

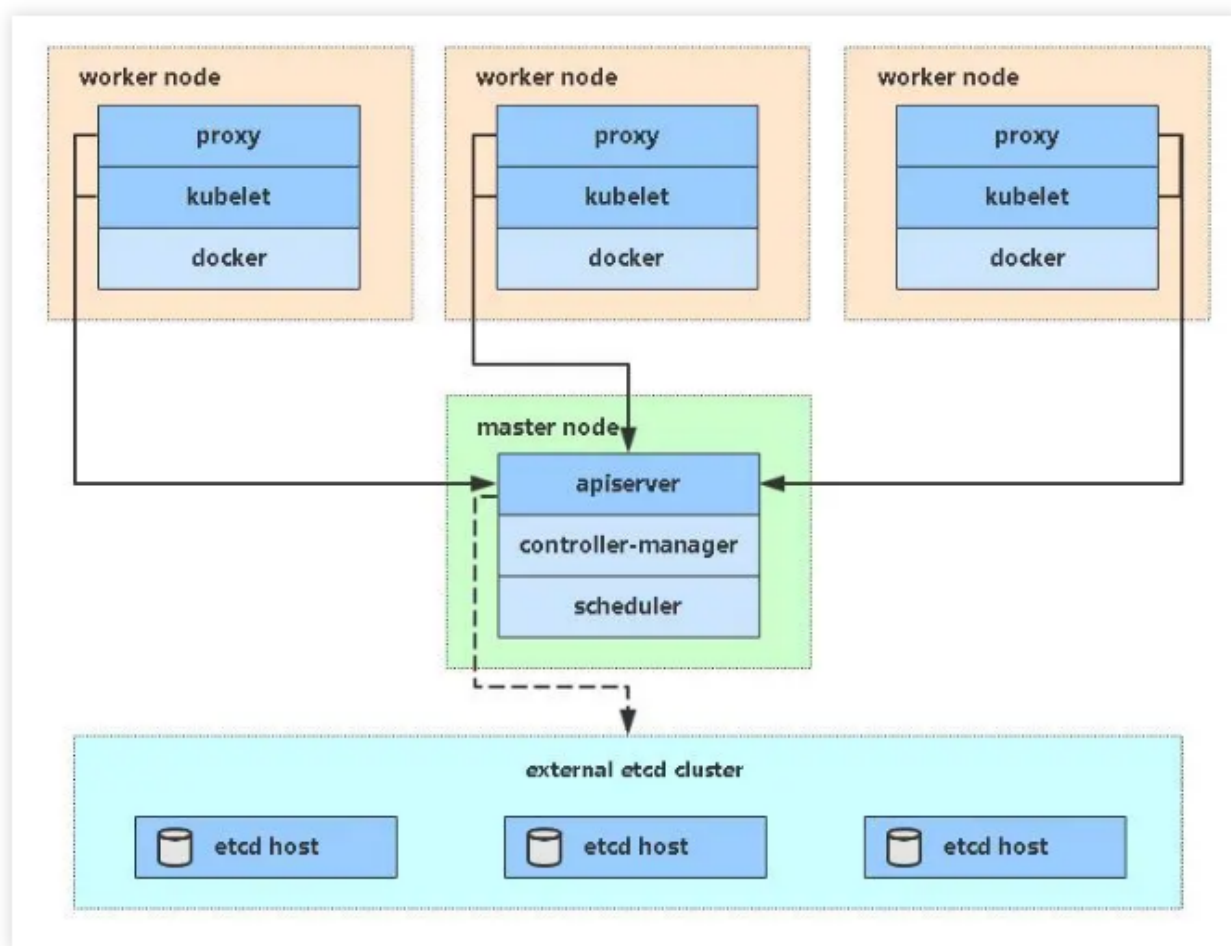
# Kubernetes集群搭建

## 1. 安装要求

在开始之前，部署Kubernetes集群机器需要满足以下几个条件：

- 一台或多台机器，操作系统 CentOS7.x-86\_x64
- 硬件配置：2GB或更多RAM，2个CPU或更多CPU，硬盘30GB或更多
- 集群中所有机器之间网络互通
- 可以访问外网，需要拉取镜像
- 禁止swap分区

