Day03 kubernetes笔记

课程主要内容：

1、pod核心原理（底层组成结构，工作机制）

2、副本控制器（replicationController,ReplicaSet）

3、Deployment

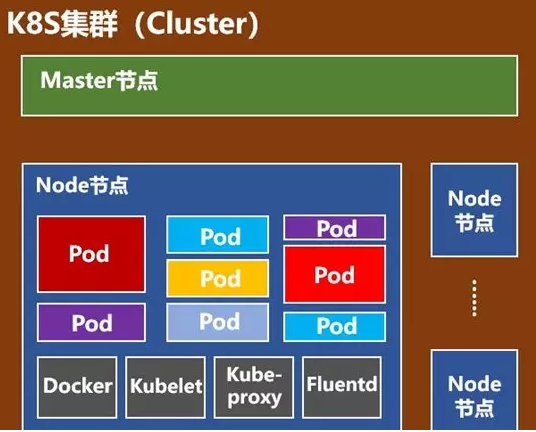
4、StatefulSet

5、DeamonSet

6、Volume

7、label

# 1、课程回顾（k8s架构）



Kubernetes集群架构（高可用集群：多个master,每一个对应一群node节点）

1、一个master对应一群node节点，node节点负责存储pod

2、node内部可以有一个pod,或者是多个pod，具体情况看pod所消耗的资源（cpu,内存），同时也必须看硬件资源

3、pod是k8s操作最小单元，pod内部运行的是一个，或者多个容器。

问题1： k8s是什么？？？

K8s是一个容器编排（管理）工具，但是k8s不直接管理docker容器，而是间接通过管理pod的管理（编排）容器。

# 2、Pod核心原理

## 2.1、Pod是什么？

Pod的特点：容器

1、有自己的IP地址

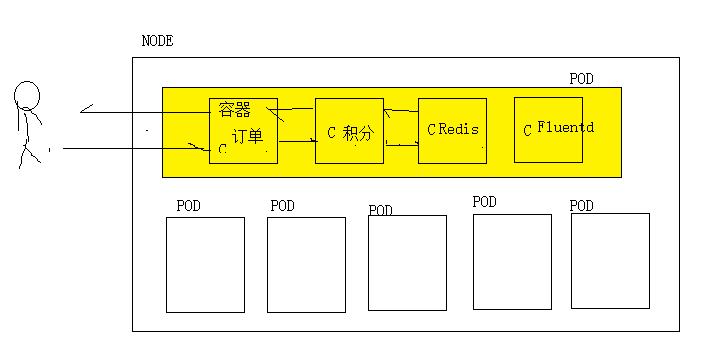
2、有自己的hostname

Pod实际上也相当于是一个独立的容器（虚拟机器），而这个pod容器内部封装的是由docker引擎所创建的容器，可以理解为pod就是一个虚拟化分组，pod内存可以存储一个或者多个容器。

## 2.2、Pod作用

Pod内部封装的是容器，容器内部运行是开发的应用程序。Pod管理上线的运行的应用程序

定义：在通常情况下，在服务上线部署的时候，pod通常被用来部署一组相关的服务。（什么一组相关的服务？）



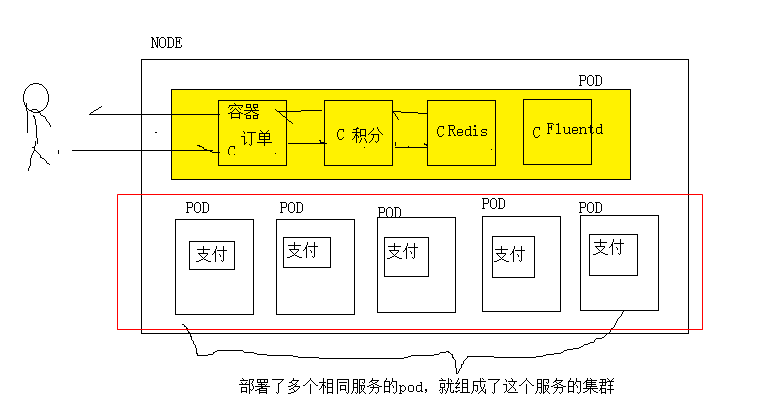
一组相关服务： 在一个请求链路上的服务，叫做一组相关服务。通常情况下，这一组相关的服务在调用链路上处于上下游的关系。

