

**K8的Service**是标准的微服务 --- 刚才的点都符合 -----



**---- 可以把Pod理解为Docker容器**

**部署在不同的节点上** ----- 这样 **每一个Service都有一个虚拟的IP地址 和 虚拟的端口号**

【这个Service就是图中的】 **这个虚拟的IP称为Cluster IP**

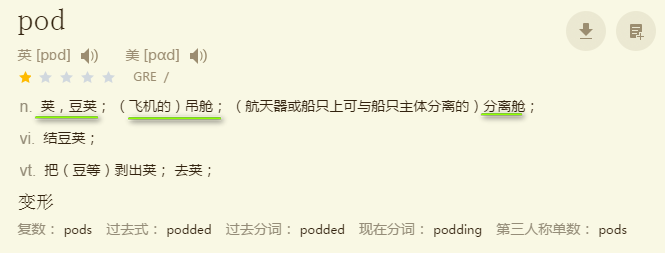
比如容器的端口是80 但是 Service的Cluster IP是8080 **并且具备负载均衡的分发能力**

【**从图上看 POD的功能必须相同 否则 就没有办法进行负载均衡了**】

----- 这样收到Client的请求之后 通过Cluster IP 和Port 分发到后面的Port上面

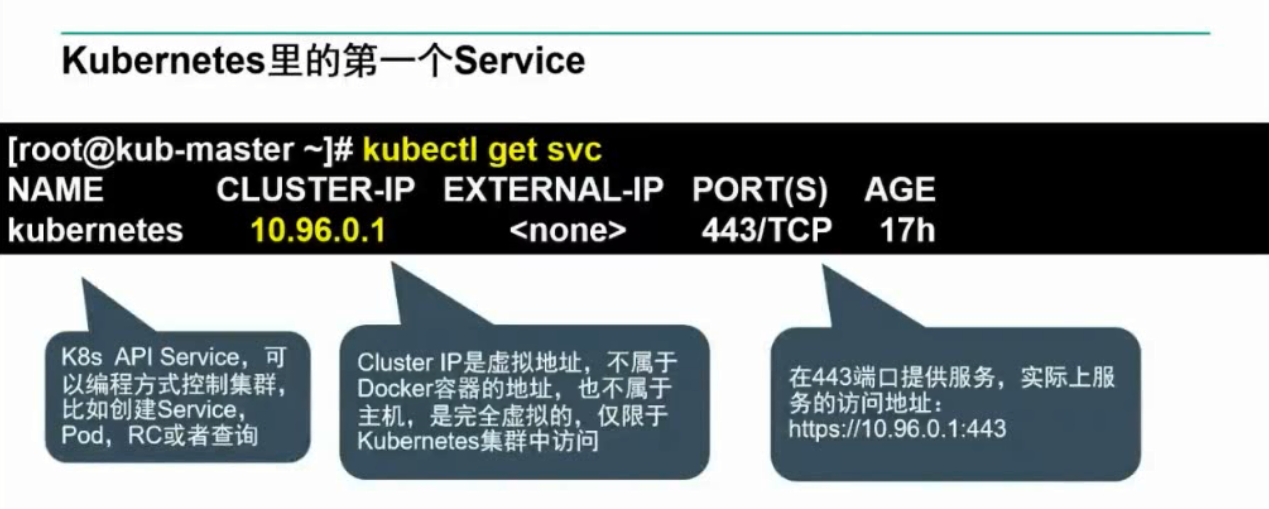
**===== 这样客户端只要访问这个Service的Cluster IP和Port 就能访问这个服务 不管Service的后面有几个实例 任何变化都不会影响 【前面说过 这个ClusterIP 和Port是不变的】 ---- 这个就是kubernetes service**

**===== 建模上 我们的系统哪些组件建模是Service 这个Service需要多少Pod副本 能否自动迁移 服务的数据存储在哪里 配置如何提供 这就是用好kubernetes建模的关键**



----- 把第一个Hello World的例子做好了 看这样的一个问题

**Kubectl get svc 列出 Kubernetes所有的服务列表**

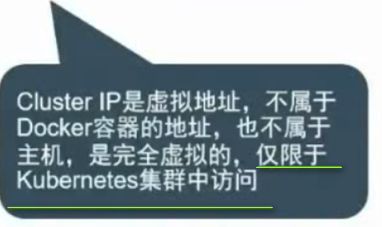


使用Kubectl get svc ---- NAME 是Kubernetes Cluster-IP是 这个IP既不是宿主机的IP 也不是Docker的网段 是第三重网路段 ---- 一旦分配之后 就不变了

