



kubernetes

# 如何更优雅的部署kubernetes集群

赵梓旗

- kubernetes 部署方案的发展
- kubernetes 部署 kubernetes 集群
- kubernetes 解析
- kubernetes 使用常见问题
- kubernetes 使用技巧

# kubernetes 部署方案的发展

# kubernetes 部署难

kube-up.sh 一键部署k8s集群脚本

缺陷

- 初始化配置复杂
- 功能随着kubernetes特性的增加变的庞杂
- shell编写，理解代码困难，很难定位问题

# kubernetes 部署难

## Setup

### [Picking the Right Solution](#)

#### ▶ [Independent Solutions](#)

#### ▶ [Hosted Solutions](#)

#### ▶ [Turn-key Cloud Solutions](#)

#### ▶ [Custom Solutions](#)

#### [Installing Addons](#)

#### [Configuring Kubernetes with Salt](#)

#### [Building Large Clusters](#)

#### [Running in Multiple Zones](#)

#### [Building High-Availability Clusters](#)

#### [Downloading or Building Kubernetes](#)

## Picking the Right Solution

Kubernetes can run on various platforms: from your laptop, to VMs on a cloud provider, to rack of bare metal servers. The effort required to set up a cluster varies from running a single command to crafting your own customized cluster. Use this guide to choose a solution that fits your needs.

If you just want to “kick the tires” on Kubernetes, use the [local Docker-based solution using MiniKube](#).

When you are ready to scale up to more machines and higher availability, a [hosted solution](#) is the easiest to create and maintain.

[Turnkey cloud solutions](#) require only a few commands to create and cover a wide range of cloud providers.

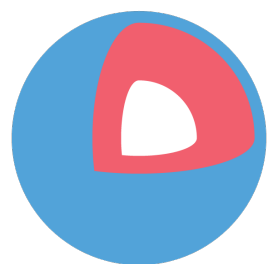
If you already have a way to configure hosting resources, use [kubeadm](#) to easily bring up a

# 成熟的社区解决方案

<https://github.com/coreos/tectonic-installer>

<https://github.com/kubernetes-incubator/kubespray> (Ex Kargo)

<https://github.com/apprenda/kismatic>



Core OS



KISMATIC

# 成熟的社区解决方案

缺陷:

- 学习曲线高
- 灵活性有限
- 社区力量有限

# kubeadm

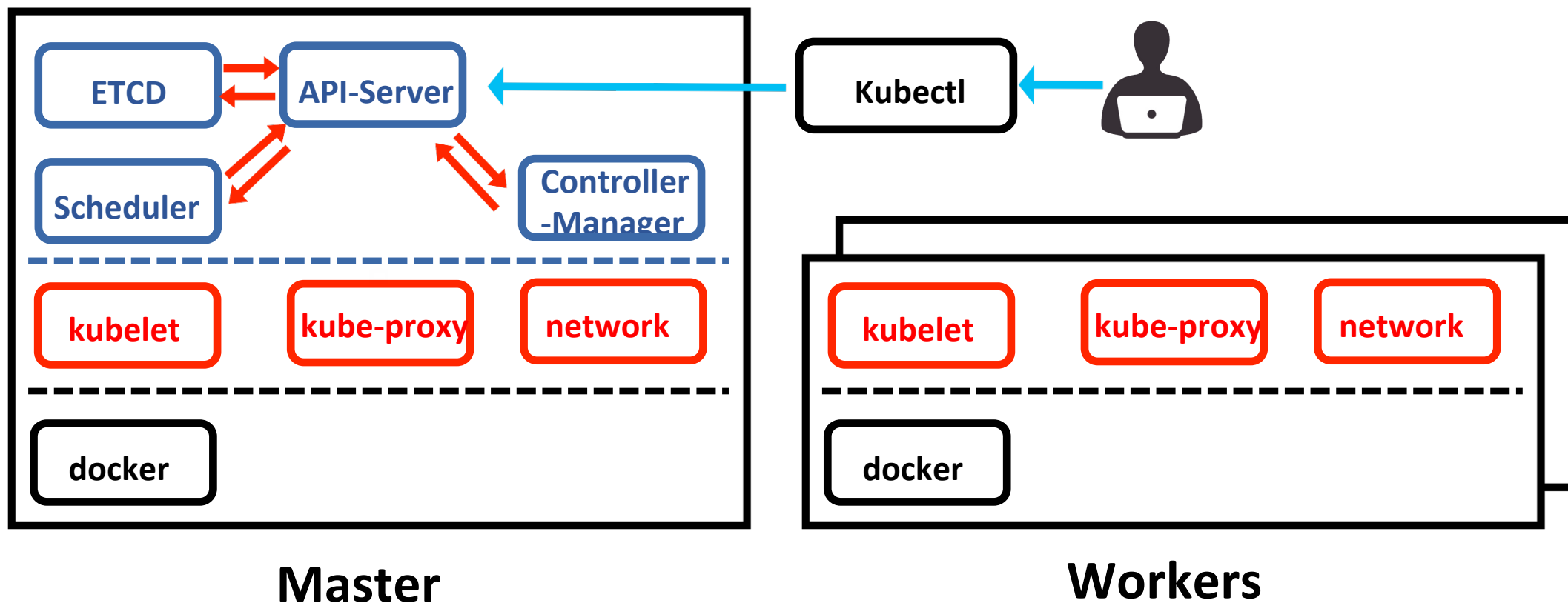
# kubeadm

- ✓ 下一代部署管理工具。 **Not kube-up.sh 2.0 !!**
- ✓ 使用方式友好
- ✓ 配置灵活性高
- ✓ 社区力量雄厚， SIG Cluster LifeCycle 专门维护开发
- ✓ 全面的测试
- ✓ 话语权



