

阿里云云原生容器ACP认证培训

云原生容器技术概要



课程目标

学习完本课程后，你将能够：

1. 掌握云原生的基础概念及价值
2. 掌握云原生与容器技术发展历程
3. 掌握容器技术Docker基础知识
4. 掌握容器编排技术Kubernetes基础知识



课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
4. 容器编排技术基础-Kubernetes

云原生成为下一代云计算的技术“内核”

- 阿里巴巴自从2011年迈入容器大门，到2019年双11阿里电商核心系统100%上云。并部署了超过200万个容器，成为全球最大规模的云原生实践
- 阿里巴巴成立云原生技术委员会，云原生升级为阿里技术新战略
- 阿里云正在重新定义下一个十年的云计算技术——云原生



阿里巴巴核心系统全面云原生化

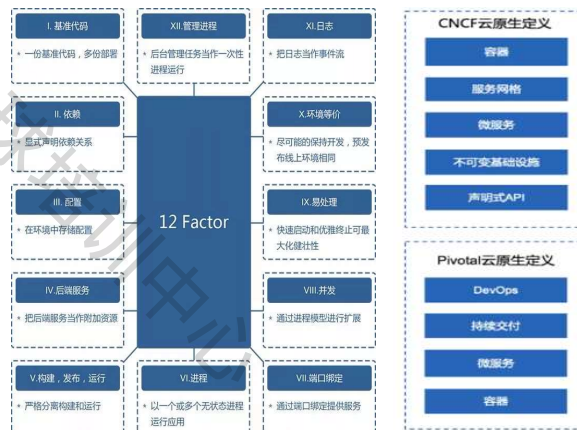
云计算的下一站，就是云原生

IT 架构的下一站，就是云原生架构

不断更新的云原生定义

• 很多人都会问“到底什么是云原生？”

- Heroku 于 2011 年提出了**十二因子**的应用定义，该定义可以适用于任何编程语言，通常被认为是最早的云原生应用的技术特征。
- Pivotal 于 2015 年提出了“Cloud Native”的概念：**云原生**是一种可以充分利用云计算优势的构建和运行应用的方式。
- 云原生计算基金会（CNCf）关于云原生定义的最新版本是：**云原生的代表技术包括容器、服务网格、微服务、不可变基础设施和声明式API。**



阿里巴巴对云原生的定义

• 云原生，是一条最佳路径

- 云原生是一条使用户能：
 - 低心智负担的、
 - 敏捷的、
 - 以可扩展、可复制的方式，
- 最大的利用“云”的能力，发挥“云”的价值的最佳路径

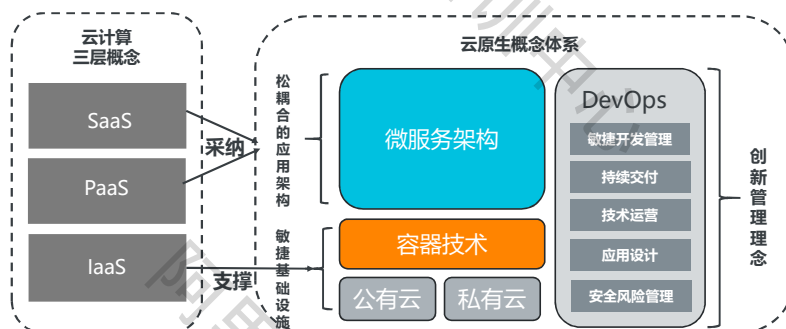


• 云原生的愿景

- 软件从诞生起就“生在云上、长在云上”、全新的软件开发、发布和运维模式

云原生是种架构模式及软件开发的新的思想理念

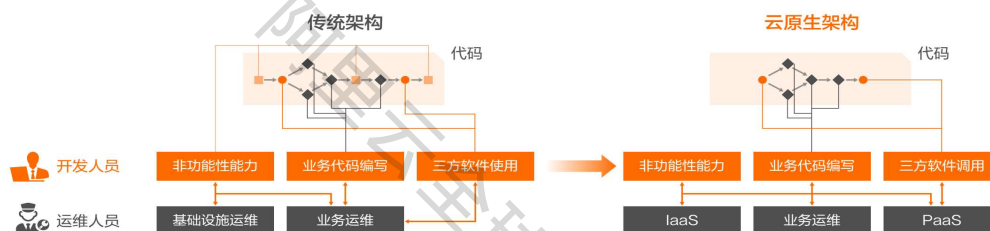
- 云原生基于云计算理念的深化，是面向云应用设计的一种新的架构设计理念
- 充分发挥云效能的最佳实践路径
- 帮助企业构建弹性可靠、松耦合、易管理可观测的应用系统
- 提升交付效率，降低运维复杂度



6

云原生架构与传统架构对比

- 传统软件架构，用户不仅仅需要关注基础设施的能力及其运维，还需要采购及维护大量的第三方软件以及非功能性能力，降低软件开发效率，并且没有充分利用云计算IaaS与PaaS的能力。



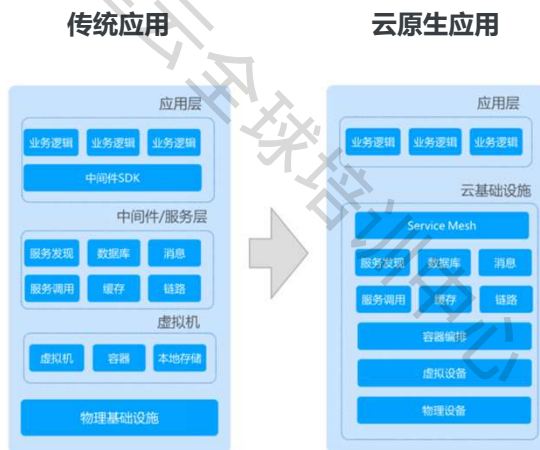
云原生架构与传统架构的对比

- 从技术的角度，云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化的剥离，从而让云设施接管应用中原有的大量非功能特性（如弹性、韧性、安全、可观测性、灰度等），使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。

7

云原生技术改变了软件开发及运维模式

- 开发者需要关注从基础设施到应用层所有细节
- 资源无法发挥弹性效能
- 技术栈技术标准不统一
- 系统缺乏敏捷，交付周期长
- 数据不成体系，无关联性
- 系统开放性差，系统交互复杂



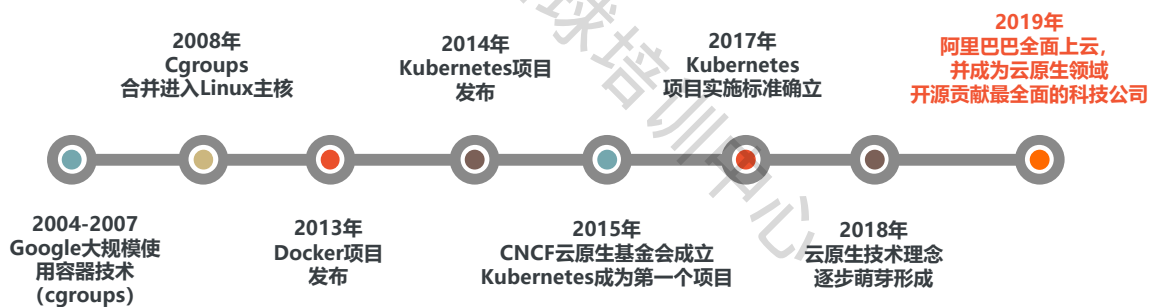
- 开发者仅需关注应用业务逻辑
- 资源最大限度弹性伸缩
- 业界标准统一技术栈
- 持续交付、敏捷开发
- 数据规范性好
- 统一标准API，开放服务

课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
4. 容器编排技术基础-Kubernetes

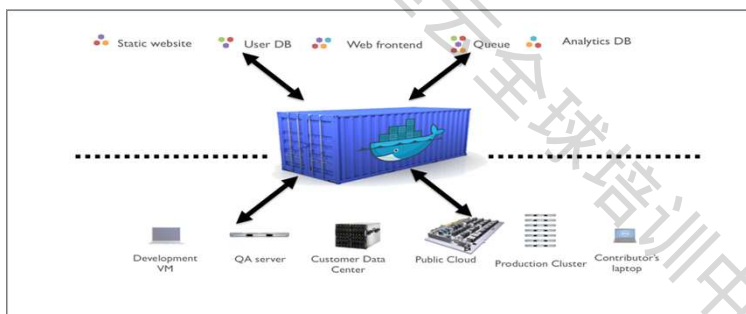
云原生与容器技术发展简史

- 聊容器技术避不开云原生，聊云原生也避不开容器技术。容器技术和云原生就是一对双螺旋体，容器技术催生了云原生思潮，云原生生态推动了容器技术发展。



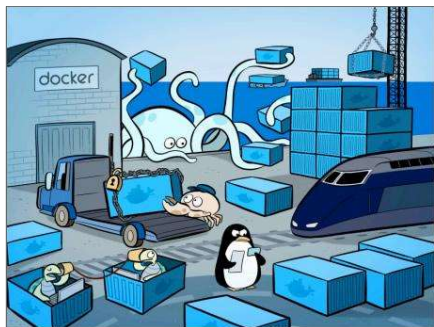
容器技术Docker重塑软件交付流程，成为云原生生态的基石

- 容器技术已经为了云原生时代软件开发和运维的标准基础设施，使用容器是迈向云原生开发的重要第一步。
- 容器技术将集装箱的思想应用到了软件的打包和部署上，采用标准化的方式来处理开发和运维间的交付改变了软件开发及交付方式。



- 容器技术Docker可以将任何应用及其依赖环境打包为一个轻量级、可移植、自包含的独立运行环境
- 容器可以运行在几乎所有的操作系统之上

容器技术带来的价值



- 容器是轻量级的
- 容器是易于移植的
- 容器可以实现秒级部署
- 容器是互相隔离的

12

容器技术推动容器编排技术的崛起

- 随着容器技术的快速成熟与推广，容器技术解决了开发人员的生产力问题，使 DevOps 工作流变得异常流畅。
- 开发人员可以创建容器镜像，运行容器并在该容器中开发代码，再将其部署在本地数据中心或公共云环境中。



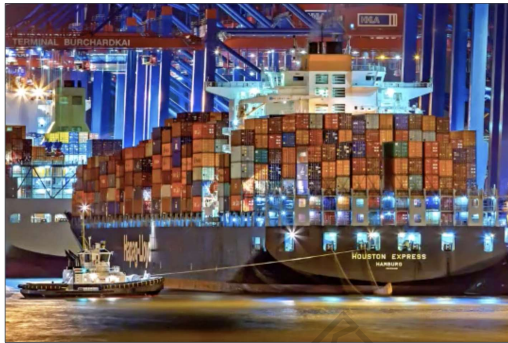
- 越来越多的企业开始使用容器，并把企业自身的生产业务封装进容器，导致**容器管理和编排遇到困难**
- 在生产环境中，会涉及到多个容器；这些容器必须**跨多个服务器主机进行部署**。
- 用户需要容器实施分组管理，以便跨所有容器提供**网络、安全、监控等服务**。

13

容器编排带来的价值

- 容器编排

- 指自动化容器的部署、管理、扩展和联网
- 可以为需要部署和管理成百上千个容器和主机的企业提供便利



- 大规模容器自动化部署
- 敏捷高效的资源调度
- 强大的弹性伸缩能力
- 容器服务自愈能力
- 自动化的服务发现

14

阿里云

随着云原生的发展逐步形成了云原生技术事实标准



容器化

应用与运行环境解耦
构建、分发的标准



编排调度

资源管理与基础设施解耦
工作负载交付、运维的标准

DevOps

微服务治理

全链路可观测

云原生

云原生可观测性、
微服务治理、
DevOps应用架构与
开发方式演进的标准

15

阿里云

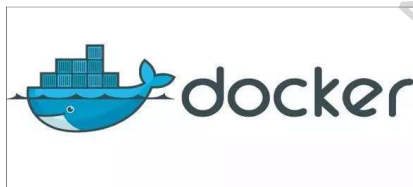
课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
 - 3.1 Docker基础概念
 - 3.2 Docker架构与使用
4. 容器编排技术基础-Kubernetes

Docker容器基础概念

• Docker 容器

- 是资源分割和调度的基本单位
- 封装整个服务的运行时环境，用于构建、发布和运行分布式应用的一个框架
- 是一个跨平台、可移植并且简单易用的容器解决方案



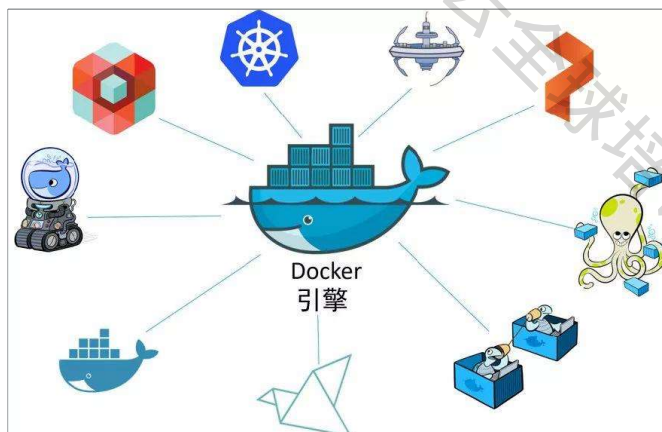
• Docker是世界领先的软件容器平台。

• Docker使用Google公司推出的Go语言进行开发实现，基于Linux内核的cgroup, namespace, 对进程进行封装隔离。

• Docker能够自动执行重复性任务，例如搭建和配置开发环境，从而解放了开发人员以便他们专注在真正重要的事情上：构建杰出的软件。

• 用户可以方便地创建和使用容器，把自己的应用放入容器。容器还可以进行版本管理、复制、分享、修改，就像管理普通的代码一样。

为什么Docker成了容器的事实标准？



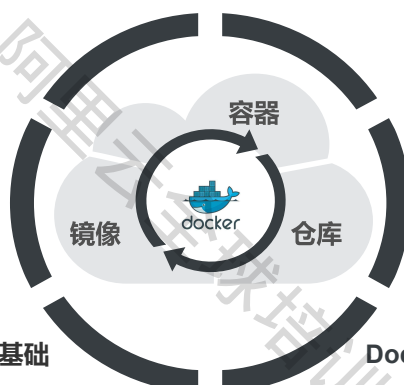
- 一致的运行环境
- 秒级启动，更快速的启动时间
- 资源隔离，容器之间相互隔离不受影响
- 弹性伸缩、快速扩展
- 迁移方便，无需担心运营环境
- 持续交付和部署，提升交付效率

18

阿里云

Docker的三大核心概念

容器是从镜像创建的运行实例



容器镜像为标准化打包提供了基础

Docker仓库是集中存放镜像文件的场所

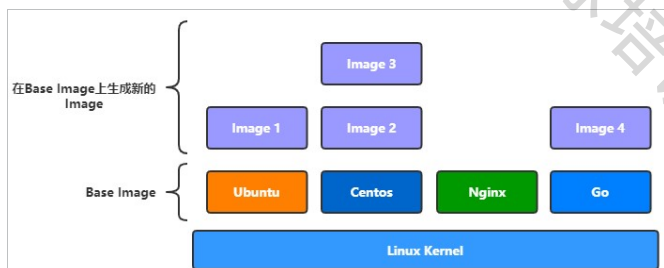
19

阿里云

容器镜像-容器封装的标准基础

• Docker镜像

- 是一个Linux的文件系统
- 包含运行的程序以及程序、库、资源、配置等数据



• Docker 镜像：

- 类似于一个只读压缩包
- 用于创建容器
- 分层构建，前一层时后一层的基础
- 包含了一个精简的操作系统、应用运行所必须的文件和依赖包
- 一次构建，到处运行

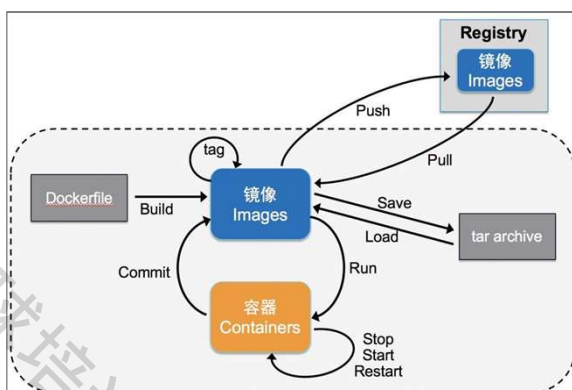
• 镜像构建（Build）的方法：

- docker commit 生成镜像
- Dockerfile（建议）

容器-镜像创建运行实例进程

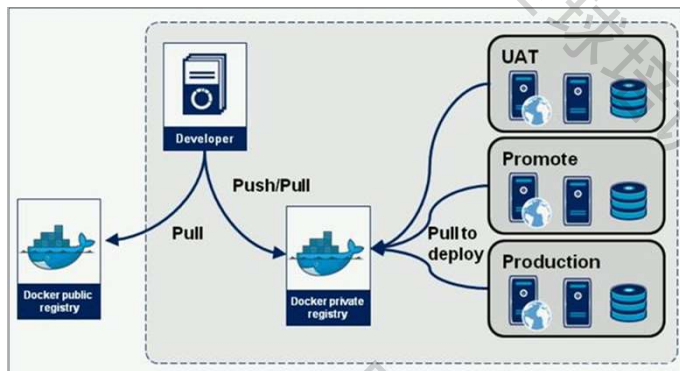
• Docker中的容器和镜像有什么区别？

- 容器是镜像的运行实体
- 容器可以被创建、启动、停止、删除、暂停等
- 容器运行特定应用
- 容器之间相互隔离
- 容器层是多层只读镜像之上运行的可写层
- 容器停止/删除，可写层数据丢失
- 数据持久化：使用宿主存储/网络存储



镜像仓库-镜像文件存储的场所

- Docker 镜像仓库类似于代码仓库，它是 Docker 集中存放镜像文件的场所。

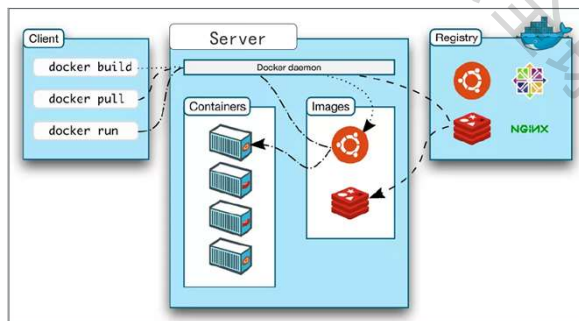


- 实现Docker镜像的全局存储
- 提供API接口
- 提供Docker镜像的下载/推送/查询
 - docker pull 镜像下载
 - docker push 镜像推送
 - docker search 镜像查询

课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
 - 3.1 Docker基础概念
 - 3.2 Docker架构与使用
4. 容器编排技术基础-Kubernetes

容器Docker体系结构详解



Docker Registry: 镜像仓库

- 官方的仓库: Docker Hub
- 私有镜像仓库搭建: 基于开源项目Harbor 或 阿里云提供的镜像仓库服务

Docker Server: Docker 服务端

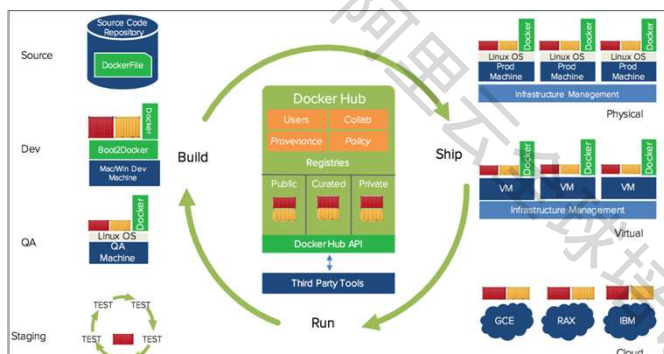
- 宿主机中运行守护进程(Docker daemon)
- 接收Docker客户端 (Client) 发送的指令来执行拉取镜像、缓存、启动容器等操作

Docker Client: Docker 客户端

- 主要负责通过docker命令行对容器进行基本操作, 如拉取镜像, 构建镜像, 运行容器等等;

Build, Ship, and Run Any App, Anywhere

- Docker 运行过程也就是去仓库把镜像拉到本地, 然后用一条命令把镜像运行起来变成容器。
- 我们也常常将Docker称为码头工人或码头装卸工, 这和Docker的中文翻译搬运工人如出一辙。



- **Build (构建镜像)**: 镜像就像是集装箱, 包含文件以及运行环境等资源;

- **Ship (运输镜像)**: 在宿主机和仓库间进行运输, 这里仓库就像是超级码头;

- **Run (运行镜像)**: 运行的镜像就是一个容器, 容器就是运行程序的地方。

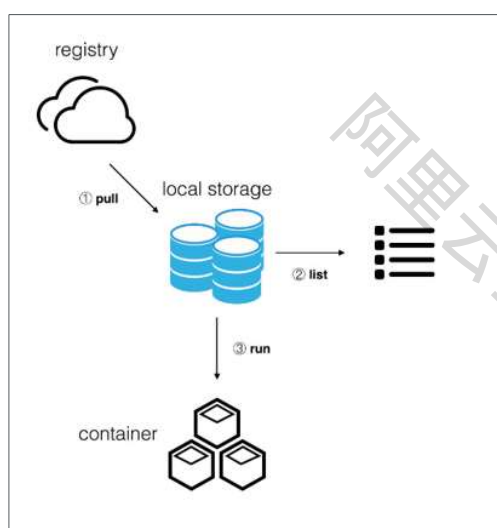
安装Docker

- Docker 在主流的操作系统和云平台上都可以使用
- 包括 Linux 操作系统(如 ubuntu、Debian、CentOS、Redhat 等)、MacOS 操作系统和 Windows 操作系统



- 推荐尽量使用 Linux 操作系统来运行 Docker，因为目前 Linux 操作系统对 Docker 的支持是原生的，使用体验最好。
- 用户可以访问 Docker 官网的 Get Docker(<https://www.docker.com/products/overview>)页面，查看获取 Docker 的方式，以及 Docker 支持的平台类型

如何运行一个容器？



- 从Docker registry 下载镜像
 - - `docker pull busybox:1.25`
- 查看本地镜像列表
 - - `docker images`
- 选择相应的镜像并运行
 - - `docker run [-d] --name demo busybox:1.25 top`

课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
4. 容器编排技术基础-Kubernetes
 - 4.1 Kubernetes基础概念
 - 4.2 Kubernetes核心架构
 - 4.3 Kubernetes流程及场景

28



Kubernetes容器编排的事实标准

• Kubernetes

- Google开源的一个容器编排引擎
- 支持自动化部署、大规模可伸缩、应用容器化管理
- 在生产环境中部署一个应用程序时，通常要部署该应用的多个实例以便对应用请求进行负载均衡。



• 核心功能

- 服务发现与负载均衡
- 容器的自动装箱
- 存储编排
- 自动化容器恢复
- 自动发布与回滚
- 配置与密文管理
- 批量执行
- 水平伸缩

Kubernetes 是一个自动化的容器编排平台，它负责应用的部署、应用的弹性以及应用的管理，这些都是基于容器的。

29

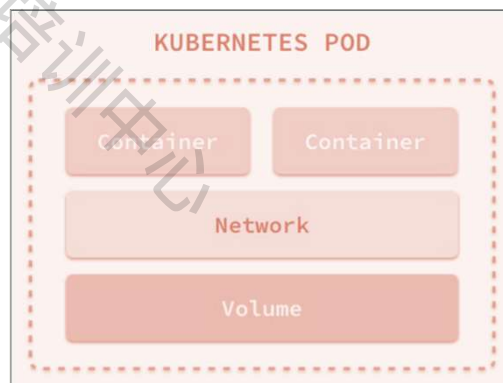


Kubernetes核心概念-Pod

- Pod

- 是Kubernetes中能够创建和部署的最小单元
- 是Kubernetes集群中的一个应用实例，总是部署在同一个节点Node上
- 包含了一个或多个容器，还包括了存储、网络等各个容器共享的资源
- 支持多种容器环境，Docker则是最流行的容器环境

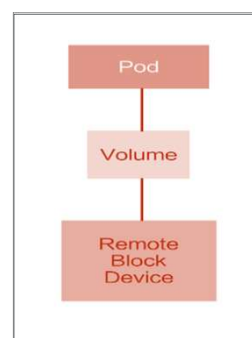
- 最小的调度及资源单位
- 由一个或者多个容器组成
- 定义容器的运行方式
- 提供给容器共享的运行环境



Kubernetes核心概念-Volume

- Volume

- Pod中能够被多个容器共享的磁盘目录
- 用来管理 Kubernetes 存储的
- 用来声明在 Pod 中的容器可以访问文件目录的
- 一个卷可以被挂载在 Pod 中一个或者多个容器的指定路径下面
- 声明在Pod中的容器可访问的文件目录
- 可以被挂载在 Pod 中一个或者多个容器的路径
- 支持多种后端存储的抽象：本地存储、分布式存储、阿里云存储等

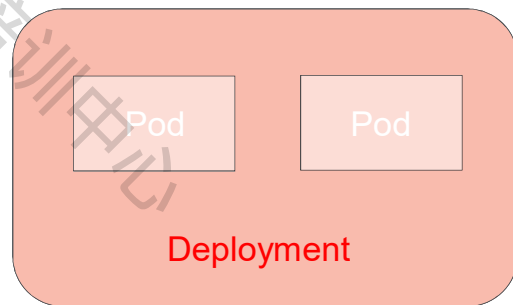


Docker提供了Volume机制以便实现数据的持久化，Kubernetes中Volume的概念与Docker中的Volume类似

Kubernetes核心概念-Deployment

- **Deployment**

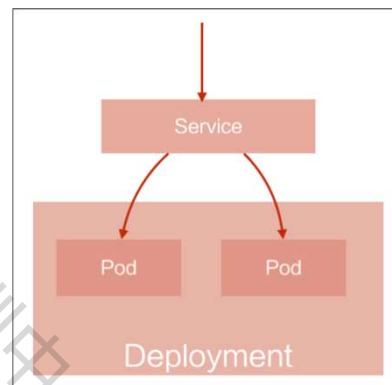
- 是Kubernetes中部署应用最常见的一种方式
- 做应用的真正的管理
- Pod 是组成 Deployment 最小的单元
- 定义一组 Pod 的副本数目、版本等
- 通过控制器维持Pod数目，自动恢复失败的Pod
- 通过控制器以指定的策略控制版本



Kubernetes核心概念-Service

- **Service**

- 是Kubernetes核心的概念
- 通过创建Service，可以为一组具有相同功能的容器应用提供一个统一的入口地址
- 将请求进行负载均衡分发到后端的各个容器应用上
- 提供了一个或者多个 Pod 实例的稳定访问地址
- 支持多种访问方式实现：
 - Cluster IP
 - NodePort
 - LoadBalance

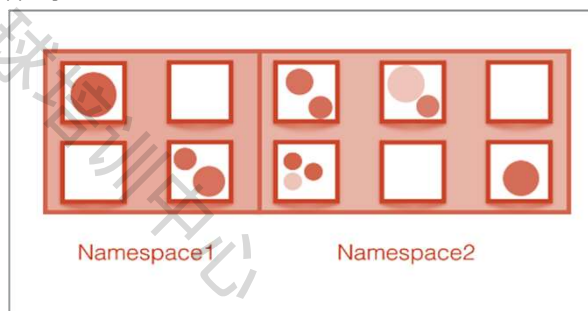


Kubernetes核心概念-Namespace

- **Namespace**

- 用来做一个集群内部的逻辑隔离的，它包括鉴权、资源管理等

- 每个资源都属于一个Namespace
- 同一个Namespace中的资源名唯一
- 不同Namespace中资源可重名



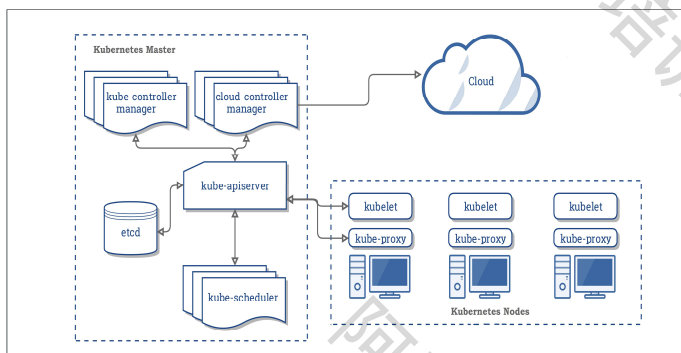
Namespaces可以理解为Kubernetes集群中的虚拟化集群。在一个Kubernetes集群中可以拥有多个命名空间，它们在逻辑上彼此隔离,他们可以为软件开发团队提供组织，安全、性能等方面的保障。

课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
4. **容器编排技术基础-Kubernetes**
 - 4.1 Kubernetes基础概念
 - 4.2 Kubernetes核心架构**
 - 4.3 Kubernetes流程及场景

Kubernetes核心架构

- Kubernetes系统用于管理分布式节点集群中的微服务或容器化应用程序
- 提供了零停机时间部署、自动回滚、缩放和容器的自愈（其中包括自动配置、自动重启、自动复制的高弹性基础设施，以及容器的自动缩放等）等功能。



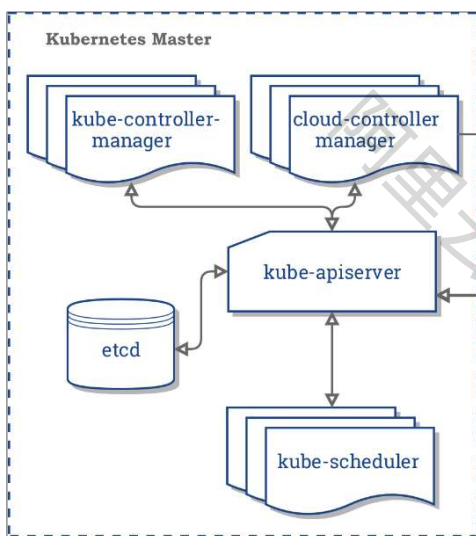
• Kubernetes Master (主控节点) :

- 集群的“大脑”，负责管理所有节点 (Node)
- 负责调度Pod在哪些节点上运行
- 负责控制集群运行过程中的所有状态

• Kubernetes Node (工作节点) :

- 负责管理所有容器 (Container)
- 负责监控/上报所有Pod的运行状态

Kubernetes架构- Master服务端



• kube-apiserver:

- 集群的HTTP REST API接口，是集群控制的入口

• kube-controller-manager:

- 集群中所有资源对象的自动化控制中心

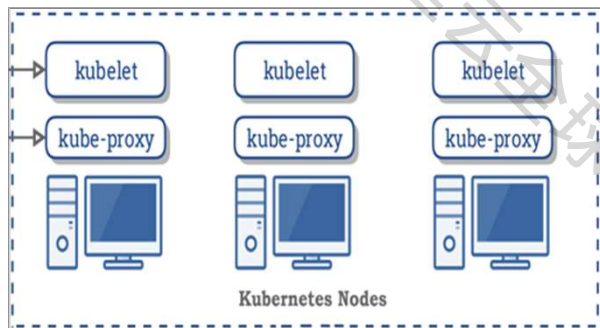
• kube-scheduler:

- 集群中Pod资源对象的调度服务

• Etcd:

- 一个分布式的存储系统

Kubernetes架构- Node客户端组件



•kubelet组件:

- 负责管理节点上容器的创建、删除、启停等任务，与Master节点进行通信。

•kube-proxy组件:

- 负责Kubernetes服务的通信及负载均衡服务。

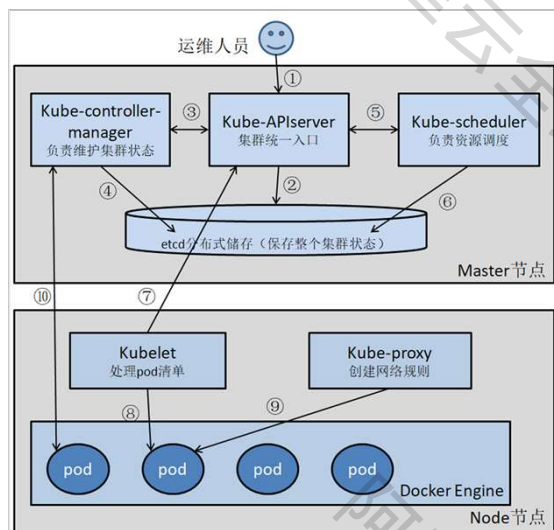
•Container Runtime:

- 负责容器的基础管理服务，接收kubelet组件的指令。

课程目录

1. 云原生技术概述
2. 云原生容器技术概述
3. 容器技术基础-Docker
- 4. 容器编排技术基础-Kubernetes**
 - 4.1 Kubernetes基础概念
 - 4.2 Kubernetes核心架构
 - 4.3 Kubernetes流程及场景

Kubernetes典型使用流程



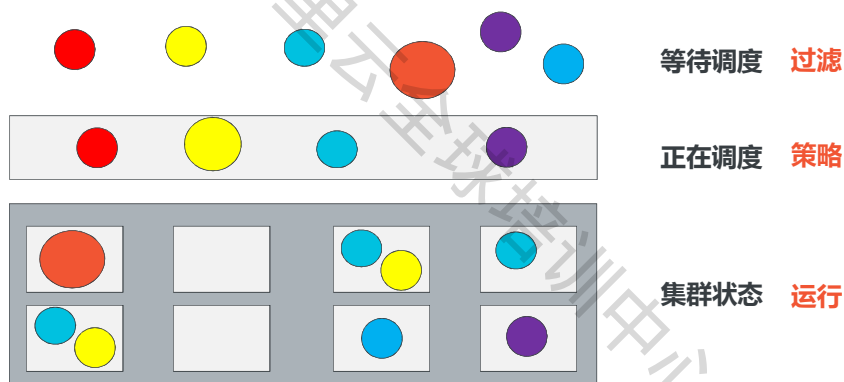
40

阿里云

Kubernetes使用场景-调度

调度器

- Kubernetes核心组件之一，承载着整个集群资源的调度功能
- 根据特定调度算法和策略，将Pod调度到最优工作节点上
- 更合理与充分的利用集群计算资源，使资源更好的服务于业务服务的需求

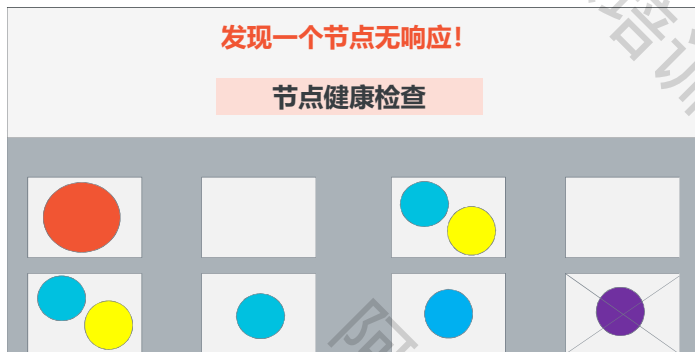


41

阿里云

Kubernetes使用场景-自动恢复

- Kubernetes的主要好处之一是它具有管理和维护集群中容器的能力
- 可以提供服务零停机时间的保障。
- Pod或容器出现故障时Kubernetes还可以让系统实现"自愈"

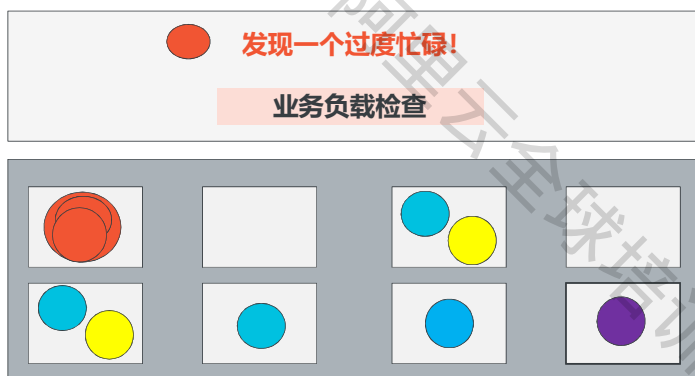


状态检查与修复

集群状态

Kubernetes使用场景-弹性伸缩

- Kubernetes 有业务负载检查的能力，会根据业务承担的负载，对业务进行扩容，实现资源弹性伸缩。



状态检查与修复

集群状态

本节课总结及回顾

本章节主要讲述内容：

1. 云原生是种架构模式及软件开发的新的思想理念
2. 阿里巴巴对云原生的定义：云原生是一条使用户能低心智负担的、敏捷的、以可扩展、可复制的方式，最大的利用“云”的能力，发挥“云”的价值的最佳路径
3. Docker容器的基本概念：用于构建、发布和运行分布式应用的一个框架。它是一个跨平台、可移植并且简单易用的容器解决方案
4. Docker的三大核心概念：容器、镜像、镜像仓库
5. Kubernetes关键概念：Pod、Volume、Deployment、Service、Namespace
6. Kubernetes核心架构：Kubernetes系统架构遵循客户端/服务端（C/S）架构，系统架构分为Master和Node两部分，Master作为服务端，Node作为客户端

阿里云云原生容器ACP认证培训

—— 阿里巴巴容器技术演进



课程目标

学习完本课程后，你将能够：

1. 掌握容器技术在阿里巴巴发展历程
2. 掌握基于Kubernetes 为代表的容器技术正在成为云原生的新界面
3. 掌握云原生容器服务产品体系



课程目录

1. 容器技术在阿里巴巴落地历程
2. 阿里云容器服务体系介绍

<#>



全面揭秘阿里巴巴双 11 的云原生支撑力

- 2020年天猫“双11”全球狂欢节如约而至，更直观展现了数字经济的先发优势和巨大潜能。11月11日零点零分26秒，天猫双11的订单创建峰值就达到58.3万笔/秒，阿里云又一次扛住全球最大规模流量洪峰。



- 与此前不同的是，继去年天猫双11核心系统上云后，阿里巴巴基于数字原生商业操作系统，实现了全面云原生化，底层硬核技术升级带来了澎湃动力和极致效能。
- 以支撑订单创建峰值为例，每万笔峰值交易的IT成本较四年前下降了80%。

<#>



容器服务在“双11”关键价值及作用

- 从2019“双11上云”到2020“云上双11”，阿里核心系统在2020年实现了全面云原生化
- 革命性重构了双11“技术引擎”，技术引擎升级带来了澎湃动力和极致效能
- 其中容器服务在“双11”支撑过程起到了不可替代的关键价值



- 2020年双11创造了诸多“云原生的第一次”：例如80%核心业务部署在阿里云容器ACK上，可在1小时内扩展超百万容器
- 容器提供了应用分发和交付标准，将应用与底层运行环境进行解耦。

阿里集团向云原生容器界面的演进

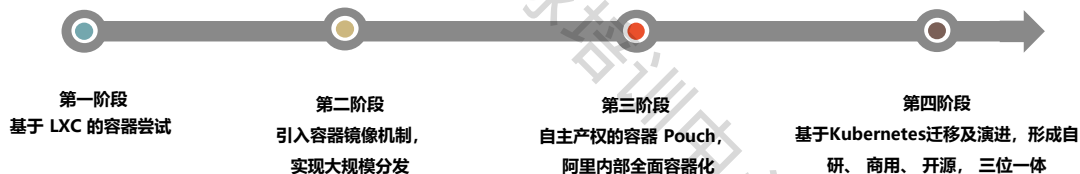
- 在过去十年，阿里集团内的容器技术，经历了从自研LXC容器T4，到Pouch容器化，再到Kubernetes云原生轻量级容器的演进历程。每一次转变升级，都是基于不同时期的业务背景，所做出的技术迭代和自我革新。

阿里集团在2011年时发起对LXC和Linux Kernel的定制，在内部上线了基于LXC的T4容器。

2015年，阿里引入Docker的镜像机制，将Docker和T4的功能整合，并在此基础上形成内部产品AliDocker。同时引入P2P镜像分发机制

阿里集团基于自研Pouch在2016年双11前，已经实现了全网的容器化。通过这种方式，极大加速了云原生的进一步演进。

2018年，阿里巴巴集团调度系统开始从内部定制的Sigma到Kubernetes逐步演进，最终形成自研、商用、开源，三位一体融合互补服务体系。



课程目录

1. 容器技术在阿里巴巴落地历程
2. 阿里云容器服务体系介绍

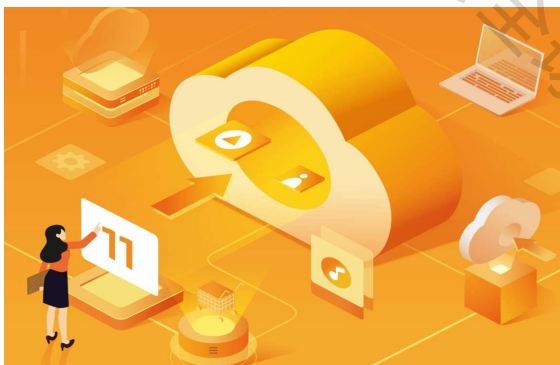
阿里云基于Kubernetes 为代表的容器技术，形成云计算的新界面

- 阿里巴巴经过多年在云计算领域的探索和实践，基于Kubernetes 为代表的容器技术正成为阿里云云计算的新界面。



- 向下封装基础设施，屏蔽底层架构的差异性
- 拓展云计算新边界，云边端一体化管理
- 向上支撑多种工作负载和分布式架构

阿里云容器服务普惠技术红利



- **敏捷**
 - 使用容器技术可以实现 3~10 倍交付效率提升
- **弹性**
 - 1 小时内扩展超百万容器，支撑上亿用户访问
- **可移植性**
 - 容器已经成为应用分发和交付的标准，可以将应用与底层运行环境解耦

<#>

阿里云

最高等级全球云原生容器规模化评测

- Gartner 发布 2020 年公共云容器报告，据报告显示：

- **阿里云拥有最丰富的产品布局，覆盖 9 项产品能力，排名第一**



Competitive Landscape of Public Cloud Container Service Market

Analysis Criteria	Generally Available		Preview		AWS		Google		IBM		Microsoft		Oracle	
	Alibaba	Alibaba	AWS	Google	IBM	Microsoft	Oracle	Alibaba	Alibaba	AWS	Google	IBM	Microsoft	Oracle
Container Orchestration Services	Kubernetes	Alibaba Cloud Container Service for Kubernetes (ACK)	Amazon Elastic Kubernetes Service (EKS)	Google Kubernetes Engine (GKE)	IBM Cloud Kubernetes Service (AKS)	Azure Kubernetes Service (AKS)	Oracle Container Engine for Kubernetes (OKE)							
	Other Orchestration Technologies	Alibaba Cloud Container Service for Kubernetes (ACK)	Amazon Elastic Container Service (ECS)											
Serverless Container Services	Serverless Kubernetes	Alibaba Cloud Serverless Kubernetes (ASK)	AWS EKS on Fargate			Azure Container Instances (ACI)								
	Serverless Container Instance	Alibaba Cloud Elastic Container	AWS ECS on Fargate	Google Cloud run on GKE		Azure Container Instances (ACI)								
Kubernetes Based Managed aPaaS	Red Hat OpenShift		Red Hat OpenShift Dedicated ¹	Red Hat OpenShift Dedicated ¹	Red Hat OpenShift on IBM Cloud	Azure Red Hat OpenShift								
	Service Mesh (Managed)	Alibaba Cloud Service Mesh	AWS App Mesh	Anthos Service Mesh	OpenShift Service Mesh									
Container Related Other Services	Registry													
	Marketplace	ACK on Alibaba Cloud	ECS/EKS on AWS Outposts	Anthos (for the Other Clouds) ²	IBM Cloud Pak	Azure Arc								
Hybrid and Multicloud Solutions	Managed Container Solutions	ACK												
	First Production Release Date	May 2016	April 2015	August 2015	March 2017	March 2018	May 2018							
General Criteria	Pricing Model for Container Services ³	Free	10 Cents per Hour per Cluster (EKS/Managed EKS)	10 Cents per Hour per Cluster (GKE/Managed GKE)	Free	Free	Free							

<#>

阿里云

云原生容器服务产品体系

- 阿里云容器服务产品家族可以在公共云、边缘计算和专有云环境提供企业容器平台，以为全球上万家企业客户提供了服务。



- 容器服务Kubernetes 版 (ACK)
- Serverless Kubernetes (ASK)
- 镜像服务 (ACR)
- 服务网格 (ASM)
- 边缘容器服务ACK@Edge

<#>

阿里云

本节课总结及回顾

本章节主要讲述内容：

- 容器服务在“双11”支撑过程起到了不可替代的关键价值，80% 核心业务部署在阿里云容器 ACK 上，可在 1 小时内扩展超百万容器。
- 阿里集团内的容器技术，经历了从自研 LXC容器T4，到Pouch容器化，再到Kubernetes云原生轻量级容器的演进历程。
- 阿里巴巴经过多年在云原生领域的探索和实践，基于Kubernetes 为代表的容器技术正成为阿里云云原生的新界面。
- 阿里云容器服务产品体系：容器服务Kubernetes 版 (ACK)、Serverless Kubernetes (ASK)、镜像服务 (ACR)、服务网格 (ASM)、边缘容器服务ACK@Edge。

<#>

阿里云

阿里云

阿里云云原生容器ACP认证培训

阿里云容器服务Kubernetes版 (ACK) 详解

阿里云

课程目标

学习完本课程后，你将能够：

1. 掌握阿里云容器产品的三种服务形态,以及不同的适用场景。
2. 掌握阿里云容器服务ACK产品功能、优势、应用场景等。
3. 掌握阿里云容器服务Kubernetes托管版 ACK Pro产品功能、优势等。
4. 掌握容器服务托管版ACK入门操作-集群创建、集群查看及连接、创建无状态应用、创建有状态StatefulSet应用。

课程目录

1. 容器服务Kubernetes版（ACK）概述

1.1 容器服务 ACK 集群

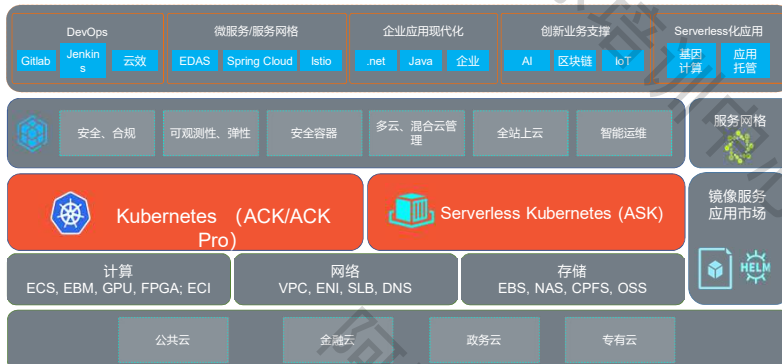
1.2 容器服务 ACK Pro版集群

1.3 容器服务 ACK 应用场景

2. 容器服务Kubernetes版（ACK）入门操作

容器服务正在成为云原生时代的基础设施

- 容器服务 Kubernetes 版基于原生 Kubernetes 进行适配和增强
- 简化集群的搭建和扩容等工作，整合阿里云虚拟化、存储、网络和安全能力
- 打造云端最佳的 Kubernetes 容器化应用运行环境



• 阿里云容器服务产品三种形态：

- 专有版Kubernetes
- 托管版Kubernetes
- Serverless Kubernetes

阿里云容器服务给客户带来的价值



全球部署

- 单元化、快速部署
- 全球应用分发



极致效率

- 提升资源利用率
- 网络0损耗



灵活弹性

- 及时应对突发流量
- 优化计算成本



混合云管理

- 统一纳管多地域集群
统一流量管理

Kubernetes容器服务产品的三种形态

比较项	专有版Kubernetes	托管版Kubernetes	Serverless Kubernetes
主要特点	您需要自行创建Master节点及Worker节点。	您只需创建Worker节点，Master节点由ACK创建并托管。	您无需创建Master节点及Worker节点。
核心优势	可以对集群基础设施进行更细粒度的控制，需要自行规划、维护、升级服务器集群。	简单、低成本、高可用，无需管理Master节点。	无需管理任何节点，可直接启动应用程序。
收费方式	集群管理免费，但需要承担Master节点、Worker节点以及其他基础资源费用。	•标准版：集群管理免费，但需要承担Worker节点以及其他基础资源的费用。 •Pro版：按照集群数量或者包年包月方式收费。	按容器实例的使用资源量和时长（秒）计费。
应用场景	适用于所有场景。	适用于所有场景。	适用于批量任务、突发扩容，以及CI/CD测试。
容量规划	需要，支持自动伸缩	需要，支持自动伸缩	不需要
使用场景	全场景支持	全场景支持	突发负载，如，批处理，CI/CD任务

<#>

阿里云

阿里云容器服务三种形态的适用用户



<#>

阿里云

阿里云容器服务Kubernetes版 ACK

- 阿里云容器服务Kubernetes版ACK
 - 基于Kubernetes 云原生技术
 - 整合阿里云虚拟化、存储、网络和安全能力
 - 提供高性能可伸缩的容器应用管理能力
 - 简化集群的搭建和扩容等工作，让您专注于容器化的应用的开发与管理。



<#>

阿里云

容器服务ACK基于K8S的增强

- 在支持标准kubernetes的基础上，ACK（阿里云容器Kubernetes服务）提供了更多与用户息息相关的能力提升



<#>

阿里云

容器服务ACK与自建集群对比

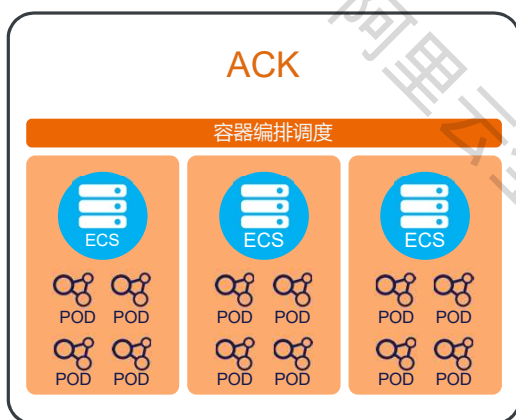
	容器服务Kubernetes版	自建Kubernetes
集群管理	通过控制台一键创建集群，支持GPU实例和裸金属服务器，支持创建跨AZ高可用的集群。 提供容器优化的OS镜像，提供稳定测试和安全加固的Kubernetes和Docker版本。 支持多集群管理，支持跨AZ高可用集群，支持集群联邦管理。	用户手动部署集群并自行开发。 用户自行探索和开发。
应用管理	支持灰度发布，支持蓝绿发布。 支持应用监控、应用弹性伸缩。 内置应用商店，支持Helm应用一键部署；支持服务目录，简化云服务集成。	用户自行探索和开发。
网络管理	提供针对阿里云优化的高性能VPC/ENI网络插件，性能优于普通网络方案20%。 支持容器访问策略和容器带宽限制。	需要挑选社区网络插件进行适配。 用户自行探索和开发。
存储管理	支持阿里云云盘、本地盘、NAS、CPFS和OSS，提供标准的CSI、FlexVolume驱动。 支持存储卷自动创建、迁移。	用户自行探索和开发。
运维管理	支持Kubernetes新版本一键升级，支持Kubernetes集群自动添加节点。 提供高性能日志采集Agent，支持10万QPS。	用户自行探索和开发。
服务保障	阿里云专业容器团队作为技术支持，为集群提供及时的稳定性和安全响应。 历经阿里大规模实践，是目前国内最大规模的公共云容器服务。 CNCF白金会员，已通过一致性验证。 2019年Forrester报告国内排名第一、2019~2020连续两年Gartner竞争格局国内唯一入选。	需要组建专门团队。

