Kubernetes架构介绍

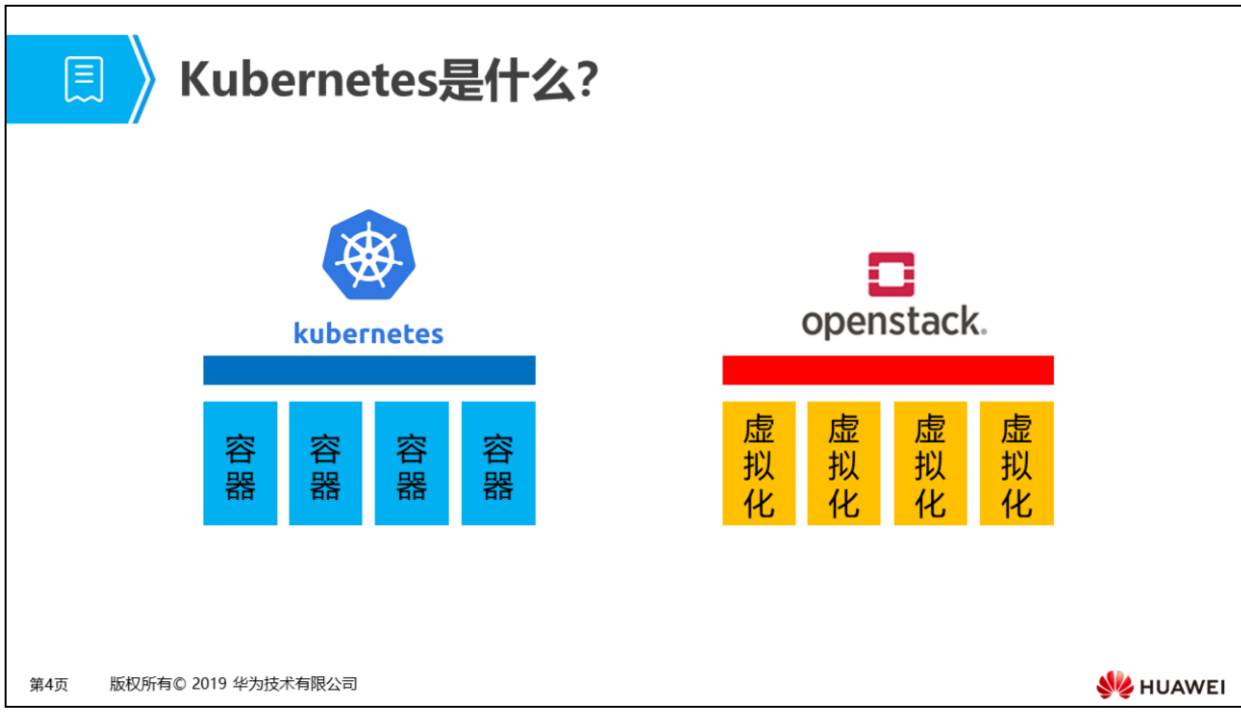
前言

- 本章节主要讲述了Kubernetes的概念、架构、组件。介绍了namespace的概念和使用。

目标

- 学完本课程后，您将能够：
  - 描述Kubernetes架构
  - 区分不同组件的功能
  - 查看master和node节点中的组件
  - 创建namespace

目录

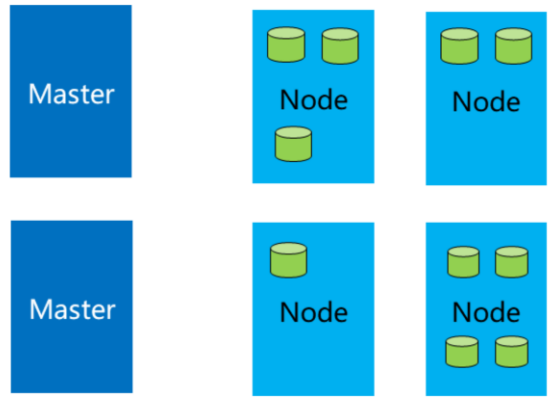


- Kubernetes是一个跨主机集群的开源容器调度平台。他可以管理各种不同的底层容器。与Openstack类似，OpenStack通过管理底层大量的虚拟化节点构成一个庞大的集群，提供云服务。而Kubernetes通过管理底层的容器节点，自动化应用容器的部署、扩展和操作，提供以容器为中心的基础架构。

