# 搭建Kubernetes集群

此文以Kubernetes 1.18.5版本为例！

如未指定，下述命令在所有节点执行！

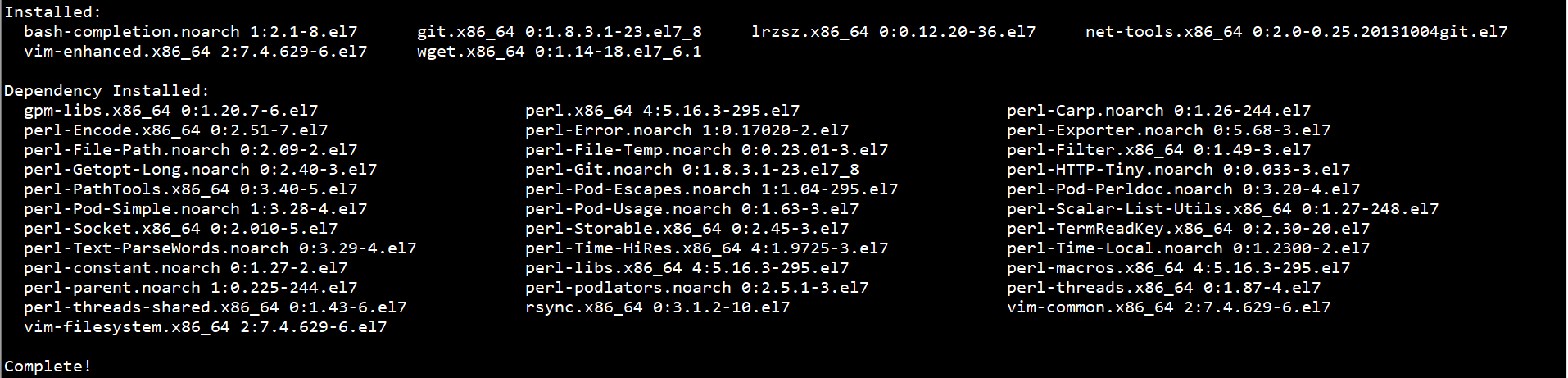
## 一、系统资源规划

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **节点名称** | **系统名称** | **CPU/内存** | **网卡** | **磁盘** | **IP地址** | **OS** |
| Master | master | 4C/8G | ens33 | 64G | 192.168.0.10 | CentOS7 |
| Worker1 | worker1 | 4C/8G | ens33 | 64G | 192.168.0.11 | CentOS7 |
| Worker2 | worker2 | 4C/8G | ens33 | 64G | 192.168.0.12 | CentOS7 |

