ALIBABA CLOUD

阿里云

专有云敏捷版云原生PaaS场景

容器服务ACK敏捷版产品简介

产品版本：V1.2.0

文档版本：20240924

阿里云

法律声明

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

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
警告	适用于可能会产生风险的场景。介绍用户在操作前就必须充分了解的信息、操作前必须要注意的事项或已具备的条件。	警告 重启操作将导致业务短暂中断，建议您在业务低峰期执行重启操作，或确保已完成数据备份。如有必要，请联系阿里云技术支持提供协助。
重要	在操作前需要用户了解的提示信息、补充信息、注意事项、限制信息等。	重要 再次登录系统时，您需要修改登录账户的初始密码。
说明	用于额外的补充说明、最佳实践、窍门等，不是用户必须了解的信息。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**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。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.什么是容器服务 05
2.什么是容器服务ACK敏捷版 06
3.功能特性 07
4.产品优势 09
5.应用场景 11
5.1. 容器化应用上云 11
5.2. 容器化DevOps 11
5.3. 微服务支持 11
5.4. 弹性伸缩 11
5.5. 混合云 12
5.6. 机器学习 12
6.注意事项 14
7.基本概念 17

1.什么是容器服务

容器服务提供高性能可伸缩的容器应用管理服务，支持用 Kubernetes 进行容器化应用的生命周期管理，提供多种应用发布方式和持续交付能力并支持微服务架构。可以直接部署在企业已有的x86 等硬件基础之上并加以管理，支持研发运维一体化、云原生应用架构和机器学习等场景，支持混合云管理，允许应用在公共云，和自有数据中心物理机统一部署管理，支持应用无缝迁云、弹性伸缩应对突发流量等场景。

2.什么是容器服务ACK敏捷版

Kubernetes是流行的开源容器编排技术。为了让用户可以方便地使用Kubernetes管理容器应用，容器服务ACK敏捷版（ACK Agility）提供了Kubernetes集群支持。

您可以通过容器服务ACK敏捷版创建一个安全高可用的Kubernetes集群，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。

Kubernetes支持对容器化应用程序的部署、扩展和管理。它具有以下功能：

集群管理

- 通过控制台一键创建Kubernetes集群。
- 提供容器优化的OS镜像，提供**稳定测试和安全加固**的Kuberntes和Docker版本。

一站式容器生命周期管理

- **网络**

  支持基于Vxlan的Overlay网络。

  支持Underlay和Overlay混部，支持由单一网络平滑升级到混部模式。

  灵活的IP分配特性，如有状态Pod固定/指定IP、按Pod指定网络类型及Subnet/Network、不连续网段分配、灵活网段扩缩容等。

- **存储**

  支持对接Open-Local、NFS和vCNS存储。

  支持对接通用CSI。

  支持存储卷动态创建。

- **日志**

  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。

- **应用管理**

  支持应用监控，应用弹性伸缩。

3.功能特性

资源调度

- 提供多种集群形态，可以管理客户混合、多云、分布式等异构基础设施。
- 支持信创环境，支持ARM与X86混合管理。
- 统一虚拟和容器的调度，统一资源分配，提升运维效率；统一网络和存储管理；基于物理机，减少不必要的虚拟化开销，减少虚拟化导致的碎片化问题。

微服务

- 内置通用的服务注册、发现、路由、负载均衡等机制，对开发语言和中间件无特殊需求。
- 提供声明式方式配置，无需编码。

DevOps

- 内置容器化DevOps最佳实践，可以实现一键式从代码提交到应用变更上线的全自动流程。
- 支持与三方、开源CI/CD方案整合。
- 提供不间断发布和蓝绿发布、金丝雀发布等灰度发布机制，支持灵活、可控的服务更新。

