

# K8S基础概念

K8S:一个基于容器部署的分布式系统。可为容器化应用部署运行、资源调度、服务发现和动态资源伸。(多层次安全防护、准入机制、多租户、服务发现注册、负载均衡、滚动更新、可扩展在线扩容功能)

k8s vs docker : docker主要提供容器的生命周期和镜像构建。主要是将环境与程序打包成在一个镜像中的功能。k8s主要是提供良好的容器编排系统。

## 容器基础:

- **容器功能:**通过打包镜像的方式,解决应用环境部署问题。容器隔离在与宿主机看似完全无关的世界中,但其实仍然是个普通进程。

步骤:

- a. 启用linux namespace配置
  - b. 设置指定的cgroup参数
  - c. 切换容器进程的根目录
- 采用**namespace技术**:让容器进程看到一个全新的空间。(网络、pid、mount、user等)
  - 采用**CGroup技术**:为进程设置资源限制的功能。(包括:CPU 磁盘 内存 网络带宽)
  - 引入层的概念,将dockerfile中的镜像制作变为一层层的增量层(**联合文件系统技术**)

作法:一个子系统加上一组资源限制文件的组合完成的

- 编写dockerfile
- docker命令

**附注:**docker在使用时,并没有使用docker运行容器,docker是帮助用户启动应用进程时,自动追加各种namespace参数,但本质还是在速主机上运行。(仅仅是看起来而已)

**虚拟机的缺点:**在宿主机上运行虚拟机时,虚拟机对宿主os调用时需要经过虚拟化软件的拦截与处理,是一种很大的性能消耗。

## 容器云k8s

各组件:

**Master节点:** api server + controller + scheduler + etcd

**node节点:**

**kubelet:**主要负责容器运行时的沟通与汇报功能,调用网络插件与存储插件为容器配置网络与持久化存储。

**CRI接口:**定义容器运行的核心操作。ex:启动容器时需要的参数。(只要容器运行的标准容器镜像，可以通过此接口接到k8s中)

**CNI:**

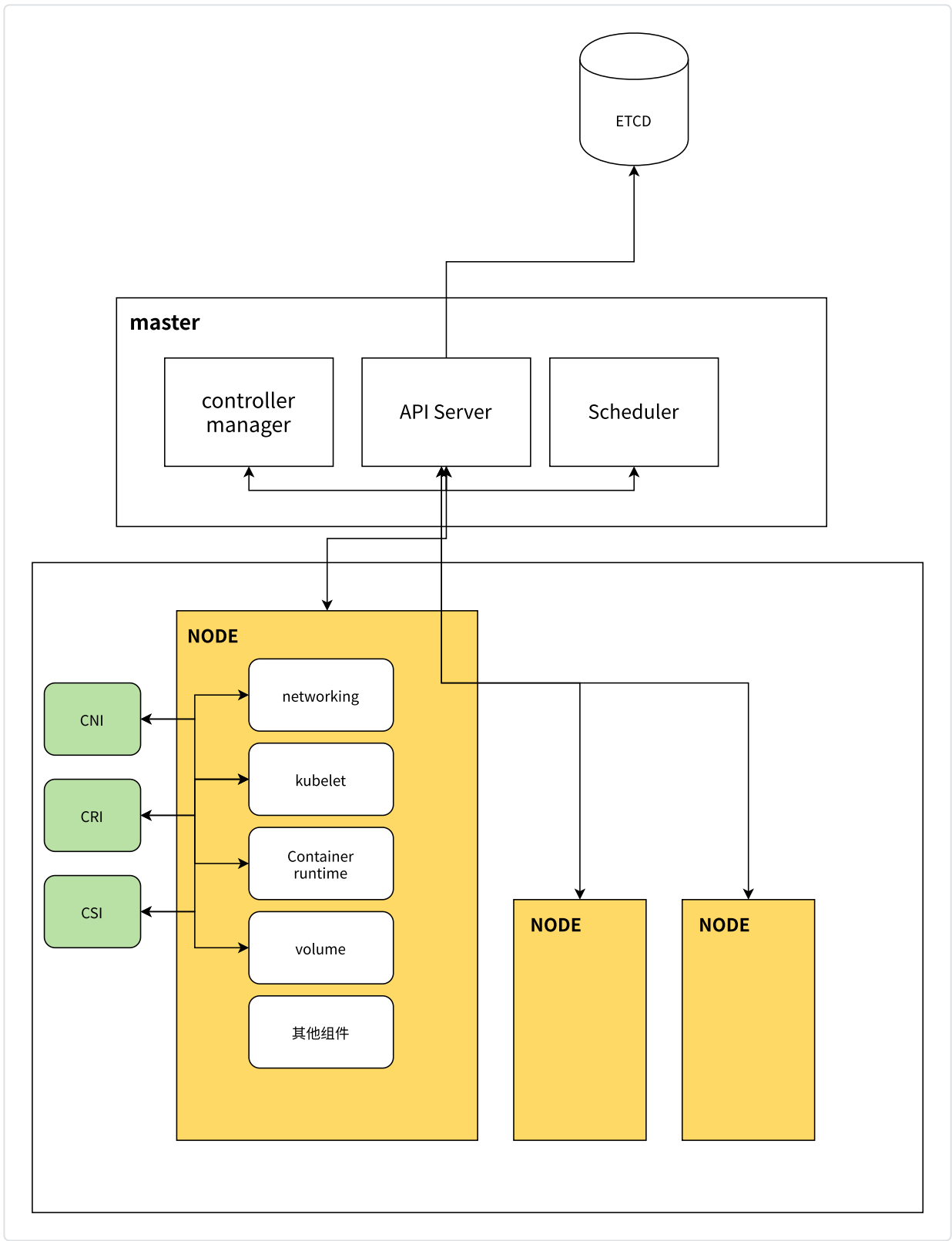
**CSI:**

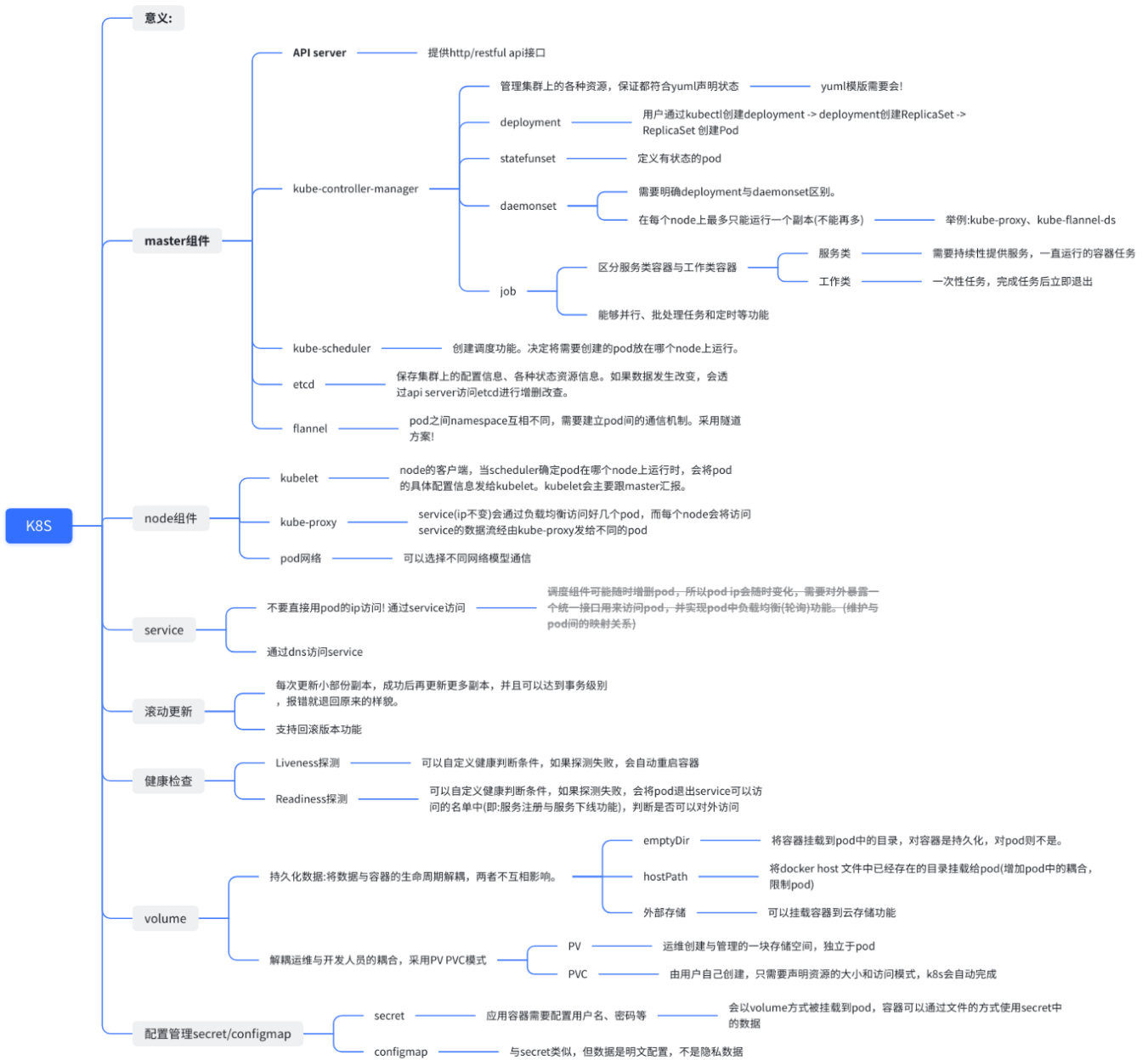
**pod:**k8s中最基础的原子调度单位。pod内部共享同一个网络namespace、同个数据卷。(可以看成是一个"进程组"的概念)

