一.centos配置

1.centos下载地址:推荐大家使用centos7.6以上版本。

```
http://mirrors.aliyun.com/centos/7/isos/x86_64/
```

2.查看centos系统版本命令

```
cat /etc/centos-release
```

3.配置阿里云yum源

```
1.下载安装wget
yum install -y wget

2.备份默认的yum
mv /etc/yum.repos.d /etc/yum.repos.d.backup

3.设置新的yum目录
mkdir -p /etc/yum.repos.d

4.下载阿里yum配置到该目录中，选择对应版本
wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

5.更新epel源为阿里云epel源
wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

6.重建缓存
yum clean all
yum makecache

7.看一下yum仓库有多少包
yum repolist
yum update
```

4.升级系统内核

```
rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm
yum --enablerepo=elrepo-kernel install -y kernel-lt
grep initrd16 /boot/grub2/grub.cfg
grub2-set-default 0

reboot
```

```
#查看centos系统内核命令：
uname -r
uname -a
```

```
#查看cpu命令
lscpu
```

```
#查看内存命令
free
free -h
```

```
#查看硬盘信息
fdisk -l
```

5.关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

6.关闭selinux

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
setenforce 0
```

7.网桥过滤

```
vim /etc/sysctl.conf

net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-arptables = 1
net.ipv4.ip_forward=1
net.ipv4.ip_forward_use_pmtu = 0

#生效命令
sysctl --system

#查看效果
sysctl -a|grep "ip_forward"
```

8.开启IPVS

```
#安装IPVS
yum -y install ipset ipvsdm

#编译ipvs.modules文件
vim /etc/sysconfig/modules/ipvs.modules

#文件内容如下
#!/bin/bash
modprobe -- ip_vs
modprobe -- ip_vs_rr
modprobe -- ip_vs_wrr
modprobe -- ip_vs_sh
modprobe -- nf_conntrack_ipv4

#赋予权限并执行
chmod 755 /etc/sysconfig/modules/ipvs.modules && bash
/etc/sysconfig/modules/ipvs.modules &&lsmod | grep -e ip_vs -e nf_conntrack_ipv4

#重启电脑，检查是否生效
```

```
reboot
lsmod | grep ip_vs_rr
```

9.同步时间

```
#安装软件
yum -y install ntpdate

#向阿里云服务器同步时间
ntpdate time1.aliyun.com

#删除本地时间并设置时区为上海
rm -rf /etc/localtime
ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

#查看时间
date -R || date
```

10.命令补全

```
#安装bash-completion
yum -y install bash-completion bash-completion-extras

#使用bash-completion
source /etc/profile.d/bash_completion.sh
```

11.关闭swap分区

```
#临时关闭
swapoff -a

#永久关闭
vim /etc/fstab

#将文件中的/dev/mapper/centos-swap这行代码注释掉
#/dev/mapper/centos-swap swap swap defaults 0 0

#确认swap已经关闭：若swap行都显示 0 则表示关闭成功
free -m
```

12.hosts配置

```
vim /etc/hosts

#文件内容如下:
192.168.238.180 master
192.168.238.181 node01
192.168.238.182 node02
192.168.238.183 node03
```

二.docker安装

```
# step 1: 安装必要的一些系统工具
sudo yum install -y yum-utils device-mapper-persistent-data lvm2

# Step 2: 添加软件源信息
sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo

# Step 3: 更新并安装 Docker-CE
sudo yum makecache fast
sudo yum -y install docker-ce

# Step 4: 开启Docker服务
sudo systemctl start docker
```

```
#查看docker更新版本
#yum list docker-ce --showduplicates | sort -r
#安装指定版本:
#yum -y install docker-ce-18.09.8
```

```
#配置阿里云镜像加速器
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://8jr2yxwm.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
#设置docker开启启动服务
systemctl enable docker
```

```
#修改Cgroup Driver
#修改daemon.json，新增
vim /etc/docker/daemon.json
"exec-opts": ["native.cgroupdriver=systemd"]

#重启docker服务
systemctl daemon-reload
systemctl restart docker

#查看修改后状态
docker info | grep Cgroup
```

三.kubeadm快速安装

软件	kubeadm	kubelet	kubectl	docker-ce
版本	初始化集群管理集群 版本:1.17.6	用于接收api-server指令，对pod生命周期进行管理版本:1.17.6	集群命令行管理工具 版本:1.17.6	推荐使用版本:19.03.8

1.安装yum源

```
#新建repo文件
vim /etc/yum.repos.d/kubernates.repo

#文件内容
[kubernetes]
name=Kubernetes
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
       https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
```

```
#更新缓存
yum clean all
yum -y makecache
```

```
#验证源是否可用
yum list | grep kubeadm

#如果提示要验证yum-key.gpg是否可用，输入y。
#查找到kubeadm。显示版本
```

```
#查看k8s版本
yum list kubelet --showduplicates | sort -r

#安装k8s-1.17.6
yum install -y kubelet-1.17.6 kubeadm-1.17.6 kubectl-1.17.6
```

2.设置kubelet