<#>



容器服务ACK产品核心功能-集群管理

- 通过控制台一键创建 Kubernetes 集群，支持跨可用区高可用。



- 集群形态：ACK/ACK Pro/ASK
- 组件升级：一键升级K8s版本
- 弹性伸缩：支持垂直伸缩、水平伸缩等多种扩容模式
- 多云管理：支持多云、混合云多集群管理：
- 安全授权：支持RAM授权和RBAC权限管理

<#>



容器服务ACK产品核心功能-应用管理

- 一站式容器全生命周期管理

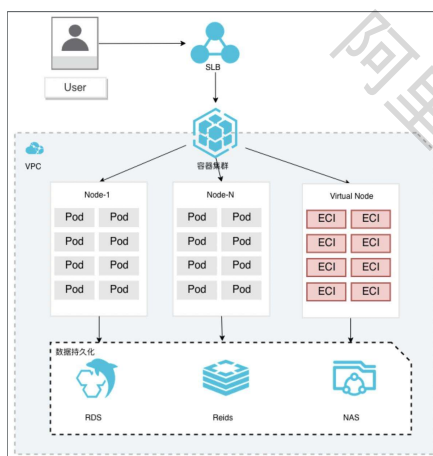


- 应用创建：从镜像、模版的创建，支持环境变量、应用健康的配置
- 应用全生命周期管理：支持应用查看、更新、删除、回滚等
- 应用调度：节点亲和性调度、应用间亲和性调度和反亲和性调度
- 应用伸缩：支持手动伸缩应用容器实例，HPA自动伸缩策略
- 应用发布：支持灰度发布和蓝绿发布
- 应用目录：支持应用目录，简化云服务集成

<#>

容器服务ACK产品核心功能-存储与网络

- 整合阿里云虚拟化、存储、网络和安全能力



- 存储支持：阿里云云盘、文件存储NAS、对象存储 OSS 等持久化存储支持
- 网络支持：高性能 Flannel VPC/Terway 容器网络

<#>

容器服务ACK产品核心功能-运维与安全

- 提供全方位的安全管理体系：基础架构安全、供应链安全、容器运行时安全。



- 监控**：支持集群、节点、应用、容器实例监控
- 日志**：日志管理及日志查看
- 安全中心**：提升容器安全整体纵深防御能力
- 安全沙箱**：提供安全沙箱环境，拥有独立内核
- 机密计算**：支持机密计算，提供可信执行加密环境

<#>

阿里云

容器服务ACK产品核心功能-弹性伸缩

弹性伸缩

- ACK上被广泛采用的功能，典型的场景包含在线业务弹性、大规模计算训练、深度学习GPU或共享GPU的训练与推理、定时周期性负载变化等。



弹性伸缩分为两个维度：

- 调度层弹性
- 资源层弹性

调度层弹性组件包括：

- 容器水平伸缩（HPA）
- 容器垂直伸缩（VPA）
- 容器定时伸缩（CronHPA）
- 弹性负载（Elastic-Workload）

<#>

阿里云

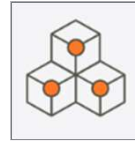
容器服务ACK产品优势



强大的集群管理



极致弹性的资源扩缩



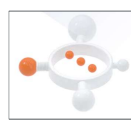
一站式容器管理



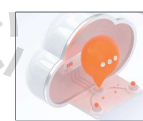
丰富的工作节点



最优的IaaS层能力



企业级的安全稳定



全天候技术支持

<#>

阿里云

课程目录

1. 容器服务Kubernetes版（ACK）概述
 - 1.1 容器服务 ACK 集群
 - 1.2 容器服务 ACK Pro版集群
 - 1.3 容器服务 ACK 应用场景
2. 容器服务Kubernetes版（ACK）入门操作

<#>

阿里云

ACK Pro版集群概述

- ACK Pro版集群
 - 在ACK托管版基础上针对企业大规模生产环境进一步增强了可靠性、安全性
 - 继承了原托管版集群的所有优势
 - 提供可赔付的SLA的Kubernetes集群



- 增强之一：更可靠
- 增强之二：高安全
- 增强之三：高调度性能

ACK Pro版集群和标准版集群的对比详情

属性分类	ACK Pro版	ACK托管标准版
集群规模	最大5000节点。	最大100节点（现有集群不受影响，可以升级到Pro版）。
SLA	99.95%（支持赔付）。	99.9%（不支持赔付）。
API Server	自动弹性伸缩机制。 可用性监控。	不支持。
etcd	高频冷热备机制，异地容灾。	不支持。
安全管理	开放高级版，支持数据加密	开放标准版
托管节点池*	支持	不支持

- 托管节点池是ACK全新推出的具有自动升级、节点自愈能力的免运维型节点池，为您提供统一的、全托管的节点生命周期管理。

ACK Pro版集群-可靠性强化集群

- 更可靠的托管Master节点，API Server自动弹性，保障集群平滑扩容海量节点。



<#>

阿里云

ACK Pro版集群-安全性强化集群

- 开放安全管理，并提供针对运行中容器更强检测和自动修复能力的安全管理高级版



<#>

阿里云

ACK Pro版集群-调度性强化集群

- 集成更强调度性能的kube-scheduler，支持多种智能调度算法
- 优化在大规模数据计算、高性能数据处理等业务场景下的容器调度能力



<#>

阿里云

ACK Pro版集群适用场景



- **互联网企业**，大规模业务上线生产环境，对管控的稳定性、可观测性和安全性有较高要求。



- **大数据计算企业**，大规模数据计算、高性能数据处理、高弹性需求等类型业务，对集群稳定性、性能和效率有较高要求。



- **开展中国业务的海外企业**，对有赔付标准的SLA以及安全隐私等非常重视。



- **金融企业**，需要提供赔付标准的SLA。

<#>

阿里云

课程目录

1. 容器服务Kubernetes版 (ACK) 概述

1.1 容器服务 ACK 集群

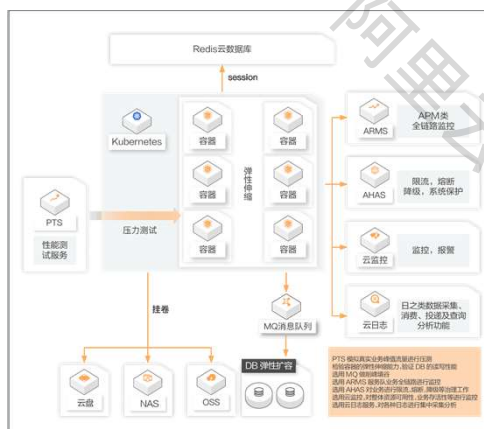
1.2 容器服务 ACK Pro版集群

1.3 容器服务 ACK 应用场景

2. 容器服务Kubernetes版 (ACK) 入门操作

容器服务ACK应用场景-弹性伸缩架构

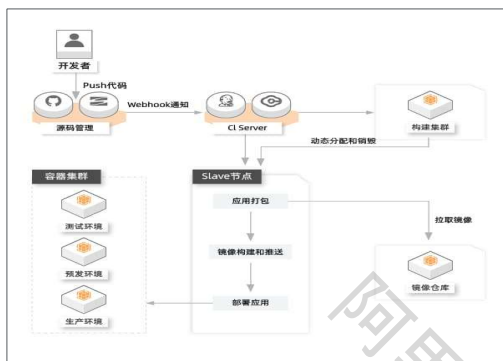
- 容器服务可以根据业务流量自动对业务扩容/缩容，不需要人工干预
- 避免流量激增扩容不及时导致系统挂掉，以及平时大量闲置资源造成浪费



- 峰值时极速扩容
- 高可靠性保障、低运维工作量、低成本
- 波峰结束后，自动缩容。策略化弹缩支撑业务

容器服务ACK应用场景- DevOps 持续交付

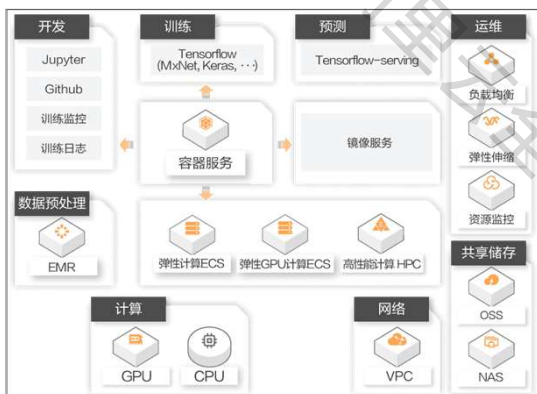
- 随着企业业务需求变化频繁，业务测试上线需求频繁，流程也越来越复杂
- 面对项目和环境的增加，人手不足的问题也导致应对这些变化时压力较大，响应缓慢
- 因此，为解决这些问题，我们在充分研究后利用容器技术帮助企业实现自动化代码构建、企业系统版本管理、持续集成



- DevOps自动化
- 环境一致性
- 持续反馈
- 持续交付

容器服务ACK应用场景-云原生AI

- 深度学习领域需要处理的三个核心问题是性能、效率和成本
- 利用ACK容器服务部署机器学习应用，无缝的整合了云的计算、存储、负载均衡等服务，同时贯穿了深度学习的全生命周期



- 容器化技术封装深度学习框架
- 多种计算资源灵活调度（GPU、CPU、云计算资源）
- 轻松集成，快速发布

容器服务ACK应用场景-微服务架构

- 企业生产环境中，通过合理微服务拆分，将每个微服务应用存储在阿里云镜像仓库帮您管理
- 您只需迭代每个微服务应用，由阿里云容器服务Kubernetes版ACK提供调度、编排、部署和灰度发布能力
- 帮助企业快速实现敏捷开发和部署落地，加速企业业务迭代



- **负载均衡和服务发现**
 - 支持4层和7层的请求转发和后端绑定。
- **丰富的调度和异常恢复策略**
 - 支持服务级别的亲和性调度，支持跨可用区的高可用和灾难恢复。
- **微服务监控和弹性伸缩**
 - 支持微服务和容器级别的监控，支持微服务的自动伸缩。

<#>

阿里云

容器服务ACK应用场景-混合云架构

- 在容器服务控制台上同时管理云上云下的资源，不需在多种云管理控制台中反复切换
- 基于容器基础设施无关的特性，使用同一套镜像和编排同时在云上云下部署应用
- 基于阿里云容器服务Kubernetes版ACK实现统一运维多个云端资源



- **在云上伸缩应用**
 - 业务高峰期，在云端快速扩容，把一些业务流量引到云端。
- **云上容灾**
 - 业务系统同时部署到云上和云下，云下提供服务，云上容灾。
- **云下开发测试**
 - 云下开发测试后的应用无缝发布到云上。

<#>

阿里云

本节课小结及回顾

本章节主要讲述内容：

1. 阿里云Kubernetes容器服务产品的三种形态？之间有什么区别？

- 专有版ACK、托管版ACK（标准版、Pro版）、Serverless Kubernetes(ASK)；
- 专有版ACK需要自行规划、创建、升级服务器集群；
- 容器服务托管版ACK（标准版、Pro版）简单、低成本、高可用，无需管理Master节点；
- 容器服务ASK无需管理任何节点，可直接启动应用程序。

2. 阿里云Kubernetes容器服务ACK基础功能及优势。

3. 容器服务ACK应用场景？

弹性伸缩架构应用场景， DevOps 持续交付应用场景，云原生AI应用场景、微服务架构、混合云架构

课程目录

1. 容器服务Kubernetes版（ACK）概述

2. 容器服务Kubernetes版（ACK）入门操作

2.1 入门实践集群创建

2.2 集群查看及管理

2.3 使用镜像快速创建无状态应用

2.4 使用镜像快速创建有状态应用

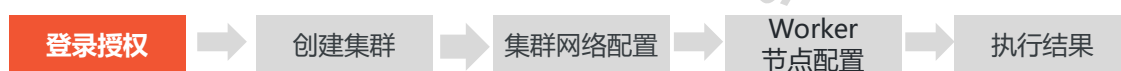
容器服务ACK的使用流程



<#>

阿里云

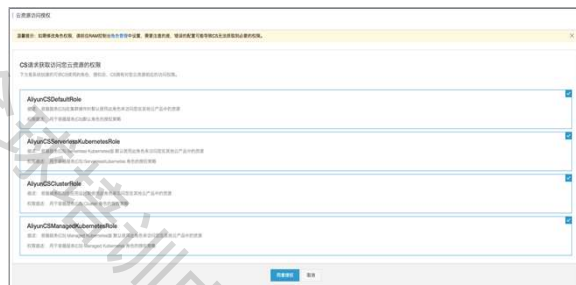
容器服务ACK的入门实践-集群创建



- 容器服务需要创建默认角色
- 点击"前往RAM进行授权"



- 给ACK需要用的云资源访问进行授权



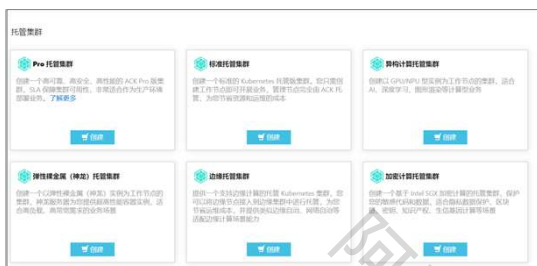
<#>

阿里云

容器服务ACK的入门实践-集群创建



- 登录容器服务管理控制台
- 在控制台左侧导航栏中，单击集群
- 在集群列表页面中，单击页面右上角的创建集群
- 在ACK托管版页面，完成集群配置



- 创建kubernetes集群：
- 输入自定义集群名称（ACA-TEST）
- 选择地域



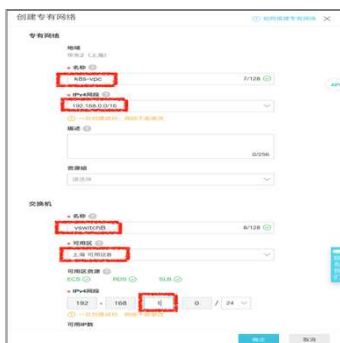
<#>



容器服务ACK的入门实践-集群创建



创建专有网络



- 自定义vpc名称，选择vpc的ipv4网段，自定义交换机名称，选择交换机可用区，指定交换机的IPv4网段，点击确定，完成。

创建详情



选择网络配置

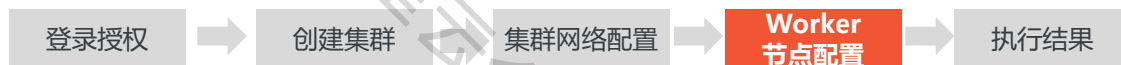


- 回到创建kubernetes集群控制台，点击专有网络右侧的刷新按钮，会自动选上刚刚我们创建的vpc和vswitch

<#>



容器服务ACK的入门实践-集群创建



- 给worker节点选型，选择数量，选择规格为共享内存型，数量为1台，此处我们暂时不挂载数据盘（生产环境建议大家勾选挂载数据盘，该盘是专门给docker存放镜像和容器相关的文件使用）



- 选择操作系统类型
- 选择worker节点登录方式，设置密码



组件配置

<#>



容器服务ACK的入门实践-集群创建



产品名称	产品类型	产品规格	数量	计费方式	购买策略
ACK集群	ACK集群	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD Service CIDR: 172.17.0.0/16	1	按量付费	无
容器镜像服务	容器镜像服务	镜像仓库: ECR 网络带宽: 10 Gbps 计费方式: 按量付费	1	按量付费	无
容器网络	容器网络	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器存储	容器存储	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器安全	容器安全	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器监控	容器监控	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器日志	容器日志	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器网络	容器网络	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器存储	容器存储	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器安全	容器安全	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器监控	容器监控	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无
容器日志	容器日志	规格: g6t2 (50%) 内核数: 2, 1.8 GiB内存 网络带宽: 10 Gbps 系统盘: 100 GB SSD 数据盘: 100 GB SSD	1	按量付费	无

- 点击创建集群按钮后，需要我们确认配置，系统会帮我们检查当前的配置是否通过。

依赖检查

依赖项	名称	状态	说明
容器网络	容器网络	通过	容器网络配置正确
容器存储	容器存储	通过	容器存储配置正确
容器安全	容器安全	通过	容器安全配置正确
容器监控	容器监控	通过	容器监控配置正确
容器日志	容器日志	通过	容器日志配置正确

集群创建成功



- 集群创建成功后，您可以在容器服务管理控制台的集群列表页面查看所创建的集群。

<#>



课程目录

1. 容器服务Kubernetes版 (ACK) 概述
2. 容器服务Kubernetes版 (ACK) 入门操作
 - 2.1 入门实践集群创建
 - 2.2 集群查看及管理
 - 2.3 使用镜像快速创建无状态应用
 - 2.4 使用镜像快速创建有状态应用

容器服务ACK的入门实践-集群查看

- 容器服务Kubernetes版集群提供集群概览展示页，包括应用状态、组件状态和资源监控等功能板块，方便您快速了解集群的健康状态信息。

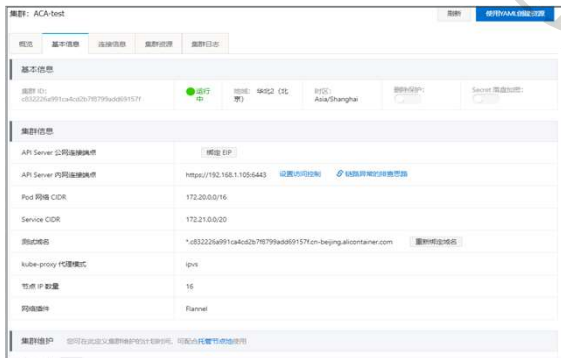
选择所需的集群和命名空间，您可查看应用状态、组件状态和资源监控图表。



- 应用状态:** 显示当前运行的部署、容器组和副本集的状态示意图，绿色图标代表正常，黄色图标代表异常。
- 节点状态:** 显示当前集群的节点状态。
- 组件状态:** 查看核心组件状态。
- 资源监控:** 提供CPU和内存的监控图表。
- 事件:** 显示集群的事件信息，例如警告和错误事件等。

容器服务ACK的入门实践-集群查看

- 查看集群的基本信息、连接信息、集群资源、集群日志以及集群时区。



集群基本信息

- 基本信息：**可以查看集群ID、地域、API Server连接端点以及其他网络信息。
- 连接信息：**获取公网和内网环境下KubeConfig文件的配置内容，用于配置通过KubectI客户端访问集群。
- 集群资源：**查看集群所使用的相关云资源。
- 集群日志：**查看集群日志。
- 集群时区：**查看集群的托管侧时区和Worker节点时区。

<#>

多种方式连接集群

- 容器服务ACK提供了多种方式连接和接入Kubernetes集群



- 通过kubectI连接Kubernetes集群
- 利用CloudShell通过kubectI管理Kubernetes集群
- 通过SSH访问Kubernetes集群
- 使用ServiceAccount Token访问Kubernetes集群
- 通过公网访问集群API Server

<#>

多种方式连接集群-通过SSH访问Kubernetes集群

- 如果您在创建集群时，选择不开放公网SSH访问，您将无法SSH访问Kubernetes集群，而且也无法通过kubectl连接Kubernetes集群。
- 如果创建集群后，您需要SSH访问您的集群，可以手动为ECS实例绑定EIP，并配置安全组规则，开放SSH（22）端口。
 1. 登录容器服务管理控制台。
 2. 在控制台左侧导航栏中，单击**集群**。
 3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。



4. 在**集群信息**页面，单击**集群资源**页签，找到您的**API Server 负载均衡（SLB）**，并单击实例ID链接。
5. 在左侧导航栏，选择**实例 > 实例管理**，单击**监听**页签，然后单击**添加监听**。
6. 添加SSH监听规则。
7. 监听创建完成后，您就可以使用该负载均衡的服务地址SSH访问您的集群了。

<#>

阿里云

课程目录

1. 容器服务Kubernetes版（ACK）概述
2. 容器服务Kubernetes版（ACK）入门操作
 - 2.1 入门实践集群创建
 - 2.2 集群查看及管理
 - 2.3 使用镜像快速创建无状态应用
 - 2.4 使用镜像快速创建有状态应用

<#>

阿里云

容器服务操作实践-使用镜像快速创建无状态应用

镜像创建

应用基本信息

容器配置

高级配置

创建并查看

- 使用镜像快速创建一个可公网访问的应用。



1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在集群列表页面中，单击目标集群名称查看详情。
4. 在集群管理页左侧导航栏中，选择**工作负载** > **无状态**。
5. 在无状态页签中，单击**使用镜像创建**。
6. 在应用**基本信息**页签，设置应用的基本信息

容器服务操作实践-使用镜像快速创建无状态应用

镜像创建

应用基本信息

容器配置

高级配置

创建并查看

创建

应用基本信息 容器配置 高级配置 创建完成

应用名称

名称为1-63个字符，可包含数字、小写字母以及短划线(-)，且不能以短划线(-)开头

副本数量

类型

标签

注解

时区同步 ☐ 容器与节点使用相同时区

返回 下一步

应用名称	设置应用的名称。
命名空间	设置应用部署所处的命名空间，默认使用default命名空间。
副本数量	应用包含的Pod数量。
类型	定义资源对象的类型。
标签	为该应用添加一个标签，标识该应用。
注解	为该应用添加一个注解（annotation）。
时区同步	容器与节点是否使用相同的时区。

容器服务操作实践-使用镜像快速创建无状态应用

镜像创建

应用基本信息

容器配置

高级配置

创建并查看

镜像名称	选择镜像，在弹出的对话框中选择所需的镜像并单击确定，本例中为hello-world
镜像Tag	选择镜像Tag选择镜像的版本。若不指定，默认为最新版。三种镜像拉取策略
资源限制	可指定该应用所能使用的资源上限
所需资源	即为该应用预留资源额度
容器启动项	Stdin和tty
特权容器	开启或者不开启
Init Container	表示创建一个Init Container。Init（一些实用的工具）

<#>

阿里云

容器服务操作实践-使用镜像快速创建无状态应用

镜像创建

应用基本信息

容器配置

高级配置

创建并查看

访问设置	您可以设置暴露后端Pod的方式。本例中选择ClusterIP服务和路由（Ingress），构建一个公网可访问的Helloworld应用。
伸缩配置	在伸缩配置区域，配置是否开启指标伸缩和定时伸缩，从而满足应用在不同负载下的需求。本案例设置为指标伸缩。
调度配置	您可设置升级方式、节点亲和性、应用亲和性、应用非亲和性和调度容忍，本案例无需设置。
标签和注释	为该Pod添加一个标签，为该Pod添加一个注解。

<#>

阿里云

容器服务操作实践-使用镜像快速创建无状态应用

镜像创建

应用基本信息

容器配置

高级配置

创建并查看

创建应用任务已提交

创建应用任务已提交		
创建部署	nginx-aca	成功
创建ingress	hello-word	成功
创建指标伸缩		成功

查看应用详情

查看容器基本信息

nginx-aca									
名称	nginx-aca	创建时间	2021-01-12 11:00:00	创建人	root	容器引擎	docker	容器镜像	nginx/nginx-aca
描述									
状态	就绪	就绪时间	2021-01-12 11:00:00	就绪人	root	容器引擎	docker	容器镜像	nginx/nginx-aca
配置									
容器配置									
高级配置									
应用									
名称	nginx-aca	部署	nginx-aca	容器引擎	docker	容器镜像	nginx/nginx-aca	容器配置	nginx/nginx-aca
状态	就绪	就绪时间	2021-01-12 11:00:00	就绪人	root	容器引擎	docker	容器镜像	nginx/nginx-aca
配置									
容器配置									
高级配置									
应用									

<#>

阿里云

课程目录

1. 容器服务Kubernetes版（ACK）概述
2. 容器服务Kubernetes版（ACK）入门操作
 - 2.1 入门实践集群创建
 - 2.2 集群查看及管理
 - 2.3 使用镜像快速创建无状态应用
 - 2.4 使用镜像快速创建有状态应用

<#>

阿里云

Deployment和StatefulSet的区别

- 容器服务ACK集群容器部署可分为：
 - Deployment（用于部署无状态服务）
 - StatefulSet（用来部署有状态服务）

服务类型	服务	应用场景
无状态服务	deployment	web服务
有状态服务	StatefulSet	mysql、redis等

- Deployment创建的Pod是无状态的**，当挂在Volume之后，如果该Pod挂了，Replication Controller会再启动Pod一个来保证可用性，但是由于是无状态的，Pod挂了的时候与之前的Volume的关系就已经断开了，新起来的Pod无法找到之前的Pod。但是对于用户而言，他们对底层的Pod挂了没有感知，但是当Pod挂了之后就无法再使用之前挂载的磁盘了。
- 而对于有状态的应用部署则需要通过**StatefulSet创建部署**。

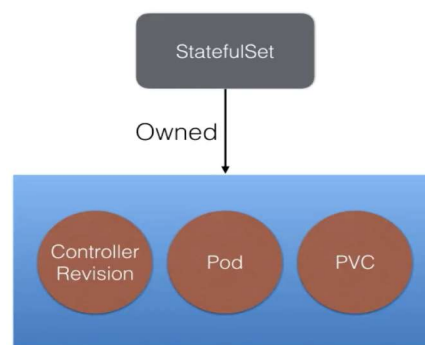
<#>



StatefulSet应用集特点

有状态应用集的特点

- 稳定且需要唯一的网络标识符；
- 稳定且持久的存储；
- 要求有序，平滑的部署和扩展；
- 要求有序，平滑的终止和删除；
- 有序的滚动更新。



- StatefulSet 可能会创建三种类型的资源：**
 - ControllerRevision**，通过这个资源，StatefulSet 可以很方便地管理不同版本的 template 模板。
 - PVC**，StatefulSet 会在创建 Pod 之前，先根据这个模板创建 PVC，并把 PVC 加到 Pod volume 中。
 - Pod**，StatefulSet 按照顺序创建、删除、更新 Pod，每个 Pod 有唯一的序号。

<#>



使用镜像快速创建有状态StatefulSet应用

镜像创建

应用基本信息

容器配置
(数据卷)

高级配置

创建并查看



1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在集群列表页面中，单击目标集群名称查看详情。
4. 在集群管理页左侧导航栏中，选择**工作负载 > 有状态**。
5. 在有状态页签中，单击**使用镜像创建**。
6. 在应用基本信息页签，设置应用的基本信息。

<#>

阿里云

使用镜像快速创建有状态StatefulSet应用

镜像创建

应用基本信息

容器配置
(数据卷)

高级配置

创建并查看

- 在**应用基本信息**页签中，设置应用的基本信息。
- 本例中选择**有状态**类型，创建StatefulSet类型的应用。

创建

应用基本信息 容器配置 高级配置 创建并查看

应用名称

名称为1-63个字符，可包含数字、小写字母以及划线(-)，且不能以划线(-)开头

副本数量

类型 无状态

标签 [添加](#)

注解 [添加](#)

时区同步 ☒ 容器与节点使用相同时区

[返回](#) [下一步](#)

应用名称	设置应用的名称。
命名空间	设置应用部署所处的命名空间，默认使用default命名空间。
副本数量	应用包含的Pod数量。
类型	定义资源对象的类型。 本例中选择有状态类型
标签	为该应用添加一个标签，标识该应用。
注解	为该应用添加一个注解（annotation）。
时区同步	容器与节点是否使用相同的时区。

<#>

阿里云

使用镜像快速创建有状态StatefulSet应用

镜像创建

应用基本信息

容器配置
(数据卷)

高级配置

创建并查看

The screenshot shows the 'Container Configuration' step in a wizard. It includes fields for 'Image Name' (with a 'Select Image' button), 'Image Tag' (with a 'Select Image Tag' button), and a checkbox for 'Always Pull Image'. Below these are resource limits for CPU (0.5 Core) and Memory (128 MiB Ephemeral-Storage). There are also fields for 'Container CPU' (0.25 Core) and 'Container Memory' (512 MiB Ephemeral-Storage). At the bottom, there are checkboxes for 'Container Stdin/Tty', 'Privileged Container', and 'Init Container'.

7. 容器基本配置

镜像名称	选择镜像，在弹出的对话框中选择所需的镜像并单击确定，本例中为hello-word
镜像Tag	选择镜像Tag选择镜像的版本。若不指定，默认为最新版。三种镜像拉取策略
资源限制	可指定该应用所能使用的资源上限
所需资源	即为该应用预留资源额度
容器启动项	Stdin和tty stdin：将控制台输入发送到容器。 tty：将标准输入控制台作为容器的控制台输入。
特权容器	开启或者不开启
Init Container	表示创建一个Init Container。Init（一些实用的工具）

<#>

阿里云

使用镜像快速创建有状态StatefulSet应用

镜像创建

应用基本信息

容器配置
(数据卷)

高级配置

创建并查看

设置容器相关的配置。

8. 可选配置：数据卷

- 支持增加本地存储和云存储声明（PVC）

The screenshot shows the 'Data Volumes' section with two options: 'Add Local Storage' and 'Add Cloud Storage Declaration (PersistentVolumeClaim)'. Each option has a table with columns for 'Storage Type', 'Mount Source', and 'Container Path'.

- 本例中配置了一个云存储类型的数据卷声明disk-ssd，将其挂载到容器的/tmp路径下。

9.完成容器配置后，单击下一步。

<#>

阿里云

使用镜像快速创建有状态StatefulSet应用

镜像创建

应用基本信息

容器配置
(数据卷)

高级配置

创建并查看

10. 可选：进行高级配置。

访问设置

您可以设置暴露后端Pod的方式。本例中选择ClusterIP服务和路由（Ingress），构建一个公网可访问Nginx的应用。

伸缩配置

在伸缩配置区域，配置是否开启指标伸缩和定时伸缩，从而满足应用在不同负载下的需求。本案例设置为指标伸缩。

调度配置

您可设置升级方式、节点亲和性、应用亲和性、应用非亲和性和调度容忍，本案例无需设置。

标签和注释

为该Pod添加一个标签，为该Pod添加一个注解。

11. 设置完成后，单击创建。

<#>

阿里云

使用镜像快速创建有状态StatefulSet应用

镜像创建

应用基本信息

容器配置
(数据卷)

高级配置

创建并查看

12. 创建成功后，默认进入创建完成页面，会列出应用包含的对象，您可以单击[查看应用详情](#)进行查看。默认进入有状态副本集详情页面。

名称	状态	镜像
nginx-0	运行中	nginx
nginx-1	运行中	nginx
nginx-2	运行中	nginx

• 验证服务伸缩性：选择所需的Nginx应用，单击右侧伸缩。

- 在弹出的对话框中，将容器组数量设置为3，您可发现扩容时，扩容容器组的排序依次递增；反之，进行缩容时，先按Pod次序从高到低进行缩容。这体现StatefulSet中Pod的次序稳定性。
- 单击左侧导航栏中的[存储卷](#) > [存储声明](#)，您可发现，随着应用扩容，会随着Pod创建新的云存储卷。缩容后，已创建的PV/PVC不会删除。

<#>

阿里云

本节课小结及回顾

本章节主要讲述内容：

1. 容器服务Kubernetes版ACK入门操作使用流程
2. 容器服务Kubernetes版ACK集群创建基础操作指引
3. 基于ACK集群使用镜像快速创建无状态应用基础操作指引
4. 基于ACK集群使用镜像快速创建有状态StatefulSet应用基础操作指引

阿里云云原生容器ACP认证培训

阿里云容器服务ACK基础与进阶



课程目标

学习完本课程后，你将能够：

1. 掌握阿里云容器服务ACK容器网络技术实现方式，以及容器网络规划与部署；
2. 掌握阿里云容器服务ACK存储管理技术实现方式，以及集群存储规划与部署；
3. 掌握阿里云容器服务ACK可观测性体系，以及容器服务日志与监控管理解决方案；
4. 掌握阿里云容器服务ACK弹性伸缩技术实现方式，以及弹性伸缩实践操作；
5. 掌握阿里云容器服务ACK集群管理功能及实践操作；
6. 掌握阿里云容器服务ACK安全管理体系；
7. 掌握阿里云容器服务应用与发布管理，以及灰度发布与基于Helm的发布管理实践。



课程目录

1. 容器服务Kubernetes版ACK基础

1.1 容器网络管理

1.2 存储管理

1.3 日志管理

1.4 监控管理

1.5 弹性伸缩

2. 容器服务Kubernetes版ACK进阶

<#>

阿里云

容器集群网络设计目标

- 集群网络系统是 Kubernetes 的核心部分，它需要解决下面四个核心网络问题：

- 容器与容器之间的通信
- Pod与Pod之间的通信
- Pod与Service之间的通信
- 外部世界与Service之间的通信

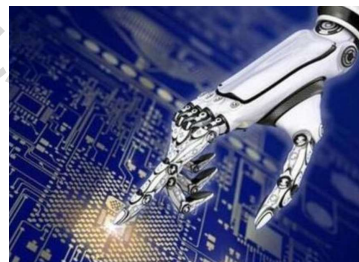


<#>

阿里云

Kubernetes网络接入的三个基本原则

- 在容器网络接入进来，需要满足如下基本原则：
 - Pod无论运行在任何节点都可以互相直接通信，而不需要借助NAT地址转换实现。
 - Node与Pod可以互相通信，在不限制的前提下，Pod可以访问任意网络。
 - Pod拥有独立的网络栈，Pod看到自己的地址和外部看见的地址应该是一样的，并且同一个Pod内所有的容器共享同个网络栈。

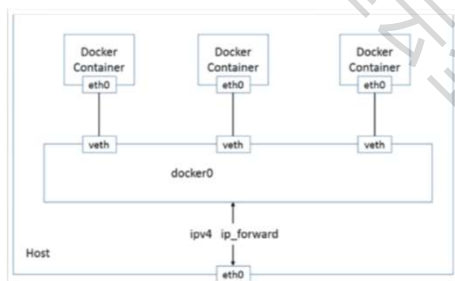


<#>

阿里云

容器到容器之间的通信

- 在容器中，容器之间的网络通信实现是通过docker0网桥，凡是连接到docker0的容器，就可以通过它来进行通信。
- 要想容器能够连接到docker0网桥，我们也需要类似网线的虚拟设备Veth Pair来把容器连接到网桥上。



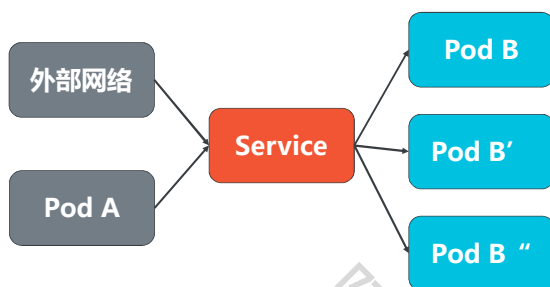
- 当容器在一台宿主机上，访问该宿主机的容器的IP地址时，这个请求的数据包：
 - 先根据路由规则到达docker0网桥
 - 然后，被转发到对应的Veth Pair设备
 - 最后，出现在容器里

<#>

阿里云

通过Service实现内外部统一访问

- 在Kubernetes集群中为什么需要服务发现（Service）？
 - Kubernetes集群应用是通过 Pod部署的，我们知道 Pod 的生命周期过程中，随时可能销毁或者重新部署变化
 - 应用创建Pod组需要统一访问入口以及负载均衡。

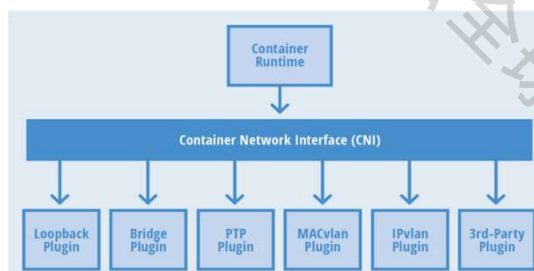


- 应用服务需要暴露到外部去访问，需要提供给用户通过外部网络进行访问
- Pod之间的网络通信也可以通过Kubernetes Service实现访问
- Kubernetes Service可以实现访问负载到一组Pod上面。

<#>

跨主机网络通信

- 在Docker的默认配置下，不同宿主机上的容器通过IP地址进行互相访问是根本做不到的。
- Kubernetes为了更好的控制网络的接入，推出了CNI即容器网络的API接口。
 - 它是Kubernetes中标准的一个调用网络实现的接口
 - kubelet通过这个API来调用不同的网络插件以实现不同的网络配置，实现了这个接口的就是CNI插件，它实现了一系列的CNI API接口。
 - 目前比较常用的Terway、Flannel、Calico、Weave等等。



- CNI，是Kubernetes与底层网络插件之间的一个抽象层，为k8s屏蔽了底层网络实现的复杂度，同时解耦了k8s的具体网络插件实现。
- CNI，目前是Kubernetes官方社区推荐的容器网络方案。

<#>

Kubernetes集群网络插件方案

- 当前Kubernetes集群中使用较多的几种网络方案：

- **Flannel:** Flannel是最早CoreOS团队开源的网络插件，用于让集群中不同节点创建的容器都具有集群内全局唯一的网络(集群外无法感知)，也是当前Kubernetes开源方案中比较成熟的方案，支持HostGW和VXLAN模式
- **Calico:** Calico是一个纯3层的数据中心网络方案，支持IPIP和BGP模式，后者可以无缝集成OpenStack这种IaaS云架构，能够提供可控的VM、容器、裸机之间的IP通信。但是，需要网络设备对BGP的支持(阿里云vpc子网内应该是不支持BGP的)。同时，可以支持基于iptables的网络策略控制
- **Contiv:** Contiv是思科开源的用于跨虚拟机、裸机、公有云或私有云的异构容器部署的开源容器网络架构，可支持2层、3层网络(通常也需要BGP的支持)
- **Terway:** Terway是阿里云开源的基于VPC网络的CNI插件，支持VPC和ENI模式，后者可实现容器网络使用vpc子网网络

<#>

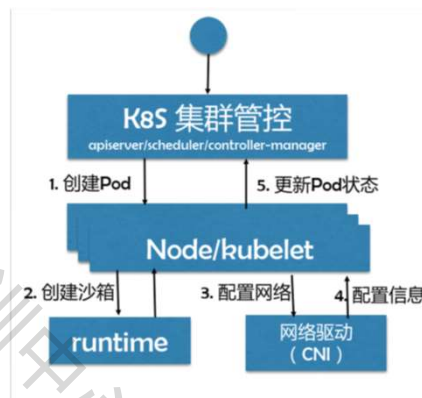


阿里云ACK容器网络技术实现

- CNI是Container Network Interface的缩写

- 它是一个通用的容器网络插件的Kubernetes网络接口。
- 开源社区里已经有了很多实现容器网络的方案，不同的网络实现方案在k8s内都是以插件调用的形式工作，所以这里需要一个统一的标准接口。

- Kubernetes使用CNI接口调用网络插件
- 网络插件(CNI)能分配且唯一的IP地址
- 网络插件(CNI)可以配置Pod的网络和打通Pods之间的访问



<#>

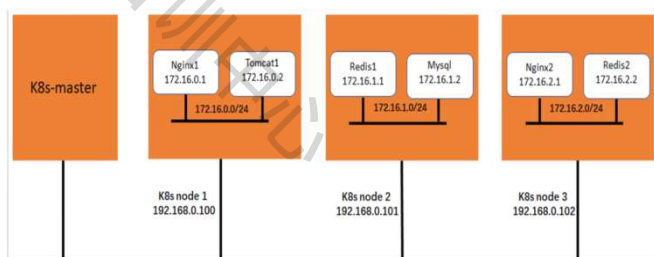


阿里云容器服务ACK容器网络模式

- 容器服务将Kubernetes网络、阿里云VPC、阿里云SLB进行深度集成，提供了稳定高性能的容器网络。

在容器服务中，支持以下类型的互联互通：

- 同一个容器集群中，Pod之间相互访问
- 同一个容器集群中，Pod访问Service
- 同一个容器集群中，ECS访问Service
- Pod直接访问同一个VPC下的ECS
- 同一个VPC下的ECS直接访问Pod



<#>

阿里云容器服务ACK容器网络规划与实现

- 在创建ACK Kubernetes集群时，您需要指定专有网络VPC、虚拟交换机、Pod网络CIDR（地址段）和Service CIDR（地址段）。
- 建议您提前规划ECS地址、Kubernetes Pod地址和服务地址。

VPC网络规划

- 规划专有网络VPC
- 创建专有网络
- 指定创建虚拟交换机



容器网络规划

- Kubernetes网段规划
- 选择Terway网络模式和Flannel网络模式
- Pod虚拟交换机（Terway网络模式）配置
- Pod网络CIDR（Flannel）配置



服务与路由规划

- 创建Service
- Service服务类型选择
- 创建路由Ingress
- 路由Ingress配置

<#>

阿里云容器服务ACK容器网络规划注意事项

• VPC网段

- 在创建VPC选择的地址段。只能从10.0.0.0/8,172.16.0.0/12,192.168.0.0/16三者当中选择一个。

• 交换机网段

- 在VPC里创建交换机时指定的网段，必须是当前VPC网段的子集（可以跟VPC网段地址一样，但不能超过）。
- 交换机下面的ECS所分配到的地址，就是从这个交换机地址段内获取的。
- 一个VPC下，可以创建多个交换机，但交换机网段不能重叠。

• Pod地址段

- Pod是Kubernetes内的概念，每个Pod具有一个IP地址。
- 在阿里云容器服务上创建Kubernetes集群时，可以指定Pod的地址段，不能和VPC网段重叠。

• Service地址段

- Service也是Kubernetes内的概念，每个Service有自己的地址。
- Service地址段也不能和VPC地址段重合，而且Service地址段也不能和Pod地址段重合。
- Service地址只在Kubernetes集群内使用，不会出集群。

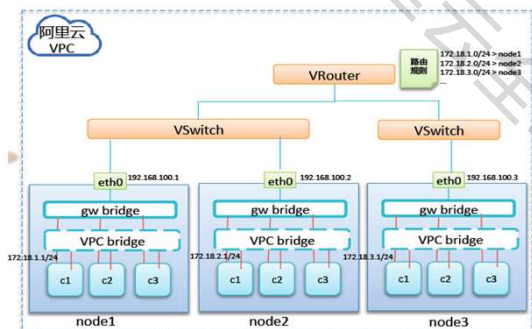
<#>



容器网络规划-VPC专有网络

• 专有网络VPC帮助您基于阿里云构建出一个隔离的网络环境

- 可以完全掌控自己的虚拟网络，包括选择自有IP地址范围、划分网段、配置路由表和网关等。
- 容器服务通过配置VPC路由表的方式将容器对容器的访问转发到容器IP网段所对应的ECS机器上。



- 容器服务是依赖VPC的路由表做容器IP到ECS的流量转发。
- 在VPC的路由表配置中，我们可以看到容器服务配置的网段到ECS的配置，这个是容器服务自动完成的，如果配置不小心被删除掉了，可以对照节点上的docker info找到本节点上对应的网段，手动恢复到VPC的路由表中。

<#>



容器网络规划-VPC专有网络规划

- 阿里云容器网络与阿里云VPC专有网络深度集成
 - 在创建阿里云容器集群之前，首先必须创建专有网络VPC和制定虚拟交换机
 - 需要结合具体的业务来规划VPC和交换机的数量及网段等

集群节点规模	目的	VPC规划	可用区
小于100个节点	一般性业务	单VPC	1个
任意	需要多可用区	单VPC	2个及以上
任意	对可靠性有极致要求、需要多地域	多VPC	2个及以上

- 可以使用

- 192.168.0.0/16、
- 172.16.0.0/12、
- 10.0.0.0/8

私网网段及其子网作为VPC的私网地址范围。



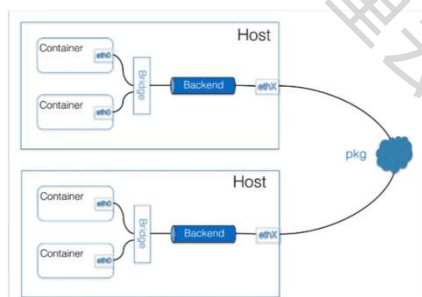
单VPC

地域多VPC

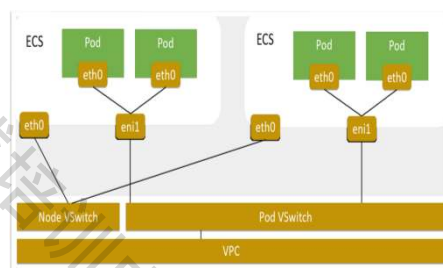
系统隔离多VPC

容器网络规划-容器网络插件

- 在容器服务ACK集群中，可以通过两种网络插件实现上述容器网络的能力：
 - Flannel网络插件
 - Terway网络插件



Flannel网络插件

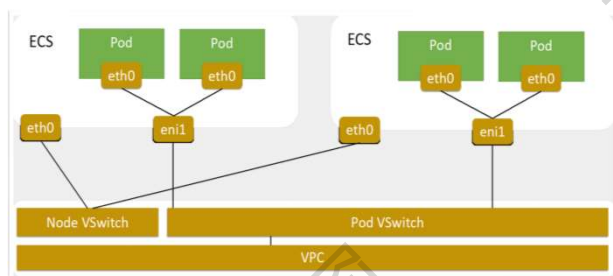


Terway网络插件

容器网络规划- Terway网络插件

• Terway

- 是阿里云开源的基于专有网络VPC的容器网络接口CNI（Container Network Interface）插件
- 支持基于Kubernetes标准的网络策略来定义容器间的访问策略
- 将原生的弹性网卡分配给Pod实现Pod网络
- 支持基于Kubernetes标准的网络策略（Network Policy）来定义容器间的访问策略，并兼容Calico的网络策略。

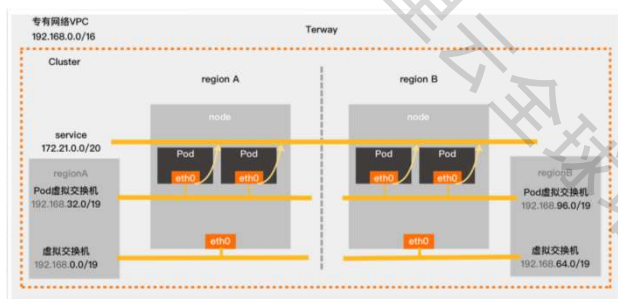


- Terway所提供的VPC互通的网络方案，方便对接已有的基础设施
- 同时，没有overlay网络封包解包的性能损耗，简单易用，出现网络问题方便诊断

<#>

容器网络规划- Terway网络模式

- Terway网络模式采用的是云原生的网络方案
- 直接基于阿里云的虚拟化网络中的弹性网卡资源来构建的容器网络
- Pod会通过弹性网卡资源直接分配VPC中的IP地址，而不需要额外指定虚拟Pod网段。



- 在Terway网络模式中，每个Pod都拥有自己网络栈和IP地址。
- 同一台ECS内的Pod之间通信，直接通过机器内部的转发，跨ECS的Pod通信、报文通过VPC的弹性网卡直接转发。
- 由于不需要使用VxLAN等的隧道技术封装报文，因此Terway模式网络具有较高的通信性能。

<#>

容器网络规划- Flannel与Terway对比

• Flannel

- 使用的是简单稳定的社区的Flannel CNI插件，配合阿里云的VPC的高速网络，能给集群高性能和稳定的容器网络体验
- 但功能偏简单，支持的特性少。

• Terway

- 是阿里云容器服务自研的网络插件，功能上完全兼容Flannel
- 支持将阿里云的弹性网卡分配给容器
- 支持基于Kubernetes标准的NetworkPolicy来定义容器间的访问策略
- 支持对单个容器做带宽的限流。

对于不需要使用Network Policy的用户，可以选择Flannel，其他情况建议选择Terway。

对比项	Terway	Flannel
性能	Pod地址即为VPC中地址，无NAT损耗支持独占ENI模式，几乎无损。	配合阿里云VPC路由，Pod地址为虚拟地址，存在NAT转换损耗。
安全	支持使用网络策略Network Policy。	不支持使用网络策略Network Policy。
地址管理	无需按节点分配地址段，随用随分配，地址无浪费。	节点维度划分地址段，大规模集群下地址浪费多。
SLB	SLB后端直接对接Pod，支持业务无中断升级。	SLB后端不能直接对接Pod，需要通过NodePort转发。

<#>



容器网络规划- Service网络

- 通过服务（Service）抽象，能够解耦前端和后端的关联，从而实现松耦合的微服务设计，以及自动负载均衡实现快速的业务弹性。
- ACK容器服务采用Service方式为一组容器提供固定的访问入口，并对这一组容器做负载均衡。

• 服务访问的方式主要支持类型为：

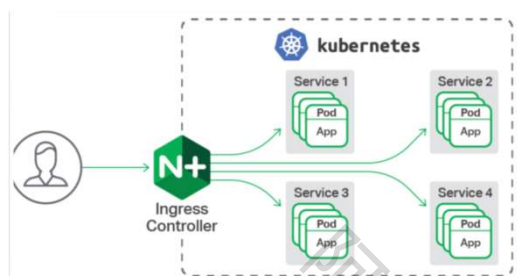
服务访问方式	实现机制
虚拟集群IP（ClusterIP）	通过集群的内部IP暴露服务。选择该值，服务只能够在集群内部可以访问，这也是默认的Service类型。
节点端口（NodePort）	通过每个Node上的IP和静态端口（NodePort）暴露服务。NodePort服务会路由到ClusterIP服务，这个ClusterIP服务会自动创建。通过请求<NodeIP>:<NodePort>，可以从集群的外部访问一个NodePort服务。
负载均衡（LoadBalancer）	指阿里云提供的负载均衡服务（SLB），可选择公网访问或内部访问。阿里云负载均衡服务可以路由到NodePort服务和ClusterIP服务。

<#>



容器网络规划- Ingress

- 在Kubernetes集群中，Ingress作为集群内服务对外暴露的访问接入点，其几乎承载着集群内服务访问的所有流量。
- Ingress是Kubernetes中的一个资源对象，用来管理集群外部访问集群内部服务的方式。
- 通过Ingress资源来配置不同的转发规则，从而达到根据不同的规则设置访问集群内不同的Service后端Pod。

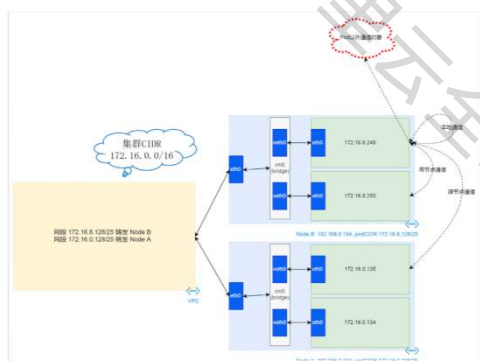


- 阿里云容器服务提供高可靠的Ingress Controller组件，集成了阿里云SLB服务，为您的Kubernetes集群提供灵活可靠的路由服务（Ingress）。

<#>

容器网络规划-容器网络搭建完成

- ACK容器网络搭建包括：
 - 容器网络规划、VPC专有网络配置阶段、容器网络配置阶段、Service网络配置阶段
 - 根据集群业务需求，实现容器网络的配置完成



- 容器与容器之间的通信
- Pod与Pod之间的通信
- Pod与Service之间的通信
- 外部世界与Service之间的通信

<#>

容器网络最佳实践：选用Flannel网络插件实现容器网络

- 本实践在首先创建一个VPC的基础环境（包括一套VPC专有网络、以及三个不同区域的交换机）
- 基于VPC基础环境，创建**Flannel网络模式**类型的ACK容器集群，并在集群上构建一个nginx应用，并实现服务访问。

• 本实践分为三个步骤：

创建专有网络

- 创建VPC专有网络
- 创建交换机

配置容器网络模式

- 创建ACK标准版托管集群
- 配置集群信息（VPC专有网络）
- 配置Worker实例信息（Flannel模式）
- 完成集群创建

创建应用

- 使用镜像创建无状态应用
- 配置应用基本信息
- 选择容器镜像和容器高级配置（Service）
- 完成应用创建

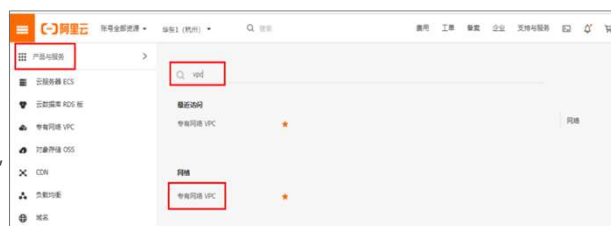
<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

• 1. 创建专有网络

- **步骤1：**登录阿里云管理控制台。
- **步骤2：**通过产品与服务导航，定位到专有网络VPC，单击进入专有网络控制台。
- **步骤3：**在专有网络页面，将地域设置为华东2（上海），并单击创建专有网络。



<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤4：在创建专有网络侧面页面，完成以下配置，并单击确定。

类型	配置项	说明
专有网络	名称	docker_test
	IPv4网段	192.168.0.0/16
交换机	名称	subnet1
	可用区	上海可用区E
交换机	名称	subnet2
	可用区	上海可用区F
交换机	IPv4网段	192.168.1.0/24
	名称	subnet3
交换机	可用区	上海可用区G
	IPv4网段	192.168.1.0/24



<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤5：等待专有网络和交换机创建成功，单击完成。

- 成功创建专有网络。您可以在专有网络列表中看到新建的专有网络（docker_test）。



<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

2. 创建跨可用区普通ACK集群

- 步骤1：登录阿里云管理控制台。
- 步骤2：通过产品与服务导航，定位到容器服务ACK，单击进入容器服务控制台。
- 步骤3：前往集群页面，单击创建集群。



- 步骤4：在选择集群模板页面，单击标准托管集群下的创建。



<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤5：在创建Kubernetes集群页面，完成以下配置，并单击创建集群。

配置项	说明
版本	Kubernetes托管版
集群名称	cluster-standard
地域	华东2（上海）
专有网络	选择docker_test
虚拟交换机	勾选subnet1、subnet2和subnet3
Worker实例	新增实例
节点类型	按量付费
实例规格	x86计算16核64G（ecs.g5.4xlarge）
数量	3台
系统盘	高效云盘120GiB
Kubernetes版本	1.12.6-aliyun.1
操作系统类型	Linux
登录方式	设置密码
登录密码&确认密码	设置符合要求的root账号的密码
网络插件	Flannel
配置SNAT	默认勾选为专有网络配置SNAT
Ingress	默认勾选安装Ingress组件



<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤6：在创建Kubernetes集群页面，完成Worker实例配置，并单击创建集群。

The screenshot displays the 'Worker instance' configuration page in the Alibaba Cloud console. On the left, the 'Instance type' is set to 'ecs.g5-4large' and the 'Image' is 'Alibaba Cloud Linux 2'. The 'Network' section on the right shows the 'VPC' and 'Subnet' selected, with the 'CIDR' block set to '172.20.0.0/16'. The 'Flannel' network plugin is chosen for the cluster.

