

目录

目录	1
镜像管理	2
应用管理	2
访问接入	4
存储管理	5
集群管理	5

# 镜像管理

## 基本概念

镜像是一个模板，是容器应用打包的标准格式，由多个镜像层叠加起来的一个文件系统，用于创建容器。除了提供容器运行时所需的程序、库、资源、配置等文件外，还包含了一些为运行时准备的配置参数（如匿名卷、环境变量等）。镜像可以来自于 Docker Hub、KENC 公开镜像服务、或用户私有的镜像仓库等。在金山云边缘节点计算产品中，业务快速部署的基础为：将镜像快速下发到各个边缘节点，实现服务运行环境的快速构建。

## 镜像仓库基本操作

### 1. 开通镜像仓库

点击控制台导航栏【边缘计算】-【镜像管理】，用户若是首次使用“镜像管理”，需要进行初始化重置镜像仓库密码。

用户名：用户名默认是金山云账号，暂不可变更

密码：密码用于通过 docker login 来登陆金山云容器镜像仓库。初始密码为 Kenc123456，客户首次使用时即可重置密码，密码规范是 8-20 个字符，必须包含字母、数字

### 2. 创建项目

项目名是客户创建的私人镜像地址的前缀，用于对客户的容器镜像进行分类。点击【新建项目】，执行创建项目操作。

项目名称：项目名称由小写字符与数字，中划线分隔

访问级别：决定了该项目下镜像的访问级别是公开或是私有

### 3. 推送镜像到镜像仓库

这里我们以北京地域为例：

```
$ sudo docker login --username=[username] registry.kenc.ksyun.com
```

[username] 是您的金山云账号，输入密码后即登陆完成，密码是默认密码或客户在使用镜像仓库时重置的密码。

### 4. 上传镜像

```
$ sudo docker tag [ImageId] registry.kenc.ksyun.com/[项目名]/[ImageName]:[tag] $ sudo docker push registry.kenc.ksyun.com/[项目名]/[ImageName]:[tag]
```

[ImageId] 请根据您的实际镜像 ID 信息进行填写

[tag] 请根据您的镜像版本信息进行填写

[项目名] 是镜像管理中新建的项目

[ImageName] 是在本地镜像的名称

### 5. 下载镜像

```
$ sudo docker pull registry.kenc.ksyun.com/[项目名]/[ImageName]:[tag]
```

[tag] 请根据您的需要下载的镜像版本信息进行填写

根据上述的镜像管理流程操作之后，用户可以将自身业务所需的镜像文件快速推送至边缘节点，做资源的快速创建及生成。

# 应用管理

## 应用概述

应用是基于 Kubernetes 工作负载（Workload）的一种资源组合的表现实体。是 KENC 用于描述业务的运行载体。分为无状态应用（Deployment）、有状态应用（StatefulSet）、任务-暂未开放（Job）、定时任务-暂未开放（CronJob）、守护进程-暂未开放（DaemonSet）。同时其他的 Kubernetes 资源如服务 Service 等通过标签（Tag）app:[appname] 与应用关联起来。形成有机的整体发布到边缘节点对外提供服务。

在发布至边缘节点时，与应用相关联的访问接入（Service、Ingress、EIP）、存储、配置集等会跟随应用一同发布。这样可以有效的简化发布流程，避免资源漏发等问题。

## 应用类型

### 1. 无状态应用

Kubernetes的Deployment，无状态应用支持弹性伸缩与滚动升级，适用于自身基本独立，其不需要在服务端本地保存其服务上下文关系的应用。例如Nginx、memcache等。

### 2. 有状态应用

Kubernetes的StatefulSet，有状态应用支持实例有序部署和删除，支持持久化存储，适用于实例间存在互访，且需要持久化保存服务状态的场景，例如etcd、mysql等。

### 3. 任务

Kubernetes的Job，任务是一次性运行的短任务，部署完成之后即自动运行。一般用于做一些不需对外提供服务的处理操作。

### 4. 定时任务

Kubernetes的CronJob，定时任务是按照指定时间周期运行的短任务。诸如服务间时间同步，定时上传日志等一般用其实现。

### 5. 守护进程集

Kubernetes的DaemonSet，守护进程集确保某些（或全部）节点都运行一个容器组（Pod），适用于每个节点都需要运行一个固定实例的场景，诸如日志收集等。