```
#如果不配置kubelet，可能会导致K8S集群无法启动。
#为实现docker使用的cgroupdriver与kubelet 使用的cgroup的一致性。
vim /etc/sysconfig/kubelet

KUBELET_EXTRA_ARGS="--cgroup-driver=systemd"
```

```
#设置开机启动
systemctl enable kubelet
```

3.初始化镜像(只需执行一次，后续可通过load tar方式导入镜像)

```
#查看安装集群需要的镜像
kubeadm config images list
```

```
#编写执行脚本
mkdir -p /data
cd /data
vim images.sh
#!/bin/bash
```

```
# 下面的镜像应该去除"k8s.gcr.io"的前缀，版本换成kubeadm config images list命令获取到的版本
images=(
    kube-apiserver:v1.17.6
    kube-controller-manager:v1.17.6
    kube-scheduler:v1.17.6
    kube-proxy:v1.17.6
    pause:3.1
    etcd:3.4.3-0
    coredns:1.6.5
)
for imageName in ${images[@]} ;
do
    docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/$imageName
    docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/$imageName k8s.gcr.io/$imageName
    docker rmi registry.cn-hangzhou.aliyuncs.com/google_containers/$imageName
done

docker save -o k8s.1.17.5.tar $images
```

```
#给脚本授权
chmod 777 images.sh
#执行脚本
./images.sh
```

```
#保存镜像
docker save -o k8s.1.17.6.tar \
k8s.gcr.io/kube-proxy:v1.17.6  \
k8s.gcr.io/kube-apiserver:v1.17.6  \
k8s.gcr.io/kube-controller-manager:v1.17.6  \
k8s.gcr.io/kube-scheduler:v1.17.6  \
k8s.gcr.io/coredns:1.6.5  \
k8s.gcr.io/etcd:3.4.3-0  \
k8s.gcr.io/pause:3.1
```

4.导入镜像

```
#导入master节点镜像tar包
#master节点需要全部镜像
docker load -i k8s.1.17.6.tar
```

```
#导入node节点镜像tar包
docker load -i k8s.1.17.6.tar
```

5.初始化集群

```
#calico官网地址
#官网下载地址
https://docs.projectcalico.org/v3.14/manifests/calico.yaml
#github地址
https://github.com/projectcalico/calico
#镜像下载
docker pull calico/cni:v3.14.2
docker pull calico/pod2daemon-flexvol:v3.14.2
docker pull calico/node:v3.14.2
docker pull calico/kube-controllers:v3.14.2
```

```
#镜像备份：
docker save -o calico3.14.tar \
calico/node:v3.14.2 \
calico/pod2daemon-flexvol:v3.14.2 \
calico/cni:v3.14.2 \
calico/kube-controllers:v3.14.2
```

```
#修改主机名称
hostnamectl set-hostname master
```

```
#集群所有节点都需要导入备份
docker load -i calico3.14.tar
```

```
#初始化集群信息:calico网络
kubeadm init --apiserver-advertise-address=192.168.238.180 --kubernetes-version
v1.17.6 --service-cidr=10.1.0.0/16 --pod-network-cidr=10.81.0.0/16
```

```
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.
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 192.168.238.180:6443 --token qn0o02.mfdxq3uvojsez1rg \
    --discovery-token-ca-cert-hash sha256:5a86ef3babfa53d8b04ef162bb2b13951149f2df5ea640749b379d6859abfeb1
[root@master data]#
```

```
#执行配置命令
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
#node节点加入集群信息
kubeadm join 192.168.238.180:6443 --token qn0o02.mfdxq3uvojsez1rg \
    --discovery-token-ca-cert-hash
sha256:5a86ef3babfa53d8b04ef162bb2b13951149f2df5ea640749b379d6859abfeb1
```

```
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

[root@node01 data]#
```

```
[root@master data]# kubectl get nodes
NAME     STATUS     ROLES    AGE     VERSION
master   NotReady   master   4m52s   v1.17.6
node01   NotReady   <none>   63s     v1.17.6
node02   NotReady   <none>   57s     v1.17.6
node03   NotReady   <none>   46s     v1.17.6
```

```
#执行命令安装网络
kubectl apply -f calico.yaml

#查看集群状态
kubectl get nodes
```

```
[root@master data]# kubectl get nodes
NAME     STATUS   ROLES    AGE     VERSION
master   Ready    master   9m52s   v1.17.6
node01   Ready    <none>   6m3s    v1.17.6
node02   Ready    <none>   5m57s   v1.17.6
node03   Ready    <none>   5m46s   v1.17.6
[root@master data]#
```

```
#kubectl命令自动补全
echo "source <(kubectl completion bash)" >> ~/.bash_profile
source ~/.bash_profile
```