<#>

阿里云

容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤7：等待集群创建完成。该过程约需10分钟。成功创建集群。
- 您可以在集群列表中看到新建的集群（cluster-standard）

The screenshot shows the 'Cluster List' page in the Alibaba Cloud console. A table lists the created clusters. The first cluster, 'cluster-standard', is highlighted with a red box. It is a 'Kubernetes 托管版' (Managed Kubernetes) cluster in the '华东2' (East China 2) region, using '专有网络 VPC' (Private Network VPC) and is in a '运行中' (Running) state.

集群名称/ID	标签	集群类型	地域 (全部)	网络类型	集群状态	节点个数	创建时间	版本	操作
cluster-standard		Kubernetes 托管版	华东2	专有网络 VPC	运行中	3		1.12.6-aliyun.1	管理 查看日志 控制台 集群扩容 更多

<#>

阿里云

容器网络最佳实践：选用Flannel网络插件实现容器网络

3. 创建应用

- 步骤1：回到容器服务控制台。
- 步骤2：前往应用> 无状态页面，将集群设置为cluster-standard，并单击使用镜像创建。
- 步骤3：填写应用基本信息，并单击下一步。应用基本信息的描述见下表。

置项	说明
应用名称	deploy-standard
集群	cluster-standard
命名空间	default
副本数量	2
类型	无状态

容器网络最佳实践：选用Flannel网络插件实现容器网络

步骤4：完成容器配置。

- 在镜像名称后，单击选择镜像。
- 在镜像选择对话框中，选择nginx镜像，并单击确定。
- 在镜像版本后，单击选择镜像版本。
- 选择latest版本，并单击确定。
- 下一步进行容器高级配置。

容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤5：完成高级配置。
- 在**服务 (Service)** 后，单击**创建**。
- 在创建服务对话框中，完成以下配置，并单击**创建**。

配置项	说明
名称	deploy-standard-svc
类型	虚拟集群IP
端口映射	服务端口80 容器端口80 协议TCP

- 在容器组水平伸缩下，根据您的需要选择是否开启伸缩。
 - 如果是创建正式使用的应用，需要开启伸缩。
 - 本实践简化应用配置，选择不开启伸缩。

<#>



容器网络最佳实践：选用Flannel网络插件实现容器网络

- 步骤6：等待创建应用任务完成。

- 成功创建部署和Service。
- 您可以在集群的无状态应用列表中看到新建的应用（deploy-standard）。

<#>



容器网络最佳实践：选用Terway插件实现容器网络

- 本实践在首先创建一个VPC的基础环境基于VPC基础环境（创建一个专有网络和两个虚拟交换机），
- 在创建ACK容器集群时，选择**Terway网络模式**，并对配置配置Terway网络参数，实现基于Terway插件实现容器集群创建。
- **Terway网络模式规划注意事项：**
 - 如果要使用Terway插件，建议您选择较高规格和较新类型的ECS机型，即5代或者6代的8核以上机型。
- **单节点所支持的最大Pod数取决于该节点的弹性网卡（ENI）数：**
 - 共享ENI支持的最大Pod数 = (ECS支持的ENI数 - 1) × 单个ENI支持的私有IP数
 - 独占ENI支持的最大Pod数 = ECS支持的ENI数 - 1

<#>



容器网络最佳实践：选用Terway插件实现容器网络

- **步骤一：规划和准备集群网络**
 - 在创建ACK Kubernetes集群时，您需要指定专有网络VPC、虚拟交换机、Pod虚拟交换机和服务CIDR（地址段）
 - 如果使用Terway网络插件，您需要先创建一个专有网络VPC，然后在VPC下创建两个虚拟交换机。
 - 这里面虚拟交换机和Pod虚拟交换机需要在一个可用区下。

可以使用以下网段配置，以快速搭建Terway网络

专有网络网段	虚拟交换机	Pod虚拟交换机	Service CIDR
192.168.0.0/16	192.168.1.0/24	192.168.32.0/24	172.21.0.0/20

<#>

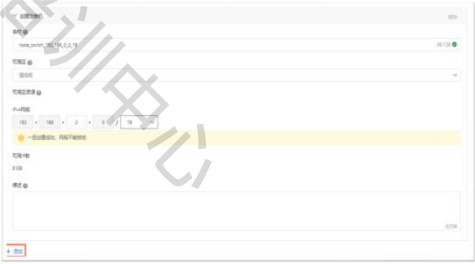


容器网络最佳实践：选用Terway插件实现容器网络

步骤二：专有网络搭建

- 1. 登录专有网络管理控制台。
- 2. 在顶部菜单栏处，选择专有网络的地域，然后单击创建专有网络。
- 3. 创建专有网络和创建交换机：

类型	配置项	说明
专有网络	名称	vpc_192_168_0_0_16
	IPv4网段	192.168.0.0/16
交换机	名称	node_switch_192_168_0_0_19
	可用区	华东1（杭州）可用区I
	IPv4网段	192.168.1.0/24
交换机	名称	pod_switch_192_168_32_0_19
	可用区	华东1（杭州）可用区I
	IPv4网段	192.168.32.0/19



专有网络的地域和要部署的云资源的地域必须相同

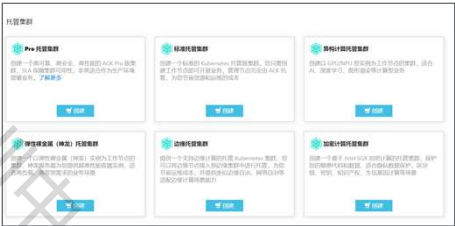
容器网络最佳实践：选用Terway插件实现容器网络

步骤三：创建ACK集群

- 1. 登录阿里云管理控制台。
- 2. 通过产品与服务导航，定位到容器服务ACK，单击进入容器服务控制台。
- 3. 前往集群页面，单击创建集群。



- 4. 在选择集群模板页面，单击标准托管集群下的创建。



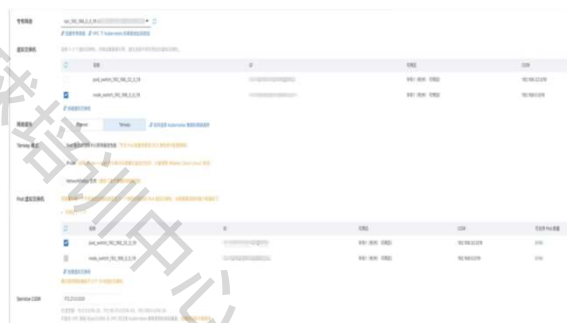
容器网络最佳实践：选用Terway插件实现容器网络

步骤四：配置Terway网络

- 为Terway网络插件配置集群网络的关键参数配置说明如下：

- 专有网络**：选择在步骤1已创建的专有网络名称。
- 虚拟交换机**：选择步骤1创建的虚拟交换机名称。
- 网络插件**：选择Terway。
- 设置网络插件为Terway时，需要配置Terway模式：
 - Terway网络插件，支持独占弹性网卡和Ipvlan（弹性网卡共享模式）两种模式。
 - NetworkPolicy支持，只在弹性网卡共享模式下支持选中，默认不选中。

- Pod虚拟交换机**：选择选择步骤1中创建的虚拟交换机名称。
- Service CIDR**：保留默认值。



独占弹性网卡注意说明 当前只有白名单用户可使用上述Pod独占弹性网卡以获得最佳性能功能。提交工单申请使用。

<#>



容器网络最佳实践：选用Terway插件实现容器网络

步骤五：配置节点实例及创建集群。

- 选择节点实例规格时，需要考虑其支持挂载的ENI数量和相应的Pod 数量。
- 在规格列表中，相应的数据已经列出，绿色数字为该型号在不同 Terway 模式下可支持的 Pod 数量。

- 其他配置基本与Flannel模式集群创建类似，完成配置后，开始创建集群。

- Terway 集群配置与 Flannel 集群相比主要有两点不同：

- Pod 需要通过弹性网卡（ENI）接入虚拟交换机，来组建 Pod 网络。
- 不同 ECS 规格所能支持挂载的弹性网卡数量不同。



<#>



本节课小结及回顾

- 本章节主要讲述内容：
- 集群网络系统是 Kubernetes 的核心部分，它需要解决下面四个核心网络问题：
 - 容器与容器之间的通信、Pod与Pod之间的通信、Pod与Service之间的通信、外部世界与Service之间的通信。
- CNI是一个通用的容器网络插件的Kubernetes网络接口，Kubernetes可以使用CNI接口调用网络插件，CNI可以配置Pod的网络和打通Pods之间的访问。
- 阿里云ACK容器网络规划与实现可以分为三个步骤：
 - VPC网络规划；
 - 容器网络规划；
 - 服务与路由规划。
- 在容器服务ACK集群中，您可以通过两种网络插件实现容器网络的能力
 - Flannel网络插件
 - Terway网络插件

<#>



课程目录

1. 容器服务Kubernetes版ACK基础

1.1 容器网络管理

1.2 存储管理

1.3 日志管理

1.4 监控管理

1.5 弹性伸缩

2. 容器服务Kubernetes版ACK进阶

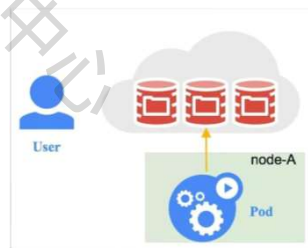
<#>



Kubernetes存储概述

• Kubernetes存储的作用

- 在Kubernetes中最小的管理单元是一个Pod
- Pod中产生的数据都是临时的，当Pod重启时里边的数据会丢失。
- 理想情况下服务应该是无状态的，但实际应用中我们很难做到服务完全无状态化，有些服务一定是有状态的，要想让这些服务能运行在集群中就不得不先解决数据持久化的问题
- 所以，在Kubernetes集群下我们需要一种机制来解决这个问题，这就是Kubernetes存储的作用



<#>

Kubernetes存储应用场景

- Kubernetes中对于存储的使用主要集中在以下几个方面：



- 服务的基本配置文件读取、密码密钥管理等；

- 服务的存储状态、数据存储等；

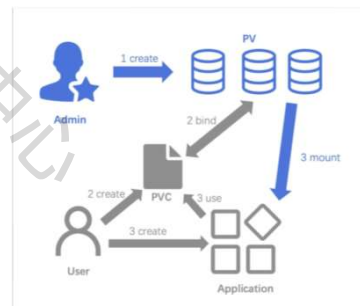


- 不同服务或应用程序间共享数据。

<#>

Kubernetes 存储系统核心概念

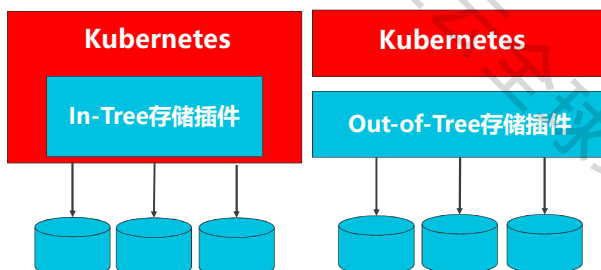
- **Volume**: Kubernetes 支持很多类型的卷。
 - Pod 可以同时使用任意数目的卷类型。
 - 临时卷类型的生命周期与 Pod 相同, 但持久卷可以比 Pod 的存活期长。
 - 因此, 卷的存在时间会超出 Pod 中运行的所有容器, 并且在容器重新启动时数据也会得到保留。
- **持久卷 (简称PV)**: 是集群中由管理员配置的一块存储。
 - 它是集群中的资源, 就和节点是集群资源一样, 包含存储的类型, 存储的大小和访问模式等。
 - PV是卷插件比如Volumes。但是, 它的生命周期独立于使用PV的任何Pod个体。例如, 当前使用它的Pod销毁时, 对PV没有影响。
- **持久卷声明 (简称PVC)**: 是用户关于存储的请求
 - 描述对PV的一个请求, 请求信息包含存储大小、访问模式等, 而PVC消耗PV资源。



<#>

Kubernetes 存储插件

- Kubernetes 可以直接调用某类存储插件, 对接后端存储服务, 现阶段Kubernetes 存储插件约有 20多类。
- Kubernetes 里的存储插件可以分为: In-tree 和 Out-of-tree 这两大类



- **In-Tree类**: 表示源码是放在 Kubernetes 内部的, 和 Kubernetes 一起发布、管理与迭代, 缺点及时迭代速度慢、灵活性差
- **Out-of-Tree 类**: Volume Plugins 的代码独立于 Kubernetes, 它是由存储商提供实现的, 目前主要有 Flexvolume 和 CSI 两种实现机制, 可以根据存储类型实现不同的存储插件。所以我们比较推崇 Out-of-Tree 这种实现逻辑。

<#>

阿里云容器服务ACK存储插件

- 目前阿里云容器服务ACK，支持两种存储插件Flexvolume和CSI。

Flexvolume插件

- Flexvolume插件是Kubernetes社区较早实现的存储卷扩展机制。
 - Flexvolume运行在host 空间，不能使用RBAC授权机制访问Kubernetes API，导致其功能极大的受限。

CSI插件

- CSI插件是当前Kubernetes社区推荐的插件实现方案。
 - ACK集群提供的CSI存储插件兼容社区的CSI特性。
 - CSI不仅仅支持Kubernetes平台存储插件接口，而是作为云原生生态中容器存储接口的标准。

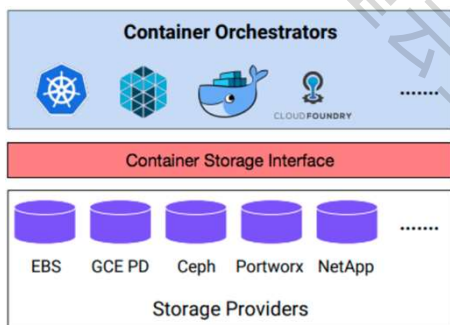
- 针对新建集群，推荐您使用CSI插件。ACK会跟随社区持续更新CSI插件的各种能力。
- 针对已经创建的集群，仍然使用已经安装的存储插件类型。ACK会持续支持Flexvolume插件。

<#>

阿里云

阿里云CSI存储插件

- 阿里云CSI存储插件
 - 遵循标准CSI规范，提供了EBS、NAS、OSS等类型阿里云存储服务的挂载能力。
 - 自ACK 1.16集群开始，部署集群时会默认安装最新版本的CSI组件，可以直接通过CSI存储插件使用阿里云存储服务。
 - CSI存储插件提供了数据卷的全生命周期管理，包括数据卷的：创建、挂载、卸载、删除、扩容等服务。



- CSI不仅仅支持Kubernetes平台存储插件接口，而是作为云原生生态中容器存储接口的标准。
- CSI逐步替换Flexvolume是K8S社区发展的趋势，新建集群中我们更推荐使用CSI类型插件。
- 由于Flexvolume、CSI依赖的Kubelet配置不同，目前ACK环境只能在创建集群时选择使用其中一种插件，暂不支持插件类型混用或者转换。

<#>

阿里云

阿里云CSI存储插件特性

- CSI插件是当前Kubernetes社区推荐的插件实现方案。ACK集群提供的CSI存储插件兼容社区的CSI特性。CSI插件包括以下两部分：
 - CSI-Plugin：实现数据卷的挂载、卸载功能。ACK默认提供云盘、NAS、OSS三种存储卷的挂载能力。
 - CSI-Provisioner：实现数据卷的自动创建能力，目前支持云盘、NAS两种存储卷创建能力。

阿里云存储	静态数据卷	动态数据卷
阿里云云盘	云盘静态存储卷支持使用CSI驱动进行挂载。通过PV和PVC方式。	支持
阿里云NAS	NAS静态存储卷支持使用CSI驱动进行挂载。通过PV和PVC方式。	支持
阿里云OSS	OSS静态存储卷支持使用CSI驱动进行挂载。通过PV和PVC方式。	不支持

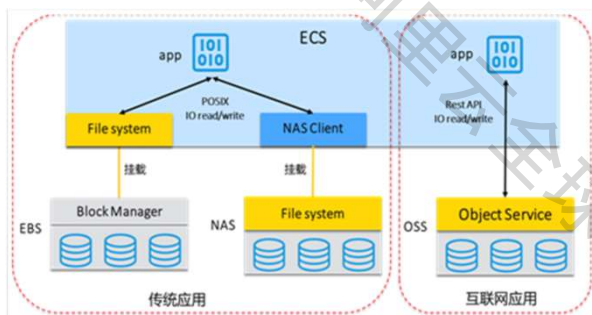
- 容器服务Kubernetes版支持在Kubernetes集群通过Pod自动绑定阿里云云盘、NAS、OSS存储服务。
- 目前CSI驱动支持静态存储卷和动态存储卷。

<#>



阿里云CSI存储插件-如何选用NAS、OSS和EBS

- 容器服务Kubernetes版，支持在Kubernetes集群通过Pod自动绑定三种存储形态：阿里云云盘、NAS、OSS存储服务。



- 阿里云云盘**是为云服务器ECS提供的块级别的数据存储产品，块存储的优势是性能高，时延低，适合于OLTP数据库、NoSQL数据库、等IO密集型的高性能、低时延应用工作负载。
- 文件存储NAS**：您无需修改应用，即可直接像访问本地文件系统一样访问文件存储NAS。因此文件存储NAS多用于多计算节点，无状态集群的共享数据访问。
- 对象存储OSS**是比较新的存储类型，相对于文件存储目录树的组织形式，对象存储OSS采用扁平的文件组织形式，采用RESTful API接口访问，不支持文件随机读写，主要适用于互联网架构的海量数据的上传下载和分发。

<#>



阿里云CSI存储插件-通过控制台使用云盘静态存储卷

- 前提条件
 - 您已经创建好一个Kubernetes集群，并且在该集群中部署CSI插件。
 - 您已经创建好一个按量付费的云盘。请参见创建云盘。

- 创建PV
 1. 登录容器服务管理控制台。
 2. 在控制台左侧导航栏中，单击集群。
 3. 在集群列表页面中，单击目标集群名称或者目标集群右侧操作列下的详情。
 4. 在集群管理页左侧导航栏中，单击存储卷。



<#>



阿里云CSI存储插件-通过控制台使用云盘静态存储卷

5. 在存储卷页签，单击右上角的创建。
6. 在创建存储卷对话框中配置参数。

配置项	说明
存储卷类型	支持云盘、NAS、OSS三种云存储类型。本文中选择为云盘。
存储驱动	支持Flexvolume和CSI。本文中选择为CSI。
访问模式	默认为ReadWriteOnce
云盘ID	您可以选择与集群属于相同地域和可用区下处于待挂载状态的云盘。
文件系统类型	您可以选择以什么数据类型将数据存储到云盘上，支持的类型包括ext4、ext3、xfs、vfat。默认为ext4。
标签	为该数据卷添加标签。



7. 参数配置完成后，单击创建。

<#>



阿里云CSI存储插件-通过控制台使用云盘静态存储卷

创建PVC

- 1. 登录容器服务管理控制台。
- 2. 在控制台左侧导航栏中，单击**集群**。
- 3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
- 4. 在集群管理页左侧导航栏中，单击**存储卷**。
- 5. 在**存储声明**页签，单击右上角的**创建**。
- 6. 在**创建存储声明**对话框中，配置参数如下表：

配置项	说明
存储卷类型	支持云盘、NAS、OSS三种云存储类型。本文中选择为云盘。
名称	创建的数据卷的名称，数据卷名在命名空间内必须唯一。
分配模式	本文中选择已有存储卷
分配模式	您可以选择与集群属于相同地域和可用区下处于待挂载状态的云盘。
已有存储卷	单击选择已有存储卷，在目标存储卷右侧操作列单击选择，选择存储卷。
总量	所创建存储卷的容量。
访问模式	默认为ReadWriteOnce。

阿里云CSI存储插件-通过控制台使用云盘静态存储卷

创建应用

- 1. 登录容器服务管理控制台。
- 2. 在控制台左侧导航栏中，单击**集群**。
- 3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**的**详情**。
- 4. 在集群管理页左侧导航栏中，单击**工作负载**。
- 5. 在**无状态**页签中，单击**使用镜像创建**。
- 6. 在**应用基本信息**页签中，设置应用的基本信息

配置项	描述
应用名称	设置应用的名称。
命名空间	设置应用部署所处的命名空间，默认使用default命名空间。
副本数量	即应用包含的Pod数量，默认数量为2。
类型	定义资源对象的类型，可选择 无状态 、 有状态 、 任务 、 定时 、 定时任务 、 守护进程集 。
标签	为该应用添加一个标签，标识该应用。
注解	为该应用添加一个注解（annotation）。
时区同步	容器与节点是否使用相同的时区。

阿里云CSI存储插件-通过控制台使用云盘静态存储卷

- 单击**下一步**。进入**容器配置**页面。

- 设置容器配置。

数据卷支持配置本地存储和云存储。

- **本地存储**：支持主机目录（HostPath）、配置项（ConfigMap）、保密字典（Secret）和临时目录，将对应的挂载源挂载到容器路径中。

- **云存储**：支持云存储类型。

- 本例中配置了一个云盘类型的数据卷，将该云盘挂载到容器中/tmp路径下，在该路径下生成的容器数据会存储到云盘中。



- 所有的信息都配置完成后，单击**创建**。创建成功后，您就可以正常使用数据卷。

<#>

阿里云

本节课小节及回顾

- **本章节主要讲述内容：**

- Kubernetes 存储系统核心概念：

- Volume
- 持久卷（简称PV）
- 持久卷声明（简称PVC）

- Kubernetes 里的存储插件可以分为：In-tree 和 Out-of-tree 这两大类

- 阿里云容器服务ACK集群支持两种存储插件模式：Flexvolume和CSI

- 容器服务Kubernetes版支持在Kubernetes集群通过Pod自动绑定三种存储形态：

- 阿里云云盘
- NAS
- OSS存储服务

<#>

阿里云

课程目录

1. 容器服务Kubernetes版ACK基础

1.1 容器网络管理

1.2 存储管理

1.3 日志管理

1.4 监控管理

1.5 弹性伸缩

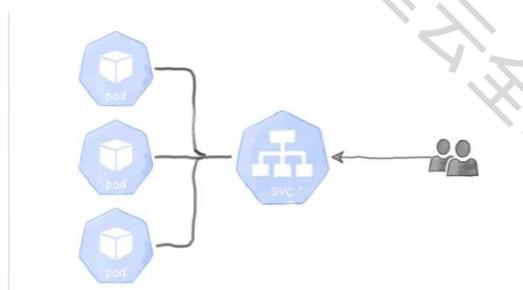
2. 容器服务Kubernetes版ACK进阶

<#>



Kubernetes可观测性体系概述

- 可观测性指如何从外部输出推断及衡量系统内部状态
- Kubernetes可观测性体系包含：监控和日志两部分，是大型分布式系统的重要基础设施
 - 监控可以帮助开发者查看系统的运行状态
 - 日志可以协助问题的排查和诊断



- 在 Kubernetes 中，监控和日志属于生态的一部分，它并不是核心组件，因此大部分的能力依赖上层的云厂商的适配。
- Kubernetes 定义了介入的接口标准和规范，任何符合接口标准的组件都可以快速集成。

<#>



阿里云ACK可观测性体系概述

- **Logging – 日志**

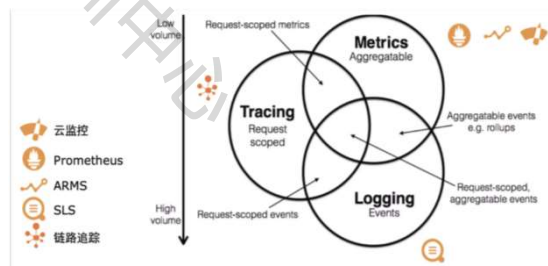
- 基于阿里云日志服务 SLS 提供了完整的日志方案，可以对应用日志进行收集、处理，并且提供了操作审计， Kubernetes 事件中心等能力。

- **Metrics – 监控指标**

- 对基础设施服务，比如 ECS、存储，网络，云监控提供了全面的监控。
- 对于业务应用的性能指标，ARMS无需修改业务代码即可对 Java 和 PHP 应用提供全方位的性能监控。

- **Tracing – 全链路追踪**

- Tracing Analysis 为开发者提供了完整的分布式应用调用链路统计、拓扑分析等工具。
- 能够帮助开发者快速发现和诊断分布式应用中的性能瓶颈，提升微服务应用的性能和稳定性。



<#>

阿里云

Kubernetes日志使用场景

- **Kubernetes日志使用场景主要分为四个大的场景：**



- **主机内核的日志**

主机内核日志可以协助开发者进行一些常见的问题与诊断：网络栈异常、驱动异常、文件系统异常、影像节点（内核）稳定的异常。



- **Runtime 的日志**

比较常见的是 Docker 的一些日志，我们可以通过 docker 的日志来排查类似删除一些 Pod Hang 这一系列的问题。



- **核心组件的日志**

APIServer日志可以用来审计，Scheduler日志可以诊断调度，etcd日志可以查看存储状态，Ingress日志可以分析接入层流量。



- **部署应用的日志**

可以通过应用日志分析查看业务层的状态，诊断异常。

<#>

阿里云

Kubernetes日志的采集

- 从采集位置上划分，主要支持如下三种：

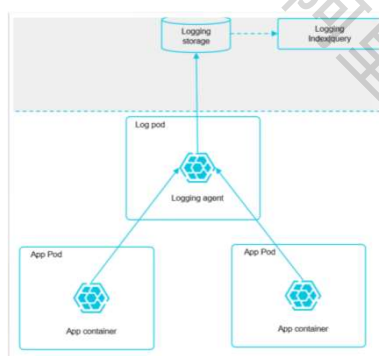


<#>

阿里云

阿里云ACK日志服务

- 阿里云容器服务Kubernetes集群集成了日志服务
 - 可在创建集群时启用日志服务，快速采集Kubernetes集群的容器日志，包括容器的标准输出以及容器内的文本文件。



集群日志服务

- 您可以在创建集群时勾选**使用日志服务**，启用Logtail组件；
- 也可以为已有集群安装Logtail组件，来进行开启日志服务；

应用配置日志服务

- 您可以在创建应用的同时配置日志服务，从而对容器的日志进行采集。
- 目前支持控制台向导和YAML模板两种方式创建应用。

<#>

阿里云

通过日志服务采集Kubernetes容器日志-创建集群启用

• 创建集群启用日志服务

- 登录容器服务管理控制台，在集群列表页面中，单击页面右上角的创建集群。
- 在组件配置中，选中日志服务配置项，表示在新建的Kubernetes集群中安装日志插件
- 当勾选使用日志服务后，会出现创建Project（日志服务管理日志的组织结构，具体可见项目）的提示

• 目前有两种方式可选：

- 选择一个现有的Project来管理采集的日志。

自动创建一个新的Project来管理采集的日志

Project会自动命名为k8s-log-{ClusterID}，ClusterID表示您新建的Kubernetes集群的唯一标识。

- 配置完成后，日志服务就会自动化采集Kubernetes容器日志相关日志。

通过日志服务采集Kubernetes容器日志-已有集群启用

• 为已有集群启用日志服务

1. 登录容器服务管理控制台，在控制台左侧导航栏中，单击**集群**。
2. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。
3. 在集群管理页左侧导航栏中，选择**运维管理 > 组件管理**，并在可选组件区域找到**logtail-ds**。
4. 在**logtail-ds**组件右侧，单击**安装**

可选组件	组件	当前版本	可升级版本	状态	操作
	acorn		2.7.1-4a8c95b-aliyun		安装
	age-metrics-collector		v1.0.0.2-c33b2d5-aliyun		安装
	AutoScaler		v1.3.1-7369cf1		安装
	kube-flannel-ds	v0.11.0.1-g5e46593e-aliyun	v0.11.0.1-g5e46593e-aliyun		无更新
	logtail-ds		v0.16.32.0-40089d3-aliyun		安装
	sgx-device-plugin		v1.0.0-5f5b5ef-aliyun		安装
	terway		v1.0.10.139-g14a4f84-aliyun		安装
	terway-enlip		v1.0.10.139-g14a4f84-aliyun		安装

5. 在**安装组件**对话框中单击**确认**。

- 配置完成后，日志服务就会自动化采集Kubernetes容器日志相关日志。

课程目录

1. 容器服务Kubernetes版ACK基础

1.1 容器网络管理

1.2 存储管理

1.3 日志管理

1.4 监控管理

1.5 弹性伸缩

2. 容器服务Kubernetes版ACK进阶

<#>



Kubernetes监控类型

- 从监控类型上划分，在Kubernetes中可以分成四个不同的类型：

- **资源监控**

CPU、内存、网络这种资源类的一个指标，通常这些指标会以数值、百分比的单位进行统计，是最常见的一个监控方式。这种监控方式在常规的监控里面都是可以做到的。

- **性能监控**

应用的内部监控，通常是通过Hook 的机制在虚拟机层、字节码执行层通过隐式调用，或者是在应用层显示注入，获取更深层次的一个监控指标，一般是用来应用的调优和诊断的。

- **安全监控**

安全监控主要是对安全进行一系列的监控策略，类似像越权管理、安全漏洞扫描等。

- **事件监控**

事件监控是 K8s 中比较另类的一种监控方式，紧密贴合Kubernetes的设计理念，补充常规监控方案的欠缺与弊端。

<#>



阿里云容器服务ACK监控管理

- 阿里云容器服务Kubernetes提供了云原生标准的监控接口
 - 将监控的数据消费能力进行了标准化解耦
 - 可以实现和不同云厂商提供的监控服务进行标准化接入
 - 进行目前已经实现多个监控云服务打通接入。



容器服务ACK可观测性组件-基础资源监控CloudMonitor

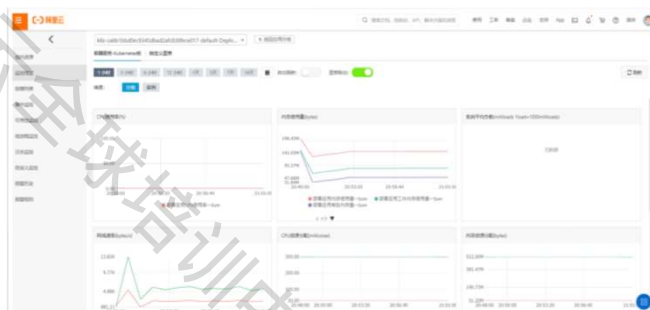
- **云监控 (CloudMonitor)**
 - 是一项针对阿里云资源和互联网应用进行监控的服务。
 - 云监控用于监控各云服务资源的监控指标，探测云服务ECS和运营商站点的可用性，并针对指定监控指标设置报警。
 - 全面了解阿里云上资源的使用情况和业务运行状况，并及时对故障资源进行处理，保证业务正常运行。

- **查看容器监控数据**

您可以查看容器服务Kubernetes版集群监控数据，时刻掌握容器服务的运行情况。

- **管理报警规则**

您可以为容器服务Kubernetes版的集群设置报警规则，便于您及时了解容器服务的异常并快速进行处理。



容器服务ACK观测性组件-SLS日志服务

- 阿里云SLS 主要负责日志的采集、分析。
- 在阿里云容器服务 Kubernetes 中，SLS 可以采集三种不同类型的日志：
 - API Server 等核心组件的日志
 - Service Mesh/Ingress 等接入层的日志
 - 应用的标准日志
- 除了采集日志的标准链路外，SLS 还提供了上层的日志分析能力
- 默认提供了基于 API Server 的审计分析能力、接入层的可观测性展现、应用层的日志分析。



<#>

阿里云

容器服务ACK观测性组件-应用实时监控服务 ARMS

- 应用实时监控服务 ARMS(Application Real-Time Monitoring Service)
 - 是一款阿里云应用性能管理(APM)类监控产品。
 - 可以基于前端、应用、业务自定义等维度，迅速便捷地为企业构建秒级响应的业务监控能力。
- ARMS 产品提供一系列监控定制功能，包括：
 - 数据接入、数据计算、数据存储、大盘展示和报警，以及下游 API 对接等环节。
- 用户只需要关心三件事：
 - 数据从哪里来；
 - 实时计算流程怎么编排；
 - 数据该怎么用。

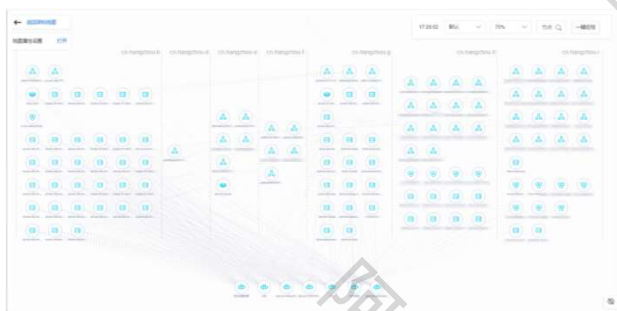


<#>

阿里云

容器服务ACK可观测性组件-架构感知监控AHAS

- AHAS 是架构感知监控，通常在 Kubernetes 集群中负载的类型大部分为微服务，微服务的调用拓扑也会比较复杂，因此当集群的网络链路出现问题时，如何快速定位问题、发现问题、诊断问题则成为了最大的难题。
- AHAS 通过网络的流量和走向，将集群的拓扑进行展现，提供更高层次的问题诊断方式。



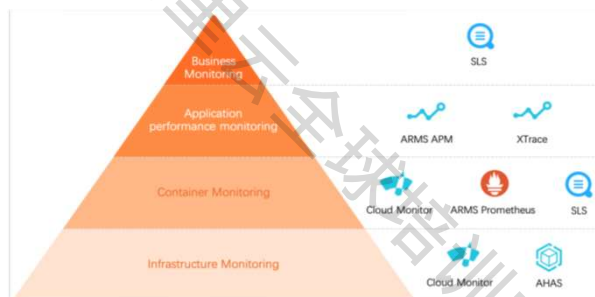
- 为Kubernetes应用安装AHAS探针后，AHAS能自动识别系统中的Pod、Deployment、Service和它们与其他组件的依赖关系。
- 通过采集Kubernetes元信息，构建Kubernetes资源的物理拓扑关系，支持自建Kubernetes集群、阿里云容器服务Kubernetes集群。

<#>

阿里云

阿里云容器服务ACK监控生态解决方案

- 从可观测性的角度，以ACK为基础的系统架构可以粗略分为4个层次，自下而上分别是：
 - 基础设施层
 - 容器性能层
 - 应用性能层
 - 用户业务层



用户业务层场景的可观测性

应用性能层场景的可观测性

容器性能层场景的可观测性

基础设施层场景的可观测性

<#>

阿里云

阿里云容器服务ACK监控方案详解（1/2）

名称	解决方案	方案介绍	适用场景
基础设施层可观测性	架构可视化感知方案	Kubernetes集群中的业务是运行在节点组成的资源池上，使得定位Pod的调用链路以及拓扑关系非常复杂。那么如何更好地可视化集群中流量的吞吐是非常重要的问题。推荐使用架构感知AHAS云产品提供产品能力。	适用全部场景
	基础设施指标监控方案	资源监控是Kubernetes中最常见的底层资源监控方式。通过资源监控可以快速查看负载的CPU、内存、网络等指标的使用率。在阿里云容器服务中，资源监控已经与云监控互通，新建的集群默认安装与集成云监控（Cloud Monitor）。	适用全部场景
集群、容器的性能指标监控	云监控容器服务ACK的监控方案	阿里云容器服务ACK新建的集群默认安装与集成云监控（Cloud Monitor），提供集群、容器的部分性能指标监控，并集成在容器服务控制台中展示。	适用部分场景，定制化提供基础的容器层性能指标可观测能力。
	阿里云托管版Prometheus监控方案	Prometheus也是社区官方的容器场景云原生指标可观测方案。阿里云Prometheus监控全面对接开源Prometheus生态，支持类型丰富的组件监控，提供多种开箱即用的预置监控大盘，且提供全面托管的Prometheus服务。借助阿里云Prometheus监控，您无需自行搭建Prometheus监控系统，因而无需关心底层数据存储、数据展示、系统运维等问题。推荐使用阿里云托管版Prometheus（ARMS Prometheus）云产品。	适用全部场景，包括微服务（ServiceMesh）场景、集群如自身组件指标，以及定制监控能力等高级可观测能力。
	开源Prometheus监控方案	阿里云容器服务在应用市场中提供了开源Prometheus监控方案的集成。	适用全部场景，包括微服务（ServiceMesh）场景、集群如自身组件指标等场景。

<#>



阿里云容器服务ACK监控方案详解（2/2）

名称	解决方案	方案介绍	适用场景
集群、容器事件监控	事件的监控方案	事件监控是Kubernetes中的站在事件角度的另一种监控方式，可以弥补资源监控在实时性、准确性和场景上的缺欠。开发者可以通过获取事件，实时诊断集群的异常与问题。推荐使用阿里云日志服务 SLS产品提供的事件中心监控能力。	适用全部场景。
应用性能层可观测性	无侵入应用监控APM监控方案	推荐使用阿里云应用性能监控ARMS作为应用性能层监控方案。只要为部署在容器服务Kubernetes版中的Java应用安装ARMS应用监控组件，您无需修改任何代码，就能借助ARMS对Java、PHP应用进行全方位监控，以便您更快速地定位出错接口和慢接口、发现系统瓶颈，从而大幅提升线上问题诊断问题的效率。	适用部分场景，包括JAVA、PHP应用的应用监控，方案接入支持无侵入方式，无需进行代码改造。
	侵入式应用监控APM监控方案	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，可以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。	适用所有场景，包括微服务（ServiceMesh）以及多种开发语言的应用，方案接入需要侵入式代码引入改造。
用户业务层可观测性	自定义日志监控方案	推荐使用阿里云日志服务SLS（Log Service）作为自定义指标的观测方案。您可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业务大盘，观测自己的业务情况，或做系统审计。	适用全部场景，如流量监控、成本审计统计、业务订单走势统计等。

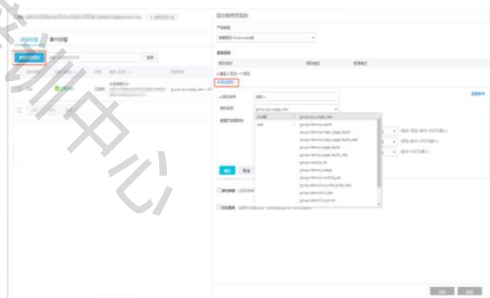
<#>



阿里云容器服务ACK-基础资源监控

- 资源监控是Kubernetes中最常见的监控方式
 - 通过资源监控可以快速查看负载的CPU、内存、网络等指标的使用率。
 - 在阿里云容器服务中，资源监控已经与云监控互通，新建的集群默认安装与集成云监控。

1. 登录**容器服务管理控制台**。
2. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
3. 在集群管理页左侧导航栏中，单击**工作负载**。
4. 在页面上方选择**命名空间**，单击**无状态**页签。
5. 选择所需的Deployment，单击右侧的**监控**，进入到云监控的相应的**监控视图**页面。
6. 您可以单击**维度**中的**分组**或**实例**，分别通过两个维度查看监控数据。
7. **可选**：如需设置告警，您可以在左侧导航栏单击**报警规则**并进行设置，分组级别的指标以**group**开头，实例级别的指标以**pod**开头。



<#>

阿里云

本章小节及回顾

- **本章节主要讲述内容：**
- 阿里云容器服务ACK版可观测性体系包括：
 - Logging-日志、Metrics-监控指标、Tracing-全链路追踪
- 阿里云容器服务ACK版集成了日志服务，在创建集群时启用日志服务，快速采集Kubernetes集群的容器日志，包括容器的标准输出以及容器内的文本文件。
- 阿里云容器服务ACK版提供了云原生标准的监控接口
 - 它将监控的数据消费能力进行了标准化解耦
 - 可以实现和不同云厂商提供的监控服务进行标准化接入
 - 目前已经实现多个监控云服务打通接入
- 阿里云容器服务ACK版监控生态解决方案分为：
 - 基础设施层场景的可观测性、容器性能层场景的可观测性、应用性能层场景的可观测性、用户业务层场景的可观测性

<#>

阿里云

课程目录

1. 容器服务Kubernetes版ACK基础

1.1 容器网络管理

1.2 存储管理

1.3 日志管理

1.4 监控管理

1.5 弹性伸缩

2. 容器服务Kubernetes版ACK进阶

<#>



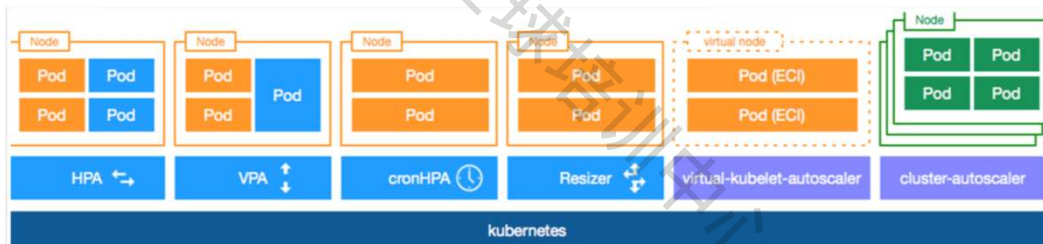
容器服务ACK弹性伸缩概述

- 弹性伸缩是ACK上被广泛采用的功能，典型的场景包含：

- 在线业务弹性
- 大规模计算训练
- 深度学习GPU或共享GPU的训练与推理
- 定时周期性负载变化等

- 弹性伸缩分为两个维度：

- **调度层弹性**：主要是负责修改负载的调度容量变化。
- **资源层弹性**：主要是集群的容量规划不能满足集群调度容量时，会通过扩充资源的方式进行调度容量的补充。

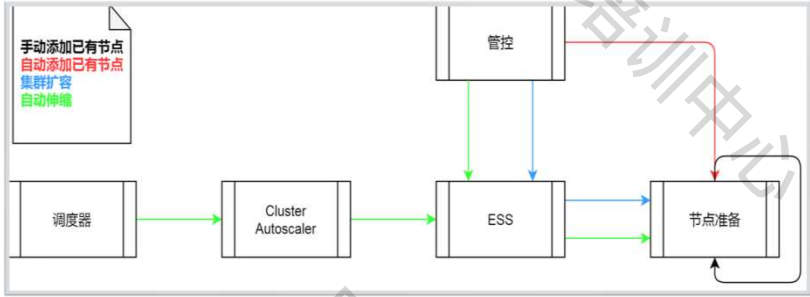


<#>



容器服务ACK弹性伸缩-资源层节点弹性介绍

- 阿里云Kubernetes集群的一个重要特性，是集群的节点可以动态的增加或减少。
- 有了这个特性，集群才能在计算资源不足的情况下扩容新的节点。
- 同时，也可以在资源利用率降低的时候，释放节点以节省费用。



- 资源层扩缩容可以分为：
 - 手动扩缩容
 - 自动扩缩容

容器服务ACK弹性伸缩-资源层弹性组件

组件名称	组件介绍	适用场景	资源交付速度
cluster-autoscaler 节点自动伸缩	Kubernetes社区开源组件，节点水平伸缩组件，阿里云提供了独有的调度、弹性优化、成本优化的功能。	全场景支持，适合在线业务、深度学习、大规模成本算力交付等。	•以1000节点为一个交付批次为例：标准模式：120s •极速模式：60s •标准模式-Qboot镜像：90s •极速模式-Qboot镜像：45s
virtual-node 虚拟节点	ACK开源组件，提供无服务器运行时环境。开发者无需关心节点资源，只需针对Pod按量付费即可。	部分场景支持，主要包括在线突发流量、CI/CD、大数据作业。	•以1000个Pod为一个交付批次为例：未开启镜像缓存：30s •已开启镜像缓存：15s
virtual-kubelet-autoscaler	ACK服务组件，提供无服务弹性伸缩能力。	部分场景支持，主要包括在线突发流量、CI/CD、大数据作业。	•以1000个Pod为一个交付批次为例：未开启镜像缓存：30s •已开启镜像缓存：15s

容器服务ACK资源层弹性-节点自动伸缩机制

- 在ACK容器服务中，节点自动伸缩的工作原理，与传统意义上基于使用率阈值的模型有所差别。
- 这个也是很多开发者在从传统的IDC或者其他编排系统，例如：Swarm等迁移到Kubernetes后最难理解的地方。
 - 传统的弹性伸缩模型**是基于使用率的
 - 例如：一个集群中有3个节点，当集群中的某个节点CPU、内存使用率超过特定的阈值时，此时弹出新的节点。
- Kubernetes节点伸缩**是怎么解决这个问题的呢？
 - Kubernetes**是通过调度与资源解耦的两层弹性模型来解决的。
 - 简单的理解，可以基于资源的使用率来触发应用副本的变化，也就是调度单元的变化。
 - 而当集群的调度水位达到100%的时候会触发资源层的弹性扩容，当资源弹出后，无法调度的单元会自动调度到新弹出的节点上，从而降低整个应用的负载状况。

<#>



容器服务ACK资源层弹性-节点自动伸缩实现原理

- 阿里云容器服务ACK的自动伸缩能力，是通过节点自动伸缩组件（**cluster-autoscaler**）实现的。
- 可以按需弹出普通实例、GPU实例、竞价付费实例，支持多可用区、多实例规格、多种伸缩模式，满足不同的节点伸缩场景。