## 应用与各个资源的关系

### 1. 容器

一个应用由一个或多个容器组（Pod）组成。一个容器组由多个容器实例构成，每个容器组都是通过应用所配置的镜像启动的，在此即可以使用前期上传至镜像仓库中的镜像，或者直接使用公共镜像仓库中的镜像。

### 2. 访问接入

访问接入分为三种类型，服务（Service）、路由（Ingress）、弹性IP（EIP）。对于服务和路由，应用通过app:[appname]标签与其关联，发布应用到边缘集群时，会同时发布访问接入；如果应用绑定弹性IP，容器直接对外暴露公网IP并提供服务。

### 3. 存储

KENC提供的存储主要为容器存储和本地持久化存储卷（LocalPV）的形式，其与应用同样通过标签关联和发布。带有LocalPV的应用发布到边缘节点后，会自动在宿主机创建存储卷，同时每次调度时都会保证应用调度到存储卷所在的宿主机上。

### 4. 配置集和加密字典

与应用同样通过标签关联和发布。一个应用可以关联多个配置集，同样一个配置集也可以关联多个应用。

## 应用发布与下线

### 1. 关联发布

应用发布时除了发布工作负载（Workload）本身外，还会将与应用关联的各种资源同时发布到指定的边缘节点。简化发布步骤，达到一次发布即可上线服务的目的。

### 2. 关联下线

在应用下线时，因考虑到应用关联的配置集等资源有可能会被其他应用所应用。所以应用下线时可以选择是否将相关资源与应用同时下线。

### 3. 模板管理

应用的每一次更改都会生成一个新的应用模板。平台支持对不同的边缘节点发布不同的应用模板，以应对更多的业务场景。但是更推荐的做法是将不同边缘节点上有区别的配置通过配置集的方式挂载到应用中。

## 日志与审计

### 1. 日志

在控制台页面可以通过日志栏查看当前应用启动的容器组的运行日志。另外也可以通过使用KENC的日志服务实现日志搜集和检索。

### 2. 审计

应用的审计主要为控制台操作的审计，会审计包括应用修改、发布、下线等通过控制台操作应用的行为。

# 访问接入

KENC提供三层、四层、七层网络接入方式，并且各节点均部署了经过金山云定制的LVS做三层、七层负载均衡接入。KENC并未使用开源LVS常规的DR/NAT等接入方式，而是使用自研的ENAT做网络接入，足以应对超大广域网流量，单点负载可以跑满10G网卡。下面将针对负载均衡、EIP、路由访问接入方式进行阐述。

## 负载均衡

在KENC平台中，负载均衡的访问接入模式，支持在创建应用时同步选择，也支持单独创建，并与应用进行绑定关联。

### 1. 创建负载均衡模板

在负载均衡功能模块，点击创建负载均衡，填写负载均衡的名称、发布说明、服务配置（负载均衡类型，包含LoadBalancer、ClusterIP，端口信息，选择器），通过选择器实现负载均衡与应用的绑定和关联，提交之后负载均衡模板创建成功。

### 2. 发布负载均衡

在负载均衡详情处，选择负载均衡模板，点击右侧的发布按钮，即可以发布到与容器或应用关联的边缘节点。

### 3. 下线负载均衡

在负载均衡详情处，选择负载均衡模板，点击右侧的下线按钮，勾选需要做下线处理的边缘节点名称，即可实现将该负载均衡在选定的边缘节点下线。

### 4. 删除负载均衡模板

负载均衡模板没有在任何边缘节点发布的状态下，可以删除。直接点击右侧删除按钮即可。

## EIP

为了实现容器的三层网络直通，即将VIP和容器内网IP一一对应，通过自研EIP来实现容器支持弹性IP。EIP和后端容器实例是一一对应的关系，EIP绑定到后端容器，该容器拥有公网IP，可以直接对外提供服务。弹性IP可作为访问接入的方式之一伴随着应用的创建而创建，也可以单独创建之后与应用进行绑定（当前只支持与应用一同创建），其原理一致，具体操作说明如下：

### 1. 创建EIP模板

在EIP功能模块，填写EIP名称、发布说明、弹性IP生成规则以及容器IP的地址（可以选择静态IP或者选择器），静态IP直接映射到后端容器的内网地址，通过选择器可以绑定应用。配置完成，即完成EIP模板的创建。

