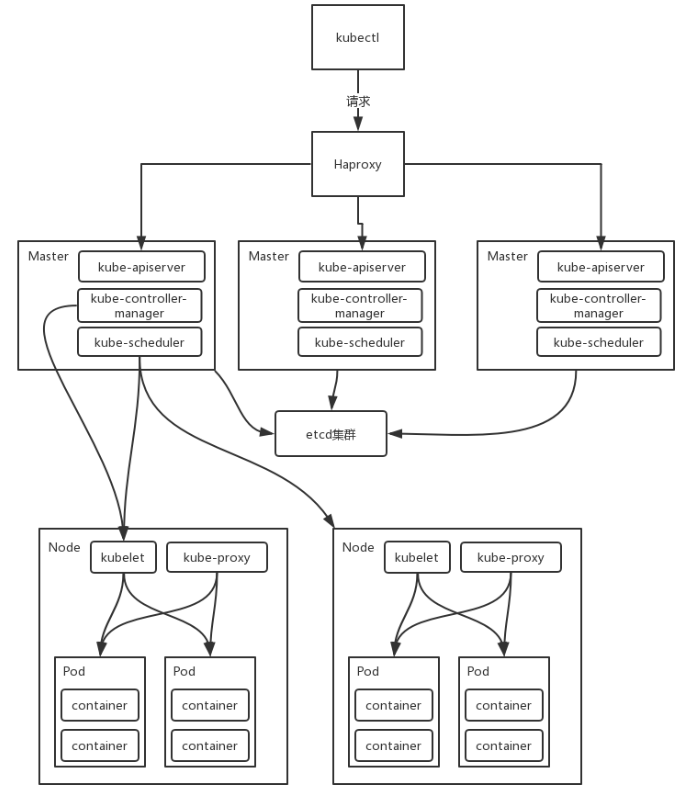
Cluster(集群)

1.

Kubernetes (K8S， 将ubernete这八个字母缩写成8) 是Google在2014年发布的一个**开源项目**，使用Go语言，用于自动化容器化应用程序的部署、扩展和管理。Kubernetes通常结合docker容器工作，并且整合多个运行着docker容器的主机集群。

2.

Kubernetes的整体架构：



Kubernetes中有Master节点以及Node节点。

🡪**Master节点**：集群控制节点，负责整个集群的管理和控制. 主要的组件为：

kube-apiserver组件：提供了http rest接口，是kubernetes里所有资源增删改查的唯一入口. 所有组件之间通信都需要通过 ETCD，实际上，他们并不是直接访问 ETCD，而是访问一个代理，这个代理是通过标准的RESTFul API，重新封装了对 ETCD 接口调用，除此之外，这个代理还实现了一些附加功能，比如身份的认证、缓存等。这个代理就是 API Server。

kube-controller-manager组件：kubernetes所有资源的自动化控制中心.于etcd进行交流，发现需要改变的状态并且启动去满足需求。

kube-scheduler组件：负责整个资源的调度. Scheduler监听到有新的资源需要调度（新的Pod），就会根据整个集群的状态，给Pod分配到具体的节点上.

etcd: 是用来存储整个 Kubernetes 的集群状态的，它除了具备状态存储的功能（以key-value的形式），还有事件监听和订阅、Leader选举的功能，所谓事件监听和订阅，各个其他组件通信，都并不是互相调用 API 来完成的，而是把状态写入 ETCD（相当于写入一个消息），其他组件通过监听 ETCD 的状态的的变化（相当于订阅消息），然后做后续的处理，然后再一次把更新的数据写入 ETCD。所谓 Leader 选举，其它一些组件比如 Scheduler，为了做实现高可用，通过 ETCD 从多个（通常是3个）实例里面选举出来一个做Master，其他都是Standby。

Kubectl: Kubectl: 是一个命令行工具，它会调用 API Server发送请求写入状态到ETCD，或者查询ETCD的状态。

🡪**Node节点**：工作节点，当一个节点宕机，其上的负载会被Master自动转移到其他工作节点上去。主要的组件为：

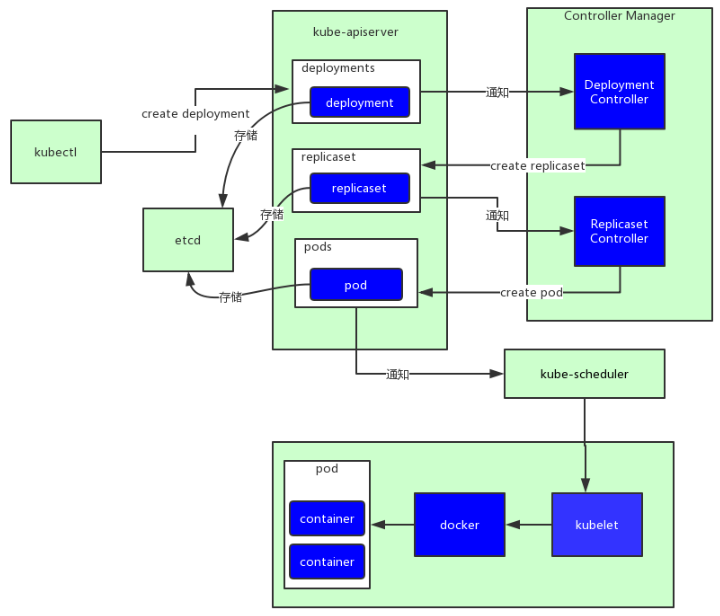
kubelet组件：在每个Node上运行的代理，负责跟Master节点协作来并控制docker如何工作，确保容器在pod中运行。它运行在Node节点上，监听ETCD中的Pod信息，发现有分配给它所在节点的Pod需要运行，就在节点上运行相应的Pod，并且把状态更新回到ETCD。

kube-proxy组件：实现kubernetes的service与pod中的容器通信和实现负载均衡机制。

Pods: pod是Kubernetes处理的基本单元，包含多个containers在里面，在同一个Pod中的containers相互配合，共享一样的生命周期，环境，以及IP空间。通常Pod中一个container完成主要的workload其他container负责一些相关的helper辅助功能。一些用户不需要直接管理pod,而是通过其他高层组件间接的管理Pod.

3.

Kubernetes简易的完整调度过程：



通过kubectl调用apiserver接口创建一个deployment资源，apiserver接收到命令后，校验完配置信息没问题后先把数据存储到etcd之后会立即返回成功信息到kubectl客户端，并通知deployment controller manager。

deployment controller manager收到通知后，会去调apiserver接口去创建一个replicaset资源并存储数据到etcd，并通知replicaset controller manager

replicaset controller manager收到通知后，同样会去调apiserver接口去创建一个pod资源并存储数据到etcd，并通知scheduler

scheduler收到通知后通过一定的调度策略找到对应的Node节点上的kubelet代理，再由kubelet调用docker接口去创建相应的pod和container。

4.