- 伸缩配置**：通过控制台配置伸缩节点的规格属性。
- 自动伸缩组件监听**：自动伸缩组件cluster-autoscale，它以Pod的形式运行在K8S集群中，监听Pod状态，在Pod因为节点资源不足而不能被调度的状态时，启动自动伸缩规则。
- 伸缩规则启动**：根据“1.伸缩配置”的内容，启动ESS弹性伸缩服务。
- ESS弹性伸缩服务创建实例**：通过ESS弹性伸缩服务按照配置创建实例后，继续进行集群监听。

<#>



容器服务ACK节点自动伸缩操作

- 容器服务ACK节点自动伸缩分为三个步骤：

执行自动伸缩



弹性伸缩服务ESS授权



配置自动伸缩

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面，执行以下操作进入**自动弹性伸缩配置**页面。

您可以通过以下两种路径配置自动弹性伸缩。

路径一： 在目标集群右侧的操作列下，选择**更多 > 自动伸缩**。

路径二：

- ① 选择目标集群然后单击操作列的**详情**。
- ② 在集群应用配置左侧导航栏，选择**节点管理 > 节点池**。
- ③ 在节点池页面右上方，单击**自动弹性伸缩配置**。



<#>



容器服务ACK节点自动伸缩操作

- 容器服务ACK节点自动伸缩分为三个步骤：

执行自动伸缩



弹性伸缩服务ESS授权



配置自动伸缩

对于节点权限被收敛的集群，您需要授权RAM角色**弹性伸缩服务ESS服务**。

1. 开通ESS服务

- ① 单击提示对话框中的第一个链接，进入**弹性伸缩服务ESS**页面。
- ② 开通成功后，在**开通完成**页签，单击**管理控制台**，进入**弹性伸缩服务ESS**页面，并前往授权，同意授权表。



2. 授权角色

- ① 单击弹出对话框中的第二个链接。
- ② 在**云资源访问授权**页面，单击**同意授权**。



<#>



容器服务ACK节点自动伸缩操作

- 容器服务ACK节点自动伸缩分为三个步骤：



1. 在自动伸缩页面，填写以下信息，并单击提交。

配置	说明
集群	目标集群名称。
缩容阈值	cluster-autoscaler管理的伸缩组中，每一个节点的资源申请值（Request / 每一个节点的资源容量。当低于配置的阈值时，节点会进行缩容。 说明 弹性伸缩中，扩容会基于调度自动触发，只需设置缩容条件即可。
缩容触发时延	集群满足配置的缩容阈值时，在配置的缩容触发时延到达后，集群开始缩容。单位：分钟。默认情况下是10分钟。
静默时间	在集群删除节点后，在静默时间内，集群不会再次触发缩容，单位：分钟。默认情况下是10分钟。

2. 根据所需要弹性伸缩的资源类型（普通、GPU、抢占式），单击操作列创建。

<#>



容器服务ACK节点自动伸缩操作

3. 在伸缩组配置填写以下信息。

配置	说明
地域	所创建伸缩组将要部署到的地域。与伸缩组所在集群的地域一致，不可变更。
专有网络	所创建伸缩组的网络，与伸缩组所在集群的网络一致。
虚拟交换机	所创建伸缩组的虚拟交换机，包含伸缩组的可用区及Pod地址段。

5. 单击**确定**，创建伸缩组。

4. 配置Worker节点的信息。

配置	说明
节点类型	所创建的伸缩组的节点类型，与创建集群时所选择的节点类型一致。
实例规格	伸缩组内实例的规格。
已选规格	所选择的伸缩组的实例规格，最多可以选择10种实例规格。
系统盘	伸缩组的系统盘。
挂载数据盘	是否在创建伸缩组时挂载数据盘，默认情况下不挂载。
实例数量	伸缩组所包含的实例数量。 •说明实例不包含客户已有的实例。
密钥对	登录伸缩后的节点时所使用的密钥对。可以在ECS控制台新建密钥对。 说明 目前只支持密钥对方式登录。
伸缩模式	支持 标准模式 和 极速模式 。
RDS白名单	弹性伸缩后的节点可以访问的RDS实例。
节点标签	在集群中添加节点标签（Label）后，会自动添加到弹性伸缩扩容出的节点上。
污点（Taints）	添加污点后，Kubernetes将不会将Pod调度到这个节点上。

<#>



容器服务ACK节点自动伸缩操作

- 查看配置结果

1. 在**自动伸缩**页面，可以在普通实例下看到新创建的一个伸缩组。



2. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。

3. 集群管理页左侧导航栏中，选择**工作负载 > 无状态**。

4. 在**无状态**页签，选择kube-system命名空间，可以看到名称为cluster-autoscaler的组件，表示伸缩组已创建成功。

<#>

阿里云

容器服务ACK弹性伸缩-调度层弹性介绍

- 调度层弹性组件是指所有的弹性动作都是和 Pod 相关的，并不关心具体的资源情况。



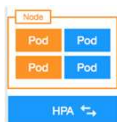
- 调度层弹性组件:

- 容器水平伸缩 (HPA)
- 容器垂直伸缩 (VPA)
- 容器定时伸缩 (CronHPA)

<#>

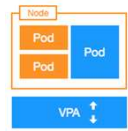
阿里云

容器服务ACK弹性伸缩-调度层弹性组件

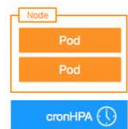


- **HPA**：Pod 水平伸缩的组件

- 除了社区支持的 Resource Metrics 和 CustomMetrics，阿里云容器服务 Kubernetes 还提供了 external-metrics-adapter
- 支持云服务的指标作为弹性伸缩的判断条件
- 目前已经支持，例如：Ingress 的 QPS、RT，RMS 中应用的 GC 次数、慢 SQL 次数等等多个产品不同维度的监控指标。



- **VPA**：Pod 的纵向伸缩的组件，主要面向有状态服务的扩容和升级场景。



- **CronHPA**：定时伸缩组件

- 主要面向的是周期性负载，通过资源画像可以预测有规律的负载周期，并通过周期性伸缩，实现资源成本的节约

<#>



容器服务ACK弹性伸缩-调度层弹性组件适用场景及限制

组件名称	组件介绍	适用场景	使用限制
HPA 水平伸缩组件	Kubernetes内置组件，主要面向在线业务。	在线业务	适用于Deployment、StatefulSet等实现scale接口的对象。
VPA 容器垂直伸缩	开源社区组件，主要面向大型单体应用。	大型单体应用	适用于无法水平扩展的应用，通常是在Pod出现异常恢复时生效。
CronHPA 容器定时伸缩	ACK开源的组件，主要面向应用资源使用率存在周期性变化的场景。	周期性负载业务	适用于Deployment、StatefulSet等，实现了scale接口的对象。此外CronHPA提供了HPA对象的兼容能力，您可以同时使用CronHPA与HPA。

<#>



通过容器服务ACK控制台创建HPA应用

- 阿里云容器服务已经集成了HPA，您可以简便地通过容器服务控制台进行创建。您可以在创建应用的时候创建HPA，也可以在已有应用的基础上开启HPA。

1. 登录**容器服务管理控制台**。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。
4. 在集群管理页左侧导航栏中，单击**工作负载**。
5. 在**无状态**页面单击**使用镜像创建**。
6. 在**应用基本信息**页填写应用的名称，设置应用部署集群和命名空间，单击**下一步**。

配置项	描述
应用名称	设置应用的名称。
命名空间	设置应用部署所处的命名空间，默认使用default命名空间。
副本数量	即应用包含的Pod数量，默认为2。
类型	定义资源对象的类型，可选择 无状态 、 有状态 、 任务 、 定时 、 定时任务 、 守护进程集 。
标签	为该应用添加一个标签，标识该应用。
注解	为该应用添加一个注解（annotation）。
时区同步	容器与节点是否使用相同的时区。



<#>



通过容器服务ACK控制台创建HPA应用

7. 在**容器配置**页进行容器设置，选择镜像，并设置所需的资源。然后单击**下一步**。
8. 在**高级设置**页的**访问设置**区域，单击服务的**创建**，设置服务的相关信息
9. 在**高级设置**页选中**指标伸缩**的**开启**，设置伸缩的条件和配置。

- **指标**：支持CPU和内存，需要和设置的所需资源类型相同。
- **触发条件**：资源使用率的百分比，超过该使用量，容器开始扩容。
- **最大副本数**：该Deployment可扩容的容器数量上限。
- **最小副本数**：该Deployment可缩容的容器数量下限。

10. 单击右下角的**创建**，一个支持HPA的Deployment就已经创建完毕。

• 验证结果：

- 单击**查看应用详情**或者在**工作负载** > **无状态**中单击**应用名称**或**操作列**的**详情**，单击**指标伸缩（HPA）**，您可在部署的详情中查看伸缩组信息。

<#>



本节课小节及回顾

- 本章节主要讲述内容：

- 阿里云容器服务ACK弹性伸缩分为两个维度：

- 调度层弹性：主要是负责修改负载的调度容量变化。
- 资源层弹性：主要是集群的容量规划不能满足集群调度容量时，会通过扩充资源的方式进行调度容量的补充。

- 阿里云容器服务Kubernetes弹性调度模型：

通过调度与资源解耦的两层弹性模型来解决的。当集群的调度水位达到设置的阈值的时候会触发资源层的弹性扩容，当资源弹出后，无法调度的单元会自动调度到新弹出的节点上，从而降低整个应用的负载状况。

- 阿里云容器服务ACK集群调度层弹性组件：

- 容器水平伸缩（HPA）
- 容器垂直伸缩（VPA）
- 容器定时伸缩（CronHPA）

<#>



课程目录

1. 容器服务Kubernetes版ACK基础

2. 容器服务Kubernetes版ACK进阶

- 2.1 集群管理

- 2.2 安全管理

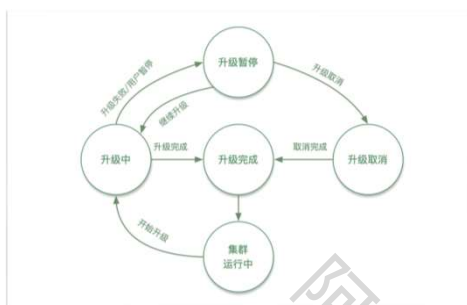
- 2.3 应用与发布管理

<#>



容器服务ACK集群-升级集群

- 可以通过容器服务管理控制台，可视化升级集群的Kubernetes版本
 - 可以在Kubernetes集群列表页面查看集群的Kubernetes版本，以及当前是否有新的版本可供升级。
- 升级集群的过程包含升级前置检查、升级Master（专有版会展示当前正在升级的Master编号）、升级Node（会展示已经升级的节点数和总节点数）。



• 集群升级策略：

- 定义了您将使用怎样的策略对集群进行升级。
- 目前默认策略为分批升级。
- 分批升级会在**升级Node**阶段对集群内的节点进行分批升级。其具体策略为：
 - 第一批升级的节点数为1，后续的批次以2的幂数进行增长。暂停后重新恢复升级的第一批次为1，后续也是以2的幂数进行增长。
 - 每一批节点的最大数量不会超过节点总数的10%。

<#>

容器服务ACK集群-升级集群

- 集群升级前置检查在您开始集群升级之后，ACK会为您自动启动集群升级前置检查。
- 该检查会对集群进行多项健康检查，以确保您的集群可以顺利的完成此次升级。
- 如果您的集群存在不合理配置或者潜在风险，则无法通过前置检查，如下图所示：



<#>

容器服务ACK集群-升级集群

- 单击**查看详情**按钮。跳转到集群运维页面，查看具体的失败原因。



- 集群升级暂停**：通过集群升级暂停功能，您可以在集群升级的任意阶段对其升级进程进行暂停。
- 集群升级取消**：您可以在暂停升级后，单击**取消**，对本次升级进行取消操作。

<#>



容器服务ACK集群-管理维护组件

- 登录**容器服务管理控制台**。
- 在控制台左侧导航栏中，单击**集群**。
- 在**集群列表**页面，选择目标集群，并在目标集群右侧**操作**列下，选择**更多 > 系统组件管理**。
- 在**组件管理**页面，您可以进行以下操作：
 - 选择需要安装的组件，单击列**安装**。
 - 选择需要卸载的组件，单击列**卸载**。
 - 选择需要升级的组件，单击列**升级**。
- 组件主要包含：**
 - 核心组件
 - 应用管理组件
 - 日志与监控组件
 - 其他组件
 - 存储组件
 - 网络组件
 - 安全组件



<#>



容器服务ACK集群-扩容集群

- 通过容器服务管理控制台，您可以根据实际业务需要对Kubernetes集群的Worker节点进行扩容。
 - 登录容器服务管理控制台。
 - 在控制台左侧导航栏中，单击**集群**。
 - 在集群列表页面，选择目标集群，并在目标集群右侧操作列下，选择**更多 > 集群扩容**。
 - 在**集群扩容**页面进行扩容配置。



- 根据业务需求扩容所需要的节点配置
- 配置内容与Worker节点配置一致。

<#>

课程目录

- 容器服务Kubernetes版ACK基础
- 容器服务Kubernetes版ACK进阶
 - 集群管理
 - 安全管理
 - 应用与发布管理

<#>

阿里云容器服务Kubernetes版ACK安全体系

- 阿里云容器服务Kubernetes版安全体系

- 提供从基础设施到上层用户业务应用全链条安全体系支撑
- 支持运行时刻的安全策略管理，应用安全配置巡检和运行时刻的安全监控和告警
- 提升容器安全整体纵深防御能力



- 运行时安全
- 可信软件供应链
- 基础架构安全

<#>

阿里云

ACK安全体系-基础架构安全

- ACK安全体系基础架构安全

- 从公有云、专有云、混合云等多角度覆盖
- 通过集群安全加固、身份验证管理、细粒度安全访问控制、集群日志审计、敏感数据加密等多个环节保障集群基础设施安全稳定高效运行



- 集群默认安全
- 身份管理
- 细粒度访问控制
- 集群审计
- 敏感数据落盘加密

<#>

阿里云

ACK安全体系-可信软件供应链

- 安全是企业应用容器技术时最大的顾虑之一
- 为了系统化地提升容器平台的安全性，就需要全方位进行安全防护
- 阿里云的做法是：
 - 首先，把 DevOps 升级为 DevSecOps
 - 将安全概念融入到整个软件生命周期之中
 - 将安全防护能力转移到开发和交付阶段。



• 镜像扫描

- 容器镜像服务支持所有基于Linux的容器镜像安全扫描，可以识别镜像中已知的漏洞信息，大幅降低使用容器的安全风险。

• 镜像签名

- 在容器镜像管理中，可以通过容器镜像数字签名保障镜像来源的安全性及不被篡改，降低在您的环境中运行意外或恶意代码的风险。

• 云原生应用交付链

- 在容器安全高效交付场景中，可以使用容器镜像服务的云原生应用交付链功能。

ACK安全体系-运行时安全

- 安全巡检
 - ACK集群提供了应用运行时刻的安全配置巡检能力，帮助您实时了解当前状态下运行应用的配置是否存在安全隐患。
- 策略管理
 - 安全策略PSP是Kubernetes中Pod部署时重要的安全校验手段，能够有效地约束应用运行时的行为安全。
- 运行时监控和告警
 - 阿里云容器服务和云安全中心深度集成了告警处理和漏洞检测能力，在应用运行时提供监控和告警能力。
- 安全沙箱管理
 - 安全沙箱为您提供的一种新的容器运行时选项，可以让您的应用运行在一个轻量虚拟机沙箱环境中，拥有独立的内核，具备更好的安全隔离能力。



课程目录

1. 容器服务Kubernetes版ACK基础
2. 容器服务Kubernetes版ACK进阶
 - 2.1 集群管理
 - 2.2 安全管理
 - 2.3 应用与发布管理

<#>

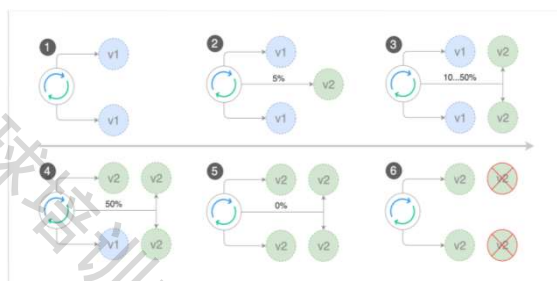
容器服务ACK发布管理- 灰度发布概念

• 灰度发布，又称金丝雀发布

- 是将应用的旧版本A与新版本B同时部署在环境中，业务请求可能会被路由到旧版本A的后端上，也可能被路由到旧版本B的后端上。您可以自定义灰度发布策略，快速调整版本A和B的流量占比。

• 灰度发布可以在发布新版本应用时：

- 自定义控制新版本应用流量比重
- 渐进式完成新版本应用的全量上线
- 最大限度地控制新版本发布带来的业务风险，降低故障带来的影响
- 同时支持快速回滚



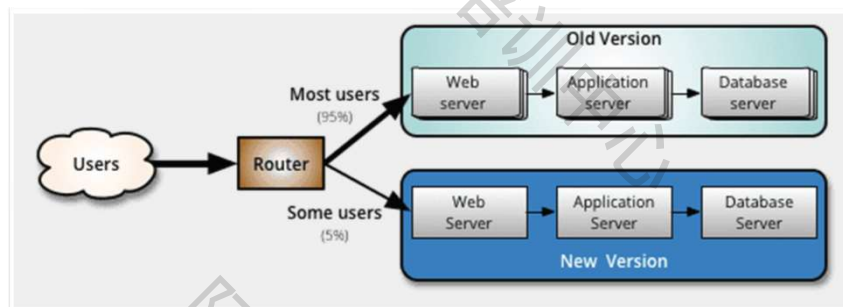
灰度发布流程图

<#>

容器服务ACK发布管理- 灰度发布实现

灰度发布是可以帮助您渐进式更新Deployment的工具

- 它能结合路由控制器（Ingress Controller）实现多版本共存，发布暂停，流量百分比切换，健康检查等功能
- 极大解放灰度发布过程中的手动操作，全自动化实现线上灰度流量切换



<#>

发布管理-安装灰度发布组件

- 在ACK中安装对应的组件后才可以使⤵灰度发布功能。
- 在灰度发布页面，直接通过单击**开始安装**即可实现对应的组件安装。
 1. 登录容器服务管理控制台。
 2. 在控制台左侧导航栏中，单击**集群**。
 3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
 4. 在集群管理页左侧导航栏中，单击**灰度发布**。
 5. 单击**开始安装**。完成灰度发布组件安装后，控制台自动跳转到灰度发布页面。



<#>

发布管理-使用手动发布模式创建灰度发布

- 在使用灰度发布的**手动发布**方式的时候，您需要创建一个Ingree、Service和Deployment结构的**应用**。
- 目前灰度发布只支持基于Deployment和Ingress的灰度发布
- 本实践举例介绍使用手动发布方式的灰度发布流程。
 - **步骤一：创建测试应用**
 - 本文使用以下YAML文件模板作为测试示例。该示例使用Deployment部署一个Web应用，并通过路由（Ingress）向外暴露服务（Service）。

```
apiVersion: apps/v1 kind: Deployment metadata: name: podinfo labels: app: podinfo spec: replicas: 4 minReadySeconds: 5 revisionHistoryLimit: 5 progressDeadlineSeconds: 60 strategy: rollingUpdate: maxUnavailable: 1 type: RollingUpdate selector: matchLabels: app: podinfo template: metadata: labels: app: podinfo spec: containers: - name: podinfo image: registry.cn-hangzhou.aliyuncs.com/acs/rollouts-demo:green imagePullPolicy: IfNotPresent ports: - name: http containerPort: 8080 protocol: TCP readinessProbe: tcpSocket: port: 8080 initialDelaySeconds: 5 timeoutSeconds: 5 --- apiVersion: v1 kind: Service metadata: name: my-podinfo-svc spec: selector: app: podinfo ports: - protocol: TCP port: 80 targetPort: 8080 --- apiVersion: extensions/v1beta1 kind: Ingress metadata: name: podinfo labels: app: podinfo spec: rules: - host: app.example.com http: paths: - backend: serviceName: my-podinfo-svc servicePort: 80
```

YAML模板



发布管理-使用手动发布模式创建灰度发布

- **步骤二：创建灰度发布**
 - 创建灰度发布时，设置上述创建的Deployment、Service、Ingress，然后设置发布方式为**手动发布**。

配置	说明
应用部署集群	应用部署集群名称
命名空间	应用的命名空间选择，默认选项即可
Deployment	选择需要发布的应用名称，本列中创建的应用实例为：podinfo
Service	选择应用的服务，本列中创建的服务为：my-podinfo-svc
Ingress	选择应用的路由，本列中创建的路由为：podinfo
发布方式	然后设置发布方式为手动发布。



发布管理-使用手动发布模式创建灰度发布

- 完成创建灰度发布后，单击创建的灰度发布名称，发布状态显示**初始化完毕**。
- 如果显示**初始化中**的话，可以多次单击**刷新**。



<#>

阿里云

发布管理-使用手动发布模式创建灰度发布

- 步骤三：执行灰度发布**
 - 初始化完毕后，就可以进行灰度发布。
 - 开始灰度发布前，您需要先更新一下应用。
- 本文示例以更新镜像tag为例，修改镜像tag中的green为blue。先发布两个Pod为新版本，同时将新版本的灰度流量设置为10%。
 - 单击目标灰度发布名称，然后在详情页面，单击**发布对象**名称后面的编辑图标。
 - 在**查看YAML**面板中，将green修改为blue，然后单击**确定**。
 - 在灰度发布**基本信息**区域，编辑**当前权重**[0-100]为10，**新版容器组副本数**[0-4]为2。



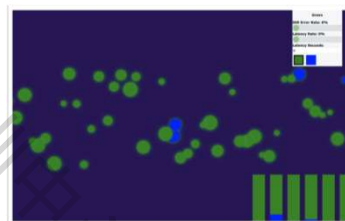
<#>

阿里云

发布管理-使用手动发布模式创建灰度发布

- 可以看到，新版本的Pod已经创建。这时候访问应用观察流量情况，可以看到流量中绿色占比90%，蓝色占比10%。

名称	状态	创建时间	版本	部署名称
podinfo-752252525-blue	运行中	2021年2月16日 19:57:48	nginx:1.19.3-alpine:demo-blue	podinfo-752252525-blue
podinfo-752252525-blue	运行中	2021年2月16日 19:57:48	nginx:1.19.3-alpine:demo-blue	podinfo-752252525-blue
podinfo-752252525-green	运行中	2021年2月16日 19:57:48	nginx:1.19.3-alpine:demo-green	podinfo-752252525-green
podinfo-752252525-green	运行中	2021年2月16日 19:57:48	nginx:1.19.3-alpine:demo-green	podinfo-752252525-green
podinfo-752252525-green	运行中	2021年2月16日 19:57:48	nginx:1.19.3-alpine:demo-green	podinfo-752252525-green
podinfo-752252525-green	运行中	2021年2月16日 19:57:48	nginx:1.19.3-alpine:demo-green	podinfo-752252525-green



- 根据业务的实际情况，不断调整新旧版容器数量，将旧版Pod扩容，新版Pod扩容，然后将更多的流量导入到新版本的Pod中。
- 当测试完毕且没有问题时，单击页面右上方的**确认发布**，完成此次发布。
- 发布成功后，可以查看流量情况。当业务的所有流量都转向了新的Pod时，就完成了一次应用的发布。您可以单击页面右上方的**完成**，结束此次发布动作。

<#>

阿里云

发布管理-使用手动发布模式创建灰度发布

- 查看历史记录**
 - 每次发布完成后，您可以查看发布的记录：在**灰度发布**页面的**历史记录**区域，您可以查看所有创建的灰度发布的发布记录。
 - 在每个创建的灰度发布详情页面下方的**发布日志**区域，查看发布的过程。

名称	发布对象	部署名称	发布状态	创建时间
podinfo-default	Deployment podinfo	default	初始化完成	2021年2月16日 19:57:48
2021年2月16日 19:58:04	创建灰度发布	状态: Initializing -> Initialized		
2021年2月16日 19:57:48	创建灰度发布	状态: -> Initializing		
2021年2月16日 19:57:48	创建灰度发布			

<#>

阿里云

发布管理-使用自动发布模式创建灰度发布

- 容器服务ACK的灰度发布功能除了手动发布方式外，还提供了自动发布方式。
- 自动发布模式可以让您全自动化地完成灰度发布流程。
 - 步骤一：创建测试应用**
 - 本文使用以下YAML文件模板作为测试示例。该示例使用Deployment部署一个Web应用，并通过路由（Ingress）向外暴露服务（Service）。

```
apiVersion: apps/v1 kind: Deployment metadata: name: podinfo labels: app: podinfo spec: replicas: 4 minReadySeconds: 5 revisionHistoryLimit: 5 progressDeadlineSeconds: 60 strategy: rollingUpdate: maxUnavailable: 1 type: RollingUpdate selector: matchLabels: app: podinfo template: metadata: labels: app: podinfo spec: containers: - name: podinfo image: registry.cn-hangzhou.aliyuncs.com/acs/rollouts-demo:green imagePullPolicy: IfNotPresent ports: - name: http containerPort: 8080 protocol: TCP readinessProbe: tcpSocket: port: 8080 initialDelaySeconds: 5 timeoutSeconds: 5 --- apiVersion: v1 kind: Service metadata: name: my-podinfo-svc spec: selector: app: podinfo ports: - protocol: TCP port: 80 targetPort: 8080 --- apiVersion: extensions/v1beta1 kind: Ingress metadata: name: podinfo labels: app: podinfo spec: rules: - host: app.example.com http: paths: - backend: serviceName: my-podinfo-svc servicePort: 80
```

YAML模板

<#>



发布管理-使用自动发布模式创建灰度发布

- 步骤二：创建灰度发布**
 - 创建灰度发布时，设置上述创建的Deployment、Service、Ingress，然后设置发布方式为**自动发布**。

自动发布的高级选项配置

配置	说明
权重调整步长	切换新的流量百分比。例如设置此参数为20，则说明新流量每次增加20%。
健康检查路径	在自动发布的过程中，填入一个URL地址。允许灰度发布控制器可以通过这个URL检查网络的连通性。此健康检查必须是一个HTTP模式的URL，且返回的HTTP Code必须是2xx才会认为是通过，否则就认为失败。
允许失败次数	需要配合健康检查路径来使用。如果健康检查失败次数超过了允许失败的次数，就会自动回滚。
超时时间	如果在自动发布过程中，新建Pod出现无法启动，或者健康检查卡住无法返回时超过设定的超时时间，就会认为发布失败。如果超过了允许失败的次数，系统就会触发自动回滚。
钉钉通知地址	生成的钉钉机器人Webhook。将Webhook填入到对应的钉钉通知地址。

<#>



发布管理-使用自动发布模式创建灰度发布

- 完成创建灰度发布后，发布流程会全自动化地进行。
- 当一个新版本的Pod创建出来后，流量切换百分之20至新版本Pod，此时旧版本的一个Pod会被删除，然后系统会再继续创建一个新的Pod删除一个旧版Pod，以此类推，直到完毕。

- **在配置自动发布模式时，需要注意以下两点：**

- 和手动发布不同，使用自动发布模式时，需要在创建灰度发布的时候编辑**发布对象**里面的YAML以设置需要更新的新版本Deployment。本文示例将发布的版本镜像**blue**修改成**green**。
- 需要设置**权重调整步长**。该值设置每次以多少步长来调整新版本的权重流量。本文以20为例，期初会以0%的流量切至新版本。随着新版本的Pod创建出来后，第一次切换20%的流量至新版本Pod，然后随着新版本的Pod数量变多，流量逐步切换到40%，60%，80%，直到100%。

<#>



发布管理-基于Helm的发布管理

- 阿里云Kubernetes服务集成了Helm包管理工具，帮助您快速构建云上应用，因为chart可以多次发布（release），这就带来一个发布版本管理的问题。
- 因此，阿里云Kubernetes服务提供了发布功能，通过Web界面的方式对通过Helm发布的应用进行管理。

- **前提条件**

- 您已经成功创建一个Kubernetes集群。
- 您已经使用应用目录或服务目录功能，安装了Helm应用，本例中是一个tf-model应用。

- **Helm是由Deis发起的一个开源工具，有助于简化部署和管理Kubernetes应用。**
- **Helm可以理解为Kubernetes的包管理工具，可以方便地发现、共享和使用Kubernetes构建的应用。**

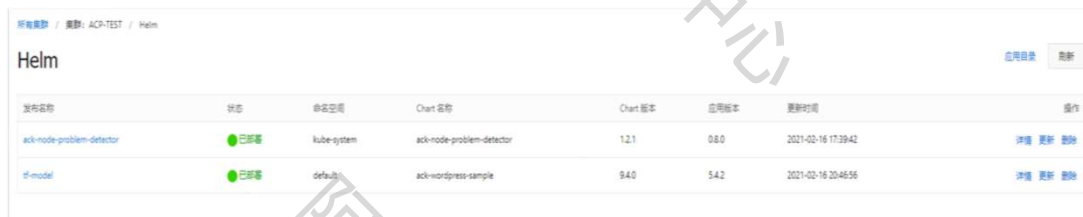
<#>



发布管理-基于Helm的发布管理

• 查看发布的详情

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，单击**应用**。
5. 单击**Helm**页签，进入发布列表页面。



发布名称	状态	命名空间	Chart 名称	Chart 版本	应用版本	更新时间	操作
ack-node-problem-detector	已部署	kube-system	ack-node-problem-detector	1.2.1	0.8.0	2021-02-16 17:39:42	详情 更新 删除
tf-model	已部署	default	ack-wordpress-sample	0.4.0	0.4.2	2021-02-16 20:46:56	详情 更新 删除

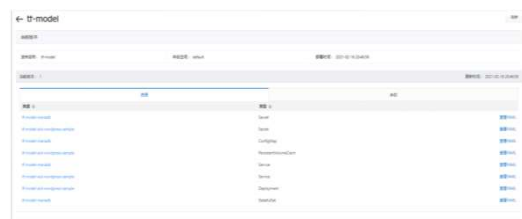
<#>



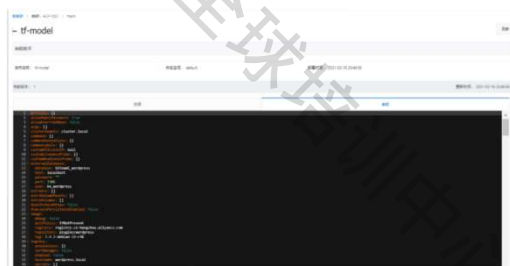
发布管理-基于Helm的发布管理

6. 以tf-model为例，您可查看发布的详情信息，单击右侧的**详情**，进入该发布的详情页面。

- 可以查看该发布的当前版本和历史版本等信息，当前版本为1，无历史版本。
- 还可查看tf-model的资源信息，如资源名称、资源类型，以及查看YAML信息。



7. 单击**参数**页签，您可查看该Helm包安装参数配置。



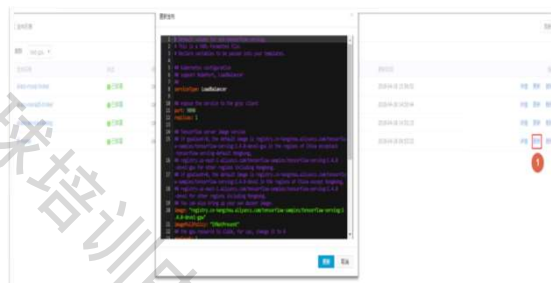
<#>



发布管理-基于Helm的发布管理

• 更新发布的版本

1. 在控制台左侧导航栏中，单击**集群**。
2. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
3. 在集群管理页左侧导航栏中，单击**应用**。
4. 单击**Helm**页签，进入发布列表页面。
5. 以tf-model为例，您可更新该发布，单击右侧的**更新**，弹出更新发布对话框
6. 在**更新发布**对话框中修改相关参数，随后单击**更新**，可对该发布进行更新。



<#>

阿里云

发布管理-基于Helm的发布管理

- 在发布列表页面，您可以看到当前版本变为2，您可以在历史版本菜单下找到版本1，单击 **回滚到该版本**，可进行回滚。



• 删除发布

1. 单击**Helm**页签，进入发布列表页面。
2. 以tf-model为例，您可删除该发布，单击右侧的**删除**。
3. 在**删除应用**对话框中，勾选是否**清除发布记录**，然后单击**确定**，您可以删除tf-model应用，其包含的service、deployment等资源都会一并删除。



<#>

阿里云

本节课小节及回顾

- 本章节主要讲述内容：
- 阿里云容器服务ACK集群管理：
 - 升级集群、管理维护核心组件、扩容集群等功能。
- 阿里云容器服务ACK安全体系：
 - 提供从基础设施到上层用户业务应用全链条安全体系支撑，支持运行时刻的安全策略管理，应用安全配置巡检和运行时刻的安全监控和告警，提升容器安全整体纵深防御能力。
- 阿里云容器服务ACK版灰度发布：
 - 可以帮助您渐进式更新部署工具。它能结合路由控制器实现多版本共存，发布暂停，流量百分比切换，健康检查等功能，极大解放灰度发布过程中的手动操作，全自动化实现线上灰度流量切换。
- 阿里云容器服务ACK版灰度发布分为手动发布模式和自动发布模式两种。

阿里云云原生容器ACP认证培训

Serverless Kubernetes (ASK) 概述



课程目标

学习完本课程后，你将能够：

1. 了解 Serverless 容器进化过程
2. 了解 ASK 的产品功能
3. 了解 ACK 与 ASK 的区别
4. 熟悉 弹性容器 ECI 的特性
5. 熟悉 ECI 与 ECS 的区别
6. 掌握 ASK 的核心功能



课程目录

1. Serverless 容器进化

1.1 Serverless 容器技术的基本概念

1.2 Serverless 容器技术的发展趋势

1.3 Serverless 容器技术的构架思考

2. 阿里云 Serverless 容器 (ASK) 产品介绍

3. 阿里云 Serverless 容器 (ASK) 适用场景

4. 阿里云 Serverless 容器 (ASK) 核心功能

5. 阿里云弹性容器实例 ECI 产品概述

<#>



Serverless容器 – 基本概念

• 特点

- 用户无需购买和管理服务器
- 直接部署容器应用
- 提高容器应用部署的敏捷度和弹性能力
- 降低用户计算成本
- 让用户聚焦业务应用

• 优势

- 敏捷部署、安全隔离、生态链接、高移植性
- 无需节点容量规划
- 无需OS和系统软件维护
- 零基础设施运维
- “无限”容量、秒级扩容、基于容器扩容
- 更高的资源利用率、更低计算成本

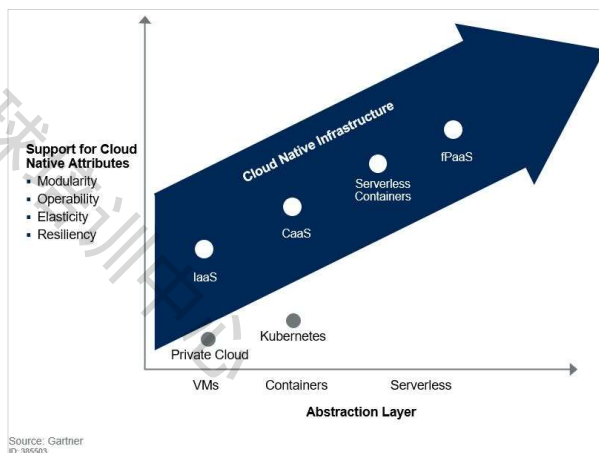


<#>



Serverless容器 – 发展趋势

- 两种技术趋势的整合
 - 云平台托管的后端服务 BaaS
 - 无状态计算模型：函数服务 FaaS
- Gartner
 - 到2023年，70% AI任务会通过容器、Serverless等计算模型构建
- AWS
 - 在2019年 40%的ECS新客户采用Serverless Container
- 采纳 Serverless 技术的行业广泛
 - 外包经济、金融业、服务业
- Serverless 是云计算必经的一场革命
- Serverless 会越来越流行



<#>

Serverless容器 – 发展趋势

- Gartner在 2020年 Public Cloud Container Service Market评估报告中把Serverless容器作为云厂商容器服务平台的主要差异化之一
 - 其中将产品能力划分为Serverless 容器实例和Serverless Kubernetes两类
 - 这与阿里云ECI/ASK的产品定位高度一致
- Gartner报告中也谈到 Serverless容器业界标准未定，云厂商有很多空间通过技术创新提供独特的增值能力，其对云厂商的建议是：
 - 扩展Serverless容器应用场景和产品组合，迁移更多普通容器workload到serverless容器服务。
 - 推进Serverless容器的标准化，减轻用户对云厂商锁定的担忧。

<#>

Serverless容器 – 构架思考

- 在阿里云Kubernetes服务中使用Serverless容器的场景，我们同时支持在K8s集群中使用ECI的方案ACK on ECI，和针对Serverless Kubernetes的极致优化的产品ASK。二者可以实现互补，覆盖满足不同客户的诉求。
- 不同于标准K8s，Serverless K8s与IaaS基础设施深度整合，其产品模式更利于公有云厂商通过技术创新，提升规模、效率和能力。
- 在架构层面我们将Serverless容器分成**容器编排**和**计算资源池**两层，下面我们将对这两层进行深度剖析，分享我们对Serverless容器架构和产品背后的关键思考。

Serverless容器 – 构架思考：容器编排

- Kubernetes在容器编排的成功不止得益于Google的光环和CNCF（云原生计算基金会）的努力运作。背后是其在Google Borg大规模分布式资源调度和自动化运维领域的沉淀和升华。
- 其中几个技术要点：
 - 声明式API
 - 可扩展性架构
 - 可移植性

Serverless容器 – 构架思考：设计原则

- Serverless Kubernetes 必须要能兼容Kubernetes生态，提供K8s的核心价值，此外要能和云的能力深度整合。
 - 用户可以直接使用Kubernetes的**声明式API**，兼容Kubernetes的应用定义，Deployment, StatefulSet, Job, Service等无需修改。
 - 全兼容Kubernetes的**扩展机制**，这个很重要，这样才能让Serverless Kubernetes支持更多的工作负载。此外Serverless K8s自身的组件也是严格遵守K8s的状态逼近的控制模式。
 - Kubernetes的能力尽可能充分利用云的能力来实现，比如资源的调度、负载均衡、服务发现等。根本性简化容器平台的设计，提升规模，降低用户运维复杂性。
 - 这些实现应该是对用户透明的，保障**可移植性**，让用户现有应用可以平滑部署在Serverless K8s之上，也应该允许用户应用混合部署在传统容器和Serverless容器之上。

<#>

阿里云

Serverless容器 – 构架思考：Nodeless

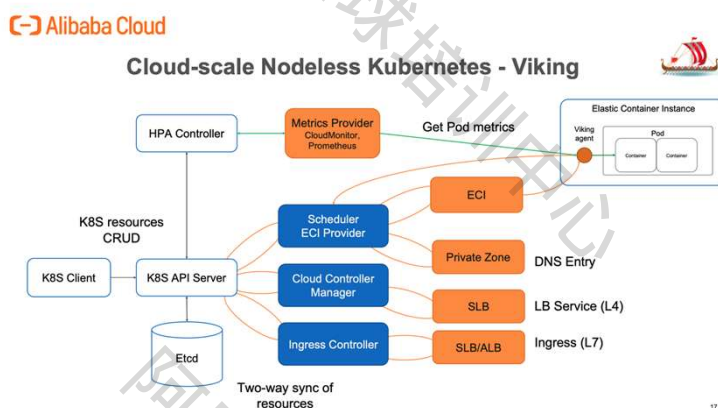
- 传统的Kubernetes采用以节点为中心的架构设计：
 - 节点是pod的运行载体，Kubernetes调度器在工作节点池中选择合适的node来运行pod，并利用Kubelet完成对Pod进行生命周期管理和自动化运维
 - 当节点池资源不够时，需要对节点池进行扩容，再对容器化应用进行扩容。
- 对于Serverless Kubernetes而言
 - 最重要的一个概念是将容器的运行时和具体的节点运行环境解耦。
 - 用户无需关注node运维和安全，降低运维成本
 - 极大简化了容器弹性实现，无需按照容量规划，按需创建容器应用Pod即可
 - 此外，Serverless容器运行时可以被整个云弹性计算基础设施所支撑，保障整体弹性的成本和规模。

<#>

阿里云

Serverless容器 – 构架思考：Viking

- 在现有Kubernetes的设计实现上，进行了扩展和优化。构建了Cloud Scale的Nodeless K8s架构 - 内部的产品代号为Viking，因为古代维京战船以迅捷和便于操作而著称。



<#>

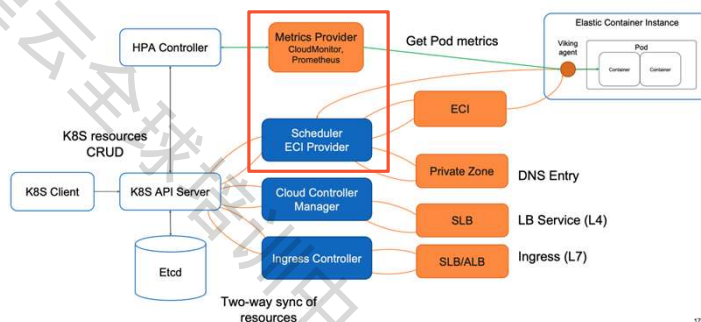
阿里云

Serverless容器 – 构架思考：优化方向

- Scheduler
 - Metrics Provider
 - Scheduler ECI Provider

Alibaba Cloud

Cloud-scale Nodeless Kubernetes - Viking



17

<#>

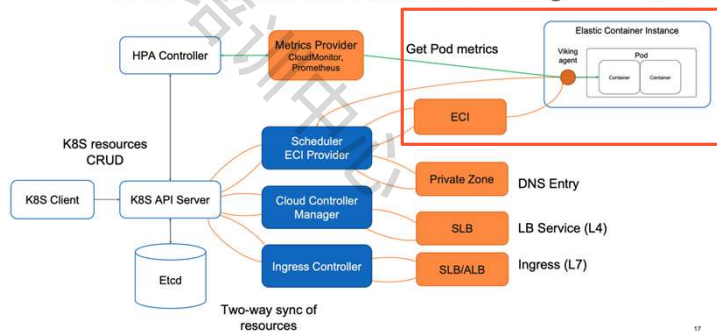
阿里云

Serverless容器 – 构架思考：优化方向

- 可伸缩性
 - ECI

Alibaba Cloud

Cloud-scale Nodeless Kubernetes - Viking



17

<#>

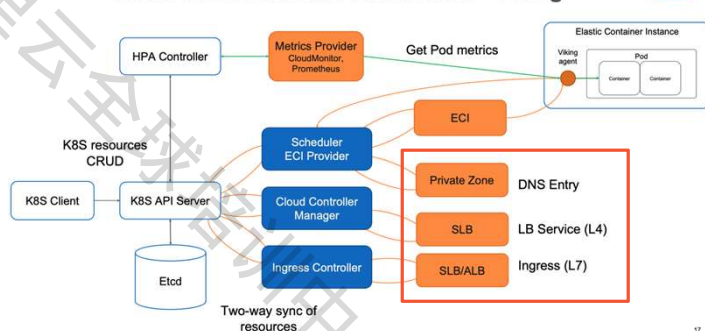
阿里云

Serverless容器 – 构架思考：优化方向

- 基于云的控制器实现
 - Private Zone
 - SLB
 - SLB/ALB

Alibaba Cloud

Cloud-scale Nodeless Kubernetes - Viking



17

<#>

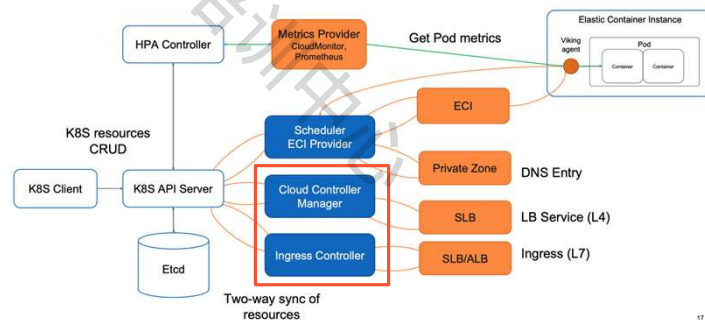
阿里云

Serverless容器 – 构架思考：优化方向

- 面向负载的深度优化
 - Knative
 - 垂直优化

Alibaba Cloud

Cloud-scale Nodeless Kubernetes - Viking



17

<#>

阿里云

课程目录

1. Serverless容器进化
2. 阿里云 Serverless 容器 (ASK) 产品介绍
 - 2.1 ASK 产品简介
 - 2.2 ASK 核心优势
 - 2.3 对Kubernetes做减法
 - 2.4 容器生态支持
3. 阿里云 Serverless 容器 (ASK) 适用场景
4. 阿里云 Serverless 容器 (ASK) 核心功能
5. 阿里云弹性容器实例 ECI 产品概述

<#>

阿里云

ASK 产品简介

• ASK集群

- 是阿里云推出的无服务器Kubernetes容器服务
 - 无需购买节点即可直接部署容器应用，无需对集群进行节点维护和容量规划，并且根据应用配置的CPU和内存资源量进行按需付费
 - 提供完善的Kubernetes兼容能力，同时降低了Kubernetes使用门槛，让您更专注于应用程序，而不是管理底层基础设施。
-
- ASK集群中的Pod基于阿里云弹性容器实例ECI运行在安全隔离的容器运行环境中。
 - 每个Pod容器实例底层通过轻量级虚拟化安全沙箱技术完全强隔离，容器实例间互不影响。

<#>



ASK 核心优势

简单易用

秒级伸缩

安全隔离

降低成本

原生兼容

服务集成

<#>



对Kubernetes做减法

- Kubernetes 作为通用的容器编排系统，承载了广泛的应用和场景，包括 CI / CD，数据计算，在线应用，AI 等，然而由于其通用性和复杂性，管理一个 Kubernetes 集群对于很多用户而言还是充满挑战的，主要体现在：
 - 学习成本高；
 - 集群运维管理成本高，包括节点管理、容量规划，以及各种节点异常问题的定位；
 - 计算成本在很多场景中没有达到最优，比如对于一个定时运行 Jobs 的集群，长期持有资源池对于用户来说是浪费的行为，资源利用率不高。
- 因此，对 Kubernetes 做减法成为了必然。

<#>

阿里云

无节点管理

- 我们相信未来用户会更加关注应用的开发，而不是基础设施的维护。体现在 Kubernetes 集群中，我们希望用户能够关注在 pod/service/ingress/job 等应用编排语义上，对底层 node 则可以减少关注。
- 无需管理节点也可以显著降低集群的运维管理成本，经统计 Kubernetes 常见的异常问题中大多数与节点相关，比如 Node NotReady 问题，也无需担忧 Node 的安全问题，以及基础系统软件的升级和维护。
- 在 ASK 集群中，我们使用虚拟节点 virtual-kubelet 代替 ecs 节点，虚拟节点的容量可以认为是“无限大”，用户不需要为集群的容量担忧，无需预先做容量规划。

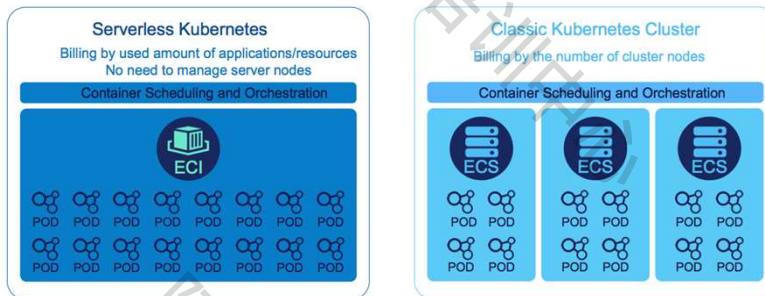
<#>

阿里云

Serverless Kubernetes vs. Classic Kubernetes

- Serverless Kubernetes 将 Classic Kubernetes中的ECS转化为更加灵活、现代的ECI。

Serverless Kubernetes vs. Classic Kubernetes



<#>



无Master管理与极简的K8s

- 和 ACK 托管版一样，ASK 的 Master(apiserver, ccm, kcm 等)资源被容器服务平台托管，用户无需管理这些核心组件的升级和运维，也不用付出成本。
- 除了无需管理节点和 Master 外，ASK还对 Kubernetes 集群管理做了大量简化，包括默认托管很多 addon，用户无需再管理一些基础的 addon，也不需要为这些 addon 付费。
- 依赖阿里云原生的网络和存储等能力，以及独特的托管架构设计，ASK提供了极度简化但功能完备的 Kubernetes 基础运行环境。

<#>



简化弹性伸缩

- 因为无需管理节点和容量规划，当集群需要扩容时也就不需要考虑节点层面的扩容，只需要关注 pod 的扩容，这对于扩容的速度和效率都是极大的提升。
- ASK/ECI 的方式被客户用来快速应对业务流量高峰。
- 当前 ASK/ECI 支持 30s 完全启动 500 个 pod（至 Running 状态），单个 pod 启动可以达到 10s 以内。

更低的成本

- 除去 ASK 集群本身的低成本创建外，pod 的按需使用也让很多场景下资源利用率达到最优。对于很多 Jobs 或者数据计算场景而言，用户并不需要长期维护一个固定的资源池，这时 ASK/ECI 可以很好的支持这些诉求。
- 经验证，当 pod 一天中运行时间少于 16 个小时，则 ASK/ECI 的方式相比保有 ecs 资源池更节省经济成本。

ECI：快速交付容器资源的弹性计算服务

- ECI 是阿里云基于 ECS IaaS 资源池提供的稳定、高效、高弹性容器实例服务。
- ECI 让容器成为了公有云的第一等公民，用户无需购买和管理 ECS 就可以直接部署容器应用，这种简化的容器实例产品形态和 ASK 形成了一个完美的组合。
- 用户可以直接使用 ECI Open API 创建容器实例资源，但在容器场景中用户普遍需要一个编排系统，来负责容器的调度、高可用编排等能力，而 ASK 正是这样的 Kubernetes 编排层。
- 对于 ASK 而言
 - ECI 让 ASK 容器服务免去了搭建后台计算资源池的必要，更不用为底层计算资源池的容量而担忧。
 - 基于 ECI 就意味着基于整个阿里云 IaaS 规模化资源池，天然拥有了库存和弹性优势（比如可以通过 Annotation 的方式指定底层 ECI 对应的 ECS 规格，大部分 ECS 规格都可以在 ASK 中使用，满足多种计算场景的需求）。
 - 另外 ECI 和 ECS 复用资源池意味着我们可以最大化释放规模化红利，给用户提供更低成本的服务。

