，从上图右边图片中看出 A 方式的位置在四项中相对较好。但在此基础上也与其他产品进行了对比，例如 B 是预付费/RI+按量方式，C 是预付费/RI+Spot 方式，D 是按量+Spot 方式，E 是全按量方式，F 是全 Spot 方式。

因为 Spot 场景价格非常低，所以 F 方式的成本很低，而系统带来的一些释放行为，也是导致 F 方式的稳定性最差的原因。因此希望在 F 方式的基础上作出成本更低稳定性更高的使用方式，所以推出了成本优化模式新功能。

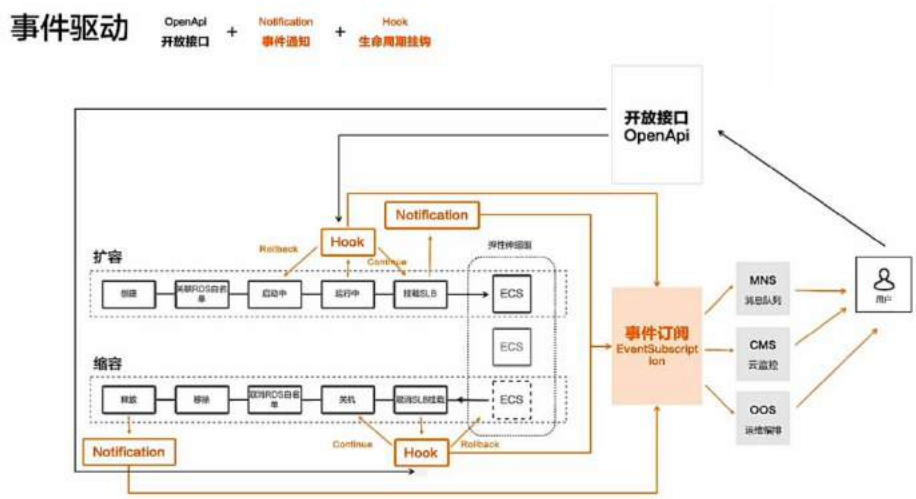
F 方式（全 Spot）在原先全部适用于一种实例规格的基础上，在此实例规格上做出一些调整，允许实例打散，将多种实例规格组合来满足资源的交付。下图中从左边一种实例规格到右边多种实例规格混合交付的方式，如果一次性交付是三种或四种，这样即便有一种实例规格出现问题也不会影响整体使用关联性。



另外，Spot 在此基础上又提供了提前补偿机制，Spot 实例在释放前五分钟可通过事件示例订阅此消息，在 Spot 实例释放前五分钟时自动开启补偿机制，知道某些实例被释放后就开始创建替换实例，创建完后立刻将 Spot 释放的实例替换掉，这样就不会出现图中左边断崖式释放。否则，在 Spot 实例释放完后再重新创建替换实例中间的时间周期会导致一定的抖动，而右边提前五分钟在 Spot 实例还没有释放完时新的实例已经创建出来，所以业务上不会出现抖动情况。需要注意的一点是使用 Spot 实例时一定要保证业务是无状态的。

6. 事件驱动

弹性伸缩提供的事件驱动能力，阿里在提供开放接口（OpenApi）的基础上又提供事件通知（Notification）和生命周期挂钩（Hook）的能力。通过这三种方式的结合可以让用户拥有更全面控制弹性伸缩组的能力。



关于事件通知和生命周期挂钩的职能，从图中可以看到扩容时需要创建资源、关联 RDS 白名单、启动中、运行中、挂载 SLB 并且最终加入到 ESC 弹性伸缩组中，而缩容时是从 ESC 弹性伸缩组挪出、取消 SLB 挂载、关机、取消 RDS 白名单、移除最终释放。

无论是扩容还是缩容最后执行结果时都会有事件通知，该事件通知可以通过事件订阅，例如 MNS 消息队列、CMS 云监控、OOC 运维编排方式订阅信息，然后将信息进行相关处理，可集成到自身系统中待资源成功时添加到系统当中，如果失败可进行自动化重试。

而生命周期挂钩在每一次扩容或缩容时会先发出消息，在用户接受到消息时可以判断当前这次扩容或缩容能否进行，例如创建实例后在挂载 SLB 之前会给用户发送消息，订阅消息后就可知道当前多少实例创建成功，并通过自身业务场景进行判断这些实例是否需要继续往下走，是否需要挂载 SLB，是否需要最终交付到弹性伸缩组中。如果不需要，可直接选择 Rollback，这些资源将会自动清理掉，如果需要就继续往下走最后添加到弹性伸缩组中。

缩容也是同样，当某些资源需要被释放时会将消息发送给用户，提示当前要释放哪些资源，用户在获取到消息后可以进行验证，看资源上的任务是否进行完毕，是否有数据需要进行同步和备份。

如果任务还未进行完毕时，用户可以选择等待或者直接取消，如果所有任务进行完毕且符合预期则可以继续往下走并进行释放。

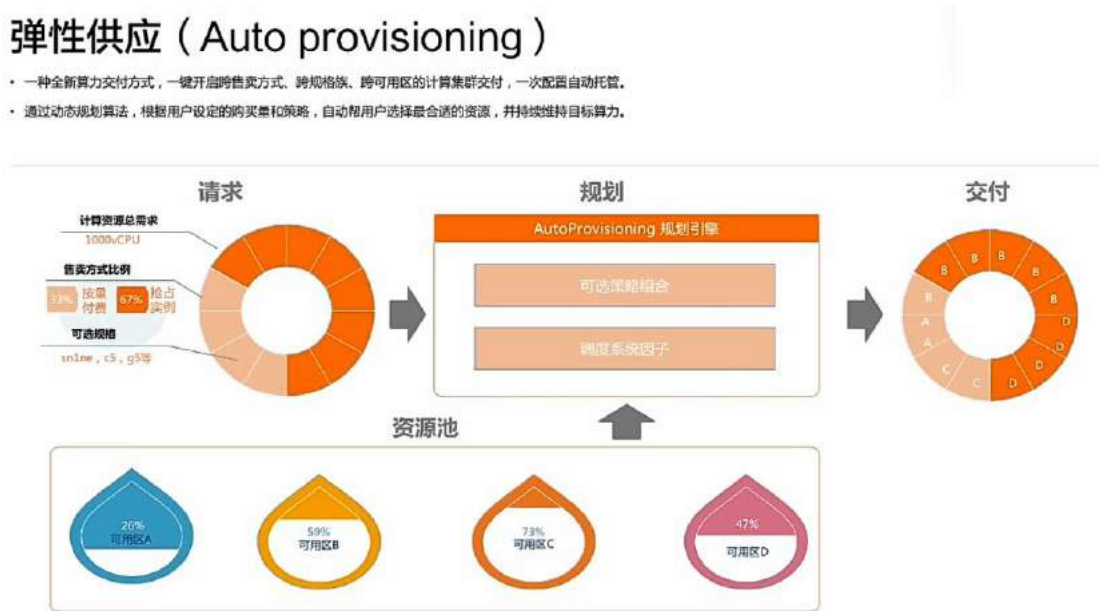
通过生命周期挂钩能力和事件通知能力可以让用户了解到一个事件中所有行为的全面感知，这样无论是自己做业务报警还是系统集成，都拥有更全面的控制能力。