日志与监控

- 提供企业级日志采集和输出方案。无缝集成容器日志采集，支持采集标准输出或指定目录的日志。
- 可以选择对接阿里云日志服务，支撑超万台服务器，IOPS超2万。也可以对接ELK等三方开源或企业已有日志框架。
- 提供容器级别、应用级别和宿主机级别的多维度监控，提供服务和应用视角聚合数据。
- 支持脚本、URL等自定义监控。
- 可以选择对接阿里云云监控，也可以对接三方开源监控解决方案。
- 在云端提供对边缘设备、边缘应用的统一监控能力，保证边缘设备及边缘智能应用少运维、高可用。

安全

- 与企业用户目录无缝集成，支持统一用户认证管理。
- 支持基于角色的授权模型，对集群资源灵活控制。

镜像管理

- 支持高可用镜像仓库，支持高可用存储。
- 支持镜像安全扫描和数字证书签名，实现安全可控的协作和应用分发机制。

容器存储

- 本地磁盘池化为卷组统一调度、分配，支持存储卷快照、监控、扩容，以及一键化自动创建存储卷。
- 支持对接OpenLocal、NFS、vCNS等存储，支持对接通用CIS。
- 支持跨租户多服务共享同一存储资源、实现灵活资源配额管理。
- 提供块存储和文件存储，支持跨集群存储供给。

容器网络

- 提供CNI网络组件，支持基于Vxlan的Overlay网络
- 支持容器间高性能跨宿主机网络通信。
- 支持与企业现有网络方案对接。
- 灵活的 IP 分配特性，如有状态Pod 固定/指定 IP 、按Pod指定网络类型及Subnet/Network、不连续网段分配、灵活网段扩缩容等。

应用管理

- 支持应用灰度发布，蓝绿发布。
- 内置应用目录，支持Helm应用一键部署和升级。
- 支持服务目录，简化服务集成。
- 根据历史数据，给出应用资源用量的推荐。

混合云

- 与阿里云公共云无缝结合，实现混合云。
- 支持对自有数据中心和公共云容器集群的统一资源、镜像、应用和安全管理。
- 支持工作负载动态迁移。
- 支持云突发、云灾备、异地多中心管理等典型场景。

自动化运维与管理

- 基于阿里云飞天基础架构管理平台。
- 提供硬件资产管理、自动化监控、运维能力。
- 支持基础网络服务管理。

无缝对接阿里云公共云

- 提供通过专线、VPN等网络互通方式。
- 复用阿里云提供数据复制机制，可实现数据迁移、异地灾备等多种混合云业务应用场景。

4.产品优势

指标项	指标参数	
稳定	技术成熟	专有云敏捷版PaaS容器服务，和阿里云公有云容器服务，技术体系相同，代码相同。容器服务领域沉淀5余年，有着非常丰富的容器技术经验。容器服务支持的客户集群规模有10000+，容器数达百万级。
	高可用	平台所有组件不存在单点故障问题
	高性能	实现应用秒级启动
异构计算	异构OS	单容器平台有能力同时纳管Linux、Windows计算节点
	异构资源	有能力纳管CPU、GPU、FPGA等计算资源
	自主可控CPU	支持自主可控CPU。例如arm64、申威等
	虚机物理机混部	同时支持Kubernetes纳管虚拟机和物理机的计算节点
网络	容器固定IP能力	支持创建容器，暂停，重启，迁移后，自身IP保持不变
	多网络方案	支持Terway、Flannel、Calico等多Kubernetes网络方案，支持underlay/overlay两大主场景
智能运维	广播任务	支持Kubernetes发起广播任务，实现各节点上执行指定任务，例如资源清理，文件预热等
	有状态应用灰度	支持有状态应用的原地升级能力，保持IP不变
	原生可观测能力	支持容器平台本身以及运行业务应用的监控、日志、追踪功能，实现业务与平台异常问题根因定位收敛在到1分钟内

