**k8s环境配置**

**1、系统基础环境**

**（1）配置yum源**

不建议使用CentOS7自带的yum源，因为安装软件和依赖时会非常慢甚至超时失败。使用阿里云的源替换，执行如下命令，替换文件/etc/yum.repos.d/CentOS-Base.repo

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

或

wget -O /etc/yum.repos.d/CentOS-Base.repo <http://mirrors.163.com/.help/CentOS7-Base-163.repo>

yum clean all #清除缓存

yum makecache #生成缓存

**（2）关闭防火墙**

防火墙一定要提前关闭，否则在后续安装K8S集群的时候是大麻烦。执行下面语句关闭，并禁用开机启动：

systemctl stop firewalld #关闭防火墙

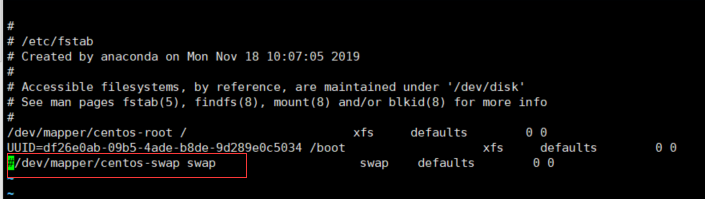
systemctl disable firewalld #禁用开机启动

**（3）关闭Swap**

类似ElasticSearch集群，在安装K8S集群时，Linux的Swap内存交换机制是一定要关闭的，否则会因为内存交换而影响性能以及稳定性。这里，我们可以提前进行设置：

