

阿里云 容器服务Kubernetes版

产品简介

文档版本：20190911

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[]或者[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }或者{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 什么是容器服务 ACK.....	1
2 产品优势.....	3
3 产品架构.....	5
4 应用场景.....	6
5 基本概念.....	12
5.1 基本概念.....	12
5.2 与原生Kubernetes名词对照.....	14
5.3 容器服务 Kubernetes 集群三种形态对比.....	16
6 使用限制.....	18
7 Kubernetes版本支持机制.....	20

1 什么是容器服务 ACK

容器服务 ACK 提供高性能可伸缩的容器应用管理服务，支持企业级 Kubernetes 容器化应用的生命周期管理。容器服务 ACK 简化集群的搭建和扩容等运维工作，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。容器服务是全球首批通过 Kubernetes 一致性认证的服务平台，可以为您提供专业的容器支持和服务。

容器服务 ACK 包含了专有版 Kubernetes（Dedicated Kubernetes）、托管版 Kubernetes（Managed Kubernetes）、Serverless Kubernetes 三种形态，方便您按需选择。

- 专有版 Kubernetes：需要创建3个 Master（高可用）节点及若干 Worker 节点，可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。
- 托管版 Kubernetes：只需创建 Worker 节点，Master 节点由容器服务创建并托管。具备简单、低成本、高可用、无需运维管理 Kubernetes 集群 Master 节点的特点，您可以更多关注业务本身。
- Serverless Kubernetes：无需创建和管理 Master 节点及 Worker 节点，即可通过控制台或者命令配置容器实例的资源、指定应用容器镜像以及对外服务的方式，直接启动应用程序。

产品功能

- 集群管理
 - 通过控制台一键创建专有版 Kubernetes 集群、托管版 Kubernetes 集群、Serverless Kubernetes 集群，支持 GPU 实例和裸金属服务器，支持创建跨可用区高可用的集群。
 - 提供容器优化的 OS 镜像，提供经过稳定测试和安全加固的 Kubernetes 集群和 Docker 版本。
 - 支持多集群管理，支持跨可用区高可用集群，支持集群联邦管理。

- 一站式容器管理

- 网络

提供高性能 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流控限制。

- 存储

支持阿里云云盘、文件存储 NAS、对象存储 OSS，提供标准的 FlexVolume 驱动。

支持存储卷的动态创建和迁移。

- 日志

支持日志采集及将采集的日志集成到日志服务。

支持和第三方开源日志解决方案集成。

- 监控

支持容器级别和 VM 级别的监控。您还可以和第三方开源监控解决方案进行集成。

- 权限

支持集群级别的 RAM 授权管理。

支持应用级别的权限配置管理。

- 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。

- 高可用调度策略，轻松打通上下游交付流程

- 支持服务级别的亲和性策略和横向扩展。

- 支持跨可用区高可用和灾难恢复。

- 支持集群和应用管理的 OpenAPI，轻松对接持续集成和私有部署系统。

2 产品优势

容器服务 Kubernetes 版的优势

使用便捷

- 通过 Web 界面一键创建 Kubernetes 集群。
- 通过 Web 界面一键完成 Kubernetes 集群的升级。

您在使用自建 Kubernetes 集群的过程中，可能需要同时处理多个版本的集群（包括 1.8.6、1.9.4、以及 1.10 及之后的版本）。每次升级集群的过程都是一次大的调整和巨大的运维负担。容器服务的升级方案使用镜像滚动升级以及完整元数据的备份策略，允许您方便地回滚到先前版本。