<#>



容器生态支持

- ASK 对 Kubernetes 容器生态提供了完善的支持，目前已有大量客户使用 ASK 来支撑如下各种场景：
 - **CI/CD**: gitlab-runner, jenkins/jenkins-x
 - **数据计算**: spark/spark-operator, flink, presto, argo
 - **AI**: tensorflow/arena
 - **ServiceMesh**: istio, knative
 - **测试**: locust, selenium
- ASK 集群不支持 Helm v2，近期 ACK/ASK 会发布 Helm v3 的支持，之后用户可以非常方便的在 ASK 集群中部署 Charts。

<#>



课程目录

1. Serverless 容器进化
2. 阿里云 Serverless 容器 (ASK) 产品介绍
3. 阿里云 Serverless 容器 (ASK) 适用场景
 - 3.1 ASK 与 ACK 对比
 - 3.2 适用场景
4. 阿里云 Serverless 容器 (ASK) 核心功能
5. 阿里云弹性容器实例 ECI 产品概述

<#>



ASK与ACK 对比

功能	ACK	ASK
存储	需要部署 aliyun-disk-controller/flexvolume	无需部署 (正在支持中)
CNI 网络	需要部署 terway/flannel daemonset	无需部署, 基于 vpc 网络通信
coredns 服务发现	需要部署 2 个 coredns 副本	无需部署, 基于 privatezone 访问
kube-proxy	需要部署 kube-proxy daemonset	无需部署, 基于 privatezone 访问
Ingress	需要部署 nginx-ingress-controller	无需部署, 基于 SLB 七层转发
免密拉取 ACR 镜像	需要部署 acr-credential-helper	无需部署, 默认支持
sls 日志收集	需要部署 logtail daemonset	无需部署, 默认支持
metrics 统计	需要部署 metrics-server	无需部署, 开箱即用
挂载 eip	需要部署 terway	无需部署, 使用 annotation 指定
云盘随 pod 创建挂载	依赖 aliyun-disk-controller	无需部署, 默认支持
弹性伸缩	需要部署 cluster-autoscaler	无需部署
GPU 插件	需要部署 Nvidia-docker	无需部署, 开箱即用

<#>



ASK 适用场景

应用托管

在线业务弹性

数据计算

CI/CD持续集成

定时任务

测试环境

<#>

阿里云

课程目录

1. Serverless 容器进化
2. 阿里云 Serverless 容器 (ASK) 产品介绍
3. 阿里云 Serverless 容器 (ASK) 适用场景
4. 阿里云 Serverless 容器 (ASK) 核心功能
 - 4.1 虚拟节点
 - 4.2 Pod安全隔离与配置
 - 4.3 应用负载管理与弹性伸缩
 - 4.4 网络、存储与日志管理
5. 阿里云弹性容器实例 ECI 产品概述

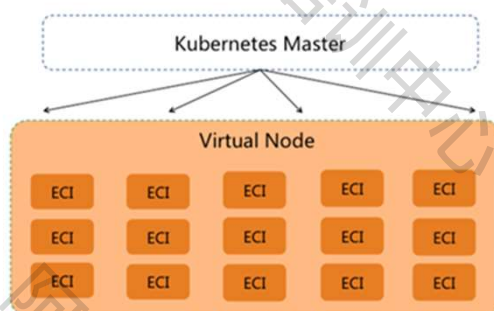
<#>

阿里云

虚拟节点

- ASK集群中基于虚拟节点创建Pod。
 - 虚拟节点实现了Kubernetes与弹性容器实例ECI的无缝连接，让Kubernetes集群获得极大的弹性能力，而不必关心底层计算资源容量。

ASK (Serverless Kubernetes)



<#>

Pod安全隔离

ASK集群中的Pod基于阿里云弹性容器实例ECI运行在安全隔离的容器运行环境中。

每个容器实例底层通过轻量级虚拟化安全沙箱技术完全强隔离，容器实例间互不影响。

实例在调度时尽可能分布在不同的物理机上，进一步保障了高可用性。

ECI的容器实例底层运行在Alibaba Cloud Linux2操作系统之上，ASK集群作为Serverless容器服务，您无法访问ECI底层OS运行环境。

<#>

Pod配置

- Pod基于弹性容器实例ECI创建
 - 支持原生的Kubernetes Pod功能，包括启动多个容器、设置环境变量、设置RestartPolicy、设置健康检查命令和挂载volumes、preStop等。
- 同时支持执行命令kubect logs访问容器日志和执行kubect exec进入容器。
- ECI支持多种规格配置的方式来申请资源和计费。
 - 指定CPU和Memory
 - 指定容器的CPU和Memory
 - 指定Pod的CPU和Memory
 - 指定Pod的ECS规格
 - 抢占式实例

<#>



应用负载管理

- 支持Deployment、StatefulSet、Job/CronJob、Pod、CRD等原生Kubernetes负载类型。
- 不支持DAEMONSET：Serverless集群中不支持节点相关的功能。

<#>



弹性伸缩

- ASK集群中没有真实节点
 - 无需考虑节点的容量规划
 - 无需考虑基于cluster-autoscaler的节点扩容
 - 只需要关注应用的按需扩容。
- 建议您配置HPA或者CronHPA策略进行Pod的灵活按需扩容。

<#>

网络管理

- 群中的ECI Pod默认使用Host网络模式，占用交换机VSwitch的一个弹性网卡ENI资源，与VPC内的ECS、RDS互联互通。
 - Service
 - Ingress
 - 服务发现
 - 弹性公网IP

<#>

存储管理

- Pod支持挂载阿里云块存储和文件存储。
 - 阿里云块存储 (Disk)
 - 阿里云文件存储 (NAS)

日志管理

- 在ASK集群中无需部署logtail daemonset即可收集Pod的stdout和文件输出日志。
- 在Serverless Kubernetes集群中将业务容器的标准输出和日志文件收集到阿里云日志服务。
- ECI模式下不支持DaemonSet，通过将Job类任务挂载NAS盘，把输出的日志存储在NAS盘，再通过另一个同样挂载NAS盘的Pod来采集Job任务标准输出到日志系统中。
- 日志服务支持在ECI中通过Sidecar模式采集日志。

课程目录

1. Serverless 容器进化
2. 阿里云 Serverless 容器 (ASK) 产品介绍
3. 阿里云 Serverless 容器 (ASK) 适用场景
4. 阿里云 Serverless 容器 (ASK) 核心功能

5. 阿里云弹性容器实例 ECI 产品概述

5.1 弹性容器实例 ECI 概述

5.2 ECI 与 ECS 的关系

5.3 ECI 的关键技术

5.4 ECI 的优势

<#>



ECI 概述

- 弹性容器实例 ECI (Elastic Container Instance) 是阿里云在云原生时代为用户提供的基础计算服务，是阿里云云原生时代下的云计算基础设施。
- ECI 改变了以往计算服务以整台机器作为交付形态的传统，通过结合容器技术与无服务器 (Serverless) 技术为用户提供了一款安全便捷的 Serverless 运行服务。
- 使用 ECI 的过程中用户不再需要关注繁冗的底层基础设施维护工作，仅需提供打包好的容器镜像即可运行用户服务。
- 在颠覆传统资源交付形态的同时，用户还可以享受按实际消耗资源付费等便利，降低用户的实际使用成本。

<#>



ECI概述

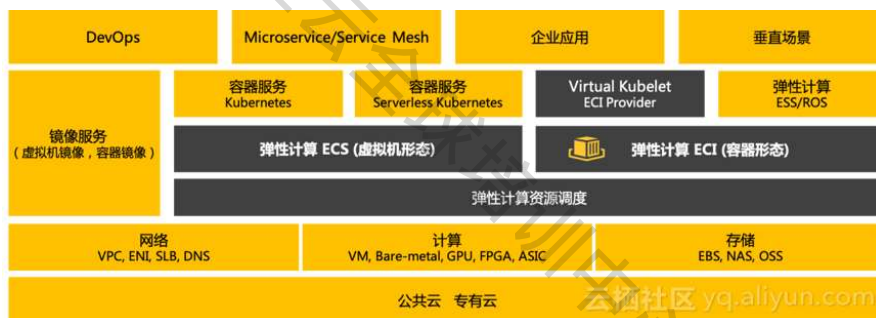
- ECI为Kubernetes提供了基础的容器Pod运行环境，但业务间的依赖、负载均衡、弹性伸缩、定期调度等能力依然需要Kubernetes来提供。
- **阿里云弹性容器实例ECI (Elastic Container Instance)**
 - 是面向容器的无服务器弹性计算服务
 - 提供免运维、强隔离、快速启动的容器运行环境
 - 使用ECI无需购买和管理底层ECS服务器，让您更加关注在容器应用而非底层基础设施的维护工作。

<#>



ECI/ECS关系

- 阿里云弹性计算团队推出的云原生基础设施，ECI 定位为 ECS 同级别的计算资源类产品。
 - 依托弹性计算团队多年研发 ECS 产品积累的技术和经验，ECI 在资源弹性、安全、运维监控体系等方面继承了大量弹性计算团队在此前产品上的积累
 - 针对云原生时代的需求进行了有针对性的改进和优化。



<#>



ECI/ECS对比

- **更低的计算成本**：弹性成本要低于ECS，long run应用成本要接近ECS包年包月
- **更高的弹性效率**：ECI扩容速度要远高于ECS
- **更大的弹性规模**：与传统ECS节点扩容不同，一个大规模容器应用动辄需要数万核的弹性算力。
- **持平的计算性能**：ECI计算效能需要和同规格ECS有一致的性能表现
- **更低的迁移成本**：与现有容器应用生态完美集成
- **更低的使用成本**：全自动化安全和运维能力

ECI关键技术

- 基于轻量化Micro VM的安全容器运行时
- 基于 Pod 的基本调度单位和标准、开放的API接口
- ECI与ECS并池架构

ECI的优势

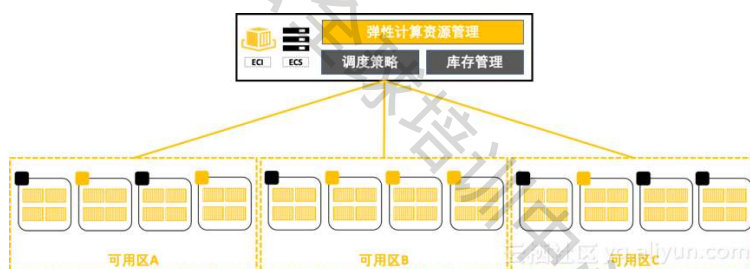
- ECI 服务依托阿里云弹性计算团队，继承了弹性计算团队在库存管理，资源调度，虚拟化，安全，操作系统及内核，存储及虚拟化网络方面的技术积累，为用户打造了安全、弹性的 Serverless 运行服务。



<#>

弹性

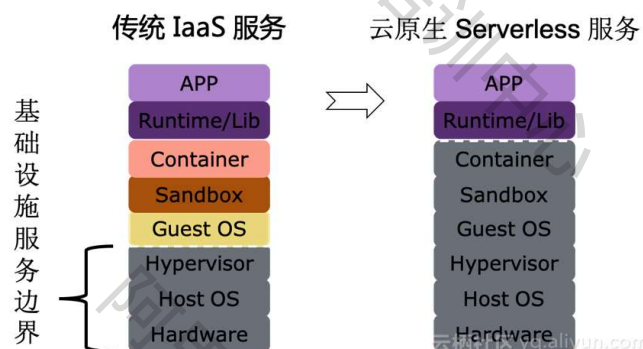
- 弹性是云计算带给用户的最大价值之一，通过将业务部署到云上，用户基础设施的扩缩容等方面能够更加灵活。
- 当前 ECI 服务完全构建在弹性计算整体资源调度平台之上，与 ECS 服务使用相同的库存管理与调度策略。依托弹性计算强大的库存管理系统，ECI 服务能够切实保证充足的库存，满足用户对极致弹性的需求



<#>

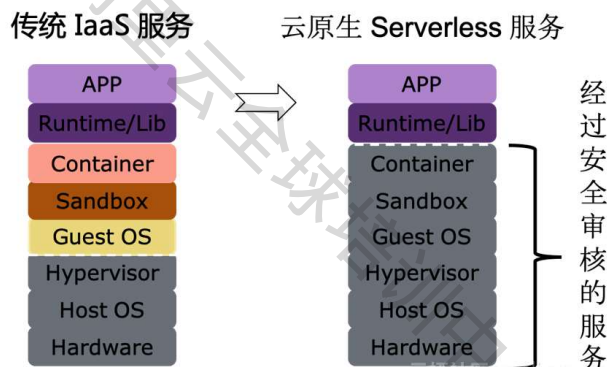
Serverless

- 无服务器 (Serverless) 技术的核心是将用户从繁冗的基础设施运维问题中拯救出来，让用户能够将精力专注于自身业务和服务上。
- ECI 通过结合容器技术和无服务器技术，将云厂商的运维边界从基础设施的整机层面提升到了容器运行时层面。



安全

- 安全问题可以细化为两个层面的问题
 - 云平台自身的安全
 - 用户自身的安全问题



安全

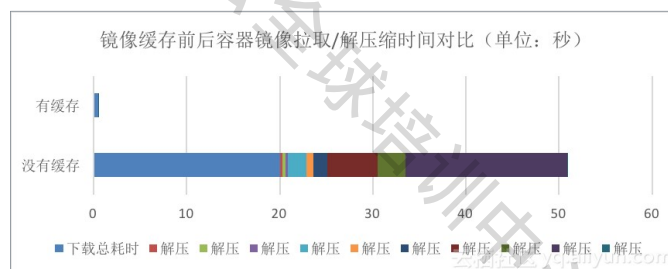
- ECI选择基于袋鼠云原生容器引擎和轻量化Micro VM来实现安全、隔离的容器运行时。除了运行时的资源隔离之外，不同客户之间的网络、存储、quota、弹性SLO等一系列能力也基于阿里云基础设施，实现了严格的多租隔离。
- 在性能方面，除了袋鼠容器引擎在OS/容器方面的高度优化之外，ECI在容器执行上优化集成了现有阿里云基础设施能力

<#>



易用性

- ECI 服务在管控、虚拟化、镜像拉取等多个层面对交付时间进行了优化。
- 借助与 ECS 服务共生的关系，通过优化 ECI 的管控系统能够更快的获取各类底层基础资源的情况，着重对创建链路进行优化，加快管控系统中 ECI 相关创建事件的处理速度。
- 在容器运行环境的创建中也采用了特殊的虚拟化技术，快速批量创建实例。
- 我们也对容器镜像的拉取过程进行了一定的优化。



<#>



课程小结

- 本节课的内容主要包括:

1. Serverless 和容器技术演进路线
2. 阿里云 Serverless 容器 ASK 产品介绍
3. 阿里云 ACK/ASK 产品区别与适用场景
4. 阿里云 ASK 产品的核心功能介绍
5. 阿里云弹性容器实例 ECI 功能概述
6. 阿里云 ECI/ECS 产品的功能对比

阿里云云原生容器ACP认证培训

Serverless Kubernetes (ASK) 使用与实践



课程目标

学习完本课程后，你将能够：

1. 了解ASK的结构概况
2. 了解ASK的各模块的管理方法
3. 使用ASK进行实例的操作



课程目录

1. ASK集群管理

1.1 创建ASK集群

1.2 查看ASK集群

1.3 删除ASK集群

1.4 管理和访问ASK集群

2. ASK应用管理

3. ASK网络、存储、日志、监控管理

4. ASK弹性伸缩管理

5. ECI Pod管理

<#>



ASK集群管理 - 创建

• 通过容器服务控制台创建

- 在控制台左侧导航栏中，单击**集群**。
- 在**集群列表**页面中，单击页面右上角的**创建集群**。



<#>



ASK集群管理 - 创建

• 通过容器服务控制台创建

- 单击ASK集群页签，然后完成集群配置。
- 在页面右侧，单击创建集群，在弹出的当前配置确认页面，单击确定，启动部署。

创建集群

ACK 托管版 ACK 专有版 **ASK 集群** ACK 边缘托管版 注册集群

Serverless Kubernetes 所调度的 ECI 容器实例为按量收费资源，请您在创建应用前务必了解 ECI 的计费标准，详情请参见 ECI 计费标准

容器 ECI 免费试用3个月，限量100元代金券，[立即领取](#)

* 集群名称 ASK-Demo
名称为1~63个字符，可包含数字、汉字、英文字母或下划线 (-)

资源组 未选择 您可以在控制台顶栏中选择资源组

地域 华北2 (北京) 华北3 (张家口) 华北5 (呼和浩特) 华北6 (乌兰察布) 华东1 (杭州)
华东2 (上海) 华南1 (深圳) 华南2 (河源) 华南3 (广州) 西南1 (成都)
中国 (香港) 日本 (东京) 新加坡 澳大利亚 (悉尼) 马来西亚 (吉隆坡)
印度尼西亚 (雅加达) 印度 (孟买) 美国 (弗吉尼亚) 美国 (硅谷) 美国 (伦敦)
德国 (法兰克福)

时区 Asia/Shanghai (UTC+08:00)

当前配置

地域: cn-beijing

创建集群

开通按量付费的云资源，如账户余额少于100.00元，需充值后方可开通，否则创建集群会失败。[去充值](#)

开通云资源，负载均衡 需要先进行实名认证，[马上认证>>](#)

您目前可以创建 20 个集群。如需更高配额，请提交工单申请

如需更多实例规格，请提交工单申请

ASK集群管理 - 创建

- 集群创建成功后，您可以在容器服务控制台的集群列表页面查看所创建的集群。

容器服务 - Kubernetes

集群列表

查看当前集群与节点配额 刷新 集群模板 创建集群

双12惊喜折扣来袭，容器镜像服务企业版 ACR EE 7.5 折起

名称	标签	集群类型 (全部)	地域 (全部)	集群状态	节点数	使用量	创建时间	版本	操作
ASK-Demo		Serverless Kubernetes	华北2 (北京)	运行中			2021-02-03 14:33:33	v1.18.8-aliyun.1	详情 应用管理 查看日志 集群升级 更多

ASK集群管理 - 创建

- 通过阿里云CLI创建
 - 登录阿里云Cloud Shell
 - 创建工作目录和配置文件
 - 执行命令创建ASK集群

```
shell@Alicloud:~/ask-cli-demo$ rm -rf tutorial-aliyun-cli/
shell@Alicloud:~/ask-cli-demo$ cat create.json
{
  "cluster_type": "Ask",
  "name": "ask-cli-demo",
  "kubernetes_version": "1.18.8-aliyun.1",
  "region_id": "cn-beijing",
  "zoneid": "cn-beijing-h",
  "endpoint_public_access": false,
  "nat_gateway": true,
  "private_zone": false
}
shell@Alicloud:~/ask-cli-demo$ aliyun cs POST /clusters --header "Content-Type=application/json" --body "$(cat create.json)"
{
  "cluster_id": "T-1234567890",
  "instanceId": "I-1234567890",
  "request_id": "R-1234567890",
  "task_id": "T-1234567890"
}
```

```
aliyun cs POST /clusters --header "Content-Type=application/json" --body "$(cat create.json)"
```

<#>



ASK集群管理 - 创建

- 创建成功后，执行以下命令查看集群实例。

```
aliyun cs GET /clusters/<YOUR-CLUSTER-ID>
```

```
{
  "_": "PayByTraffic",
  "capabilities": null,
  "cluster_healthy": "",
  "cluster_id": "T-1234567890",
  "cluster_spec": "",
  "cluster_type": "Ask",
  "created": "2021-02-04T12:36:22+08:00",
  "current_version": "v1.18.8-aliyun.1",
  "data_disk_category": "cloud",
  "data_disk_size": 0,
  "deletion_protection": false,
  "docker_version": "",
  "enabled_migration": false,
  "external_loadbalancer_id": "L-1234567890",
  "gw_bridge": "",
  "init_version": "v1.18.8-aliyun.1",
  "instance_type": "",
  "maintenance_info": {
    "cluster_upgrade_info": {
      "nodepools": null
    }
  },
}
```

<#>



ASK集群管理 – 查看

- 通过容器服务控制台查看
- 在集群页面的控制台左侧导航栏中，单击的**集群信息**



ASK集群管理 – 查看

- 在集群的详情信息页面，可通过切换页签查看以下信息。
 - **基本信息**：查看集群ID、地域、API Server连接端点以及其他网络信息。
 - **连接信息**：获取公网和内网环境下KubeConfig文件的配置内容。
 - **集群资源**：查看集群所使用的相关云资源。通过单击相应的资源ID可以跳转至对应的控制台。
 - **集群日志**：查看集群日志。



ASK集群管理 – 删除

• 通过容器服务控制台删除

- 在控制台左侧导航栏中，单击**集群**。
- 在**集群列表**页面，选择所需的集群并选择**更多 > 删除**。



ASK集群管理 – 删除

• 通过容器服务控制台删除

- 在弹出的**删除集群**对话框中，阅读并选中**我已知晓以上信息并确认删除集群**，单击**确定**。



ASK集群管理 – 管理和访问

- 通过kubectll连接Kubernetes集群
- 可以通过登录容器服务控制台，在集群信息页面查看集群凭据。
 - 在控制台左侧导航栏中，单击**集群**。
 - 在**集群列表**页面中，单击目标集群名称或者目标集群右侧的**详情**。



ASK集群管理 – 管理和访问

- 通过kubectll连接Kubernetes集群
 - 在**基本信息**页签，可以在**集群信息**处查看集群的连接地址。



ASK集群管理 – 管理和访问

- 通过kubectl连接Kubernetes集群
 - 在连接信息页签，可以在配置集群凭据查看集群凭据。



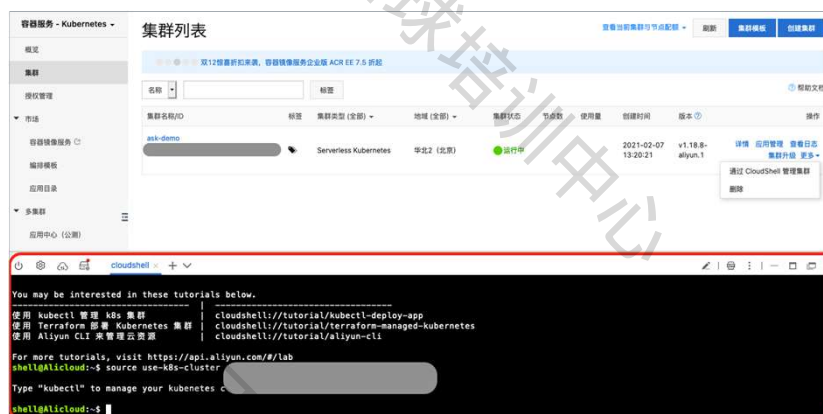
ASK集群管理 – 管理和访问

- 在CloudShell上通过kubectl管理Kubernetes集群
 - 登录容器服务控制台。
 - 在控制台左侧导航栏中，单击集群。
 - 在集群列表页面中，单击目标集群右侧操作列下的更多 > 通过CloudShell管理集群。



ASK集群管理 – 管理和访问

- 在CloudShell上通过kubectl管理Kubernetes集群
 - 在CloudShell中可通过kubectl工具管理容器服务Kubernetes集群。



<#>



ASK集群管理 – 管理和访问

- 通过公网访问集群API Server
 - 使用弹性公网IP EIP (Elastic IP Address) 暴露ASK集群的API Server后, 将获得从公网访问集群API Server的能力。
 - 可在创建集群的时候, 选中**使用EIP暴露API Server**选项以启用该功能。

公网访问

☒ 使用 EIP 暴露 API Server

默认将为 API Server 创建一个内网 SLB 实例, 若删除该实例会导致 API Server 无法访问
开启后, 将获得从公网访问集群 API Server 的能力

<#>



本章小结

- **ASK集群管理：**

1. 了解ASK集群管理的主要功能。
2. 了解ASK集群创建、查看、删除的方式。
3. 了解ASK集群的管理和访问的接口。

课程目录

1. ASK集群管理
2. **ASK应用管理**
 - 2.1 通过命令管理应用
 - 2.2 使用镜像创建应用
 - 2.3 服务管理
 - 2.4 容器查看
3. ASK网络、存储、日志、监控管理
4. ASK弹性伸缩管理
5. ECI Pod管理

ASK应用管理 - 通过命令管理应用

- 通过命令创建应用

- 在集群的CloudShell中运行简单的容器。

```
shell@Alicloud:~$ kubectl run nginx --image=registry.cn-hangzhou.aliyuncs.com/spacexnice/netdia:latest  
pod/nginx created
```

- 为该容器创建一个服务入口。

```
shell@Alicloud:~$ kubectl expose pod nginx --port=8080 --target-port=80 --type=LoadBalancer  
service/nginx exposed
```

- 通过命令查看容器

```
shell@Alicloud:~$ kubectl get pods  
NAME      READY   STATUS    RESTARTS   AGE  
nginx     1/1     Running   0           2m33s
```

<#>



ASK应用管理 - 使用镜像创建应用

- 配置应用基本信息

- 登录容器服务控制台。
- 在控制台左侧导航栏中，单击**集群**。
- 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。



<#>



ASK应用管理 - 使用镜像创建应用

• 配置应用基本信息

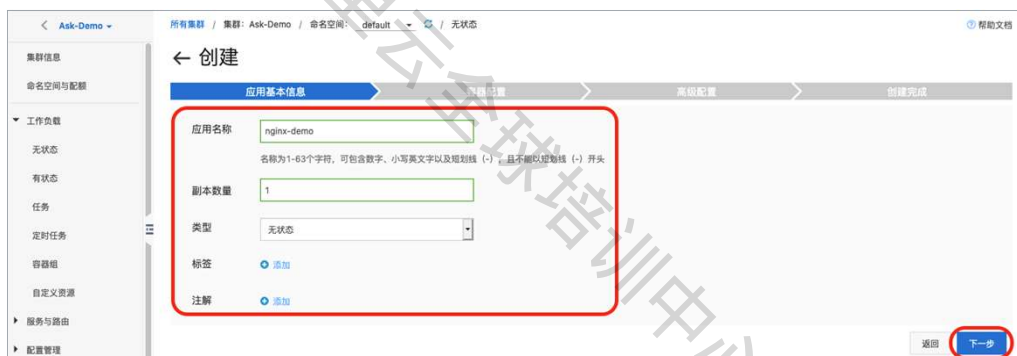
- 在集群管理页左侧导航栏中，单击工作负载。
- 在无状态页签中，单击使用镜像创建。



ASK应用管理 - 使用镜像创建应用

• 配置应用基本信息

- 在应用基本信息页面，设置应用的基本信息。
- 单击下一步。进入容器配置页面。



ASK应用管理 - 使用镜像创建应用

- 配置容器

- 在基本配置区域，完成容器的基本配置。

基础配置

镜像名称: [选择镜像](#)

镜像Tag: [选择镜像Tag](#)

镜像拉取策略: [设置镜像密钥](#)

所需资源: CPU 内存

容器启动项: ☐ stdin ☐ tty

Init Container ☐

<#>

阿里云

ASK应用管理 - 使用镜像创建应用

- 配置容器

- 在端口设置区域，单击新增设置容器的端口。
- 在环境变量区域，单击新增设置环境变量。
- 在健康检查区域，根据需要开启存活检查和就绪检查。

端口设置

新增

名称	容器端口	协议
nginx	80	TCP

环境变量: 新增

健康检查

存活检查 ☐ 开启

就绪检查 ☐ 开启

<#>

阿里云

ASK应用管理 - 使用镜像创建应用

• 配置容器

- 在生命周期区域，设置容器的生命周期。

如何设置生命周期

启动执行:	命令	示例: sleep 3600 或 ["sleep", "3600"]
	参数	示例: ["--log_dir=test", "--batch_size=150"]
启动后处理:	命令	示例: echo hello world 或 ["/bin/sh", "-c", "echo hello world"]
停止前处理:	命令	示例: echo hello world 或 ["/bin/sh", "-c", "echo hello world"]

<#>



ASK应用管理 - 使用镜像创建应用

• 配置容器

- 在数据卷区域，增加本地存储或云存储声明PVC (Persistent Volume Claim) 等。

数据卷:

+ 增加本地存储				
存储卷类型	挂载源	容器路径		
+ 增加云存储声明 (PersistentVolumeClaim)				
存储卷类型	挂载源	容器路径		
+ 增加 NAS				
名称	NAS 地址	主机路径	容器路径	
+ 增加云盘				
名称	文件系统	云盘实例 ID	云盘大小	容器路径

云盘实例 ID 和大小二选一即可，仅支持添加一个云盘。当没有选择云盘实例 ID 时，将自动创建一块云盘（云盘随 pod 释放）

<#>



ASK应用管理 - 使用镜像创建应用

配置容器

- 单击下一步，进行高级设置。

The screenshot shows the 'Create' (创建) page in the ASK console, specifically the 'Container Configuration' (容器配置) step. The left sidebar contains navigation links: 集群信息, 命名空间与配额, 工作负载, 无状态, 有状态, 任务, 定时任务, 容器组, 自定义资源, 服务与路由, 配置管理, 存储, and 应用. The main content area has a progress bar with four steps: 应用基本信息, 容器配置 (current), 高级配置, and 创建完成. Under '容器配置', there is a '容器1' section with a '+ 添加容器' button. The configuration fields include: 镜像名称 (Image Name) set to 'pc.cn-hangzhou.aliyuncs.com/acs-sample/nginx', 镜像Tag (Image Tag), 镜像拉取策略 (Image Pull Policy), 设置镜像密钥 (Set Image Key), 所需资源 (Resources) with CPU '1Core' and 内存 (Memory) '2GB', 容器启动项 (Container Start Command) with checkboxes for 'stdin' and 'tty', and an 'Init Container' section. On the right, a '配置概述' (Configuration Overview) panel lists: 基本配置, 端口设置, 环境变量, 健康检查, 生命周期, and 数据卷. At the bottom right, there are '上一步' (Previous Step) and '下一步' (Next Step) buttons, with '下一步' highlighted by a red circle.

<#>

阿里云

ASK应用管理 - 使用镜像创建应用

完成高级配置

- 在访问设置区域，设置暴露后端Pod的方式。
 - 您可以设置暴露后端Pod的方式。本例中选择虚拟集群IP和路由（Ingress），构建一个公网可访问的Nginx应用。

The screenshot shows the '创建服务' (Create Service) dialog box in the ASK console. It has a title bar with a close button. The dialog is divided into two main sections. The top section, '服务 (Service)', has a '+ 添加' (Add) button and a '路由 (Ingress)' button. The bottom section, '创建服务' (Create Service), contains the following fields: 名称 (Name) set to 'nginx-demo-svc', 命名空间 (Namespace) set to 'default', 类型 (Type) set to '负载均衡' (Load Balancing), 公网访问 (Public Access) set to '是' (Yes), 新建sld (New sld) set to 'sld.x1.small', 外部流量策略 (External Traffic Policy) set to 'Local', and a table for '端口映射' (Port Mapping) with columns for 名称 (Name), 服务端口 (Service Port), 容器端口 (Container Port), and 协议 (Protocol). The table has one row: 名称 'nginx', 服务端口 '8080', 容器端口 '80', and 协议 'TCP'. At the bottom, there are '注册' (Register) and '标签' (Tag) buttons, both with '+ 添加' (Add) options. At the bottom right, there are '创建' (Create) and '取消' (Cancel) buttons, with '创建' highlighted by a red circle.

<#>

阿里云

ASK应用管理 - 使用镜像创建应用

• 完成高级配置

- 在伸缩配置区域，勾选是否开启容器组水平伸缩，从而满足应用在不同负载下的需求。
- 在标签和注释区域，单击添加设置容器组的标签和注释。
- 单击创建。



<#>

阿里云

ASK应用管理 - 使用镜像创建应用

• 查看应用

- 在创建完成页签中单击查看应用详情。您可以看到新建的nginx-demo出现在无状态列表下。



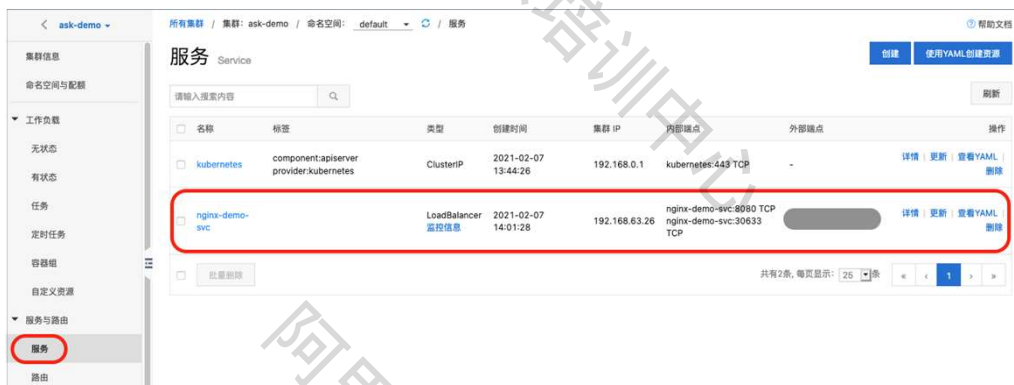
<#>

阿里云

ASK应用管理 - 使用镜像创建应用

• 查看应用

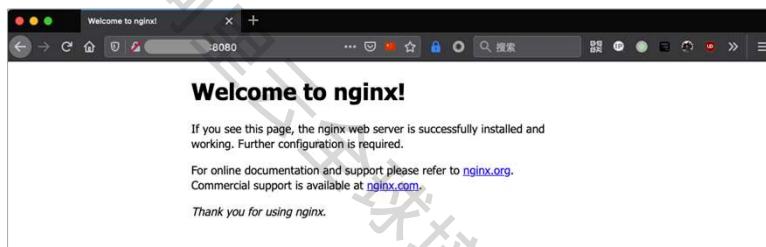
- 在集群管理页左侧导航栏中，单击**服务**，可以看到新建的服务nginx-demo-svc出现在服务列表下。



ASK应用管理 - 使用镜像创建应用

• 查看应用

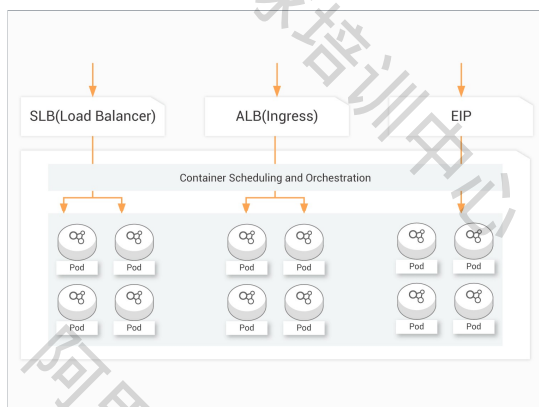
- 在浏览器中访问外部端点，您可访问Nginx欢迎页，本示例为Nginx。



ASK应用管理 – 服务管理

• 背景信息

- Kubernetes Service定义了这样一种抽象：一个Pod的逻辑分组，一种可以访问它们的策略，通常称为微服务。这一组Pod能够被Service访问到，通常是通过Label Selector来实现。



<#>

阿里云

ASK应用管理 – 服务管理

• 创建服务

- 登录容器服务控制台。
- 在控制台左侧导航栏中，单击**集群**。
- 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。



<#>

阿里云

ASK应用管理 – 服务管理

• 创建服务

- 在集群管理页左侧导航栏中，单击**服务**。
- 单击页面右上角的**创建**。
- 在弹出的创建服务对话框中，进行配置。
- 单击**创建**，nginx-demo-svc服务出现在服务列表中。



创建服务对话框，配置如下：

- 名称：nginx-demo-svc
- 类型：负载均衡
- 公网访问：公网访问
- 新建slb：新建slb
- 关联：nginx-demo
- 外部流量策略：Local
- 端口映射：

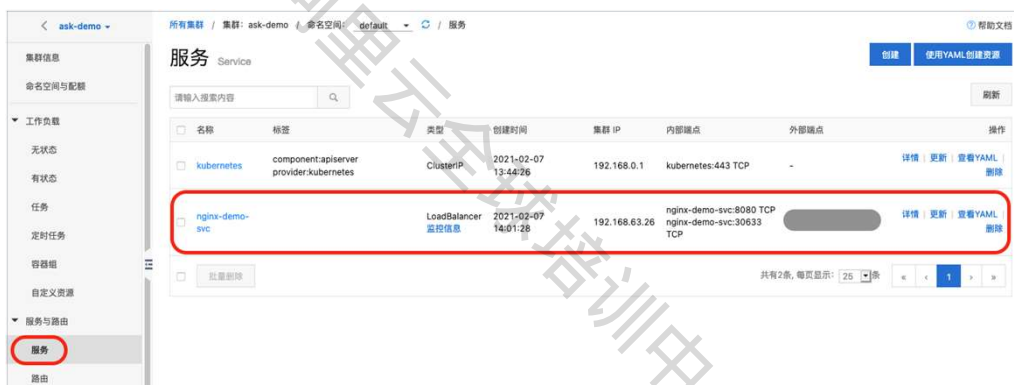
名称	服务器端口	容器端口	协议
nginx	8080	80	TCP
- 注解：添加
- 标签：添加

底部有**创建**和**取消**按钮。

ASK应用管理 – 服务管理

• 查看服务

- 可以在服务（Service）页面，查看部署的服务。



服务列表页面，显示如下：

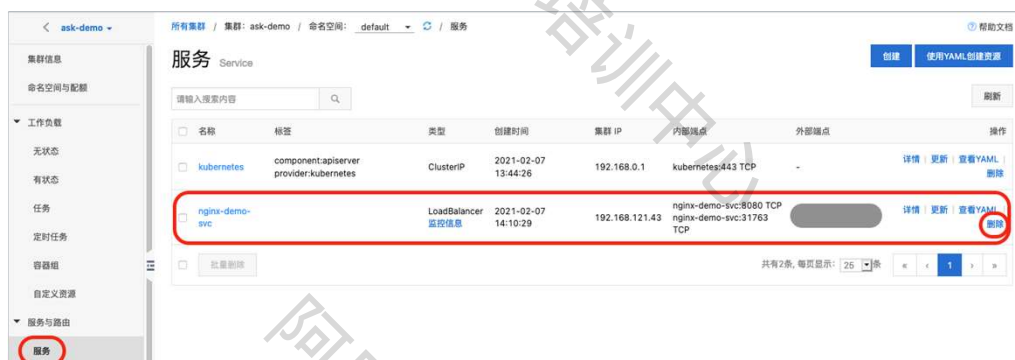
名称	标签	类型	创建时间	集群 IP	内部端口	外部端口	操作
kubernetes	component:apiserver provider:kubernetes	ClusterIP	2021-02-07 13:44:26	192.168.0.1	kubernetes:443 TCP	-	详情 更新 查看YAML 删除
nginx-demo-svc		LoadBalancer	2021-02-07 14:01:28	192.168.63.26	nginx-demo-svc:8080 TCP nginx-demo-svc:30633 TCP		详情 更新 查看YAML 删除

底部有**创建**和**使用YAML创建资源**按钮。

ASK应用管理 – 服务管理

• 删除服务

- 在集群管理页左侧导航栏中，单击服务。
- 选择目标服务，单击右侧操作列下的删除。



ASK应用管理 – 服务管理

• 删除服务

- 在弹出的窗口中，单击确定，确认删除，该服务在服务列表中消失。



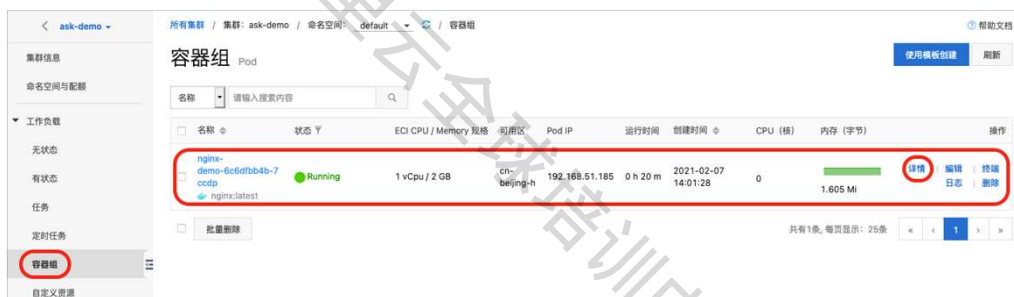
ASK应用管理 – 查看容器

- 登录容器服务控制台。
- 在控制台左侧导航栏中，单击**集群**。
- 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。



ASK应用管理 – 查看容器

- 在集群管理页左侧导航栏中，单击**工作负载**。
- 单击**容器组**页签。
- 选择目标容器组，单击右侧的**详情**。



ASK应用管理 – 查看容器

- 进入容器组的详情页，您可查看该容器组的详情信息。

The screenshot displays the ASK application management interface. On the left is a sidebar with navigation options: 集群信息, 命名空间与配额, 工作负载, 无状态, 有状态, 任务, 定时任务, 容器组 (selected), 自定义资源, 服务与路由, 配置管理, 存储, 应用, 运维管理, and 安全管理. The main content area shows the details for the container group 'nginx-demo-6c6dfbb4b-7ccdp'. It includes basic information such as name, namespace, status (Running), node, pod IP, and labels. Below this is a '现状详情' (Current Status) table showing the progression of the container group's state.

类型	状态	更新时间	原因	消息
Initialized	True	2021-02-07 14:01:40	-	-
Ready	True	2021-02-07 14:01:48	-	-
ContainersReady	True	2021-02-07 14:01:48	-	-
PodScheduled	True	2021-02-07 14:01:40	-	-

Below the status table, there are tabs for 容器 (Containers), 事件 (Events), 创建者 (Creator), 初始化容器 (Init Containers), 存储 (Storage), and 日志 (Logs). The '容器' tab is active, showing a table with columns for 名称 (Name), 镜像 (Image), and 端口 (Port). The table lists one container named 'nginx-demo' using the image 'registry.cn-hangzhou.aliyuncs.com/acs-sample/nginx:latest' on port 'TCP 80'.

本章小结

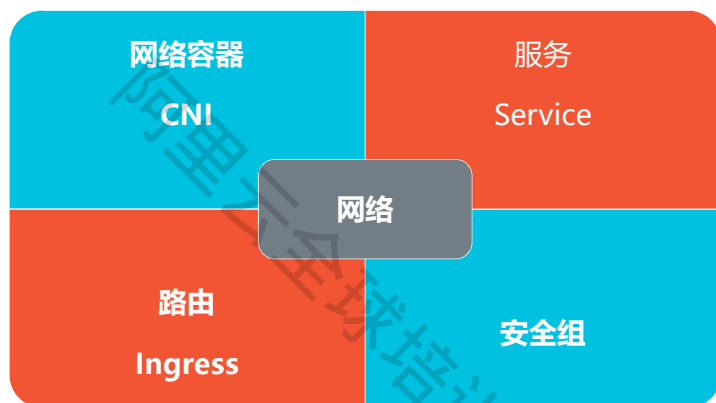
- ASK应用管理：
 - 了解ASK应用管理的主要功能。
 - 了解ASK服务创建、查看、删除的方式。
 - 了解ASK容器的查看。

课程目录

1. ASK集群管理
2. ASK应用管理
3. **ASK网络、存储、日志、监控管理**
 - 3.1 网络管理
 - 3.2 存储管理
 - 3.3 日志管理
 - 3.4 监控管理
4. ASK弹性伸缩管理
5. ECI Pod管理

<#>

ASK网络、存储、日志、监控管理 – 网络管理



<#>

ASK网络、存储、日志、监控管理 – 网络管理

• 容器网路CNI

- 阿里云Serverless Kubernetes服务、虚拟节点推出Pod挂载弹性公网IP功能，此功能使某些Serverless容器应用的部署和服务访问变得更加简单和便利。

无需创建VPC NAT网关即可让单个Pod访问公网

无需创建Service也可让单个Pod暴露公网服务

可以更加灵活而且动态的绑定Pod和EIP

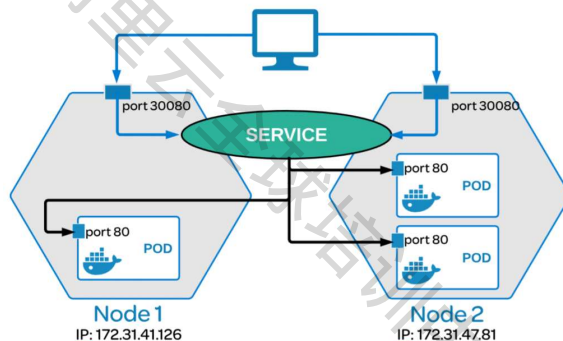
<#>

阿里云

ASK网络、存储、日志、监控管理 – 网络管理

• 服务 Service

- Kubernetes Service定义了这样一种抽象：一个Pod的逻辑分组，一种可以访问它们的策略，通常称为微服务。



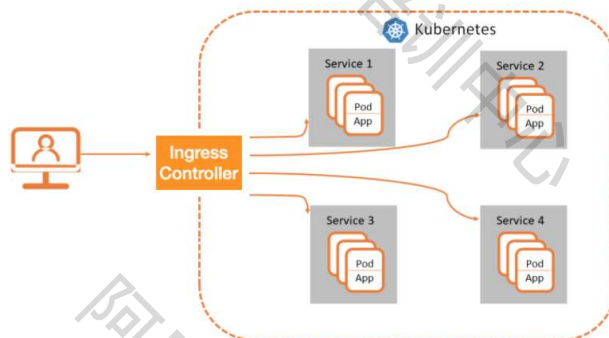
<#>

阿里云

ASK网络、存储、日志、监控管理 – 网络管理

• 路由 Ingress

- 在Kubernetes集群中，Ingress是Kubernetes中的一个资源对象，用来管理集群外部访问集群内部服务的方式。

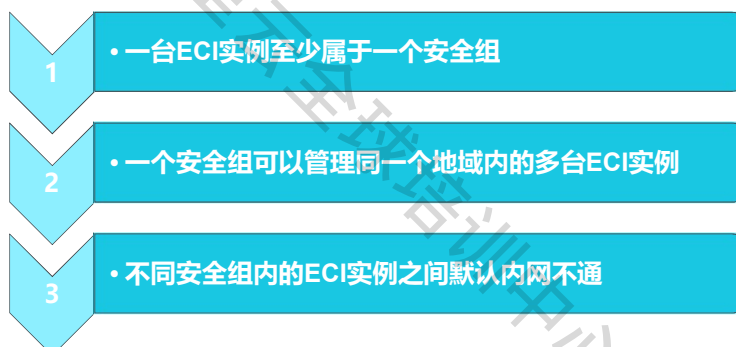


<#>

ASK网络、存储、日志、监控管理 – 网络管理

• 安全组

- 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。



<#>

ASK网络、存储、日志、监控管理 – 存储管理

• 概述

- 容器服务ASK支持自动绑定阿里云云盘、阿里云文件存储NAS（Network Attached Storage）、阿里云对象存储OSS（Object Storage Service）存储服务。
- 容器服务支持静态存储卷和动态存储卷，每种数据卷的支持情况如下。

阿里云存储	静态数据卷	动态数据卷
阿里云云盘	云盘静态存储卷支持以下两种使用方式： - 直接通过volume使用 - 通过PV/PVC使用	支持
阿里云NAS	NAS静态存储卷支持以下两种使用方式： - 通过flexvolume插件使用 - 通过volume方式使用 - 通过PV/PVC使用 - 通过Kubernetes的NFS驱动使用	不支持

<#>



ASK网络、存储、日志、监控管理 – 存储管理

• 存储插件说明

- 目前阿里云容器服务ASK集群支持两种存储插件Flexvolume和CSI。

• Flexvolume和CSI存储插件的区别：

插件名称	插件特性	参考文档
Flexvolume	Flexvolume插件是Kubernetes社区较早实现的存储卷扩展机制。ASK从上线起，即支持Flexvolume类型数据卷服务。Flexvolume插件包括以下三部分。 - Flexvolume：负责数据卷的挂载、卸载功能。ASK默认提供云盘、NAS两种存储卷的挂载能力。 - Disk-Controller：负责云盘卷的自动创建能力。 - Nas-Controller：负责NAS卷的自动创建能力。	有关Flexvolume的详细概述，请参见 Flexvolume概述 。 有关如何升级Flexvolume存储插件，请参见 管理组件 。
CSI	CSI插件是当前Kubernetes社区推荐的插件实现方案。ASK集群提供的CSI存储插件兼容社区的CSI特性。CSI插件包括以下两部分： - CSI-Plugin：实现数据卷的挂载、卸载功能。ASK默认提供云盘、NAS两种存储卷的挂载能力。 - CSI-Provisioner：实现数据卷的自动创建能力，目前支持云盘、NAS两种存储卷创建能力。	有关CSI的详细概述，请参见 CSI概述 和 alibaba-cloud-csi-driver 。 有关如何升级CSI存储插件，请参见 CSI存储插件升级指南 。

<#>



ASK网络、存储、日志、监控管理 – 日志管理

• 通过阿里云日志服务采集日志

- 打开集群工作负载下无状态页面，并点击使用模板创建。
- 新建并部署YAML模板。
 - 通过环境变量来创建您的采集配置，所有与配置相关的环境变量都采用aliyun_logs_作为前缀。
- 当YAML编写完成后，单击创建，即可将相应的配置交由集群执行。

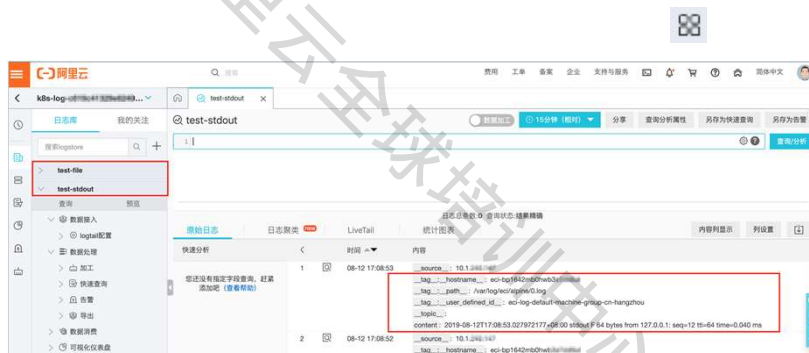
```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: alpine
    name: alpine
  name: alpine
spec:
  replicas: 2
  selector:
    matchLabels:
      app: alpine
  template:
    metadata:
      labels:
        app: alpine
    spec:
      containers:
        - image: alpine
          imagePullPolicy: Always
          args:
            - ping
            - 127.0.0.1
          name: alpine
          env:
            ##### 配置 环境变量 #####
            ##### project, 如果使用k8s集群默认的project可以不填 #####
            - name: aliyun_logs_test-stdout_project
              value: k8s-log-xxx
            - name: aliyun_logs_test-file_project
              value: k8s-log-xxx
            ##### 配置volume mount, 该volume就是为了上面的自定义的日志收集目录 #####
            volumeMounts:
              - name: volume-als
                mountPath: /log
            volumes:
              - name: volume-als
                emptyDir: {}
```

