版本1.15.1集群初始化centos7

笔记本: kubernetes
创建时间: 2019/10/7 12:22 **更新时间:** 2019/11/13 20:24
作者: 长诗佐酒

```
#免密登录
[root@k8s-master01 ~]# ssh-keygen -t rsa
[root@k8s-master01 ~]# ssh-copy-id node01.k8s
```

```
#设置系统主机名及Host文件的相互解析
hostnamectl set-hostname apiserver.demo
```

```
cat >>/etc/hosts<<EOF
192.168.10.200 master.k8s
192.168.10.201 node01.k8s
192.168.10.202 node02.k8s
192.168.10.203 node03.k8s
192.168.10.204 node04.k8s
EOF
```

```
#安装依赖包
yum install -y \
yum-utils \
device-mapper-persistent-data \
lvm2 \
conntrack \
ntpdate \
ntp \
ipvsadm \
ipset \
jq \
iptables \
curl \
sysstat \
libseccomp \
wget \
vim \
net-tools \
git

#设置防火墙为iptables并设置空规则
systemctl stop firewalld && systemctl disable firewalld
yum install -y iptables-services && systemctl start iptables &&
systemctl enable iptables && iptables -F && service iptables save

#关闭selinux
setenforce 0
sed -i "s/SELINUX=enforcing/SELINUX=disabled/g"
/etc/selinux/config
#关闭swap分区

swapoff -a

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

#调整内核参数，对于k8s
cat > /etc/sysctl.d/kuberbetes.conf << EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind=1
net.ipv4.tcp_tw_recycle = 0
vm.swappiness=0 #禁止使用swap空间，只有当系统ooM时才允许使用它
vm.overcommit_memory=1 #不检查物理内存是否够用
vm.panic_on_oom=0 #开启oom
fs.inotify.max_user_instances=8192
fs.inotify.max_user_watches=1048576
fs.file-max=52706963
fs.nr_open=52706963
net.ipv6.conf.all.disable_ipv6=1
net.netfilter.nf_conntrack_max=2310720
EOF

sysctl -p /etc/sysctl.d/kuberbetes.conf

#设置系统时区为中国上海
timedatectl set-timezone Asia/Shanghai
#将当前UTC时间写入硬件时钟
timedatectl set-local-rtc 0
```

```
#重启依赖于系统时间的服务
systemctl restart rsyslog
systemctl restart crond

#关闭系统不需要的服务
systemctl stop postfix && systemctl disable postfix

#设置rsyslogd和systemd journald
mkdir -p /var/log/journal #持久化保存日志的目录
mkdir -p /etc/systemd/journald.conf.d
cat > /etc/systemd/journald.conf.d/99-prophet.conf <<EOF
[Journal]
#持久化保存到磁盘
Storage=persistent
#压缩历史日志
Compress=yes
SyncIntervalSec=5m
RateLimitInterval=30s
RateLimitBurst=1000
#最大占用空间10g
SystenMaxUse=10G
#单日志文件最大200m
SystemMaxFileSize=200M
# 日志保存时间为2周
MaxRetentionSec=2week
#不将日志转发到syslog
ForwardToSyslog=no
EOF
systemctl restart systemd-journald

#升级系统内核为4.44
rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm
yum --enablerepo=elrepo-kernel install -y kernel-lt
#设置开机从新内核启动
grub2-set-default "CentOS Linux (4.4.182-1.el7.elrepo.x86_64) 7 (Core)" && reboot

#kube-proxy开启ipvs的前置条件
cat > /etc/sysconfig/modules/ipvs.modules <<EOF
#!/bin/bash
modprobe -- ip_vs
modprobe -- ip_vs_rr
modprobe -- ip_vs_wrr
modprobe -- ip_vs_sh
modprobe -- nf_conntrack_ipv4
EOF
chmod 755 /etc/sysconfig/modules/ipvs.modules && bash /etc/sysconfig/modules/ipvs.modules && lsmod | grep -e ip_vs -e nf_conntrack_ipv4

#安装docker
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
yum install -y docker-ce-18.09.7 docker-ce-cli-18.09.7 containerd.io
#yum update -y ; yum install -y docker-ce
systemctl enable docker &&  systemctl start docker
#mkdir /etc/docker
cat > /etc/docker/daemon.json <<EOF
{
"registry-mirrors": ["https://v16stybc.mirror.aliyuncs.com"],
"exec-opts": ["native.cgroupdriver=systemd"],
"insecure-registries": ["192.168.10.200:5000"],
"log-driver": "json-file",
        "log-opts": {
                "max-size": "10m",
                "max-file":"5"
        }
}
EOF
mkdir -p /etc/systemd/system/docker.service.d
#重启docker服务
systemctl daemon-reload && systemctl restart docker

#安装kubeadm
cat <<EOF > /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 install -y kubelet-1.15.1 kubeadm-1.15.1 kubectl-1.15.1
systemctl enable kubelet.service

#初始化主节点
kubeadm config print init-defaults > kubeadm-config.yaml

#修改kubeadm-config.yaml至如下
apiVersion: kubeadm.k8s.io/v1beta2
bootstrapTokens:
- groups:
  - system:bootstrappers:kubeadm:default-node-token
  token: abcdef.0123456789abcdef
  ttl: 24h0m0s
  usages:
  - signing
  - authentication
kind: InitConfiguration
localAPIEndpoint:
  advertiseAddress: 192.168.10.200  #当前主节点ip
  bindPort: 6443
nodeRegistration:
  criSocket: /var/run/dockershim.sock
  name: master.k8s
  taints:
  - effect: NoSchedule
    key: node-role.kubernetes.io/master
---
apiServer:
  timeoutForControlPlane: 4m0s
apiVersion: kubeadm.k8s.io/v1beta2
certificatesDir: /etc/kubernetes/pki
clusterName: kubernetes
#controlPlaneEndpoint: "192.168.10.199:8443" #keeplive虚拟的ip
controllerManager: {}
dns:
  type: CoreDNS
etcd:
  local:
    dataDir: /var/lib/etcd
```

```
imageRepository: registry.cn-hangzhou.aliyuncs.com/google_containers  #修改为国内镜像源
kind: ClusterConfiguration
kubernetesVersion: v1.15.1   #当前版本
networking:
  dnsDomain: cluster.local
  podSubnet: "10.244.0.0/16"  #添加，此为flannel网络默认网段,也可修改插件yaml文件中的网段，也可只写""
  serviceSubnet: 10.96.0.0/12
scheduler: {}
---
apiVersion: kubeproxy.config.k8s.io/v1alpha1   #末尾添加，修改为ipvs调度模式
kind: KubeProxyConfiguration
featureGates:
  SupportIPVSProxyMode: true
mode: ipvs
```