## 二、系统软件安装与设置

### 1、安装基本软件

yum -y install vim git lrzsz wget net-tools bash-completion

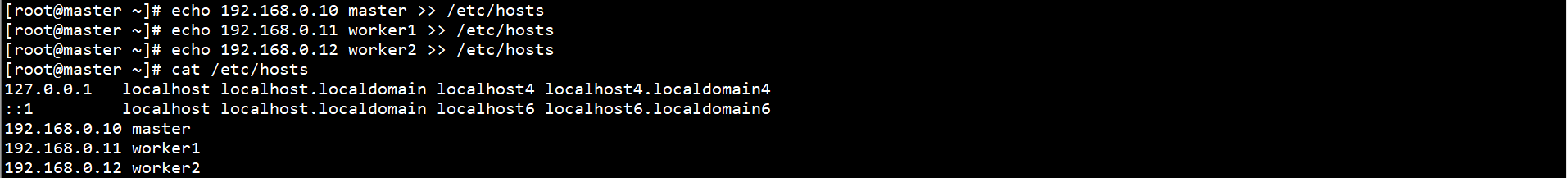


### 2、设置名称解析

echo 192.168.0.10 master >> /etc/hosts

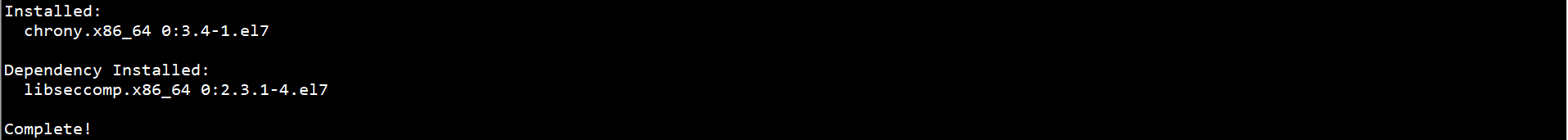
echo 192.168.0.11 worker1 >> /etc/hosts

echo 192.168.0.12 worker2 >> /etc/hosts



### 3、设置NTP

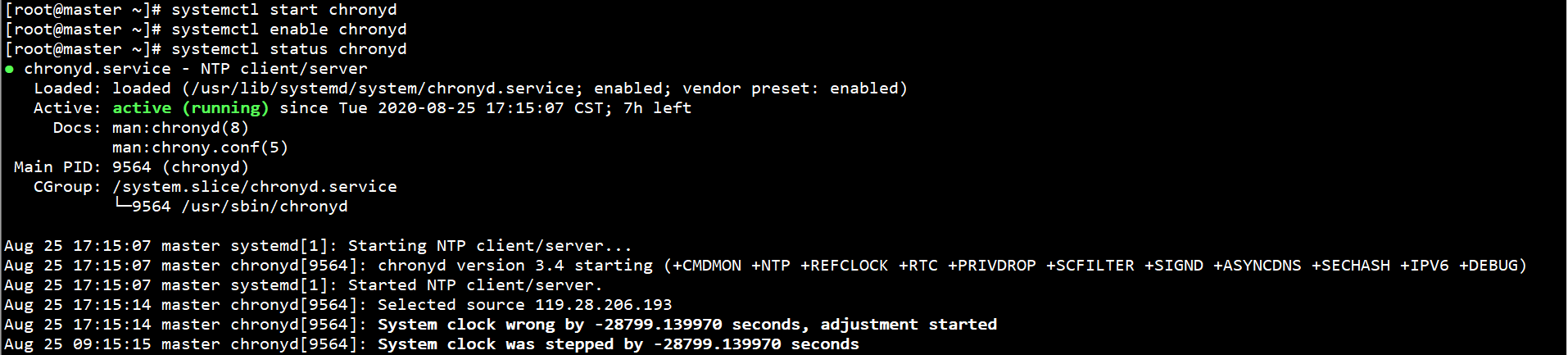
yum -y install chrony



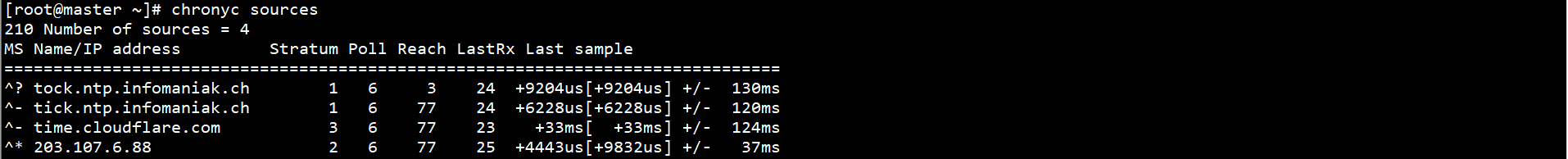
systemctl start chronyd

systemctl enable chronyd

systemctl status chronyd



chronyc sources



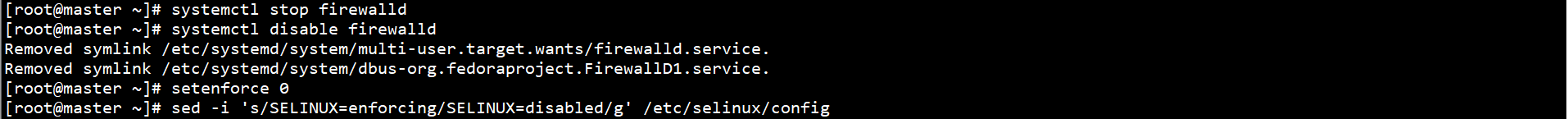
### 4、设置SELinux、防火墙

systemctl stop firewalld

systemctl disable firewalld

setenforce 0

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config



### 5、设置网桥

配置L2网桥在转发包时会被iptables的FORWARD规则所过滤，CNI插件需要该配置

创建/etc/sysctl.d/k8s.conf文件，添加如下内容：

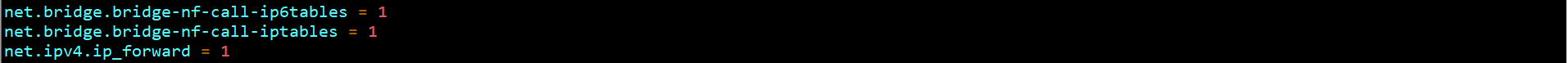
cat > /etc/sysctl.d/k8s.conf << EOF

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

net.ipv4.ip\_forward = 1

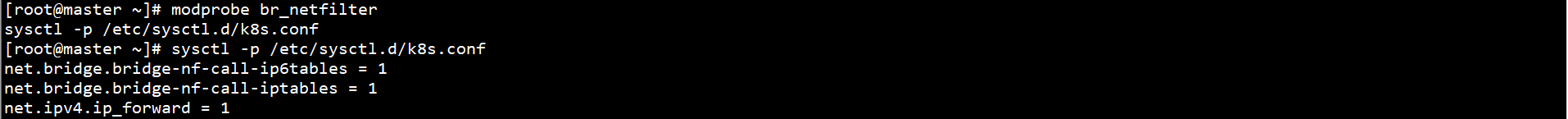
EOF



执行命令，使修改生效：

modprobe br\_netfilter

sysctl -p /etc/sysctl.d/k8s.conf



### 6、设置swap

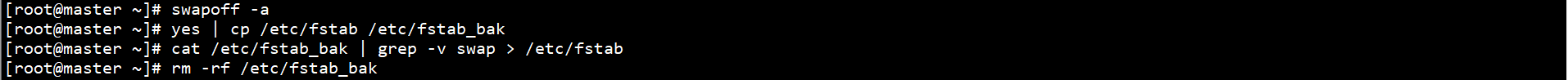
关闭系统swap分区：

swapoff -a

yes | cp /etc/fstab /etc/fstab\_bak

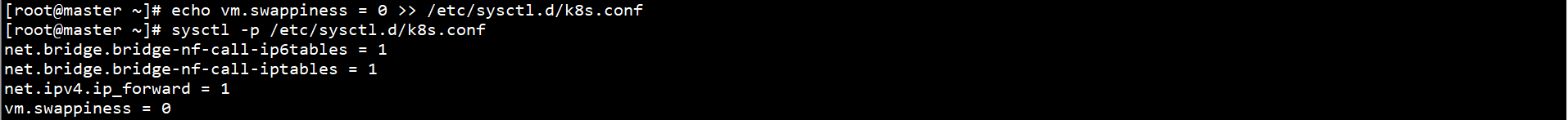
cat /etc/fstab\_bak | grep -v swap > /etc/fstab

