

## 目录

目录	1
容器服务	2
Serverless容器服务	2
产品简介	2
核心优势	2
简单易用	2
原生支持	2
生态兼容	2
安全隔离	2
应用场景	2
在线业务	2
离线计算	2
实时音视频处理	2
产品功能	2
集群管理	2
镜像管理	2
名词解释	3
基本概念	3
Kubernetes相关概念	3
产品优势	4
安全可靠	4
高效部署	4
简单易用	4
低成本	4
使用场景	4
微服务架构	4
持续集成/持续交付	5
弹性伸缩	6

# 容器服务

金山云容器服务（Kingsoft Cloud Container Engine, KCE）基于原生的Kubernetes进行开发和适配，整合了金山云虚拟化、网络、存储等能力，为客户提供高可靠、高性能、高度可扩展的容器管理服务。支持资源调度、服务发现、灰度升级，弹性伸缩和服务编排等功能特性，方便用户在金山云容器平台轻松部署、运维容器化的应用。

## Serverless容器服务

### 产品简介

Serverless容器服务（Kingsoft Cloud Serverless Kubernetes，简称KSK）是一种基于容器实例，同时完全兼容Kubernetes生态的容器服务。您无需购买和管理底层节点即可直接部署容器应用，并通过Kubernetes原生方式进行管理，低门槛实现Kubernetes应用的灵活部署。

KSK集群中的Pod基于金山云容器实例KCI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技术完全强隔离，容器实例间互不影响。

### 核心优势

#### 简单易用

无需购买和管理底层节点即可快速创建全托管的Serverless集群，用户能更专注于应用程序。

#### 原生支持

完全兼容原生的Kubernetes集群管理方式，支持原生的k8s api，用户可无缝迁移Kubernetes应用。

#### 生态兼容

与金山云其他产品深度集成，如VPC/SLB/EBS等，提供整合公有云能力的多种解决方案。

#### 安全隔离

基于容器实例采用的安全容器技术，提供虚拟机级别的安全隔离能力。

### 应用场景

#### 在线业务

基于集群中的监控指标与Kubernetes的HPA机制，根据应用的实际负载高效灵活伸缩，降低运维成本的同时提升业务效率。

#### 离线计算

对于离线批量的静态数据处理，利用KSK集群中提供的原生Job/Crono job资源实现离线报表、数据分析等一次性/定时周期性任务。KSK集群仅需对工作负载实际消耗的资源进行付费，无需为冗余资源投入额外成本。

#### 实时音视频处理

流媒体直播加速服务中使用KSK集群承载编解码、转码等实时音视频处理任务，应对直播业务场景对稳定流畅、支持高并发的需求。使用KSK集群，无需考虑节点扩容周期，以秒级启动速度实现容器实例级别扩容

## 产品功能

### 集群管理

1. 用户可以方便高效管理容器运行的集群，支持金山云现有的所有主机类型。
2. 用户独享集群资源，在底层资源层面实现多租户的隔离，保证用户业务的隔离性。
3. 集群支持在VPC内部跨可用区部署，有效保证用户业务的高可用性。
4. 集群支持伸缩功能，用户可根据自己业务的负载情况，自由调整集群规模。
5. 支持集群内节点维度的监控，用户可以自定义监控告警策略。

### 镜像管理

1. 提供Docker官方镜像下载加速服务。
2. 提供安全可靠的企业级私有镜像仓库，用户可将私有的镜像托管至私有的镜像仓库中，方便管理。
3. 提供用户私网下载私有镜像和Docker官方镜像的能力，加快镜像拉取的速度。
4. 提供全地域、公网获取镜像的能力。