# kubeadm 部署 kubernetes 流程

# kubernetes 架构



# kubeadm 部署 kubernetes 流程

## 1. 准备正确配置的机器

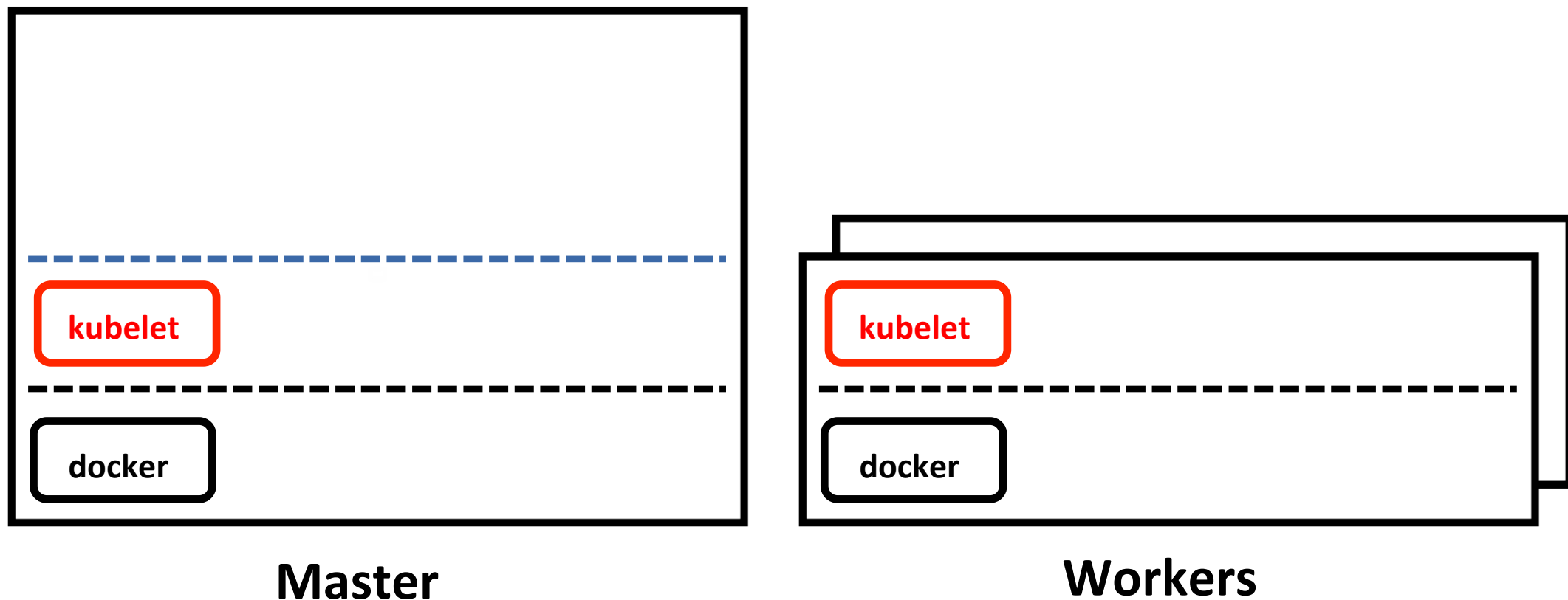
- Ubuntu 16.04 / Debian 9 / CentOS 7 / RHEL 7 / Fedora 25 / HypriotOS v1.0.1
- 至少1GB内存
- 集群内的物理机节点之间网络互通
- 物理机，虚拟机，云主机都可以

# kubeadm 部署 kubernetes 流程

## 2. 安装必要的软件(deb/rpm)

- docker
  - before 1.8, 1.12.06- recommended
  - after 1.8, 1.17.03- recommended
- kubelet (systemd 管理, 开机自启动)
- kubeadm
- kubectl (作为依赖安装)

# kubeadm 部署 kubernetes 流程



# kubeadm 部署 kubernetes 流程

## 3. 启动master节点

启动方法:

