**k8s部署Master主节点**

参考：http://docs.kubernetes.org.cn/457.html

<https://www.kubernetes.org.cn/5462.html>

<https://www.kubernetes.org.cn/5846.html>

http://www.coderdocument.com/docs/kubernetes/v1.14/index.html

**1、前言**

搭建好了基础的虚拟机环境后，把现有的虚拟机centos-master用作**主节点**。为了减少工作量，在centos-master安装Kubernetes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

centos-node1：Woker

centos-node2：Woker

**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

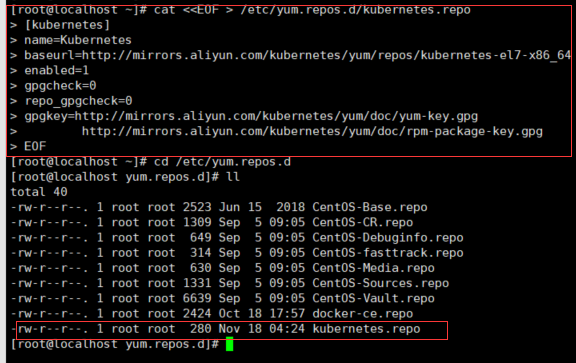
gpgcheck=0

repo\_gpgcheck=0

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

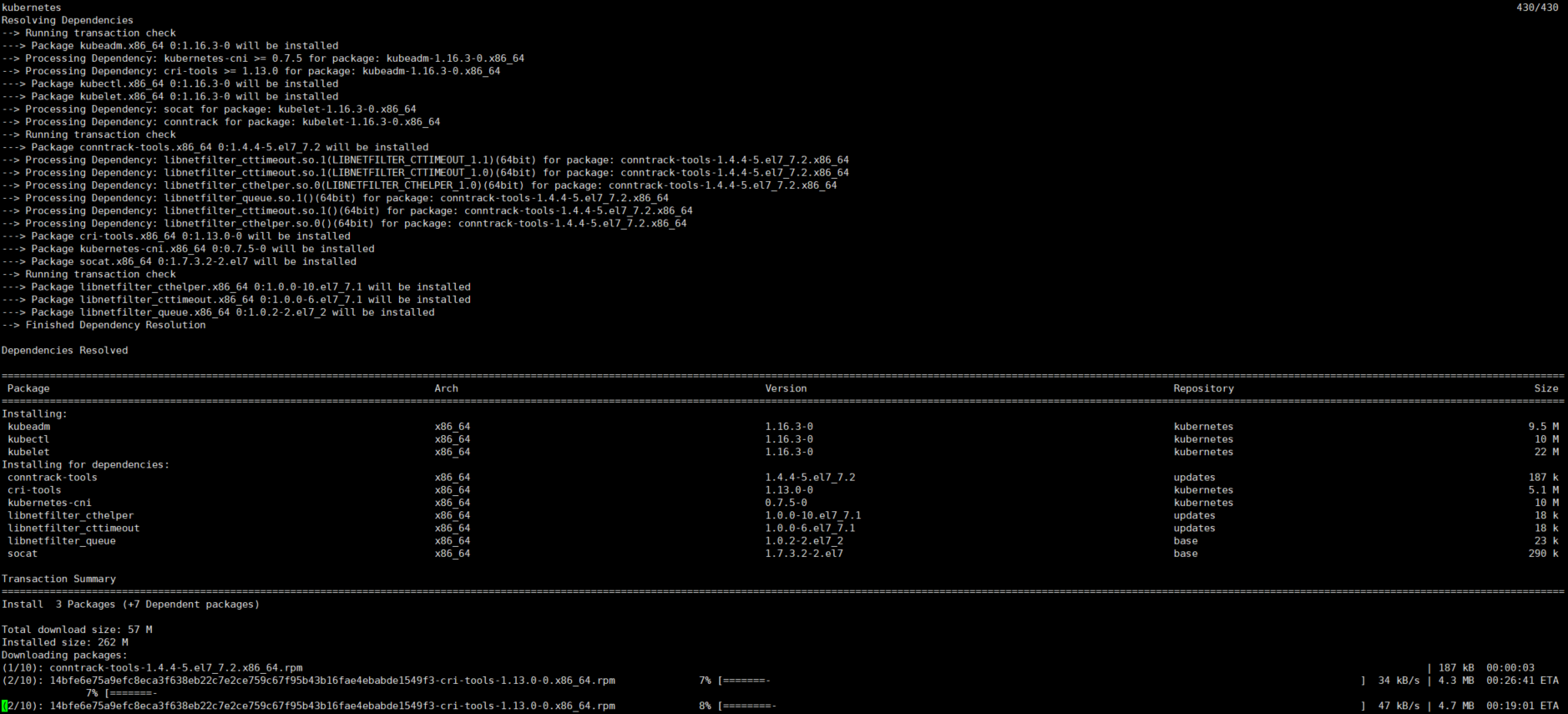
http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

