

[VIP课]

# Kubernetes入门与实战

DO ONE THING AT A TIME AND DO WELL.

➤ 大白老师QQ号: 1828627710



## 咕泡学院- 大白老师

### 前大众点评架构师

十余年Java经验，曾任职于1号店、大众点评、同程旅游、阿里系公司，担任过技术总监、首席架构师、team leader、系统架构师。有着多年的前后台大型分布式项目架构经验，在处理高并发、性能调优上有独到的方法论。精通Java、J2EE架构、Redis、MongoDB、Netty，消息组件如Kafka、RocketMQ。



# 课程目标



- 了解Docker产生的背景
- 了解Docker与传统虚拟化技术的差异
- 了解Docker的应用场景
- 深入理解Docker架构和核心组件
- 学会使用常用命令
- 学会编写Dockerfile文件
- 了解Docker的数据卷的管理并熟练使用





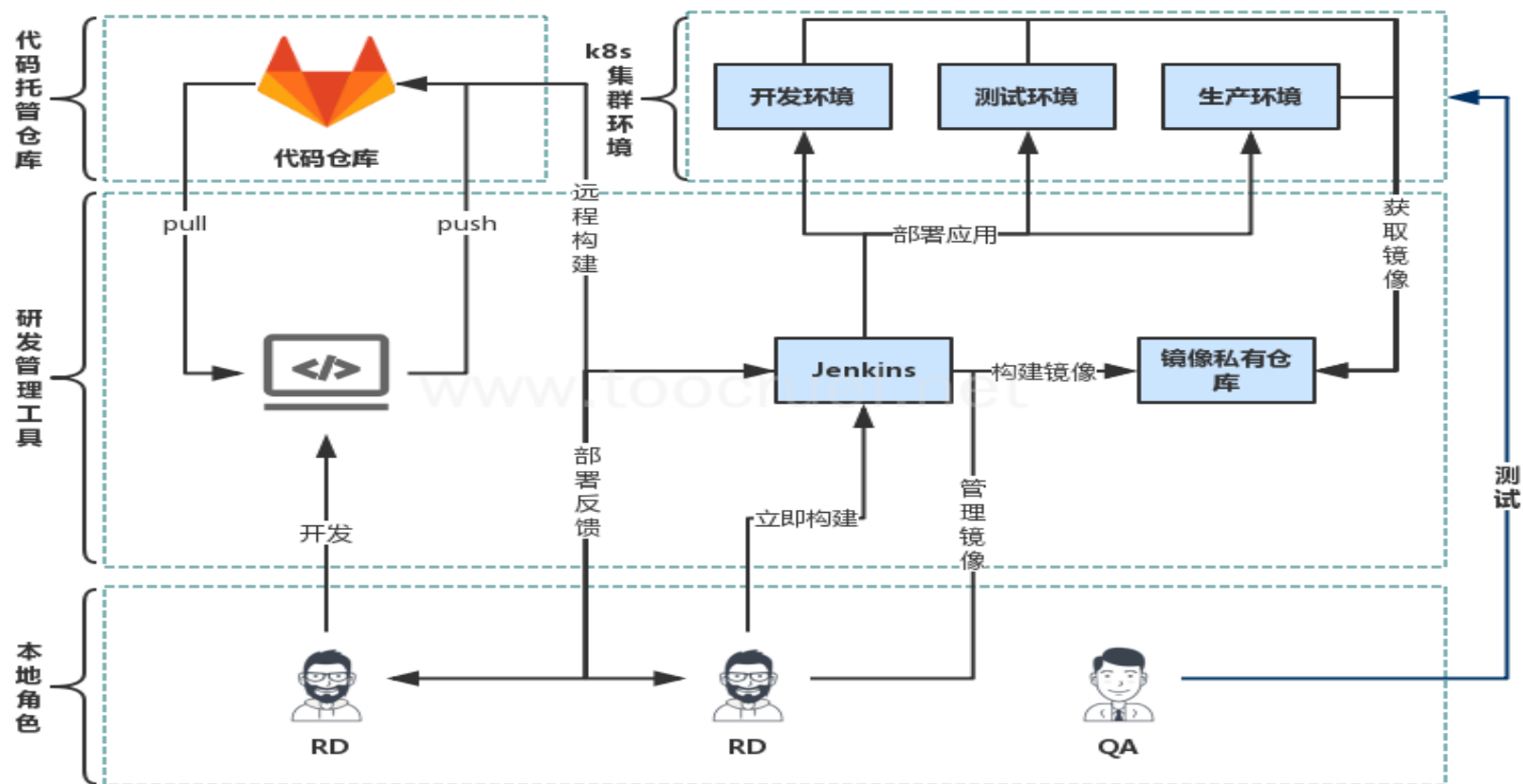