并且主机是ping不通

----- 可以指定这个Cluster-IP是哪一个地址段 只要和虚拟机 Docker网段不通就好

----- 这个是.1 就是第一个 ----- 就是把这个网段的第一个IP给Service

---- 这个是在443端口上做的 --- 也就是一个https访问 完全虚拟的 



现在监控某一个Service的Pod

**我们通过kubernetes service可以操作Kubernetes本身 ------**

为什么提供者服务 --- 就是让你有机会调用这个服务 进行编程

---- 现在监控某一个Service的Pod 如果这个ServicePod发生了变化 我就做一些其他的操作

---- 这时候创建自己的Pod Pod里面调用Kubernetes的Service ---- 然后查Kubernetes的API进行监控 ---非常重要 ==== 让你在容器内部调用这个服务 进行监控

试试下面的这个命令

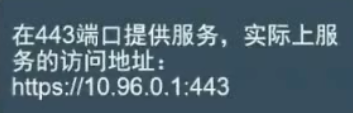


**Kubectl delete svc <SERVICE-NAME>** 然后 再看看 这个Service有没有被delete掉？

看是否成功 是否真的删除？【看QQ】

这个Service不能被删除了 但是 命令提示是删掉了 ---- 这个不是bug 很多第三方插件都依赖这个kubernates service 去做一些操作 [指的是 我的这个service的名字 就是kubernetes]

AGE没有问题 的确是重新创建了一个

访问的是  这个只能在Kubernetes集群中访问 但是在主机上是访问不了的

============ 

**Kubectl describe svc <SERVICE-NAME>**



描述信息

Kubernetes的endpoints  的ip就是宿主机的ip 这个6443对应的是什么进程？ Netstat –tlnp | grep 6443

进程的名称是什么？ 需要懂Linux 【QQ中 竟然没有人答对 老师说百度一下也是可以的】

**6443到底是被那个进程占据？**

----- 这个端口绑定的kubernetes的api server

【  我猜测是这个】

--- 是kubernetes中最最重要的一个独立的组件 独立的进程 那么 通过6443可以访问他的 ---- 实际上 你是通过kubectl命令也是可以访问他的 --- 日志中 都要链接6443 大家看看   
 **这样说明一个什么问题？ 就是kubernetes service对应的后端有什么特点？**

为了对比这个 所有装好系统的 你们都装了自己的service 比如 有创建的nginx service

--- 看看nginx service 的endpoint是什么样子的 最后对比kubernetes service的endpoints ------ 就是Kubernetes service 谁能发现 并且总结一下？ 对比 这两个endpoints的不同？【这块是老师在启发大家怎么发现问题 ---- 用心良苦】

有人发现了特点 但是没有给出答案

---- Kubernetes的endpoint 对应的本机的ip 和Port 有一个进程

但是 nginx的service的endpoint 是**一个pod的地址** --- 这个service产生了一个pod ---- 大家描述一下Pod的地址 --- 就是 80的nginx server --- 学习一下get pod的命令

Kubectl get pod 看看nginx 这个pod的ip 地址 是不是nginx service的endpoints的IP地址 ---- **大家截图为证？**

【QQ大家上】

看到了 endponits 的确是10.32.0.3 端口是80 ---- 也就是**service记录所有和它挂钩的pod的地址** ---- 这个pod就是endpoint【单数】 --- 但是 可能存在一种service 他的endpoints和他的pod是没有关系的 ---- endpoints是你指定的任何一个地址 --- 这个service就是Kubernetes service

Kubernetes service既可以是你指定的IP地址 也可以是容器的IP地址 ---- Oracle数据库部署在外面 --- 然后 把你的Oracle的IP port 放在Kubernetes service这里 ---- Kubernetes service就可以通过这个你给定的IP port 来访问这个Oracle --- 但是Oracle到底在什么地方不影响

关键点

\*\* Kubernetes service的IP地址可以是任何一个 可以是集群中的某一个容器 也可以是某一个节点 也可以是公网上的某一个ip 都可以建模 这就是特殊的service 你来控制endpoints

标准情况下 Service Pod之前并非紧耦合 --- 但是大多数情况下 Service 和Pod之间是挂钩的 --- 挂钩的方式 是通过label ---也就是标签的形式