在 **master** 节点上, 运行

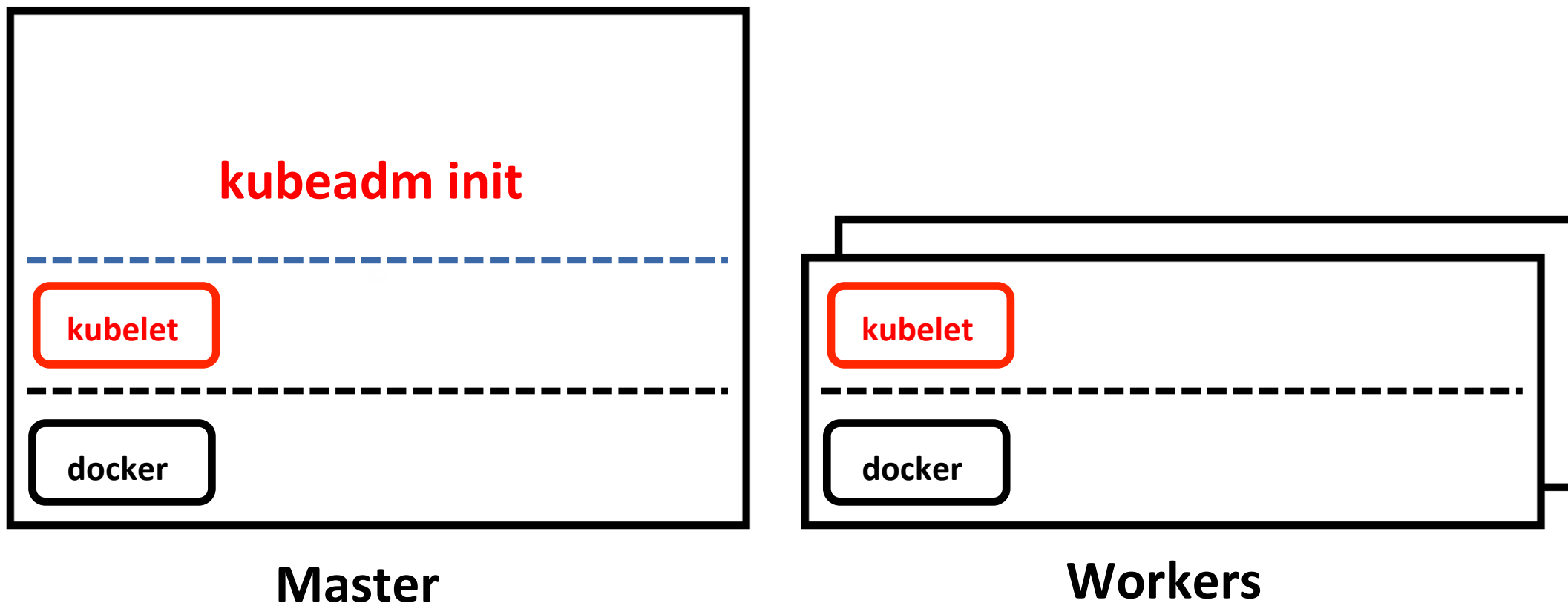
**kubeadm init**

启动master相关组件, add-on组件

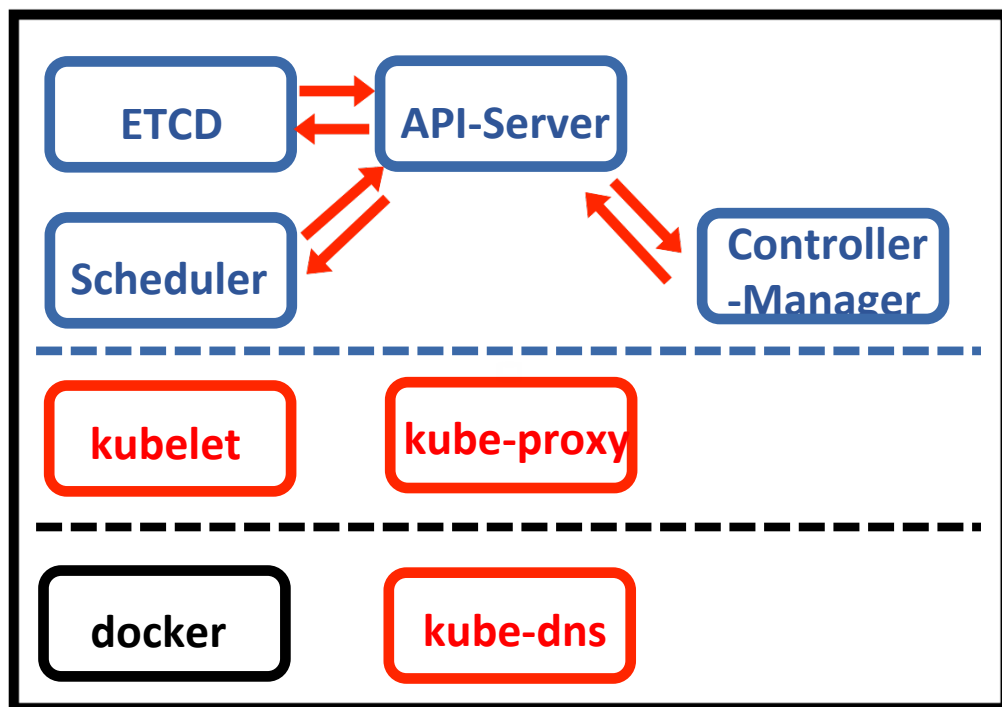
终端输出:

```
kubeadm join --token xxxxxx.xxxxxxxxxxxx master_ip:master_port
```