1

# Kubernetes入门与实战

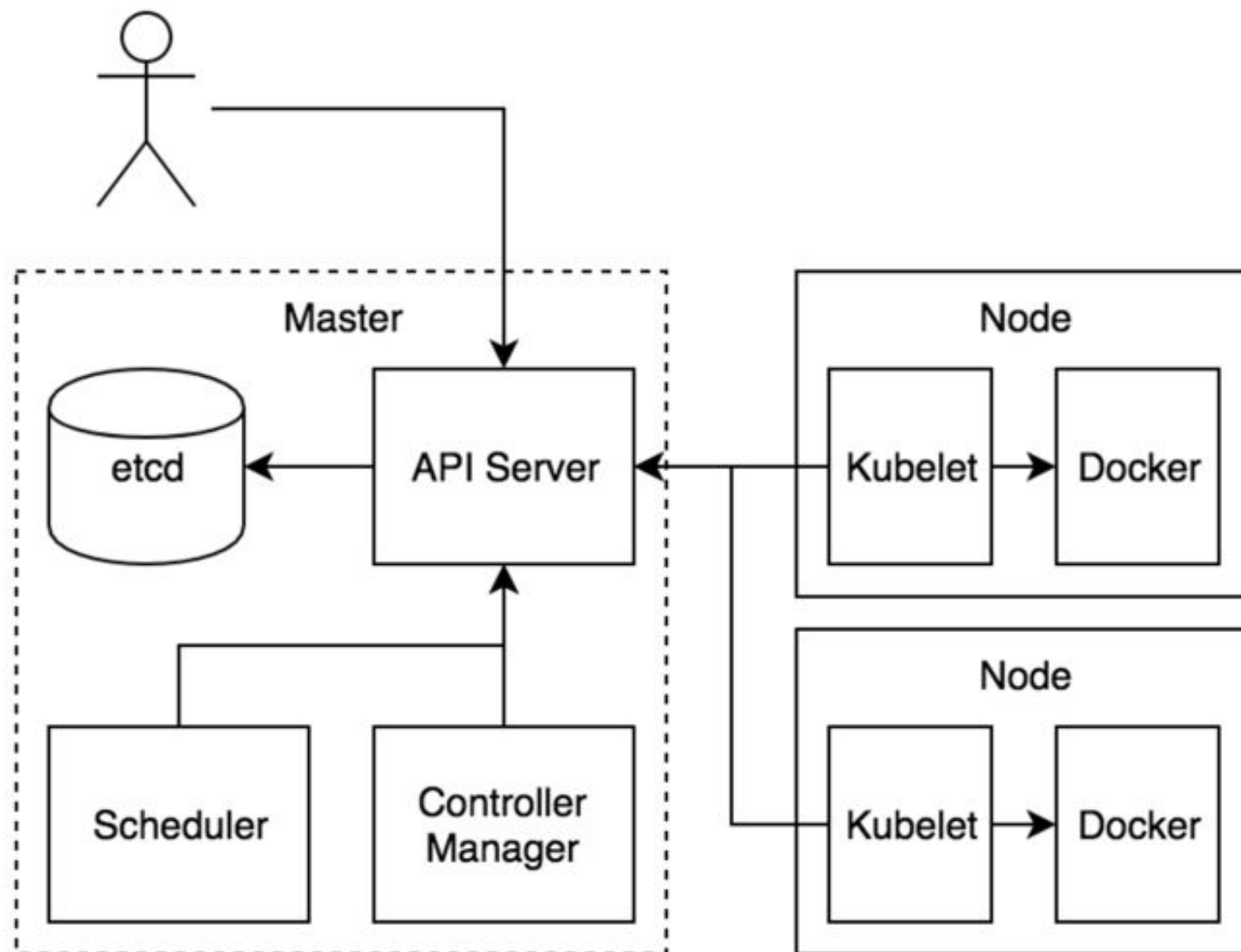


## 一、项目介绍

### 基于springboot + gradle + docker + Kubernetes



## 一、Master整体架构



## 一、整体架构

### ➤ Cluster state store

存储状态，提供 watch 的功能支持

### ➤ API Server

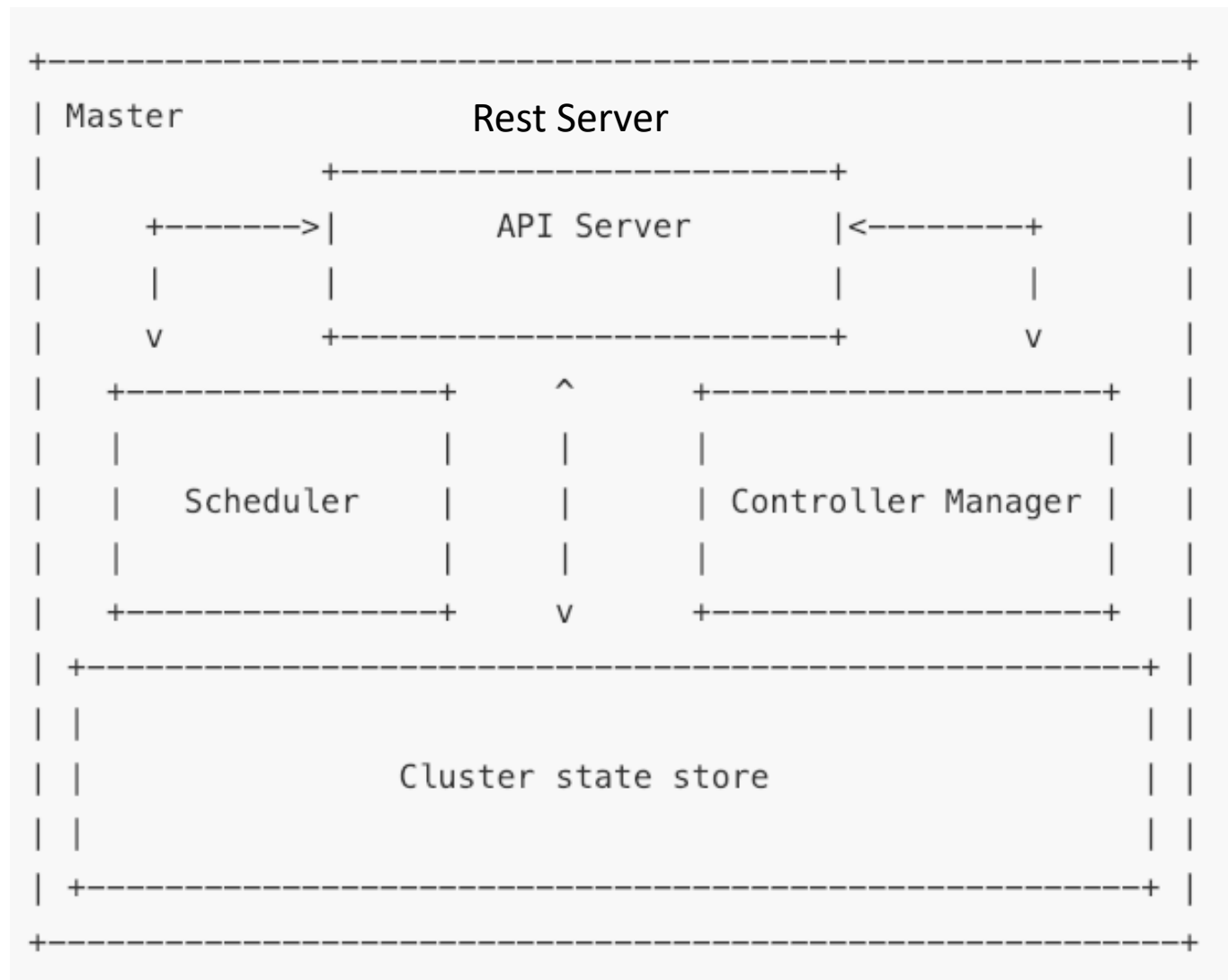
集群的网关

### ➤ Controller Manager

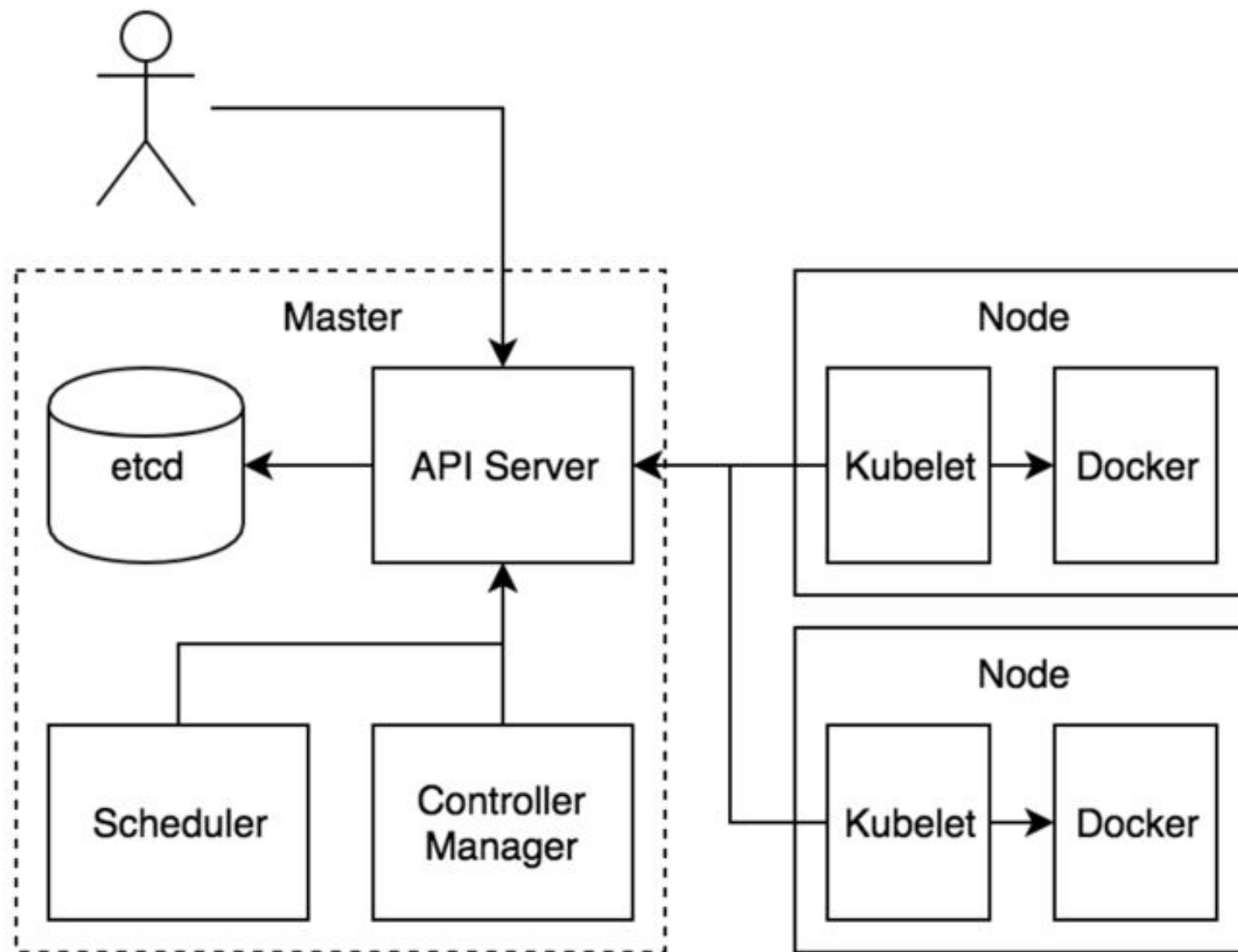
调节集群的状态

### ➤ Scheduler

集群的调度器



## 一、Master整体架构





## 二、Node整体架构

### ➤ Kubelet

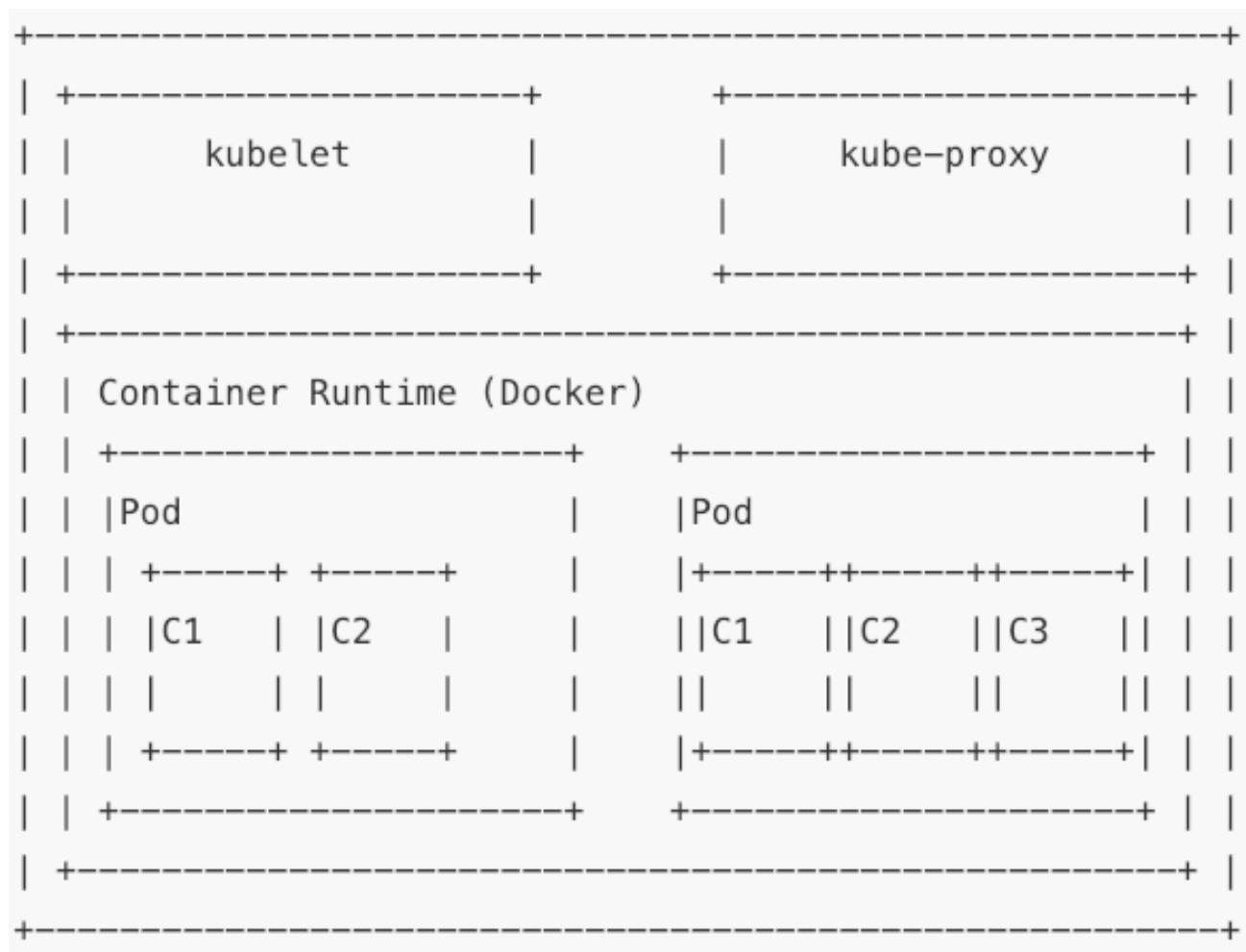
