K8s

leolinuxer

October 19, 2020

Contents

1 背景

`https://zhuanlan.zhihu.com/p/103124918`

k8s 全称 kubernetes。k8s 是为容器服务而生的一个可移植容器的编排管理工具，越来越多的公司正在拥抱 k8s，并且当前 k8s 已经主导了云业务流程，推动了微服务架构等热门技术的普及和落地，正在如火如荼的发展。

首先，我们从容器技术谈起，在容器技术之前，大家开发用虚拟机比较多，比如 vmware 和 openstack，我们可以使用虚拟机在我们的操作系统中模拟出多台子电脑（Linux），子电脑之间是相互隔离的，但是虚拟机对于开发和运维人员而言，存在启动慢，占用空间大，不易迁移的缺点。举一个我亲身经历过的场景吧，之前在 vmware 中开发了一个线下平台，为了保证每次能够顺利使用，我们就把这个虚拟机导出为 OVF，然后随身携带，用的时候在服务器中部署，这里就充分体现了虚拟机的缺点。

接着，容器化技术应运而生，它不需要虚拟出整个操作系统，只需要虚拟一个小规模的环境即可，而且启动速度很快，除了运行其中应用以外，基本不消耗额外的系统资源。Docker 是应用最为广泛的容器技术，通过打包镜像，启动容器来创建一个服务。但是随着应用越来越复杂，容器的数量也越来越多，由此衍生了管理运维容器的重大问题，而且随着云计算的发展，云端最大的挑战，容器在漂移。在此业务驱动下，k8s 问世，提出了一套全新的基于容器技术的分布式架构领先方案，在整个容器技术领域的发展是一个重大突破与创新。

那么，K8S 实现了什么?

从架构设计层面，我们关注的可用性，伸缩性都可以结合 k8s 得到很好的解决，如果你想使用微服务架构，搭配 k8s，真的是完美，再从部署运维层面，服务部署，服务监控，应用扩容和故障处理，k8s 都提供了很好的解决方案。

