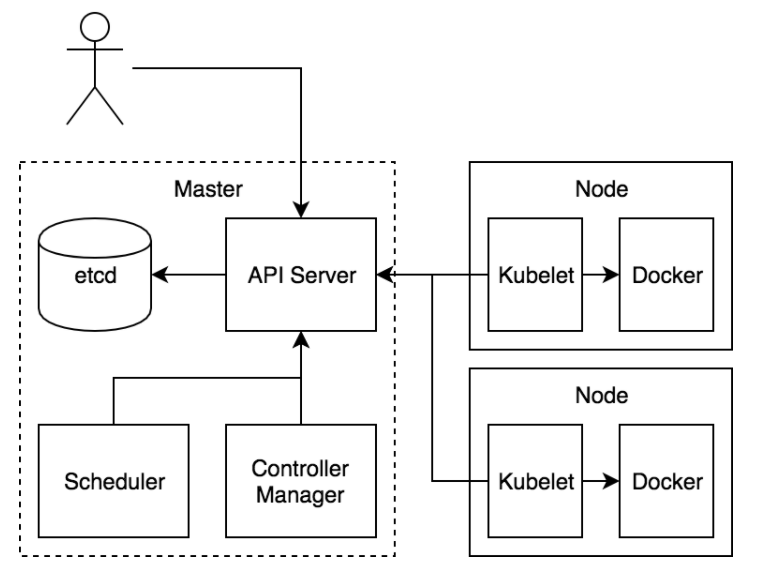
# 参考

K8S训练营：<https://www.qikqiak.com/k8strain/>

Prometheus与Grafana在kubernetes上的结合使用：B站搜索左侧关键字

# 基础

## 2.1 架构



Kubernetes 由 Master 和 Node 两种节点组成，这两种角色分别对应着控制节点和工作节点（可以理解为老板和员工）。

**Master 节点**由三个独立的组件组成，它们分别是

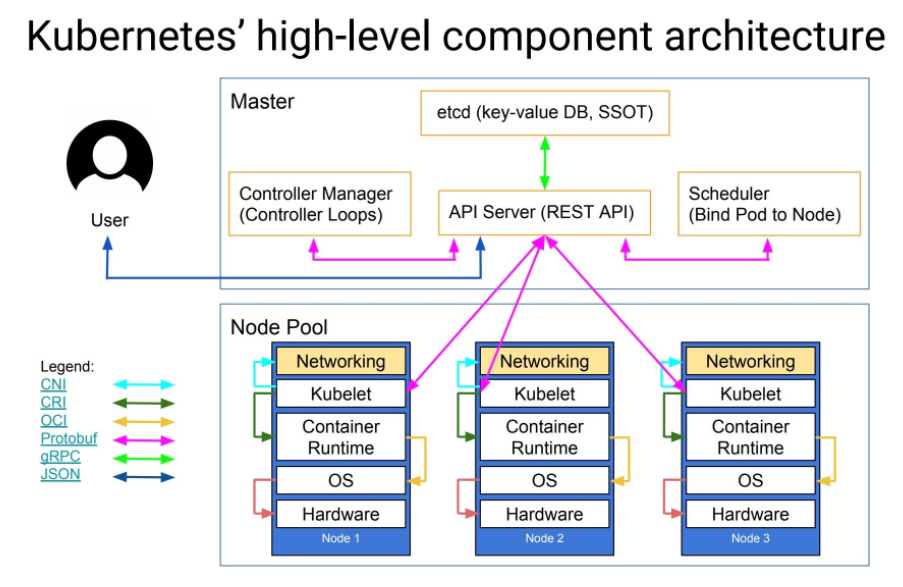
* 负责整个集群通信的 API 服务的 **kube-apiserver**
* 负责容器调度的 **kube-scheduler**
* 负责维护集群状态的 **kube-controller-manager** 组件。

整个集群的数据都是通过 kube-apiserver 保存到 etcd 数据库中的，而其他所有组件的通信也都是通过 kube-apiserver 和 etcd 数据库进行通信的，都不会直接和 etcd 进行通信。

**工作节点：**

最核心的组件就是 kubelet，当然还有底层的容器运行时，比如 Docker。其中 kubelet 就是主要来实现和底层的容器运行时进行通信的，这个通信的过程也被 Kubernetes 抽象成了一个 CRI（Container Runtime Interface）的远程调用接口，这个接口里面定义了容器运行时的所有标准操作，比如创建容器、删除容器等等，只是目前 kubelet 里面内置了 Docker 关于这个 CRI 实现的一个 shim，所以如果我们底层是 Docker 容器的话就不需要单独安装这个 CRI 的实现的组件了。其他的容器运行时是需要提供这样的一个接口实现组件的。所以对于 Kubernetes 来说他根本不关心你部署的到底是什么容器运行时，只要你这个容器运行时可以实现 CRI 接口就可以被 Kubernetes 来管理。

kubelet 的另外一个重要功能就是调用网络插件（CNI）和存储插件（CSI）为容器配置网络和存储功能，同样的 kubelet 也是把这两个重要功能通过接口暴露给外部了，所以如果我们想要实现自己的网络插件，只需要使用 CNI 就可以很方便的对接到 Kubernetes 集群当中去。



**更多每个组件的详细介绍**，参考<https://www.qikqiak.com/k8strain/> 组件章节

## 2.2 核心资源对象

Kubernetes 作为容器编排引擎，在 Kubernetes 集群中抽象了很多集群内部的资源对象，我们可以通过这些资源对象去操作容器的编排工作。

### Pod

### Label

### Namespace

### Deployment

### Service