

docker、docker-compose、docker swarm和k8s的区别

1、Docker

Docker 这个东西所扮演的角色，容易理解，它是一个容器引擎，也就是说实际上我们的容器最终是由Docker创建，运行在Docker中，其他相关的容器技术都是以Docker为基础，它是我们使用其他容器技术的核心。

2、Docker-Compose

Docker-Compose 是用来管理你的容器的，有点像一个容器的管家，想象一下当你的Docker中有成百上千的容器需要启动，如果一个一个的启动那得多费时间。有了Docker-Compose你只需要编写一个文件，在这个文件里面声明好要启动的容器，配置一些参数，执行一下这个文件，Docker就会按照你声明的配置去把所有的容器启动起来，只需docker-compose up即可启动所有的容器，但是Docker-Compose只能管理**当前主机**上的Docker，也就是说不能去启动**其他主机**上的Docker容器

3、Docker Swarm

Docker Swarm 是一款用来管理**多主机**上的Docker容器的工具，可以负责帮你**启动容器，监控容器状态**，如果容器的状态不正常它会帮你重新帮你启动一个新的容器，来提供服务，同时也提供服务之间的负载均衡，而这些东西Docker-Compose 是做不到的

4、Kubernetes

Kubernetes它本身的**角色定位是和Docker Swarm 是一样的**，也就是说他们负责的工作在容器领域来说是相同的部分，都是一个**跨主机的容器管理平台**，当然也有自己一些不一样的特点，k8s是谷歌公司根据自身的多年的运维经验研发的一款容器管理平台。而Docker Swarm则是由Docker 公司研发的。

既然这两个东西是一样的，那就面临选择的问题，应该学习哪一个技术呢？实际上这两年Kubernetes已经成为了很多大公司的默认使用的容器管理技术，而Docker Swarm已经在这场与Kubernetes竞争中已经逐渐失势，如今容器管理领域已经开始已经逐渐被Kubernetes一统天下了。所以建议大家学习的时候，应该多考虑一下这门技术在行业里面是不是有很多人在使用。

需要注意的是，虽然Docker Swarm在与Kubernetes的竞争中败下阵来，但是这个跟Docker这个容器引擎没有太大关系，它还是整个容器领域技术的基石，Kubernetes离开他什么也不是。

总结

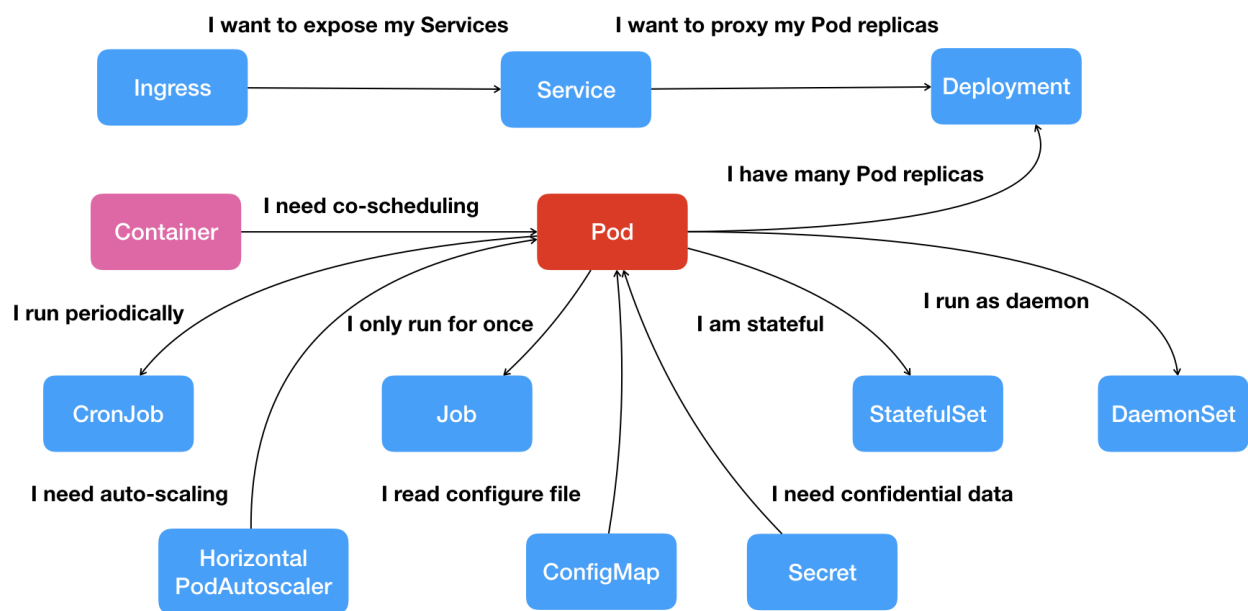
Docker是容器技术的核心、基础，Docker Compose是一个**基于Docker的单主机容器编排工具**。而**k8s是一个跨主机的集群部署工具**，功能并不像Docker Swarm和Kubernetes是基于Docker的跨主机的容器管理平台那么丰富