大家把Nignx service的label describe出来 看看是什么

Service上面实际上是有一些标签选择器 label selector ---- **来指定具有哪些label的pod**



这里面有三个pod label都是 app=backend 标签

Service 上有个标签选择器 要选择的是 label是app=backend的pod --- 这时候 Kubernetes发现里面的pod有这样标签的pod 就会把这些pod加到这个service的endpoints里面去 也就是当你产生一个这个表现的Pod 那么这个pod就会被加到对应标签的service里面去了 ---这是自动的

当你这个pod消除之后 你的这个service的endpoints就会消除这个pod

大家试试这个例子 把nginx的pod删掉 再看看nginx service的endpoints里面是否有着地址 是否空了？ 看看谁能完成？

这时候 浏览器再访问nginx这个端口 看看有什么效果

---- 删除这个pod有这样的一个效果 就是 你的端口可以连通 但是没有结果

---- 如果一个service的pod没有了 很难排查问题 就是可以连接 但是 没有响应 这是一个比较头疼的问题

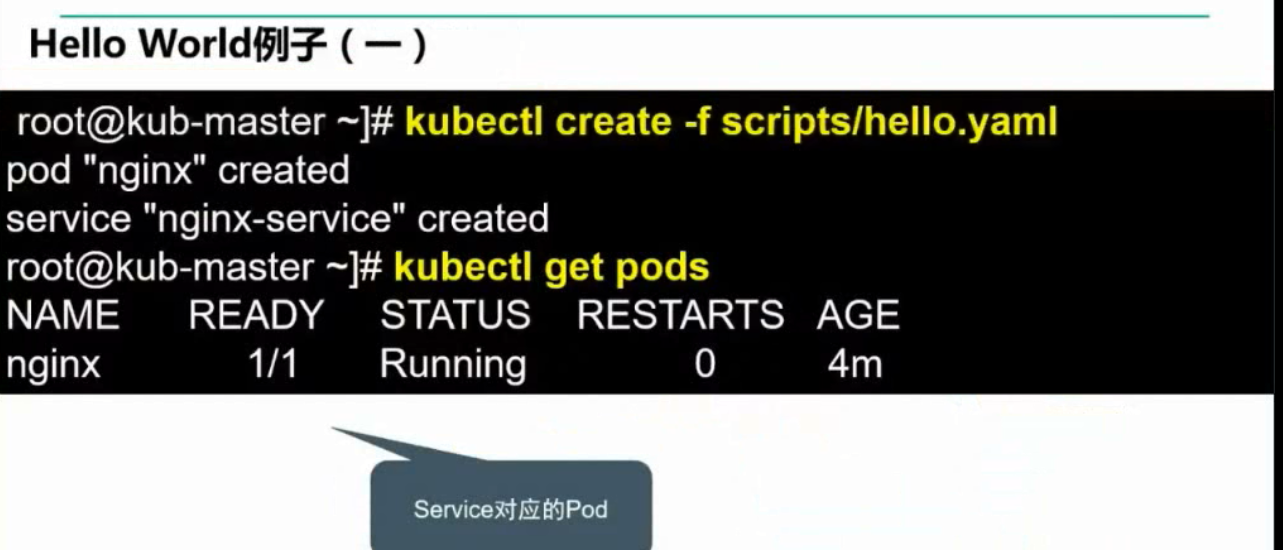
---- 本机Curl这个端口 会出问题

Service中有一个标签选择器 来选择具有某些label特征的pod ---- service自身也有一个标签 这个标签和pod的标签没有关系 和 标签选择器也没有关系