Kubernetes架构

- 一个基础的Kubernetes集群包含一个或多个master节点和多个node节点。每个节点可以是一台物理机，也可以是一台虚拟机。

Master节点提供的集群控制，对集群做出全局性决策，例如调度等。通常在master节点上不运行用户容器。

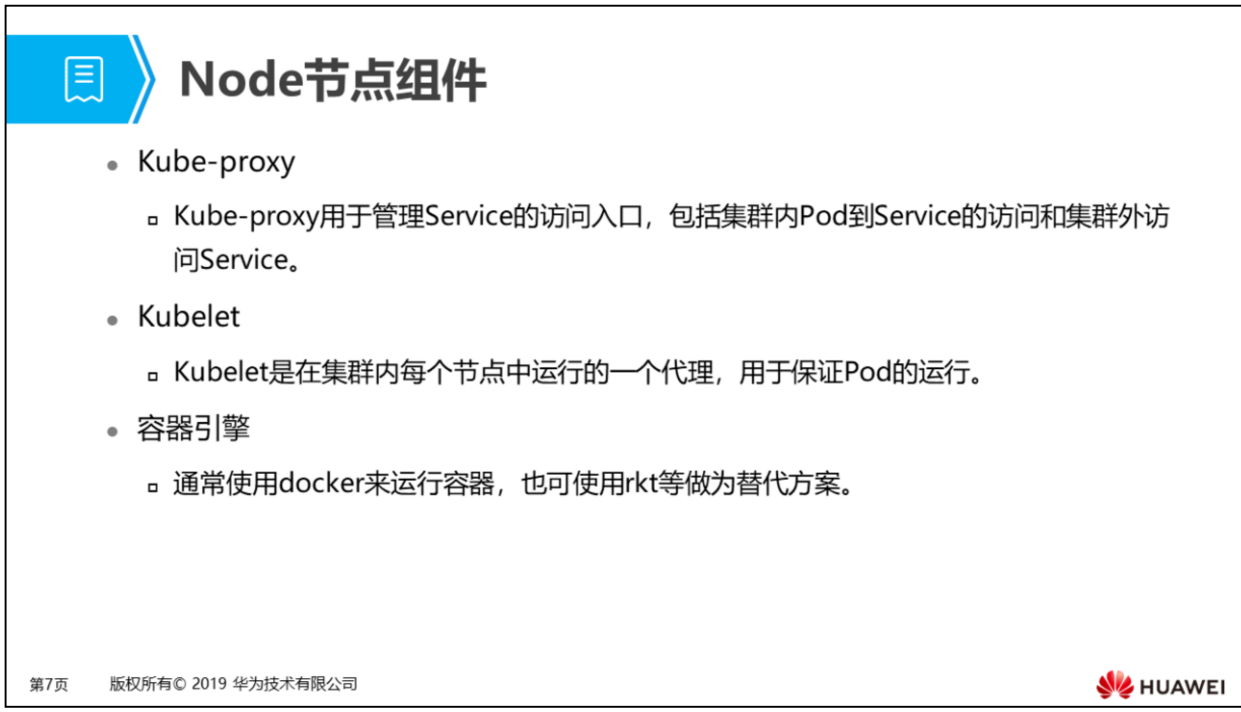


节点组件运行在每一个Node节点上。维护运行的pod并提供Kubernetes运行时环境。

Master节点组件

- Kube-apiserver
  - kube-apiserver对外暴露了Kubernetes API。它是Kubernetes的前端控制层。它被设计为水平扩展，即通过部署更多实例来缩放。
- Etcd
  - etcd用于Kubernetes的后端存储。所有集群数据都存储在此处，始终为Kubernetes集群的etcd数据提供备份计划。
- kube-controller-manager
  - 运行控制器，它们是处理集群中常规任务的后台线程。逻辑上，每个控制器是一个单独的进程，但为了降低复杂性，它们都被编译成独立的可执行文件，并在单个进程中运行。
- kube-scheduler
  - 监视没有分配节点的新创建的Pod，选择一个节点供他们运行。

- 可以使用命令行查看Master节点上的组件。



- Kubelet的功能
  - 挂载Pod所需要的数据卷(Volume)。
  - 下载Pod的secrets。
  - 通过Docker(或通过rkt)运行Pod中的容器。
  - 周期性的对容器生命周期进行探测。
  - 如果需要，通过创建镜像Pod（Mirror Pod）将Pod的状态报告回系统的其余部分。
  - 将节点的状态报告回系统的其余部分。