三、弹性供应

1. 弹性供应（Auto provisioning）介绍

弹性供应是一种全新算力交付方式，可以整合包年包月、按量和 Spot 多种能力的付费类型实例，并且跨实例规格族、跨可用区的计算集群交付。

其优势在于相比以前交付 1000 资源，现在在弹性供应上交付 1000vCPU、交付 1000Memory 或者用户自身定义纬度应该交付的资源权重的样式，按照权重定义资源数量。



图中显示可定义当前伸缩组所有资源的比例，如 33%按量资源和 67%Spot 资源，这样用户可在整体成本上自行控制，在自身想要的成本情况下来满足自身业务需要。最终交付的成果如图 A、B、C、D，是多种实例规格多种可用区资源的整体结构。

2. 弹性供应的基本组件

弹性供应的基本组件

名词	概念
目标容量	指算力的总数量，单位可以是vCPU个数，也可以是实例个数
实例权重	指每个实例规格对目标容量的贡献度，权重越大，单台实例满足计算力需求的能力越大，所需的实例数量越小。 权重根据指定实例规格的计算力与集群单节点最低计算力得出，假设单节点最低算力为8 vCPU/60GB，则8 vCPU/60GB的实例权重为1，16 vCPU/120GB 实例规格权重为2，也可以将每个实例规格的权重与其vCPU数量保持一致
实例优先级	指交付算力时选择每种实例规格的先后顺序，优先创建优先级高的实例；与按量实例的优先级策略配合使用，0表示优先级最高，随着数字增大而降低
InstancePoolToUseCount	指在成本优化策略时，希望选择最便宜的实例规格数量。

目标容量：定义目标容量，可以定义 vCPU 的维度、实例维度或者其他需要的维度。

实例权重：为每个实例规格都定义自己的权重，如果是按照 vCPU 维度定义的权重，最终交付的资源整体容量是以 vCPU 为维度，如果是以用户自己业务产品计算力或是推出能力去定义，最终的效果就是以自己产品的权重容量交付，此能力最大的好处在于完全依照自身方式去定义期望的结果。

实例优先级：交付实例时可以定义每一种实例先后顺序，这样在创建时可以优先极高的实例。

Instance Pool To Use Count：指在成本优化策略时，可以选择最便宜的实例规格数量，进行一定程度的多种实例规格打散，在弹性收缩中使用一种 Spot 实例时有可能出现大面积的 Spot 资源全部清零的情况，然而现在可以用 Instance Pool 设定几种可接触的最低价的 Spot 实例数量，例如数量设定为三种时，资源创建的时候救可以选择最低价的三种 Spot 实例进行创建，这样即便出现一种 Spot 实例释放也不会导致整体业务受到损失。

3. 弹性供应的产品优势

超低成本：如果全部使用 Spot 实例交付时，最高可节省 90%的成本，同时也可设置全局和单个实例规格价格上限，这样用户可以完全保证低成本，如果超过上限阿里就会帮助取消创建的资格，只有满足用户定义的范围之内才可进行创建。

多种策略组合：可以支持所有的按量资源的实例交付策略或是具体成本的数量。

快速交付：单个供应组支持 20 种实例规格和多可用区部署，可分钟级快速交付 2000 实例。

智能打散：降低 Spot 被整体释放的风险，自动托管，分钟级巡检，动态保证集群的整体算力。

产品链接

如果用户对弹性伸缩或弹性供应相关产品有兴趣的话，可以点进链接查看，该产品是在云上解决希望用云上弹性降低整体成本的需求。

现在无论是电商或是视频类等许多用户都在使用该产品解决自身的业务问题，使用的效果也非常不错，可以以更低的价格保证自己的业务运行。尤其现在大部分用户都对成本有需求时，阿里也提供了很多种新功能，包括滚动升级等功能帮助用户在使用产品的过程中可以更好的融合到业务中。

因为资源交付置换的下一步就是如何进行应用发布等流程。用户后续如果有相关的问题可以在系统中提出，阿里也会在后续的具体细节中与用户进行沟通。

弹性伸缩

产品介绍页：https://www.aliyun.com/product/ecs/ess

产品文档：https://help.aliyun.com/document_detail/25857.html

产品控制台：https://essnew.console.aliyun.com/

弹性供应

产品文档：https://help.aliyun.com/document_detail/120020.html

产品控制台：https://ecs.console.aliyun.com/#/fleet/region/

3.2 基于弹性计算网络能力提升容器密度最佳实践

摘要：云原生和容器化是主流的趋势，实现容器化时推荐大家使用云厂商的容器服务，如阿里云 ACK。但由于部分用户因为一些原因需要自建容器，此时不得不面临一个问题，就是如何能够在一台宿主机上提升容器数量，降低容器成本。
本次分享由阿里云高级技术专家姜文锋(令吾)为大家介绍三种基于阿里云弹性计算网络能力提升容器密度的主要方法和最佳实践。

演讲嘉宾简介：姜文锋(令吾)，阿里云高级技术专家。来自阿里云弹性计算控制&体验团队，主要负责弹性计算（ECS）网络、安全组相关组件和产品的研发工作。

本次分享主要围绕以下四个方面：

一、弹性网卡直通
二、弹性网卡多 IP
三、弹性网卡中继
四、创建容器网络方案总结

云原生和容器化是主流的趋势，实现容器化时推荐大家使用云厂商的容器服务，如阿里云 ACK。但由于部分用户因为一些原因需要自建容器，此时不得不面临一个问题，就是如何能够在一台宿主机上提升容器数量，降低容器成本。

借此机会，本次分享由阿里云高级技术专家姜文锋(令吾)为大家介绍基于阿里云弹性计算网络能力，具体讲解三种构建容器的网络架构方案，同时横向对比各种方案在提升容器密度方面的优缺点。下面主要介绍三种构建容器的网络架构方案，分别是弹性网卡直通方案、弹性网卡多 IP 方案、弹性网卡中继方案。

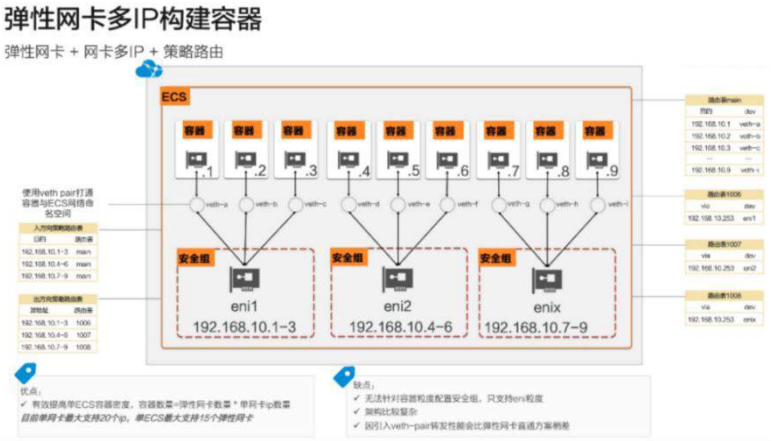
一、弹性网卡直通

所谓弹性网卡直通就是将一个 ECS 上的弹性网卡直接加入到容器命名空间内。这种架构非常简单，无技术风险，同时缺点也非常明显：ECS 容器数量受限于 ECS 可挂载弹性网卡数量。而目前阿里云最高规格实例大约支持 20 块网卡左右，对容器密度敏感的用户，这不是合适的选项。

