# kubeadm安装kubernetes

## 前提准备

```
# 三台的hosts文件改下
172.17.0.2      node-1
172.17.0.3      node-2
172.17.0.4      master
# 配置免密
ssh-keygen -t rsa   # 全部回车就行了
ssh-cory-id -i ~/.ssh/id_rsa.pub root@node-1
```

## 1.containerd安装

```
wget
https://github.com/containerd/containerd/releases/download/v1.5.5/cri-
containerd-cni-1.5.5-linux-amd64.tar.gz
tar -zxvf cri-containerd-cni-1.5.5-linux-amd64.tar.gz -C /
mkdir -p /etc/containerd
systemctl enable containerd --now
rm /etc/containerd/config.toml
systemctl restart containerd
# 打印版本信息，能正常输出就没问题
crictl version
```

## 2.安装Kubernetes

## 1、基本环境

```
#设置每个机器自己的hostname(三台机器分别是master、node-1、node-2)
hostnamectl set-hostname xxx

# 将 SELinux 设置为 permissive 模式（相当于将其禁用）
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'
/etc/selinux/config

#关闭swap
swapoff -a
sed -ri 's/.*swap.*/#&/' /etc/fstab

#允许 iptables 检查桥接流量
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
br_netfilter
EOF

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
```

## 2、安装kubelet、kubeadm、kubectl

```
#配置k8s的yum源地址
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=http://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-
x86_64
enabled=1
gpgcheck=0
```

```
repo_gpgcheck=0
gpgkey=http://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
   http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
EOF

yum makecache fast


#安装 kubelet, kubeadm, kubectl
sudo yum install -y kubelet-1.22.4 kubeadm-1.22.4 kubectl-1.22.4

#启动kubelet
sudo systemctl enable  kubelet

#所有机器配置master域名
echo 172.17.0.2      node-1 \
172.17.0.3      node-2 \
172.17.0.4      master >> /etc/hosts\
```

## 3.kubeadmm初始化集群

```
touch kubeadm.yaml
```

这是我的yaml，可以参考下

```
apiVersion: kubeadm.k8s.io/v1beta2
kind: InitConfiguration
localAPIEndpoint:
  advertiseAddress: 172.17.0.4
  bindPort: 6443
nodeRegistration:
  criSocket: /run/containerd/containerd.sock
  taints:
  - effect: PreferNoSchedule
    key: node-role.kubernetes.io/master
```

```
---
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
kubernetesVersion: v1.22.0
imageRepository: registry.aliyuncs.com/google_containers
networking:
  podSubnet: 10.244.0.0/16
---
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
cgroupDriver: systemd
failSwapOn: false
---
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
mode: ipvs
```

尝试拉下镜像

```
kubeadm config images pull --config kubeadm.yaml
```

初始化master节点

```
kubeadm init --config kubeadm.yaml
```

把最后输出的记录下来

```


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

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed
at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on
each as root:

kubeadm join 172.17.0.4:6443 --token a0qym3.m9ra9pyc0gy89cn5 \
  --discovery-token-ca-cert-hash
sha256:988f5fdd912dbb4b75e733f50b4a9b44ef126371c012747d0c5ea423eb0a1fb7
```

照着提示

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

然后执行以下命令查看master是否正常启动

```
kubectl get node
```

新节点加入

```
kubeadm join 172.17.0.4:6443 --token a0qym3.m9ra9pyc0gy89cn5 \
  --discovery-token-ca-cert-hash
sha256:988f5fdd912dbb4b75e733f50b4a9b44ef126371c012747d0c5ea423eb0a1fb7
```