负责 Node 和 Pod 相关的管理任务。

### ➤ Container runtime

Docker

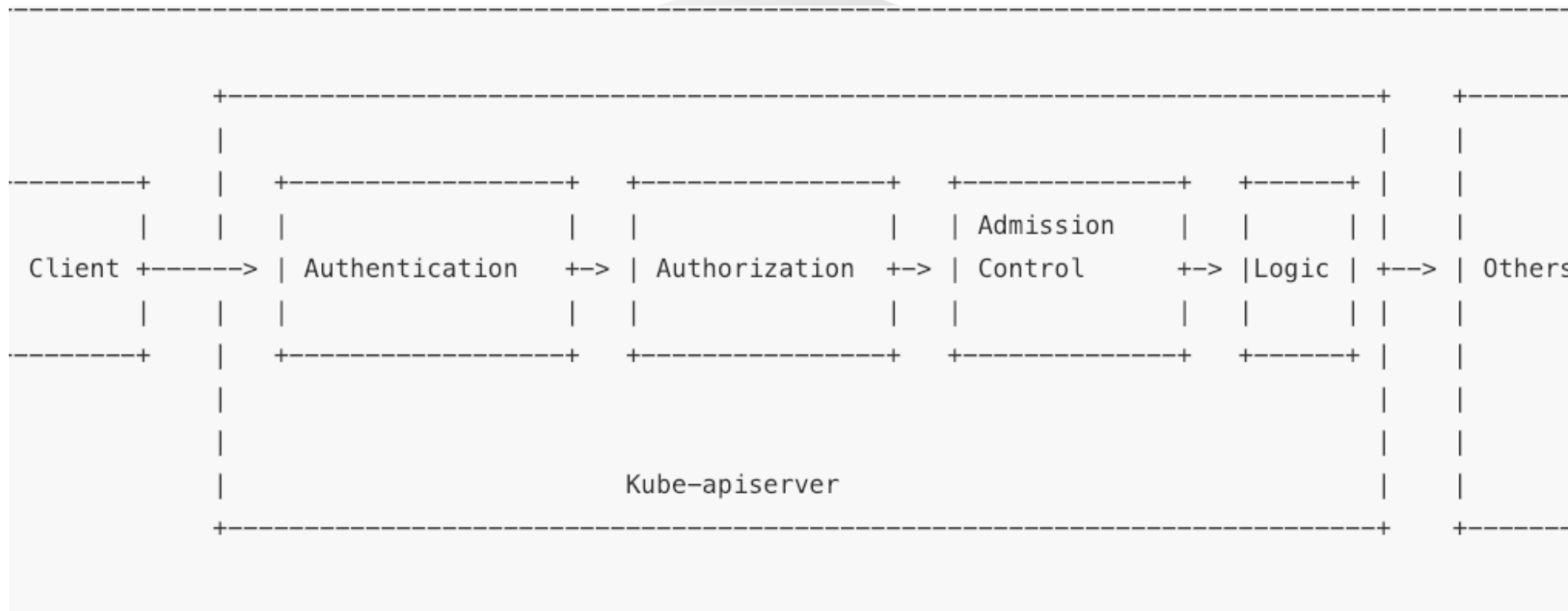
### ➤ Kube Proxy

提供一种代理的服务，让你可以通过 Service 访问到 Pod。



## 三、kube-apiserver

- 6443
- 认证
- 授权
- 准入
- 处理



# Kubernetes 核心组件深入分析

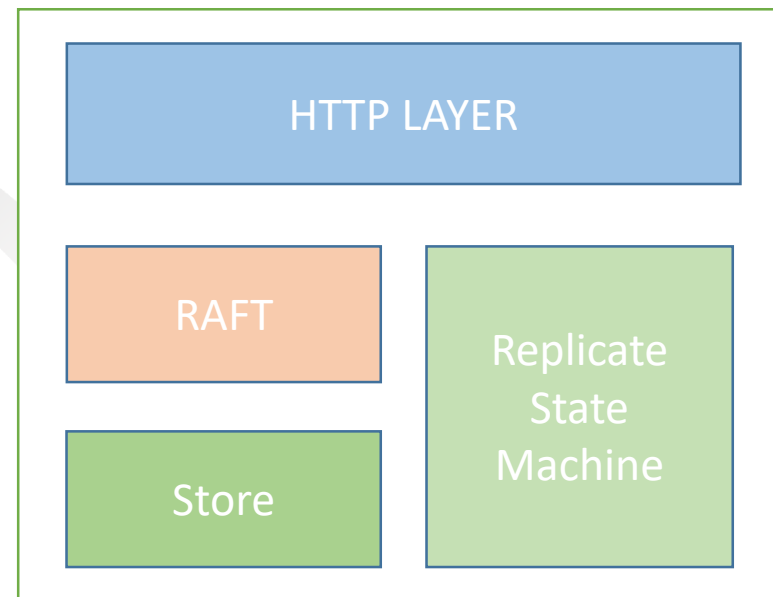
## 四、ETCD-SERVER整体架构

网路层：提供网路数据读写、监听服务端口，完成集群节点之间的数据通信，收发客户端数据

Raft模块:完整实现了Raft协议

存储模块:KV存储，WAL文件，SNAPSHOT管理

复制状态机：状态机的数据维护在内存中，定期持久化到磁盘，每次写请求会持久化到WAL文件，并根据写请求的内容修改状态机的数据



# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT

每个节点有三个状态，他们会在这三个状态之间进行变换。客户端只能从主节点写数据，从节点里读数据。

### 节点的三种状态



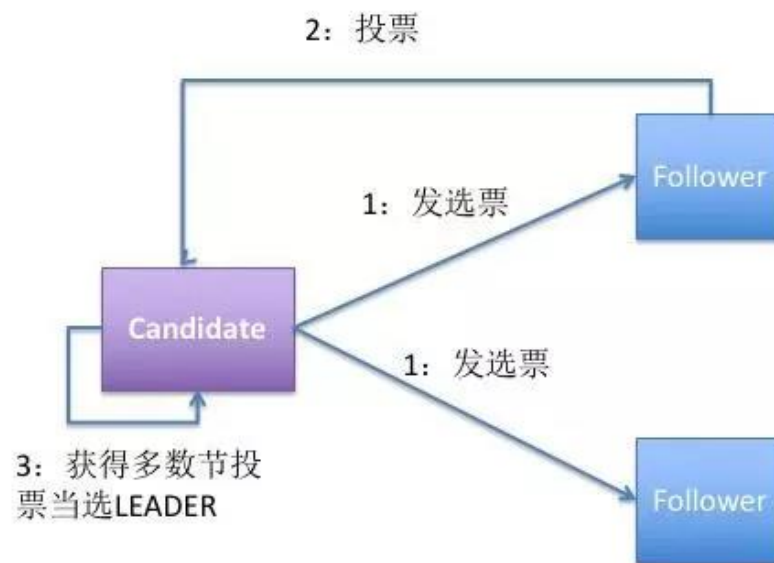
# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT(续)

### 选主流程

初始是Follower状态节点，等100-300MS没有收到LEADER节点的心跳就变候选人  
候选人给大家发选票，候选人获得大多数节点的选票就变成了LEADER节点。

### 选主流程





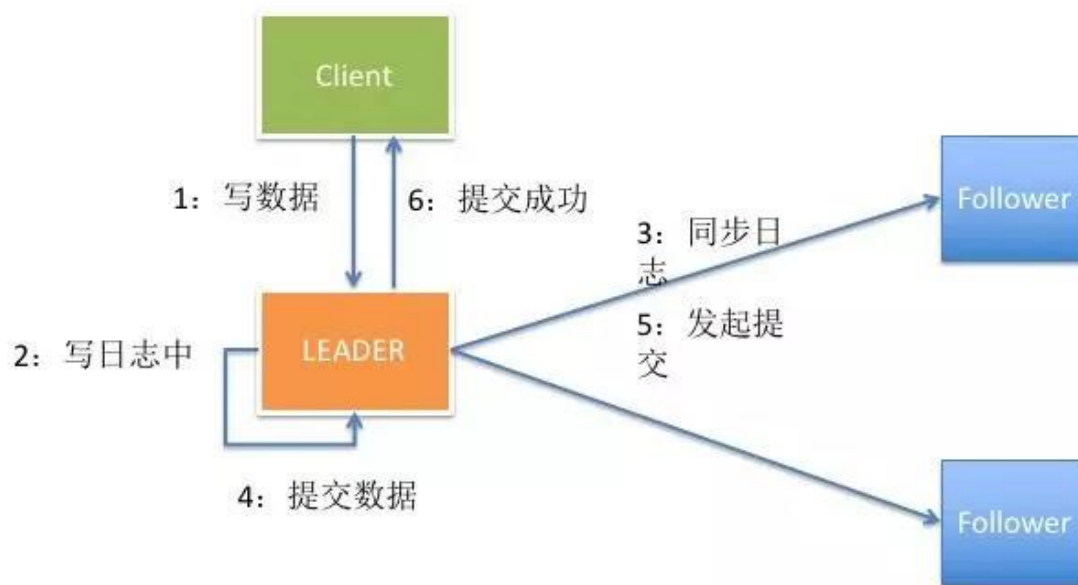
# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT(续)