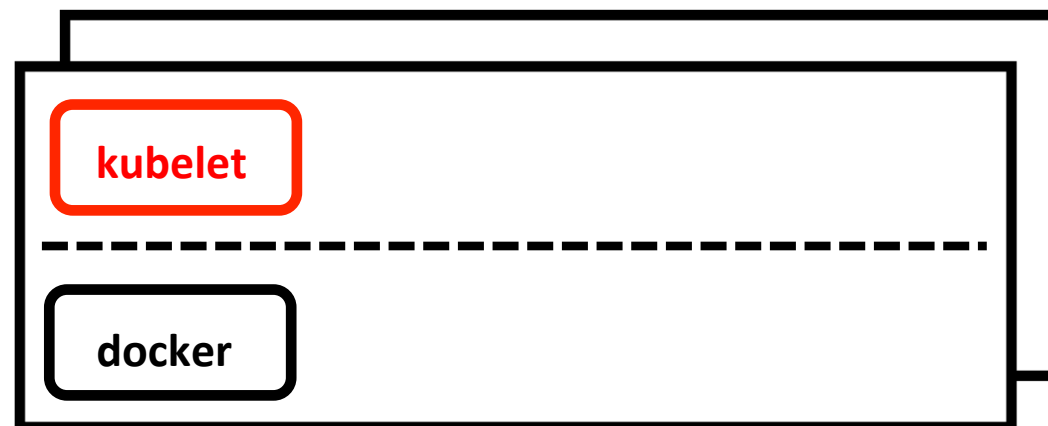
# kubeadm 部署 kubernetes 流程



# kubeadm 部署 kubernetes 流程



Master



Workers



# kubeadm 部署 kubernetes 流程

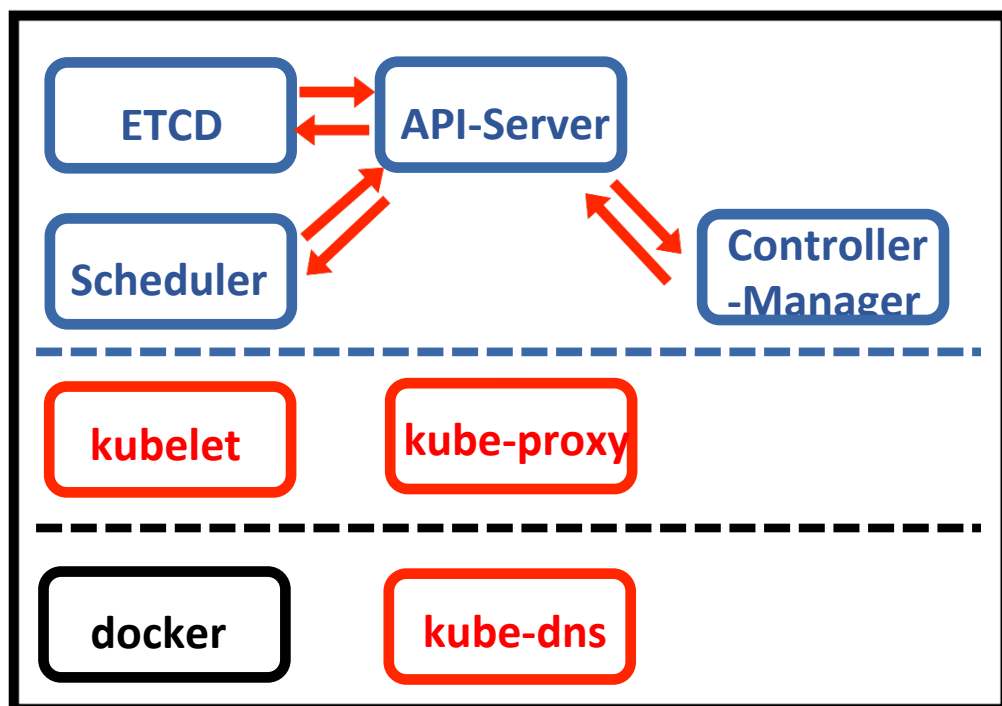
## 4. 将worker节点加入集群

加入方法:

在 **worker** 节点上, 运行

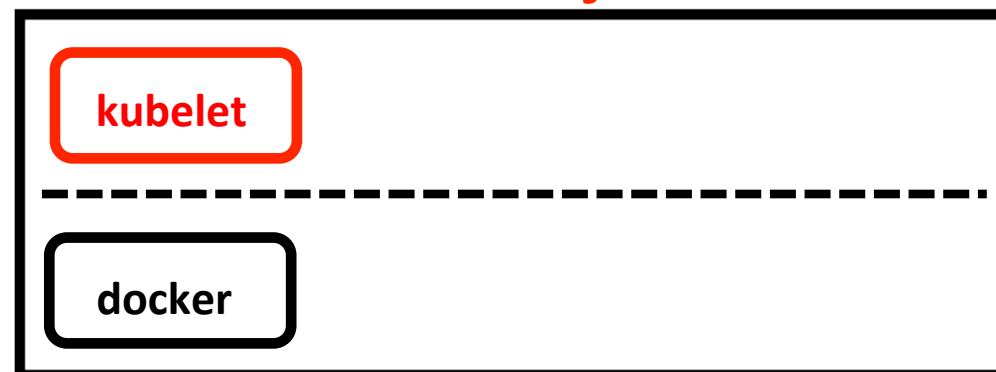
```
kubeadm join --token xxxxxx.xxxxxxxxxxxx  
master_ip:master_port
```

