

学习内容：

Kubernetes 教程

收获：

- 两种实战方式 kubectl 和 Kubeboard
- Pod 容器组包括共享存储、网络和容器的基本信息
- 每个 Kubernetes Node 至少运行 Kubelet 和容器运行环境
- Service 通过设置配置文件中的 spec.type 字段的值: ClusterIP、NodePort 和 LoadBalancer
- 设置 spec.replicas 来给应用程序扩容
- 节点的状态包含如下信息：
 1. Addresses: HostName、ExternalIP 和 InternalIP
 2. Conditions: OutOfDisk、Ready、MemoryPressure、PIDPressure、DiskPressure 和 NetworkUnavailable
 3. Capacity and Allocatable: CPU、内存、该节点可调用的最大数量
 4. Info: Linux 内核版本、Kubernetes 版本、Docker 版本和操作系统名称
- Kubernetes 在 APIServer 上建一个节点 API 对象（节点的描述），并且基于 metadata.name 字段对节点进行健康检查
- 节点控制器是一个负责管理节点的 Kubernetes master 组件
- 控制 kubelet 的启动参数 --register-node 选择自动或手动创建
- 从 master (apiserver) 到 Cluster 存在着两条主要的通信路径：
 1. apiserver 访问集群中每个节点上的 kubelet 进程
 2. 使用 apiserver 的 proxy 功能，从 apiserver 访问集群中的任意节点、Pod、Service
- 控制器都是通过与集群中 API Server 交互来达到调整状态的目的，某些特殊的控制器将从 API Server 中读取关于目标状态的信息，并直接调用外部接口以实现调整目标
- 在.yaml 文件中必须填写：apiVersion、kind、metadata、spec
- 管理 Kubernetes 对象的方式：

| 管理方式 | 操作对象 | 推荐的环境 | 参与编辑的人数 | 学习曲线 |
|----------|-------------------|-------|---------|------|
| 指令性的命令行 | Kubernetes 对象 | 开发环境 | 1+ | 最低 |
| 指令性的对象配置 | 单个 yaml 文件 | 生产环境 | 1 | 适中 |
| 声明式的对象配置 | 包含多个 yaml 文件的多个目录 | 生产环境 | 1+ | 最高 |

- 通过 namespace+name 唯一性地确定一个 RESTFUL 对象
- 三种广泛使用的资源名称的限制类型：DNS Subdomain Names、DNS Label Names、Path Segment Names
- UID 是由 Kubernetes 系统生成的，唯一标识某个 Kubernetes 对象的字符串