----- 这个是贯穿Kubernetes 所有对象的关联 都是通过标签选择器匹配的

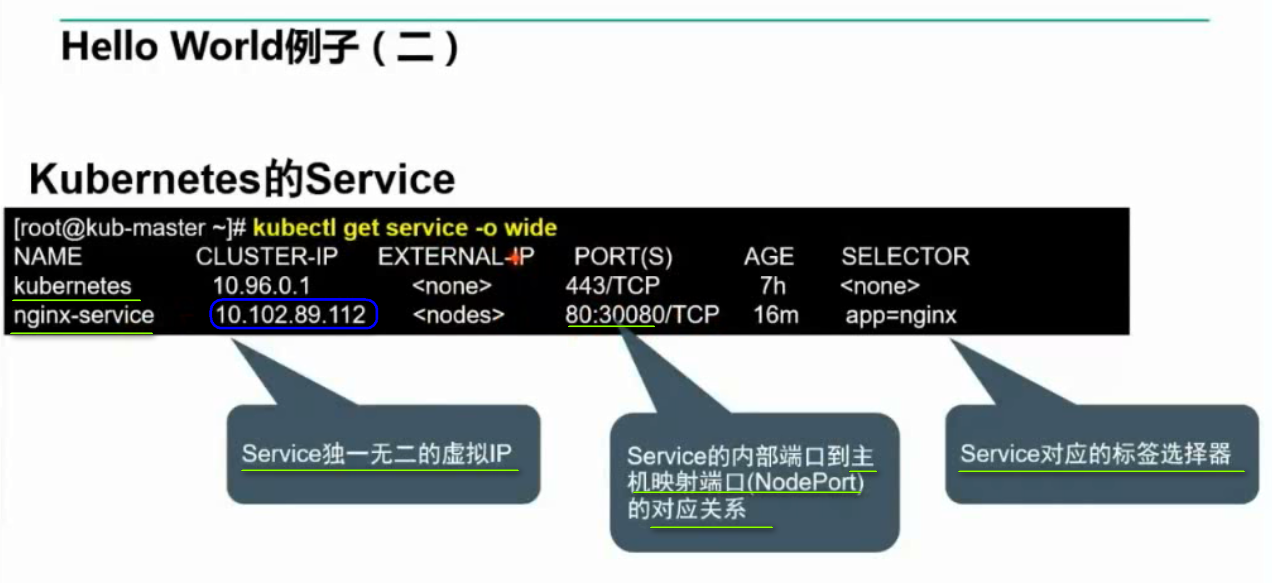
 没匹配上 是不会报错的 这个是比较麻烦的

做好虚机的 把这个例子完成 hello world



这些命令都要掌握 Create –f delete –f就把所有的delete掉

刚才的nginx有一个映射关系



Kubectl get service **–o wide** [-o wide可以给出更多的信息]

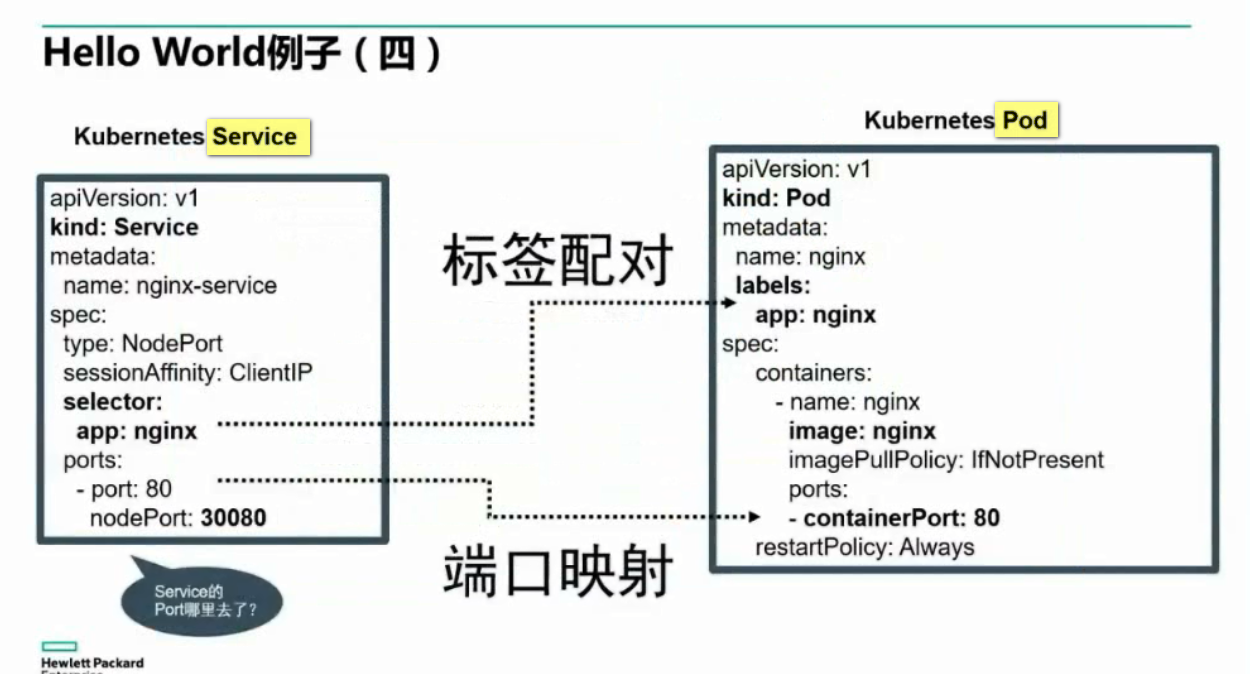
这里面有一个端口映射  也就是 80端口 对应到宿主机的30080 --- 也就是你可以通过宿主机的30080 来进行访问 ---- 这个就是node pod的端口映射