阿里云

ASK网络、存储、日志、监控管理 – 日志管理

• 查看日志

- 打开日志服务控制台，在Project列表区域选择Kubernetes集群对应的Project（默认为k8s-log-{Kubernetes集群ID}），进入日志库列表页签。
- 在列表中找到相应的Logstore（采集配置中指定），单击，在下拉框中选择查询分析。



阿里云

ASK网络、存储、日志、监控管理 – 监控管理

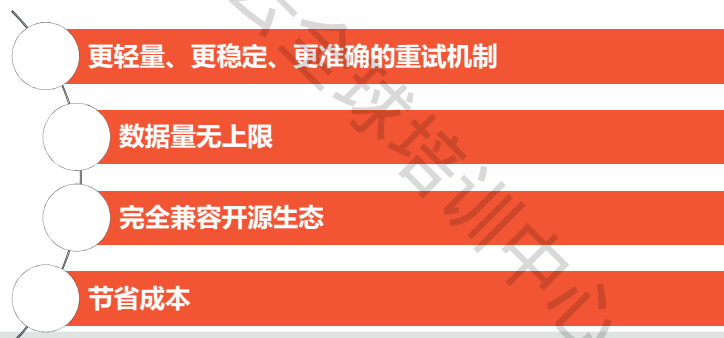
- 监控管理



<#>

ASK网络、存储、日志、监控管理 – 监控管理

- 阿里云Prometheus监控
 - 阿里云Prometheus监控全面对接开源Prometheus生态
 - 支持类型丰富的组件监控
 - 提供多种开箱即用的预置监控大盘
 - 提供全面托管的Prometheus服务。



<#>

ASK网络、存储、日志、监控管理 – 监控管理

• 开启阿里云Prometheus监控

- 登录容器服务管理控制台。
- 在控制台左侧导航栏中，选择**市场 > 应用目录**。
- 在应用目录页面的阿里云应用页签中，搜索并单击**ack-arms-prometheus**。
- 在右侧的**创建**区域中，选择目标集群，然后单击**创建**。



ASK网络、存储、日志、监控管理 – 监控管理

• 执行结果

- 单击ARMS 控制台右侧的**Prometheus监控**，找到目标集群并单击已安装的插件进入大盘查看监控数据。



本章小结

- **ASK网络、存储、日志、监控管理：**

1. 了解ASK以下管理模块的主要功能和特性。

- 网络管理
- 存储管理
- 日志管理
- 监控管理

课程目录

1. ASK集群管理
2. ASK应用管理
3. ASK网络、存储、日志、监控管理
4. **ASK弹性伸缩管理**
 - 4.1 概述
 - 4.2 容器水平伸缩
 - 4.3 容器定时伸缩
 - 4.4 阿里云指标容器水平伸缩
5. ECI Pod管理

ASK弹性伸缩管理 – 概述

• 背景介绍

- 弹性伸缩是ASK上被广泛采用的功能，典型的场景包含在线业务弹性、大规模计算训练、深度学习GPU或共享GPU的训练与推理、定时周期性负载变化等。



<#>

阿里云

ASK弹性伸缩管理 – 概述

• 调度层弹性组件

组件名称	组件介绍	适用场景	使用限制
HPA	Kubernetes内置组件，主要面向在线业务。	在线业务	适用于Deployment、StatefulSet等实现scale接口的对象。
VPA [alpha]	开源社区组件，主要面向大型单体应用。	大型单体应用	适用于无法水平扩展的应用，通常是在Pod出现异常恢复时生效。
CronHPA	ACK开源的组件，主要面向应用资源使用率存在周期性变化的场景。	周期性负载业务	适用于Deployment、StatefulSet等，实现了scale接口的对象。此外CronHPA提供了HPA对象的兼容能力，您可以同时使用CronHPA与HPA。
Elastic-Workload	ACK组件，主要面向精细化角度场景，例如希望在不同的可用区进行负载分布。	精细调度场景	适用于在线业务精细化控制的场景。例如，一个Deployment部分的应用运行在ECS上，而超出的部分运行在ECI上。

• 资源层弹性组件

组件名称	组件介绍	适用场景	资源交付速度
cluster-autoscaler	Kubernetes社区开源组件，节点水平伸缩组件，阿里云提供了独有的调度、弹性优化、成本优化的功能。	全场景支持，适合在线业务、深度学习、大规模成本算力交付等。	以1000节点为一个交付批次为例： - 标准模式：120s - 极速模式：60s - 标准模式-qboot：90s - 极速模式-qboot：45s

<#>

阿里云

ASK弹性伸缩管理 – 容器水平伸缩 (HPA)

• 通过容器服务控制台创建HPA应用

- 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。
- 在**工作负载 > 无状态**页面中，单击**使用镜像创建**。
- 在**应用基本信息**页填写应用的名称，设置应用部署集群和命名空间，单击**下一步**。

应用名称: nginx-demo
名称为1-63个字符，可包含数字、小写字母以及短划线 (-)，且不能以短划线 (-) 开头

副本数量: 1

类型: 无状态

标签: 添加

注解: 添加

应用基本信息 容器配置 高级配置 创建完成

应用1 添加容器

下一步

ASK弹性伸缩管理 – 容器水平伸缩 (HPA)

• 通过容器服务控制台创建HPA应用

- 在**容器配置**页进行容器设置，选择镜像，并设置所需的资源。然后单击**下一步**。

镜像名称: pc.cn-hangzhou.aliyuncs.com/acs-sample/nginx 选择镜像

镜像Tag: 选择镜像Tag

镜像拉取策略: 设置镜像密钥

所需资源: CPU 1Core 内存 2GB

容器启动项: ☐ stdin ☐ tty

Init Container: ☐

配置概述

基本配置
端口设置
环境变量
健康检查
生命周期
数据卷

应用1 添加容器

下一步

ASK弹性伸缩管理 – 容器水平伸缩 (HPA)

• 通过容器服务控制台创建HPA应用

- 在高级设置页的访问设置区域，单击服务的创建，设置服务的相关信息。
- 在高级设置页选中容器组水平伸缩的开启，设置伸缩的条件和配置。
- 单击右下角的创建，一个支持HPA的Deployment就已经创建完毕。

指标伸缩 ☒ 开启

指标: CPU使用量

触发条件: 使用量 70 %

最大副本数: 10 可选范围: 2-100

最小副本数: 1 可选范围: 1-100

定时伸缩 ☐ 开启

ASK弹性伸缩管理 – 容器水平伸缩 (HPA)

• 验证结果

- 在工作负载 > 无状态中单击应用名称或操作列的详情，单击容器伸缩，在详情中查看指标伸缩 (HPA) 信息。

nginx-demo

基本信息

名称: nginx-demo

命名空间: default

选择器: nginx-demo

注解: deployment.kubernetes.io/revision

状态: 就绪: 0/1个, 已更新: 1个, 可用: 0个

容器组 访问方式 事件 容器伸缩 历史版本 日志 触发器

指标伸缩 (HPA) 指标伸缩对象最多一个

名称	目标使用率	当前使用率	最小副本数	最大副本数	当前副本数	创建时间	操作
nginx-demo	CPU: 70%		1	10	1	2021-02-08 16:04:41	状态 事件 编辑 删除

定时伸缩 (CronHPA) ack-kubernetes-cronhpa-controller 组件未安装。 点击安装

名称	任务名称	状态	调度周期	目标副本数	最近调度	创建时间	操作
----	------	----	------	-------	------	------	----

ASK弹性伸缩管理 – 容器定时伸缩 (CronHPA)

通过容器服务控制台创建HPA应用

- 创建一个新的Deployment并设置相关信息。
- 在高级设置页选中**定时伸缩的开启**（如没有安装“ack-kubernetes-cronhpa-controller”组件，可使用**点击安装按钮**一键安装）。
- 设置**定时任务名称**、**目标副本数**、**调度周期**。
- 单击右下角的**创建**，一个支持定时伸缩的Deployment就已经创建完毕。



<#>



ASK弹性伸缩管理 – 容器定时伸缩 (CronHPA)

验证结果

- 在**工作负载 > 无状态**中单击应用名称或操作列的详情，单击**容器伸缩**，在详情中查看**定时伸缩 (CronHPA)** 信息。



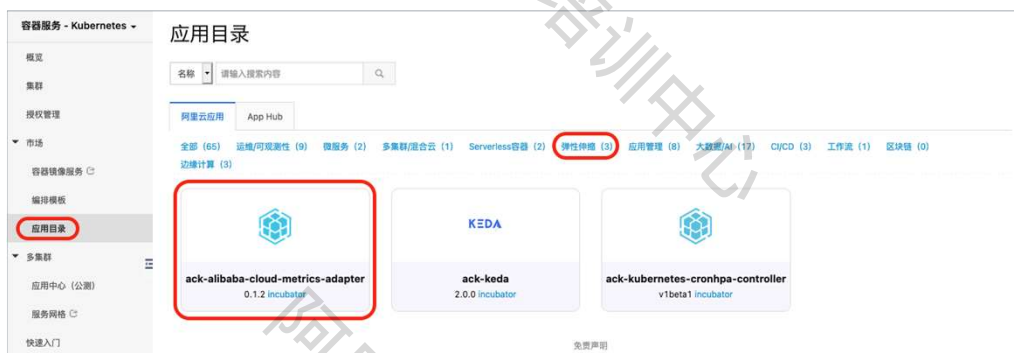
<#>



ASK弹性伸缩管理 – 阿里云指标容器水平伸缩

- 部署alibaba-cloud-metrics-adapter组件

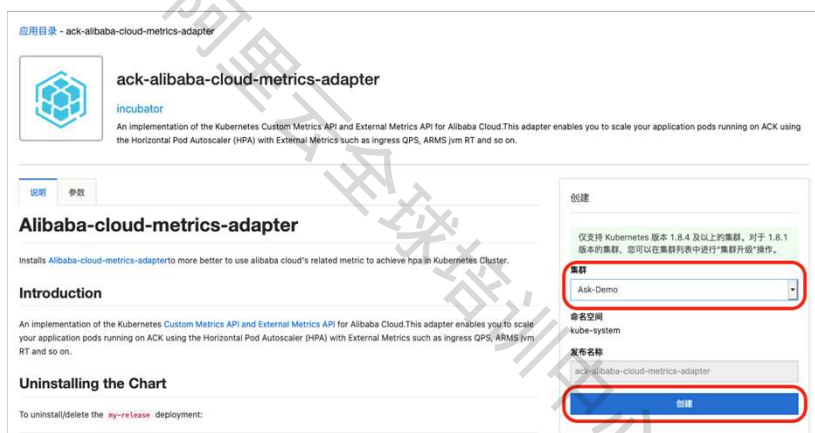
- 在容器服务管理控制台页面的左侧导航栏，选择市场 > 应用目录，并在应用目录页面，选择阿里云应用 > 弹性伸缩 > ack-alibaba-cloud-metrics-adapter。



ASK弹性伸缩管理 – 阿里云指标容器水平伸缩

- 部署alibaba-cloud-metrics-adapter组件

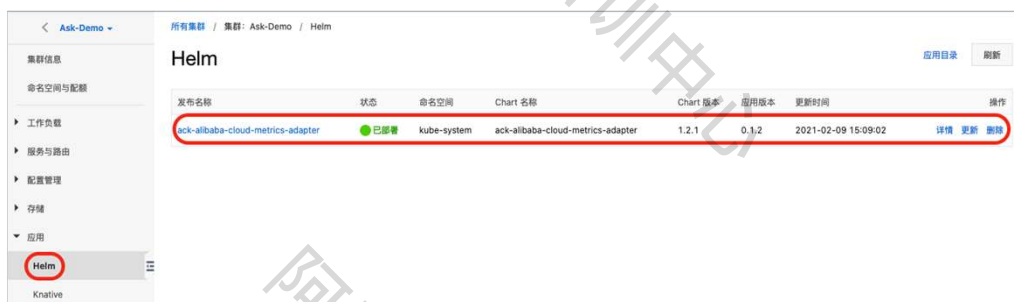
- 在右侧创建区域配置集群信息，并单击创建。



ASK弹性伸缩管理 – 阿里云指标容器水平伸缩

• 部署alibaba-cloud-metrics-adapter组件

- 在控制台左侧导航栏单击**集群**，在**集群列表**页面中，单击目标集群名称。
- 在集群管理页面选择**应用**，在Helm页面看到ack-alibaba-cloud-metrics-adapter组件已经成功部署到集群中。



<#>



ASK弹性伸缩管理 – 阿里云指标容器水平伸缩

• 示例

- 在**集群列表**页面中，单击目标集群右侧操作列下的**应用管理**。
- 在**无状态**页面中，单击**使用模板创建**。通过部署YAML文件创建一个应用，并暴露一个ClusterIP的Service，最后单击**创建**。
- 在**路由**页面中，在页面右上方，单击**创建**。在创建面板中填写相关信息，点击**创建**并完成后，系统自动跳转至**路由**页面。
- 在生成的Ingress地址右侧，单击**详情**查看路由信息。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: default
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
  type: ClusterIP
```

<#>



ASK弹性伸缩管理 – 阿里云指标容器水平伸缩

• 示例

- 在控制台左侧导航栏选择**市场** > **编排模板**，并在编排模板页面右上方，单击**创建**。
- 部署YAML文件创建一个HPA模板。
- 在**模板列表**页面，选择刚创建的模板并单击**创建应用**。

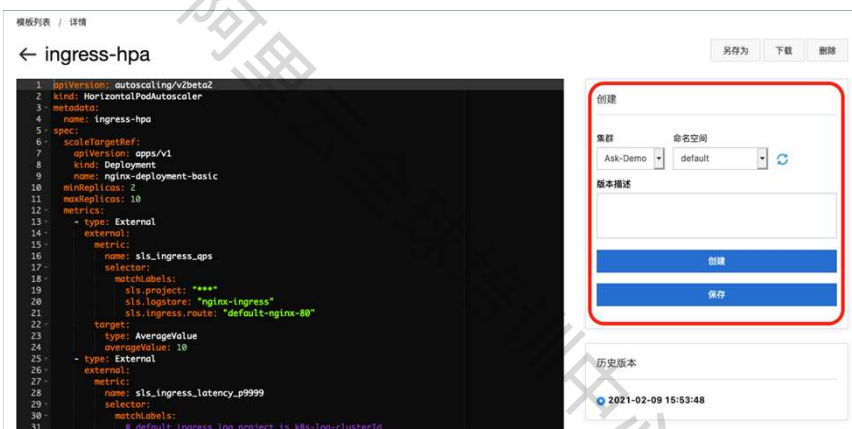


```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: ingress-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deployment-basic
  minReplicas: 2
  maxReplicas: 10
  metrics:
    - type: External
      external:
        metric:
          name: sls_ingress_qps
          selector:
            matchLabels:
              sls.project: "*"
              sls.logstore: "nginx-ingress"
              sls.ingress.router: "default-nginx-80"
          target:
            type: AverageValue
            averageValue: 10
    - type: External
      external:
        metric:
          name: sls_ingress_latency_p9999
          selector:
            matchLabels:
              # default ingress log project is k8s-log-clusterid
              sls.project: "*"
              # default ingress logstore is nginx-ingress
              sls.logstore: "nginx-ingress"
              # namespace-svc-port
              sls.ingress.router: "default-nginx-80"
              # sls vpc endpoint, default true
              # sls.internal.endpoint.ttl
          target:
            type: Value
            # sls_ingress_latency_p9999 > 10ms
            value: 10
```

ASK弹性伸缩管理 – 阿里云指标容器水平伸缩

• 示例

- 在创建面板中填写相关信息，然后单击**创建**。



ASK弹性伸缩管理 - 阿里云指标容器水平伸缩

- 示例

- 设置好HPA后，执行以下脚本进行压测。

```
#!/bin/bash
##使用Apache Benchmark对ingress暴露的服务设置并发为10，时间为300s的压测。
ab -t 300 -c 10 <ingress配置的域名>
```

- 验证指标伸缩状态。

```
$ kubectl get hpa ingress-hpa
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
ingress-hpa	Deployment/nginx-deployment-basic	27/10 (avg)	2	10	10	7m49s

<#>



本章小结

- ASK弹性伸缩管理：

1. 了解ASK弹性伸缩管理模块的主要功能。
2. 了解ASK弹性伸缩的两种种类和实现。
3. 了解ASK弹性伸缩的定制化实现。

<#>



课程目录

1. ASK集群管理
2. ASK应用管理
3. ASK网络、存储、日志、监控管理
4. ASK弹性伸缩管理
5. ECI Pod管理
 - 5.1 概述
 - 5.2 创建ECI Pod
 - 5.3 特殊实例

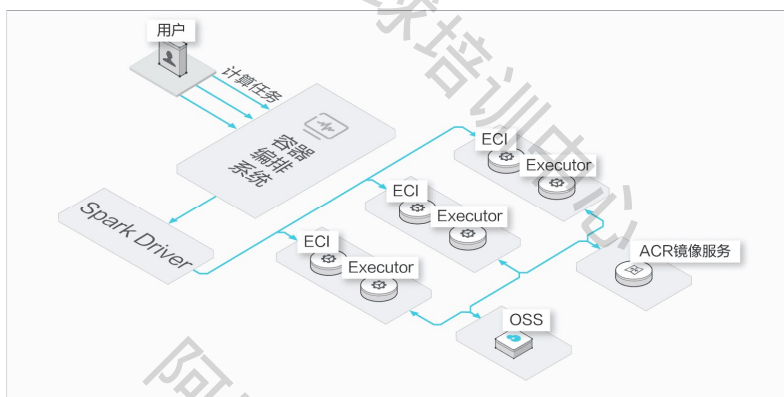
ECI Pod管理 – 概述

- 弹性容器实例ECI Pod的配置组成
 - 安全隔离
 - CPU/Memory资源和规格配置
 - 存储
 - 网络
 - 镜像拉取
 - 日志收集

ECI Pod管理 – 概述

- 安全隔离

- 每个ECI实例底层通过轻量级安全沙箱技术完全强隔离，实例间互不影响。



<#>

ECI Pod管理 – 概述

- CPU/Memory资源和规格配置
- ECI支持多种规格配置的方式来申请资源和计费。

指定CPU和Memory

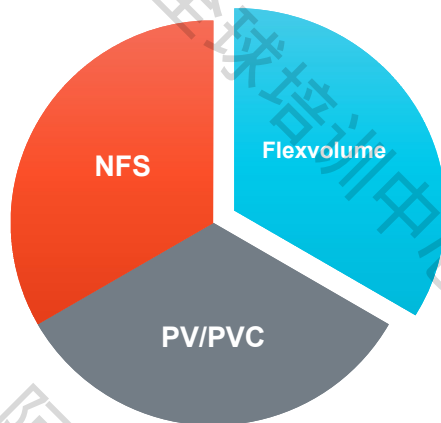
指定Pod的ECS规格

抢占式实例

<#>

ECI Pod管理 – 概述

- 存储
- 支持多种使用存储的方式：



<#>

ECI Pod管理 – 概述

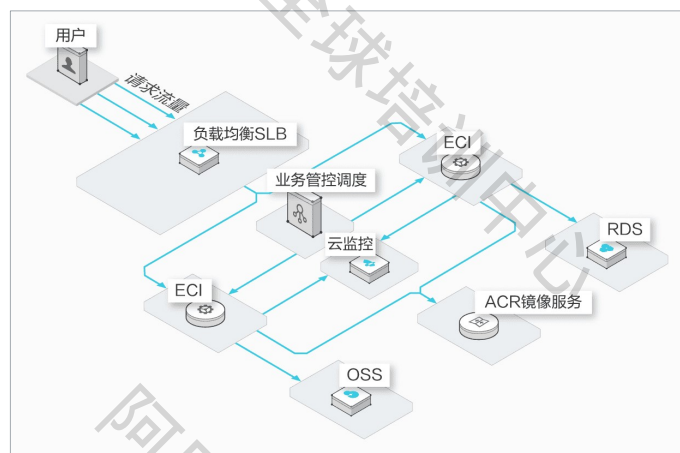
- 网络
 - ECI Pod默认使用Host网络模式，占用交换机vSwitch的一个弹性网卡ENI资源。



<#>

ECI Pod管理 – 概述

- 镜像拉取和日志收集



<#>

ECI Pod管理 – 创建ECI Pod

- 指定ECI实例内容器规格

- 通过Kubernetes标准方式配置单个容器的CPU和内存，ECI的资源则是Pod内所有容器所需资源的总和。
- 每个ECI实例支持最多20个容器，实例内每个容器的资源支持自定义配置，但汇总到ECI实例级别需要满足CPU和内存约束。

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
          resources:
            requests:
              cpu: "500m"
              memory: "1024Mi"
        - name: busybox
          image: busybox:latest
          ports:
            - containerPort: 80
          resources:
            requests:
              cpu: "500m"
              memory: "1024Mi"
```

<#>

ECI Pod管理 – 创建ECI Pod

- 指定ECI Pod规格

- 在指定ECI Pod级别的CPU和内存模式下，对于指定的CPU和内存规格，ECI会尝试使用多种ECS规格进行支撑，以提供比ECS单规格更好的库存和弹性能力。

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    annotations:
      k8s.aliyun.com/eci-use-specs: "2-4Gi"
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
        - name: busybox
          image: busybox:latest
          ports:
            - containerPort: 80
      resources:
        limits:
          cpu: "500m"
          memory: "1024Mi"
```

<#>



ECI Pod管理 – 创建ECI Pod

- 通过指定ECS规格创建ECI Pod

- ECI支持通过指定ECS规格进行创建。

- 目前支持的实例规格族如下所示:

- 通用型 (1:4) 实例规格族g6、g5、sn2ne (网络增强)
- 计算型 (1:2) 实例规格族c6、c5、sn1ne (网络增强)
- 内存型 (1:8) 实例规格族r6、r5、se1ne (网络增强)
- 密集计算型 (1:1) 实例规格族ic5
- 高主频计算型 (1:2) 实例规格族hfc6、hfc5
- 高主频通用型 (1:4) 实例规格族hfg6、hfg5
- GPU计算型实例规格族gn6i、gn6v、gn5i、gn5 (不支持本地存储)
- 突发性能实例规格族t6、t5

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    annotations:
      k8s.aliyun.com/eci-use-specs: "ecs.c5.large"
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

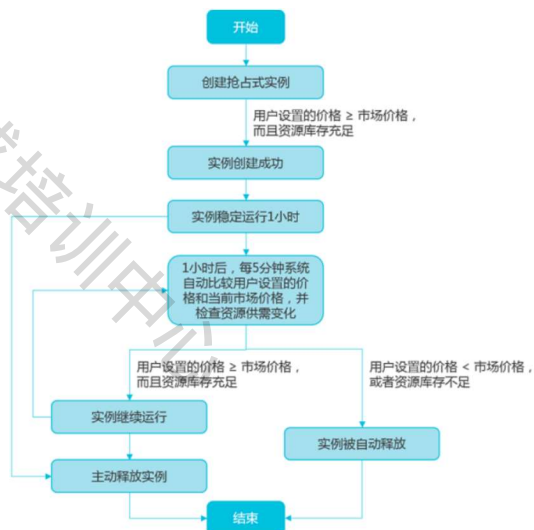
<#>



ECI Pod管理 – 特殊实例

• 抢占式实例

- 抢占式实例是一种低成本竞价型实例，您可以对阿里云当前闲置的资源出价，获得资源后运行容器，直到因为其他客户的更高出价而被回收容器资源，从而降低部分场景下使用ECI实例的成本。



<#>

ECI Pod管理 – 特殊实例

• 抢占式实例

- 抢占式实例适用于无状态的应用场景，例如可弹性伸缩的Web站点服务、图像渲染、大数据分析和大规模并行计算等。
- 应用程序的分布度、可扩展性和容错能力越高，越适合使用抢占式实例节省成本和提升吞吐量。



<#>

ECI Pod管理 – 特殊实例

• GPU实例

- ECI支持通过指定ECS GPU规格来创建ECI GPU实例。
- GPU实例内置了对应的Docker镜像，因此使用ECI GPU实例时无需安装Tensorflow、CUDA Toolkit等软件。

gn6v	• NVIDIA V100, 例如: ecs.gn6v-c8g1.2xlarge
gn6i	• NVIDIA T4, 例如: ecs.gn6i-c4g1.xlarge
gn5	• NVIDIA P100, 例如: ecs.gn5-c4g1.xlarge
gn5i	• NVIDIA P4, 例如: ecs.gn5i-c2g1.large

<#>



ECI Pod管理 – 特殊实例

• AMD实例

- ECI支持通过指定ECS AMD规格进行实例的创建。
- AMD实例依托神龙架构，将大量虚拟化功能卸载到专用硬件，降低虚拟化开销，提供稳定可预期的超高性能。



<#>



ECI Pod管理 – 特殊实例

- 多可用区的ECI Pod

- 使用ASK的多可用区特性可以提高ECI实例的创建成功率。

集群名称: ECI-multizone

资源组: 未选择

地域: 华北2 (北京)

Kubernetes 版本: 1.16.9-aliyun.1

专有网络: defaultvpc

虚拟交换机: 选择 1~10 个虚拟交换机, 为保证集群高可用, 建议选择不同可用区的虚拟交换机。

名称	ID	可用区	CIDR
vsw-1124	vsw-1124	华北2 (北京) 可用区E	172.17.0.0/16
vsw-1124-1	vsw-1124-1	华北2 (北京) 可用区E	172.17.0.0/16

<#>

阿里云

课程回顾

- 本节课主要包括:

1. 了解如何对ASK集群进行管理
2. 了解如何对ASK应用进行管理
3. 了解ASK网络、存储、日志、监控管理的功能
4. 了解ASK弹性伸缩的功能和使用方式
5. 了解ECI Pod的特性和使用方式

<#>

阿里云

阿里云

阿里云云原生容器ACP认证培训

Serverless Kubernetes (ASK) 场景应用

阿里云

课程目标

学习完本课程后，你将能够：

1. 了解ASK在线业务弹性伸缩的概况
2. 了解大数据Spark on Kubernetes的概况
3. 了解Knative Serverless应用的概况
4. 了解CI/CD流水线解决方案的概况

课程目录

1. 在线业务弹性伸缩
 - 1.1 弹性伸缩特性概述
 - 1.2 容器水平伸缩
 - 1.3 容器定时伸缩
 - 1.4 指标容器水平伸缩
2. 大数据 Spark on Kubernetes
3. Knative Serverless应用
4. CI/CD流水线解决方案

弹性伸缩特性概述

- 特性
 - 根据业务需求和策略
 - 经济地自动调整弹性计算资源
- 维度
 - 调度层弹性：修改负载的调度容量变化
 - 资源层弹性：补充ECI和集群容量规划
- 场景
 - 在线业务弹性
 - 大规模计算训练
 - 深度学习或共享GPU的训练与推理
 - 定时周期性负载变化

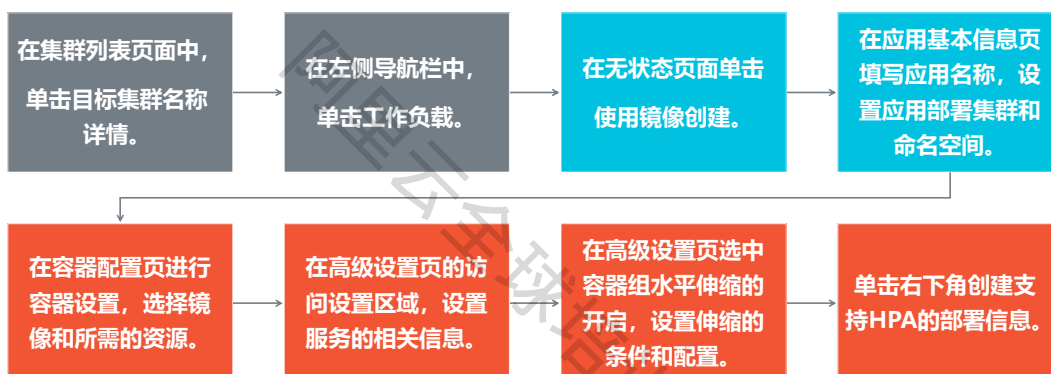


<#>

阿里云

容器水平伸缩

- 容器水平伸缩操作步骤：



- 查看应用详情，单击容器组水平伸缩，可在部署的详情中查看伸缩组信息
- 在实际使用环境中，应用会根据CPU负载进行伸缩

<#>

阿里云

容器定时伸缩

- kubernetes-cronhpa-controller：按照类似crontab的策略，定时地对容器服务进行扩缩容

```
apiVersion:
  autoscaling.alibabacloud.com/v1beta1
kind: CronHorizontalPodAutoscaler
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: cronhpa-sample
  namespace: default
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deployment-basic
  excludeDates:
    # exclude November 15th
    - " * * * 15 11 * *"
    # exclude every Friday
    - " * * * * 5"
  jobs:
    - name: "scale-down"
      schedule: "30 */1 * * * *"
      targetSize: 1
    - name: "scale-up"
      schedule: "0 */1 * * * *"
      targetSize: 3
    runOnce: true
```

scaleTargetRef

指定去扩缩容对象。

如果对象支持scale子资源，cronhpa即可支持。

excludeDates

日期数组。

当遇到符合excludeDates描述的日期时任务将会被跳过。

最小单位为天。

如您想在11月15日不运行任务，可像以下示例一样指定excludeDates。

excludeDates: - " * * * 15 11 * *"

jobs

支持在一个spec中设定多个cronhpa任务。

每个cronhpa任务可以配置以下字段：

name：唯一，可以通过name来区分不同的任务。

schedule：策略和crontab类似。

targetSize：到调度时间时，您想扩缩容到的Pod的数目。

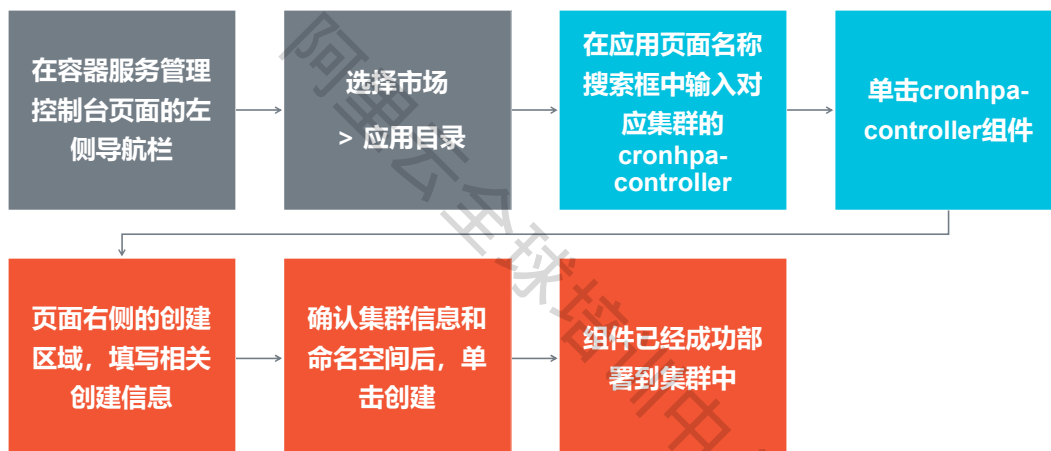
runOnce：如果设置为true，那么任务将只执行一次。

<#>

阿里云

容器定时伸缩

- kubernetes-cronhpa-controller 组件安装



<#>

阿里云

指标容器水平伸缩

- alibaba-cloud-metrics-adapter：由监测指标来指导集群的扩缩容



<#>

课程目录

1. 在线业务弹性伸缩
2. 大数据 Spark on Kubernetes
 - 2.1 Apache Spark 概述
 - 2.2 ASK 运行 Spark 的优势
 - 2.3 运行 Apache Spark 的工具
3. Knative Serverless应用
4. CI/CD流水线解决方案

<#>

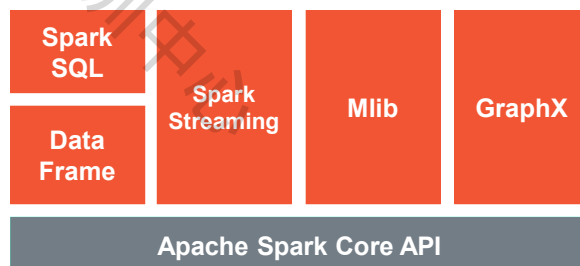
Apache Spark 概述

- 应用角色

- 数据分析领域快速构建集群计算框架
- 大数据和机器学习工作负载

- 应用架构

- Spark SQL / DF : 处理结构化和半结构化数据
- Spark Streaming : 处理实时流数据
- Mlib : 机器学习库
- GraphX : 图形计算库
- Apache Spark Core API : 应用程序的平台



<#>



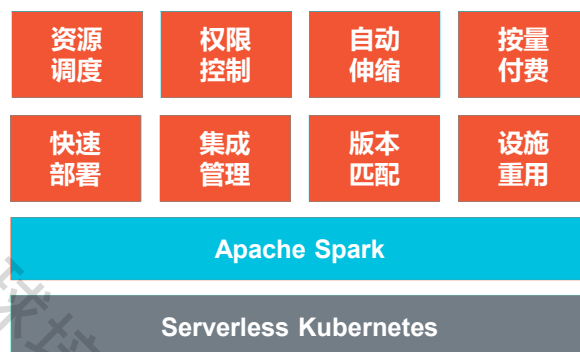
ASK 运行 Spark 的优势

- 在 Kubernetes 上运行Spark工作负载

- 应用的快速部署
- 集成化生命周期管理
- 解决版本匹配、兼容和依赖问题
- 重用基础设施，减少运维成本
- 支持多租户和用户粒度的资源调度
- 基于Kubernetes的控制权限

- 在 ASK 集群上运行 Spark 工作负载

- 按需按量创建 Pod
- 结束后停止计费
- 无需为Spark计算任务预留计算资源
- 无需担心集群算力扩容问题



<#>



运行 Apache Spark 的工具

- Spark Operator
 - 由Google官方支持的产品 spark-on-k8s-operator
 - 在Kubernetes上像其他工作负载一样用通用的方式方便地运行Spark应用
 - 使用Kubernetes custom resource来配置、运行Spark应用，并展现其状态
- Alluxio
 - 面向云的数据分析和人工智能的开源的数据编排技术。
 - 为数据驱动型应用和存储系统构建桥梁，使数据更容易被访问。
- TPC-DS Benchmark
 - 社区支持的第三方性能压测工具，协助确定解决方案的工业标准



<#>



课程目录

1. 在线业务弹性伸缩
2. 大数据 Spark on Kubernetes
3. Knative Serverless应用
 - 3.1 Knative 简介
 - 3.2 ASK Knative 优势
 - 3.3 ASK Knative 最佳实践
4. CI/CD流水线解决方案

<#>



Knative 简介

- 角色

- 基于 Kubernetes 的 Serverless 解决方案
- 标准化 Serverless 技术架构
- 简化学习成本

- 组件

- Build: 源到容器的构建和编排
- Event: 消息传递层, 事件交付管理
- Serving: 请求计算, 基于负载自动伸缩

- 优势

- 便利性
- 生态成熟
- 标准化
- 自动伸缩
- 服务间解耦
- 应用监控



<#>

ASK Knative 优势

	社区 Knative	ASK Knative
资源托管	• 默认使用Istio作为Gateway • 安装Istio的Controller需要额外支出部分IaaS成本 • Knative自身的Controller部署 • 需要额外支出部分IaaS成本	• ASK上提供了Knative的托管服务 • 无需为Knative Controller付出任何成本
服务网关	• 默认支持Istio、Gloo、Contour等多种实现方案	• 通过阿里云SLB实现了Knative Gateway • 不需要常驻资源
保留实例	• 默认在没有流量时可以把应用实例缩容到零 • 从零到一创建一个Pod是有明显的冷启动时间	• 在没有流量的时候不把应用实例数缩容到零个 • 通过低成本的保留实例来平衡成本和冷启动时长

<#>

ASK Knative 最佳实践

- 观测服务的QPS、RT和Pod扩缩容趋势
 - 配置Logstore
 - 配置QPS统计
 - 配置RT统计
 - 配置Pod扩缩容趋势统计
 - 观测服务运行状况
- 观测服务的CPU和Memory使用情况
 - ASK集群已开通Knative功能
 - 集群已开通阿里云Prometheus监控功能
 - 配置Prometheus监控目标ASK集群



<#>

阿里云

课程目录

1. 在线业务弹性伸缩
2. 大数据 Spark on Kubernetes
3. Knative Serverless应用
4. CI/CD流水线解决方案
 - 4.1 方案背景
 - 4.2 方案架构
 - 4.3 实践概要

<#>

阿里云

方案背景

- 场景需求

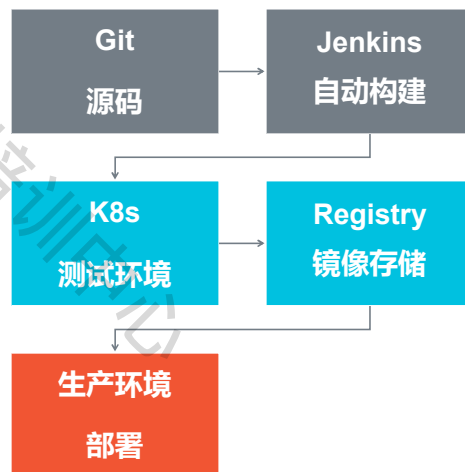
- 基于Jenkins构建自动化CI/CD集群系统
- 集群资源合理利用，控制成本
- 集群高可用性需求
- 集群资源快速弹性伸缩

- 方案优势

- 高可用服务
- 自动弹性伸缩，资源合理利用
- 可扩展性好

- 解决问题

- 集群Master节点单点故障
- 集群资源利用率低
- 集群资源可扩展性差



<#>

方案架构

- 服务高可用

- 避免Master单点故障导致集群流程不可用

- 弹性伸缩

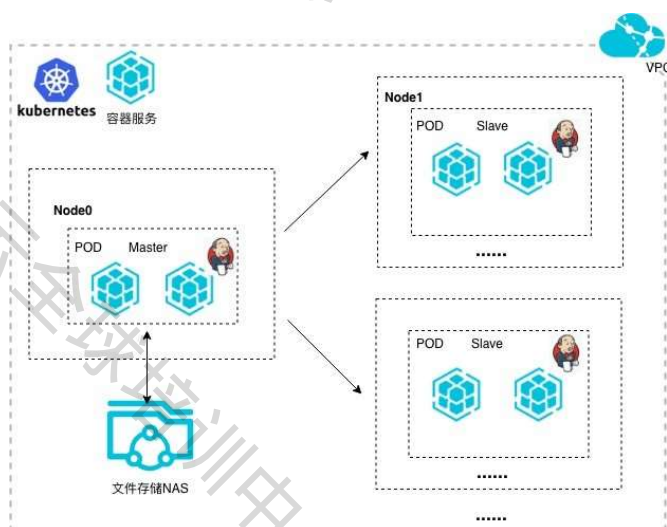
- 每次运行Job时，自动构建Jenkins Slave
- Job完成以后，自动注销并删除容器
- 资源自动释放，节省成本

- 资源合理利用

- 动态分配Slave到空闲节点
- 降低出现由于节点资源限制的排队等待情况

- 扩展性好

- 集群资源严重不足时，可以快速添加节点
- 降低集群资源不足导致Job排队等待情况



<#>

实践概要



<#>

课程小结

- 本节课的内容主要包括:

1. ASK的弹性伸缩特性
2. ASK的弹性伸缩部署流程
3. Apache Spark 在 ASK 上的部署方案
4. ASK Knative 的优势和最佳实践
5. ASK 上部署 CI/CD 流水线的解决方案

<#>



阿里云云原生容器ACP认证培训

服务网格 ASM



课程基础

学习本课程，需要了解以下前置知识：

1. 云原生
2. Kubernetes
3. 微服务架构
4. 服务网格架构
5. Envoy Proxy
6. Istio
7. 阿里云容器服务



kubernetes



ACK | ASK



envoy



Istio

<#>

阿里云

课程目标

学习完本课程后，你将能够：

1. 了解阿里云服务网格的功能架构
2. 了解阿里云服务网格的产品特点
3. 了解阿里云服务网格的应用场景
4. 掌握阿里云服务网格的实例管理
5. 掌握阿里云服务网格的网关配置
6. 掌握阿里云服务网格的应用部署
7. 掌握阿里云服务网格的资源定义

<#>

阿里云

课程目录

1. 阿里云服务网格概述

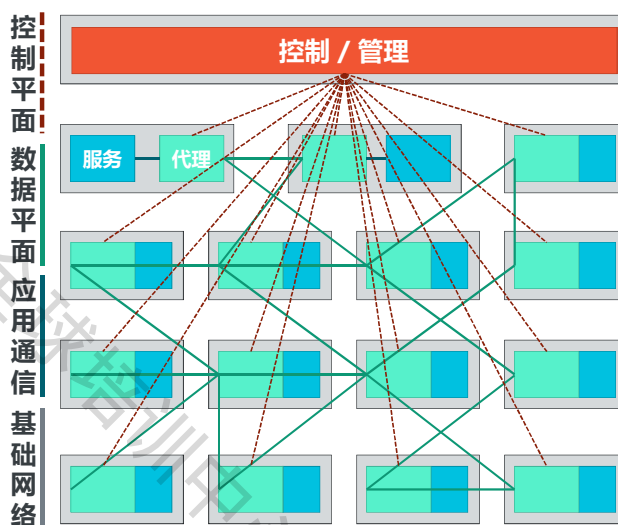
- 1.1 阿里云服务网格的技术概要
- 1.2 阿里云服务网格的产品架构
- 1.3 阿里云服务网格的功能特点
- 1.4 阿里云服务网格的产品优势
- 1.5 阿里云服务网格的应用场景

2. 阿里云服务网格使用

阿里云服务网格概述 – 技术概要

• 服务网格 Service Mesh

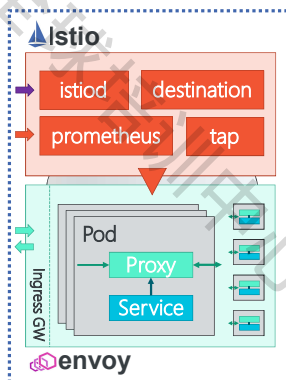
- 基础设施层
- 微服务通信
 - 保证请求可靠传递
 - 隐藏通信细节
 - 减少框架耦合
 - 语言无关
- 数据平面
 - 边车模式
 - 通信代理服务
 - 独立、平行、同步、透明
- 控制平面
 - 集中式控制面板
 - 统一运维功能



阿里云服务网格概述 – 技术概要

• 服务网格平台 Istio

- 连接
 - 通信路由
 - 流量管理
- 安全
 - 加密通信
 - 身份验证
- 控制
 - 自动策略
 - 分布式资源调配
- 观察
 - 量度指标
 - 日志
 - 跟踪



<#>

阿里云

阿里云服务网格概述 – 技术概要

• 阿里云服务网格 Alibaba Cloud Service Mesh (ASM)

- 统一管理微服务应用流量、兼容Istio的托管式平台
- 通过流量控制、网格观测以及服务间通信安全等功能，服务网格可以全方位地简化服务治理
- 为运行在异构计算基础设施上的服务提供统一的管理能力
- 适用于Kubernetes集群、Serverless Kubernetes集群、ECS虚拟机以及自建集群
- 与阿里云产品深度融合，历经数个版本迭代优化，多区域覆盖覆盖



<#>

阿里云

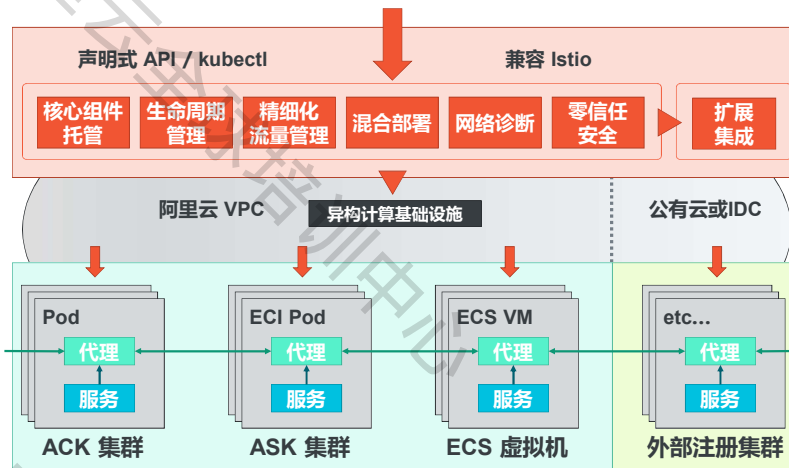
阿里云服务网格概述 – 产品架构

• 控制平面

- 完全托管的Istio实例
- 与开源社区兼容
- 声明式定义路由规则
- 统一服务流量管理
- 阿里云组件集成

• 数据平面

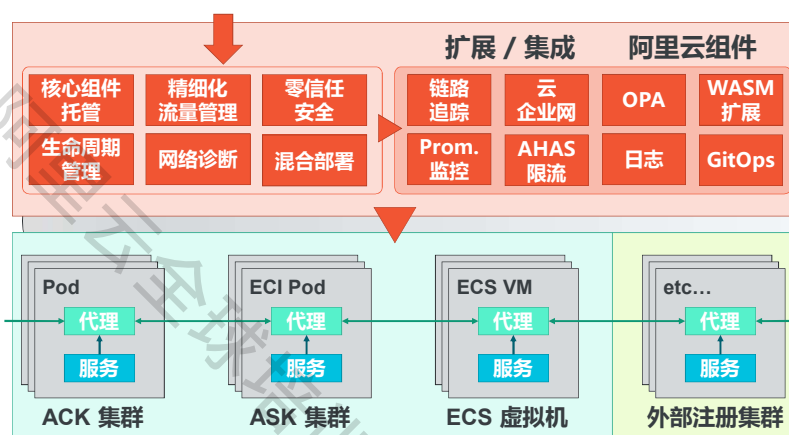
- Kubernetes集群的应用服务
- ECI Pod上的应用服务
- 声明式定义路由规则
- 统一服务流量管理



阿里云服务网格概述 – 产品架构

• 扩展与集成

- 可观测性服务
- 网络互连CEN
- 整合社区开源软件
 - OPA
 - 授权服务
- WebAssembly
 - 协议
 - 指标
 - 日志
 - 过滤器
- GitOps
 - CI / CD



阿里云服务网格概述 – 功能特点

• 一致的管理方式

- 以一致的方式来管理运行于不同集群的应用服务
- 支持混合云、多云场景和异构环境
- 提供一致的可观测性和流量控制

• 统一的流量管理

- 支持容器或者虚拟机混合环境下统一的流量管理

• 控制平面核心组件托管化

- 托管控制平面的核心组件
- 最大限度地降低用户资源开销和运维成本

安装
升级
回滚

流量拦截
重定向

协议
支持

多集群
支持

Sidecar
代理部署

CRD
支持

Istio入口
网关负载
均衡

用户界面

负载均衡

安全策略

可观测性

Istio Mixer
代替方案

<#>

阿里云

阿里云服务网格概述 – 功能特点

功
能
场
景

流量管理

- 基于配置的流量管理
- 将流量管理与基础设施管理分隔，独立流量管理过程
- 管理网格内的服务发现、流量路由和负载均衡，简化服务级属性的配置

服务安全

- 服务之间的双向TLS认证
- 服务之间的授权和不同粒度的权限访问控制
- 自动生成、分发、轮换与撤销密钥和证书

故障恢复

- 开箱即用的故障恢复功能
- 基于Istio的混沌工程能力
- 熔断、重试、故障注入

可观测性

- 实现服务之间的可观测性
- 借助监控功能，快速有效地检测和修复问题
- 通过阿里云链路追踪服务，提高开发诊断效率

微服务架构

- 实现敏捷开发和部署落地，加速企业业务迭代
- 基于阿里云的应用镜像仓库
- 基于阿里云的服务调度、编排、部署和灰度发布能力

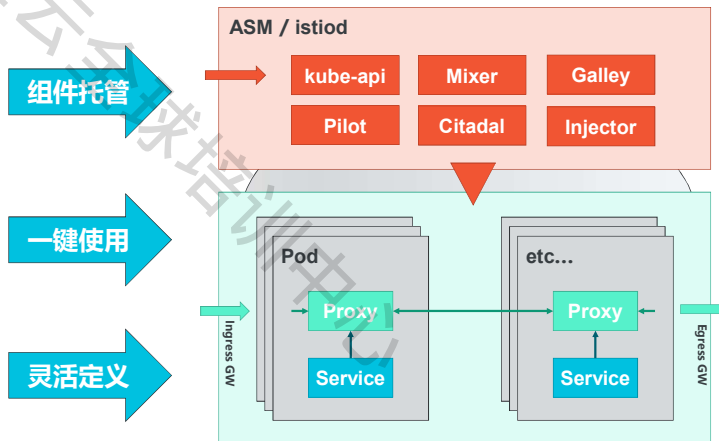
<#>

阿里云

阿里云服务网格概述 – 产品优势

• 使用便捷

- 部署挑战
 - 复杂的架构和组件关系
 - 大量个性化配置
 - 多集群、混合云、异构应用
- 基于托管组件
 - 提高社区兼容性
 - 高可用、免运维、内建安全
 - 数据平面可观测性
 - 统一管理异构混合云
- 一键式使用
 - 创建服务网格
 - 升级服务网格
- 灵活定义资源
 - Istio 虚拟服务
 - 目标规则
 - 入口网关