二、弹性网卡多 IP 构建容器

1. 弹性网卡+网卡多 IP+策略路由

网卡多 IP 是阿里云为了提升 IP 数量而提供的方案，可以让一个弹性网卡支持多个 IP，这样就具备了在 ECS 上创建更多容器的基础。下图展示了结合弹性网卡+网卡多 IP+策略路由构建容器的架构。与弹性网卡直通对比下，引入的复杂度是路由设置问题。



由于弹性网卡直通方案中容器与网卡的比例为 1：1，路由极其简单，无需管理。而引入了网卡多 IP 之后，容器与网卡的比例关系是 n：1，需要正确的管理路由，实现入方向和出方向上容器与网卡正确的对应。

具体做法：首先使用 veth pair 打通容器与 ECS 网络命名空间，使得容器 IP 与 ECS 弹性网卡 IP 可见；之后进行策略路由的设置，实现容器与弹性网卡的正确对应。如下图所示，入方向策略路由表表明当访问 192.168.10.1-3 时需要查找 main 路由表进行包转发，main 路由表把不同的 ip 指向不同的 veth pair 设备，这样就可以路由到正确的容器内。出方向上如果源地址是 192.16.10.1-3 的话，则需要去 1006 路由表。路由表 1006 表示要将所有包路由到 eni1 上，从而找到正确的网卡。

整个方案的优点是可以有效提高单 ECS 容器密度，创建可观的容器数量。目前单网卡最大支持 20 个 ip，单 ECS 最大支持 15 个弹性网卡。

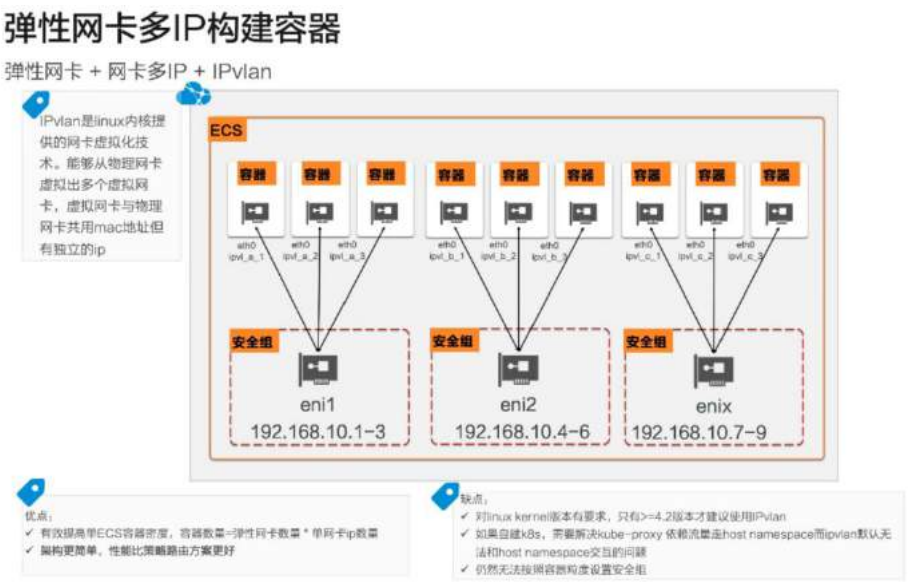
但弹性网卡多 IP 缺点主要有三点：首先，安全组是实现云安全的基本能力，而安全组只支持 eni 粒度，无法针对容器粒度配置安全组，这意味着下图中关联到同一个网卡的三个容器的安全配置完全相同，无法对每一个单独的容器做安全配置。另外，相对而言架构更加复杂。最后由于引入了 veth-pair，转发性能会比弹性网卡直通方案稍差。

2. 弹性网卡+网卡多 IP+IPvlan

基于弹性网卡多 IP 的方案还有更加简单的做法：IPvlan。

IPvlan 是 linux 内核提供的网卡虚拟化技术，能够从物理网卡虚拟出多个虚拟网卡。多个虚拟网卡有相同 mac 地址，但是有独立的 IP。

使用 IPvlan 之后，从容器往下看，ECS 便有了更多的网卡，如下图中 ECS 有三个弹性网卡，每个网卡 3 个 IP，每个 IP 对应一个 IPvlan 设备。将 IPvlan 设备直接放到容器的命名空间内，打通整个链路。



相对于弹性网卡+网卡多 IP+策略路由方案，IPvlan 方案架构更加简单，性能更好，同时可以起到提升容器密度的效果。但缺点同样有三点：首先对 linux kernal 版本有要求，只有大于等于 4.2 版本才建议使用 IPvlan。其次如果自建 k8s，需要解决 kube-proxy 依赖流量走 host namespace，而 IPvlan 默认无法和 host namespace 交互。还有依然无法支持容器粒度设置安全组。

三、弹性网卡中继构建容器

如果即要提升容器密度，又要求安全组支持容器粒度，那么阿里云推荐弹性网卡中继（EniTrunking）构建容器的方案。EniTrunking 是阿里云提供的一种提升单 ECS 挂载弹性网卡数量的技术。下面介绍 EniTrunking 中的几个主要概念：

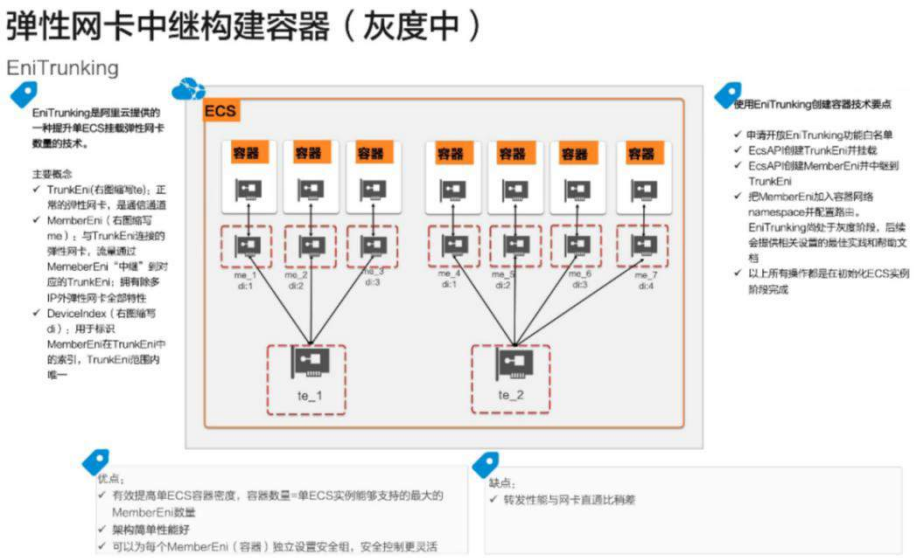
- 首先是 TrunkEni（如下面图中的 te_1，te_2），它是正常的弹性网卡，方案中充当通信通道的角色。
- 另外就是 MemberEni（如下图中的 me_i），是与 TrunkEni 连接的弹性网卡，流量通过 MemberEni“中继”到对应的 TrunkEni，拥有弹性网卡的绝大部分特性，但是不支持弹性网卡多 IP。
- 最后是 DeviceIndex，一个 TrunkEni 对应多个 MemberEni，DeviceIndex 是给 MemberEni 分配一个 Index，表示 TrunkEni 中的位置，在此范围内保持唯一。

