

K8S

基础概念

- 源于google内部的borg，提供了面向应用的容器集群部署和管理系统。k8s的目标旨在消除编排物理/虚拟运算，网络和存储基础设施的符单，并使应用程序运营商和开发人员完全将重点放在以容器为中心的原语上进行自助运营。（具备完善的集群管理能力）
- 一个全新的基于容器技术的分布式架构方案,在Docker的基础上,为容器化的应用提供了部署运行，资源调度，服务发现和动态伸缩等一系列的完整功能，提高了大规模容器集群管理的便捷性。
- kubernetes主要有以下几个核心组件组成：
 - etcd保存了整个集群的状态
 - kube-apiserver提供了资源操作的唯一入口，并提供认证、授权、访问控制、api注册和发现等机制
 - kube-controller-manager负责维护集群的状态，比如故障检测、自动扩展、滚动更新等
 - kube-scheduler负责资源的调度，按照预定的调度策略将pod调度到响应的机器上；
 - kubelet负责维持容器的生命周期，同时也负责volume（CVI）和网络（CNI）的管理
 - Container runtime负责镜像管理以及pod和容器的真正运行（CRI），默认的容器运行时为docker
 - kube-proxy负责为service提供cluster内部的服务发现和负载均衡(代理节点)

架构

Master

k8s集群的管理节点，负责管理集群，提供集群的资源数据访问入口。拥有Etcd存储服务（可选），运行Api Server进程，Controller Manager服务进程及Scheduler服务进程，关联工作节点Node。Kubernetes API server提供HTTP Rest接口的关键服务进程，是Kubernetes里所有资源的增、删、改、查等操作的唯一入口。也是集群控制的入口进程；Kubernetes Controller Manager是Kubernetes所有资源对象的自动化控制中心；Kubernetes Scheduler是负责资源调度（Pod调度）的进程

Node

- Node作为集群中的工作节点，运行真正的应用程序，在Node上k8s管理的最小运行单位是Pod。
- Node上运行着k8s的Kubelet,kube-proxy服务进程，这些服务进程负责Pod的创建，启动，监控，重启，销毁，以及实现软件模式的负载均衡。
- Node包含的信息：
 - Node的地址:主机的IP地址，或者NodeID
 - Node的运行状态:Pending,Running,Terminated三种状态
 - Node Condition
 - Node系统容量：描述Node可用的系统资源，包括CPU、内存、最大可调度Pod数量等。
 - 其他：内核版本号、Kubernetes版本等。
- 查看Node信息：`kubectl describe node`

核心概念

Pod

- Pod是Kubernetes最基本的操作单元，包含一个或多个紧密相关的容器，一个Pod可以被一个容器化的环境看作应用层的“逻辑宿主机”；一个Pod中的多个容器应用通常是紧密耦合的，Pod在Node上被创建、启动或者销毁；每个Pod里运行着一个特殊的被称之为Pause的容器，其他容器则为业务容器，这些业务容器共享Pause容器的网络栈和Volume挂载卷，因此他们之间通信和数据交换更为高效，在设计时我们可以充分利用这一特性将一组密切相关的服务进程放入同一个Pod中。同一个Pod里的容器之间仅需通过localhost就能互相通信。
- PID命名空间：Pod中的不同应用程序可以看到其他应用程序的进程ID；
- 网络命名空间：Pod中的多个容器能够访问同一个IP和端口范围；
- IPC命名空间：Pod中的多个容器能够使用SystemV IPC或POSIX消息队列进行通信；
- UTS命名空间：Pod中的多个容器共享一个主机名；

- Volumes（共享存储卷）：Pod中的各个容器可以访问在Pod级别定义的Volumes；

核心概念 - Pod

Volume

核心概念 - Volume

Deployment

核心概念 - Deployment

Service

Namespaces

API

Job

DaemonSet

网络

StatefulSet

Ingress

ingress本质上是一个负载均衡服务，他被用来将集群中的用户服务通过统一的IP地址暴露给外界，它的实现可以基于多种不同的具体负载均衡工具，统一IP的好处在于，用户可以为该地址绑定域名，从而通过自定义的域名直接访问集群中内部署的服务。

运维

批量删除Evicted状态的pod

```
kubectl -n sky get pods | grep Evicted | awk '{print$1}' | xargs kubectl -n sky delete pods
```