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**

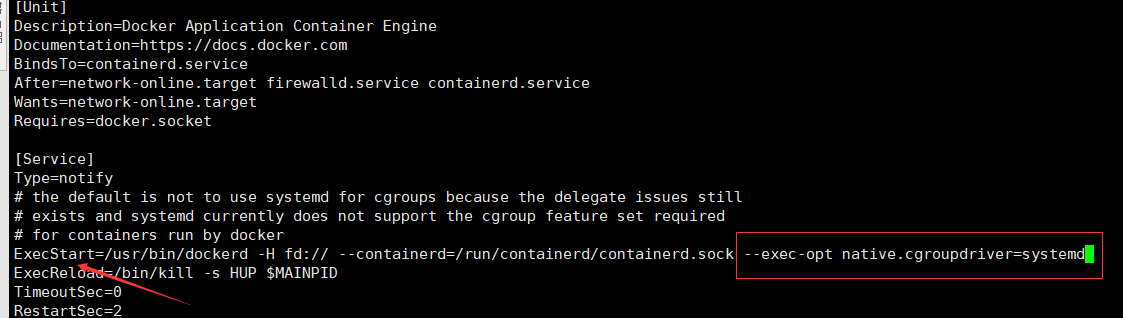
#查看

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。

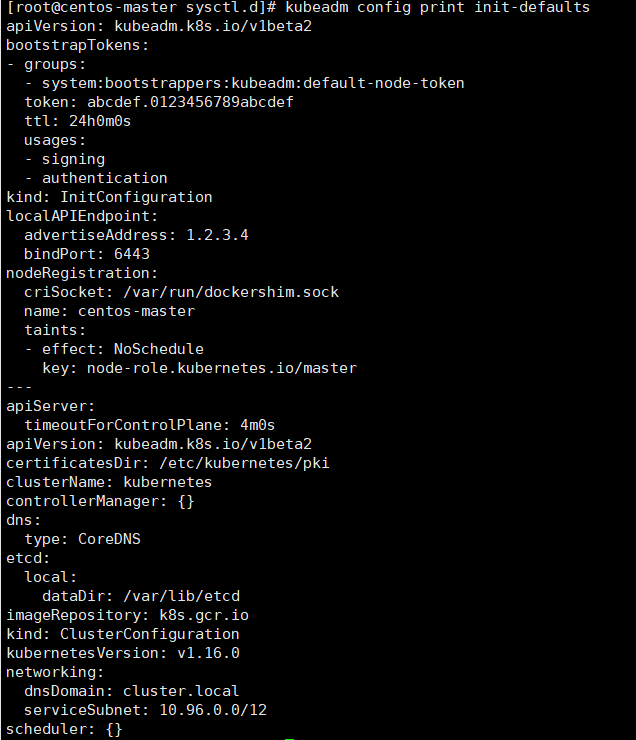
**Kubectl**是Kubernetes集群管理工具。

**=========到此，我们克隆两个虚机环境为Node1/Nodes节点再继续===========**

**9、通过指定配置文件(kubeadm.yaml)部署master 节点（此步骤可能会失败，请参考第10步）**

查看kubeadm init命令初始化默认参数

kubeadm config print init-defaults



新建kubeadm.yaml文件，将上面的内容复制一份，修改如下：

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: **192.168.1.100**

bindPort: 6443

nodeRegistration:

criSocket: /var/run/dockershim.sock

name: centos-master

taints:

- effect: NoSchedule

key: node-role.kubernetes.io/master

---

apiServer:

timeoutForControlPlane: 4m0s

apiVersion: kubeadm.k8s.io/v1beta2

certificatesDir: /etc/kubernetes/pki

clusterName: kubernetes

controllerManager: {}

dns:

type: CoreDNS

etcd:

local:

dataDir: /var/lib/etcd

imageRepository: **registry.aliyuncs.com/google\_containers**

kind: ClusterConfiguration

kubernetesVersion: v1.16.0

networking:

dnsDomain: cluster.local

serviceSubnet: 10.96.0.0/12

scheduler: {}

#执行初始化

kubeadm init –config kubuadm.yaml --ignore-preflight-error=Swap

说明：

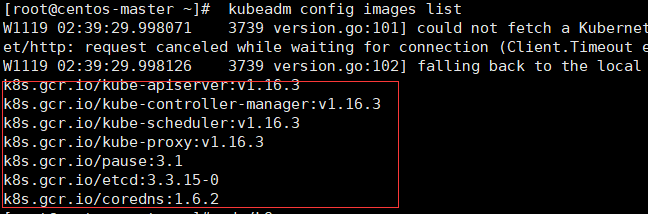
advertiseAddress：API服务器将通知它正在监听的IP地址，监听的地址为master主机IP

imageRepository:指定阿里云镜像仓库这一步很关键，由于kubeadm 默认从官网k8s.grc.io下载所需镜像，国内无法访问，因此需要通过–image-repository指定阿里云镜像仓库地址，很多新手初次部署都卡在此环节无法进行后续配置。

**10、手动下载k8s所需要的镜像（如果9步骤成功，请跳过）**

由于国外网无法访问或是访问过慢，导致无法正常执行9步骤，可以手动下载镜像，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

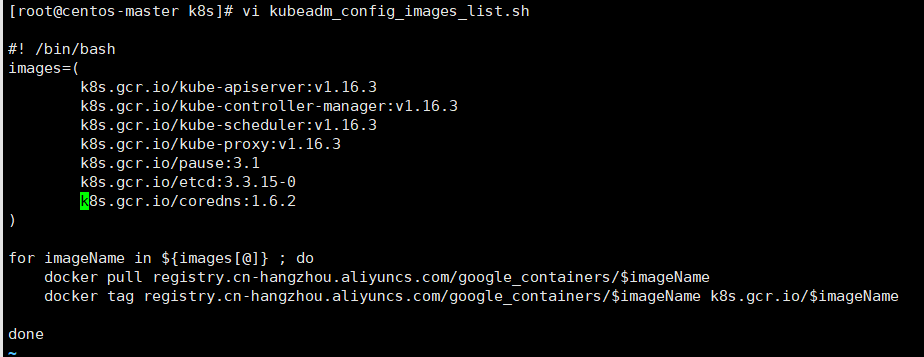
)