基于 EniTrunking 的方案主要操作步骤如下：

- 首先申请开通 EniTrunking 功能白名单。
- 然后通过 ECS API 创建几个 TrunkEni 并挂载。
- ECS API 创建 MemberEni（具体数量参考官方文档），再将 MemberEni 中继到 TrunkEni 上。
- 把 MemberEni 加入到容器网络 namespace 中，并配置路由，从而实现整个链路的打通。

MemberEni 包含大部分的弹性网卡特性，这就使得弹性网卡中继构建容器方案可以有效提高单 ECS 容器密度以及容器数量，并可以为每个 MemberEni 设置独立的安全组，安全控制更灵活。对于某些安全网络产品使用弹性网卡中继构建容器方案是非常适合的。但同时由于中间多了一个环节，所以弹性网卡中继构建容器方案转发性能会比网卡直通方案差。

目前弹性网卡中继构建容器方案在灰度阶段，后续会提供相关设置的最佳实践和帮助文档，有兴趣的同学可以关注一下。



四、创建容器网络方案总结

前面具体介绍了 4 种创建容器网络方案，下面简单做一个总结：

1）**弹性网卡直通方案**的优点是非常简单，支持容器粒度安全组。但受限于 ECS 可挂载弹性网卡数量，导致容器成本较高。弹性网卡直通方案对企业自建或容器密度不是关键考虑因素的用户比较适用。

2）**弹性网卡多 IP+策略路由方案**可以有效提升容器密度，方案较为成熟，无技术风险。但它不支持容器粒度的安全组，而且由于引入了策略路由和 veth-pair 设备导致转发性能稍差。

3）**弹性网卡多 IP+IPvlan 方案**与弹性网卡多 IP+策略路由方案类似，但架构会更加简单，因为使用了 linux 原生的网卡技术。同样，弹性网卡多 IP+IPvlan 方案也不支持容器粒度安全组，对 linux 内核版本有一定的要求，用户需要自己解决 IPvlan 默认不走 host namespace 的问题。企业自建容器或容器服务的用户可以考虑这两种方案。

4）**弹性网卡中继方案**，最大优点是即可以提升容器密度，还支持容器级别安全组。只是转发性能会比弹性直通方案略差。

创建容器网络方案总结

网络方案	优点	缺点	适用场景
弹性网卡直通	✓ 架构简单 ✓ 支持容器级别安全组	✓ 容器密度受限于单ECS可挂载弹性网卡数量，容器成本高	✓ 企业自建容器&容器密度不是关键考虑因素
弹性网卡多IP + 策略路由	✓ 可有效提高容器密度 ✓ 方案成熟	✓ 不支持容器级别安全组 ✓ 因引入策略路由和veth-pair设备转发性能稍差	✓ 企业自建容器 ✓ 容器服务
弹性网卡多IP + IPvlan	✓ 可有效提高容器密度 ✓ 架构相比策略路由方案更简单	✓ 不支持容器级别安全组 ✓ IPvlan只有 4.2及更高linux 内核版本才稳定支持 ✓ 要自己解决IPvlan默认不走host namespace问题	✓ 企业自建容器 ✓ 容器服务
弹性网卡中继	✓ 可有效提高容器密度 ✓ 支持容器级别安全组	✓ 转发性能比网卡直通方案略差	✓ 网络产品 ✓ 企业自建容器

3.3 ECS 安全组最佳实践及新特性介绍

摘要：本次内容由阿里云智能技术专家王帝（丞浩）为大家介绍如何正确使用安全组、最佳实践以及新特性；详细了解安全组为何是云端的虚拟防火墙，以及为何是重要的网络隔离手段。

演讲嘉宾简介：王帝（丞浩），阿里云技术专家，2017 年 10 月加入阿里云弹性计算团队，主要负责网络安全组的优化和演进。

使用过 ECS 的朋友一定不会对安全组陌生，他是 ECS 实例的虚拟防火墙。配置安全组是创建 ECS 实例或者发生网络属性变更必不可少的一步。下面我就为大家分享一下安全组的相关内容。

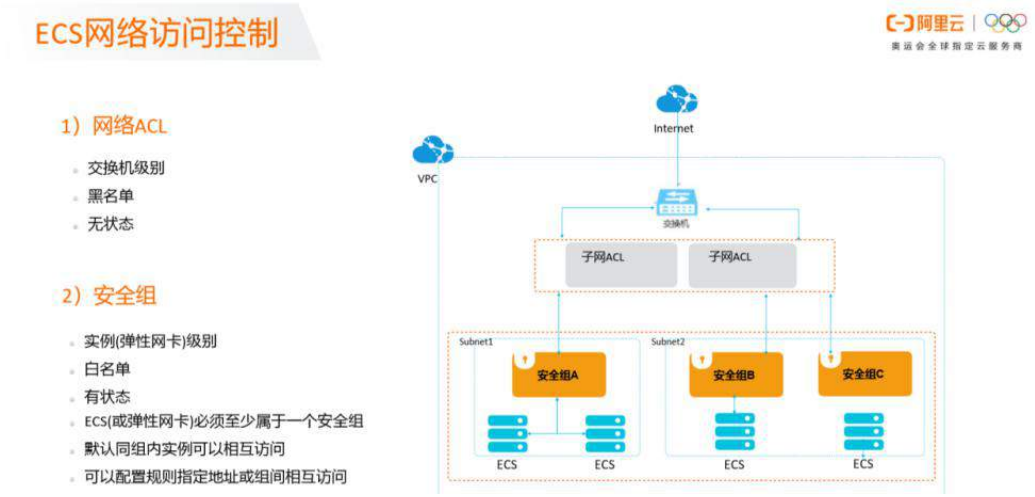
本次分享主要围绕以下三个方面：

一、安全组的简介
二、安全组的基本操作
三、最佳实践

一、安全组简介

ECS 网络访问控制

首先什么是安全组，阿里云 ECS 的网络访问控制，是由子网 ACL 和安全组两层实现的。大家知道阿里云提供 VPC 专有网络，是用户独有的云上私有网络。VPC 专用网络为用户独立出一块网络区域，使得用户可以自行规划自己的网段，在没有配置公网 IP 的情况下，VPC 是完全与外界隔离的。



交换机绑定网络 ACL，ACL 会对应一些控制规则，所有经过交换机的网络流量都需经过这些规则，一般配置的是黑名单规则(当然也支持指定白名单)，例如不允许哪些网端的数据包流入或流出。

再往下一层就是我们今天要讲的安全组，相对于子网 ACL 是生效在交换机上，安全组实例级别的防火墙，生效在 ECS 上面。所有经过用户的 ECS 网络流量都需经过安全组。

安全组是一种虚拟防火墙，具备状态检测和数据包过滤能力，用于在云端划分安全域。通过配置安全组规则，您可以控制安全组内 ECS 实例的入流量和出流量。

安全组是一个逻辑上的分组，由同一地域内具有相同安全保护需求并相互信任的实例组成。此外，安全组与子网 ACL 之间的明显区别是安全组具有状态，安全组会自动允许返回的数据流不受任何规则的影响，简单来说就是主动请求别人就一定会收到回包。而交换机则不是，入流量也会走一遍所设置的子网 ACL 规则，如果被拦截是收不到回包的。有些用户会将安全组与 iptables 对比，其实这两者是独立的。

