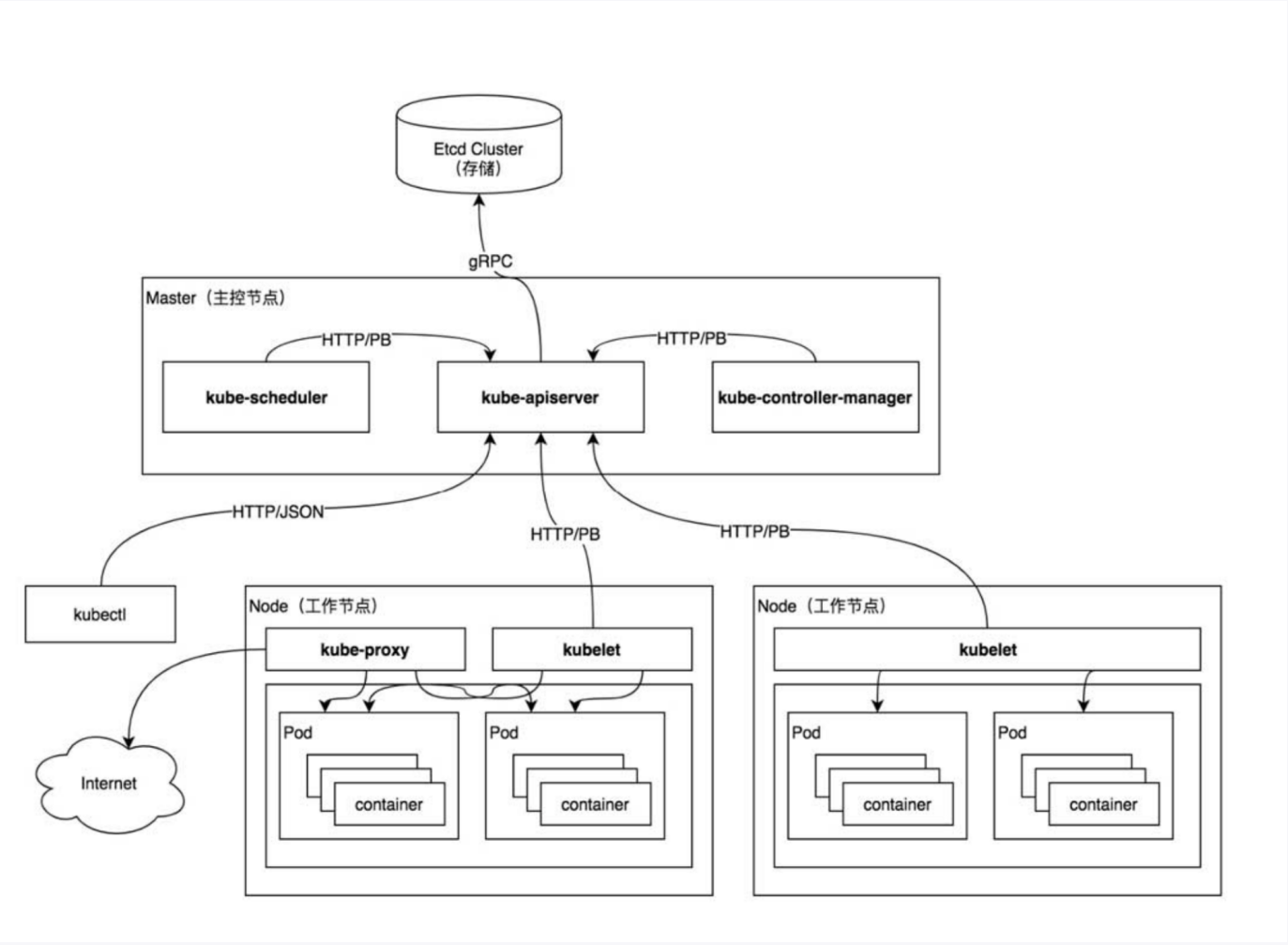


总架构图



pod pending 原因

- 镜像有问题
- 节点资源不足
- kube-scheduler 未正常工作

Kubernetes 架构

一个Pod创建流程

- 用户通过操作界面或者CLI提交了一个 Pod 给 K8S 进行部署，这个 Pod 请求被提交给了 API Server
- API Server 会把这个信息写入到存储系统 Etcd 中，然后通过 watch 机制 Scheduler 知道了有一个 Pod 需要被调度
- Scheduler 根据目前集群的资源状态进行了一次调度决策，然后汇报给 API Server 此次调度信息（这个 Pod 需要被调度到哪个节点上）
- API Server 接收到调度信息后，将其写入 Etcd，然后通知相关 Node节点的 kubelet 去执行
- Kubelet 得到通知，调用 CRI 进行容器的创建和运行，调用 Storage Plugin 去配置存储，调用 Network Plugin 去配置网络
- kube-controller-manager 会确保 pod 处于预期的状态

kube-proxy 原理

监听 api-server 中 service 和 endpoint 的变化情况, 并通过 iptables 等来为服务配置负载均衡

遵循 C/S 架构

- Master 作为服务端
 - kube-api-server 提供组件通信服务, k8s中所有的组件都会和 api-server 通信, 都哦依赖它进行消息的传递
 - kube-controller-manager 不断检查各个pod是否符合声明的状态, 如果不一致, 就进行相应的新增、删除、替换工作
 - kube-scheduler 根据每个pod对CPU, memory的要求, 负责把它放到合适的Node上
- Node 作为客户端
 - kubelet 真正去运行Pod的组件, 与 api-server 通信, 获取到所需运行的Pod状态, 通过 CRI 远程调用接口, 向 container runtime 发起创建容器的请求
 - kube-proxy 负责对集群master与node节点的服务发现网络, 利用了 iptables 来实现

etcd

- 为整个集群提供服务发现、配置协同功能
- 可信赖的分布式键值存储服务
- raft 算法, 强一致性
 - 同一时间只会存在一个leader, 所有的操作都在 leader 上
 - leader 节点负责集群中的日志复制和管理工作