**service:**pod绑定service服务，可以提供一个固定的ip地址访问节点中的pod，并且实现服务注册、负载均衡

**secret:**将隐私信息保存在etcd中

调度管理pod控制器:deployment、daemon、job、cronjob



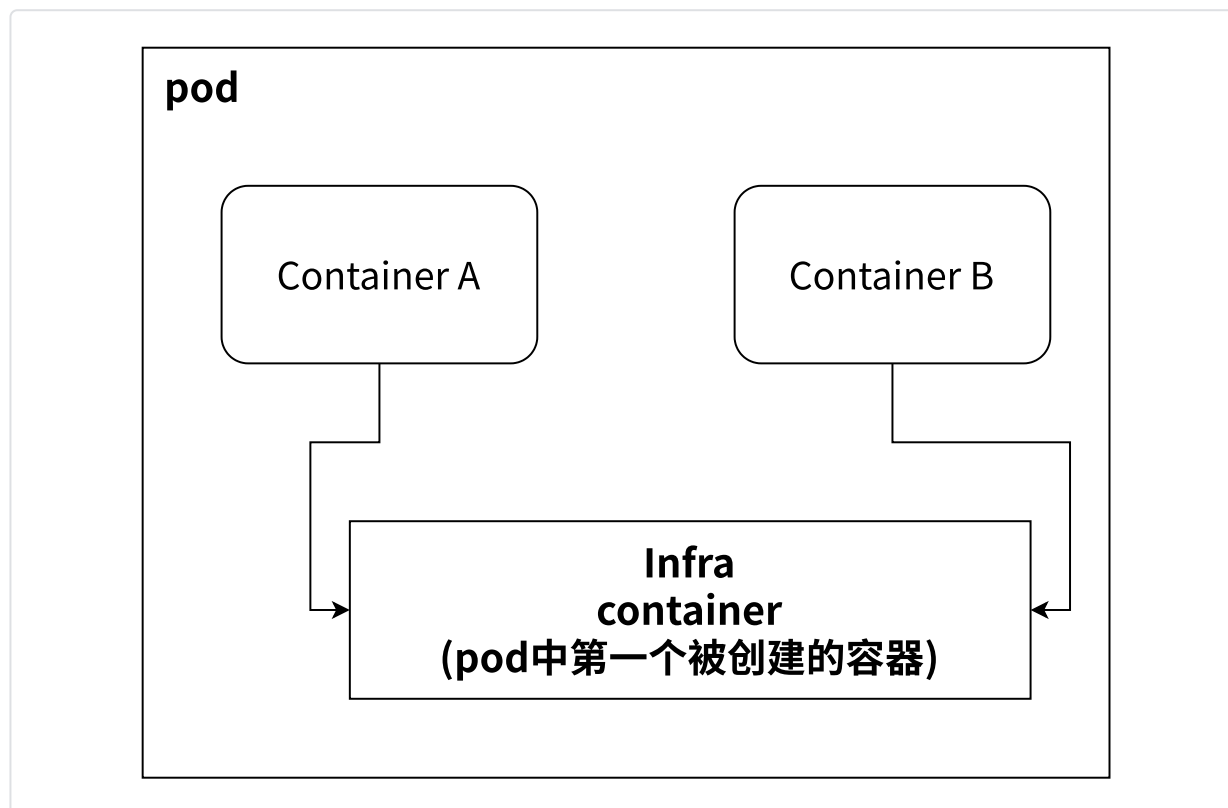


## POD

- why需要pod? 容器的本质是进程, 而pod的本质就是进程组, 进程组(pod)中可以起好多进程(container), 可以有对应的资源分配方式(namespace)、网络通信方式(容器模型)、挂载数据卷方式, 当中可以有不同种类的控制器(controller manager管理)等。
- pod是k8s中的原子调度单位, k8s调度器是按照pod而非容器的资源需求分配的。
- pod中有一个轻量级中间容器Intra容器。Intra容器永远是pod中第一个创建的容器
- 经常用**sidercar**的方式启动容器!
- pod能定义的属性:调度、网络、存储和安全相关的属性
- 