rm -rf /etc/fstab\_bak



echo vm.swappiness = 0 >> /etc/sysctl.d/k8s.conf

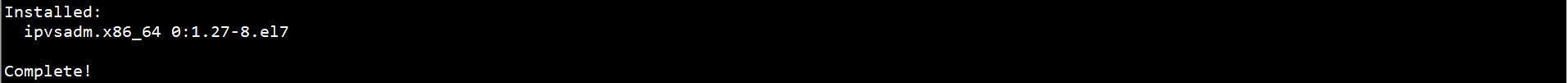
sysctl -p /etc/sysctl.d/k8s.conf



### 7、设置ipvs

安装ipvsadm ipset：

yum -y install ipvsadm ipset



创建ipvs设置脚本：

cat > /etc/sysconfig/modules/ipvs.modules << EOF

#!/bin/bash

modprobe -- ip\_vs

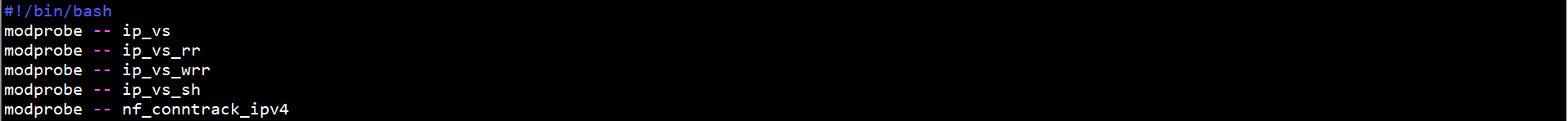
modprobe -- ip\_vs\_rr

modprobe -- ip\_vs\_wrr

modprobe -- ip\_vs\_sh

modprobe -- nf\_conntrack\_ipv4

EOF

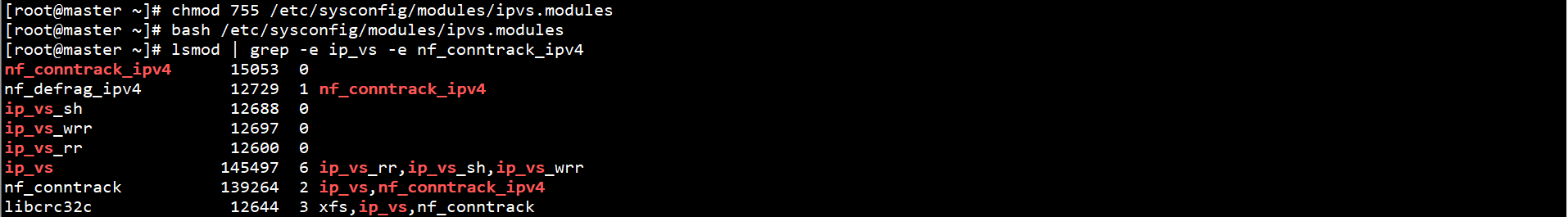


执行脚本，验证修改结果：

chmod 755 /etc/sysconfig/modules/ipvs.modules

bash /etc/sysconfig/modules/ipvs.modules

lsmod | grep -e ip\_vs -e nf\_conntrack\_ipv4

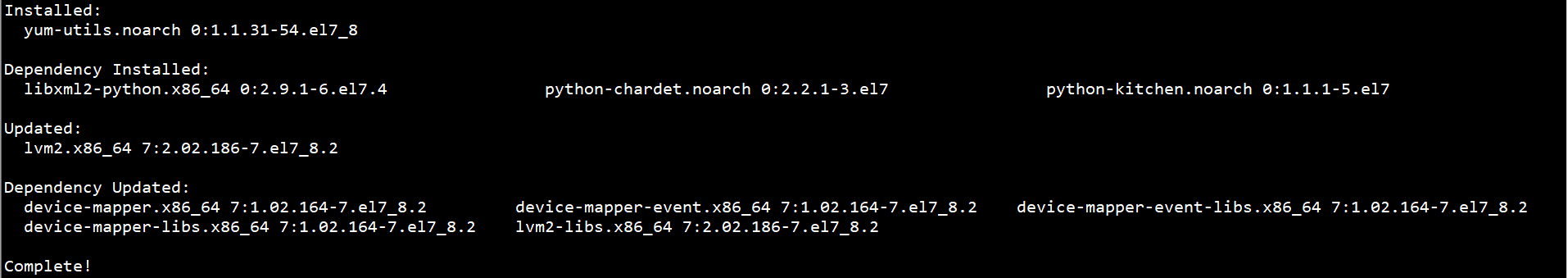


## 三、Kubernetes集群配置

### 1、安装Docker

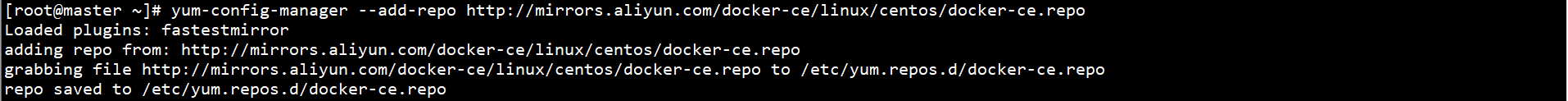
安装所需软件包：

yum -y install yum-utils device-mapper-persistent-data lvm2



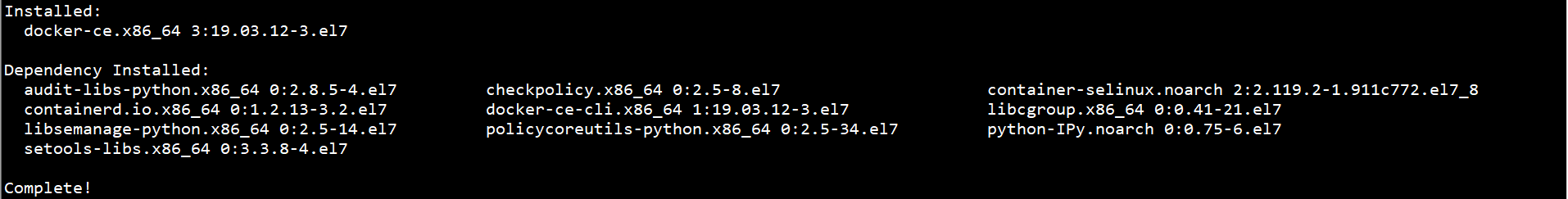
设置稳定存储库：

yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo



安装Docker CE：

yum -y install docker-ce

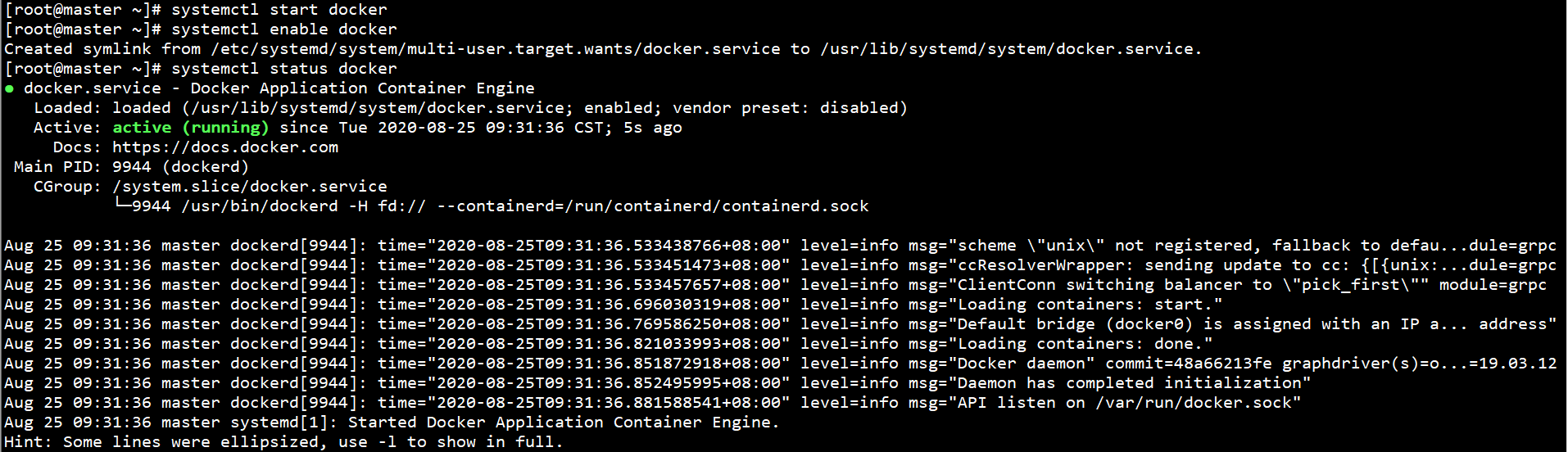


启动Docker，并设置自启动：

systemctl start docker

systemctl enable docker

systemctl status docker



### 2、安装kubelet、kubeadm和kubectl

添加kubernetes存储库：

cat > /etc/yum.repos.d/kubernetes.repo << EOF

[kubernetes]