- 通过 Web 界面轻松地实现 Kubernetes 集群的扩容和缩容。

使用容器服务 Kubernetes 集群可以方便地一键垂直扩缩容来快速应对数据分析业务的峰值。

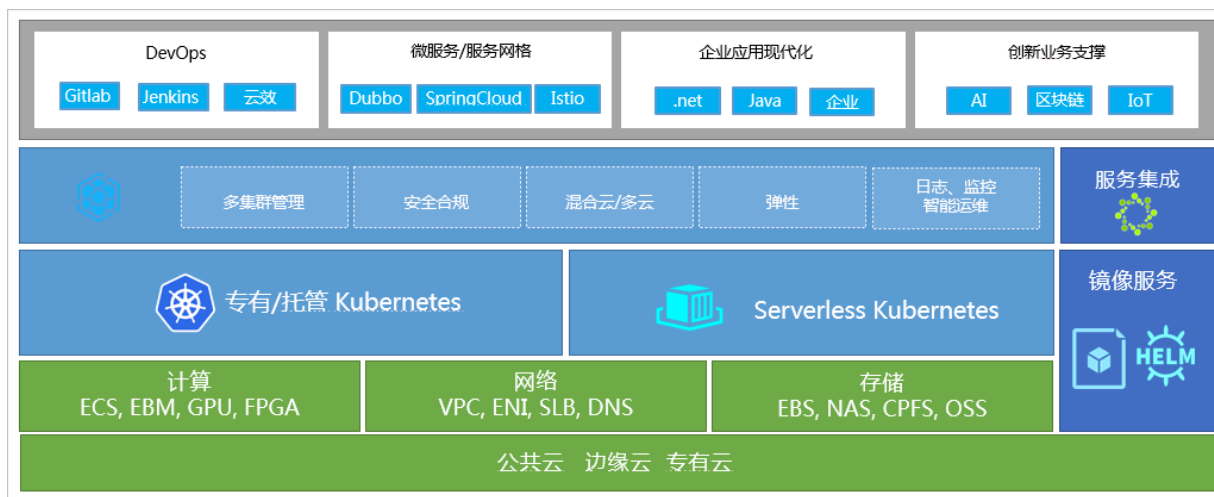
功能强大

功能	说明
网络	· 高性能 VPC 网络插件。 · 支持 network policy 和流控。 容器服务可以为您提供持续的网络集成和最佳的网络优化。
负载均衡	支持创建负载均衡实例（公网、内网）。 如果您在使用自建 Kubernetes 集群的过程中，使用自建的 Ingress 实现，业务发布频繁可能会造成 Ingress 的配置压力并增加出错概率。容器服务的 SLB 方案支持原生的阿里云高可用负载均衡，可以自动完成网络配置的修改和更新。该方案经历了大量用户长时间的使用，稳定性和可靠性大大超过用户自建的入口实现。
存储	集成阿里云云盘、文件存储NAS、块存储，提供标准的 FlexVolume 驱动。 自建 Kubernetes 集群无法使用云上的存储资源，阿里云容器服务提供了最佳的无缝集成。

功能	说明
运维	· 集成阿里云日志服务、云监控 · 支持弹性伸缩
镜像仓库	· 高可用，支持大并发 · 支持镜像加速 · 支持 p2p 分发 您如果使用自建的镜像仓库，在百万级的客户端同时拉取镜像的时候，会存在镜像仓库崩溃的可能性。使用容器服务镜像仓库的公共云版本来提高镜像仓库的可靠性，减少运维负担和升级压力。
稳定	· 专门的团队保障容器的稳定性。 · 每个 Linux 版本，每个 Kubernetes 版本都会在经过严格测试之后才会提供给用户。 容器服务提供了 Docker CE 兜底和推动 Docker 修复的能力。当您遇到 Docker Engine 挂起、网络问题、内核兼容等问题时，容器服务可以为您提供最佳实践。
高可用	· 提供多可用区支持。 · 支持备份和容灾。
技术支持	· 提供 Kubernetes 升级能力，新版本一键升级。 · 阿里云容器团队负责解决在环境中遇到的各种容器问题。

3 产品架构

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。



产品特性	特性说明
多Kubernetes集群形态	融合阿里云虚拟化技术，容器服务Kubernetes版支持Dedicated、Managed和Serverless三种集群形式。 · Dedicated：可使用ECS、EGS、神龙服务器作为集群节点，实例规格灵活配置，支持丰富的插件。 · Managed：在Dedicated的基础上，新增托管Master节点的功能，具备简单、低成本、高可用、无需运维的特点。 · Serverless：提供免服务器管理的Kubernetes serverless服务，简化底层资源管理，灵活扩容，降低资源开销。
阿里云Kubernetes集群管控服务	支持强大的网络、存储、混合集群管理、水平扩容、应用扩展等特性。
阿里云Kubernetes管理服务	支持安全镜像和Helm工具，与阿里云RAM、KMS、日志、监控等产品高度集成，提供一个安全合规的Kubernetes解决方案，为您提供混合云、容器安全、CI/CD、DevOps等增强能力。
便捷、高效的使用方式	容器服务Kubernetes版提供Web Console 和API&SDK。

4 应用场景

DevOps 持续交付

最优化的持续交付流程

配合 Jenkins 帮您自动完成从代码提交到应用部署的 DevOps 完整流程，确保只有通过自动测试的代码才能交付和部署，高效替代业内部署复杂、迭代缓慢的传统方式。