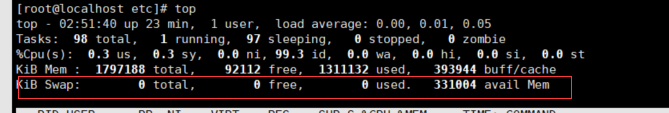
(a、执行swapoff -a可临时关闭，但系统重启后恢复

(b)、编辑/etc/fstab，注释掉包含swap的那一行即可，重启后可永久关闭，如下所示：



(c)、或直接执行：sed -i '/ swap / s/^/#/' /etc/fstab

(d)、关闭成功后，使用top命令查看，如下图所示表示正常：



**（4）关闭SeLinux**

setenforce 0

**（5）安装net-tools，用于网络命令**

yum install -y net-tools

**（6）安装Docker**

参考：《CentOS搭建Docker环境》

**2、设置主机名**

hostnamectl set-hostname centos-master

hostnamectl set-hostname centos-node1

hostnamectl set-hostname centos-node2

#查看修改结果

hostnamectl status

**3、域名解析**

编辑 /etc/hosts 文件，添加域名解析

cat <<EOF >>/etc/hosts

192.168.1.100 centos-master

192.168.1.101 centos-node1

192.168.1.102 centos-node2

EOF

**4、配置内核参数，将桥接的IPv4流量传递到iptables的链**

#创建k8s文件

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

#启用生效

modprobe br\_netfilter

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

sysctl --system

**4、确保已经关闭防火墙、Swap、SeLinux**

在《2-Hyper-V虚拟机CentOS环境配置.docx》一文有说明

**5、配置k8s的yum源**

官方仓库无法使用，建议使用阿里的仓库，执行以下命令添加kubernetes.repo仓库：

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

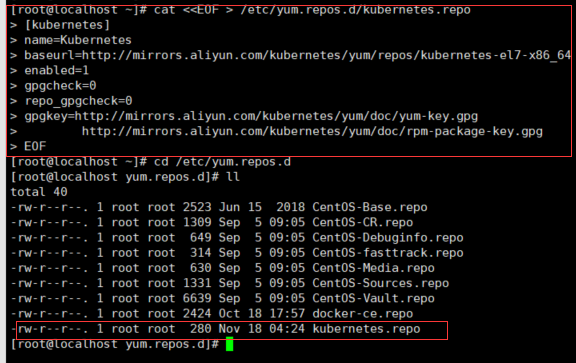
gpgcheck=0

repo\_gpgcheck=0

gpgkey=http://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg

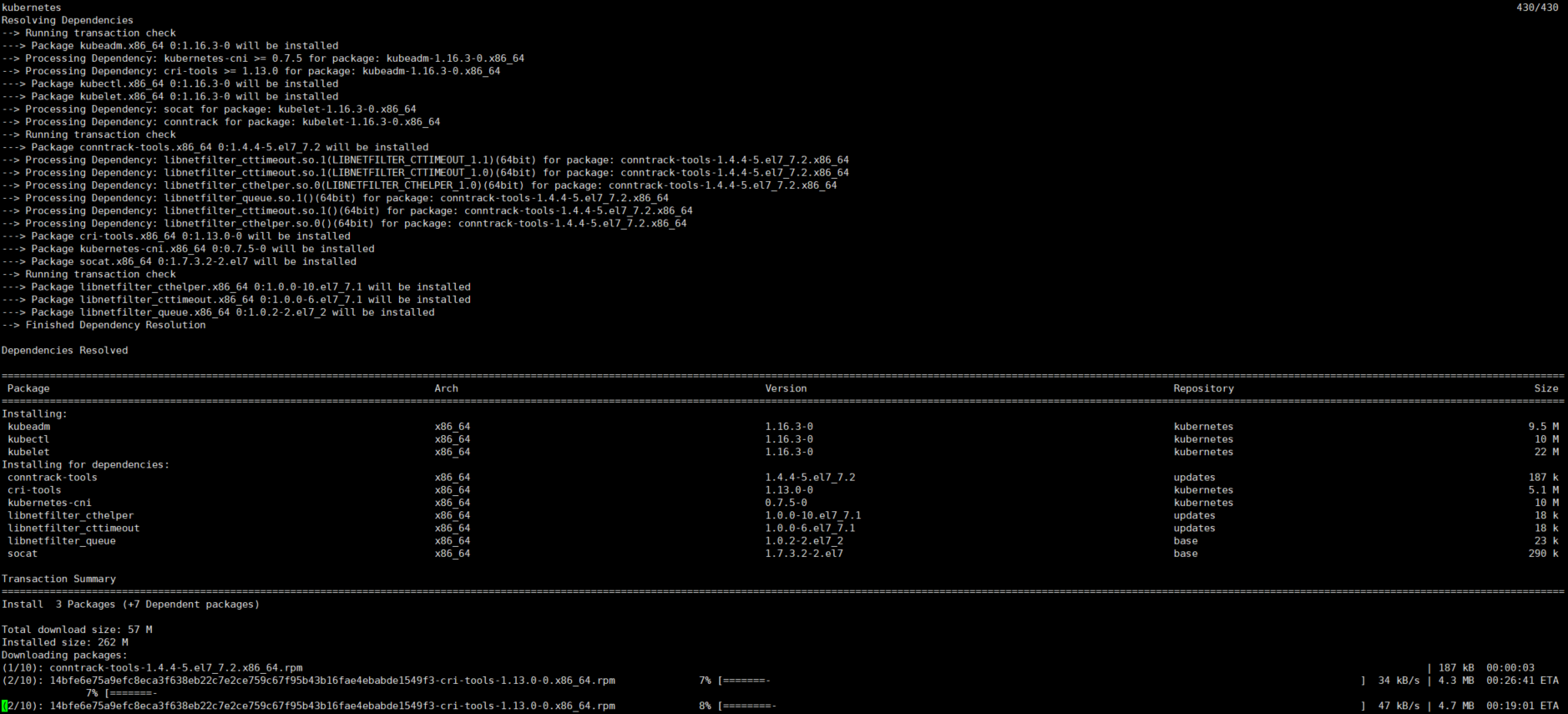
http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

EOF



**6、安装k8s组件**

yum install -y kubelet kubeadm kubectl



**如果速度太慢，可以手动下载并上传到各个节点进行rpm安装：**

地址：http://mirrors.aliyun.com/kubernetes/yum/repos/

kubulet：kubelet-1.16.3-0.x86\_64.rpm #kubeadm依赖包

kubeadm：kubeadm-1.16.3-0.x86\_64.rpm

kubectl：kubectl-1.16.3-0.x86\_64.rpm #kubeadm依赖包

kubernetes-cni：kubernetes-cni-0.7.5-0.x86\_64.rpm #kubeadm依赖包

#rpm安装

rpm -i kubernetes-cni-0.7.5-0.x86\_64.rpm kubeadm-1.16.3-0.x86\_64.rpm kubectl-1.16.3-0.x86\_64.rpm kubelet-1.16.3-0.x86\_64.rpm