#初始化时自动颁发证书，后续版本用

--upload-certs

```
kubeadm init --config=kubeadm-config.yaml --experimental-upload-certs | tee kubeadm-init.log
rm -rf /root/.kube/ ;mkdir /root/.kube/ ;cp -i /etc/kubernetes/admin.conf /root/.kube/config
```

#kubectl默认会在执行的用户家目录下面的.kube目录下寻找config文件。这里是将在初始化时[kubeconfig]步骤生成的admin.conf拷贝到.kube/config

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

在k8s-master01将证书文件拷贝至k8s-master02、k8s-master03节点

#拷贝证书至k8s-master02节点

```
[root@k8s-master01 ~]# vim /tmp/k8s-master-zhengshu.sh
#!/bin/bash
USER=root
CONTROL_PLANE_IPS="node01.k8s  node02.k8s"
for host in ${CONTROL_PLANE_IPS}; do
ssh "${USER}"@$host "mkdir -p /etc/kubernetes/pki/etcd"
scp /etc/kubernetes/pki/ca.* "${USER}"@$host:/etc/kubernetes/pki/
scp /etc/kubernetes/pki/sa.* "${USER}"@$host:/etc/kubernetes/pki/
scp /etc/kubernetes/pki/front-proxy-ca.* "${USER}"@$host:/etc/kubernetes/pki/
scp /etc/kubernetes/pki/etcd/ca.* "${USER}"@$host:/etc/kubernetes/pki/etcd/
scp /etc/kubernetes/admin.conf "${USER}"@$host:/etc/kubernetes/
done
```

```
#部署flannel网络,只在 master 节点执行（flannel的国外镜像pull失败，image: registry.cn-shanghai.aliyuncs.com/gcr-k8s/flannel:v0.10.0-amd64)
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

```
# 或者部署 calico网络,只在 master 节点执行
kubectl apply -f https://docs.projectcalico.org/v3.6/getting-started/kubernetes/installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml
#修改calico.yaml，修改CALICO_IPV4POOL_CIDR这个下面的vaule值。与kubeadm初始化文件中的serviceSubnet的值对应
```

```
# 在 master 节点执行，获得worker节点加入集群的命令
kubeadm token create --print-join-command
#所有其他master加入集群时带参数 --experimental-control-plane
```

```
#master故障后恢复
#先观察集群状态
kubectl get endpoints kube-controller-manager --namespace=kube-system -o yaml
kubectl get endpoints kube-scheduler --namespace=kube-system -o yaml
kubectl -n kube-system exec etcd-k8s-master01节点 --etcdctl  \
--endpoints=https://192.168.10.200:2379  \
--ca-file=/etc/kubernetes/pki/etcd/ca.crt  \
--cert-file=/etc/kubernetes/pki/etcd/server.crt  \
--key-file=/etc/kubernetes/pki/etcd/server.key   cluster-health
#降低已恢复master的keeplived的优先级（待验证，可能是配置问题）
#或修改  .kube/config 中的连接ip地址，否则kubelet命令会失效
#kubectl delete 故障节点
#验证etcd中的故障master信息是否删除，否则修改
kubectl edit configmaps -n kube-system kubeadm-config
```

#从健康master拷贝ca等证书到各目录

#kubeadm token create --print-join-command且带--experimental-control-plane