阿里建议用户单独使用安全组，如果用户的场景需要配置 iptables，ECS 也是完全支持的。相对于子网 ACL 的黑名单方式，安全组一般是白名单。

ECS 或弹性网卡必须至少属于一个安全组，安全组组内默认互通可以配置规则控制网络联通。

二、安全组的基本操作

下面再给大家分享一下在阿里云 ECS 控制台是如何操作管理安全组的。

为了方便理解，我们把对安全组的操作暂时分为两类，组的操作和规则的操作。

组的操作是针对安全组本身的操作，比如创建、删除、改名字，以及可能导致安全组内 IP 发生变化的操作，还有组内添加实例网卡、替换组等。规则的操作则是改变组内规则的操作，比如添加，删除，修改，克隆等操作。

1）组的操作

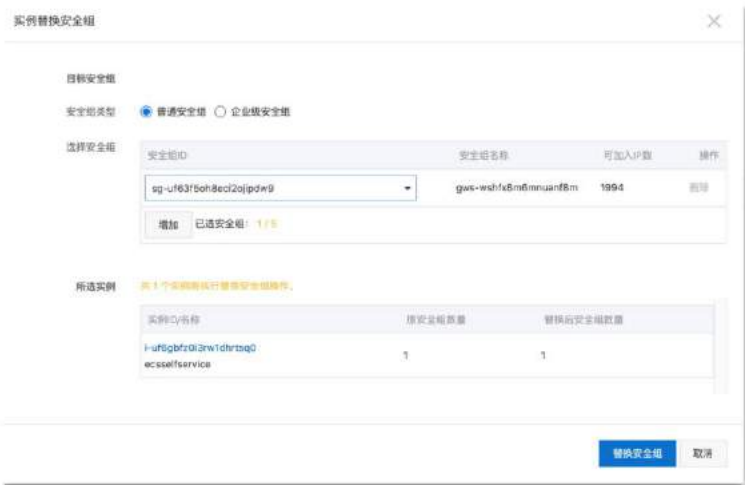
- 创建、修改、删除
- 组内添加、删除实例(弹性网卡)
- 替换组

2）规则的操作

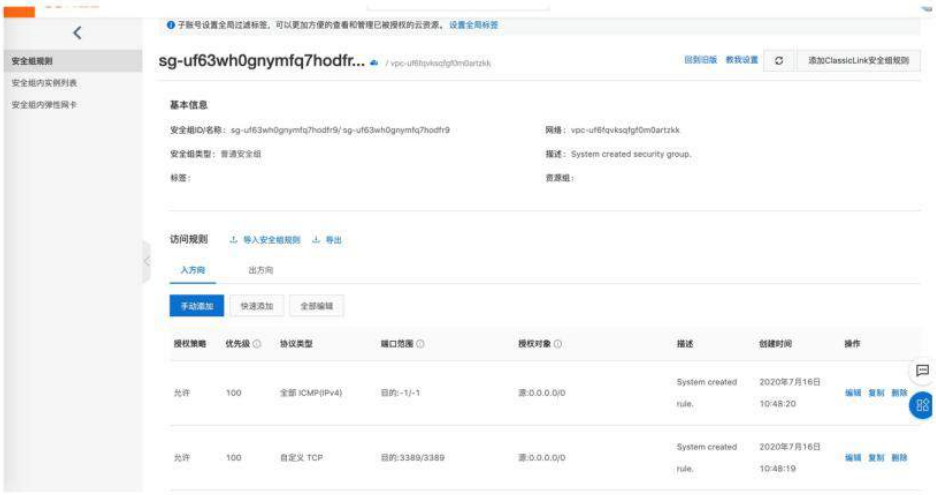
- 添加、修改、复制、删除
- 还原、导出导入
- ClassicLink
- 克隆组

先说组的操作，比如组内加减实例，网卡等操作比较简单，我们就不特别介绍了，重点介绍一下替换组。

替换组：实例可以从组 A 替换到组 B，在替换过程中不会发生网络闪断或抖动的情况，用户只需保证新老组的规则是兼容的，在整个替换过程中不会对网络有任何的影响和抖动。



再说规则的操作，我们重点介绍一下还原，导出导入，Classic Link 和克隆组。

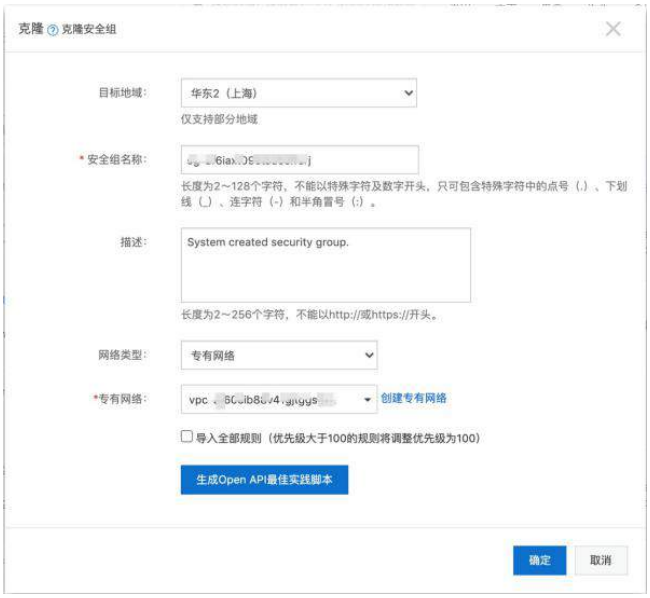


规则还原：对两组规则进行合并或替换，一般在做实验性质的网络变更之前可以先克隆出一个组，测试后可以通过规格还原功能恢复成原来的样子。下图为规格还原的页面，展示出哪些规格为新增哪些为删除。

