Part1 搭建Kubernete容器集群管理系统

1. Kubernetes和相关组件介绍
2. 实验环境：

· 三台全新的centos7.5

1. k8s概述
2. Kubernetes是Google开源的容器集群管理系统，基于Docker构建一个容器的调度服务，提供资源调度，均衡容灾，服务注册，动态扩缩等功能套件。是一个基于容器的云平台
3. Kubernetes是基于docker容器的云平台，简写为k8s
4. Openstack是基于kvm虚拟机的云平台
5. k8s常见组件介绍
6. Master：k8s管理节点
7. API Server：提供接口服务，用户通过apiserver来管理整个容器集群平台。API Server负责和etcd交互(其他组件不会直接操作etcd，只有API Server这么做)，在永恒个kubernets集群的所有的交互都是以API Server为核心的。如：1、所有对集群进行的查询和管理都要通过API来进行;2、所有模块之间并不会相互调用，而是通过和API Server打交道来完成自己那部分的工作;3、API Server提供的验证和授权保证了整个集群的安全
8. scheduler k8s调度服务
9. Replication Controllers复制，保证pod的高可用

- Replication Controller是Kuberbetes系统中最有用的功能，实现复制多个Pod副本，往往一个应用需要多个Pod来支撑，并且可以保证其他复制的副本数，机师副本所调度分配的宿主机出现异常，通过Replication Controller可以保证在其他宿主机启动同等数量的Pod。Replication Controller可以通过rep模板来创建多个Pod副本，同样也可以直接复制已存在的Pod，需要通过Label selector来关联。

1. minion：真正运行容器container的物理机。k8s中需要很多minion机器来提供运算
2. container：容器，用来运行服务和程序
3. Pod：在k8s系统中，调度的最小颗粒不是单纯的容器，而是抽象成一个Pod，Pod是一个可以被创建、销毁、调度、管理的最小部署单元。pod中可以包括一个或一组容器
4. Kube\_proxy：代理，做端口转发，相当于LVS-NAT模式中的负载调度器

- proxy解决了同一宿主机，相同服务端口冲突的问题，还提供了对外服务的能力，proxy后端使用了随机，轮询负载均衡算法

1. etcd：etcd存储k8s的配置信息，可以理解为k8s的数据库，存储着k8s容器云平台中所有节点、pods、网络等信息
2. Services：Services是k8s最外围的单元，通过虚拟一个访问IP及服务端口，可以访问我们定义好的pod资源，目前的版本是通过iptables的nat转发来实现，转发的目标端口为Kube\_proxy生成的随机端口
3. Labels标签

- Labels是用于区分Pod、Service、Replication Controller的key/value键值对，仅使用在Pod、Service、Replication Controller之间的关系识别，但对这些单元本身进行的操作时得使用name标签

1. Deployment
2. 配置yum和安装k8s及组件
3. 配置etcd和master节点
4. 配置minion1节点
5. 配置minion2节点并总结k8s所有服务和端口号