	智能升级能力	支持kubernetes无缝支持到一个新的版本，实现业务无感下的容器云平台升级
镜像	镜像安全性	实现镜像安全扫描，所有上传镜像实现漏洞报告输出
	异地仓库同步	设置跨域两个镜像仓库的镜像实现自动同步
	镜像分发效率	支持P2P镜像分发能力，支持上百节点高并发执行镜像下载任务时，镜像仓库依然提供稳定的下载服务
多集群	集群容量能力	容器云支持可扩展至10000个节点，容器云平台支持扩展至100个集群
	多集群能力	支持容器云平台管控组件，和容器云平台的业务应用，分别部署在不同的Kubernetes集群上，实现控制链路与业务数据链路的隔离
迁云	镜像迁移能力	有能力将多个源头镜像仓库的镜像迁移到指定的容器与镜像仓库
	Kubernetes对象迁移	有能力实现新老Kubernetes集群的对象导出，并实现在另一个Kubernetes集群上的重新导入运行

5.应用场景

5.1. 容器化应用上云

传统运行在数据中心的应用如果要向云上迁移面临的挑战是应用的运行环境的适配以及应用部署方式的改变。原有数据中心的环境和云环境的差异越大，迁移的成本越高。

由于容器技术具有跨平台移植的特性，可以解决不同环境下应用运行差异问题。无论用户的应用是何种语言，何种技术栈，都能够通过容器技术封装成为跨平台可用的Docker镜像。从而将原来繁重的应用迁移变成一系列标准化的操作。

一般情况下，在数据中心里运行的虚拟机镜像无法直接迁移到云上环境，原来的镜像构建和部署脚本都要重新开发。采用容器技术可以使用容器镜像替代虚拟机镜像。利用容器技术将原有的传统应用镜像化，可以在尽量减少或者没有改变原有代码的情况下快速上云，并保证部署的一致性。

利用Docker Compose部署模版，可以描述完整的应用栈和所需云资源。容器服务扩展了Compose模板支持服务治理：包括服务调度约束、服务注册/发现策略、弹性伸缩等；以及容器与外部资源的动态集成，如负载均衡，网络存储，RDS数据库服务等等。这样可以采用应用为中心的管理方式，大大提升了部署、运维效率。

这个方案的特点是：成本低、上线速度快，并且在完成迁移后可以利用容器服务的微服务支持进行进一步微服务化改造。

5.2. 容器化DevOps

开发和运维中交接的代码，由于同一代码在不同环境中的差异性，往往导致很高的调试成本，软件的交付和测试周期长。开发运维团队需要花费很大的精力来屏蔽系统差异，标准化交付手段。

容器技术作为软件的交付手段，具有同一应用在多个平台的可移植性，和不同应用在同一环境中的相互隔离的特点。利用这一特性，使容器成为开发环境、测试环境、生产环境的标准代码交付手段。在任何环境中运行的都是同一个容器，这样保证了由于环境不同所带来的差异被最大限度地屏蔽。利用这种标准交付手段，可以大大提高开发和运维的效率。

容器服务提供了容器化Jenkins的一键式部署模板，可以快速灵活地搭建一套CI/CD环境。同时还提供了Jenkins插件来充分发挥容器服务的能力，比如多种发布模式，保证新系统和服务上线的可靠性。比如通过灰度发布，可以灵活控制新老版本的流量切换，快速、可靠地回滚。

5.3. 微服务支持

微服务作为云应用常见的架构模式正在被越来越多地接受。但是随着服务实例数量的快速增长，对服务治理的要求会越来越高。容器服务提供了和语言无关的微服务治理能力，您不限定语言和开发框架，不用改变应用逻辑，就可以轻松应对微服务的管理和伸缩。所有这些能力都可以通过在编排模版中以声明的方式指定，或者在容器服务控制台中可视化地指定。

从对SLA保证角度，容器服务提供了利用自定义命令 `/HTTP` 等健康检查的能力，可以细粒度地评估应用的健康状况。容器服务还可以按照资源约束调度和再平衡，当节点故障时自动重调度，按照服务亲和性和跨可用区调度约束等机制保证可用性。