导出导入：将安全组下载成 JSON 文件或是 CSV 文件用于备份。

Classic Link：通过添加一条安全组规则实现 VPC 和经典网络之间的网络联通。

克隆组：克隆组支持跨地域或跨网络类型的安全组复制，可以从经典网络到 VPC 或是到一个新 Regen 并快速复制一个组。



三、最佳实践

最后再给大家介绍一下比较好的安全组配置实践，比如如何合理的配置规则，如何使用五元组，如何基于安全组做断网演练。

安全组规则格式

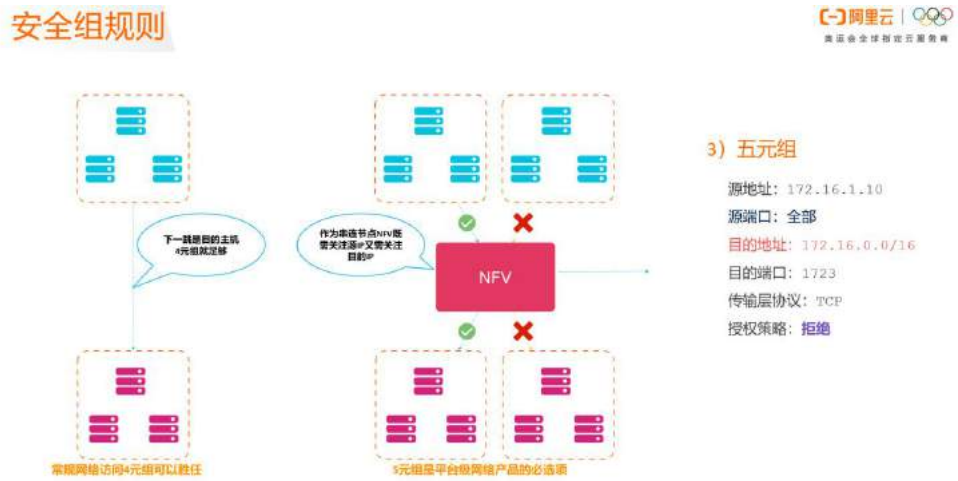
首先先介绍一下安全组规则配置有两种方式，

CIDR：图中示例展示，授权 192.168.0.0/24 的地址 DR 机器访问 22 端口。

组织授权：图中示例展示，拒绝另一组访问的所有端口，完全切断两组之间的流量。

以上两个示例都为入方向规则，一般情况下不知道对方使用哪个端口接连自己，所以不限制，区别在于自己任选哪个端口对外暴露给任意对象。

相对于上图的四元组，阿里也支持五元组。



五元组为五个参数：源地址、源端口、目的地址、目的端口、传输层协议，相比四元组多了目的地址。以入方向规则为例，使用五元组可以实现不放行整个组，可单独放行某一个 IP 段，这样某些平台内网络服务为了防范第三方产品对用户 ECS 的实例发起非法访问，

需要在安全组内设置五元组规则，更精确的控制出和入的流量。另外，如果用户组内联通策略是拒绝场景，想精确控制组内 ECS 之间的联通策略也需要使用五元组。五元组场景较为特殊，而绝大多数场景四元组是可以胜任的。

规则配置建议

先规划对于分布式应用来说，不同的应用类型应放到不同组中。使用白名单方式管理安全组，不建议用户使用先加一条低优先级全通，再逐条拒绝的方式。要慎用 0.0.0.0/0 全通策略。遵守规则最小化配置原则。尽量使用 CIDR 段，因为单组容量有限，而且 CIDR 地址段更容易扩展和维护。不限制协议使用 all，不是每个协议都配一遍。安全组规则变更非常高危，要认真写好备注以便于后续的维护。

潜在高危安全组概览

阿里云 ECS 会定期检查用户的实例，如果实例暴露在公共环境并且开放高危端口，阿里会对用户做出预警，且用户在资源概览页面中可以查看自己的高危安全组。

如何给应用划分安全组

为了避免测试环境和正式环境之间互相干扰，阿里会将测试环境和正式环境放在不同 VPC 中进行隔离。将有公网服务放在一组内网服务放在一组。不同应用类型应使用不同安全组，例如 Web 服务、应用服务、数据库或缓存，都应该放在不同安全组中。下图中的示例，由于都需使用跳板机，所以都授权了跳板机组 G1，Web 服务器需要联通应用服务器，应用服务器又需要联通数据库，所以分别做了组组授权。这样做完网络安全组的规划使得应用分层清晰，便于后续的维护，同时也满足了隔离性和安全性的要求。

使用安全组进行断网演练

基于安全组可进行断网演练用于高可用容灾或混沌工程等场景，如下图中 case，演练数据挂掉时系统的表现，可以将此 DB 加入专门的断网组，断网组+全阻断规格来实现快速规模化断网。

拆解断网过程

创建断网组：因为组内默认联通策略是组内互通的，所以先改为组内不互通。

加入实例：先将演练数据库加入断网组内，这时对业务无任何影响且正常运行。

添加规则：当真正进行断网演练时需要执行该步骤，给断网组出和入各加一条全阻断规则，加完后此时服务器的流量就会完全被切断。

删除规则：当演练完毕后，需要将全阻断规则删除，即可恢复业务。

如果需要不定期做容灾演练，只需重复步骤三、四即可。

企业级安全组

传统安全组组内容量上限是 2000IP，如要进行更大规模则需使用企业组，企业组支持单组 60000 以上的 IP，未来支持的容量会更多。

典型的企业组场景例如阿里云的容器服务 ACK 或是用户自建 K8s 集群，其实 ECS 只是作为 S 层，ECS 之间只需网络互通即可。容器的网络访问控制并不是安全组实现的，而是通过 Network Policy 等方式来实现的。下图中的示例显示是较为常见的部署方式，VPC 下挂两个虚拟交换机分别在两个可用区做跨地域容灾，将所有实例放入一个组中，无需复杂的规则配置，只需配置内网全通和出方向全通就可实现 VPC 内互通。下图中只配置了三条规则，两条入方向全通和一条出方向全通，这是较为常见的容器服务安全组架构。如果用户不需要组组授权并且对组内的实例规模有要求，那么请使用企业级安全组。如果当前传统组想切换至企业组，有两种方式可实现，第一种新建一个企业组后采用替换组的方式将实例逐个挪入到企业组中，第二种是阿里协助用户将原来的传统组升级成企业组。

云产品托管安全组

用户在使用阿里云云产品时，如云防火墙、NAT 网关等，云产品都会替用户创建安全组，为了避免用户误操作造成产品不可用，阿里采用了托管模式，托管组即用户可建不可操作，避免产生问题。

以上是对本次分享的全部，希望对大家有所帮助，谢谢各位。

3.4 Region 化部署和跨可用区容灾介绍

摘要：本次分享由阿里云弹性计算架构负责人李钟（谢顿）为大家介绍阿里云 region 化部署和跨可用区容灾的实践经验，说明多 Region 部署场景中使用阿里云弹性计算的最佳实践，并结合弹性计算的实践经验探讨如何基于阿里云多可用区实现跨地域容灾。

演讲嘉宾简介：李钟(谢顿)，阿里云智能弹性计算高级技术专家。2015 年 7 月加入阿里云弹性计算团队，目前负责阿里云弹性计算管控架构团队，主导弹性计算管控系统架构的优化和演进。

本次分享主要围绕以下三个方面：

一、Region 和可用区介绍
二、Endpoints 和资源作用域
三、多可用区容灾和 Region 化部署

弹性计算 Region 化部署和跨可用区容灾本身是非常复杂的课题，本次分享由阿里云弹性计算架构负责人李钟（谢顿）为大家介绍如何选择 Region，同时结合阿里云在 Region 化部署和跨可用区容灾的实践经验，分享多 region 部署场景中如何使用阿里云弹性计算的最佳实践，并结合弹性计算的实践经验探讨如何基于阿里云多可用区实现跨地域容灾。

一、Region 和可用区介绍

该部分主要介绍 Region 和可用区，包括阿里云建设 Region 和可用区的原因、划分等。

1. Region（地域）与 Available Zone（可用区）

Region（地域）：根据定义，Region 是相互独立的地理区域，Region 中包含多个 Available Zone。可通过阿里云 Region 和可用区分布图直观了解。

Available Zone（可用区）： Available Zone 是 Region 内网络和电力相互独立的区域，具有两个主要特点。第一，Available Zone 网络和电力相互独立，具有故障隔离能力。当一个可用区内部网络、电力出现问题时，不会影响其他可用区。第二，相同 Region 内的可用区之间内网互通，通过高速网络连接，网络延迟低（ms 级）。

即可用区之间有故障隔离，同时相同 Region 内的可用区之间网络延迟较低，可以将多个可用区视为整体提供服务，也为后续做跨可用区容灾提供了基础。

此处需要指明，并不是先规定了地域和可用区后才发现可支持高可用服务建设，而是在有高可用服务需求的基础上通过 Region 与可用区方式进行了资源划分。

2. 如何选择 Region?