能够实现：

- DevOps 自动化

实现从代码变更到代码构建，镜像构建和应用部署的全流程自动化。

- 环境一致性

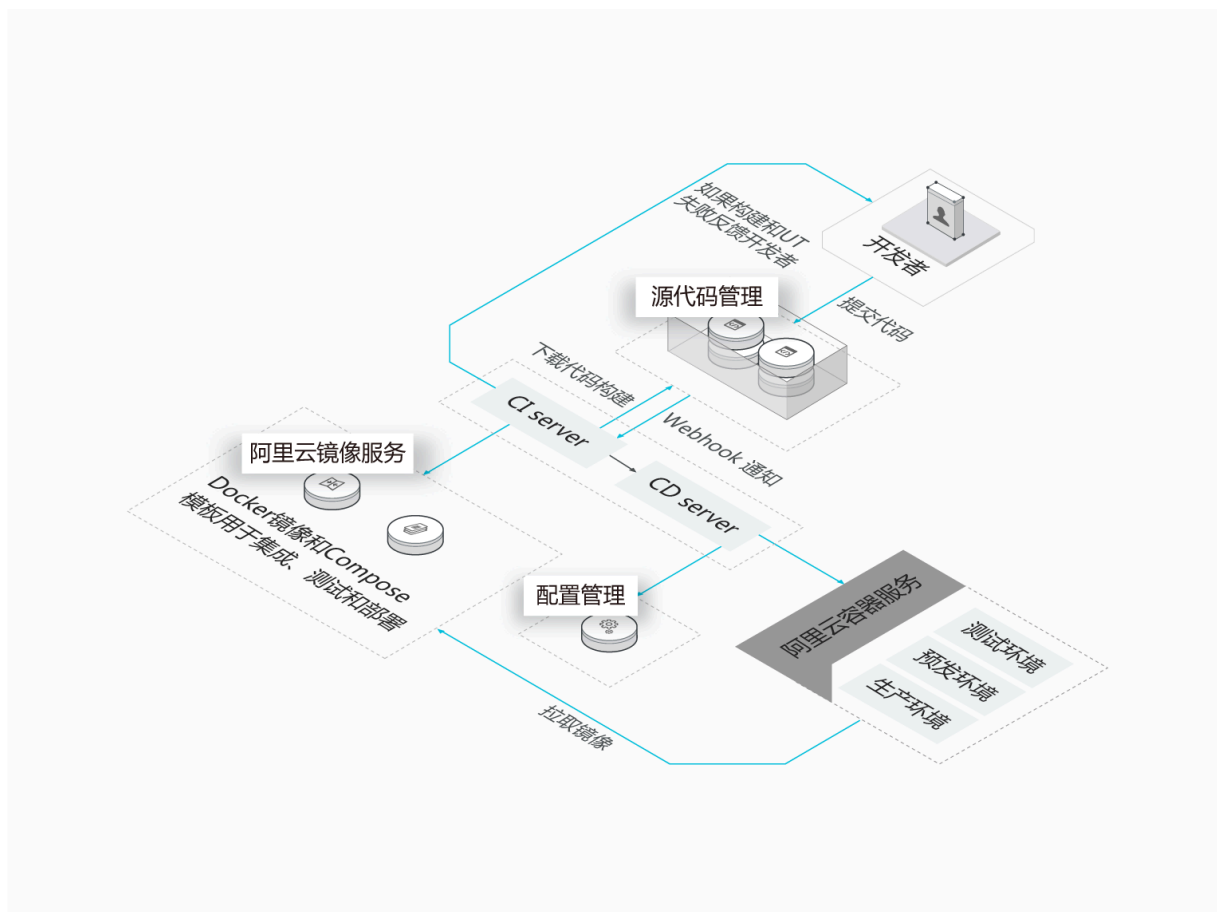
容器技术让您交付的不仅是代码，还有基于不可变架构的运行环境。

- 持续反馈

每次集成或交付，都会第一时间将结果实时反馈。

推荐搭配使用：

云服务器 ECS + 容器服务



基于高性能计算的机器学习

专注机器学习本身，快速实现从 0 到 1

帮助数据工程师在 HPC 集群上轻松部署机器学习应用，跟踪试验和训练、发布模型，数据部署在分布式存储，无需关心繁琐部署运维，专注核心业务，快速从 0 到 1。

能够实现：

- 快速弹性

一键部署机器学习应用，秒级启动和弹性伸缩。

- 简单可控

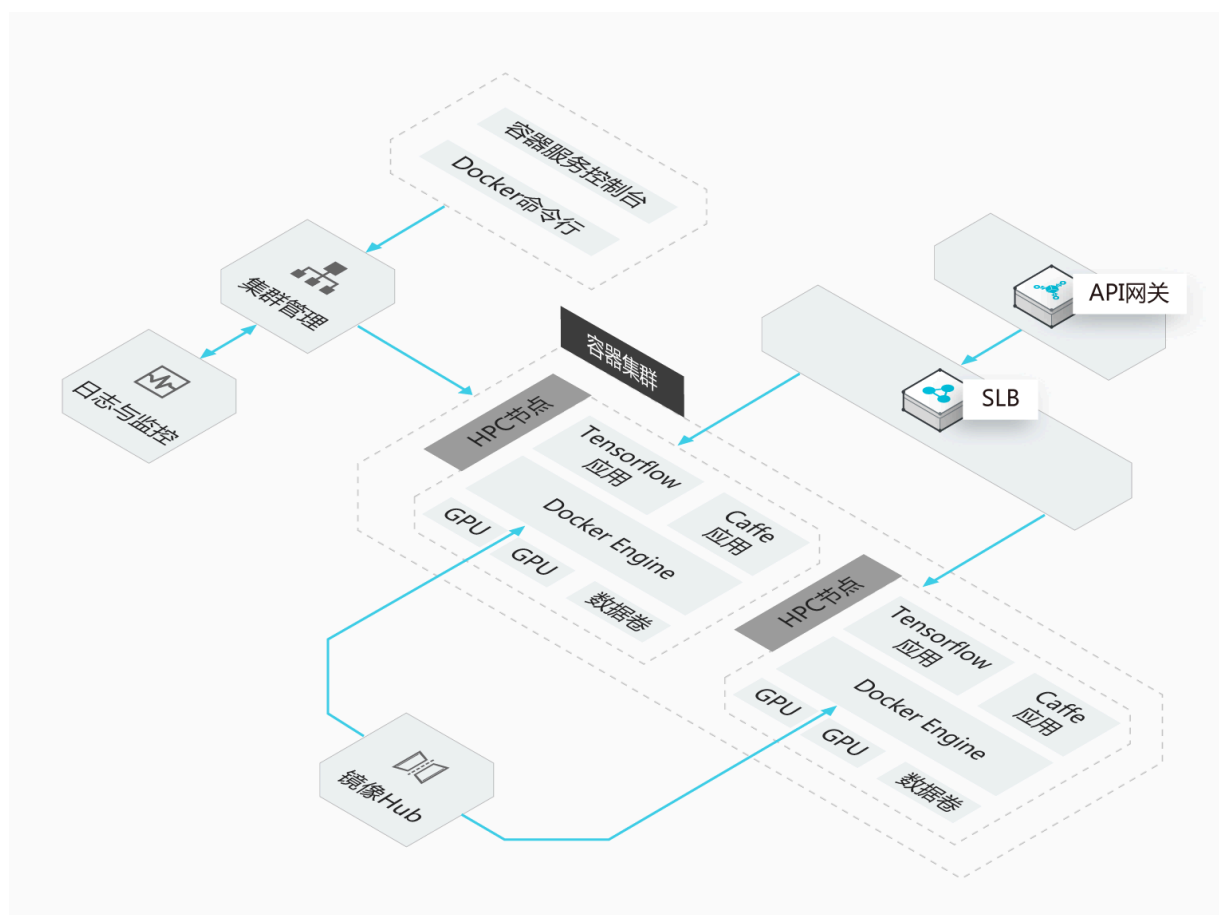