## 名词解释

### 基本概念

**集群：**指容器运行的所需要的云资源的组合，包含了若干的云服务器、负载均衡、专有网络等云资源。

**独立部署集群：**金山云容器服务提供用户具有集群全部操作权限的集群，集群的Master节点和ETCD节点独立部在用户的VPC，用户具有集群全部的操作管理权限。

**托管集群：**金山云容器服务提供Master和ETCD全托管的集群，用户只需要创建集群的Worker节点，无需关心集群的控制管理平面。

**Serverless集群：**Serverless集群是一种基于容器实例，同时完全兼容Kubernetes生态的集群。用户无需购买和管理底层节点即可直接部署容器应用，并通过Kubernetes原生方式进行管理。

**节点：**节点为注册到容器集群的单一服务器资源，可以是云服务器、裸金属服务器、GPU服务器等等。

**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。

**服务：**服务为一组配置相同的容器以及如何访问他们的规则，即微服务架构中的部署单元，即是用户业务应用的最小运行单元。

**实例：**由相关的一个或多个容器构成一个实例，对应 Kubernetes 的 pod，这些实例共享相同的存储和网络空间。

**配置项：**配置项为多个配置的集合，方便管理不同环境和不同业务。

**镜像：**对应 Kubernetes 的 Images，是对所要部署服务的操作系统、运行环境、应用的打包，用户使用镜像来部署服务中的副本。Docker镜像存储在镜像仓库中。

**镜像仓库：**用于存放Docker的镜像。

### Kubernetes相关概念

**命名空间：**一个 Kubernetes 集群支持设置多个命名空间（Namesapce），每个命名空间相当于一个相对独立的虚拟空间，不同空间的资源相互隔离互不干扰。集群可通过命名空间对资源进行分区管理。

**Pod：**Pod 是 Kubernetes 部署应用或服务的最小的基本单位。由相关的一个或多个容器构成，一个Pod中的容器共享的存储和网络空间。

**Deployment：**为pod和replicaset提供了声明式的定义方法，适用于部署无状态的应用。您可以根据业务需求，对 Deployment 中运行的 Pod 的副本数、调度策略、更新策略等进行声明。

**StatefulSet：**StatefulSet是用来管理有状态应用的工作负载对象。StatefulSet 管理基于相同容器规约的一组 Pod。和 Deployment 不同的是，StatefulSet 为它们的每个 Pod 维护了一个持久型的标识符，Pod 迁移或销毁重启后，标识符仍会保留。如果应用程序不需要任何稳定的标识符或有序的部署、删除或伸缩，建议您使用Deployment部署

**DaemonSet：**DaemonSet 确保全部（或者某些）节点上运行一个 Pod 的副本。当有节点加入集群时，也会为他们新增一个 Pod 。当有节点从集群移除时，这些 Pod 也会被回收。删除 DaemonSet 将会删除它创建的所有 Pod。DaemonSet的典型用法如在每个节点上运行集群守护程序、在每个节点运行日志采集程序等等。

**Job：**Job 会创建一个或者多个 Pods，并确保指定数量的 Pods 成功终止。

**CronJob：**是一种基于时间调度的 Jobs，可以预定计划周期性地运行一个 Job。

**Service：**用来管理集群中四层网络的服务访问。基于四层网络，Service 提供了集群内容器服务的暴露能力。

**Ingress：**Ingress是授权进站连接到达集群服务的规则集合。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。

**ConfigMap：**ConfigMap用来将非机密性的数据保存到键值对中，Pods 可以将其用作环境变量、命令行参数或者存储卷中的配置文件。ConfigMap 将您的环境配置信息和容器镜像解耦，便于应用配置的修改。

**Secret：**key-value 类型的键值对，用来保存敏感信息，例如密码、OAuth 令牌和 SSH 密钥。

**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。

**PersistentVolume (PV)** : PV是集群中的存储资源，可以由管理员事先供应，或者 使用存储类 (Storage Class) 来动态供应。

**PersistentVolumeClaim (PVC)** : 用户对存储的请求。概念上与 Pod 类似。Pod 会耗用节点资源，而 PVC 申领会耗用 PV 资源。Pod 可以请求特定数量的资源 (CPU 和内存)；同样 PVC 申领也可以请求特定的大小和访问模式。

**StorageClass:** 描述存储的类型，集群管理员可以为集群定义不同的存储类别。

## 产品优势

### 安全可靠

1. 用户独享计算资源，在底层资源层面保证了多租户的隔离，保证了用户业务的隔离性。
2. 集群部署在客户VPC内部，客户可根据业务的安全需求，自定义安全组和ACL规则。
3. 容器集群采用分布式架构部署，保证集群的高可用。

### 高效部署

1. 金山云容器服务运行在您的私有网络中，高品质的 BGP 网络保证镜像极速上传下载，轻松支持海量容器秒级启动，极大程度降低了运行开销，使您的部署更加专注于业务运行。
2. 您可以在金山云容器服务上部署业务，开发人员在 GitHub 或其他代码平台提交代码后，容器服务可立即进行构建、测试、打包集成，将集成的代码部署到预发布环境和现网环境上。

### 简单易用

1. 金山云容器服务提供超大规模的集群管理服务，客户无需安装、运维、扩展集群管理的基础设施，只需要启动容器集群并指定想要运行的任务即可。
2. 支持客户一键化部署集群，支持客户自定义集群的规模和配置。
3. 客户可以通过提供可视化界面和Open API对容器集群进行操作。

