目录

镜像（Image）

镜像是一个模板，是容器应用打包的标准格式，用于创建容器。或者说：Docker镜像是一个特殊的文件系统，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、用户等）。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。

容器（Container）

一个通过镜像创建的运行实例，一个节点可以运行若干个容器，容器的实例是进程，但与直接在宿主执行的进程不同，容器进程运行于属于自己的独立命名空间。镜像（Image）和容器（Container）的关系，就像是面向对象程序设计中的“类”和“实例”一样，镜像是静态的定义，容器是镜像运行时的实体。容器可以被创建、启动、停止、删除、暂停等。

命名空间（Namespace）

KENC的命名空间与管辖的K8S集群一一对应。Namespace是对一组资源和对象的抽象整合。在同一个集群内可以创建不同的命名空间，不同命名空间中数据彼此隔离，使得他们即可以共享同一个集群的服务，也能够互不干扰。

应用（App）

应用是基于 Kubernetes 工作负载（Workload）的一种资源组合的表现实体。分为无状态应用（Deployment，部署）、有状态应用（StatefulSet，状态副本集）、任务-暂未开放（Job，任务）、定时任务-暂未开放（CronJob，定时任务）、守护进程-暂未开放（DaemonSet，守护进程）。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。

应用模板（App Template）

在创建应用时，可以选用Deployment、StatefulSet、Job、CronJob、DeamonSet等不同工作负载，在应用发布前，可以通过应用模板确定工作负载，并对容器规格、配置信息等进行定义。应用模板创建完成之后，即可发布。用户可以根据应用模板进行发布、下线、克隆、重启、查看模板等操作。

容器组（Pod）

Pod是Kubernetes部署应用或服务的最小基本单位。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。

服务（Service）

与Kubernetes的service对应，Service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，这给扩展和维护后端带来很大好处。

路由（Ingress）

路由是用来聚合集群内服务的方式，对应的是Kubernetes的Ingress资源，后端使用了Nginx Controller来处理具体规则。Ingress可以给service提供集群外部访问的URL、负载均衡、SSL termination、HTTP路由等。

配置集（ConfigMap）

存放应用的配置信息，与Kubernetes的Config资源对应。

加密字典（Secret）

加密字典用于存放一些敏感配置，如密码、证书等信息，与Kubernetes的Secret资源对应。

弹性IP（EIP）

弹性IP功能可以满足容器三层网络直通，即公网IP和容器一一映射，相当于将公网IP绑定在容器上，实现类似于云主机的弹性IP功能。

本地存储卷（LocalPersistentVolume）

KENC针对容器本地存储进行优化配置，使得Pod可以使用集群内的存储资源，使用Local PV的Pod，总会被调度到同一个节点上（否则就调度失败）。这样可以使容器重启或后存储卷中的数据不丢失。