name=Kubernetes

baseurl=http://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_64

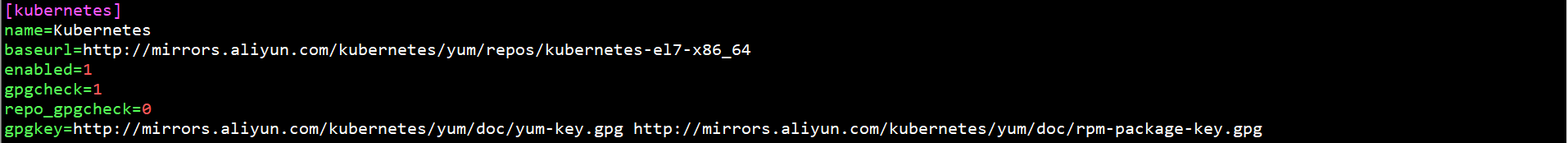
enabled=1

gpgcheck=1

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

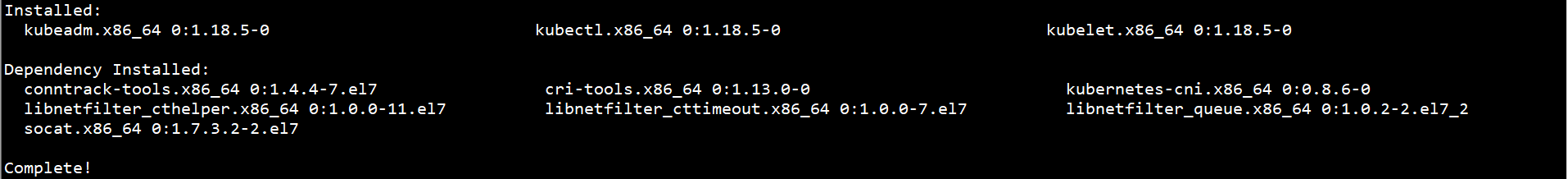
EOF



安装kubelet、kubeadm和kubectl：

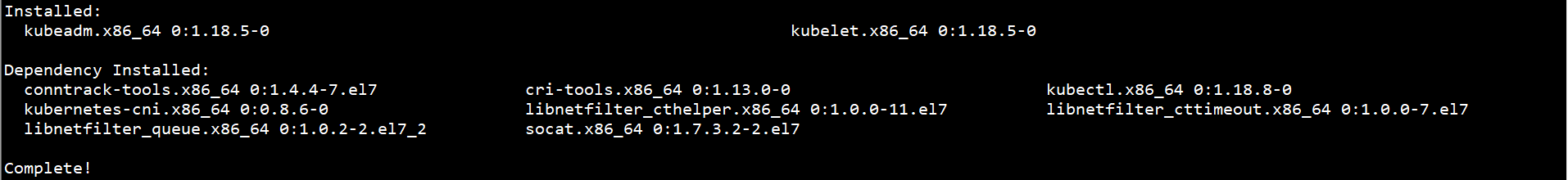
Master节点：

yum -y install kubelet-1.18.5-0 kubeadm-1.18.5-0 kubectl-1.18.5-0



Worker节点：

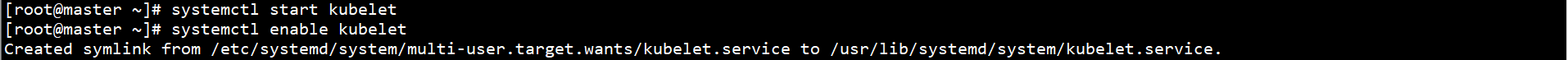
yum -y install kubelet-1.18.5-0 kubeadm-1.18.5-0



启动kubelet，并设置自启动：

systemctl start kubelet

systemctl enable kubelet



此时kubelet缺省配置文件无法启动，可忽略状态

### 3、设置cgroup driver

修改docker cgroup driver为systemd：

cat > /etc/docker/daemon.json << EOF

{

"exec-opts": ["native.cgroupdriver=systemd"],

"log-driver": "json-file",

"log-opts": {

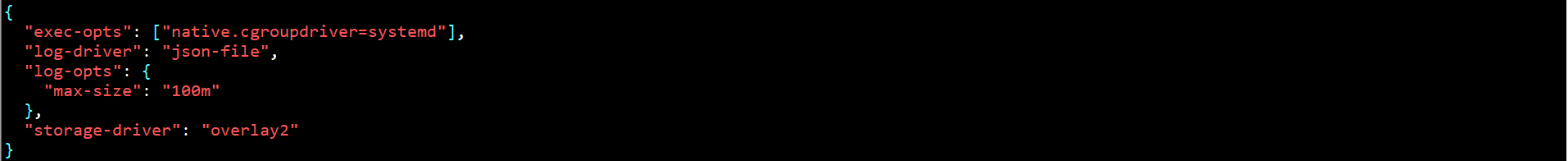
"max-size": "100m"

},

"storage-driver": "overlay2"