但是kubernetes本身没有标签选择器

 这个是给云厂商提供的 --- 我们自己的宿主机无法做到



下面是kubernetes的yaml文件



 kind表名这个是一个什么资源 是service 还是Pod

名称是在metadata下面的name 【就是name就属于元信息了 】

所有的pod都由label  service很多时候 也有label 但是很多时候 是做管理用

Labels可以加多个标签

Service中的 表示 service有自己的selector 来选择标签为app:nginx的pod 这样就和右边的匹配了

否则就不匹配了

但是 Service的type也是不一样的 kubernetes service的type就是前面显示的 是ClusterIp 【 kubernetes service的Type写的就是ClusterIP】

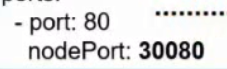
Nginx service的type就是  这个NodePort就是要做端口映射

Service有三个ports ---- 至少有两个port

port

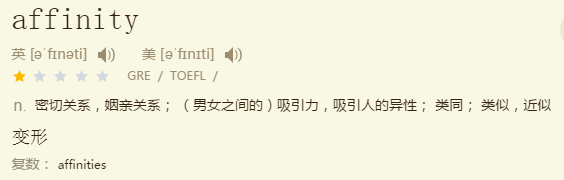
container port ----如果不写container port 那么就和port对等

nodePort --- 如果不写 kubernetes会随机进行分配给你一个可用的

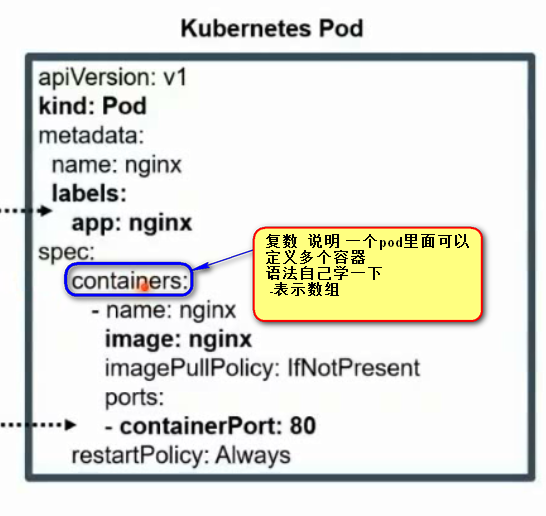
 这地方可以配置一个

目标port都是3000以上的 范围也可以改

 这个表示 是对Client的请求负载均衡的



不写这个 那么 就是第一个IP放到第一个pod 第二个放到第二个



**删掉这个容器**

如果找到这个pod对应的容器 --- 依据显示的容器的名称 把这个容器给kill掉 ---- 或者通过docker命令kill掉 看看是不是这个pod就重启了？ 重启之后 ip地址是否变化 都可以做做实验