<#>

阿里云服务网格概述 – 产品优势

功能体系	完全托管	• 一键安装部署升级 • 控制平面组件托管 • 只需专注于业务应用开发 • 兼容 Istio 社区规范
	流量路由	• 自定义流量路由规则 • 跨多个 Kubernetes 集群 • 精细化流量管理 • 开箱即用混沌工程能力
	安全功能	• 一键启用或禁用 mTLS • 以渐进方式逐步实现 mTLS • 简单易用的 RBAC 功能 • 易于第三方系统对接
	监控诊断	• 提供网格诊断能力 • 集成托管式跟踪、监控和日志服务 • 实现端到端的可见性 • 生态系统对接
	高稳定性	• 专门的团队保障托管服务的稳定性
	高可用性	• 控制平面区分规格 • 面向不同业务需求定制 SLA • 支持多可用区
	技术支持	• 控制平面核心组件托管化 • 提供工单技术支持 • 降低运维复杂性

<#>

阿里云服务网格概述 – 应用场景

• 多语言应用微服务治理

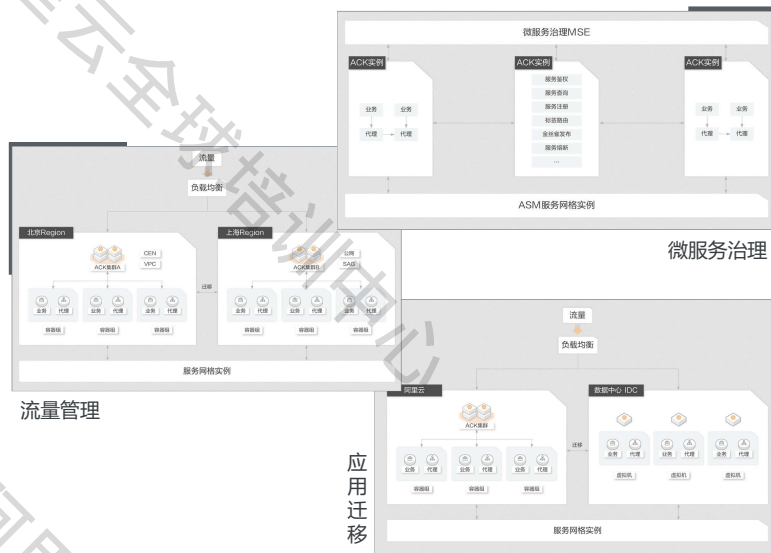
- 金丝雀发布
- 无损上下线
- 服务鉴权
- 标签路由

• 多集群应用统一流量管理

- 多地域
- 混合云
- 灵活灾备

• 应用容器化平滑上云

- 存量应用迁移
- 无状态服务架构升级



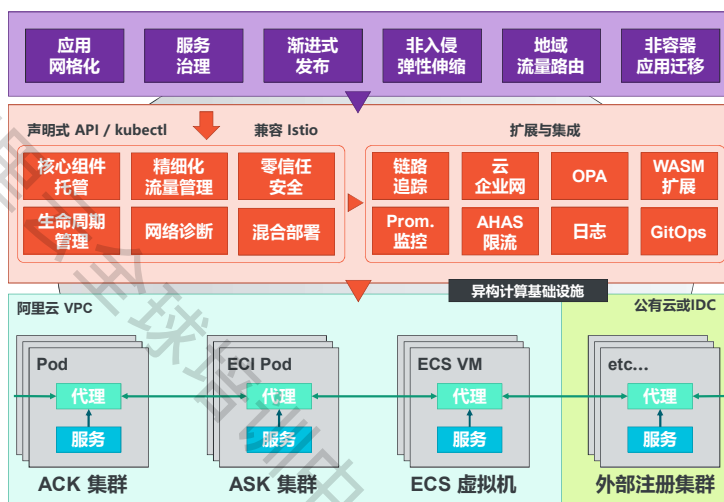
阿里云服务网格概述 – 应用场景

• 商米科技

- 维护更新迭代
- 负载均衡
- 链路追踪
- 流量统一管理

• 东风日产

- 原环境性能瓶颈
- 按需配置
- 增量部署
- 负载均衡
- 流量精细化管理
- 大数据安全



课程目录

1. 阿里云服务网格概述
2. 阿里云服务网格使用
 - 2.1 ASM入门指引
 - 2.2 创建 ASM 实例
 - 2.3 添加集群到 ASM 实例
 - 2.4 添加入口网关服务
 - 2.5 部署应用
 - 2.6 定义 Istio 资源

<#>



阿里云服务网格使用 – 入门指引

• 部署 ASM 的基础环境

- 阿里云 ECS 弹性计算云服务器
- 阿里云 Kubernetes 集群 (ACK / ASK)

• 部署 ASM 的主要步骤

- 创建一个服务网格ASM 实例（控制平面）
- 添加一个Kubernetes集群（数据平面）到服务网格ASM 实例（控制平面）中
- 为服务网格 ASM 实例（数据平面）中的集群部署入口网关
- 将应用部署到上述服务网格 ASM 实例的ACK Kubernetes集群（数据平面）中
- 为服务网格ASM 实例（控制平面）定义虚拟服务和 Istio 网关



<#>



阿里云服务网格使用 – 入门指引

- 入口：产品 -> 弹性计算 -> 容器服务 -> 服务网格 ASM



阿里云服务网格使用 – 创建实例

前提条件

- 已开通以下服务
 - 服务网格 ASM
 - 容器服务
 - 弹性伸缩 (ESS) 服务
 - 访问控制 (RAM) 服务
 - 链路追踪服务 (如需启用链路追踪功能)
- 已获得以下角色授权
 - AliyunServiceMeshDefaultRole
 - AliyunCSClusterRole
 - AliyunCSManagedKubernetesRole

容器服务

服务网格

弹性伸缩

访问控制

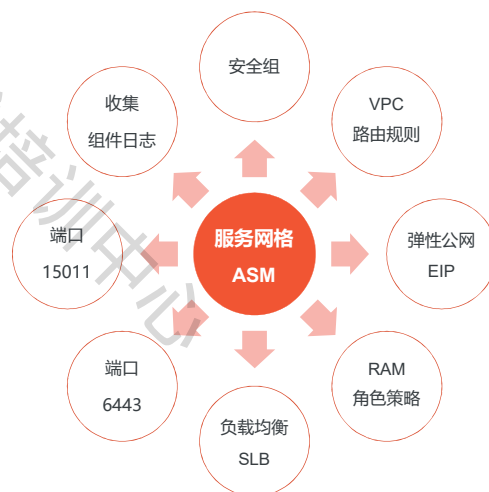
链路追踪



阿里云服务网格使用 – 创建实例

• 背景信息

- 创建ASM可能会进行如下操作：
 - 创建安全组
 - 创建VPC路由规则
 - 创建弹性公网IP (EIP)
 - 创建RAM角色及相应策略
 - 创建专有网的负载均衡 (SLB)
 - 暴露 6443 端口
 - 暴露 15011 端口
 - 收集被托管管控组件的日志信息



<#>

阿里云服务网格使用 – 创建实例

• 基本步骤

- 登录ASM控制台
- 在左侧导航栏, 选择服务网格 -> 网格管理
- 在网格管理页面单击创建新网格

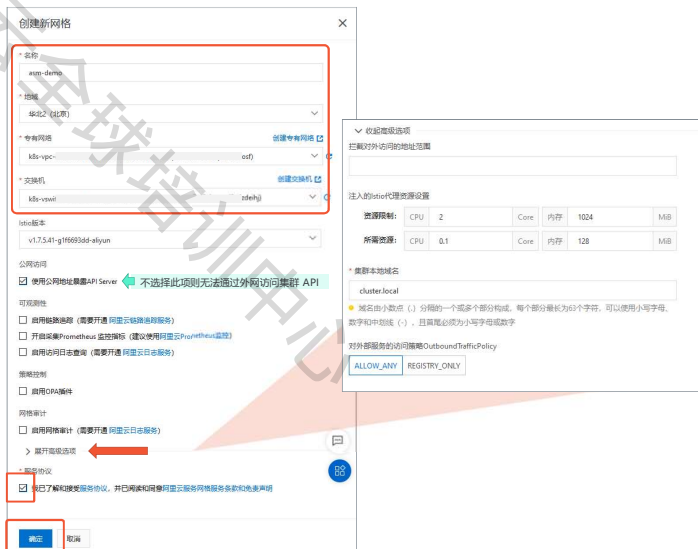


<#>

阿里云服务网格使用 – 创建实例

• 基本步骤

4. 在创建新网格页面，完成网格配置
 - 设置网格基础选项
 - 设置网格高级选项
5. 了解和接受服务协议
6. 单击确定，开始实例的创建
 - 一个ASM实例的创建时间一般约为2到3分钟



阿里云服务网格使用 – 创建实例

• 执行结果

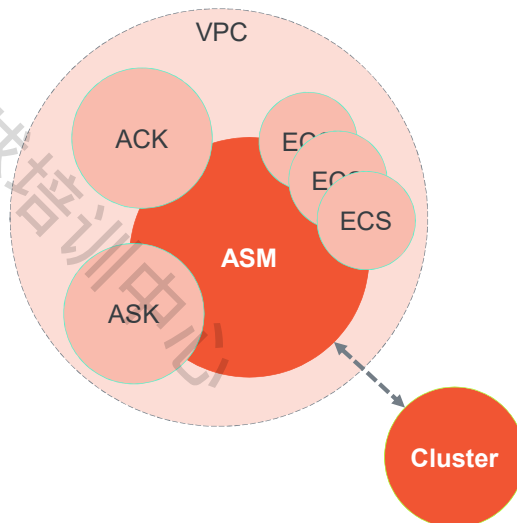
- 网格实例页面显示实例列表
- 查看已创建的实例摘要信息
- 查看实例相关的日志信息
- 查看实例相关的管理细节
 - 基本信息
 - 连接配置
 - 数据平面集群信息
 - 控制平面命名空间
 - 虚拟服务
 - 目标规则
 - Istio资源定义

网络管理						
创建新网格						
名称/ID	地域	专有网络	创建时间	状态	操作	
asm-demo cb82c5e	华北2 (北京)	vpc-2zevt...	2021年2月20日 12:19:27	初始化	管理	日志

阿里云服务网格使用 – 添加集群

• 前提条件

- 已创建至少一个ASM实例
- 已创建至少一个集群
 - Kubernetes专有版集群
 - Kubernetes托管版集群
 - Serverless ASK集群
 - ECS虚拟机
- 待添加的集群与ASM实例位于同一VPC
 - 或集群已开启公网API Server以便快速接入



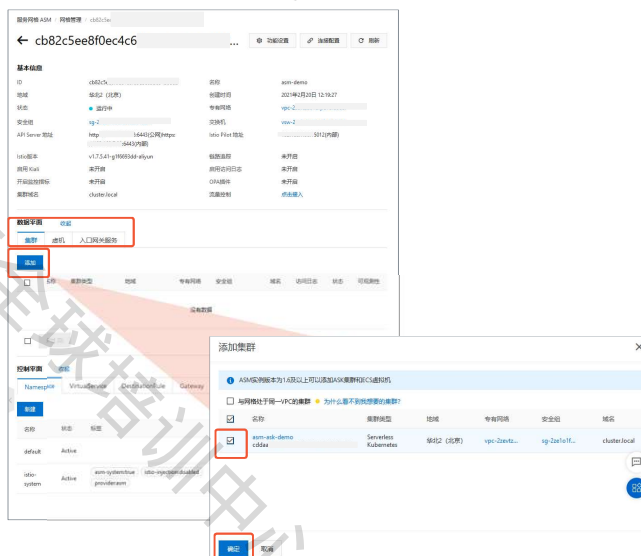
<#>

阿里云

阿里云服务网格使用 – 添加集群

• 操作步骤

1. 在网格管理页面单击目标ASM实例名称
 - 或目标ASM实例右侧操作列下的**管理**
2. 在数据平面区域，单击**集群**页签
3. 在集群页签，单击**添加**
4. 在弹出页**选**中需要添加的集群，并确定
5. 在重要提示对话框中单击确定



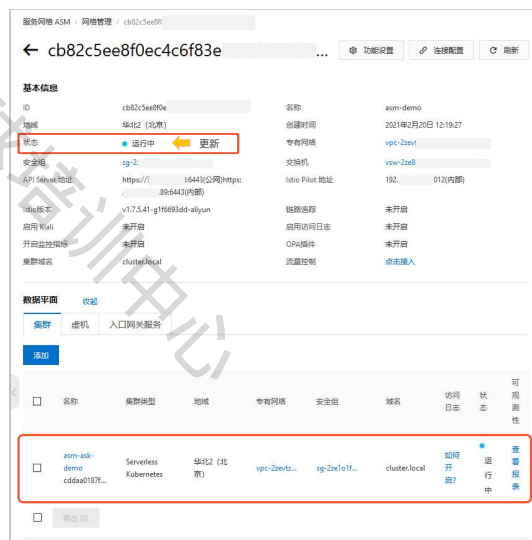
<#>

阿里云

阿里云服务网格使用 – 添加集群

• 执行结果

- 添加集群之后，ASM实例的状态变为更新中
 - 黄：更新中
 - 青：运行中
- 数秒之后，刷新信息，网格状态会变为运行中
 - 等待时长与添加的集群数量有关
- 在数据平面区域，可以查看已添加集群的信息



<#>

阿里云

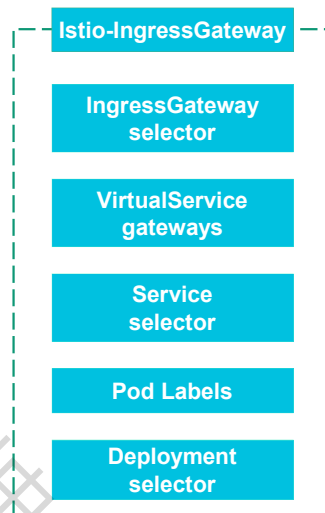
阿里云服务网格使用 – 添加入口网关

• 前提条件

- 已创建至少一个ASM实例
- 已添加至少一个集群到该实例中

• 背景信息

- 七层网关功能
- 对外提供一个统一的七层服务入口
- HTTP请求分发 / 负载均衡



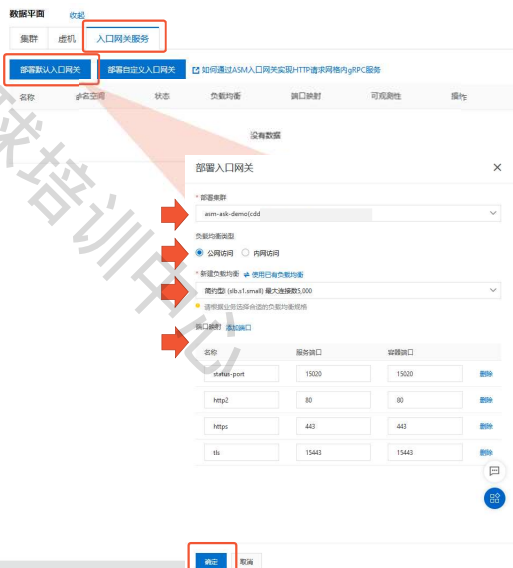
<#>

阿里云

阿里云服务网格使用 – 添加入口网关

• 操作步骤

1. 在网格管理页面，进入ASM实例管理页面
2. 在数据平面区域，单击**入口网关服务**页签
3. 在入口网关服务页签，单击**部署默认入口网关**
4. 在部署入口网关页面，为集群添加入口网关服务
 - 从部署集群列表中选择要部署入口网关服务的集群
 - 指定**负载均衡**的类型，公网访问或内网访问。
 - 选择**负载均衡**。
 - 使用已有负载均衡
 - 新建负载均衡
5. 配置**端口映射**
 - 单击添加端口
 - 在新增端口行中，输入服务端口和容器端口
6. 单击确定



阿里云

阿里云服务网格使用 – 添加入口网关

• 执行结果

- ASM 数据平面区域，**入口网关服务**页签
 - 表格中显示刚创建的 IngressGateway
- 登录**容器服务控制台**查看详情
 - 在集群列表页面中，单击目标集群名称或详情
 - 在集群管理页左侧导航栏中单击服务
 - 在服务页面，从命名空间下拉列表中选择istio-system
 - 查看入口网关服务和Pod的详细信息



阿里云

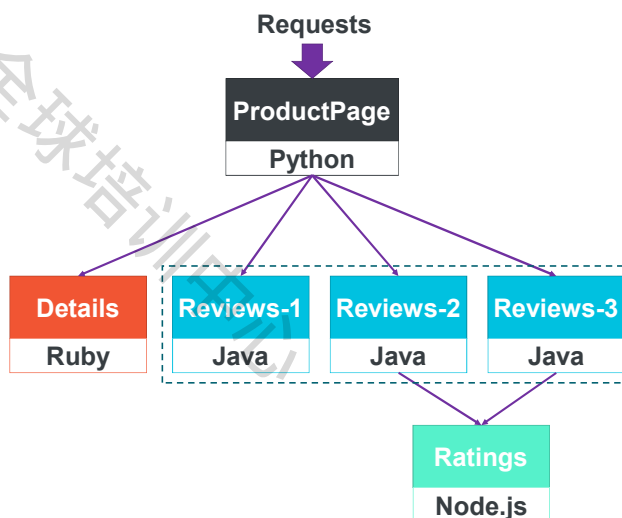
阿里云服务网格使用 – 部署应用

• 前提条件

- 已创建至少一个ASM实例
- 已添加至少一个集群到该实例中
- 通过kubectl连接到ASM实例中新添加的集群
- 已添加入口网关到该集群

• 背景信息

- 案例：Bookinfo书评应用
 - 4个微服务构成
 - 不同的技术栈
 - Reviews微服务有3个版本



<#>

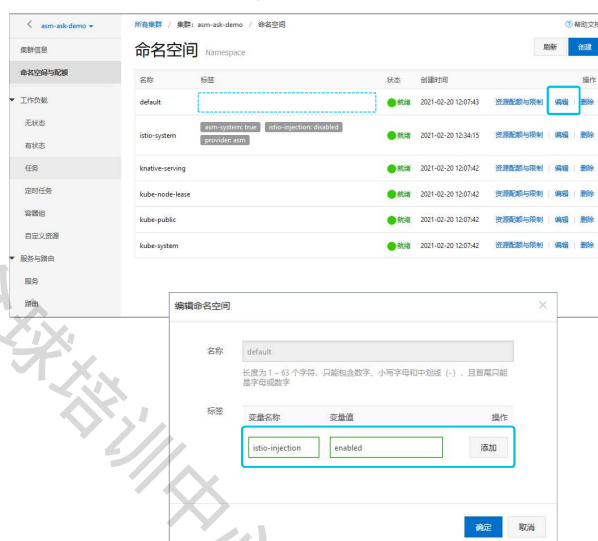
阿里云

阿里云服务网格使用 – 部署应用

• 操作步骤

1. 为命名空间配置Istio注入标签

- 登录容器服务控制台
- 在集群列表页面中，单击目标集群名称或详情
- 在集群管理页左侧导航栏中，单击命名空间
- 在命名空间页面，单击default命名空间右侧的编辑
- 在编辑命名空间对话框中，为该命名空间添加标签
 - 在变量名称文本框中输入istio-injection
 - 在变量值文本框中输入enabled
- 单击添加
- 或者：kubectl label namespace default istio-injection=enabled



<#>

阿里云

阿里云服务网格使用 – 部署应用

• 操作步骤

2. 部署Bookinfo应用

- 方法一：单击[通过CloudShell管理集群按钮](#)，开启控制台
 - 下载Bookinfo的YAML文件
`wget https://github.com/istio/istio/blob/master/samples/bookinfo/platform/kube/bookinfo.yaml`
 - 将Bookinfo应用部署到ASM实例的集群中
`kubectl apply -f bookinfo.yaml`
- 方法二：单击[使用YAML创建资源按钮](#)，开启控制台
 - 选择 default 命名空间
 - 将 bookinfo.yaml 的内容粘贴至模板编辑区
 - 单击创建按钮



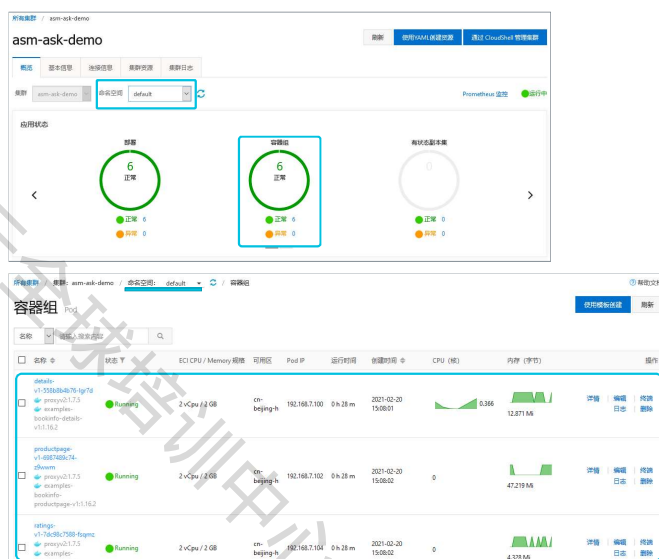
<#>



阿里云服务网格使用 – 部署应用

• 执行结果

- 查看Bookinfo应用的部署情况
 - 在控制台左侧导航栏中，单击集群
 - 在集群列表页面中，单击目标集群名称
 - 选择命名空间 default
 - 在应用状态图表中，单击容器组图表
 - 跳转至工作负载的容器组子页面
 - 页面中会显示所选命名空间中的Pod信息
 - 单击对应Pod操作列的详情，查看Pod的详细信息



<#>



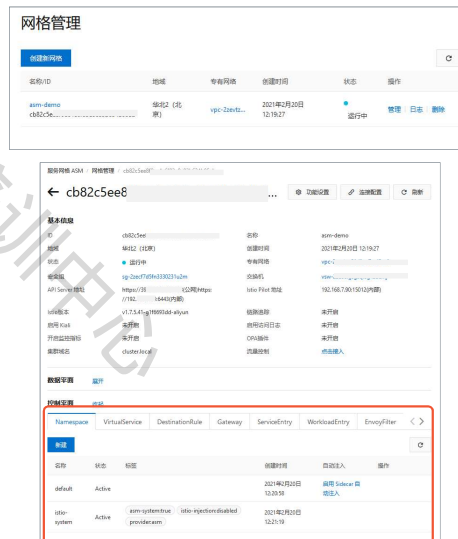
阿里云服务网格使用 – 定义资源

• 前提条件

- 已创建至少一个ASM实例
- 已添加至少一个集群到该实例中
- 已添加入口网关到该集群
- 已部署应用到该集群

• 背景信息

- 定义Istio资源的目的
 - 策略路由
 - 流量治理
 - 鉴权
 - 安全保护



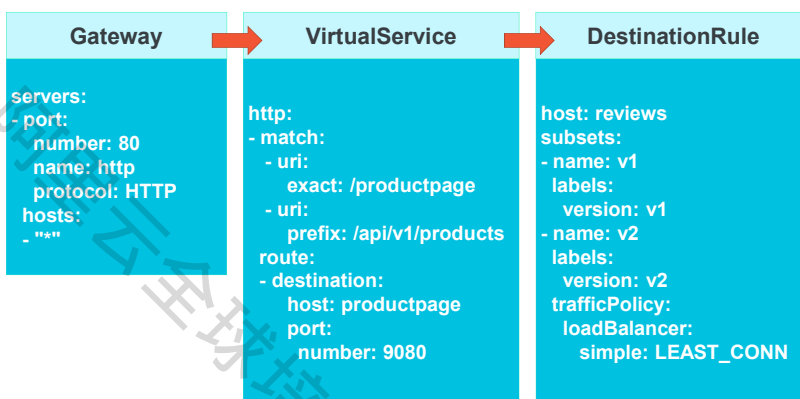
<#>



阿里云服务网格使用 – 定义资源

• 常用 Istio 资源类型

- 服务网关 Gateway
 - 暴露服务
 - 绑定端口、协议、域名
- 虚拟服务 VirtualService
 - URI匹配
 - 服务版本
 - 流量权重
- 目标规则 DestinationRule
 - 负载均衡策略
 - 流控规则
- 服务入口 ServiceEntry
 - 外部服务发现
 - 流量拦截



<#>



阿里云服务网格使用 – 定义资源

• 步骤一：定义Gateway资源

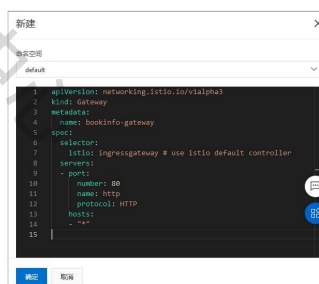
- 进入ASM实例的管理页面
 - 选择服务网格 -> 网格管理
 - 在网格管理页面，单击实例的名称或管理
- 在控制平面区域单击Gateway页签，单击新建
- 在新建页面中，按以下步骤定义服务网关，单击确定
 - 选择相应的命名空间
 - 在文本框中，定义服务网关



• 步骤结果

- 在服务网关页签可以看到新建的bookinfo-gateway网关

名称	命名空间	标签	创建时间	操作
bookinfo-gateway	default		2021年2月2日 15:03:11	详情 删除



<#>



阿里云服务网格使用 – 定义资源

• 步骤二：定义虚拟服务

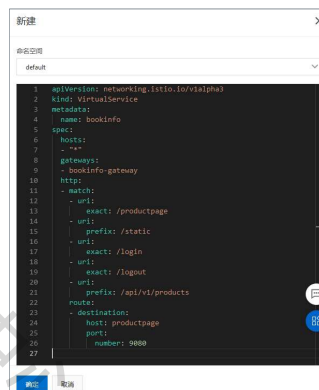
- 进入ASM实例的管理页面
 - 在控制平面区域，单击VirtualService页签，单击新建
- 在新建页面中，按以下步骤定义虚拟服务，单击确定
 - 选择相应的命名空间
 - 在文本框中，定义Istio虚拟服务



• 步骤结果

- 在虚拟服务页签可以看到新建的bookinfo服务

名称	命名空间	标签	创建时间	操作
bookinfo	default		2021年2月2日 15:04:01	详情 删除



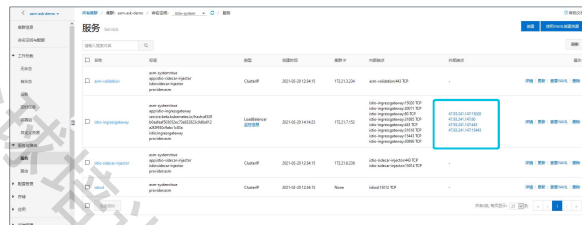
<#>



阿里云服务网格使用 – 定义资源

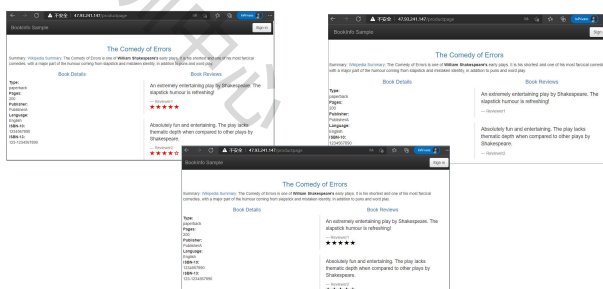
• 步骤三：访问入口网关

- 从ASM控制台查看入口网关的IP地址
- 从容器服务控制台查看入口网关的IP地址
 - 进入集群管理的服务页面
 - 在命名空间下拉列表中选择istio-system
 - 查看 istio-ingressgateway 对应的外部端点信息
- 在通过浏览器访问验证路由策略
 - `http://{入口网关IP地址}/productpage`



• 步骤结果

- 未定义Reviews的流量路由策略是轮询
- 可以依次看到3个版本



<#>



阿里云服务网格使用 – 定义资源

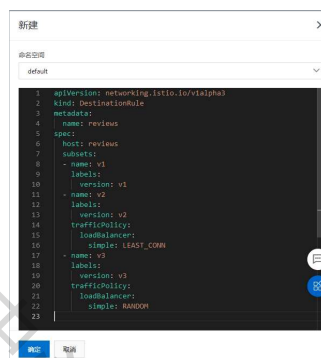
• 步骤四：定义目标规则

- 在控制平面区域，单击DestinationRule页签，单击新建
- 在新建页面中，按以下步骤定义目标规则，单击确定
 - 选择相应的命名空间
 - 在文本框中，定义目标规则
 - 负载均衡策略：轮询 / 最少连接数 / 随机



• 步骤结果

- 在目标规则页签可以看到新建的reviews目标规则



<#>



阿里云服务网格使用 – 定义资源

• 步骤五：增加新的虚拟服务

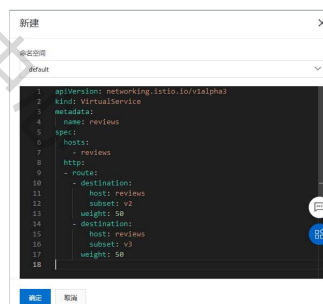
- 为Reviews微服务定义虚拟服务
- 在控制平面区域，单击VirtualService页签，单击新建
- 在新建页面中，按以下步骤定义虚拟服务，单击确定
 - 选择相应的命名空间
 - 在文本框中，定义虚拟服务



• 步骤结果

- 在虚拟服务页签可以看到新建的reviews虚拟服务

名称	命名空间	标签	创建时间	操作
bookinfo	default		2021年2月20日 16:04:01	查看 删除
reviews	default		2021年2月20日 16:11:10	查看 删除



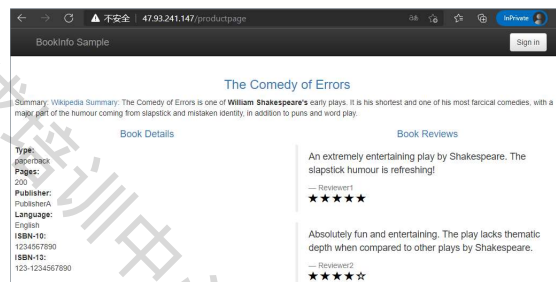
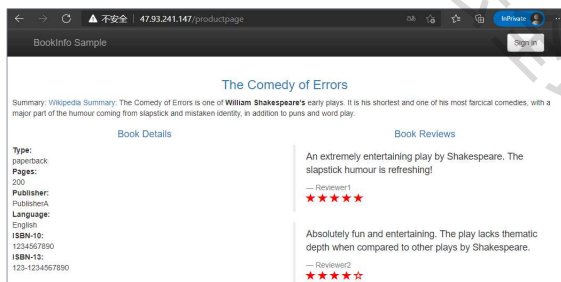
<#>



阿里云服务网格使用 – 定义资源

• 执行结果

- 查看Bookinfo应用的负载均衡效果
 - 在浏览器地址栏输入 http://{入口网关IP地址}/productpage
 - 持续刷新，可以看到Bookinfo应用页面在v2和v3之间等权重地轮流切换，显示为黑色星形图标或者红色星形图标



<#>



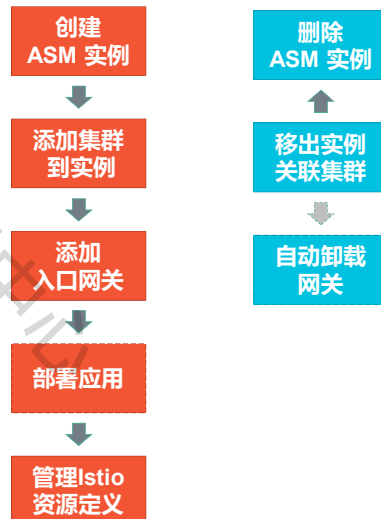
阿里云服务网格使用 – 流程总结

• 服务网格 ASM 的部署流程

- 创建一个**服务网格ASM实例**
- 添加一个**集群**到服务网格ASM 实例中
- 为服务网格 ASM 实例中的集群**部署入口网关**
- 将应用部署到上述服务网格 ASM 实例中的集群中
- 为服务网格ASM实例**定义虚拟服务和 Istio 网关**

• 服务网格 ASM 的移除流程

- 在ASM实例管理页面**移出**所有关联的集群
- 集群中部署的入口网关服务将被**自动卸载**
- 在列表页面**删除**ASM实例



<#>

阿里云服务网格使用 – 流程总结

• 不能在ASM实例中部署应用

- 控制台提供的配置用于连接ASM实例
- Istio资源在控制平面操作，无法在控制平面部署应用

• 不能在ASM管理的集群中部署Istio的虚拟服务等CRD资源

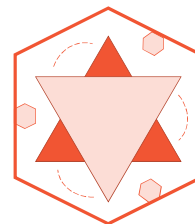
- Istio资源在控制平面操作，连接的kubernetes配置同样是控制平面提供的配置

• 不直接使用集群的连接配置操作Istio资源

- Istio CRD对应的资源保存在控制平面，控制平面完成规则转换并应用

• 服务网格控制平面的命名空间与数据平面集群的命名空间的区别

- 通过控制台或Kubernetes定义的命名空间隶属于ASM实例
- 与该ASM实例管理的数据平面集群相互独立
- 在实例和数据平面集群中的管理及维护操作互不影响



<#>

课程回顾

- 本节课的内容主要包括:

1. 服务网格的技术概要
2. 阿里云ASM的架构特点
3. 阿里云ASM的应用场景
4. 如何添加一个ASM实例
5. 如何关联ASM实例与K8s集群
6. 为集群配置入口网关
7. 常用Istio资源类型
8. 通过定义Istio资源实现流量转发策略

<#>

阿里云

阿里云

阿里云云原生容器ACP认证培训

容器镜像服务 (ACR) 概述



课程目标

学习完本课程后，你将能够：

1. 了解容器镜像存储中心
2. 了解阿里容器镜像产品功能
3. 了解ACR企业版和默认实例的区别
4. 了解ACR的产品优势及应用场景



课程目录

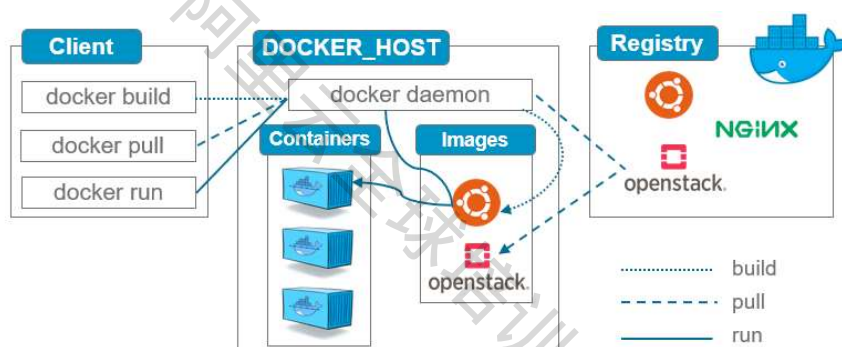
1. 容器镜像基本概念
2. 阿里容器镜像产品介绍
3. ACR 默认实例概述
4. ACR 企业版概述
5. ACR 企业版和默认实例版提供的规格
6. 产品优势及应用场景

<#>

阿里云

容器镜像基本概念

- 容器镜像基本概念



<#>

阿里云

课程目录

1. 容器镜像服务基本概念
- 2. 阿里容器镜像产品介绍**
3. ACR 默认实例概述
4. ACR 企业版概述
5. ACR 企业版和默认实例版提供的规格
6. 产品优势及应用场景

<#>



阿里容器镜像产品介绍

• 容器镜像服务（简称 ACR）

- 提供云原生资产的安全托管和全生命周期管理
- 支持多场景下镜像的高效分发，与容器服务 ACK 无缝集成，打造云原生应用一站式解决方案。



多类型资产托管

支持多架构容器镜像、Helm Chart
云原生资产全生命周期管理



多维度安全保障

支持镜像安全扫描及加签，支持操作审计，支持网络访问控制



提效应用交付

提供云原生应用交付链功能，全链路可观测、可追踪、可自主配置



云产品间无缝集成

与 ACK 无缝集成，输出云原生 DevOps、持续交付方案

<#>



阿里容器镜像产品介绍

- 与阿里云容器服务、CodePipeline深度集成，自动完成从代码提交到应用部署的DevOps完整流程



(中)

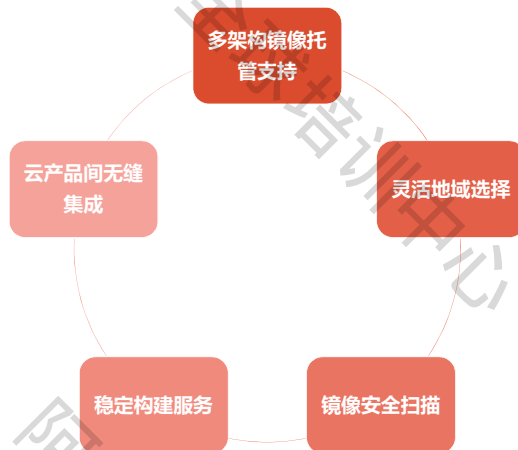
课程目录

1. 容器镜像服务基本概念
2. 阿里容器镜像产品介绍
- 3. ACR 默认实例概述**
4. ACR 企业版概述
5. ACR 企业版和默认实例版提供的规格
6. 产品优势及应用场景

(中)

ACR 默认实例概述

- 容器镜像服务ACR（Alibaba Cloud Container Registry）默认实例提供基础的容器镜像服务



<#>

课程目录

1. 容器镜像服务基本概念
2. 阿里容器镜像产品介绍
3. ACR 默认实例概述
- 4. ACR 企业版概述**
5. ACR 企业版和默认实例版提供的规格
6. 产品优势及应用场景

<#>

ACR企业版概述

• 产品功能



<#>

阿里云

ACR企业版概述

• 使用场景



<#>

阿里云

课程目录

1. 容器镜像服务基本概念
2. 阿里容器镜像产品介绍
3. ACR 默认实例概述
4. ACR 企业版概述
- 5. ACR 企业版和默认实例版提供的规格**
6. 产品优势及应用场景

<#>

阿里云

ACR企业版和默认实例版提供的规格

功能归类	细分项		默认实例版	企业版		
				基础版	标准版	高级版
制品管理	容器镜像	托管	支持	支持	支持	支持
		命名空间限额	3	15	25	50
		公开仓库限额	300	1000	3000	5000
		私有仓库限额				
		构建	支持	支持	支持	支持
		触发器	支持	支持	支持	支持
	Helm Chart	托管	-	支持	支持	支持
		命名空间限额	-	15	25	50
		公开仓库限额	-	1000	3000	5000

<#>

阿里云

ACR企业版和默认实例版提供的规格

功能归类	细分项		默认实例版	企业版		
				基础版	标准版	高级版
安全增强	容器镜像	漏洞扫描	支持	支持	支持	支持
		网络访问控制	-	支持	支持	支持
	Helm Chart	网络访问控制	-	支持	支持	支持
		操作审计	-	支持	支持	支持
应用分发	大规模分发		-	-	支持	支持
	全球同步		-	-	支持	支持
应用交付	云原生应用交付链		-	-	-	支持
周边服务	ACK集成	免密插件	支持	支持	支持	支持
	迁云指导	镜像极速导入	-	支持	支持	支持

<#>

阿里云

课程目录

1. 容器镜像服务基本概念
2. 阿里容器镜像产品介绍
3. ACR 默认实例概述
4. ACR 企业版概述
5. ACR 企业版和默认实例版提供的规格
6. 产品优势及应用场景

<#>

阿里云

ACR产品优势及应用场景

简单易用

- 无需自行搭建及运维，一键创建镜像仓库
- 支持多地域，提供稳定快速的镜像上传、下载服务

安全可控

- 完善的镜像权限管理体系，确保镜像的分享安全，团队的协作便利
- 提供镜像安全扫描功能，保证镜像漏洞可识别，漏洞级别可提示

无缝集成

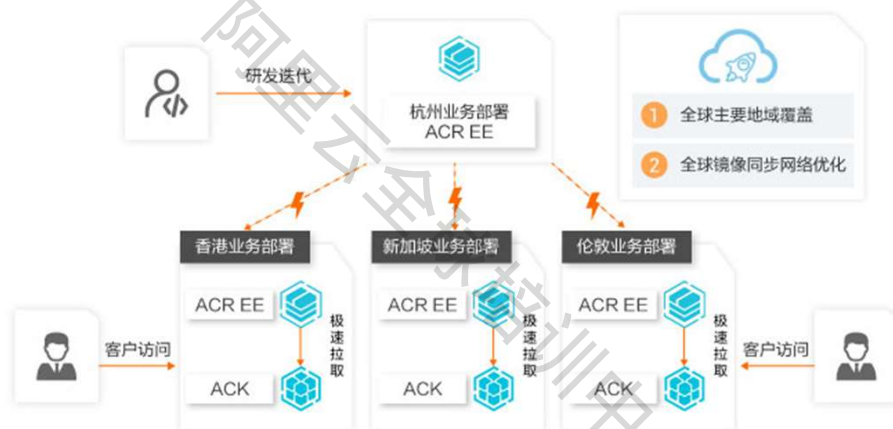
- 整合容器服务，实现新应用镜像生成后的持续部署
- 整合云 Code、CodePipeline，实现源代码到编译测试后，自动生成应用镜像

<#>

阿里云

ACR产品优势及应用场景

• 业务镜像全球化自动部署



<#>

阿里云

ACR产品优势及应用场景

• 容器化 DevSecOps 最佳实践



<#>

阿里云

课程小结

1. 了解阿里容器镜像产品功能
2. 分析ACR企业版和默认实例的区别
3. 了解ACR的产品优势及应用场景

<#>

阿里云



阿里云云原生容器ACP认证培训

容器镜像服务 (ACR) 使用与实践



课程目标

学习完本课程后，你将能够：

1. 了解如何创建企业版容器实例
2. 了解如何配置容器实例
3. 了解ACR镜像分发加速

课程目录

1. 创建企业版容器实例
2. 容器实例配置
3. ACR 镜像分发加速

创建企业版容器实例

前提条件

- 已开通容器镜像服务企业版所依赖的云产品OSS。
- 如需查询实例的实时数据，请开通云监控。
- 如需通过VPC访问实例，请开通VPC。
- 如需记录控制台操作，请开通ActionTrail。
- 已为容器镜像服务企业版授权您的OSS、VPC、云监控等资源权限。

<#>

阿里云

创建企业版容器实例

1. 登录容器镜像服务控制台。
2. 在左侧导航栏，选择**实例列表**。
3. 在**实例列表**页面，单击**创建企业版实例**。



<#>

阿里云

创建企业版容器实例

4. 在创建企业版实例页面，完成基本信息的配置，然后单击立即购买。

容器镜像服务企业版

2020年3月12日零时起，ACR EE 正式商业托管，限时优惠购买。

地域：中国 亚太 欧洲与澳洲 中东与印度

华北2（北京） 华北3（张家口） 华东1（杭州） 华东2（上海） 华南1（深圳） 中国（香港）

实例规格：基础版 标准版 高级版

【基础版】提供企业级容器镜像及 Helm Chart 托管和分发服务，支持安全扫描、网络访问控制、操作审计、镜像拉取 QPS 可达 250，详见规格说明

实例名称：acrdemo

实例名称必须不可修改，为企业级应用或默认前缀，长度为 3-30 个字符，可使用小写字母、数字、分划线“-”（不能为首尾字符）。

实例存储：默认 自定义

默认从指定账号下创建 OSS Bucket 托管容器镜像，可避免产生存储冗余公网流量费用，价格参考 OSS 定价

购买时长：1个月 2个月 3个月 6个月 1年 2年 更多时长

☐ 到期自动续费

参数	说明
地域	选择企业版实例所在的地域。
实例规格	三个规格：基础版、标准版、高级版
实例名称	输入实例名称。
实例存储	支持默认和自定义。
选择 Bucket	选择已有Bucket托管容器镜像。
购买时长	目前支持选择1、2、3、6个月和1~5年。

课程目录

1. 创建企业版容器实例
2. 容器实例配置
 - 2.1 绑定源代码托管平台
 - 2.2 创建镜像仓库
 - 2.3 设置构建规则
 - 2.4 构建镜像
3. ACR镜像分发加速

容器实例配置1

绑定源代码托管平台

- 在实例管理页面选择**仓库管理** > **代码源**。



代码平台	版本	绑定方式	触发构建限制
阿里云Code	全部	Code OAuth授权	不支持提交代码自动触发镜像构建
阿里云Codeup	全部	私人访问令牌	无
GitHub	个人版	GitHub OAuth授权	无
GitLab	全部	私人访问令牌	无
Bitbucket	全部	Bitbucket OAuth授权	无

<#>



容器实例配置2

创建镜像仓库

- 在**镜像仓库**页面单击**创建镜像仓库**。
- 在创建镜像仓库对话框中，设置**命名空间**、**仓库名称**、**仓库类型**和**摘要**，然后单击**下一步**。



<#>



容器实例配置3

3. 在设置代码源对话框中，选择之前绑定的代码源，然后单击**创建镜像仓库**。

创建镜像仓库

仓库信息

代码源

云Code GitHub Bitbucket 私有GitLab 本地仓库

构建设置

☒ 代码变更自动构建镜像 ☐ 海外机器构建 ☐ 不使用缓存

构建规则设置

请在仓库创建完成后前往构建页面设置

上一步 **创建镜像仓库** 取消

<#>

阿里云

容器实例配置4

设置构建规则

1. 在**镜像仓库**页面单击目标仓库右侧操作列中的**管理**。
2. 单击左侧导航栏中的**构建**，在**构建规则**设置区域的左侧单击**添加规则**。
3. 在**添加构建规则**对话框中设置构建规则，然后单击**确认**。

添加构建规则

* 类型 Branch

* Branch/Tag main

* Dockerfile 目录 /

* Dockerfile 文件名 Dockerfile

* 镜像版本 latest

☒ 镜像版本 latest

☐ 时间

☐ yyyyMMdd-HH ☒ yyyyMMdd-H:mm

支持英文字母、数字、"-","_",".","/","\$","@"

最近一次推送代码的 Commit ID，默认取前 6 位

0 40

确定 取消

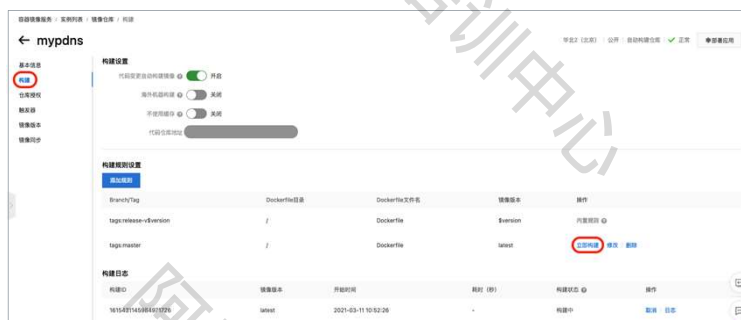
<#>

阿里云

容器实例配置6

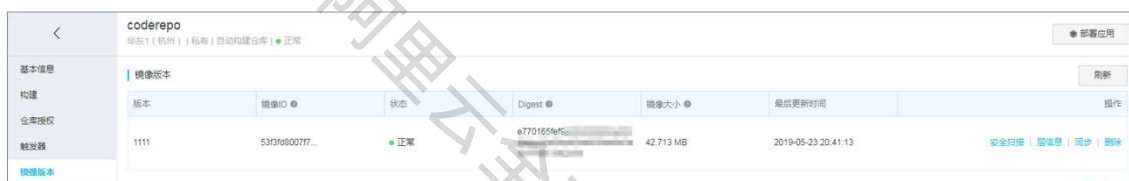
构建镜像

1. 在**镜像仓库**页面，单击目标仓库右侧操作列的**管理**。
2. 在镜像仓库详情页面，单击左侧导航栏中的**构建**。
3. 在**构建规则设置**区域，单击目标规则操作列中的**立即构建**。



容器实例配置7

4. 等待镜像构建完成后，单击左侧菜单栏中的**镜像版本**，查看已构建完成的镜像列表。



课程目录

1. 创建企业版容器实例
2. 容器实例配置
3. ACR镜像分发加速
 - 3.1 官方镜像加速
 - 3.2 P2P加速

<#>



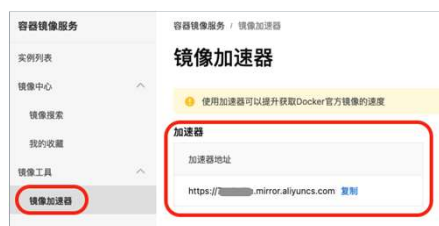
ACR镜像分发加速

官方镜像加速

1. 登录容器镜像服务控制台。
2. 在左侧导航栏选择**镜像工具** > **镜像加速器**。
3. 在**镜像加速器**页面就会显示为您独立分配的加速器地址。
4. 使用配置文件/etc/docker/daemon.json进行docker配置。

```
{
  "registry-mirrors": ["[系统分配前缀].mirror.aliyuncs.com"]
}
```

5. 重启Docker Daemon。



<#>



ACR镜像分发加速

P2P加速

当大规模容器集群批量下载镜像时，P2P加速功能可以提升镜像拉取速度，减少应用部署时间。

- 当集群规模达到300节点以上时，使用效果会更好
- 推荐使用多可用区VSwitch打散集群ECS节点
- 推荐使用本地SSD或者可分配内存较大的ECS实例
- 当集群规模较小或可分配内存不足时，可能不会起到加速的效果

课程小结

通过本节的学习，您可以了解

1. 如何创建容器实例
2. 配置容器实例的流程
3. ACR镜像分发加速方式和优点

阿里云

阿里云云原生容器ACP认证培训

容器镜像服务 (ACR) 进阶

阿里云

课程目标

学习完本课程后，你将能够：

1. 了解什么是DevOps
2. 了解如何通过容器镜像实现DevOps
3. 了解如何搭建容器DevOps
4. 了解如何使用免密组件拉取容器镜像

<#>

课程目录

1. DevOps相关概念

1.1 DevOps

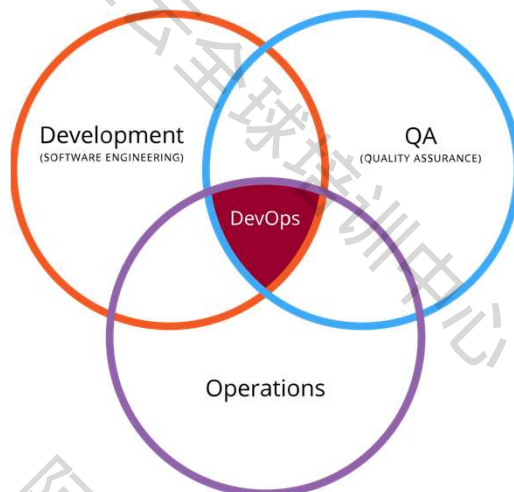
1.2 Continuous Delivery

2. 通过容器镜像实现DevOps
3. 构建容器DevOps
4. 使用免密组件拉取容器镜像

<#>

DevOps相关概念

DevOps

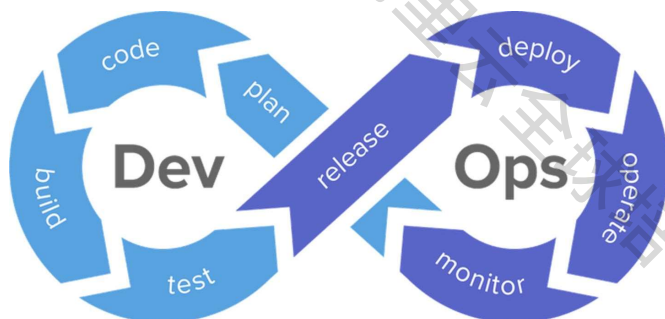


<#>

阿里云

DevOps相关概念

DevOps (Development和Operations的组合词) 是一种重视 “软件开发人员 (Dev) ” 和 “IT 运维技术人员 (Ops) ” 之间沟通合作的文化、运动或惯例。透过自动化 “软件交付” 和 “架构变更” 的流程，来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。