经验来说：（可维护的角度）

建议一个pod内部只允许部署一个容器（一个服务）。

## 2.3、服务集群

问题：服务集群该如何部署？？？？（pod内部运行就是容器，容器中运行的就是服务）



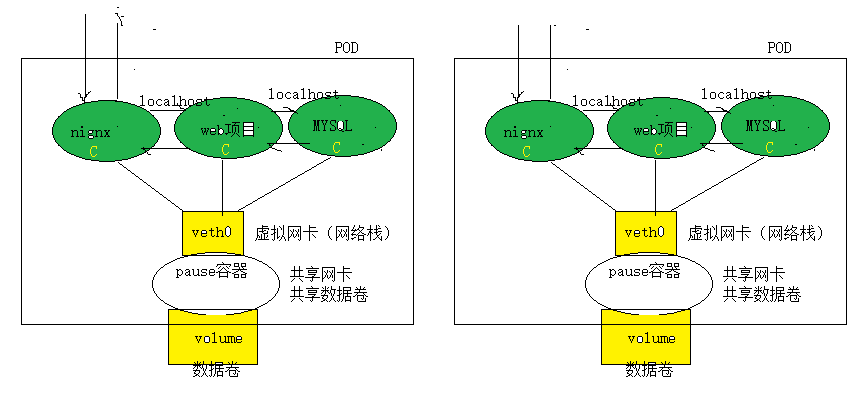
使用kubernetes构建服务集群，只需要创建多个部署了相同服务的pod即可。也就是说k8s可以实现一键式部署，一键式扩容

## 2.4、Pod原理

Pod底层核心原理：

1、数据存储

2、网络



在使用了pause容器创建共享网卡后，pod内部容器之间访问就使用localhost访问，就相当于本地访问一样，因此性能非常高。

注意： 一个pod不能分裂存储在多个node节点，但是多个pod副本可以随机分配到不同的node节点进行存储。

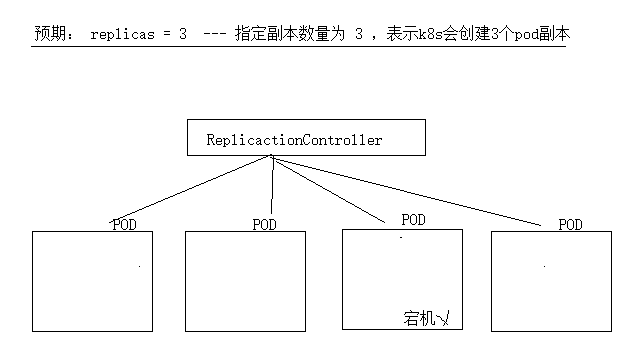
# 3、k8s核心组件

## 3.1、组件一RC

Kubernetes核心组件原理：

RC (ReplicationController) ---- 副本控制器

定义：用来确保容器应用的副本数始终保持在用户定义的副本数，即如果有容器异常退出，会自动创建新的Pod来替代，而如果异常多出的容器也会自动回收。



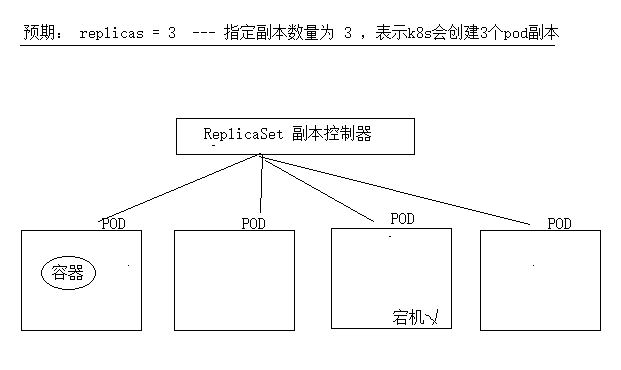
副本控制器作用：永远保证服务数量和预期的数量保持一致。也就是说服务永远不会宕机，服务永不停机，服务永远处于高可用状态。

如果pod全部宕机：全部重新创建

注意：在新版本的 Kubernetes 中建议使用ReplicaSet 来取代ReplicationController

## 3.2、组件二RS

控制器和pod控制架构---副本控制器是如何控制pod副本的？？



RS 和 RC 副本控制器到底有和区别？？ 为什么要是有RS，不使用RC呢？？

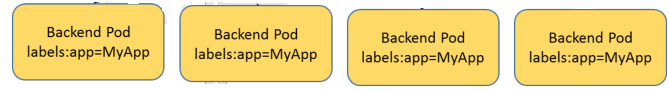
1、RC只支持单个标签的选择器

2、RS不仅支持单个标签选择器，还支持复合选择器。

Labels： 标签，是用来唯一标识一个组件的，k8s中所有的资源对象都可以被标签类进行标识。通过标签标识后，方便，灵活的查询到这个组件。

查询一组相同业务pod:

app = MyAPP 此标签可以查询出下面4个pod.这4个pod是一组相关副本的pod. 标签相同，说明是同一个服务部署了多个副本。



标签选择器：

根据标签 key=value 查询与之对应的匹配的资源对象（pod，rs, deployment……）

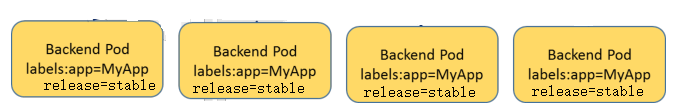
标签选择器 – 标签 – 选择资源对象 （唯一定位一个资源对象）

复合选择器：支持多个标签选择器

Selector:

app = MyApp

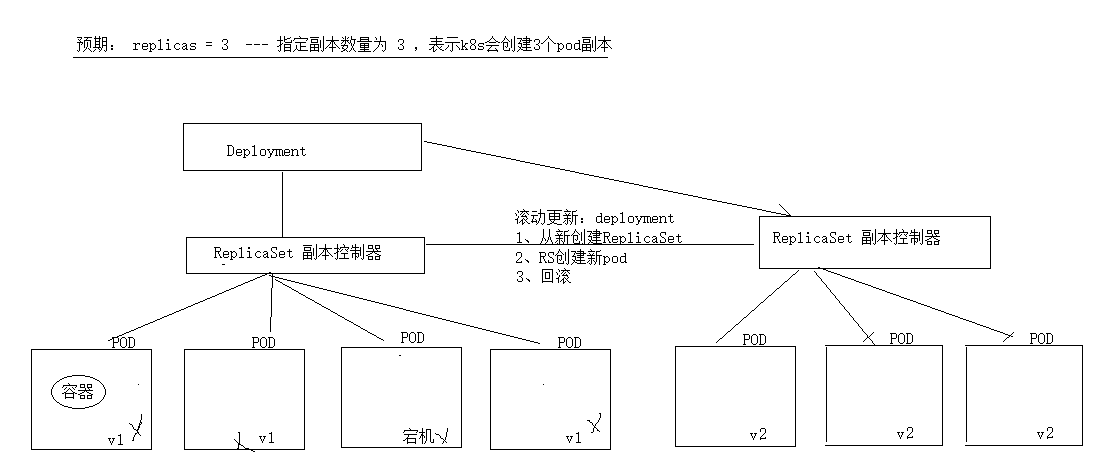
release = stable



注意： 如果单独使用rs，就不支持滚动更新。（在企业中，产品经理不停提出新的需求，项目版本不停的迭代…不停机更新？？）

## 3.3、组件三Deployment

Deployment和rs结合使用，支持滚动更新。（发新版，不停机更新）



资源对象组成关系：

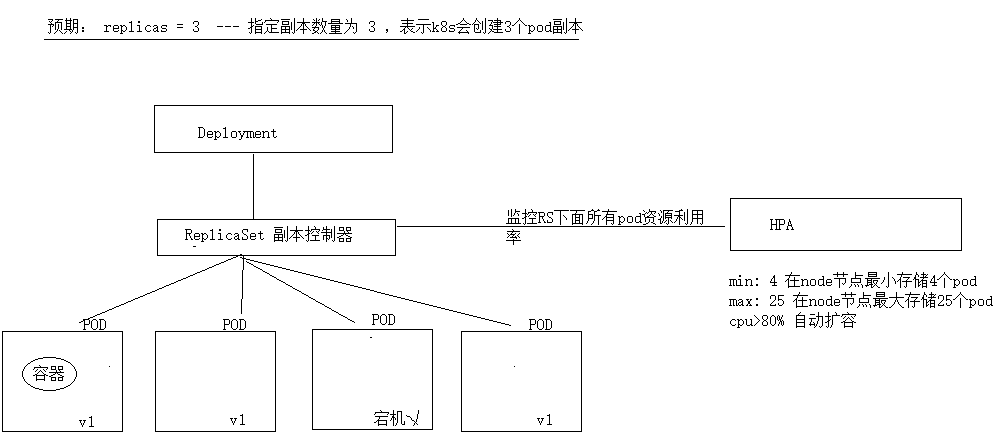
1、deployment部署资源对象，管理rs ,支持滚动更新，更新模式就是新创建一个新的rs,然后由rs创建新的版本的pod.

2、rs副本控制器，控制pod的数量与预期的数量保持一致，rs副本控制器就是来管理pod.