**这个hello这个yaml是需要研究的 --- 这个研究清楚 可以试着学习更多了**

**下面的过程 会把其他的概念继续讲解 讲解的过程仍然有例子**

**所以 虚机没有装好 必须这周把虚机转好 把该补的都补上**

=============== 十分钟进行提问 =======================

下次上课 下周的某一天 晚上 这个过程 需要自己学 基本的demo 网上的内容 都可以试验了

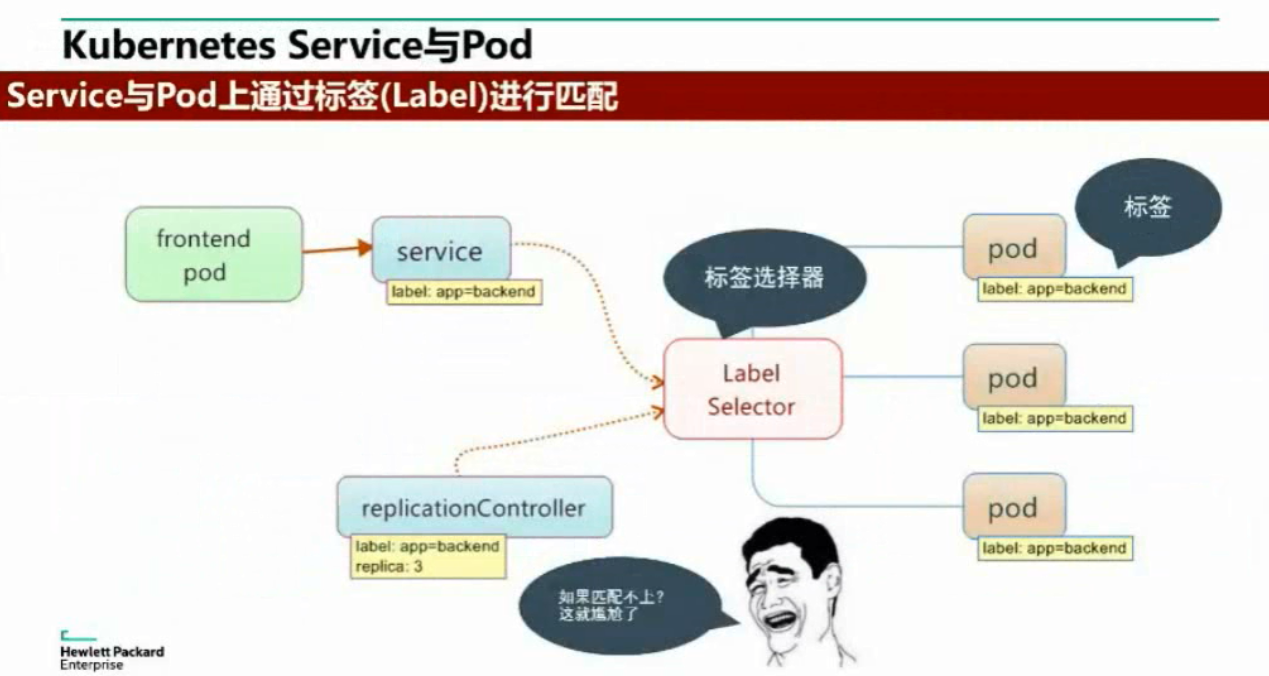
建议大家把命令整理出来 共享出来

这个课程以后 有谁有时间 可以hp做讲师 只能是工作时间 所以 有时间的 可以介绍过去做这样的一个培训

HP大学 有人做开源 讲师 能很好讲解这些高端技术 可以介绍去

互动 安排 都是惠普大学的标准模式

另外这个课程以后 可能希望这里面的经验丰富的 拿着这个课件 讲解后面的 希望锻炼自己 以后参与惠普大学的培训

 这是一张经典的图

大部分都是概念性的 这几天努力

自己写yaml文件

可以用tomcat redis mysql 做一个service 做一个pod 然后 进行访问 都是很大的成就

**题目：你用yaml去定义的mysql服务 然后 本机客户端可以访问 + PPT 然后分享**

---- docker能够做出来 kubernetes就能够做出来

--- 写一个ppt

Mysql 5.7 5.6都行

做法和hello一样的 做完可以在群里分享

做的更复杂 主从同步 更难 --- 数据同步 --- 希望做一个ppt 然后分享 特别是转型讲师 顾问的

标准镜像 联网 都可以下拉下来

认真学习的人 完成你能完成的 看看有多少人把这个作业能完成

---- 前面的ppt错过了 顶多不能吹水而已

听第二次讲课 如果没有做作业 第二次不允许参与 试着做出来

第一次题目 建议在下周三公开 不要提早公开 可以截图 但是 不能给出yaml文件

Yaml可以最后公布 上课前一天公布出来

希望在下周三之前 搞定这个 不要给yaml 可以提示

群里可以交流这些问题 任何人不要给yaml文件源码 语法可以给出

手写很多yaml 才能掌握出来

手写好几个yaml文件

--------- 想再听的人 必须做作业 否则不接受下一次听 ------------------

这个就跑master节点就可以 docker也有 都可以试验