一行配置轻松获取 GPU 计算能力，并且可以监控 GPU 的资源。

- 深度整合

无缝接入阿里云存储、日志监控和安全基础架构能力。

推荐搭配使用：

高性能计算 (Alibaba Cloud HPC) + 容器服务 + 阿里云文件存储 NAS + 对象存储 OSS



微服务架构

实现敏捷开发和部署落地，加速企业业务迭代

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，由阿里云提供调度、编排、部署和灰度发布能力。

能够实现：

- 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

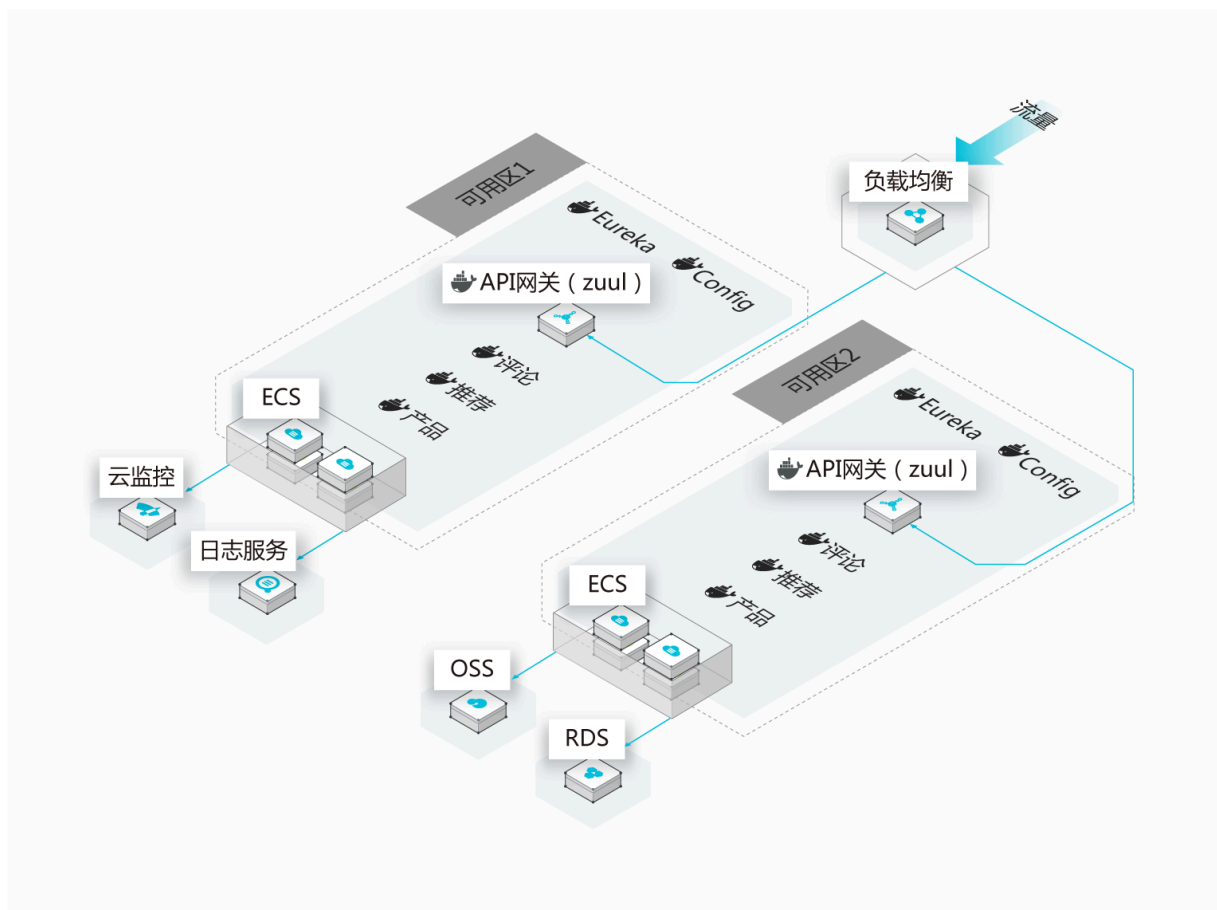
支持服务级别的亲和性调度，支持跨可用区的高可用和灾难恢复。