服务发现是微服务平台的一个重要能力，所有服务的启动和停止都能够自动注册到平台，服务之间可以非常容易地进行服务发现。更进一步，配合Routing Mesh，简单路由服务可以为应用提供灵活的4层、7层路由方式。

对于任何运维团队来说，日志的集中和分析，云资源的集中监控都必不可少。容器服务提供了声明式日志、监控采集方案，只需要简单声明要采集的日志目录，就可以自动收集日志。支持对接多种日志、监控存储。

配合监控服务，在您自定义弹性伸缩策略后，容器服务对应用的指标进行监控，在触发条件时按照策略进行应用弹性扩容和缩容。容器服务同时也支持集群弹性伸缩。

5.4. 弹性伸缩

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

5.5. 混合云

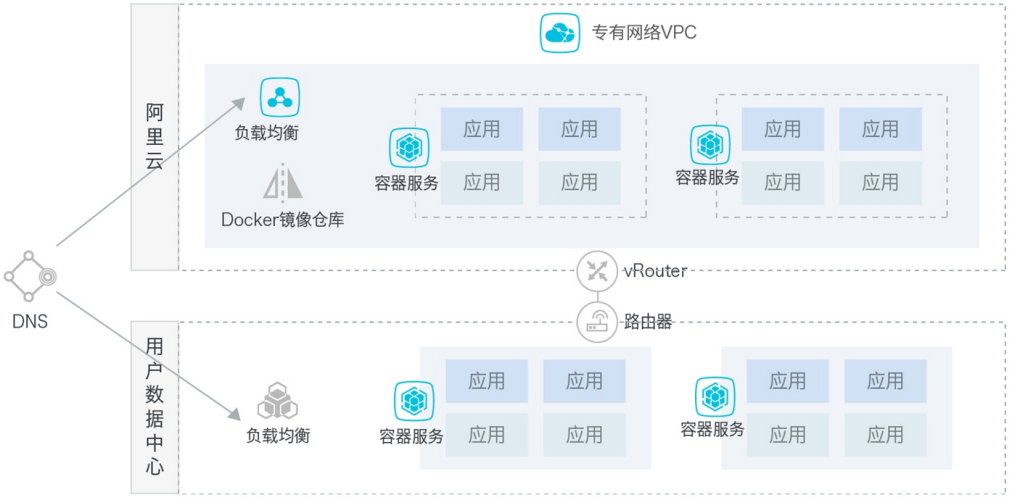
容器服务支持计算能力跨界迁移，通过容器服务可以管理本地专有云和公共云容器集群，提供了对所有环境完全一致的管理方式，您无需针对不同的其他云服务商或者IDC机器寻求特定的管理方案。使用同一套镜像和编排模板让应用可以无缝地在云间迁移。

混合云的典型场景包括：

- 场景1：应用快速在云端部署伸缩，应对大促突发流量峰值
- 场景2：异地、同城灾备
- 场景3：本地的机器开发测试，云端预发生产；或反之

针对混合云，为了提供最佳的网络实现，容器服务提供了不同的网络方案来适配不同环境。对于IDC与阿里公共云容器互通的需求，提供基于VPC的混合云互连方案，使用VPC+专线/VPN，可以实现IDC机器、IDC容器、ECS机器、ECS容器的全部互通。容器跨宿主机互连方案包括：Overlay（VXLAN）/Layer 3（Calico）等。

图 1. 混合云架构



5.6. 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中 ，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等基础架构能力。

6.注意事项

本文介绍使用容器服务ACK敏捷版时的注意事项。

环境兼容性列表

管控部分

环境	说明
芯片架构	Intel x86
部署形态	物理机、虚拟机
虚拟化环境	ECS、ZStack、VMWare、OpenStack等。
网络模型	Overlay（VPC）
存储	Host Path
操作系统	CentOS 7.6、CentOS7.9