注意：rs通过标签选择器选择受rs所管理的资源对象。

## 3.4、组件四HPA

HPA 自动更新 （k8s会根据CPU利用率来进行扩缩容），如果cpu利用率超过80%,此时就会对pod进行扩容，如果cpu利用率低于50%,对pod进行缩容



## 3.5、组件五StatefullSet

Deployment ,StatefulSet都是用来部署服务的。

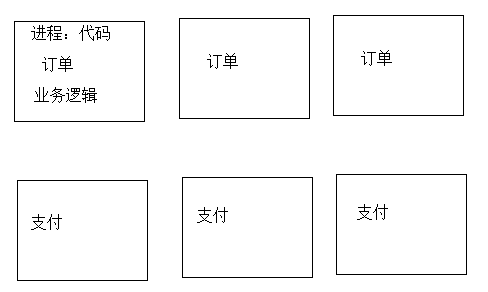
1、但是deployment被用来部署无状态服务，不适合部署有状态服务。

2、statefulset是用来部署有状态服务的。

无状态：

1、没有实时的数据需要存储

2、在服务集群网络中，把其中一个服务抽离出去，过一段时间在加入进来，对服务集群没有任何影响。



有状态：

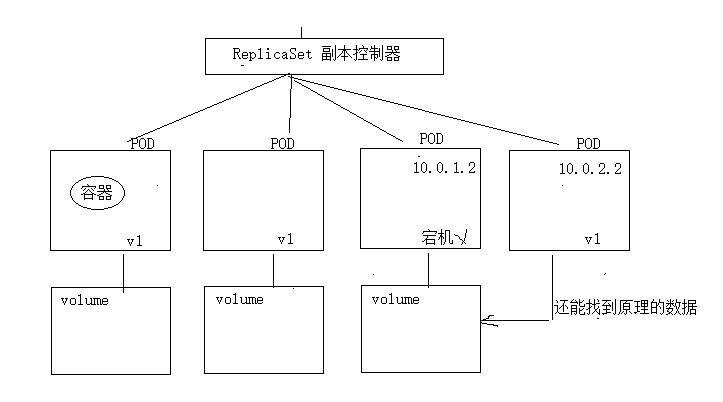
1、有实时数据需要存储

2、在服务集群网络中，把其中一个服务抽离出去，过一段时间在加入进来，对服务集群有影响。

ES/MYSQL/Mongodb/……………

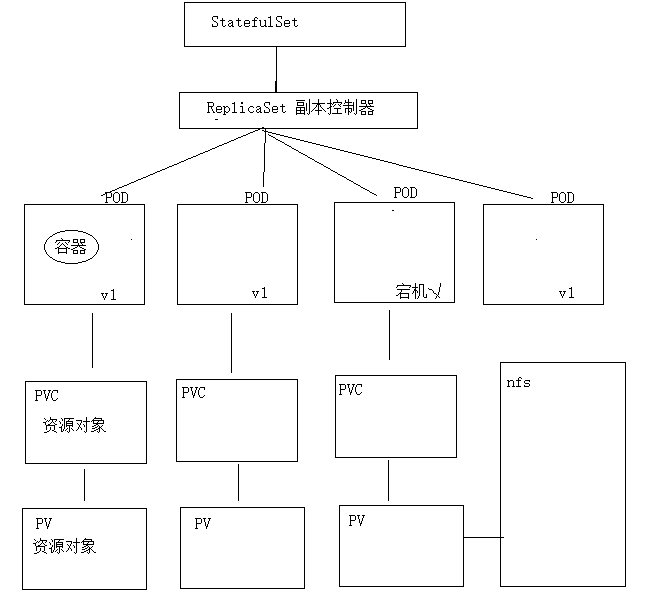
为什么pod方式，必须使用statefulSet来进行部署？？？？

前提： 副本创建，滚动更新，扩容 ，都是创建一个新的pod。当一个pod宕机后，新创建一个新的pod.因此这个新的pod如何共享之前数据，如果不做任何处理，此时这个新的pod就会丢失之前的数据，对于有状态服务来说，就是灾难。



Statefulset部署有状态服务网络拓扑结构：

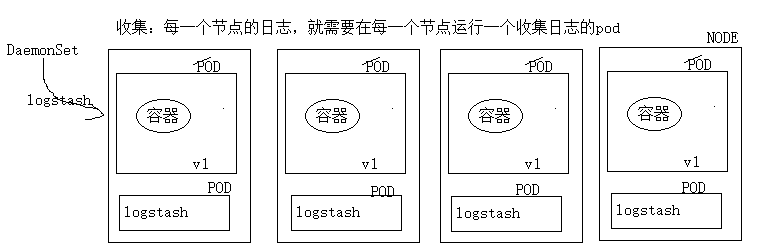
‘



Statefulset保证数据不丢失根据pod的名称来挂载数据的，也就是说可以让pod的名称永远保持不变，在pod宕机，重新创建后，根据名称找回数据。

## 3.6、组件六DaemonSet

Daemonset就是让每一个node节点都运行一个相同的pod.



# 下节课（最难：网络）

K8s服务发现

负载均衡（service）