- 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。

推荐搭配使用：

云服务器 ECS + 云数据库 RDS 版 + 对象存储 OSS + 容器服务



混合云架构

统一运维多个云端资源

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。

能够实现：

- 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

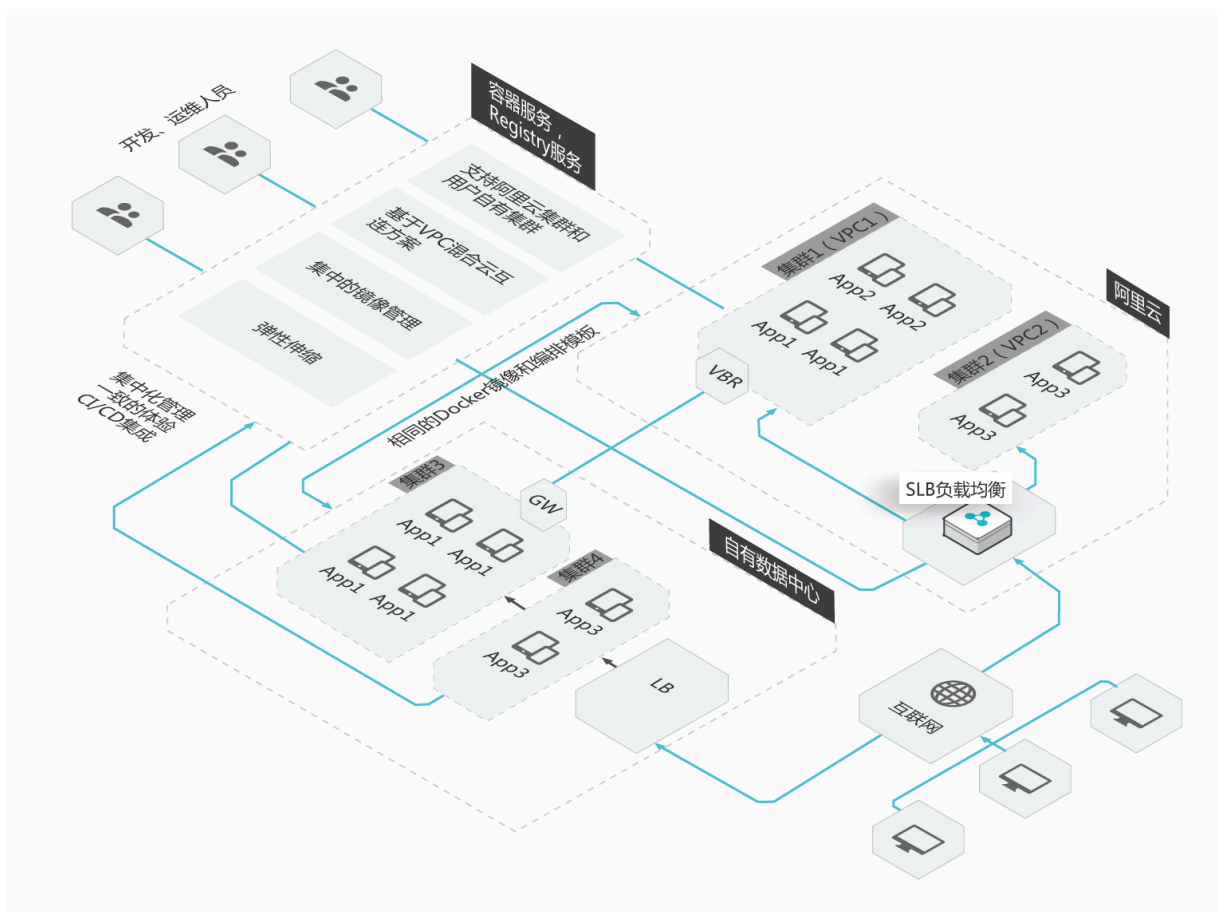
业务系统同时部署到云上和云下，云下提供服务，云上容灾。

- 云下开发测试

云下开发测试后的应用无缝发布到云上。

推荐搭配使用：

云服务器 ECS + 专有网络 VPC + 高速通道 (Express Connect)