为什么有了Compose和Swarm，还会有Kubernetes的出现？

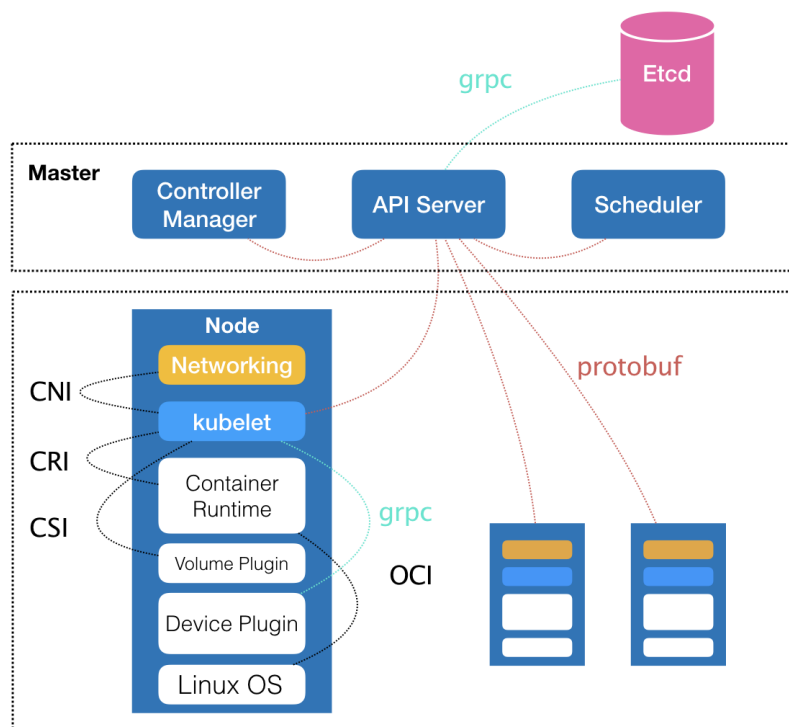
一、k8s设计思想更先进

k8s的主要设计思想，是从更宏观的角度，以统一的方式来定义任务之间的各种关系

1.k8s的核心功能图



2.k8s的全局架构图



把微服务比喻为人，服务治理解决的是人的沟通，人太多了就需要生存空间和沟通方式的优化，这就需要集群和编排。

compose和swarm可以解决少数人之间的关系，比如把手机号给你，你就可以方便的找到我，但是如果手机号变更的时候就会麻烦，人多了也会麻烦。

而k8s是站在上帝视角的高度抽象，看到了

1. 总体有哪些组织，不同组织有什么样的特点（Job、CronJob、Autoscaler、StatefulSet、DaemonSet...）
2. 不同组织之间交流可能需要什么（ConfigMap, Secret...），这样比较紧密的人在相同的pod中，通过Service-不会变更的手机号，来和不同的组织进行沟通，
3. 帮助人们快速构建组织(Deployment、RC)。

k8s就是把组织协调这项管理学落实到了计算机工程上

二、功能对比

1. swarm偏重的是容器的部署，而k8s偏重应用的部署

swarm中最小单元是容器，而k8s是pod，pod可以由多个容器组成，在pod内共享volume和namespace，同一pod内的通信更为高效

pod有什么好处？

例如有一个web容器，为了收集web日志，需要安装一个日志插件，如果把插件安装在web容器内：

1. 如果插件有更新，即使服务没有变化也要重新把镜像构建部署一遍
2. 如果插件存在内存泄露问题，整个容器都会被连累

而pod可以为日志插件和web应用各自创建一个容器，两者共享volume，web应用只需要日志保存到volume，两个容器各自有自己的镜像，更新互不影响

2. k8s比swarm有更多的调度策略，更适合大规模容器的管理

swarm只有三种调度策略：spread、binpack、random，而k8s策略数更多，还有端口冲突策略、容器挂载卷冲突策略、指定特定宿主机策略等。

Composer中，通过link将容器关联起来，如DB的连接写入环境变量供进程使用，如果DB发生变化（如镜像）集群中的节点，只要在同一network内，服务之间

3. k8s的负载均衡机制比swarm更灵活

swarm采用的是nginx+consul。

consul保存了各个docker中应用的网络信息，nginx在compose时，在dockerfile中指定consul的地址，配置到nginx配置中，从而实现负载均衡，这样有个缺点，就是新添加的容器IP和网络需要手动添加到nginx文件中

而k8s负载均衡通过service实现，没有容器IP变更问题，只要有相同的label的pod都可以通过service访问，新添加的容器IP和网络不会影响负载均衡器

4.k8s支持弹性伸缩

k8s可以根据Pod的CPU、内存自动的调整Pod的个数，保障服务的可用性，swarm则不具备这样的功能