# kubeadm 部署 kubernetes 流程



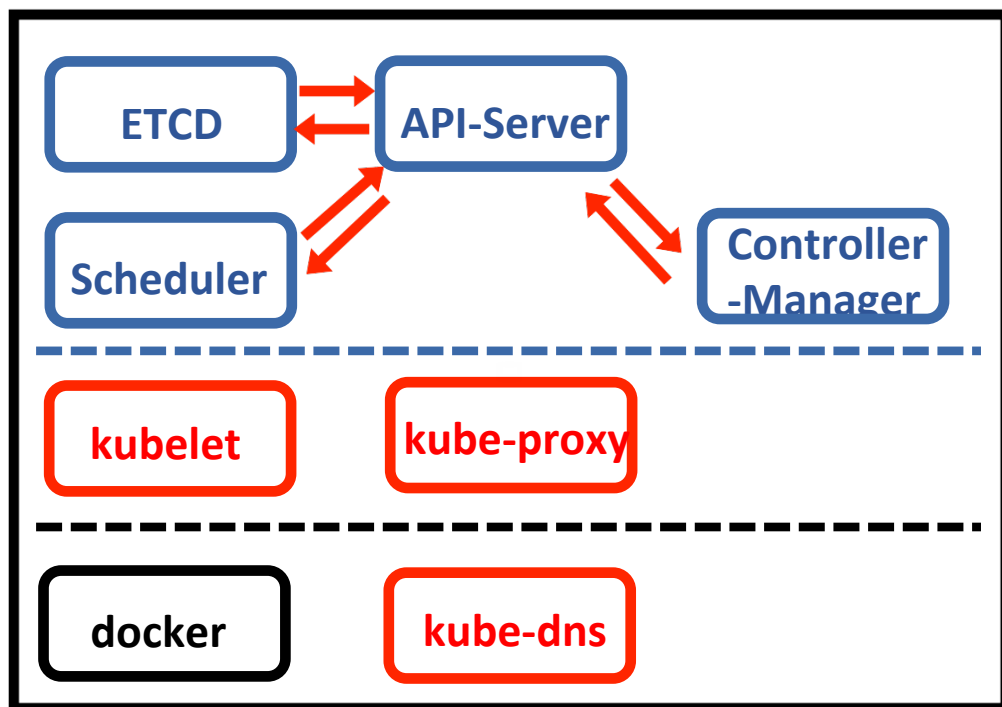
Master

kubeadm join

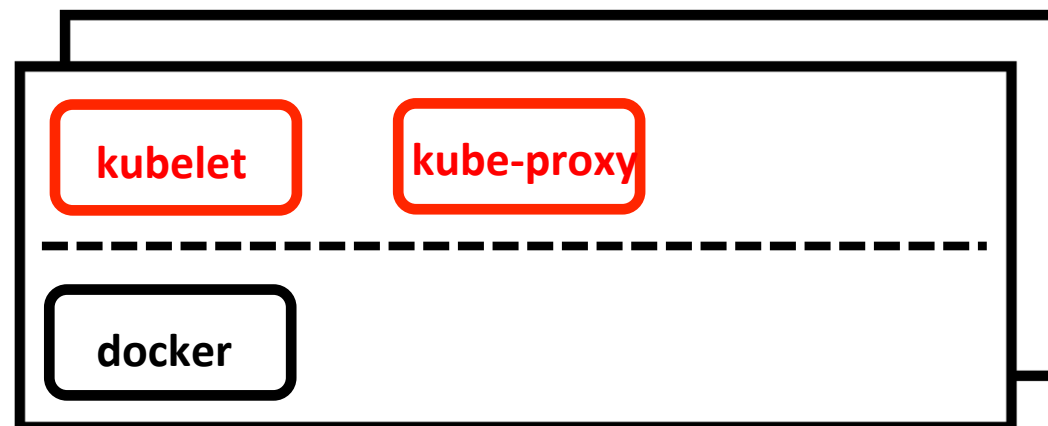


Workers

# kubeadm 部署 kubernetes 流程



Master **NotReady**



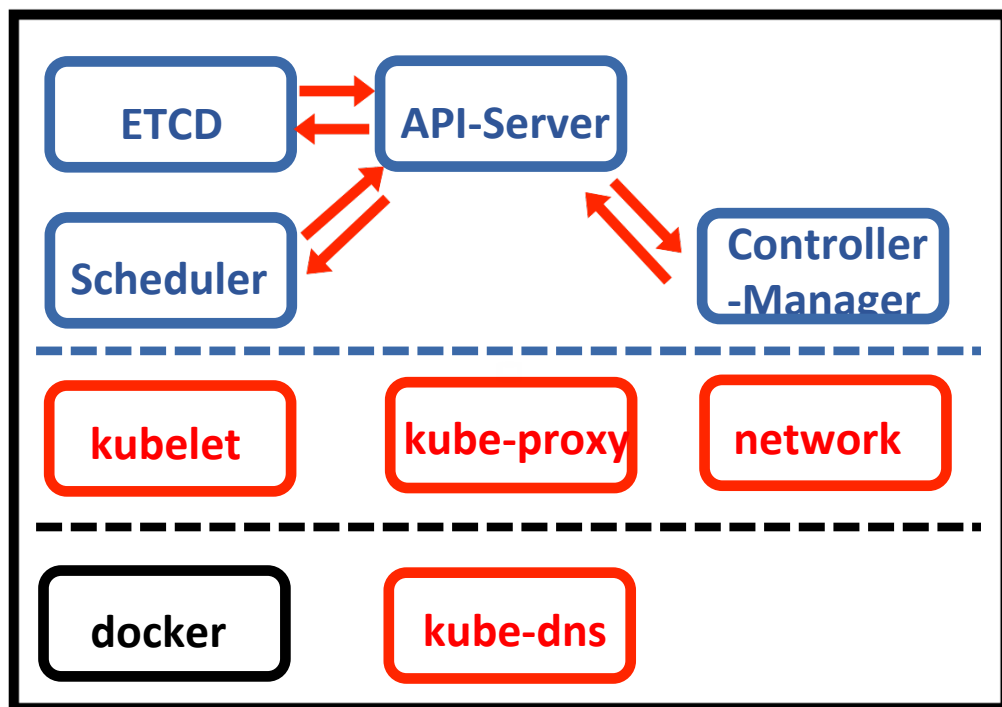
Workers **NotReady**

# kubeadm 部署 kubernetes 流程

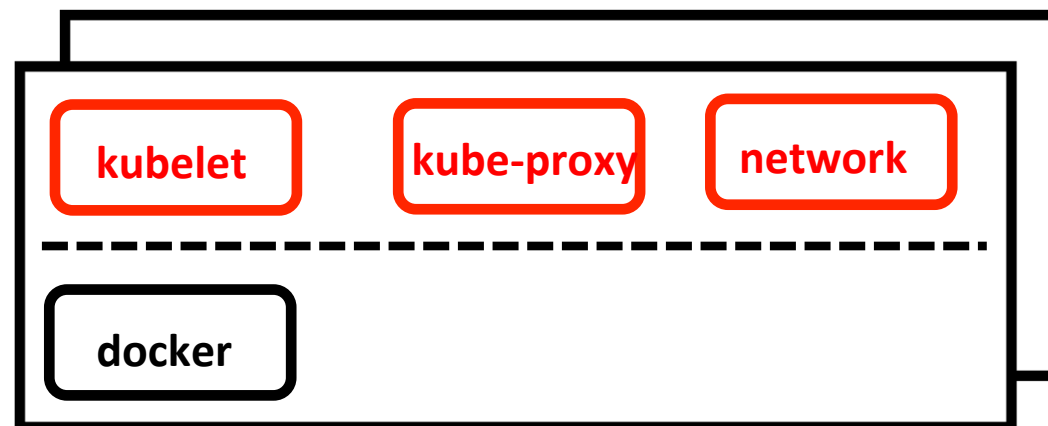
## 5. 部署网络

- 知名网络方案已经能够把组件全部运行在k8s中
- calico可以通过yaml文件直接部署到集群中
- `kubectl apply -f https://docs.projectcalico.org/v2.6/getting-started/kubernetes/installation/hosted/kubeadm/1.6/calico.yaml`

# kubeadm 部署 kubernetes 流程



Master Ready



Workers Ready

## 该集群的特性

- 集群工作在安全模式下，所有通讯都是通过TLS加密的，并且任何想要和该集群通讯的用户都必须通过客户端证书(kubeconfig)认证
- 集群重要组件除kubelet之外都采用容器化部署
- 集群只有单个master节点
- 自带kube-dns组件

# kubeadm 解析

# kubeadm 剖析

kubeadm init

启动一个kubernetes master物理机

kubeadm join

添加节点到kubernetes 集群中



# kubeadm init 解析

## 1. 前期机器检查

- k8s组件监听端口是否被绑定
- cgroups特性是否正确配置
- kubelet是否已经被安装且通过systemd管理

...