选择一个 Region 有以下三点重要影响因素：

第一，地域选择需要符合相应政策和法律合规性需要。例如向美国提供服务，要求对应数据与服务器资源必须在该国家内部，那么根据下方 Region 与可用区分布，只能申请美西、美东两处资源。

第二，需要根据所需云产品在各个地域的功能开通情况和 SLA 选择合适区域。

第三，一般选择距离用户更近且网络延迟更低的地域，保证用户快速接入。

虽然 Region 的选择看似条款化，但是实际选择时是更为简单直观的，较易得到最优选择。

3. 阿里云 Region 与可用区分布

目前为止，阿里云公共云在全球建设 21 个地域 63 个可用区，资源正在快速增长中，未来将支持更多形态。

下图所示体现了阿里云在全球多个地理位置提供服务的能力，可见 Region 是一个地理概念。例如当用户在华北 2（北京）Region 购买资源时，对应的 ECS、RDS 产品的确是在该地域范围内。

目前 Region 与可用区分布较集中在中国，其他国家地区也有分布，在南美与非洲缺少 Region 建设，是后续的发展方向。

而 Region 的使用稍显复杂。与各个 Region 交会并不直接使用其名字，而是通过 REGION_ID。REGION_ID 看似杂乱，其实具有一定特点，可大致分为以下三种：

中国 REGION_ID 为 cn-（城市名），如 cn-hangzhou。
大面积国家 REGION_ID 为（国家代码）-（方位）-（编号），如 USA-EAST-2。
其他地区 REGION_ID 为（地区）-（方位）-（编号）。

二、Endpoints 和资源作用域

该部分介绍使用 Region 与可用区时需要关注的两个关键点：Endpoints 和资源作用域。

1. Endpoints

Endpoints 是调用 ECS API 的接入地址。

调用 ECS API 首先需要获取 Endpoints，即需要知道需要调用的地域的域名，才能调用到相应服务。第二需要有 AK，通过某种阿里云特定的算法对调用串做一些加签。如此便可以完成一次请求。

如下表所示，根据不同 Region 类型，Endpoints 接入方式有如下三种：

第一种类型是中心域名 Region，其接入点域名地址是 ecs.aliyuns.com。此类老地域使用兼容模式只能通过中心域名接入，如 cn-beijing。

第二种类型是 Region 化域名，其接入点为 Region 化域名地址 ecs.${REGION_ID}.aliyuns.com。此类 Region 使用 Region 化域名调用，兼容中心域名接入，如 cn-zhangjiakou。

第三种类型开始考虑不兼容中心域名接入的 Region 化域名，只能通过 Region 化域名地址 ecs.${REGION_ID}.aliyuns.com 接入，如 cn-heyuan。

Region类型	接入点	是否可以使用 ecs.aliyuncs.com	说明
中心域名Region	ecs.aliyuncs.com	●	使用兼容模式, 通过中心域名接入 (cn-beijing, cn-shanghai, cn-hangzhou等)
Region化域名 (兼容中心域名)	ecs.${REGION_ID}.aliyuncs.com	●	Region化域名, 兼容中心域名接入方式 (cn-zhangjiakou, eu-central-1等)
Region化域名 (不兼容中心域名)	ecs.${REGION_ID}.aliyuncs.com	●	Region化域名, 不兼容中心域名 (cn-heyuan, cn-wulanchabu等)

- 非中心域名的地域, 建议使用Region化域名地址 ecs.${REGION_ID}.aliyuncs.com.
- 部分region只能使用中心域名访问, 后续会逐渐切换为Region化域名.

针对以上三种类型 Region 的接入得到了最佳实践。非中心域名的地域，建议使用 Region 化域名地址 ecs.${REGION_ID}.aliyuns.com。部分 Region 只能使用中心域名访问，后续会之间切换为 Region 化域名。

2. 资源作用域

使用 ECS 过程中会有多种资源，每一种资源有其特定的作用域。

下表罗列了部分资源。如账号、角色、AK 此类为全局性信息，所有 Region 共享。实例、云盘为可用区级别的概念，创建实例、云盘都需要选择一个特定的可用区，但是实例只能挂载相同可用区的云盘。快照、镜像资源、keypair、安全组的作用域均为 Region，

在 Region 内各个可用区均可访问。当需要跨 Region 使用快照、镜像资源时，需要先进行资源复制操作，在另一个 Region 上形成新资源才可以使用。

资源	全局性	Region	可用区	备注
RAM 账号, 角色, AK	•			账号, 角色, AK等信息都是全局的, 可以在任何地域使用
实例			•	创建实例需要选择一个特定的可用区
云盘			•	创建云盘需要选择可用区, 实例只能挂载相同可用区的云盘.
快照		•		镜像和快照的作用域都是region, 当需要跨region使用时, 需要先进行资源复制操作.
镜像		•		
keypair		•		keypair的作用域是region.
安全组		•		安全组的作用域是region, 可以选择region内的实例加入安全组.

资源作用域并非规定成章，有时会存在变化，大部分时期是上述情况。

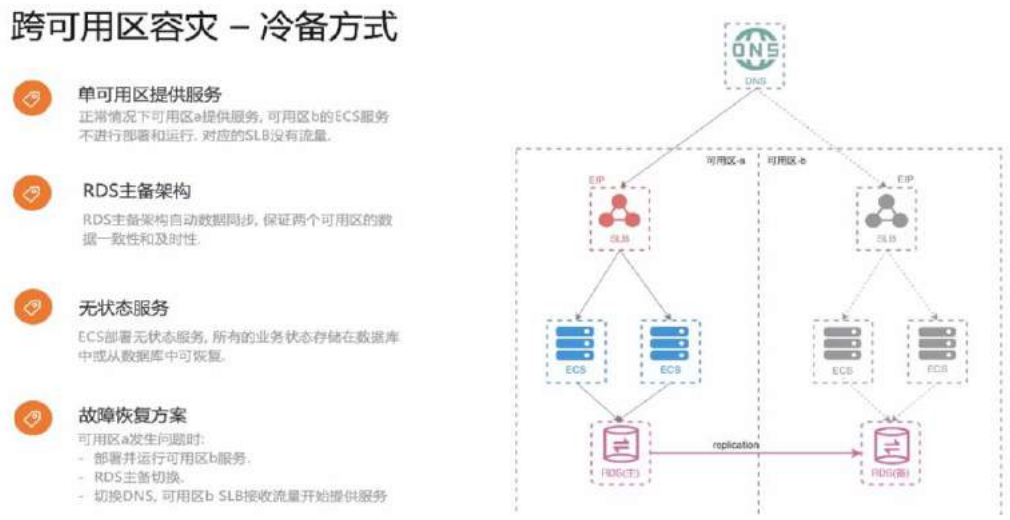
三、多可用区容灾和 Region 化部署

该部分介绍如何利用 Region 与可用区地域分布以及可用区电力、网络故障隔离和低延迟的特点实现多可用区容灾和 Region 化部署。

1. 跨可用区容灾——冷备方式

冷备方式指两个可用区，一个主可用区运行，一个不运行作为备份。一旦运行中的可用区服务故障，部署并运行备份可用区。冷备方式较易实现，其缺点为主备切换耗时大（分钟级）。