客用部分

环境	说明
芯片架构	Intel x86
部署形态	物理机、虚拟机
虚拟化环境	ECS、zStack
网络模型	Underlay（经典网络）、Overlay（VPC）
存储	本地存储，也支持外部存储（如NFS、Ceph、GlusterFS等）
操作系统	• CentOS：7.6、7.7、7.8、7.9、8.0 • Redhat：7.6、7.7、7.8、7.9、8.0
支持接入的K8s版本	1.14、1.17
支持对接的Harbor版本	1.9.3、1.10.2
存储服务支持的NFS版本	v4

容量限制

限制项	限制说明

创建集群时需要符合的资源准入条件	• Master节点磁盘要求
	  ◦ 最小节点数：3
	  ◦ 最小规格（vCPU/Mem）：16 C/32 G
	  ◦ 系统盘≥50 GB
	  ◦ 数据盘1：1.4 T （控制面所需存储）
	  ◦ 数据盘2：50 G （管控数据库所需存储，需要独占盘，推荐SSD）
	• Worker节点磁盘要求
	  ◦ 最小节点数：0
	  ◦ 最小规格（vCPU/Mem）：4 C/8 G
	  ◦ 系统盘≥50 GB
	  ◦ 数据盘1：400 G （控制面所需存储）

集群	• 支持接入集群的版本范围：1.14，1.17 • 支持数据中心个数：不限制
镜像服务	• 支持Harbor对接的版本： 1.10.2和1.9.3版本 • 支持镜像最大的大小：2 G • 仓库的容量总大小：500 G
日志服务	• 日志文件大小限制：包含3个节点，每个节点最多30 G • 日志备份次数：由您自己设定，默认一天一次
集群监控	• 监控存储时长：15天 • 告警条数容量：30 G
其他	• 存储服务支持的NFS版本：NFSv4 • 对外路由最大可配置域名个数：不限制 • 备份空间大小限制和任务个数限制：由您自己设定

性能指标

指标	说明
镜像仓库服务RTO（分）	≤1
镜像仓库服务RPO（分）	≤15
集群数（个）	3
单集群节点数（个）	≤300
集群节点总数（个）	≤300
Pod总数（个）	≤300
单集群Pod总数（个）	≤300

每个节点的Pod数量（个）	≤100
1000个单副本deploy用例 P99 Pod启动时间（s）	3
一个1000副本deploy用例 P99 Pod启动时间（s）	3
调度器吞吐量（Pod/S）	15
node/patchP99 响应时间（ms）	50
node/getP99 响应时间（ms）	50
node/listP99 响应时间（ms）	50
pod/status-patchP99 响应时间（ms）	50
pod/cluster-listP99 响应时间（ms）	100
pod/listP99 响应时间（ms）	50
pod/GetP99 响应时间（ms）	50
pod/DeleteP99 响应时间	50
10并发pull同一个镜像（s）	pull平均RT为11s
10并发pull5个不同镜像（s）	pull平均RT为11s

7.基本概念

- ***集群***
  - 一个集群指容器运行所需要的资源组合，关联了若干服务器节点等资源。
- ***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镜像创建的运行时实例，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，阿里云容器镜像服务，容器服务的DTR，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。
- ***编排模板***
  - 编排模板包含了一组容器服务的定义和其相互关联，可以用于多容器应用的部署和管理。

Kubernetes 相关概念

Kubernetes是一个 Google 开源的，用于自动部署、扩展和管理容器化应用的大规模容器编排调度系统。具有可移植、可扩展和自动调度等特性。

- ***节点（Node）***
  - Kubernetes 集群中的计算能力由 Node 提供，Kubernetes 集群中的 Node 是所有 Pod 运行所在的工作主机，可以是物理机也可以是虚拟机。工作主机的统一特征是上面要运行 kubelet 管理节点上运行的容器。
- ***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，管理员可以可以创建新的名字空间满足需要。
- ***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 的升级版本，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。您可以给 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。