}

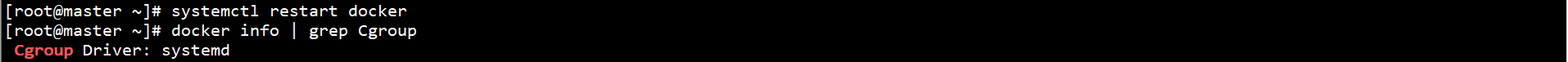
EOF



重启Docker，验证修改结果：

systemctl restart docker

docker info | grep Cgroup

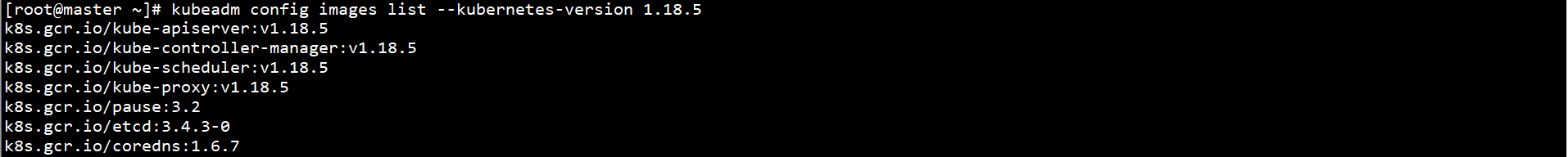


### 4、下载镜像

由于镜像在google在Registry上，国内无法访问，需要手动从阿里云或其他Registry上下载

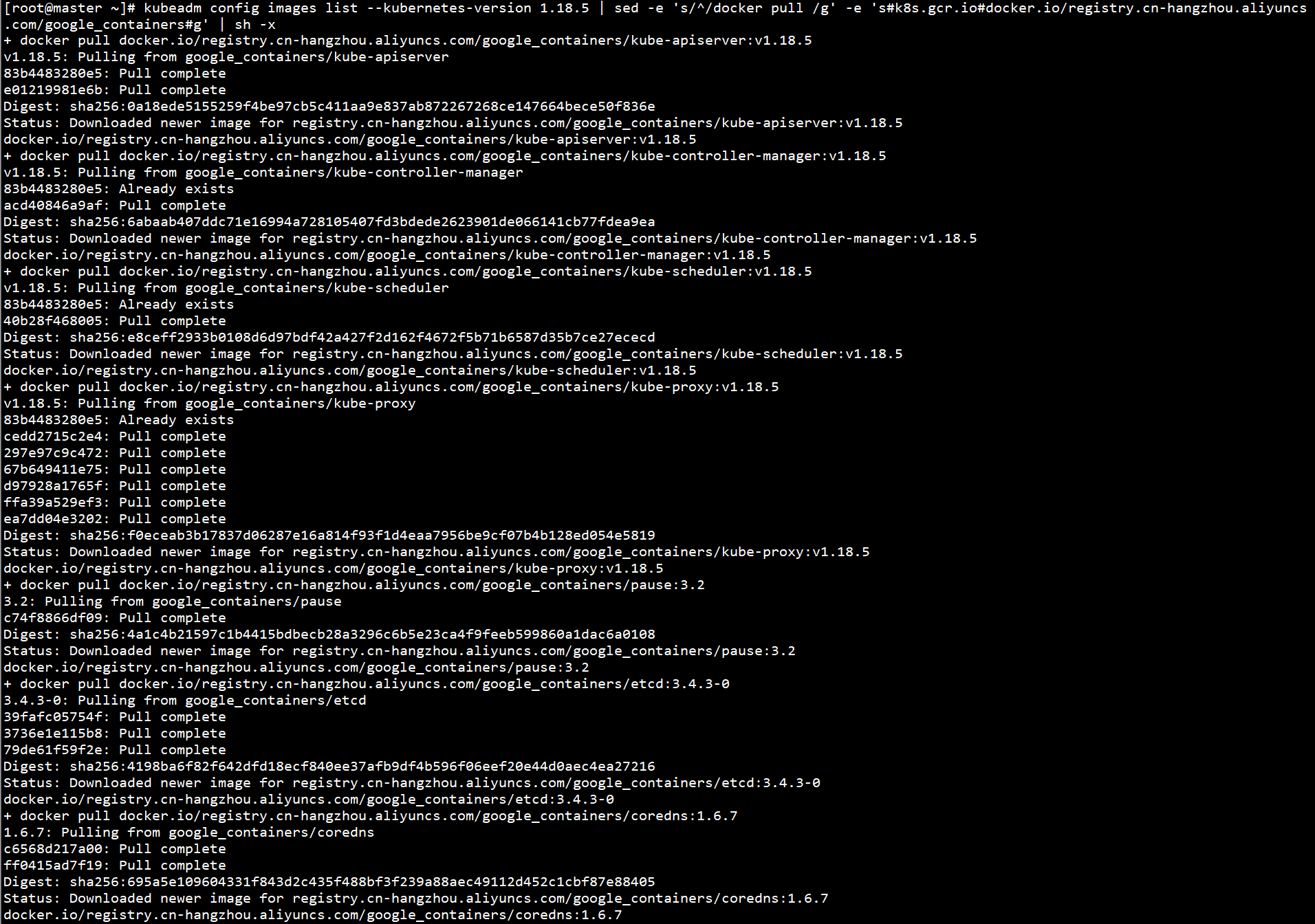
在Master节点上查看所需下载镜像：

kubeadm config images list --kubernetes-version 1.18.5



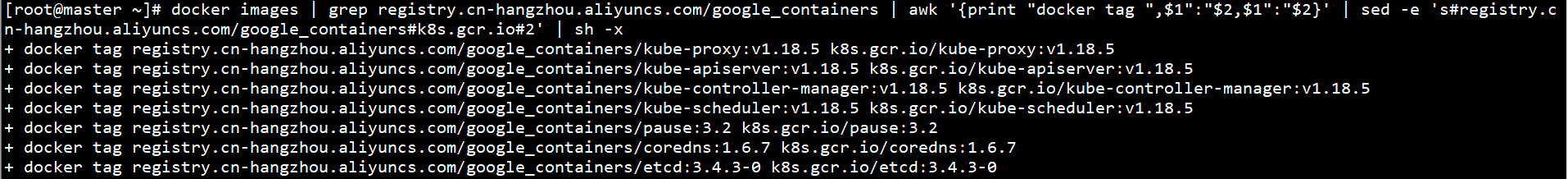
下载镜像：

kubeadm config images list --kubernetes-version 1.18.5 | sed -e 's/^/docker pull /g' -e 's#k8s.gcr.io#docker.io/registry.cn-hangzhou.aliyuncs.com/google\_containers#g' | sh -x



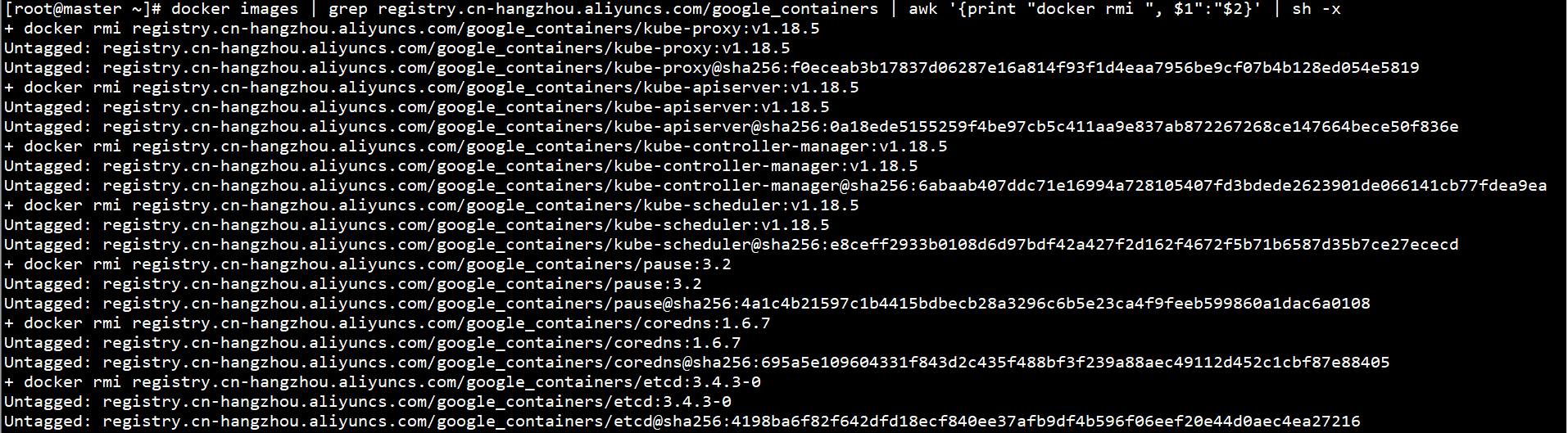
修改镜像：

docker images | grep registry.cn-hangzhou.aliyuncs.com/google\_containers | awk '{print "docker tag ",$1":"$2,$1":"$2}' | sed -e 's#registry.cn-hangzhou.aliyuncs.com/google\_containers#k8s.gcr.io#2' | sh -x



删除无用镜像：

docker images | grep registry.cn-hangzhou.aliyuncs.com/google\_containers | awk '{print "docker rmi ", $1":"$2}' | sh -x



