

拉勾教育

— 互联网人实战大学 —

# 《Kubernetes 原理剖析与实战应用》

正范

— 拉勾教育出品 —

拉勾教育

— 互联网人实战大学 —

# 加餐



### 问题 1

监听未调度的 pod 是什么意思，老师没太明白

### 解答

pod 创建出来后，还没有被调度，这种 pod 叫 未调度的 pod  
调度器要去 apiserver 监听（你可以理解为轮询、查询）拿到这些 pod  
根据调度策略，选择合适的节点





### 问题 2

一般 node 向 master 上报状态的频率设置多少比较合适？  
如果上报失败了，被标记了 nodelosy，该如何处理？

### 解答

这个没有统一的结论，要结合集群大小、apiserver 的负载能力来设置  
是一个可以调优的参数，但是不建议初学者去调试  
目前社区默认的是 10s，就使用默认配置即可，已经可以满足绝大多数的集群大小和使用场景了  
上报过程如果遇到网络短暂中断等情况，不必担心，不会因为一次失败就标记为 NodeLost 的  
如果最后被标记了，可以结合告警等手段，及时发现问题





### 问题 3

声明式还是有点糊涂，希望老师讲的更清楚点呀！

### 解答

打个比方，你现在想吃鱼

声明式 API 就是给服务员下单，我想要盘酸菜鱼，就 OK 了，其他不用操心，静静等待上桌即可

命令式 API 就是自己去 follow 所有的流程，从买鱼、杀鱼、片鱼、清洗到下锅

套用比较火的台词，声明式 API 就是“我不管你怎么想，我就要这个”

命令式 API 就是“我不管你怎么想，照我说的做”





### 问题 1

那平时运维管理这些容器用 kubernetes dashbord 可视化看吗?

### 解答

可以的





### 问题 2

还是没搞明白，kubernetes 到底是啥，能干啥？

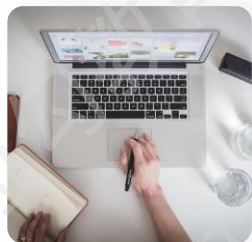
### 解答

Kubernetes 是一个容器管理平台，可以用来做容器化的应用发布部署并提供容器调度、弹性伸缩、负载均衡、自愈等诸多功能  
可以将 Kubernetes 理解成 “容器云”





### 问题 3



关于升级策略，有个疑问：假如初始版本和官方一致  
例如 1.16，0.5 年后官方升级至 1.18，我方升级至 1.17，相差 1 个小版本  
1 年后官方升级至 1.20，我方升级至 1.18，相差 2 个小版本  
1.5 年后官方升级至 1.22，我方升级至 1.19，相差 3 个小版本  
2 年后官方升级至 1.24，我方升级至 1.20，相差 4 个小版本  
那么，我方就必须在这两年内最少要做一次小版本升级

### 解答

是的

社区也意识到这个问题，所以从 1.19 版本开始，支持窗口已经延长到一年了  
也就是你可以一年内不用升级 1.19 的集群。不过对于之前的版本  
还是得抓紧升级上来的







### 问题 4

为啥不用二进制部署啊？

### 解答

对于很多刚入门的人来说，二进制安装步骤过于繁琐  
如果你已经知道一些安装的步骤，比如一些参数配置、配置文件、证书等  
可以选择二进制安装





### 问题 5

可以介绍一下二进制搭建 Kubernetes 高可用集群的方法吗？

### 解答

其实是一样的，都需要配置一些证书以及参数

可以通过 kubeadm 帮你生成一些证书然后通过 systemd 来管理各个组件的启动即可





### 问题 5

Pod 从 ContainerCreating 状态变为 Running 状态的过程大概经历了 10 分钟  
这种场景如何排查呢？

### 解答

先排查调度器是否调度，再看调度节点的情况

可以通过 `kubectl describe pod-name -n some-ns` 来看看相关的 event 信息





### 问题

假如我创建了 10G 的 PV，已经通过 PVC 挂载至 Pod 用过一段时间后，发现空间不够，需要扩展 PV，如何将它扩展至 20G？

### 解答

修改 pvc 的 size 即可





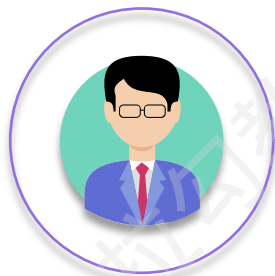
### 问题

请教下老师，服务 A 使用 nodeport 访问，服务 B 使用 host 模式，如何访问？  
宿主的 ip 不是固定的

### 解答

nodeport 的话，可以通过任一宿主机 ip+port 的方式来访问  
所以服务 A 和服务 B 可以通过任一宿主机 ip 来访问





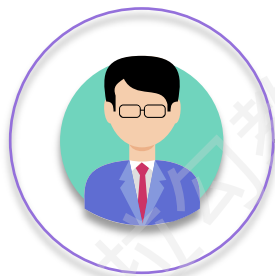
01 讲中讲到 Kubernetes 是如何火起来的



\*\*龙 同学

我常常给别人做这个比喻，docker 就像是一块通用的砖，既能盖普通的房子也能盖白宫，让你不用担心制造砖的细节和标准  
但是通过砖去盖房子的步骤要我们去探索  
k8s 则是图纸，你有多高的设计能力就能快速给你搞出房子  
k8s 的出现我们不仅要感谢 google 还有感谢 docker 毕竟是 google 看到 docker 太火了并且 ‘不听话’，才决定 ‘打’ 他的，现在已经被打到体无完肤了



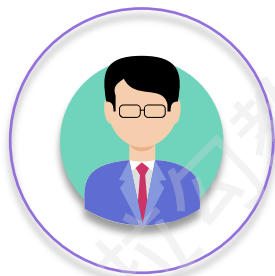


04 讲中讲到 Kubernetes 是如何搞定“不可变基础设施”的



\*\*龙 同学

不可变基础设施是容器化改造中让业务比较难理解的东西  
以前虚拟机的思想已经固化了，就想着进去改掉东西  
容器化的过程中还有业务的架构导致很难容器化，还要业务进行架构调整

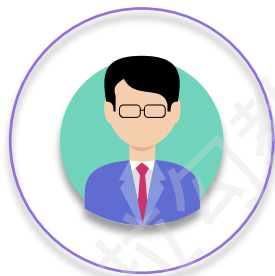


08 讲谈到 Kubernetes 管理业务配置方式



\*\*某同学

对于 ConfigMap 或者 Secret 的‘热更新’，我目前的方法比较原始  
在 Deployment 对象中增加一个 ConfigMap 或 Secret 发生变化  
则修改这个注解的值，从而触发工作负载对象的滚动升级



14 讲中，关于“如何在 Kubernetes 中做日志收集与管理”的问题



\*\*0530 同学

一般程序的业务日志都不是标准输出流的……都是自定义的file的日志  
还是使用每个node挂一个agent最常用，可以结合ELK使用哈

Next: 《24 | 调度引擎：Kubernetes 如何高效调度 Pod? 》

# 拉勾教育

— 互联网人实战大学 —



关注拉勾「教育公众号」  
获取更多课程信息