### 2. 发布EIP

在EIP详情处，选择EIP模板，点击右侧的发布按钮，即可以发布到与容器或应用关联的边缘节点。

### 3. 下线EIP

在EIP详情处，选择EIP模板，点击右侧的下线按钮，勾选需要做下线处理的边缘节点名称，即可实现将该EIP在选定的边缘节点下线。

### 4. 删除EIP模板

可以删除没有在任何边缘节点发布的EIP模板，直接点击右侧删除按钮即可。

## 路由

路由作为访问接入的方式之一，使用的前提是存在一个应用，以及和该应用相对应的、类型为ClusterIP的负载均衡。

### 1. 创建路由模板

在路由功能模块，点击创建路由，填写路由名称、发布说明、服务配置（域名、负载均衡名称及端口）的等信息，域名处填写的域名信息需要公网备案，并将映射到nginx集群的ENAT地址。提交之后，即可完成路由模板的创建。

### 2. 发布路由

在路由详情处，选择路由模板，点击右侧的发布按钮，即可以发布到完成负载均衡配置的应用所关联的边缘节点。

### 3. 下线路由

在路由详情处，选择路由模板，点击右侧的下线按钮，勾选需要做下线处理的边缘节点名称，即可实现将路由在选定的边缘节点下线。

#### 4. 删除路由模板

可以删除没有在任何边缘节点发布的路由模板，直接点击右侧删除按钮即可。

## 存储管理

### 存储概述

存储（volume）主要用于存储 Docker 容器的数据。容器服务可分为有状态和无状态服务。无状态的容器服务自身不会持久化任何数据，这意味着容器重启时，上一次运行时的数据都将丢失；有状态容器服务可通过挂载本地持久化存储卷，将数据文件保存在存储卷中，以实现数据的存储、备份或共享。

平台提供动态创建本地持久化存储卷（LocalPersistentVolume）的功能，可以将容器的数据持久化到本地的磁盘、目录中。基于kubernetes的Local存储，和远端存储类似，此时数据依然独立于 Pod 的生命周期，即使业务 Pod 被删除，数据也不会丢失。同时，和远端存储相比，本地存储可以避免网络 IO 开销，拥有更高的读写性能，所以分布式文件系统和分布式数据库这类对 IO 要求很高的应用非常适合本地存储。

#### 1. 创建存储

无状态存储会随着容器的生成而自动创建，也会伴随容器的下线而下线。此处讲解的创建流程均针对本地持久化存储。本地持久化存储可以随应用的创建而创建，如果在创建应用时，不存在存储卷，则需要先创建存储卷，对本地持久化存储卷的名称、容量进行限定，创建之后即不可变更。创建完成后，添加本地存储，将本地持久化存储卷与应用进行关联。

本地持久化存储也可以再存储管理模块单独创建，但是单独创建的本地持久化存储卷与应用没有关联关系，需要在应用管理处，针对需要关联的应用进行克隆同时添加本地存储卷，将本地持久化存储卷绑定在应用中；也可以在yaml文件通过标签关联应用和本地持久化存储卷。创建本地持久化存储卷即意味着在k8s为用户创建存储卷声明PVC，将本地持久化存储卷与应用关联时，相当于LocalPV完成创建。

#### 2. 释放空间

当与本地持久化存储卷关联的应用处于发布状态时，本地持久化存储卷不允许释放空间；当应用下线时，如果将本地持久化存储卷同步下线，存储空间即会默认释放；如果未将本地持久化存储卷同步下线，则用户可以主动选择将本地持久化存储卷做释放空间处理。空间一旦释放，数据将丢失且不可恢复。

#### 3. 解除关联

存储空间已经释放的本地持久化存储卷，相当于其PV被释放，但PVC模板仍保存，且保持与应用的关系。在此处，如果该模板需要在其他应用上使用，或者对应的应用需要更换存储卷，可以点击解除关联，将应用与PVC模板的耦合关系。

#### 4. 删除

如果存储空间已经释放，不管存储是否与应用存在关联关系，均可以选择将其删除，意为将PVC模板删除。

#### 5. 使用限制

一个PVC可以被一个Pod的多个容器挂载，但只有与当前PVC在同一个namespace下的Pod才能使用它，即PVC不能跨命名空间调用。

## 集群管理

集群管理模块的设计，旨在为用户开放边缘节点的管理能力，目前该模块的功能陆续开发中，当前支持用户选择节点导出 kubeconfig 文件，便于用户使用与kubernetes保持一致性的API接口，在选定的边缘节点上做资源开通、删除、变更、查询等操作。