查看镜像：

docker images



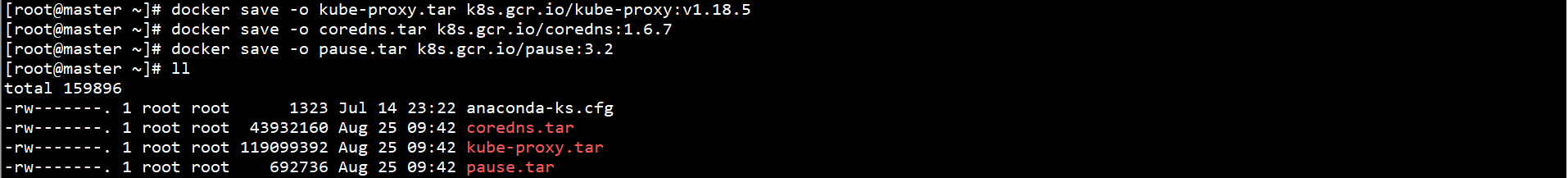
Worker节点也需部分镜像，按上述步骤下载，这里介绍另一种方法，导入镜像文件

在Master节点上保存镜像：

docker save -o kube-proxy.tar k8s.gcr.io/kube-proxy:v1.18.5

docker save -o coredns.tar k8s.gcr.io/coredns:1.6.7

docker save -o pause.tar k8s.gcr.io/pause:3.2

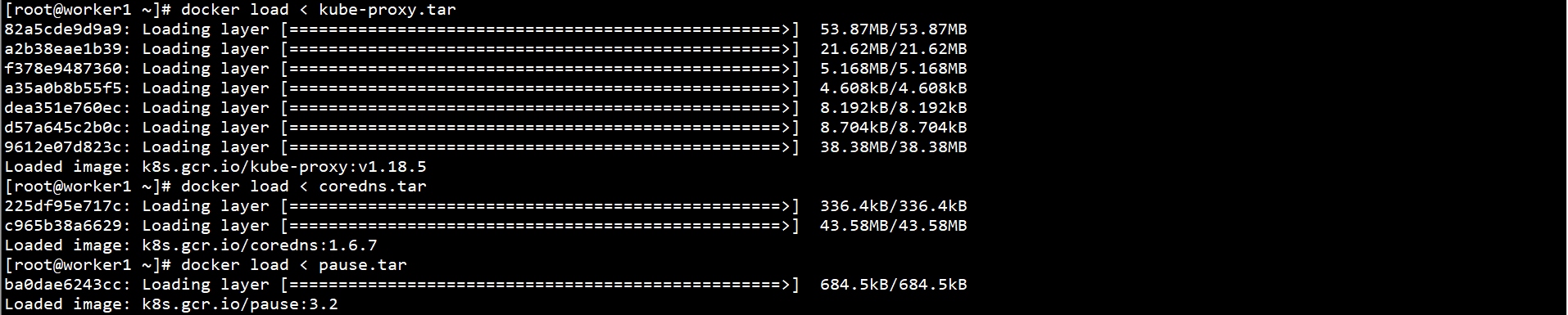


将镜像传到Worker节点上，在Worker节点上导入镜像：

docker load < kube-proxy.tar

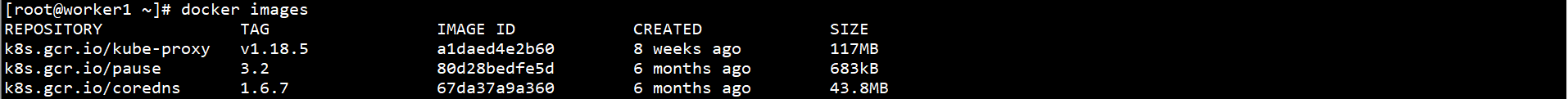
docker load < coredns.tar

docker load < pause.tar



查看镜像：

docker images



### 5、初始化集群

如下操作需在Master节点执行

初始化集群：

kubeadm init --apiserver-advertise-address=192.168.0.10 --pod-network-cidr=10.244.0.0/16 --kubernetes-version=v1.18.5



--apiserver-advertise-address表示绑定的网卡IP

--pod-network-cidr=10.244.0.0/16表示Pod将使用10.244.0.0/16网段地址，calico.yaml 文件须使用10.244.0.0/16网段，两者必须一致

--kubernetes-version=v1.18.5指定Kubernetes版本，必须与实际版本一致

### 6、初始化过程说明

初始化过程如下：

（1）kubeadm执行初始化前的检查

（2）生成token和证书

（3）生成KubeConfig文件，kubelet需要这个文件与Master通信

（4）安装Master组件，组件镜像已下载指本地

（5）安装附加组件kube-proxy和kube-dns

（6）Kubernetes Master初始化成功

（7）提示如何配置kubectl

（8）提示如何安装Pod网络

（9）提示如何注册其他节点到Cluster

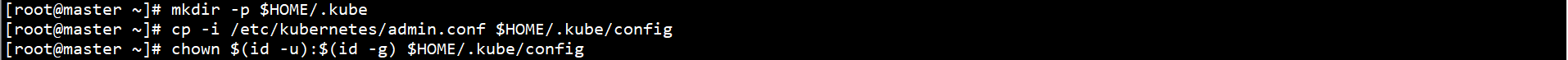
### 7、配置kubectl

在Master节点上配置kubectl：

mkdir -p $HOME/.kube

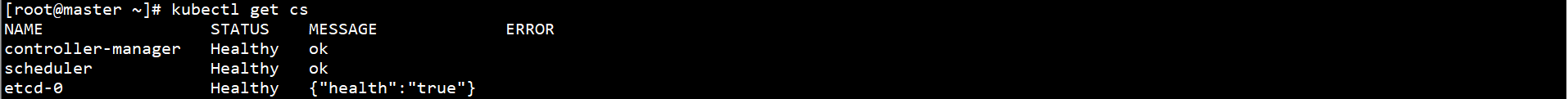
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

chown $(id -u):$(id -g) $HOME/.kube/config



查看集群状态：

kubectl get cs



### 8、安装CNI网络（calico）

在Master节点上部署CNI网络：

下载calico部署文件：

下载地址：https://docs.projectcalico.org/manifests/calico.yaml

修改calico.yaml

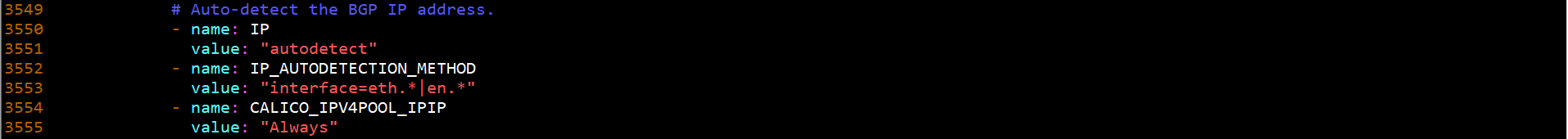
增加

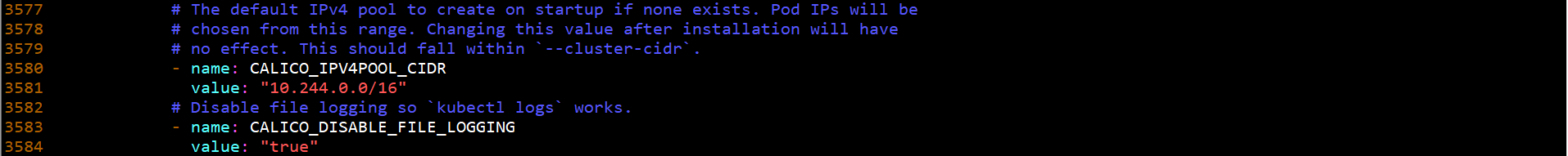
- name: IP\_AUTODETECTION\_METHOD

value: "interface=eth.\*|en.\*"