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

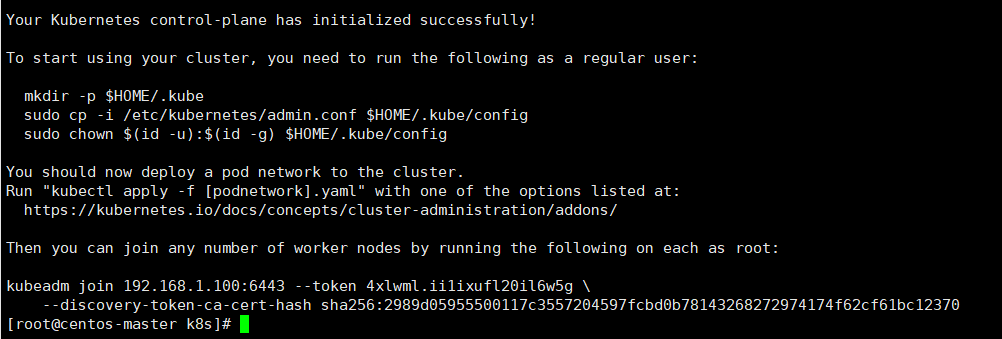
#加载镜像

docker load -i k8s.tar

**。。。。。。。。。。。。。。。。。。。。。。。。。。。。。等待下载。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。**

镜像下载完后，执行如下：

kubeadm init --apiserver-advertise-address=192.168.1.100 --pod-network-cidr=10.244.0.0/16



含义：

1.选项--pod-network-cidr=10.244.0.0/16表示集群将使用Flannel网络，这里需要提前指定Flannel的子网范围

2.选项–kubernetes-version=v1.16.3指定K8S版本，这里必须与之前导入到Docker镜像版本v1.16.3一致，否则会访问谷歌去重新下载K8S最新版的Docker镜像

3.选项--apiserver-advertise-address=192.168.1.100表示绑定的网卡IP，这里一定要绑定前面提到的eth0网卡

4.若执行kubeadm init出错或强制终止，则再需要执行该命令时，需要先执行kubeadm reset重置

kubeadm reset

ifconfig cni0 down

ip link delete cni0

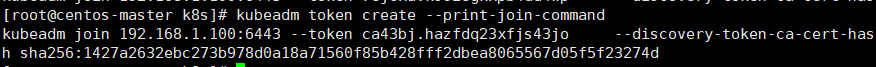
ifconfig flannel.1 down

ip link delete flannel.1

rm -rf /var/lib/cni/

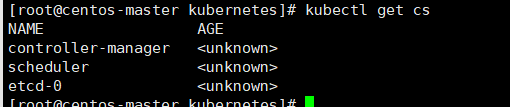
添加节点,通过以下命令获取join command

kubeadm token create --print-join-command



**11、查看一下集群状态**

执行kubectl get cs查看集群状态，



可能会出现：The connection to the server localhost:8080 was refused

原因：kubenetes master没有与本机绑定，集群初始化的时候没有设置，执行如下：

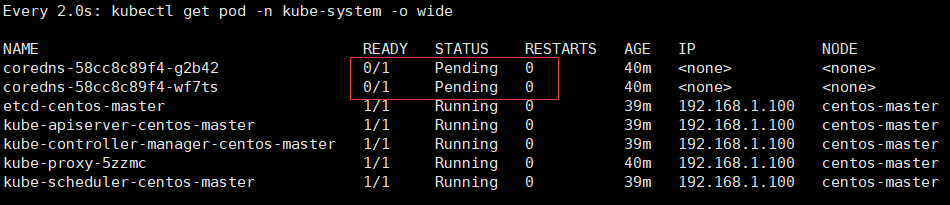
#admin.conf这个文件主要是集群初始化的时候用来传递参数的

export KUBECONFIG=/etc/kubernetes/admin.conf

**12、检查 master 初始化结果**

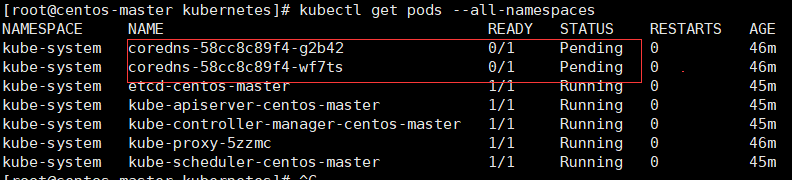
执行如下命令，等待 3-10 分钟，直到所有的容器组处于 Running 状态

watch kubectl get pod -n kube-system -o wide



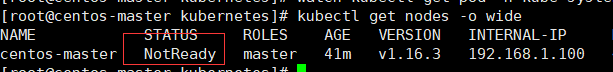
kubectl get pods --all-namespaces #查看所有pod列表