### 低成本

1. 金山云容器服务暂不收取服务本身费用，客户仅需要对使用的云资源（例如云服务器，弹性IP等）付费。

## 使用场景

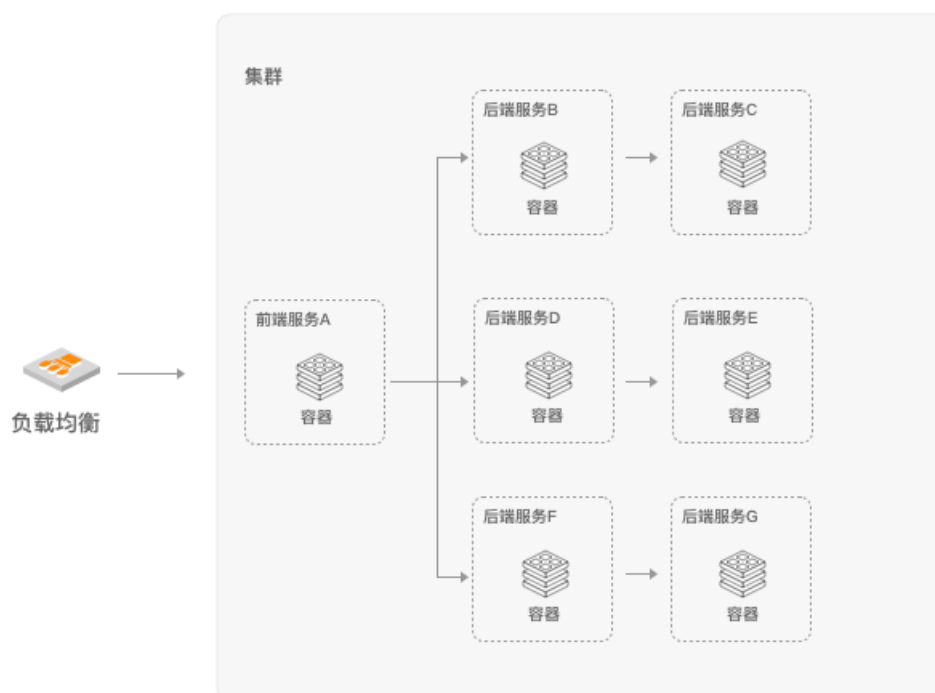
### 微服务架构

金山云容器服务提供应用微服务化的能力，用户可以将建构复杂的应用从不同维度拆分为多个微服务，每个微服务可以独立部署，具有强大的分布式能力、弹性扩展能力和容错能力。

每个微服务使用一个应用镜像作为模板进行微服务的部署。

#### 优势:

1. 支持负载均衡和服务自动发现，完美适配微服务架构。
2. 支持一键化部署集群，简化集群的部署和运维工作，轻松管理集群。
3. 与金山云生态深度融合，直接使用金山云计算，网络，安全，存储等能力。



## 持续集成/持续交付

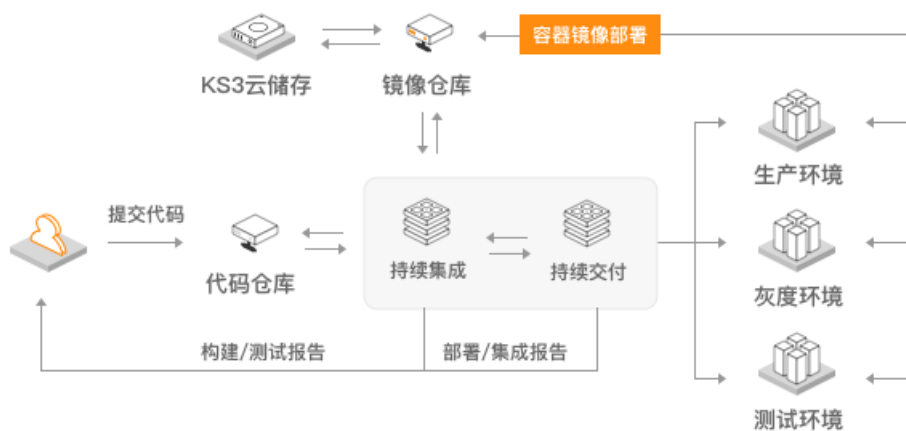
持续集成与持续交付提供的优秀的 DevOps 环境，可极大提高软件的发布效率。

持续集成：持续集成强调开发人员提交了新代码之后，立刻进行构建、（单元）测试，便于确定新代码和原有代码能否正确地打包集成在一起。

持续交付：持续交付在持续集成的基础上，将集成的代码部署到预运行环境。

### 优势：

提供持续集成/持续交付的能力，为用户业务软件开发、编译、测试、集成部署提供了全套的、完整的流程，消除开发、测试、上线过程中出现的人为低级失误，提高生产效率。



## 弹性伸缩

金山云容器服务可以根据服务的负载情况自动进行扩/缩容，当服务负载较高时，自动进行服务扩容并秒级部署，响应并发请求；在服务负载较低时，在不影响线上服务的前提下自动执行缩容，节约资源成本。全程不需要人为干预，为您免去人工部署的负担。

### 优势：

1. 扩容/缩容过程自动化，无需人为干预。
2. 服务负载达到扩容指标，秒级触发容器扩容操作。
3. 服务负载较低时，自动进行缩容，降低成本。