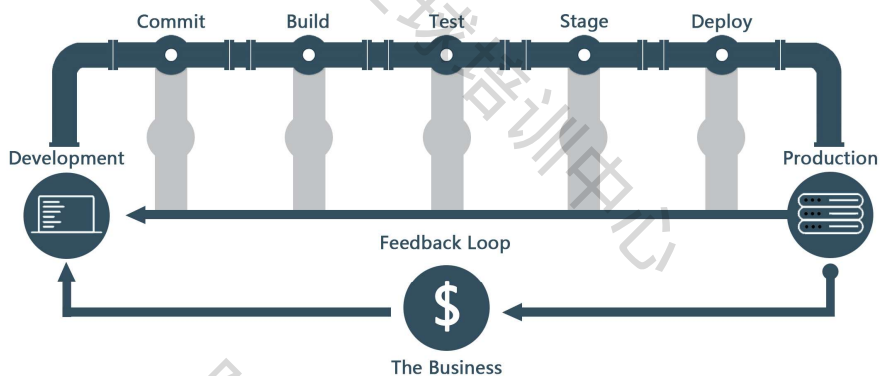
- 技术与流程自动化
- 组织管理和流程
- 软件质量和监控
- 企业文化

<#>

阿里云

DevOps相关概念

Continuous Delivery



<#>

阿里云

课程目录

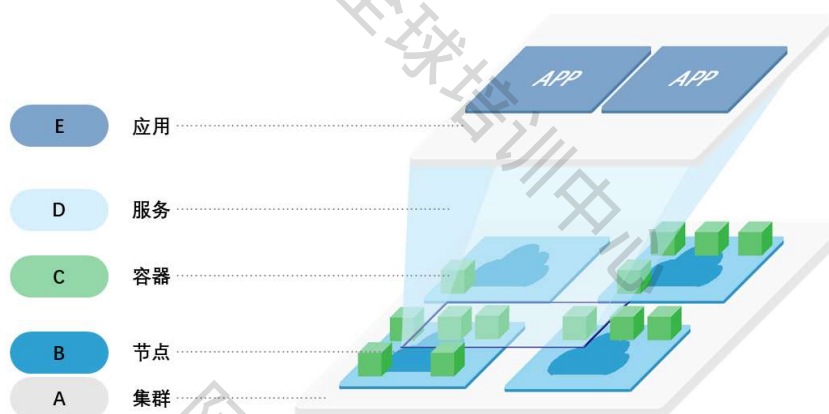
1. DevOps基本概念
2. 通过容器镜像实现DevOps
 - 2.1 阿里云容器服务
 - 2.2 完整的容器化持续交付流程
 - 2.3 Jenkins
3. 构建容器DevOps
4. 使用免密组件拉取容器镜像

<#>

阿里云

通过容器镜像实现DevOps

阿里云容器服务

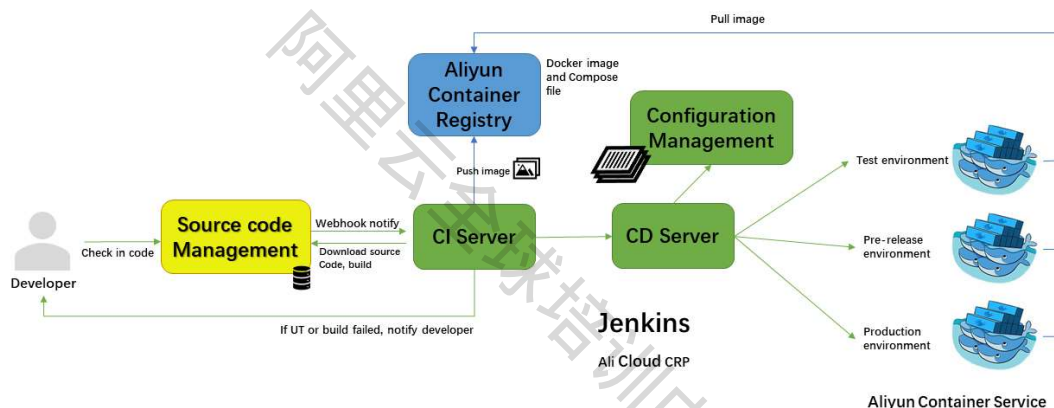


<#>

阿里云

通过容器镜像实现DevOps

完整的容器化持续交付流程

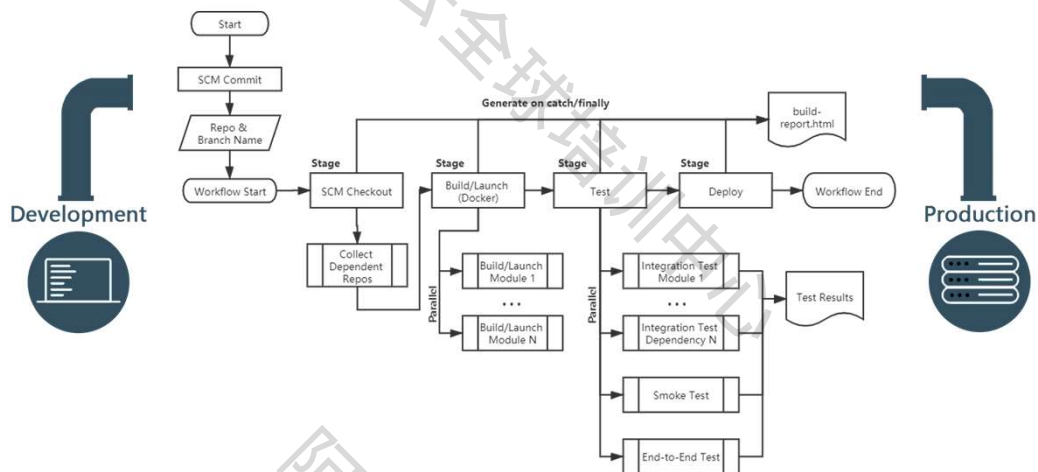


<#>

阿里云

通过容器镜像实现DevOps

Jenkins

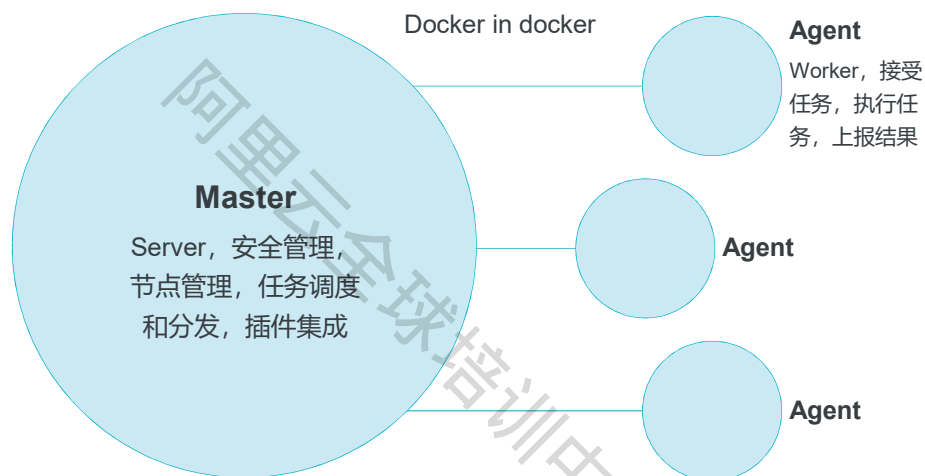


<#>

阿里云

通过容器镜像实现DevOps

Jenkins



<#>

阿里云

课程目录

1. DevOps基本概念
2. 通过容器镜像实现DevOps
3. 构建容器DevOps
 - 3.1 创建镜像仓库
 - 3.2 设置构建规则
 - 3.3 构建镜像仓库
 - 3.4 绑定容器服务触发器
4. 使用免密组件拉取容器镜像

<#>



构建容器DevOps

创建镜像仓库

1. 在镜像仓库页面单击创建镜像仓库。
2. 在创建镜像仓库对话框中，设置命名空间、仓库名称、摘要和仓库类型，本例选择私有镜像仓库类型。然后单击下一步。

创建镜像仓库对话框，包含以下信息：

- 步骤指示器：1. 仓库信息, 2. 代码源
- 地域：华东1（杭州）
- 命名空间：coderepo
- 仓库名称：coderepo
- 仓库类型：公开 ☐ 私有 ☒
- 摘要：1
- 描述信息：支持Markdown格式
- 底部按钮：下一步, 取消

<#>



构建容器DevOps

3. 在设置代码源对话框中，选择已绑定的代码源，然后单击**创建镜像仓库**。



The dialog box titled "创建镜像仓库" (Create Image Repository) shows the "代码源" (Code Source) tab selected. Under "代码源", "云Code" is chosen. Below it, a user selection dropdown shows "ZhangShiying". The "构建设置" (Build Settings) section has "代码变更自动构建镜像" (Automatically build image on code change) checked. The "构建规则设置" (Build Rule Settings) section has a note: "请在仓库创建完成后前往构建页面设置" (Please go to the build page to set up after repository creation). At the bottom right, the "创建镜像仓库" button is highlighted with a red box.

<#>



构建容器DevOps

设置构建规则

1. 在**镜像仓库**页面单击目标仓库右侧操作列中的**管理**。
2. 单击左侧导航栏中的**构建**，在**构建规则设置**区域的左侧单击**添加规则**。
3. 在**添加构建规则**对话框中设置构建规则，然后单击**确认**。



The dialog box titled "添加构建规则" (Add Build Rule) shows the following settings:

- * 类型: Branch
- * Branch/Tag: main
- * Dockerfile目录: /
- * Dockerfile文件名: Dockerfile
- * 镜像版本: latest
- * 镜像版本: ☒ 镜像版本
- * 时间: ☐ 时间
- * Commit ID: ☐ Commit ID

At the bottom right, the "确定" (Confirm) button is highlighted with a blue box.

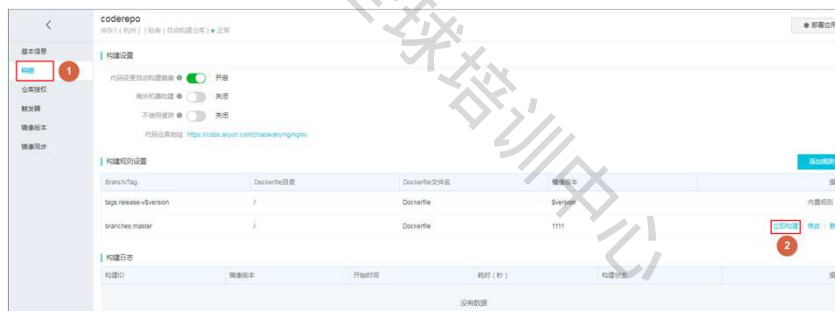
<#>



构建容器DevOps

构建镜像仓库

1. 手工构建：在构建规则设置区域，单击目标规则操作列中的立即构建。



2. 自动构建：当您在绑定的代码源的指定分支中修改并提交代码后，将触发仓库的第二条构建规则进行自动构建。

<#>

阿里云

构建容器DevOps

绑定容器服务触发器

- 镜像信息页面，单击左侧操作栏中的触发器，新建一个触发器，并填入触发器的名称、URL、触发方式。



<#>

阿里云

课程目录

1. DevOps基本概念
2. 通过容器镜像实现DevOps
3. 构建容器DevOps
4. 使用免密组件拉取容器镜像
 - 4.1 背景信息
 - 4.2 升级组件并对组件进行配置
 - 4.3 配置当前账号拉取权限

<#>



使用免密组件拉取容器镜像

背景信息

免密组件通过读取ACK集群内的kube-system命名空间中的acr-configuration的配置，进行私有镜像拉取。当前支持针对私有镜像仓库使用以下三种权限之一的策略进行配置：

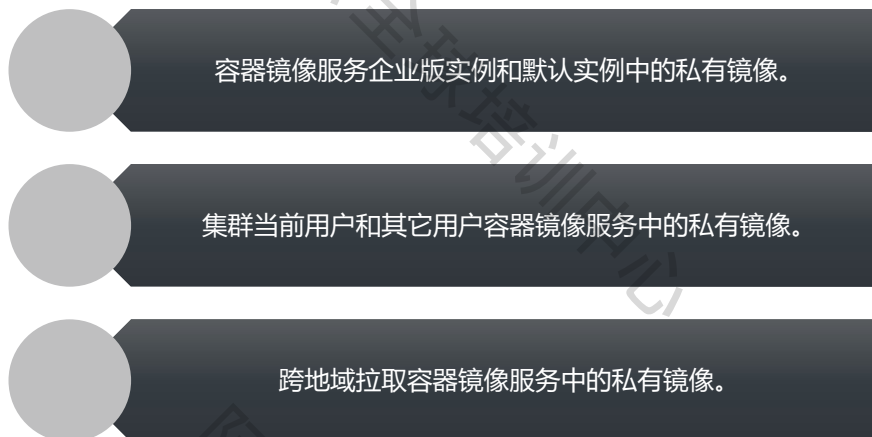
- 使用默认的ACK Worker RAM角色权限进行拉取（默认策略）
- 使用自定义RAM角色的AccessKey ID和AccessKey Secret的权限进行拉取
- 通过配置RAM AssumeRole权限，使用其他用户的权限进行拉取。

<#>



使用免密组件拉取容器镜像

镜像

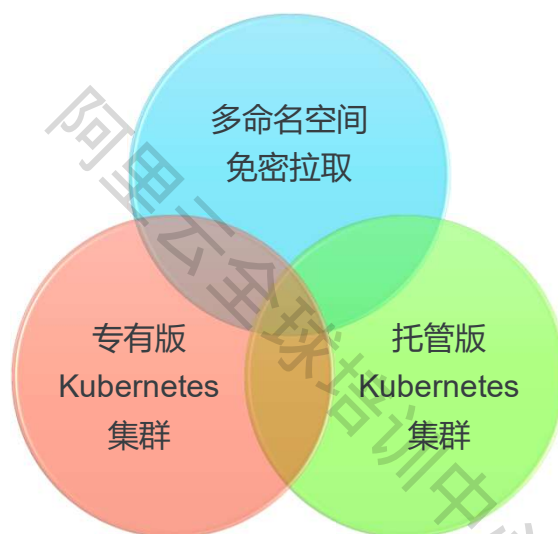


<#>

阿里云

使用免密组件拉取容器镜像

集群



<#>

阿里云

使用免密组件拉取容器镜像

升级组件并对组件进行配置

1. 升级组件

- I. 在**集群列表**页面，单击目标集群操作列下的**更多** > **系统组件管理**。
- II. 在组件列表区域中找到**aliyun-acr-credential-helper**，单击**升级**。



<#>

阿里云

使用免密组件拉取容器镜像

升级组件并对组件进行配置

2. 配置组件

- I. 在**集群信息**页面左侧导航栏，选择**应用配置管理** > **配置项**。
- II. 在**配置项**页面，选择命名空间：**kube-system**，单击**创建**添加新配置项，或单击目标配置项右侧**编辑**。



<#>

阿里云

使用免密组件拉取容器镜像

企业版实例

```
data:
  acr-api-version: "2018-12-01"
  acr-registry-info: |-
    - instanceId: "cri-xxx"
      regionId: "cn-hangzhou"
  expiring-threshold: 15m
  service-account: default
  watch-namespace: all
```

默认实例

```
data:
  acr-api-version: "2018-12-01"
  acr-registry-info: |-
    - instanceId: ""
      regionId: "cn-hangzhou"
  expiring-threshold: 15m
  service-account: default
  watch-namespace: all
```

配置项	配置项说明	默认值
service-account	使免密组件作用于指定的服务账号	Default
acr-registry-info	容器镜像的实例信息数组，YAML多行字符串格式，每个实例以三元组方式配置	空，表示免密拉取本地地域的默认容器镜像实例仓库。
watch-namespace	期望能免密拉取镜像的Namespace	Default
expiring-threshold	本地Cache Token过期阈值	15 m

<#>



使用免密组件拉取容器镜像

配置当前账号拉取权限

- 在**集群信息**页面，单击**集群资源**页签，单击Worker RAM角色右侧链接。
- 在RAM角色基本信息的**权限管理**页签，单击目标权限策略名称。

The screenshot displays the Alibaba Cloud console interface for managing RAM roles. On the left, the 'ack-demo' cluster is shown with the 'Worker RAM 角色' highlighted. On the right, the 'KubernetesWorkerRole-d9c1f3ea-6b59-473c-a276-00b0f14c307d' role is selected, and the '权限管理' (Permissions) tab is active, showing the 'K8sWorkerRolePolicy-d9c1f3ea-6b59-473c-a276-00b0f14c307d' policy.

<#>



使用免密组件拉取容器镜像

配置当前账号拉取权限

3. 单击修改策略内容。
4. 在策略内容区域增加以下字段后，单击确定。



课程小结

通过本节的学习，您可以：

1. 了解DevOps和Continuous Delivery的概念
2. 了解通过容器镜像实现DevOps的原理
3. 了解搭建容器DevOps的流程
4. 了解如何使用免密组件拉取容器镜像

阿里云

阿里云云原生容器ACP认证培训
边缘容器服务 (ACK@Edge)

阿里云

课程目录

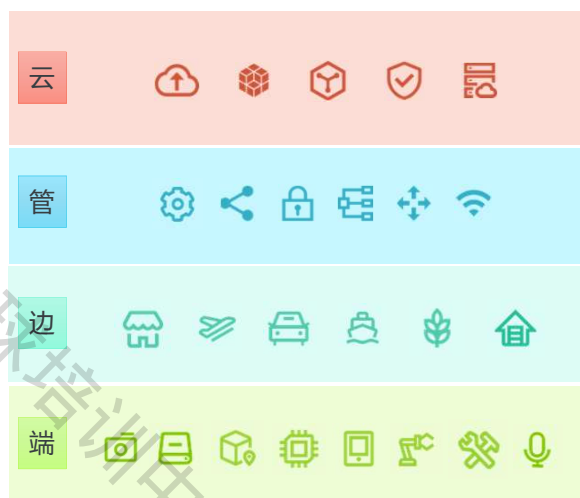
1. 边缘计算和边缘容器
 - 1.1 边缘计算概要
 - 1.2 边缘计算应用
 - 1.3 边缘计算与云原生
2. 阿里云边缘容器服务

<#>

阿里云

边缘计算 – 概要

- 核心概念
 - 分布式架构
 - 靠近物或数据源头的一侧
 - 就近提供服务
 - 计算、存储、网络一体式平台
- 技术特性
 - 邻近性、低时延、高带宽
 - 分布式、位置认知、数据入口
- 服务特点
 - 本地化、自组织
 - 可定义、可调度
 - 标准开放、安全



<#>

阿里云

边缘计算 – 应用

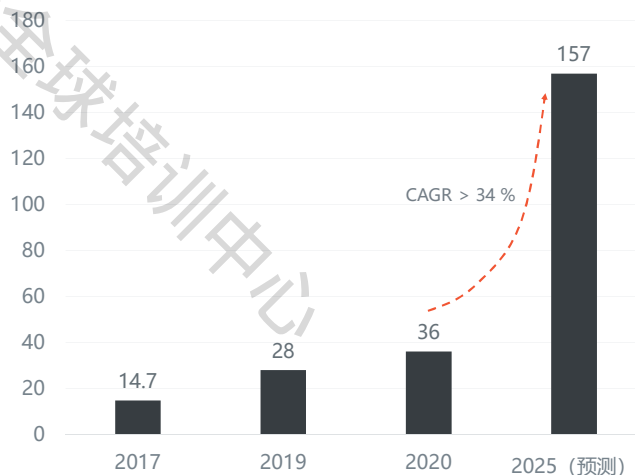
• 应用场景

- 技术场景:
 - 5G、物联网、工业互联网
 - 人工智能、AR/VR、云边协同
- 业务场景:
 - 智能制造
 - 智慧城市
 - 互动直播
 - 车联网

• 发展趋势

- 超过500亿的终端设备联网
- 超过40%数据在边缘侧计算
- 超过90%的企业在边缘部署应用
- 市场规模预计到2025年超157亿美元

世界边缘计算市场规模（亿美元）



<#>

阿里云

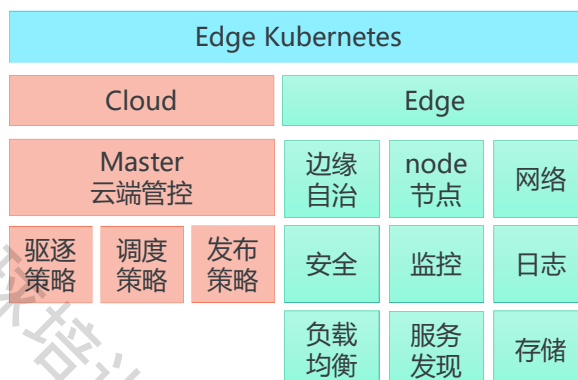
边缘计算 – 与云原生的融合

• 边缘计算 + 云原生

- 以Kubernetes为容器应用编排的标准
- 在任何基础设施上提供和云上一致的功能和体验
- 云-边-端一体化的应用分发
- 大规模应用交付、运维、管控的诉求
- 安全的工作负载运行环境
- 边缘服务治理
- 网络自恢复能力和兼容性
- 异构资源融合

• 边缘容器

- 原生 Kubernetes 容器服务能力
- Serverless边缘容器服务
- 混合云场景



<#>

阿里云

课程目录

1. 边缘计算和边缘容器
2. 阿里云边缘容器服务
 - 2.1 边缘容器服务产品简介
 - 2.2 边缘容器服务技术框架
 - 2.3 边缘容器服务功能优势
 - 2.4 边缘容器服务适用场景
 - 2.5 边缘容器服务实践概要

<#>



阿里云边缘容器服务 – 产品简介

• 阿里云边缘容器 ACK@Edge

- 依托ACK托管服务构建
- 通用的边缘容器云原生基础设施
- 非侵入式设计原则
- 原生与插件化组合方式
- 云边端一体化的云原生边缘容器产品
- 向上作为底座支撑边缘计算领域PaaS构建
- 向下支持自有节点等边缘算力资源接入
- 支持边缘自治、边缘安全容器、边缘智能
- 云端AI
- 流计算
- 边缘下沉的通道和平台
- 拓宽云产品边界

云原生边缘基础设施

- Edge K8s
- Edge Node

云边一体化

- 云产品向边缘下沉
- 拓宽云产品边界
- 推动边缘生态多样化

云 + 边

- 安全、稳定、标准

平台功能扩展

- 阿里云深度融合的组件
- 边缘智能
- 边缘Serverless

云 + 边 + 端

- 轻量、低耗
- IoT通信、异构适配

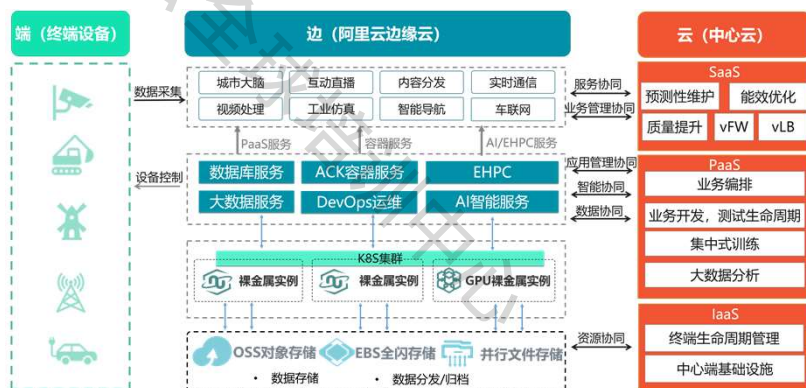
<#>



阿里云边缘容器服务 – 产品简介

• 边缘托管Kubernetes集群

- 阿里云托管
- 标准、安全、高可用
- 简化集群运维工作
- 非侵入方式增强
- 原生运维API支持
- 集群统一资源调度
- 异构边缘节点资源
- 边缘计算弱网络连接场景
- 反向运维网络通道



<#>

阿里云

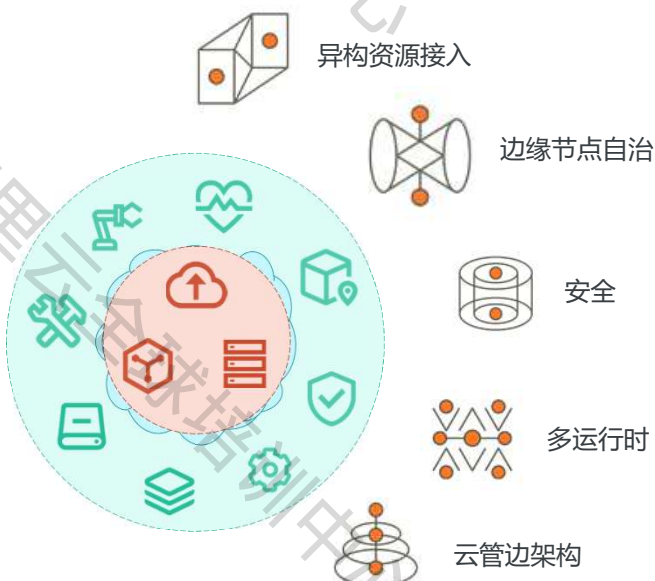
阿里云边缘容器服务 – 功能优势

• 产品功能

- 边缘Kubernetes集群和应用管理
- 多类型、异构边缘资源纳管
- EdgeUnit管理
- 边缘节点自治

• 产品优势

- 边缘异构资源接入
- 边缘节点自治
- 安全
- 多运行时
- 云管边架构



<#>

阿里云

阿里云边缘容器服务 – 适用场景：ENS节点服务

• 场景需求

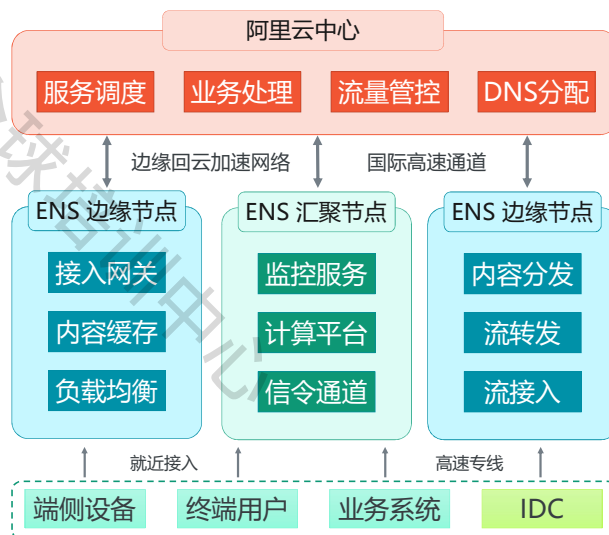
- 边缘节点间网络互通
- 异构资源共享
- 成本合理

• 方案扩展

- 边缘节点服务 ENS
 - 基于运营商边缘节点和网络构建
 - 用户业务下沉至运营商侧边缘
 - 降低计算时延和成本

• 方案优势

- 全域覆盖
- 高性价比
- 融合开发
- 安全可靠



<#>

阿里云

阿里云边缘容器服务 – 适用场景：ENS应用场景



CDN 缓存、调度、分发



互动直播 合流、切片、弹幕



在线教育 实时链路优选



视频监控 城市大脑、新零售



本地IT上云 仓储、高校、企业

更多场景尝试

SD-WAN
探测监测
弱网优化

<#>

阿里云

阿里云边缘容器服务 – 适用场景：IoT

• 场景架构：边缘智能一体机

- 完整的边缘应用
- 资源的生命周期管理
- 安全运行支撑
- 容灾容错保障
- 远程运维监控与报警等能力
- 云端AI能力服务下沉通道



阿里云边缘容器服务 – 适用场景：CDN

• 场景需求

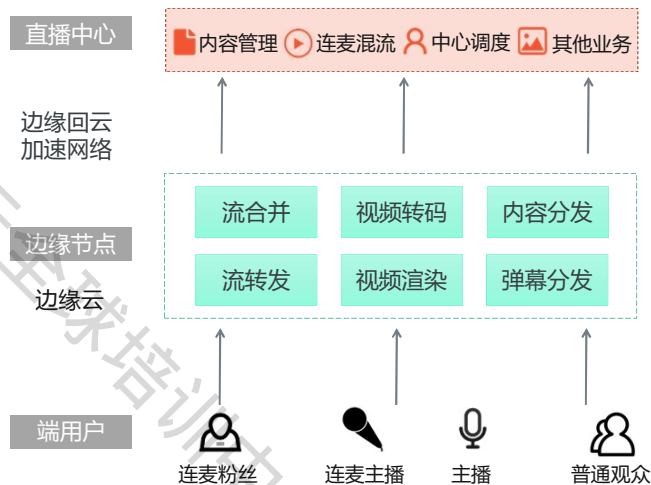
- 网络环境
 - 低延时
 - 高性能
 - 高稳定

• 用户痛点

- 自建多地IDC机房成本高
- 上中心云网络性能不满足需求
- 带宽价格昂贵

• 解决方案

- 提供低成本优质上云算力和网络



阿里云边缘容器服务 - 适用场景：CDN

- 场景架构：CDN资源统一管理
 - 海量边缘基础设施节点管理
 - CDN业务全生命周期管理能力
 - 云原生API助力构建CDN领域PaaS



<#>

阿里云

阿里云

阿里云云原生容器ACP认证培训

阿里云容器集群规划与构建案例



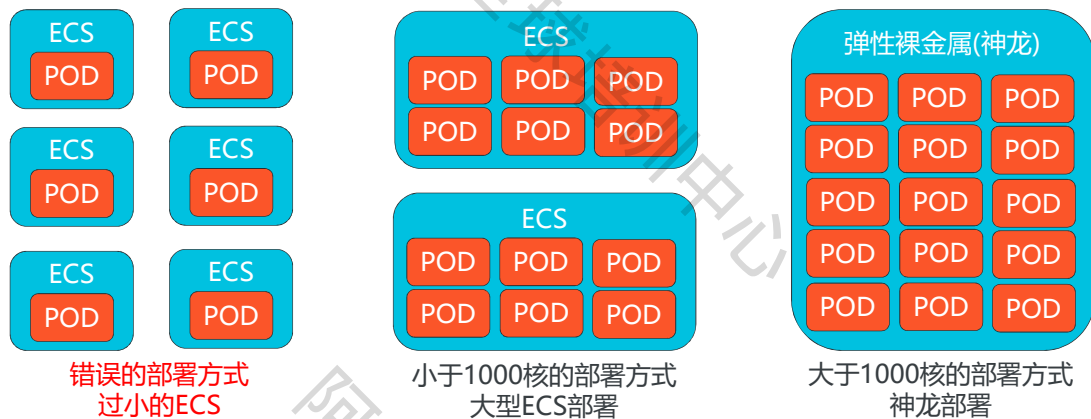
课程目录

- 1.1 集群构建规划
- 1.2 集群核心组件配置及优化
- 1.3 聚石塔如何应对上云挑战



集群构建规划——容器部署中的ECS选型

- 在创建Kubernetes集群时，物理部署实体的选型对上层应用的性能起着非常大的影响。通过大量客户在云容器部署实践的统计，很大一部应用在容器部署中往往没有采用正确的部署选型

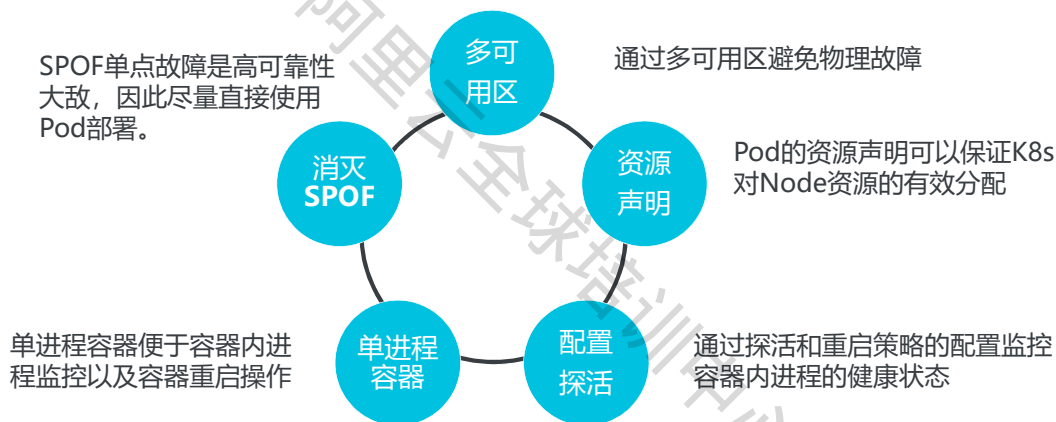


<#>

阿里云

集群构建规划——容器集群的高可靠配置

- 在容器集群部署规划中，除了ECS选型之外，还有很多配置方式会直接影响集群的可靠性。可以总结为如下5点



<#>

阿里云

集群构建规划——容器节点操作系统选择

- Alibaba Cloud Linux 2是新一代阿里云原生Linux操作系统，为云上应用程序提供安全、稳定、高性能的定制化运行环境，并针对云基础设施进行了深度优化，并提供更好的运行时体验。

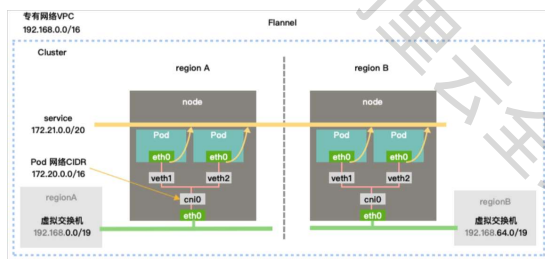


<#>

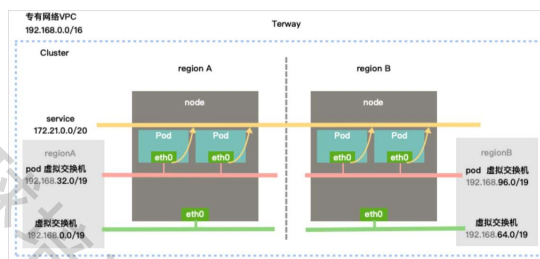
阿里云

集群构建规划——Kubernetes集群网络规划

- 在创建K8s集群时，根据网络插件的不同，需要指定专有网络VPC、虚拟交换机、Pod网络CIDR(地址段)和服务CIDR(地址段)。并做好ECS地址、K8s Pod地址和服务地址的规划工作。



Terway配置示意图



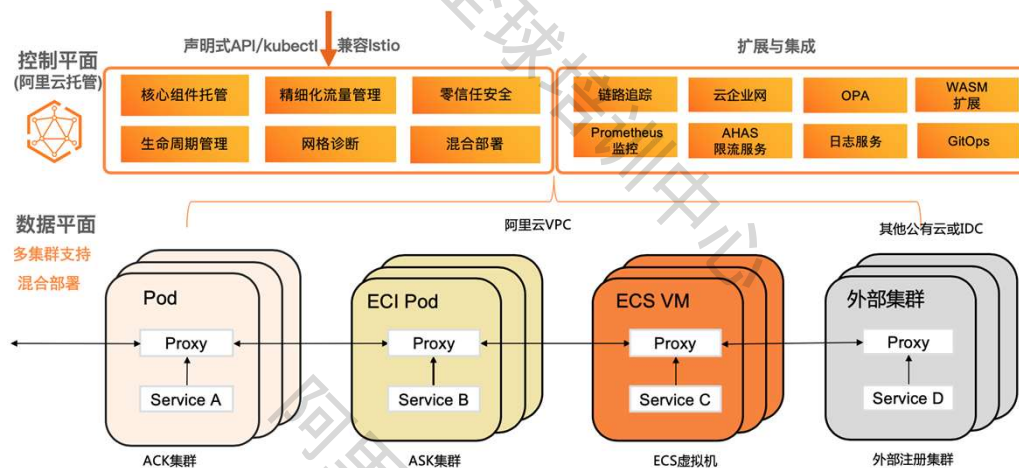
Flannel配置示意图

<#>

阿里云

集群构建规划——集群接入服务网格ASM

- 将部署容器集群中的应用接入阿里云服务网格ASM，可以简化服务的网格化治理，实现服务的可观测性，减轻开发与运维的工作负担。



课程目录

- 1.1 集群构建规划
- 1.2 集群核心组件配置及优化
- 1.3 聚石塔如何应对上云挑战

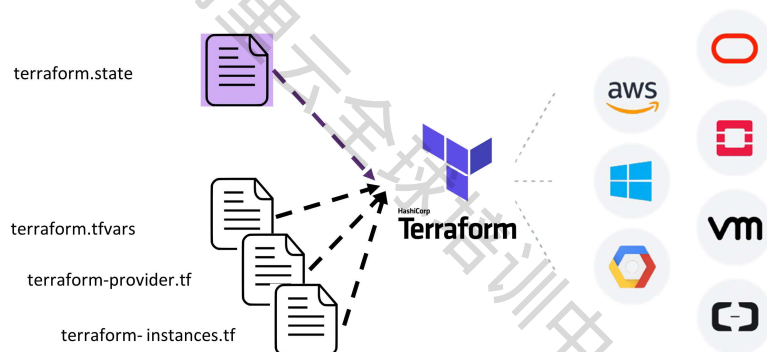
混合云部署Kubernetes集群

- 对于企业客户而言，出于数据主权和隐私安全的考虑，大多会考虑使用多云混合云方式开展业务，以K8s为代表的技术屏蔽了基础设施的差异性，推动了以应用多云混合云新架构的到来。



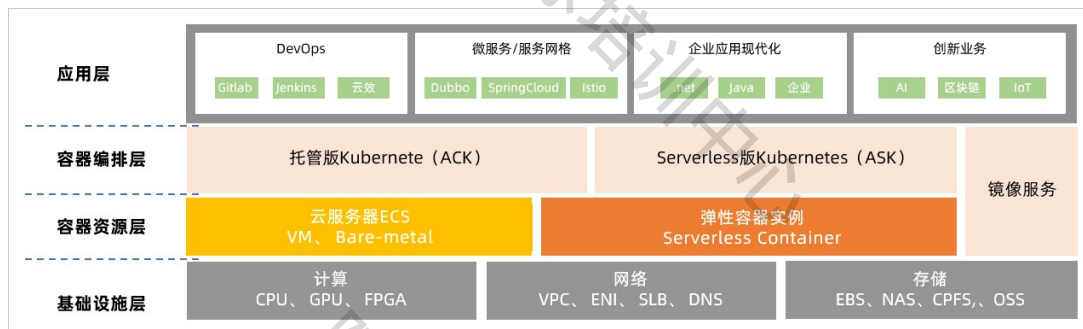
使用Terraform创建Kubernetes集群

- Terraform是HashiCorp公司提供的一种开源工具，用于安全高效地预览，配置和管理云基础设施架构和资源，帮助开发者自动化地创建、更新阿里云基础设施资源，并进行版本管理。



Kubernetes对接ECI

- ECI为K8s提供一种层次化的解决方案：即ECI负责底层Pod容器资源的调度和管理工作，K8s在ECI之上作为PaaS层来管理Deployment、Service、StatefulSet、CronJob等业务负载。

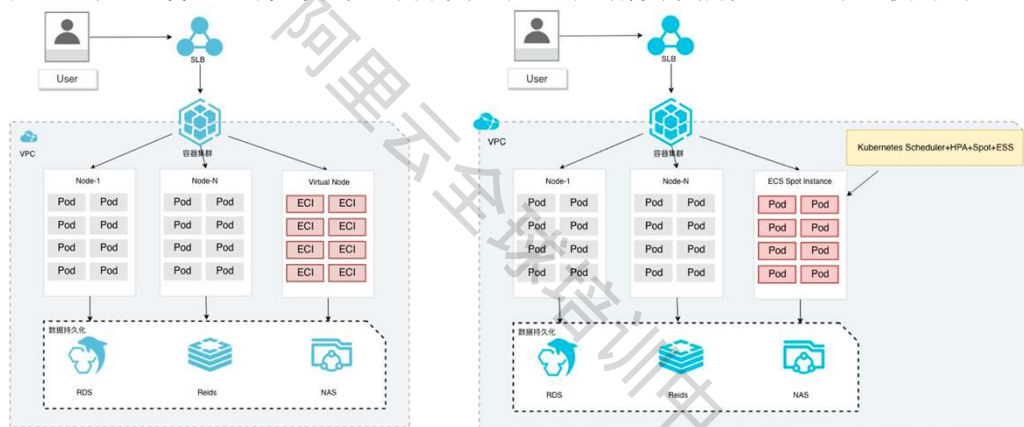


<#>

阿里云

容器集群的高弹性架构实现

- 对于有波峰波谷的在线业务。或者机器学习、基因测序等对实时性要求不高，但是对成本敏感的应用。可以通过容器高弹性架构，全自动实现动态创建或释放实例，达到成本的最优控制。



ECI方式高弹性架构

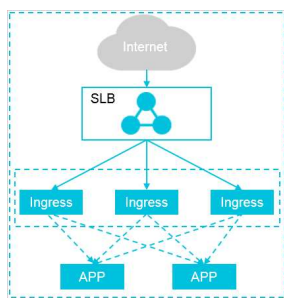
抢占式实例方式高弹性架构

<#>

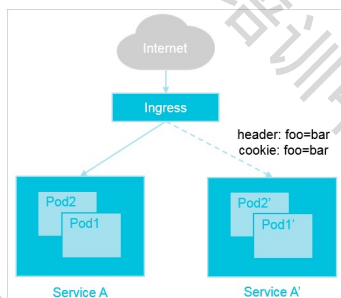
阿里云

Ingress最佳实践

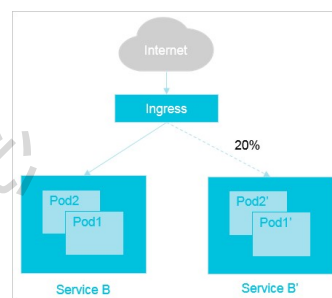
- 在K8s集群中，Ingress是授权入站连接到达集群服务的规则集合，可以提供七层负载均衡能力，通过Ingress可以实现滚动升级、分批暂停发布、蓝绿发布以及灰度发布等发布方式。以及不同集群之间的流量复制。作为集群流量接入层，Ingress的高可靠性显得尤为重要



Ingress高可靠架构



基于客户端请求的流量切分

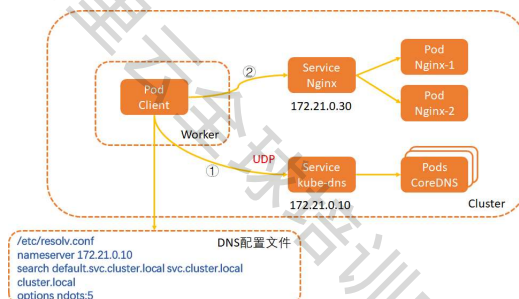


基于服务权重的流量切分

<#>

服务发现的最佳实践

- 容器集群中部署了CoreDNS作为内部DNS服务器，在集群中推荐使用Service名作为服务的访问地址，通过CoreDNS服务可以实现从Service名到Service的IP地址的解析。同时Serverless集群中支持PrivateZone进行域名解析服务发现



CoreDNS解析实现服务发现

<#>

课程目录

- 1.1 集群构建规划
- 1.2 集群核心组件配置及优化
- 1.3 聚石塔如何应对上云挑战

聚石塔业务背景与介绍

- 聚石塔最早上线于 2012 年，是阿里集团帮助商家打造的开放电商云工作平台。它汇聚包括淘宝、天猫、阿里云等整个平台资源和资源优势。同时依托于聚石塔，各种业务类型的服务商商家提供了如ERP、CRM、WMS 等业务服务。



聚石塔的业务痛点

- 聚石塔的业务大致可以分为两个部分：1. 电商订单服务链路上的系统如 ERP，CRM，WMS。2. 电商行业中直接面向客户的小程序场景，如手淘与千牛的小程序。其业务痛点如下：

标准化和稳定性问题

突发流量下的弹性问题

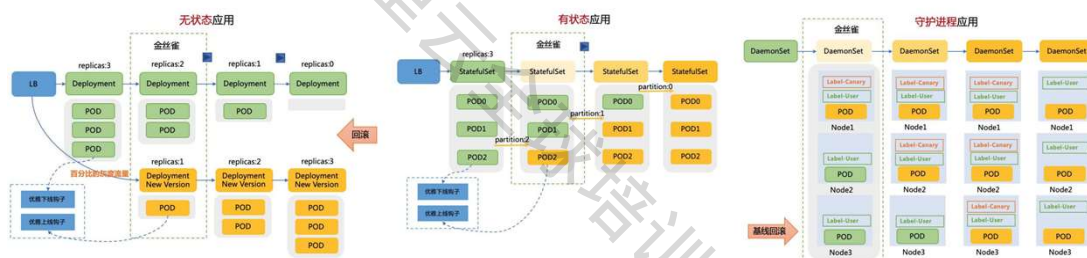
效率和成本的问题

<#>



聚石塔技术特点——应用和发布标准化

- K8s 自带的滚动发布，虽然标准化了发布环节，但在发布期间无法暂停，即使服务起来了，可能功能和业务存在问题，最终也会随着滚动不断扩大影响面；对此，聚石塔设计了支持暂停的分批发布，通过提前批的“金丝雀”验证，从而提升系统的稳定性。

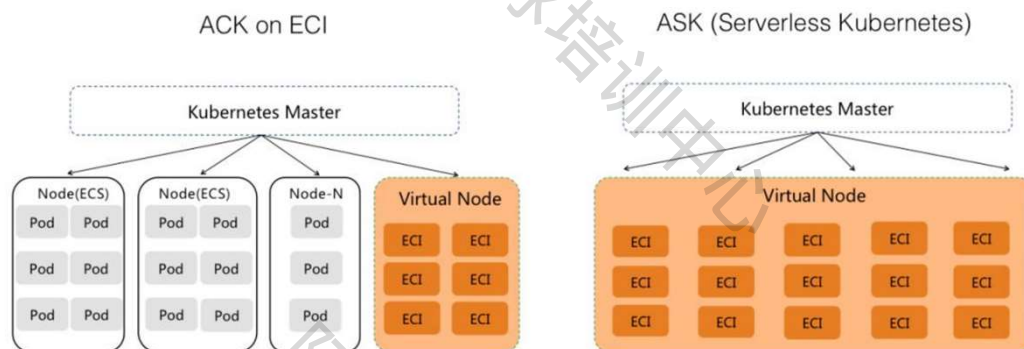


<#>



聚石塔技术特点——丰富的弹性扩容

- 对于电商等流量波动较大的场景，用户一般会把集群资源维持在日常流量的规模，并在流量高峰在来临前进行集群资源的扩容。ECI 的价格与同等规格 ECS 相近，且为按秒付费。秒级扩容。

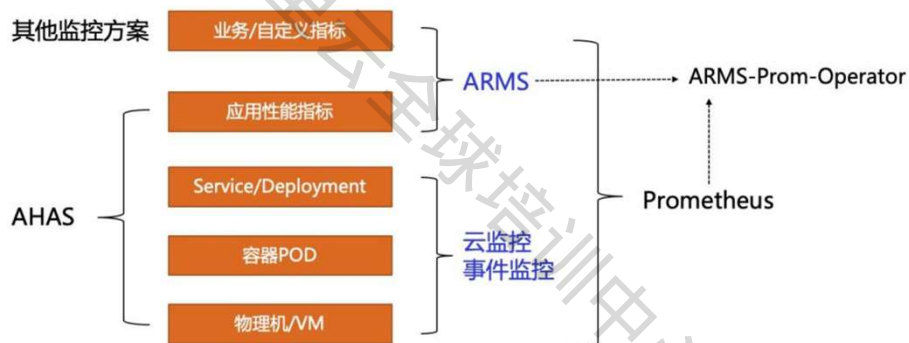


<#>

阿里云

聚石塔技术特点——应用监控

- 与传统应用相比，基于 K8s 容器实例是动态调度的、生命周期短，Deployment、Service 等容器上层抽象组件更是难以监控的，此外还需要底层 ECS 计算资源、实例生命周期、K8s 集群自身以及集群核心组件的各个维度的监控。

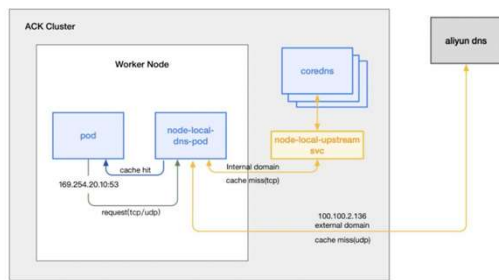


<#>

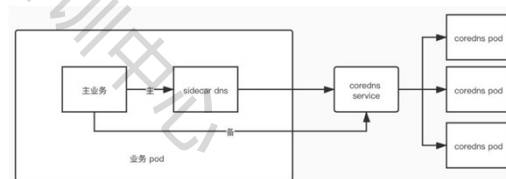
阿里云

聚石塔技术特点——DNS 生产环境优化

- 聚石塔的用户大多是电商或小程序的场景，开发语言多样化，有些语言没有很好的连接池，导致DNS解析频繁。K8s默认的CoreDNS在高并发时会遇到性能瓶颈，因此聚石塔对DNS的性能做了深入的优化，优化的手段主要有两种，分别是Node Local DNS或Sidecar DNS。



Node Local DNS架构图



Sidecar DNS架构图

<#>