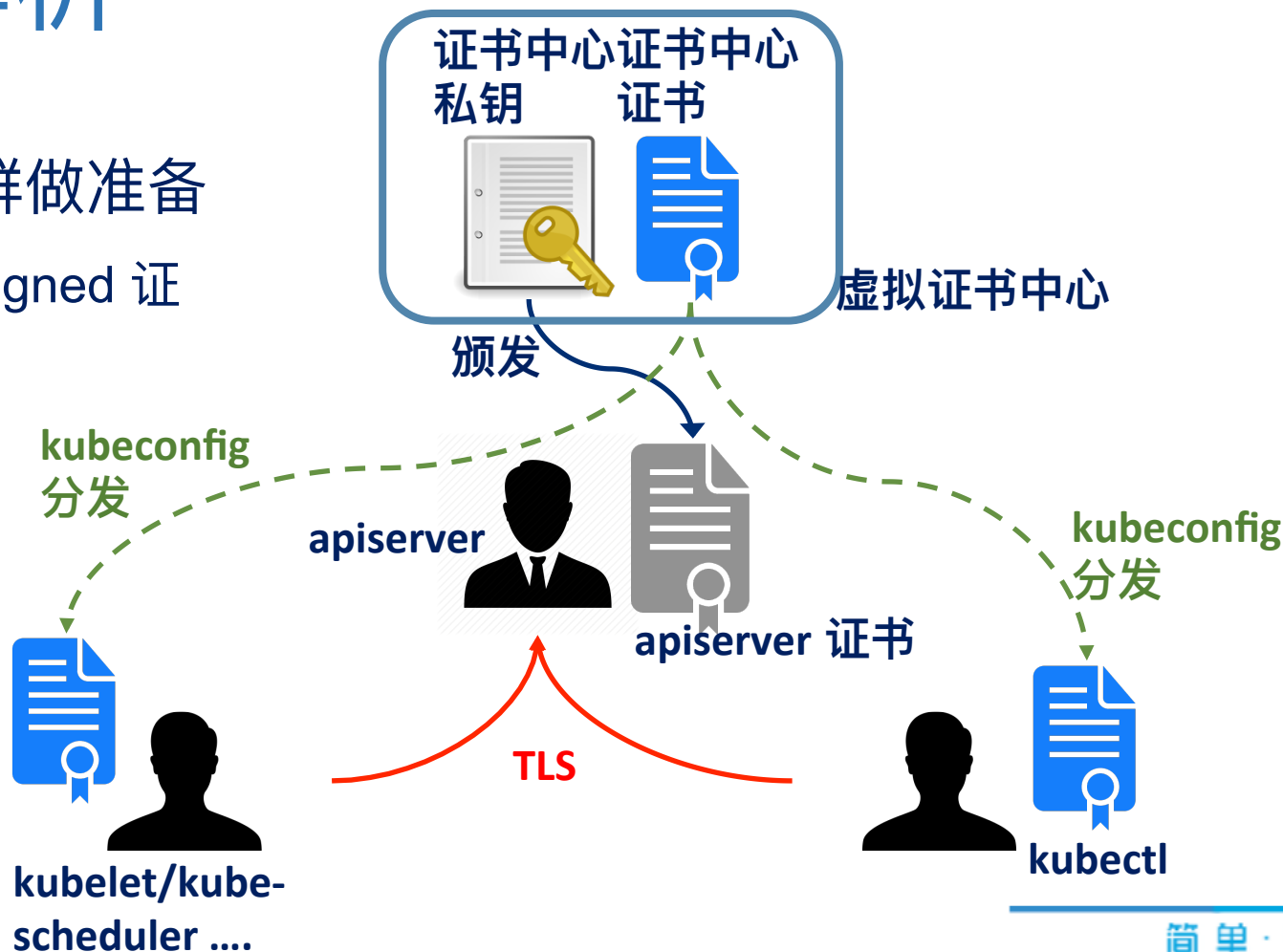
## 2. 生成验证token

添加worker节点时会使用

# kubeadm init 解析

## 3. 为创建一个安全的k8s集群做准备

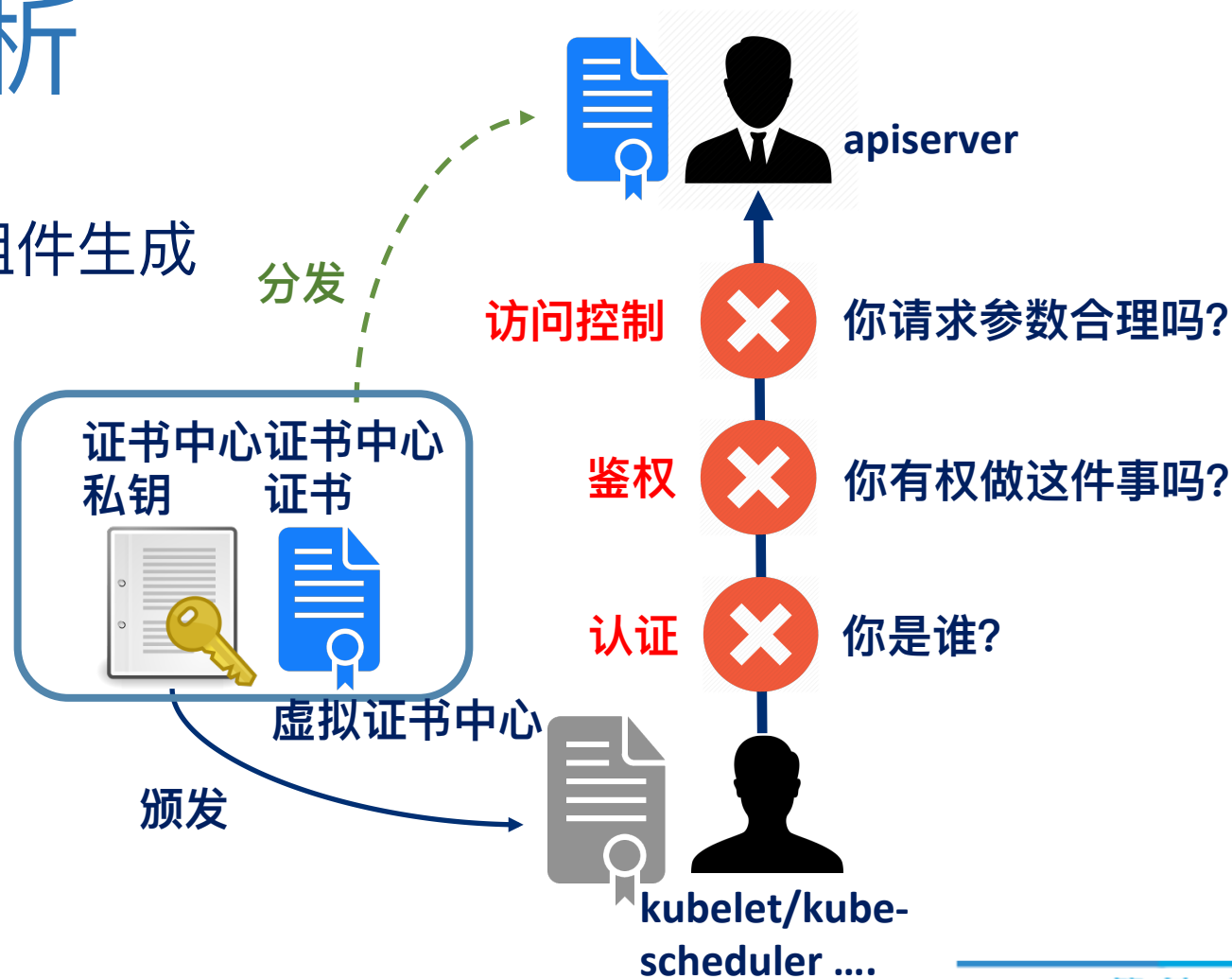
- kubeadm 创建一个 Self-Signed 证书中心
- 经典的服务器认证机制



# kubeadm init 解析

## 4. 给和apiserver组件通讯的组件生成 kubeconfig文件

- kubelet
- kube-apiserver
- kube-controller-manager
- kube-scheduler
- administrator



# kubeadm init 解析

## 5. 启动master相关组件

- kube-apiserver/controller-manager/scheduler/etcd
- 文件形式写入/etc/kubernetes/manifests
- static pod

# kubeadm init 解析

## 6. 为master节点添加label和taint

- Label: `node-role.kubernetes.io/master:`
- Taint: `node-role.kubernetes.io/master:NoSchedule`