**7-配置kubelet的cgroup driver ,确保cgroup drive一样**

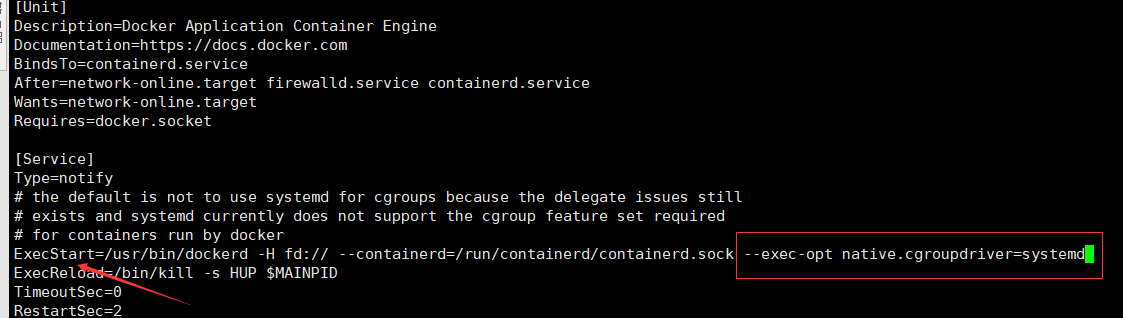
#查看

docker info | grep -i cgroup

#修改

vi /usr/lib/systemd/system/docker.service

ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --exec-opt native.cgroupdriver=systemd



#重启docker

systemctl daemon-reload

systemctl restart docker

除外，也可以创建或修改/etc/docker/daemon.json：

{

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

}

**8、启动kubelet**

注意，根据官方文档描述，安装后，要求启动kubelet：

systemctl enable kubelet && systemctl start kubelet

注：

**Kubelet**负责与其他节点集群通信，并进行本节点Pod和容器生命周期的管理。

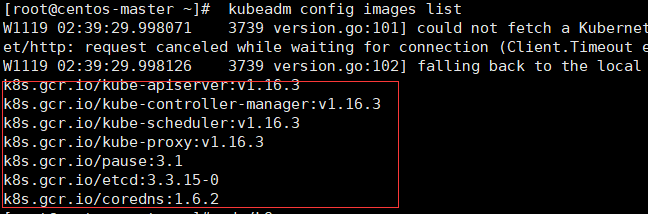
[**Kubeadm**](https://www.kubernetes.org.cn/tags/kubeadm)是Kubernetes的自动化部署工具，降低了部署难度，提高效率。

**Kubectl**是Kubernetes集群管理工具。

**9、手动下载k8s所需要的镜像**

由于国外网无法访问或是访问过慢，在部署k8s前先手动下载镜像，导入到dokcer中

kubeadm config images list #列出所需镜像



**创建批处理文件，使用国内源pull镜像**

vi kubeadm\_config\_images\_list.sh

#! /bin/bash

images=(

kube-apiserver:v1.16.3

kube-controller-manager:v1.16.3

kube-scheduler:v1.16.3

kube-proxy:v1.16.3

pause:3.1

etcd:3.3.15-0

coredns:1.6.2

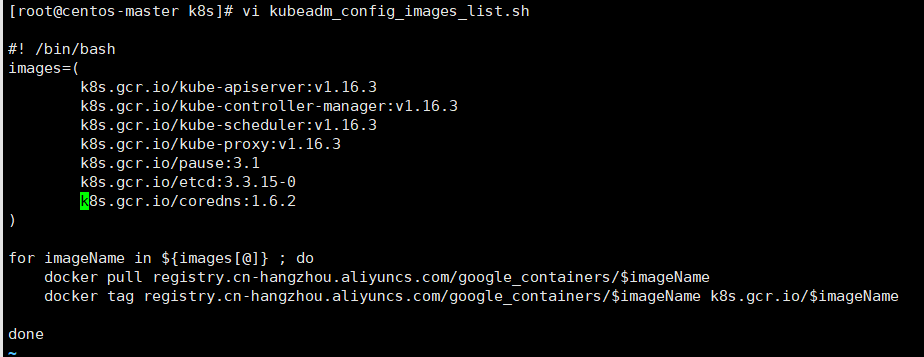
)

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

done



chmod +x kubeadm\_config\_images\_list.sh #让其变得可执行

./kubeadm\_config\_images\_list.sh #执行

**相关命令,可用于不同主机间镜像的拷贝**

打包k8s的docker镜像

docker save $(docker images | grep -v REPOSITORY | awk 'BEGIN{OFS=":";ORS=" "}{print $1,$2}') -o k8s.tar

#加载镜像

docker load -i k8s.tar

**10、准备k8s集群机器**

搭建好了k8s基础环境的虚拟机后，把现有的虚拟机centos-master用作**主节点**。为了减少工作量，在centos-master完成以上安装后，利用Hyper-V的克隆功能，克隆出两个完全一样的虚拟机作为**工作节点**centos-node1、centos-node2。三者角色为：

centos-master：Master ->192.168.1.100

centos-node1：Woker ->192.168.1.101

centos-node2：Woker ->192.168.1.102