图中的A和B:

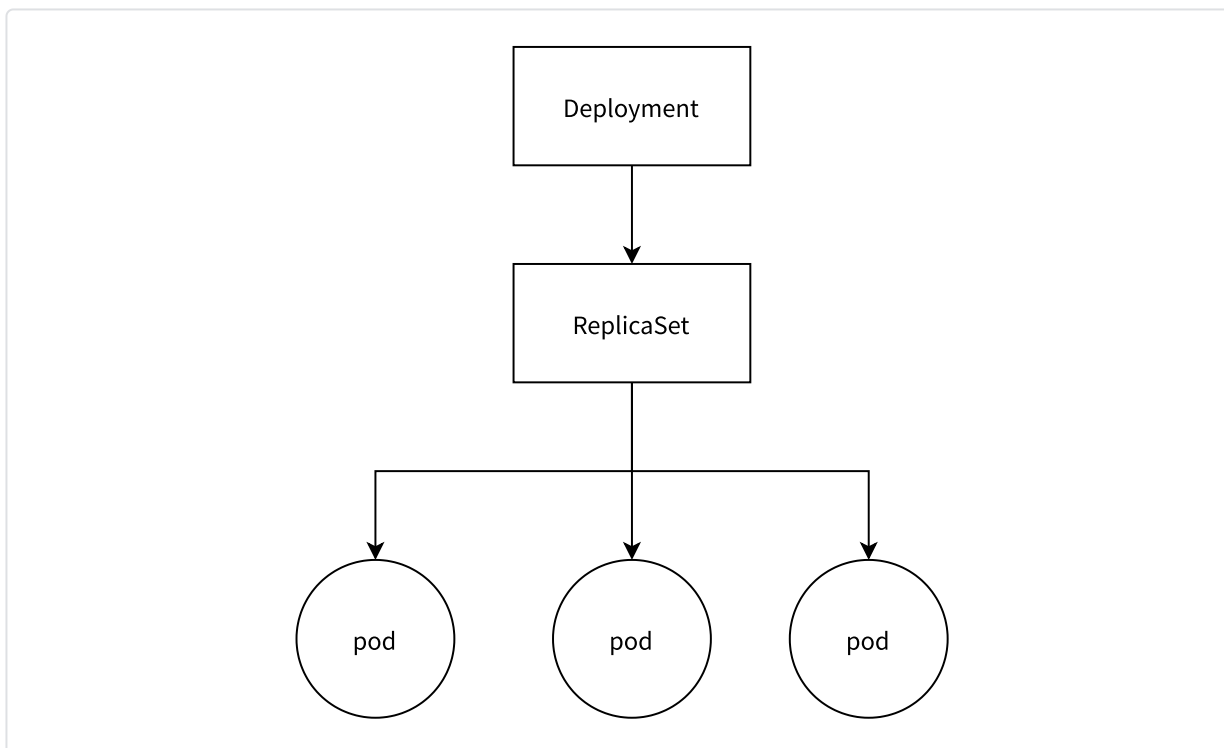
1. 可以直接由localhost通信
2. 彼此看到的网络设备与Infra container相同
3. 一个pod只有一个ip地址，因为网络namespace
4. pod生命周期和Infra container一致，和A B无关



## 控制器

### Deployment控制器

- deployment编排方式:从yaml中声明的副本个数，以循环控制的方式检查副本数量，如果副本数少于期望->增加副本，如果多于期望->减少副本。
- kubelet通过心跳汇报容器状态和节点状态、监控系统中保存的应用监控数据、控制器主动搜集的信息都状态来源。
- deployment遵循滚动更新的方式升级容器版本、实现容器副本的水平扩展与收缩操作
- ReplicaSet对象:由副本数与POD模板组成的。
- deployment控制器操作的是ReplicaSet对象，不是pod。由ReplicaSet操作pod。



## StatefulSet控制器

与deployment区别:deployment管理的pod都是创建与编排都是没有先后与优先级区分的，可以任意的方式创建与删除。而往往很多容器实例均有从属关系、先后创建关系

StatefulSet保证了pod之间的先后顺序状态，称为**有状态管理**。

在pod中的状态采用pod的名字+编号固定下来，

## DaemonSet控制器

类似守护进程的意思，被DaemonSet控制器创建的pod在每个节点中仅仅只有一个。

## Job控制器

适合工作类型服务。

与deployment区别:job任务运行结束就立即退出了，deployment则是会不断重启。

提供并行与批量运行功能