## 7. 创建两个add on组件

- `kube-proxy daemonset running`
- `kube-dns deployment pending`

# kubeadm init 解析

## 8. 为worker节点的安全添加做准备

- 双向信任 (利用Token)
- Token必须保密



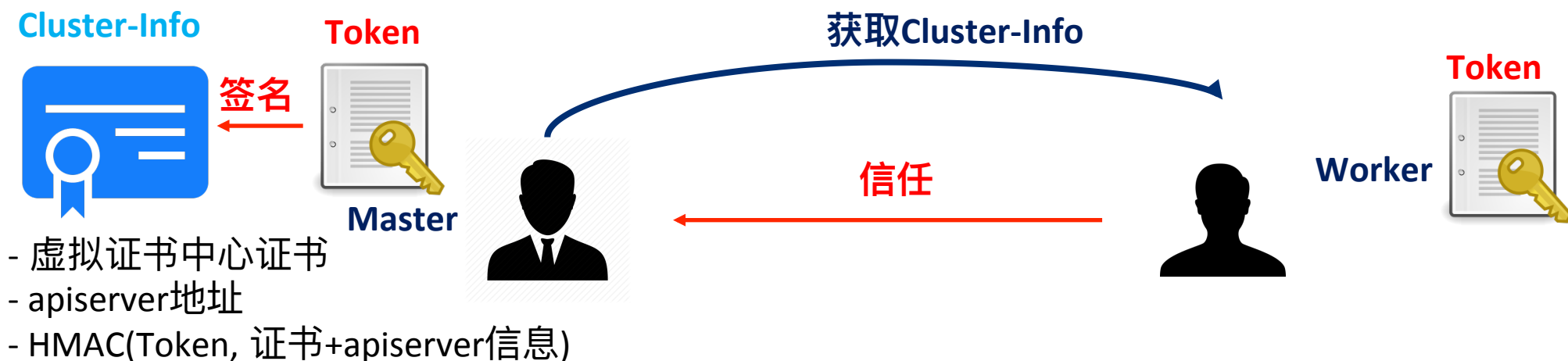
# kubeadm init/join 解析

## 8. 为worker节点的安全添加做准备

- 公共信息 Cluster-Info

## 1. 获取cluster-info

- 获取Cluster-Info, 用token检验
- 实现了虚拟证书中心证书的分发



# kubeadm init/join 解析

## 8. 为worker节点的安全添加做准备

- 配置csrApproverController

## 2. 发出签名请求

- 获取kubelet的kubeconfig文件





# kubeadm 常见问题

# 遇到的问题

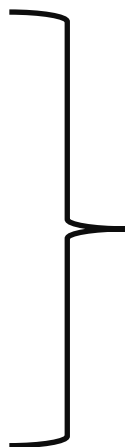
## 1. deb/rpm包无法翻墙安装问题

- kubernetes / kubelet / kubectl 都是通过deb/rpm
- 安装源为google官方源
- 中科大镜像源: <http://mirrors.ustc.edu.cn>
  - 跟kubernetes官方源保持同步


# 遇到的问题

## 2. docker镜像源gcr.io/google\_containers无法下载

kube-apiserver  
kube-controller-manager  
kube-scheduler  
etcd  
kube-proxy  
kube-dns related images



KUBE\_REPO\_PREFIX =  
registry.cn-hangzhou.aliyuncs.com/google-containers

**pause**  **kubelet --pod-infra-container-image**  
gcr.io/google\_containers/pause-amd64:3.0

# 遇到的问题

## 3. apiserver --advertise-address 参数判定问题

- 用途：apiserver组件和其他组件的通讯地址，监听地址
- 默认判定方法：default interface 网卡的ip地址
- 不适用场景
  - default interface 为公网网卡
  - 管理流量走内网网段

# 遇到的问题

## 4. kubelet --node-ip 参数判定问题

- 用途： kubelet组件和其他组件的通讯地址， 监听地址
- 默认判定方法： default interface 网卡的ip地址
- 不适用场景
  - default interface 为公网网卡
  - 管理流量走内网网段

# 遇到的问题

## 5. dns service ip 和 kubelet --cluster-dns 参数不匹配问题

- kubeadm 默认创建两个service, 根据service ip range来确定
  - 默认service ip range 10.96.0.0/16
  - kubernetes -> kube-apiserver 10.96.0.1
  - kube-dns -> kube-dns 10.96.0.10
- --cluster-dns: 指定kubelet启动的pod的nameserver
- 所以service ip range更新, 也要同步更新kubelet

# kubeadm 使用技巧

# 使用技巧

## 1. Kubeadm init --config 文件

- 提供基于配置文件的完整配置
  - kubernetes version
  - service ip range
  - master组件的命令行参数
  - token

```
apiVersion: kubeadm.k8s.io/v1alpha1
kind: MasterConfiguration
api:
  advertiseAddress: <address|string>
  bindPort: <int>
etcd:
  endpoints:
  - <endpoint1|string>
  - <endpoint2|string>
  caFile: <path|string>
  certFile: <path|string>
  keyFile: <path|string>
  dataDir: <path|string>
  extraArgs:
    <argument>: <value|string>
    <argument>: <value|string>
  image: <string>
networking:
  dnsDomain: <string>
  serviceSubnet: <cidr>
  podSubnet: <cidr>
kubernetesVersion: <string>
cloudProvider: <string>
nodeName: <string>
authorizationModes:
- <authorizationMode1|string>
- <authorizationMode2|string>
token: <string>
tokenTTL: <time duration>
selfHosted: <bool>
apiServerExtraArgs:
  <argument>: <value|string>
  <argument>: <value|string>
controllerManagerExtraArgs:
  <argument>: <value|string>
  <argument>: <value|string>
```



# 使用技巧

## 2. 为master组件启动添加额外的配置参数

- 开启新的特性
- 修改运行模式

```
apiServerExtraArgs:  
  <argument>: <value|string>  
  <argument>: <value|string>  
controllerManagerExtraArgs:  
  <argument>: <value|string>  
  <argument>: <value|string>  
schedulerExtraArgs:  
  <argument>: <value|string>  
  <argument>: <value|string>
```

# 使用技巧

## 3. 为用户在集群外通过kubectl访问集群

- apiServerCertSANs:
  - X509 subject alternative name
- /etc/kubernetes/admin.conf
- 适用场景:
  - 内网集群
  - 仅暴露一个公网ip
  - 通过公网ip和kubectl访问集群

```
apiServerCertSANs:  
- <name1|string>  
- <name2|string>
```

# 使用技巧

## 4. 如何更新static pod组件

- 把yaml文件从/etc/kubernetes/manifests move出去
- 修改这个文件
- 把yaml文件move回/etc/kubernetes/manifests下面



七牛容器云

Thank you