- name: CALICO\_IPV4POOL\_CIDR

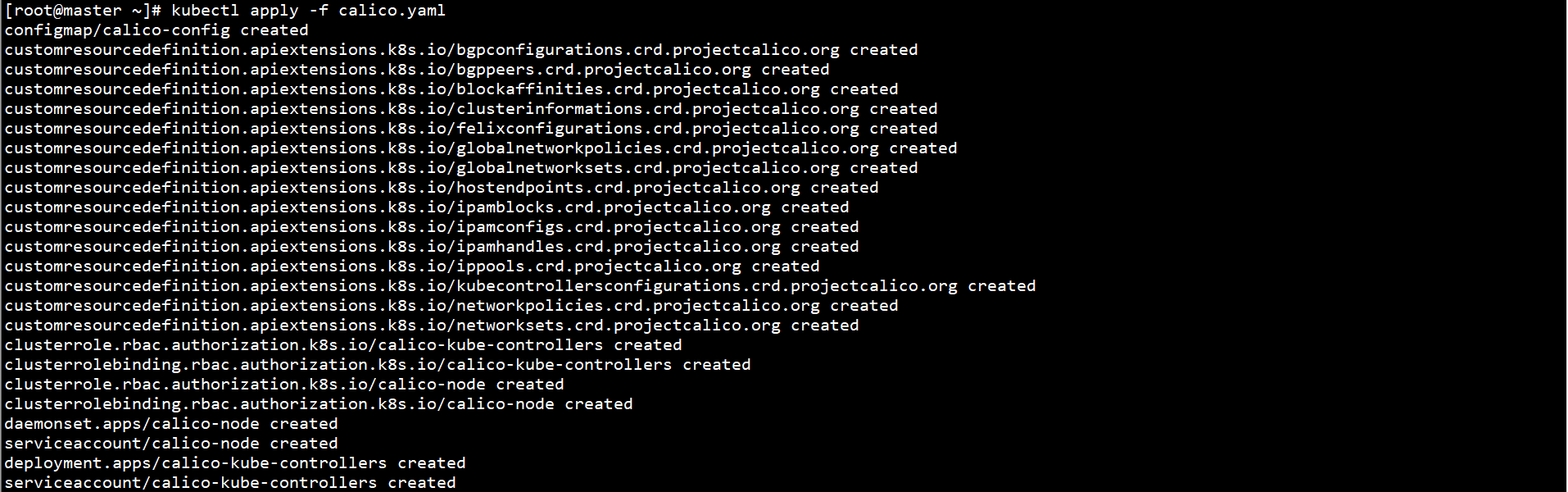
value: "10.244.0.0/16"





部署CNI网络：

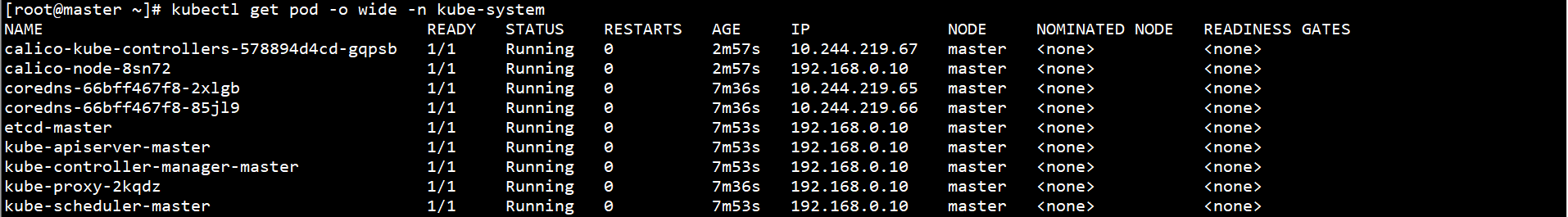
kubectl apply -f calico.yaml



calico.yaml中的CIDR需与初始化集群中的参数一致

查看Pod状态：

kubectl get pod -o wide -n kube-system

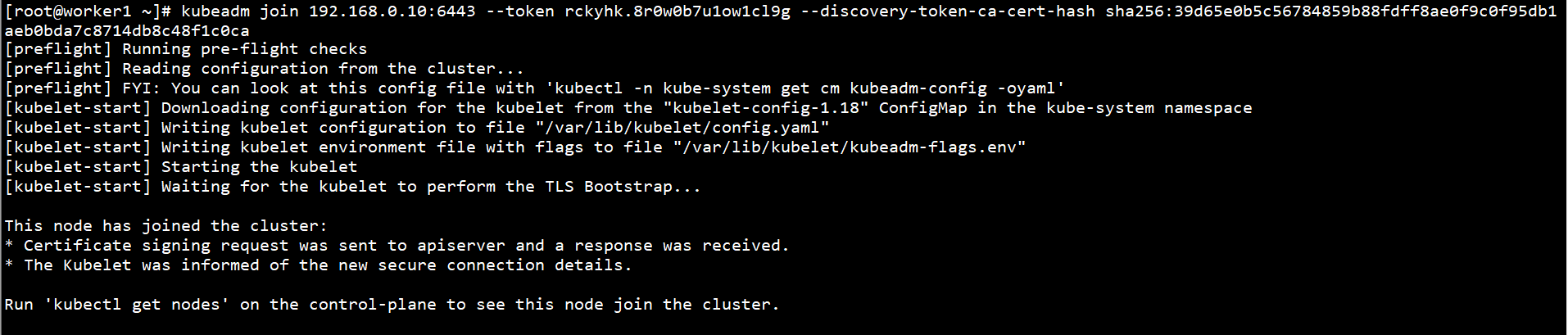


### 9、添加Worker节点

如下操作需在Worker节点执行

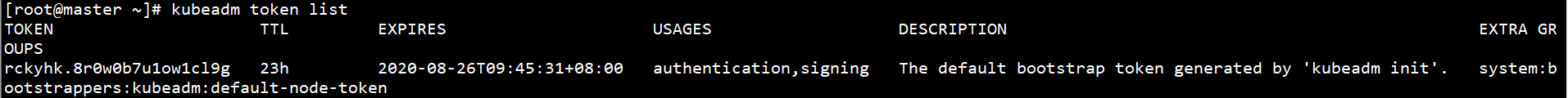
初始化集群时，显示添加节点命令：

kubeadm join 192.168.0.10:6443 --token rckyhk.8r0w0b7u1ow1cl9g --discovery-token-ca-cert-hash sha256:39d65e0b5c56784859b88fdff8ae0f9c0f95db1aeb0bda7c8714db8c48f1c0ca