弹性伸缩架构

根据业务流量自动对业务扩容/缩容

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。

能够实现：

- 快速响应

业务流量达到扩容指标，秒级触发容器扩容操作。

- 全自动

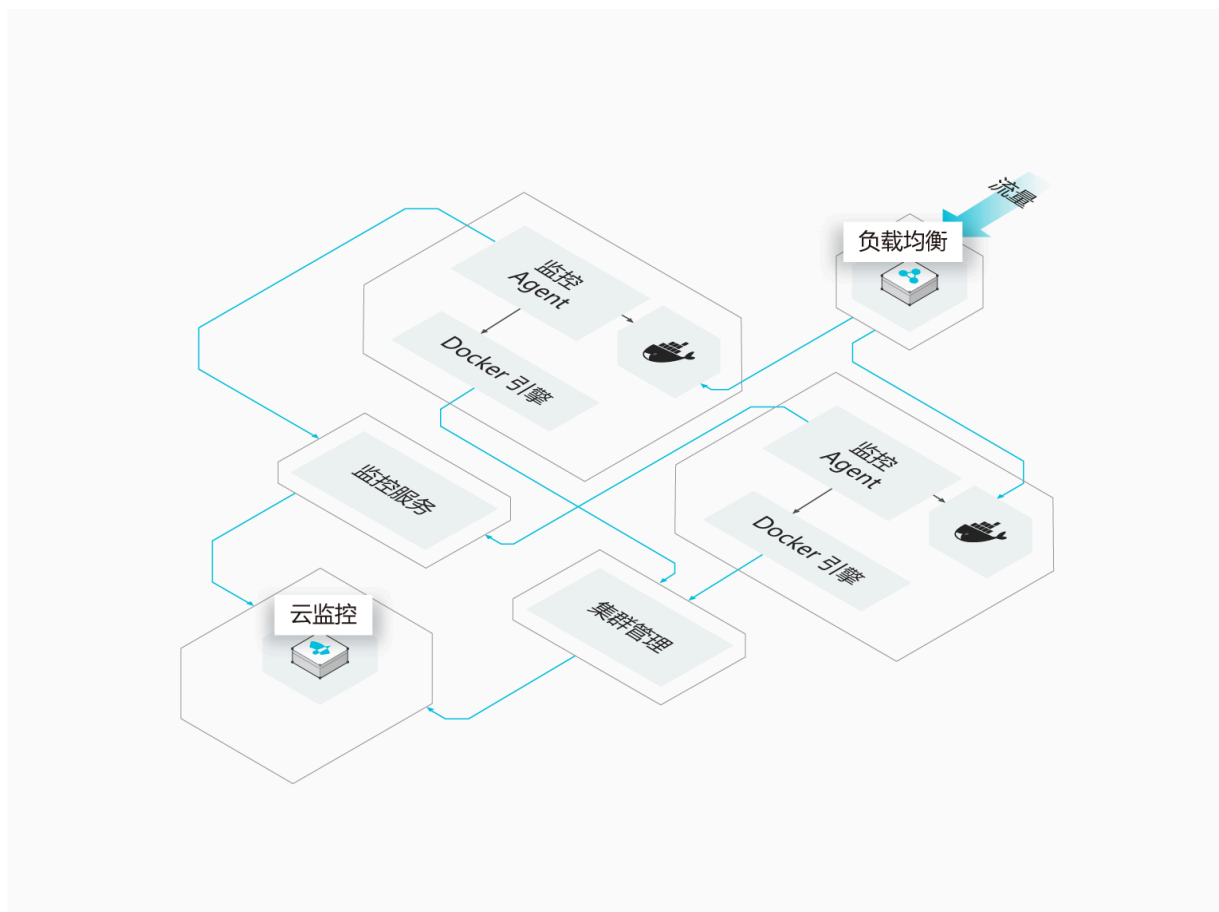
整个扩容/缩容过程完全自动化，无需人工干预。

- 低成本

流量降低自动缩容，避免资源浪费。

推荐搭配使用：

云服务器 ECS + 云监控



5 基本概念

5.1 基本概念

基本概念

集群

一个集群指容器运行所需要的云资源组合，关联了若干服务器节点、负载均衡、专有网络等云资源。

托管集群 (Managed Kubernetes Cluster)

只需创建Worker节点，Master节点由容器服务创建并托管。具备简单、低成本、高可用、无需运维管理Kubernetes集群Master节点的特点。

专有集群 (Dedicated Kubernetes Cluster)

需要创建3个Master（高可用）节点及若干Worker节点，可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。

Serverless集群 (Serverless Kubernetes Cluster)

无需创建和管理Master节点及Worker节点，即可通过控制台或者命令配置容器实例的资源、指明应用容器镜像以及对外服务的方式，直接启动应用程序。

节点

一台服务器（可以是虚拟机实例或者物理服务器）已经安装了 Docker Engine，可以用于部署和管理容器；容器服务的 Agent 程序会安装到节点上并注册到一个集群上。集群中的节点数量可以伸缩。

容器

一个通过 Docker 镜像创建的运行实例，一个节点可运行多个容器。

镜像

Docker 镜像是容器应用打包的标准格式，在部署容器化应用时可以指定镜像，镜像可以来自于 Docker Hub，阿里云镜像服务，或者用户的私有 Registry。镜像 ID 可以由镜像所在仓库 URI 和镜像 Tag（缺省为 latest）唯一确认。

Kubernetes 相关概念

管理节点 (Master Node)

管理节点是Kubernetes集群的管理者，运行着的服务包括kube-apiserver、kube-scheduler、kube-controller-manager、etcd和容器网络等组件。一般3个管理节点组成HA的架构。

工作节点 (Worker Node)

工作节点是Kubernetes集群中承担工作负载的节点，可以是虚拟机也可以是物理机。工作节点承担实际的 Pod 调度以及与管理节点的通信等。一个工作节点上的服务包括Docker运行时环境、kubelet、Kube-Proxy以及其它一些可选的Add-on组件。

命名空间 (Namespace)

命名空间为 Kubernetes 集群提供虚拟的隔离作用。Kubernetes 集群初始有 3 个命名空间，分别是默认命名空间 default、系统命名空间 kube-system 和 kube-public，除此以外，管理员可以创建新的命名空间以满足需求。

Pod

Pod是 Kubernetes 部署应用或服务的最小的基本单位。一个Pod 封装多个应用容器（也可以只有一个容器）、存储资源、一个独立的网络 IP 以及管理控制容器运行方式的策略选项。

副本控制器 (Replication Controller, RC)

RC 确保任何时候 Kubernetes 集群中有指定数量的 pod 副本(replicas)在运行。通过监控运行中的 Pod 来保证集群中运行指定数目的 Pod 副本。指定的数目可以是多个也可以是1个；少于指定数目，RC 就会启动运行新的 Pod 副本；多于指定数目，RC 就会终止多余的 Pod 副本。

副本集 (Replica Set, RS)

ReplicaSet (RS) 是 RC 的升级版，唯一区别是对选择器的支持，RS 能支持更多种类的匹配模式。副本集对象一般不单独使用，而是作为 Deployment 的理想状态参数使用。

部署 (Deployment)

部署表示用户对 Kubernetes 集群的一次更新操作。部署比 RS 应用更广，可以是创建一个新的服务，更新一个新的服务，也可以是滚动升级一个服务。滚动升级一个服务，实际是创建一个新的 RS，然后逐渐将新 RS 中副本数增加到理想状态，将旧 RS 中的副本数减小到 0 的复合操作；这样一个复合操作用一个RS是不太好描述的，所以用一个更通用的 Deployment 来描述。不建议您手动管理利用 Deployment 创建的 RS。

服务 (Service)

Service 也是 Kubernetes 的基本操作单元，是真实应用服务的抽象，每一个服务后面都有很多对应的容器来提供支持，通过 Kube-Proxy 的 port 和服务 selector 决定服务请求传递给后端的容器，对外表现为一个单一访问接口，外部不需要了解后端如何运行，这给扩展或维护后端带来很大的好处。

标签 (labels)

Labels 的实质是附着在资源对象上的一系列 Key/Value 键值对，用于指定对用户有意义的对象的属性，标签对内核系统是没有直接意义的。标签可以在创建一个对象的时候直接赋予，也可以在后期随时修改，每一个对象可以拥有多个标签，但 key 值必须唯一。