### 日志复制流程

每次改变数据先记录日志，日志未提交不能改节点的数值。然后LEADER会复制数据给其他follower节点，并等大多数节点写日志成功再提交数据

### 日志复制流程



# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT(续)

### 选举超时

每个节点随机等150到300MS，如果时间到了就开始发选票，因为有的节点等的时间短，所以它会先发选票，从而当选成主节点。但是如果两个候选人获得的票一样多，它们之间就要打加时赛，这个时候又会重新随机等150到300MS，然后发选票，直到获得最多票当选成主节点

### 选举超时



# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT(续)

### 心跳超时

每个节点会记录主节点是谁，并且和主节点之间维持一个心跳超时时间，如果没有收到主节点回复，从节点就要重新选举候选人节点。

### 心跳超时



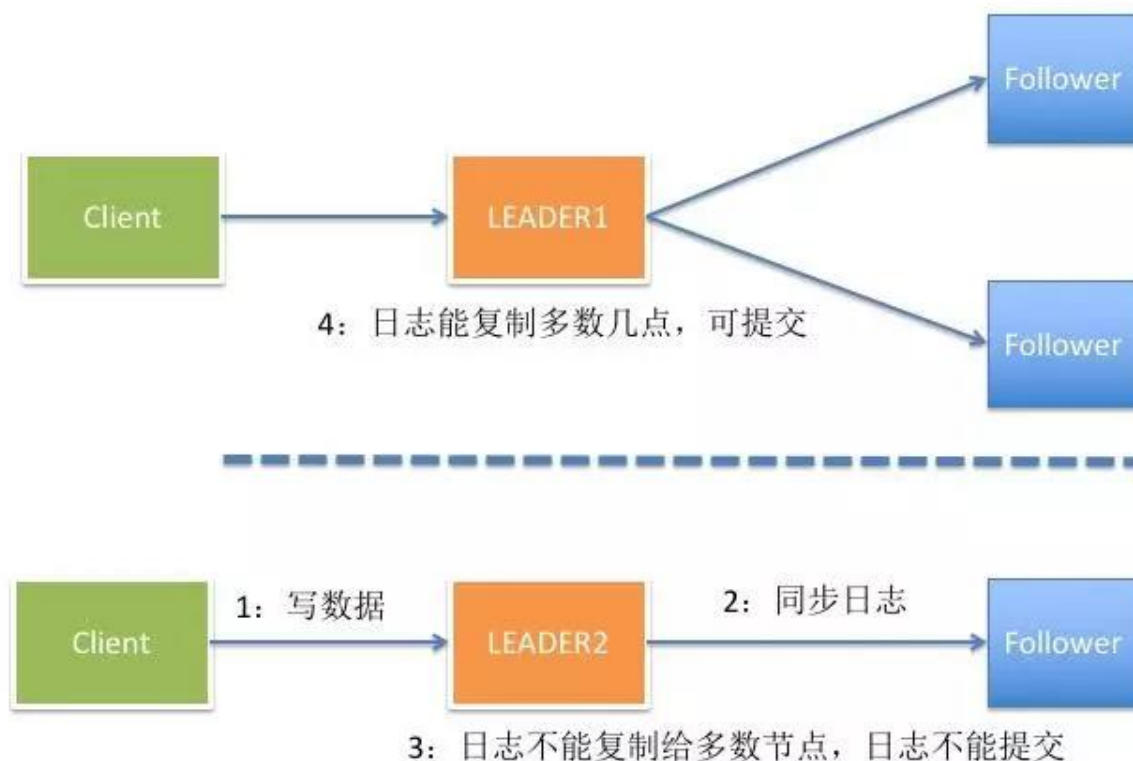
# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT(续)