单可用区提供服务：如下图所示，可用区 a 和可用区 b 同时提供两套服务，由单可用区提供服务。正常情况下可用区 a 提供服务，可用区 b 的 ECS 服务不进行部署和运行，对应的 SLB 没有流量。流量从 DNS 解析到可用区 a 的 SLB，接入请求后 SLB 会分发到某个 ECS。



RDS 主备架构：两个 ECS 访问同一个 RDS，RDS 采用主备结构，主 RDS 提供读写服务，同时有一条路径自动同步数据到备 RDS，保证两个可用区的数据一致性与及时性，为容灾切换提供基础。

无状态服务：ECS 部署无状态服务，所有业务状态存储在数据库中或可从数据库恢复。如此才能在可用区 a 故障时将整个业务切换到可用区 b。

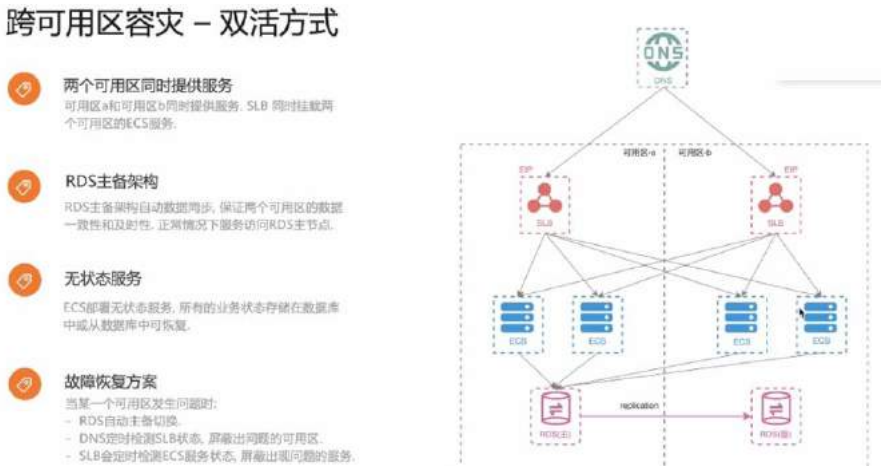
故障恢复方案：可用区 a、b 存在电力、网络的物理隔离，延迟低，发生可用区级别故障时可切换可用区提供服务。可用区 a 发生问题时，部署并运行可用区 b 服务，RDS 主备切换，可用区 a 恢复后可继续进行数据同步。切换 DNS，可用区 b SLB 接收流量开始提供服务。

劣势：冷备方式比单纯的单个可用区提供服务有更强的容灾能力，但是其缺点是切换可用区启动 ECS、启动服务等、运行 SLB 等等操作耗费一定时间，属于分钟级别的容灾。

2. 跨可用区容灾——双活方式

双活容灾方式与冷备方式区别在于两个可用区同时运行，数据同步。一旦某一可用区服务故障，另一可用区继续提供服务。双活方式同样易于实现，且主备切换耗时低（s 级）。

双可用区同时提供服务：如下图所示，有可用区 a 和可用区 b 同时提供两套服务。SLB 同时挂载两个可用区的 ECS 服务，每个可用区的 SLB 需要将请求分发给两个可用区的 ECS。



RDS 主备架构：自动同步数据，保证两个可用区的数据一致性与及时性。正常情况下两可用区服务均访问 RDS 主节点。

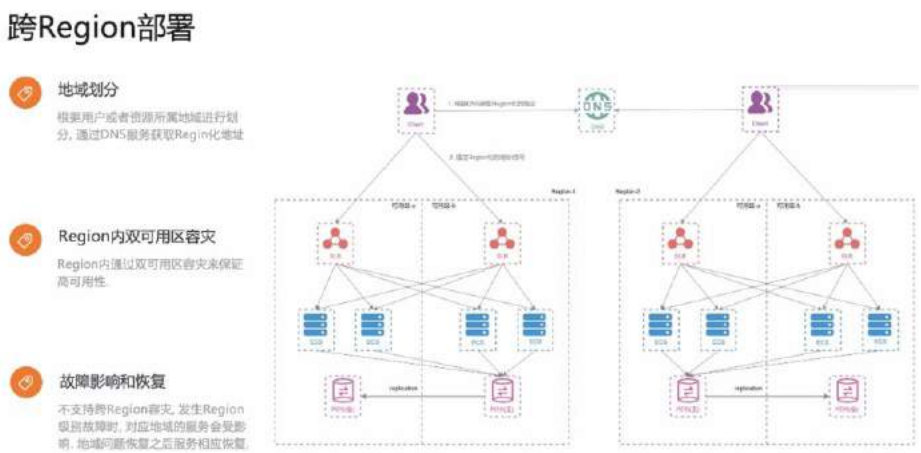
无状态服务：ECS 部署无状态服务，所有业务状态存储在数据库中或可从数据库恢复。

故障恢复方案：当一个可用区发生问题时，RDS 可自动主备切换。DNS 定时检测 SLB 状态，屏蔽故障可用区，SLB 定时检测 ECS 服务状态，屏蔽故障服务。该场景下大部分检测自动实现，部分复杂场景下可能需要手动确认故障，但是该切换过程迅速，可实现秒级切换。

以上两种跨可用区容灾方式较为实用、常见，实现条件均是可用区之间具备网络、电力的物理隔离以及低延迟特点。

3. 跨 Region 部署

此处仅介绍较为简单的跨 Region 部署模式。



首先进行地域划分。Region 化部署完成后，每个 Region 中由可用区容灾的集群保证高可用性。但是客户端调用地域时一个 Region 不能支持所有用户与资源访问。因此首先需要根据用户或者资源所属地域进行划分，通过 DNS 服务获取 Region 化地址。该 DNS 可能是阿里云提供的服务或其他服务，DNS 根据用户或资源返回相应的 Region 化地址。

例如某用户访问 Region 1，则调用 Region 1 的服务提供相应支持，某用户访问 Region 2，则调用 Region 2 的服务提供相应支持。

Region 内双可用区容灾。地域划分完成后，当可用区级别问题扩大到 Region 级别问题场景下，并未提供 Region 级别容灾能力，但是提供了隔离能力。例如当 Region1 发生问题，无法切换到 Region2 进行容灾，但是可以实现隔离，当 Region 1 出现问题时 Region 2 不受影响，即能够保存一部分工作服务的能力。

若需要提供 Region 级别容灾能力，需要实现跨 Region 的数据同步或复制，将一个 Region 的数据、用户状态等复制到另一个 Region，即可实现 Region 切换。如此将更加复杂。Region 级别容灾等复杂方案此处不做介绍，有兴趣者可自行查阅资料。

故障影响和恢复。用户最终选择的容灾方式、故障隔离级别、故障恢复方案等需要根据自身业务需求、实现难度与故障概率等进行判断，选择最适合自身业务的方案。

综上所述，本次分享介绍了 Region、可用区的定义、故障隔离与低延迟特性、关键点以及跨可用区容灾等基本概念。大家可以利用阿里云的基础架构以及业务相应支持提高服务可用性，在一定级别实现容灾，提供更好的服务。

感兴趣的同学可通过学习更加具体的场景下不同跨可用区容灾方案，更加深入了解跨 Region 容灾。

扫码查看“玩转 ECS”详情页
解锁更多课程视频

关注百晓生，笑谈云计算

阿里云开发者“藏经阁”
海量免费电子书下载