## 2. 环境准备



角色：

k8s-master	192.168.78.135
k8s-node1	192.168.78.136
k8s-node1	192.168.78.137

```
2 systemctl stop firewalld && systemctl disable firewalld
3 // 2.关闭SELinux
4 vi /etc/selinux/config,将SELINUX=enforcing改为SELINUX=disabled
5 // 3.关闭swap
6 sed -ri 's/.*swap.*/#&/' /etc/fstab
7 // 4.设置主机名
8 hostnamectl set-hostname k8s-master
9 reboot //只有马上重启才会生效
10 // 5.在 master 节点添加 hosts
11 cat >>/etc/hosts << EOF
12 192.168.78.135 k8s-master
13 192.168.78.136 k8s-node1
14 192.168.78.137 k8s-node1
15 EOF
16 // 6.将桥接的IPv4流量传递到iptables的链
17 cat >/etc/sysctl.d/k8s.conf << EOF
18 net.bridge.bridge-nf-call-ip6tables =1
19 net.bridge.bridge-nf-call-iptables =1
20 EOF
21 sysctl --system // 重启生效
22 // 7.时间同步
23 yum install ntpdate -y
24 ntpdate time.windows.com
```

## 3. 所有节点安装Docker/kubeadm/kubelet

Kubernetes默认CRI（容器运行时）为Docker，因此先安装Docker。

### 3.1 安装Docker

```
1 // 1.
2 wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O
/etc/yum.repos.d/docker-ce.repo
3 // 2.
4 yum -y install docker-ce
5 // 3.
6 systemctl enable docker && systemctl start docker
7 // 4.配置镜像加速器
8 cat >/etc/docker/daemon.json << EOF
9 {
10   "registry-mirrors":["https://v0i08fi4.mirror.aliyuncs.com"]
11 }
```

```
12 EOF
13 // 5.重启生效
14 sudo systemctl daemon-reload
15 sudo systemctl restart docker
```

## 3.2 添加阿里云YUM软件源

```
1 cat >/etc/yum.repos.d/kubernetes.repo << EOF
2 [kubernetes]
3 name=Kubernetes
4 baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
5 enabled=1
6 gpgcheck=0
7 repo_gpgcheck=0
8 gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
  https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
9 EOF
```

## 3.3 安装kubeadm, kubelet和kubectl

由于版本更新频繁，这里指定版本号部署：

```
1 yum install -y kubelet-1.18.0 kubeadm-1.18.0 kubectl-1.18.0
2 systemctl enable kubelet
```

- kubelet: systemd守护进程管理
- kubeadm: 部署工具
- kubectl: k8s命令行管理工具

## 3.4 部署Kubernetes Master

```
1 kubeadm init \
2 --apiserver-advertise-address=192.168.78.135 \
3 --image-repository registry.aliyuncs.com/google_containers \
4 --kubernetes-version v1.18.0 \
5 --service-cidr=10.96.0.0/12 \
6 --pod-network-cidr=10.244.0.0/16
```

- —apiserver-advertise-address 集群通告地址
- —image-repository 由于默认拉取镜像地址k8s.gcr.io国内无法访问，这里指定阿里云镜像仓库地址。
- —kubernetes-version K8s版本，与上面安装的一致

- —service-cidr 集群内部虚拟网络，Pod统一访问入口
- —pod-network-cidr Pod网络，与下面部署的CNI网络组件yaml中保持一致

出现如下图信息表示初始化成功：

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
```

拷贝kubectl使用的连接k8s认证文件到默认路径：

```
1 mkdir -p $HOME/.kube
2 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
3 sudo chown $(id -u):$(id -g) $HOME/.kube/config
4 // 查看节点信息
5 kubectl get nodes
```

## 4. 加入节点信息

在192.168.78.136、192.168.78.137执行。

向集群添加新节点，执行在kubeadm init输出的kubeadm join命令。

```
1 kubeadm join 192.168.31.61:6443--token esce21.q6hetwm8si29qxwn \
2 --discovery-token-ca-cert-hash sha256:00603a05805807501d7181c3d60b4787884
08cfe6cedefedb1f97569708be9c5
```

默认token有效期为24小时，当过期之后，该token就不可用了。这时就需要重新创建token，操作如下：