### 集群中断

当集群之间的部分节点失去通讯时，主节点的日志不能复制给多个从节点就不能进行提交。

### 集群中断

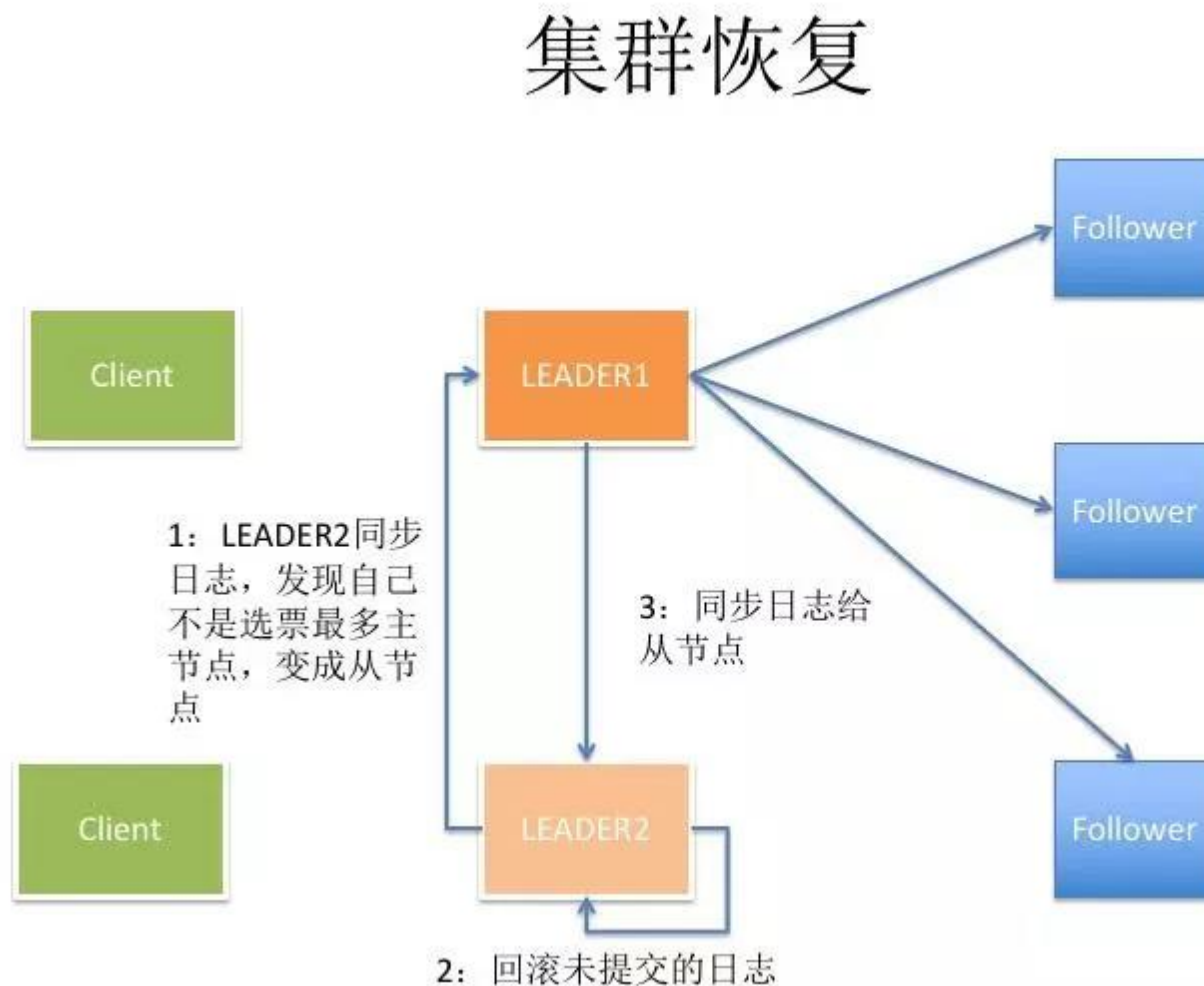


# Kubernetes 核心组件深入分析

## 五、分布式协议-RAFT(续)

### 集群恢复

当集群恢复之后，原来的主节点发现自己不是选票最多的节点，就会变成从节点，并回滚自己的日志，最后主节点会同步日志给从节点，保持主从数据的一致性。



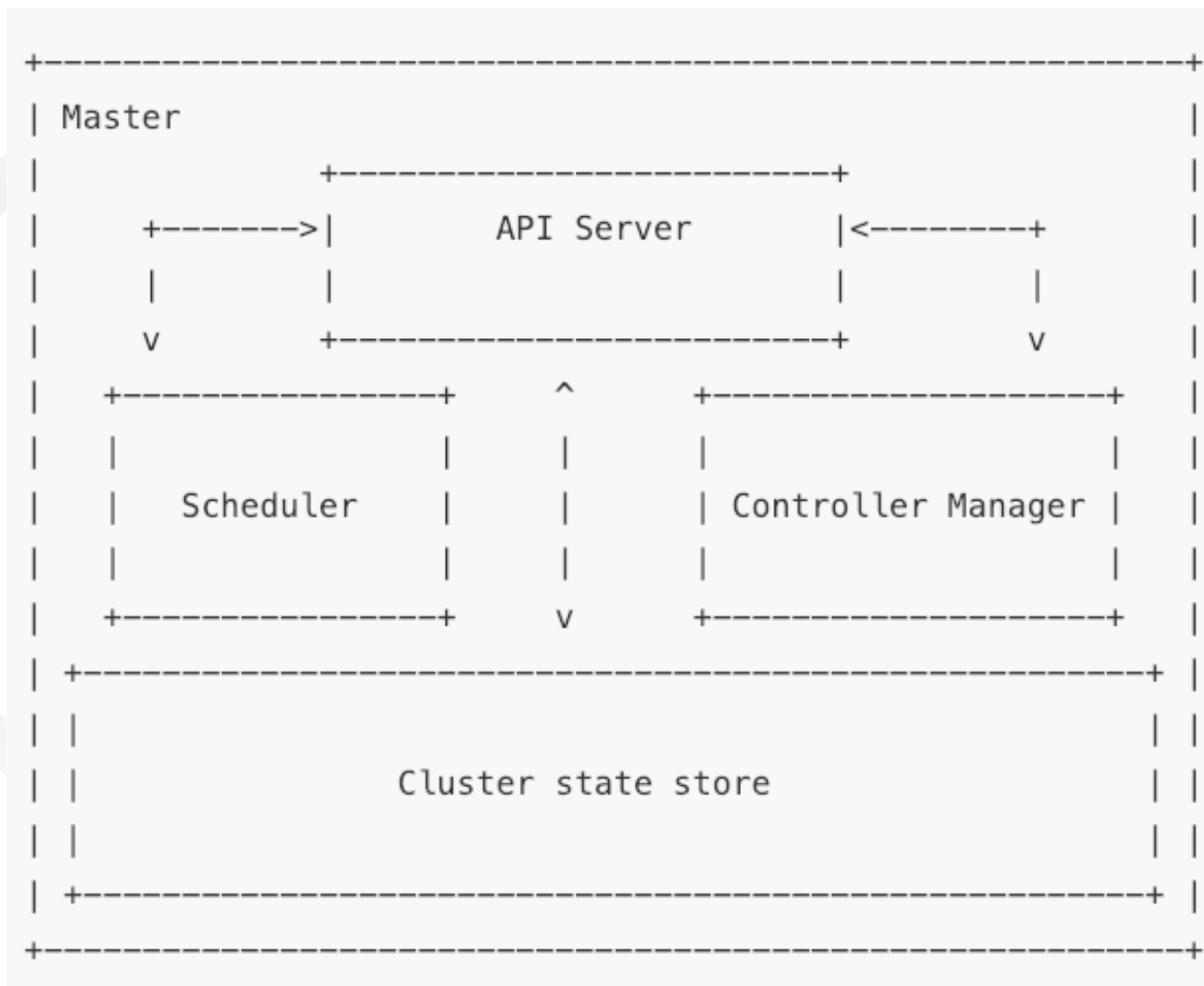


# Kubernetes 核心组件深入分析

## 六、Controller-Manager

kube-controller-manager 是一个嵌入了 K8S 核心控制循环的守护进程

- 控制循环：这里拆解为两部分：**控制** 和 **循环**，它所控制的是集群的状态；至于循环它当然是会有个循环间隔的，这里有个参数可以进行控制。
- 守护进程