存储卷 (Volume)

Kubernetes 集群中的存储卷跟 Docker 的存储卷有些类似，只不过 Docker 的存储卷作用范围为一个容器，而 Kubernetes 的存储卷的生命周期和作用范围是一个 Pod。每个 Pod 中声明的存储卷由 Pod 中的所有容器共享。支持使用 Persistent Volume Claim 即 PVC 这种逻辑存储，使用者可以忽略后台的实际存储技术，具体关于 Persistent Volumn(pv)的配置由存储管理员来配置。

持久存储卷 (Persistent Volume, PV) 和持久存储卷声明 (Persistent Volume Claim, PVC)

PV 和 PVC 使得 Kubernetes 集群具备了存储的逻辑抽象能力，使得在配置 Pod 的逻辑里可以忽略对实际后台存储技术的配置，而把这项配置的工作交给 PV 的配置者。存储的 PV 和 PVC 的这种关系，跟计算的 Node 和 Pod 的关系是非常类似的；PV 和 Node 是资源的提供者，根据集群的基础设施变化而变化，由 Kubernetes 集群管理员配置；而 PVC 和 Pod 是资源的使用者，根据业务服务的需求变化而变化，由 Kubernetes 集群的使用者即服务的管理员来配置。

Ingress

Ingress 是授权入站连接到达集群服务的规则集合。你可以通过 Ingress 配置提供外部可访问的 URL、负载均衡、SSL、基于名称的虚拟主机等。用户通过 POST Ingress 资源到 API server 的方式来请求 Ingress。Ingress controller 负责实现 Ingress，通常使用负载均衡器，它还可以配置边界路由和其他前端，这有助于以 HA 方式处理流量。

相关文档

- [Docker glossary](#)
- [Kubernetes concepts](#)

5.2 与原生Kubernetes名词对照

本文主要为您介绍Kubernetes与原生Kubernetes名词对照情况。

容器服务ACK	原生Kubernetes
集群	Cluster
节点	Node

容器服务ACK	原生Kubernetes
容器	Container
镜像	Image
命名空间	Namespace
无状态	Deployment
有状态	StatefulSet
任务	Job
定时任务	CronJob
服务	Service
路由	Ingress
标签	Label
配置项	Configmap
保密字典	Secret
存储卷	PersistentVolume
存储声明	PersistentVolumeClaim
水平弹性伸缩	HPA
虚拟集群IP	Cluster IP
节点端口	NodePort
负载均衡	LoadBalancer
节点亲和性	NodeAffinity
应用亲和性	PodAffinity
应用非亲和性	PodAntiAffinity
选择器	LabelSelector
注解	Annotation
触发器	Webhook
端点	Endpoint
资源配额	Resource Quota
资源限制	Limit Range
模板	Template

5.3 容器服务 Kubernetes 集群三种形态对比

本文将介绍容器服务 Kubernetes 集群以下三种形态的对比情况：专有版 Kubernetes（Dedicated Kubernetes）、托管版Kubernetes（Managed Kubernetes）、Serverless Kubernetes。

特点

容器服务 Kubernetes 版（简称 ACK）提供高性能可伸缩的云原生应用管理能力，支持企业级 Kubernetes 容器化应用的全生命周期管理，并提供了以下三种 Kubernetes 集群形态：

- [专有版 Kubernetes](#)

需要创建 3 个 Master（高可用）节点及若干 Worker 节点，可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。

- [托管版 Kubernetes](#)

只需创建 Worker 节点，Master 节点由容器服务创建并托管。具备简单、低成本、高可用、无需运维管理 Kubernetes 集群 Master 节点的特点，您可以更多关注业务本身。

- [Serverless Kubernetes](#)