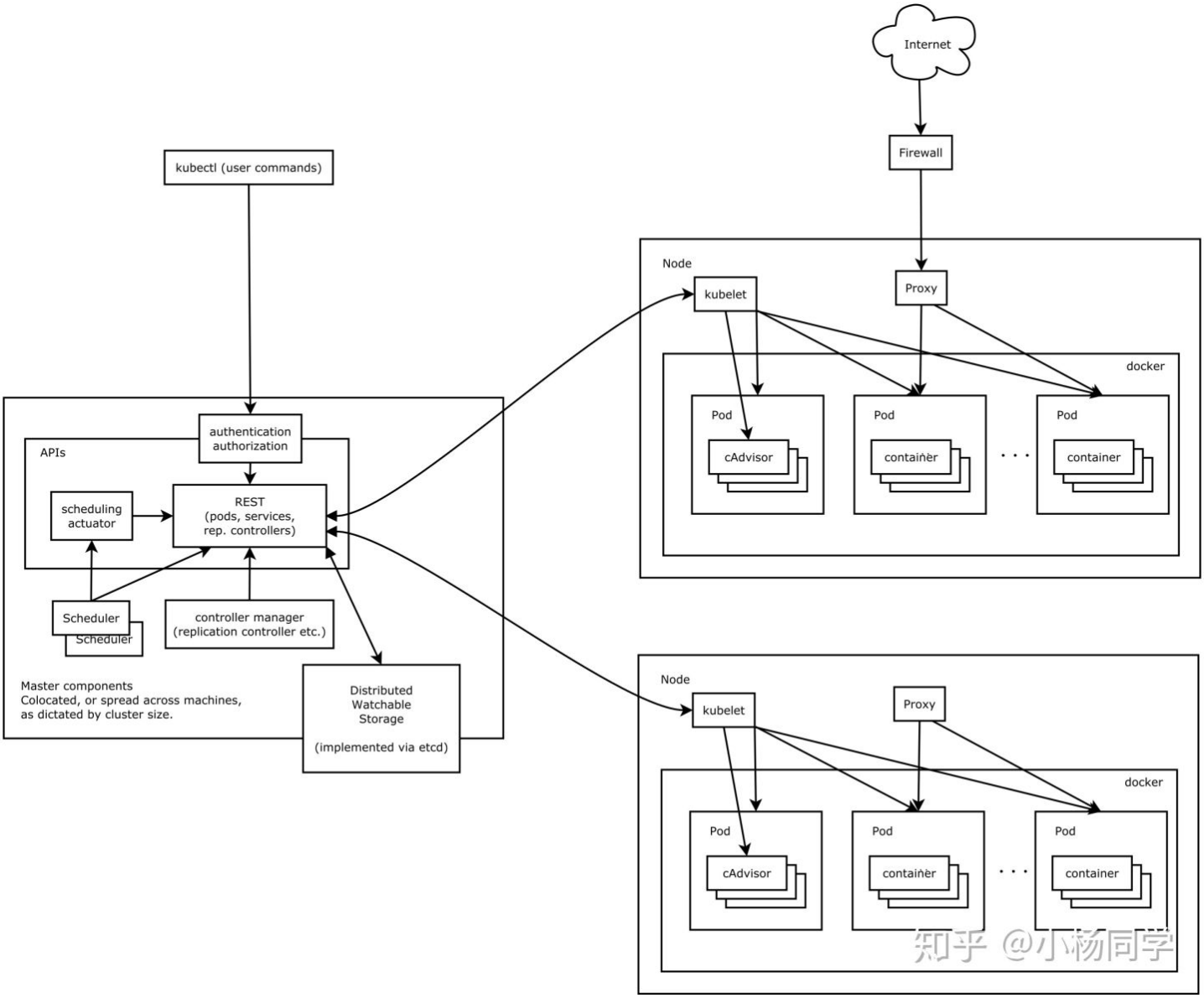
具体来说，主要包括以下几点：

- 服务发现与调度
- 负载均衡
- 服务自愈
- 服务弹性扩容
- 横向扩容
- 存储卷挂载

总而言之，k8s 可以使我们应用的部署和运维更加方便。

2 K8s 架构

最后，我们看下 k8s 的架构:



k8s 集群由 Master 节点和 Node（Worker）节点组成。

2.1 Master 节点

Master 节点指的是集群控制节点，管理和控制整个集群，基本上 k8s 的所有控制命令都发给它，它负责具体的执行过程。在 Master 上主要运行着：

1. Kubernetes Controller Manager（kube-controller-manager）：k8s 中所有资源对象的自动化控制中心，维护管理集群的状态，比如故障检测，自动扩展，滚动更新等。
2. Kubernetes Scheduler（kube-scheduler）：负责资源调度，按照预定的调度策略将 Pod 调度到相应的机器上。
3. etcd：保存整个集群的状态。

2.2 Worker 节点

除了 master 以外的节点被称为 Node 或者 Worker 节点，可以在 master 中使用命令 kubectl get nodes 查看集群中的 node 节点。每个 Node 都会被 Master 分配一些工作负载（Docker 容器），当某个 Node 宕机时，该节点上的工作负载就会被 Master 自动转移到其它节点上。在 Node 上主要运行着：

1. kubelet：负责 Pod 对应的容器的创建、启停等任务，同时与 Master 密切协作，实现集群管理的基本功能。
2. ube-proxy：实现 service 的通信与负载均衡。
3. docker（Docker Engine）：Docker 引擎，负责本机的容器创建和管理。

2.3 pod ——k8s 调度的最小单元

`https://www.jianshu.com/p/502544957c88`

一个 pod 包含一组容器，一个 pod 不会跨越多个工作节点

- pod 相当于逻辑主机，每个 pod 都有自己的 IP 地址
- pod 内的容器共享相同的 IP 和端口空间
- 默认情况下，每个容器的文件系统与其他容器完全隔离

References