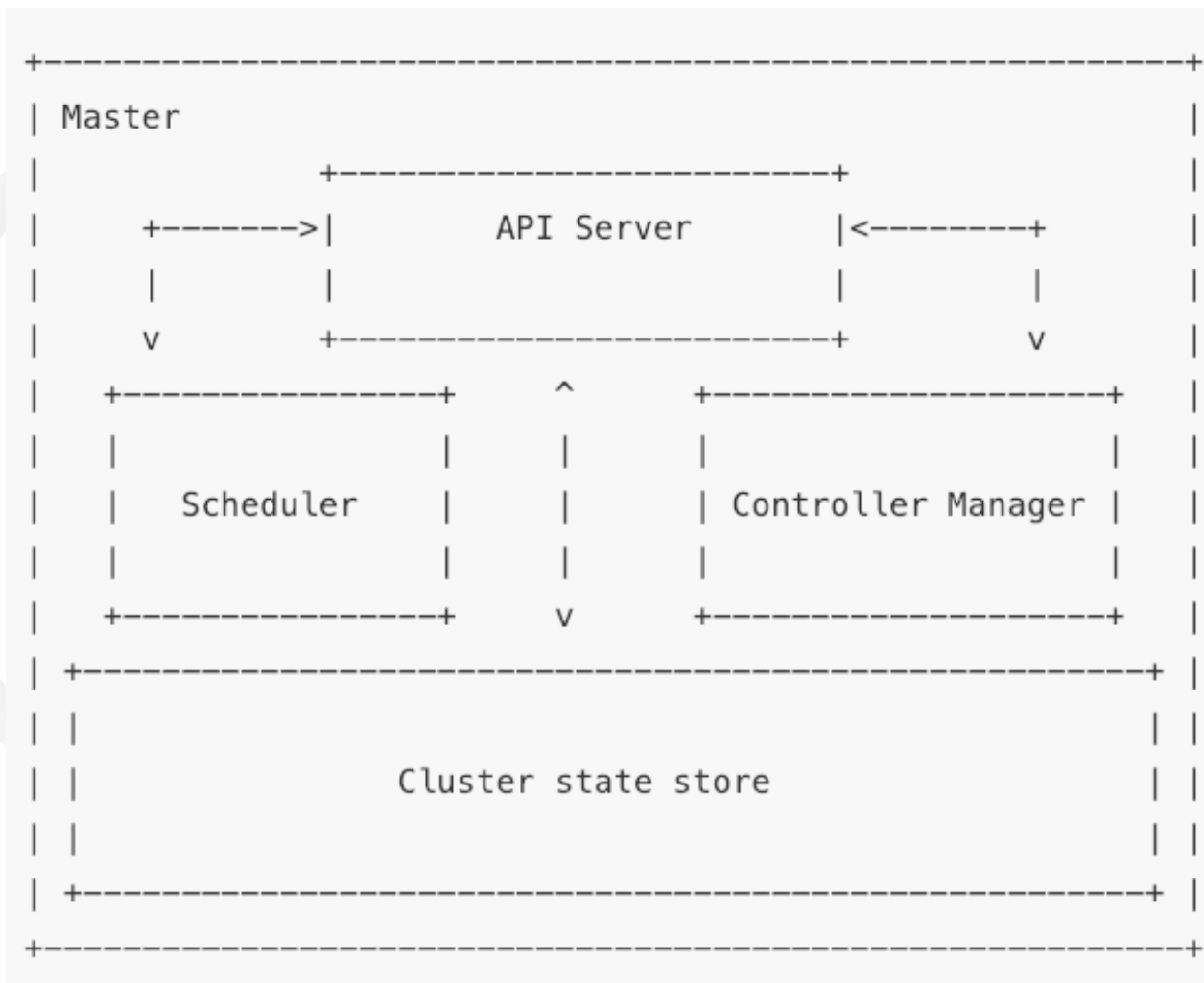


# Kubernetes 核心组件深入分析

## 七、kube-scheduler

kube-scheduler 的作用就是将待调度的 Pod 调度至最佳的 Node 上，而在这个过程中则需要根据不同的策略，考虑到 Node 的资源使用情况，比如端口，内存，存储等。

kube-scheduler 将处理阶段主要分为三个阶段  
Computing predicates, Prioritizing 和 Selecting host