无需创建和管理 Master 节点及 Worker 节点，即可通过控制台或者命令配置容器实例的资源、指明应用容器镜像以及对外服务的方式，直接启动应用程序。

收费方式

- 专有版 Kubernetes

免费。用户承担 Master 节点和 Worker 节点的资源费用。

- 托管版 Kubernetes

免费。用户承担 Worker 节点的资源费用。

- Serverless Kubernetes

按容器实例的使用资源量和时长（秒）计费。

应用场景

- 专有版 Kubernetes

适用于所有场景。

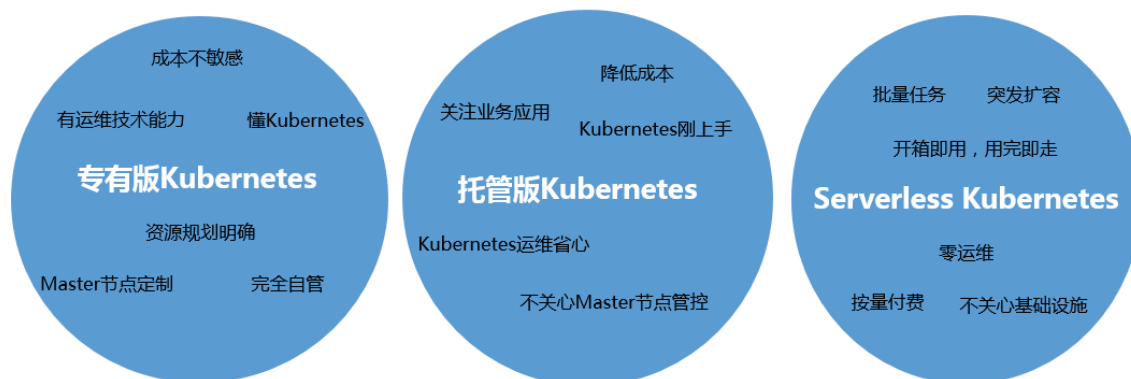
- 托管版 Kubernetes

适用于所有场景。

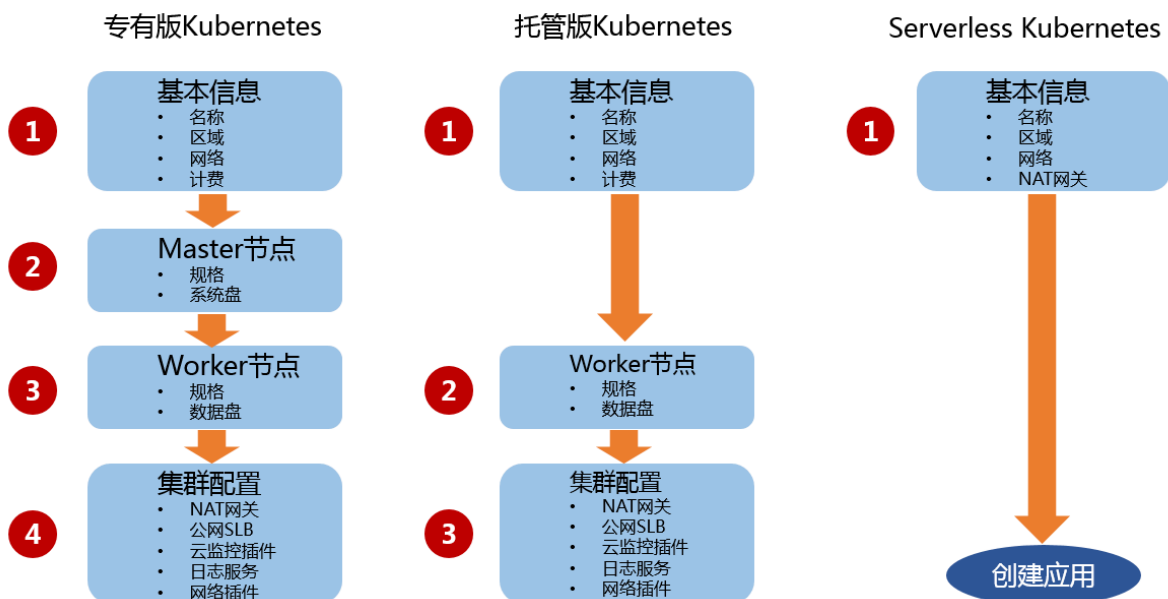
· Serverless Kubernetes

适用于批量任务、突发扩容，以及 CI/CD 测试。

三种集群形态的用户画像



集群创建流程对比



6 使用限制

阿里云容器服务 Kubernetes 集群的使用限制如下所示。

限制概述

使用容器服务Kubernetes版有下列限制：

- 集群一旦创建以后，暂不支持以下项：
 - 变更集群的VPC
 - Kubernetes集群更改为多可用区Kubernetes集群
 - Kubernetes集群更换为Kubernetes托管版集群
 - 增加公网Server Load Balancer暴露API Server
 - 增加云监控插件
 - 变更容器网络插件
- 不支持应用在不同命名空间下迁移。
- 由于创建集群时需要通过OpenAPI访问公网，因此，在创建专有版Kubernetes和托管版Kubernetes集群时，必须勾选NAT网关。
- 创建的ECS实例目前支持按量付费和包年包月，其他资源（比如负载均衡）为按量付费，您可以通过管理控制台将按量付费实例转换成包年包月实例。
- 根据 ROS 的规则，弹性缩容不会缩减创建集群时创建的节点以及您手动添加到集群中的节点，只会缩减您手动扩容出来的节点。缩减规则是按创建时间进行的，最新扩容出来的节点会被先回收。

容器服务Kubernetes版配额限制

限制项	普通用户限制	例外申请方式
例外申请方式	实名认证	没有例外
创建按量付费资源的限制	账户余额、代金券和信用度之和不得小于100元	工单申请
Kubernetes和Kubernetes托管版集群总数	10	工单申请
每个集群master和worker节点数总和	40	工单申请
每个worker节点创建容器实例最大数	创建集群时界面可设置，最大256	没有例外

限制项	普通用户限制	例外申请方式
Serverless Kubernetes集群总数	2	工单申请
每个Serverless Kubernetes集群创建容器实例最大数	300	工单申请

7 Kubernetes版本支持机制

为了更好地支持客户，让客户使用到既稳定又好用的Kubernetes，容器服务特推出Kubernetes版本支持机制，具体如下：

版本支持机制

容器服务ACK同时支持两个双数号的大版本，例如v1.12、v1.14。提供界面创建集群和后台维护的能力。

版本发布周期

- ACK保持每半年更新一次Kubernetes版本的频率。
- ACK一般在社区新版本发布后，经过2~3个月的仔细验证后推出相应版本。

版本升级和淘汰

- ACK提供Kubernetes版本的升级功能，当新版本发布上线时，会立即下线一个较早的版本；当新版本的升级功能发布后，会淘汰这个较早的版本。



说明：

- 下线：是指用户无法在界面上创建该版本的集群。
- 淘汰：是指停止对此版本的技术支持以及关闭升级通道。我们会给用户预留三个月的缓冲期来升级过老的Kubernetes版本。

例如，容器服务ACK目前支持的版本有v1.11、v1.12。当v1.14版本上线后，则较早的v1.11版本将会被下线。待v1.14版本的升级功能（新版本上线和新版本升级功能发布的间隔周期约一个月）发布后，我们会在3个月后淘汰v1.11版本。

- 版本升级功能当前只支持最近版本的升级，暂不支持跨多个版本的升级。