```
1 kubeadm token create --print-join-command
2 // 加上 --ttl 0 会生成一个用不过期的token
3 kubeadm token create --print-join-command --ttl 0
```

## 5. 部署容器网络（CNI）

这里使用Flannel作为Kubernetes容器网络方案，解决容器跨主机网络通信。

Flannel是CoreOS维护的一个网络组件，Flannel为每个Pod提供全局唯一的IP，Flannel使用ETCD来存储Pod子网与Node IP之间的关系。flanneld守护进程在每台主机上运行，并负责维护ETCD信息和路由数据包。

```
1 wget https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
2 // 修改国内镜像地址
3 sed -i -r "s#quay.io/coreos/flannel:.*-amd64#lizhenliang/flannel:v0.11.0-amd64#g" kube-flannel.yml
4 kubectl apply -f kube-flannel.yml
5 kubectl get pods -n kube-system
```

## 6. 部署官方Dashboard (UI)

```
1 wget https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.3/aio/deploy/recommended.yaml
```

默认Dashboard只能集群内部访问，修改Service为NodePort类型，暴露到外部：

vim recommended.yaml

```
[root@master01 ~]# vim recommended.yaml
kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
spec:
  type: NodePort
  ports:
    - port: 443
      targetPort: 8443
      nodePort: 30000
  selector:
    k8s-app: kubernetes-dashboard
```

```
1 kubectl apply -f recommended.yaml
2 kubectl get pods -n kubernetes-dashboard
```

访问地址：<https://192.168.78.135:30000>

## 创建service account并绑定默认cluster-admin管理员集群角色：

```
1 // 创建用户
2 kubectl create serviceaccount dashboard-admin -n kube-system
3 // 用户授权
4 kubectl create clusterrolebinding dashboard-admin --clusterrole=cluster-admin --serviceaccount=kube-system:dashboard-admin
5 // 获取用户Token
6 kubectl describe secrets -n kube-system $(kubectl -n kube-system get secret | awk '/dashboard-admin/{print $1}')
```

## 使用输出的Token登录Dashboard：

### Kubernetes 仪表盘

☒ Token

每个 Service Account 都有一个 valid Bearer Token，可用于登录 Dashboard。要了解有关如何配置和使用 Bearer Tokens 的更多信息，请参考 [Authentication](#) 部分。

☐ Kubeconfig

请选择您创建的 kubeconfig 文件以配置对集群的访问权限。要了解有关如何配置和使用 kubeconfig 文件的更多信息，请参阅 [Configure Access to Multiple Clusters](#) 部分。

输入 token \*

登录

kubernetes

default

搜索

Overview

集群

Cluster Roles

Namespaces

Nodes

Persistent Volumes

Service Accounts

Storage Classes

工作量

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Services

配置和存储

Config Maps

工作量

工作量状态

Deployments

Pods

Replica Sets

Deployments

名字	命名空间	标签	Pods	创建时间	操作
web	default	app: web	3 / 3	43 minutes ago	详情

1 - 1 of 1

Pods

名字	命名空间	标签	节点	状态	重启	CPU 使用率 (cores)	内存使用 (bytes)	创建时间
----	------	----	----	----	----	-----------------	--------------	------

## 7.参考

### 7.1 环境搭建

<https://mp.weixin.qq.com/s/PD1CCkKZgtZD7pAZEgF-rw>

<https://www.kubernetes.org.cn/7189.html>

[https://zhuanlan.zhihu.com/p/150670253?from\\_voters\\_page=true](https://zhuanlan.zhihu.com/p/150670253?from_voters_page=true)

### 7.2 错误处理

"WARNING IsDockerSystemdCheck" 错误。

```
1 vi /usr/lib/systemd/system/docker.service
2 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --exec-opt native.cgroupdriver=systemd
3 systemctl daemon-reload
4 systemctl restart docker
```

参考: <https://blog.whsir.com/post-5312.html>

"error execution phase preflight: [preflight] Some fatal errors occurred:"

```
1 kubeadm reset
```

参考: [https://blog.csdn.net/qq\\_44895681/article/details/107414231](https://blog.csdn.net/qq_44895681/article/details/107414231)