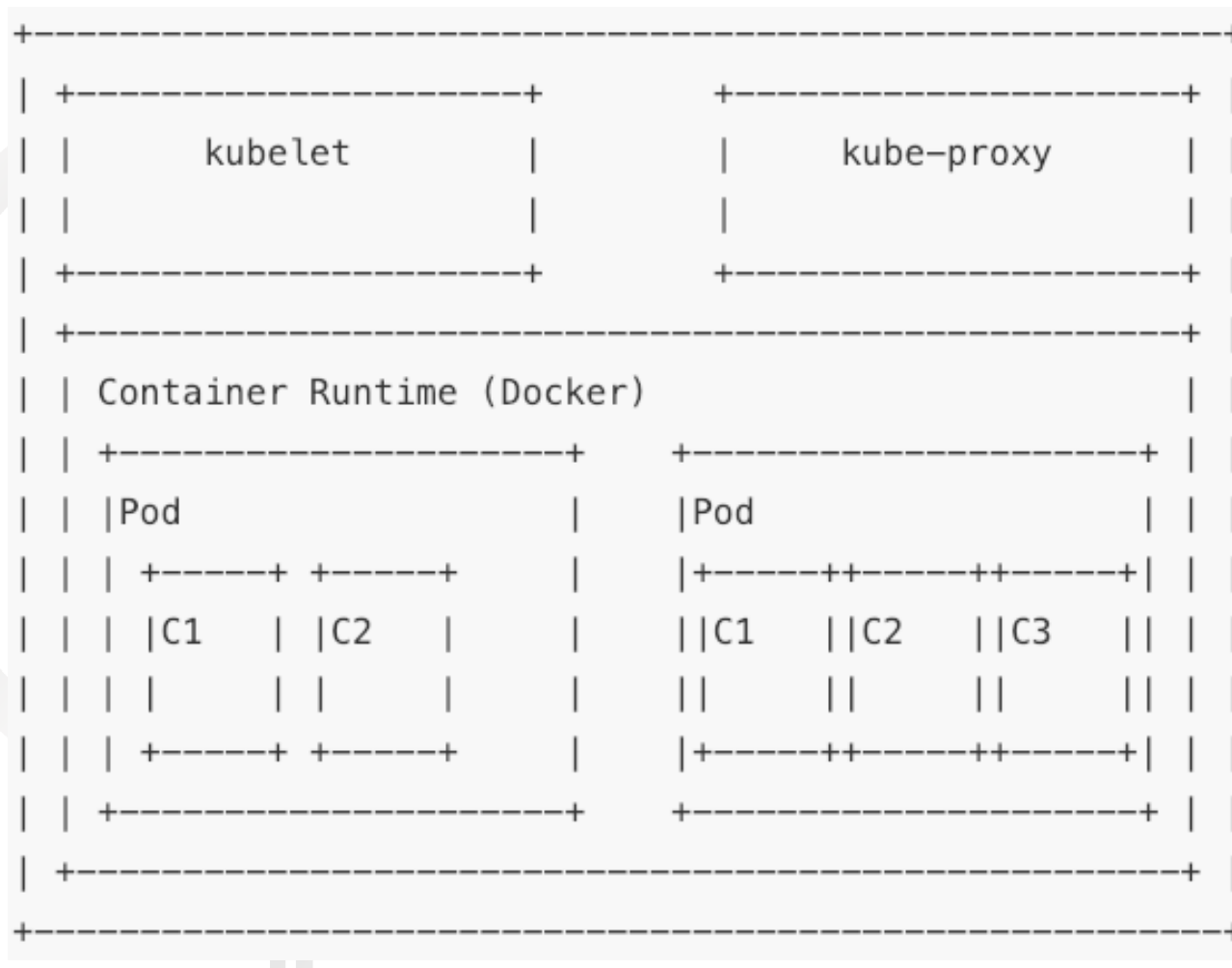


# Kubernetes 核心组件深入分析

## 八、Kubelet

**kubelet 是什么：**kubelet 便是 K8S 中的 agent，负责 Node 和 Pod 相关的管理任务。

**kubelet 有什么作用：**通常来讲 agent 这样的角色起到的作用首先便是要能够注册，让 server 端知道它的存在，所以这便是它的第一个作用：节点管理。

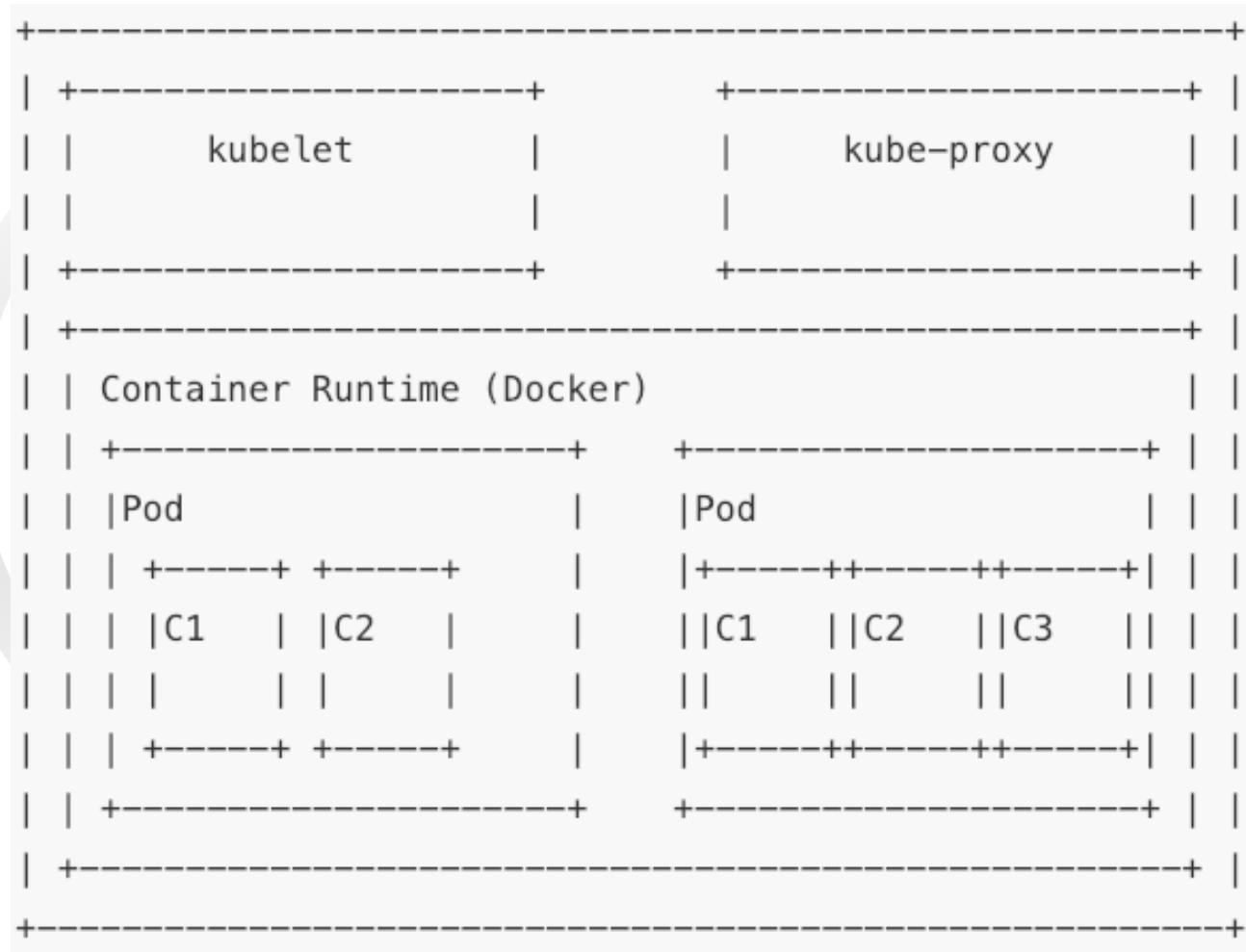


# Kubernetes 核心组件深入分析

## 九、 kube-proxy

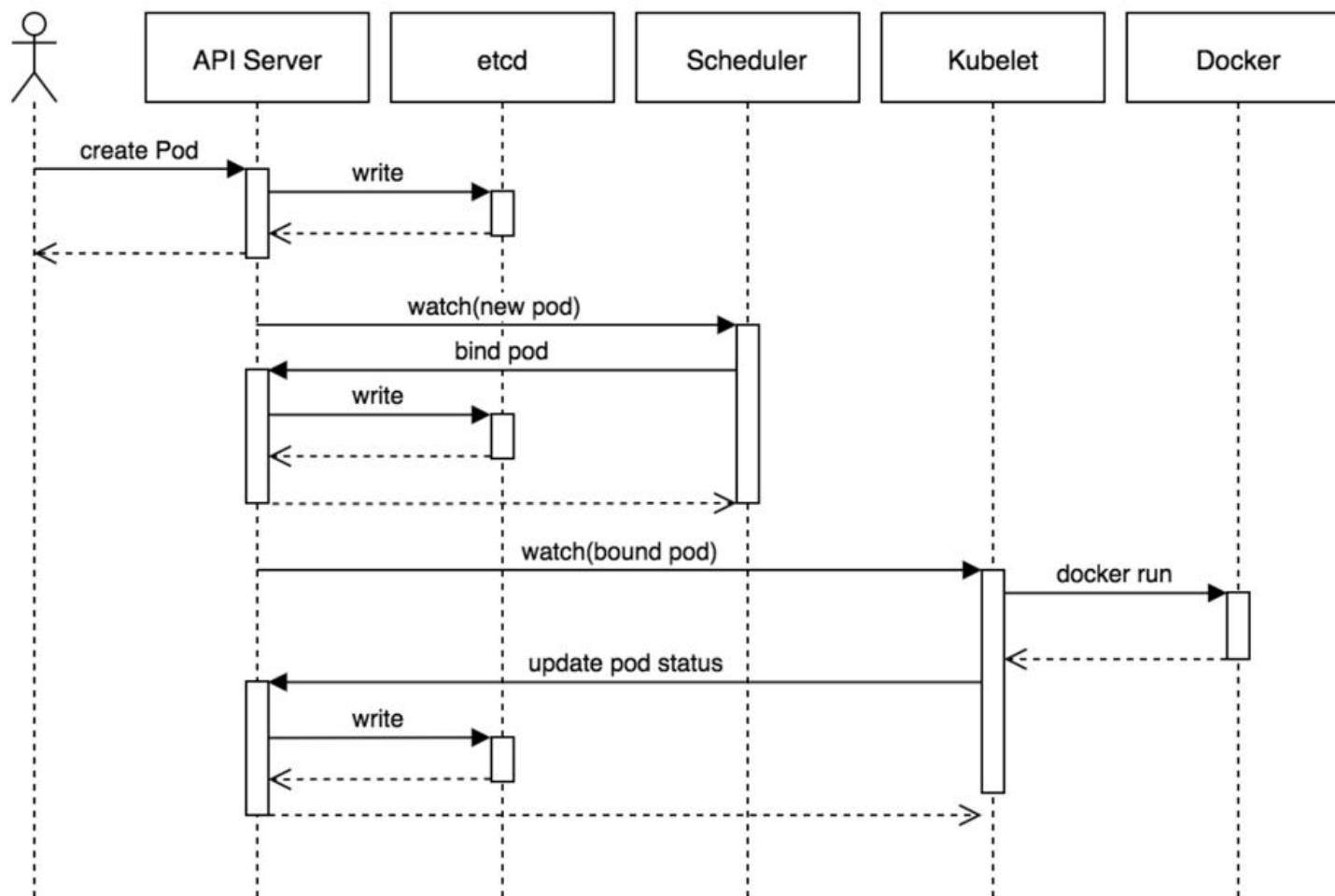
**kube-proxy 是什么：** kube-proxy 是 K8S 运行于每个 Node 上的网络代理组件，提供了 TCP 和 UDP 的连接转发支持。

**kube-proxy 如何工作：** iptables



# Kubernetes 核心组件深入分析

## 十、创建Pod的过程





# 谢谢观看

➤ XX老师QQ号: XXXXXXXX

上课老师、以及QQ号需要更新