推荐Add-ons

- 除了上述组件外，Kubernetes使用中通常需要一些额外的组件实现特定功能，常用的Add-ons包括：
  - Core-dns：为整个集群提供DNS服务
  - Ingress Controller：为Service提供外网访问入口
  - Dashboard：提供图形化管理界面
  - Heapster：提供集群资源监控
  - Flannel：为Kubernetes提供方便的网络服务

- 本课程涉及到的插件包括EFK等。

Kubeadm

- Kubeadm是社区主推的快速创建Kubernetes集群的工具。
- kubeadm通过执行必要的操作来启动和运行一个最小可用的集群。它被故意设计为只关心启动集群，而不是之前的节点准备工作。同样的，诸如安装各种各样值得拥有的插件，例如Kubernetes Dashboard、监控解决方案以及特定云提供商的插件，这些都不在它负责的范围。
  - Master节点：kubeadm init，快速初始化安装主节点组件
  - Node节点：kubeadm join，将从节点加入集群

查看组件运行状态

- 使用systemctl status指令查看组件运行状态
  - Docker:

```
[root@k8s-node1 runfile]# systemctl status docker
   docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service;
enabled; vendor preset: disabled)
   Active: active (running) since Sun 2019-04-28 18:00:36
CST; 2 weeks 3 days ago
```

  - kubelet:

```
[root@k8s-node1 runfile]# systemctl status kubelet
   kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service;
enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/kubelet.service.d
           └─10-kubeadm.conf
   Active: active (running) since Sun 2019-04-28 17:15:03
CST; 2 weeks 3 days ago
```


- 为什么kube-proxy等组件的状态无法看到呢?

Kubeadm容器化组件

- Kubeadm为了实现部署的便捷性，将一些组件封装到了Pod中。
  - Master节点

```
[root@k8s-master]# kubectl get pods --field-selector spec.nodeName=k8s-master
--namespace=kube-system
NAME                                 READY   STATUS    RESTARTS   AGE
coredns-fb8b8dccf-j4kn5              1/1     Running   3          18d
coredns-fb8b8dccf-mx6pw              1/1     Running   3          18d
etcd-k8s-master                      1/1     Running   2          18d
kube-apiserver-k8s-master            1/1     Running   3          18d
kube-controller-manager-k8s-master   1/1     Running   2          18d
kube-flannel-ds-amd64-lfslh          1/1     Running   3          18d
kube-proxy-dt69s                     1/1     Running   2          18d
kube-scheduler-k8s-master            1/1     Running   2          18d
kubernetes-dashboard-5f798cc594      1/1     Running   3          18d
```


- Master节点有5个Pod。

Kubeadm容器化组件

- 查看Node节点的所包含的系统Pod

```
[root@k8s-node1 runfile]# kubectl get pods --field-selector
spec.nodeName=k8s-node2 --namespace=kube-system
NAME                          READY   STATUS    RESTARTS   AGE
kube-flannel-ds-amd64-7tn6g   1/1     Running   2          18d
kube-proxy-q8xmm              1/1     Running   2          18d
```

  - 可以看到在Node节点中支撑Kubernetes系统的必须组件较少。

- 在Node节点上是不是只有kube-proxy一个系统组件?

目录

HUAWEI

命名空间 - namespace

- Kubernetes支持多个虚拟集群，它们底层依赖于同一个物理集群。这些虚拟集群被称为命名空间。
- 命名空间提供了良好的资源隔离，可以用于区分不同的项目、用户等。如开发测试使用的namespace，或者生产使用的namespace。
- 使用如下命令可以查看哪些对象在命名空间中：

```
kubectl api-resources --namespaced=true
```

  HUAWEI

- 命名空间是在多个用户之间划分集群资源的一种方法（通过资源配额）。

常用命名空间命令

- 查看存在哪些namespace。

```
kubectl get namespace
```

- 对指定命名空间进行操作，如创建Pod，查看Pod等。以下是我们之前使用过的查看系统域中Pod的命令：

```
kubectl get pod -namespace=kube-system
```

- 以上命令也可使用简写的方式：

```
kubectl get pod -n kube-system
```

  HUAWEI

- 常见namespace
  - Kube-system：Kubernetes系统创建对象所使用的命名空间。
  - Default：没有指明使用其他命名空间的对象所使用的默认命名空间。

实验任务

- 实验任务
  - 请按照实验手册2.2部分完成Kubernetes组件实验。

本章总结

- 本章节主要讲述了Kubernetes的概念、架构、组件。
- 理解namespace的概念和如何使用。


谢 谢
www.huawei.com