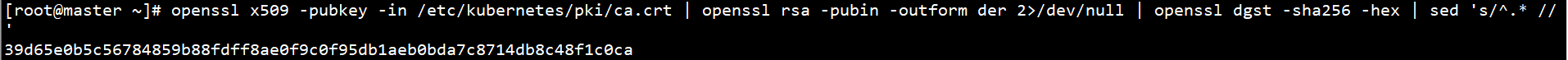


在Master节点上通过如下命令查看token和discovery-token-ca-cert-hash：

kubeadm token list



openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt | openssl rsa -pubin -outform der 2>/dev/null | openssl dgst -sha256 -hex | sed 's/^.\* //'



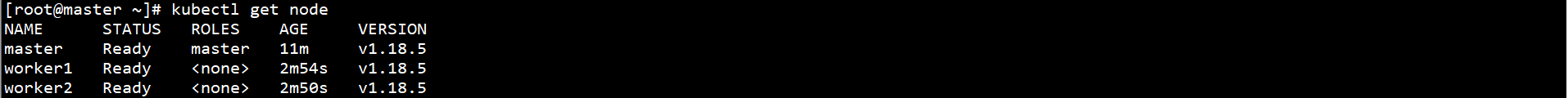
token有效时长为24h，通过如下命令创建：

kubeadm token create



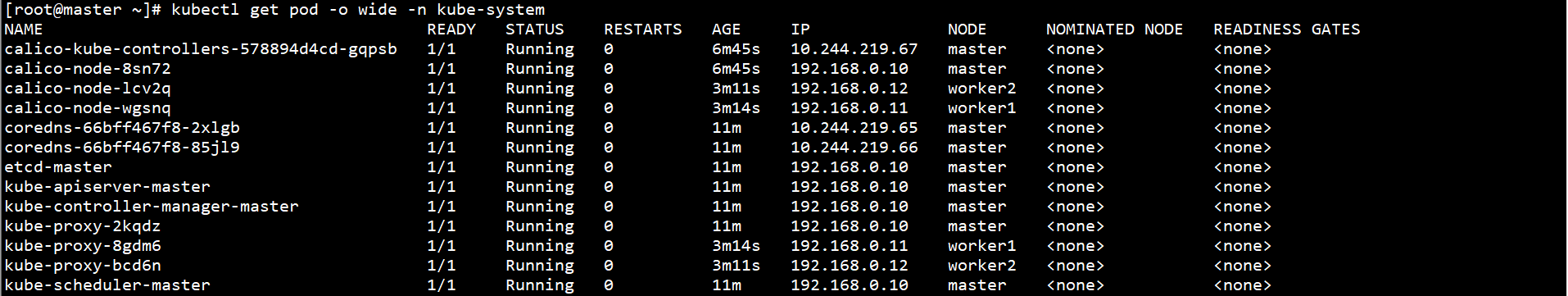
在Master节点上查看节点状态：

kubectl get node



在Master节点上查看Pod状态：

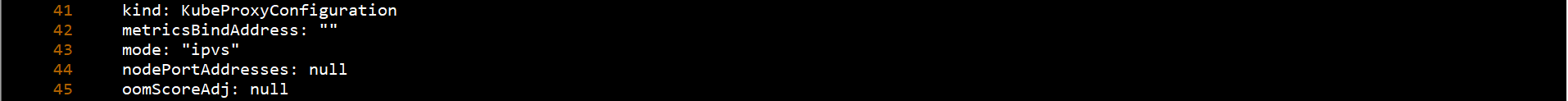
kubectl get pod -o wide -n kube-system



### 10、kube-proxy开启ipvs

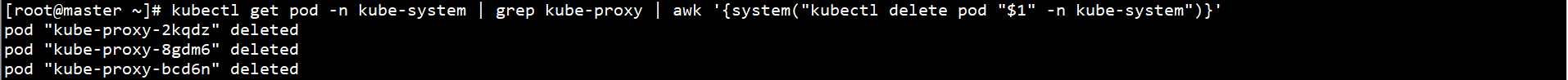
在Master节点上修改ConfigMap kube-proxy中的mode: “ipvs”：

kubectl edit configmap kube-proxy -n kube-system



在Master节点上重启各个节点上的kube-proxy pod：

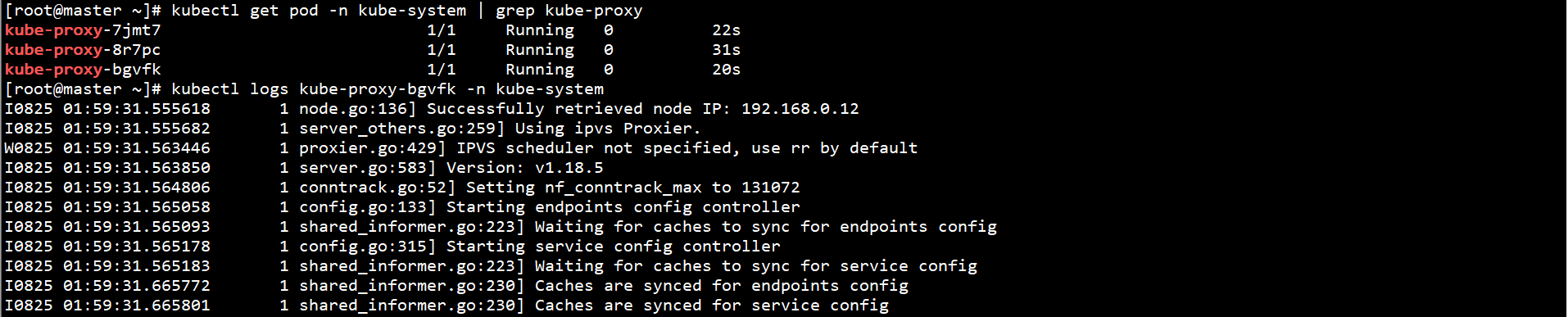
kubectl get pod -n kube-system | grep kube-proxy | awk '{system("kubectl delete pod "$1" -n kube-system")}'



验证修改成功：

kubectl get pod -n kube-system | grep kube-proxy

kubectl logs kube-proxy-4n6m8 -n kube-system



日志中打印出了Using ipvs Proxier，说明ipvs模式已经开启