kubectl get pods -n kube-system #查看指定namespacer的pod列表



查看 master 节点初始化结果

kubectl get nodes -o wide



从上面，我们发现coredns状态一直处理Pending（阻塞）状态，而centos-master状态为NotReady，原因是dns的容器是使用bridge网络，需要配置网络才能跑起来，详见13步骤

**13、安装Pod Network，选择Flannel**

参考文档：

K8s addons：https://kubernetes.io/docs/concepts/cluster-administration/addons/

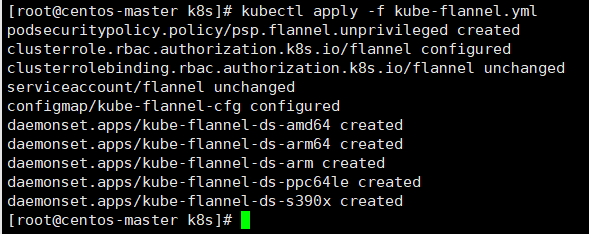
flannel：https://github.com/coreos/flannel/

#下载kube-flannel.yaml文件

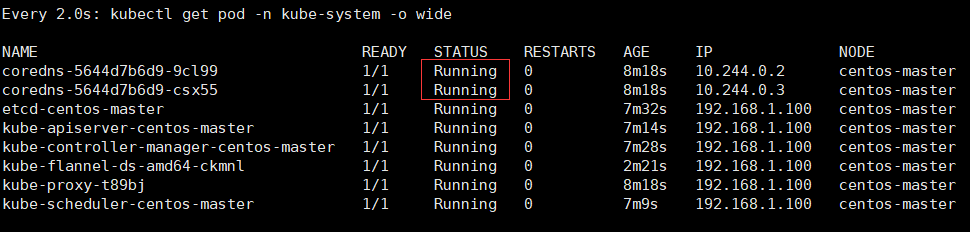
curl -O https://raw.githubusercontent.com/coreos/flannel/master/Documentation//kube-flannel.yml

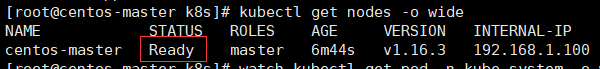
#应用到k8s

kubectl apply -f kube-flannel.yml



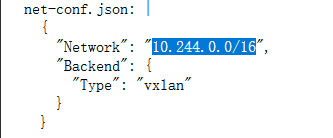
重新执行12步骤，发现coredns已正常，主节点centos-master状态为Ready

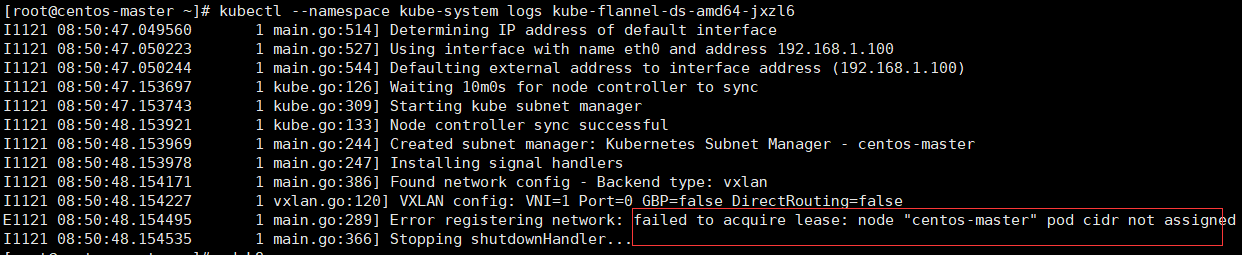




其他问题：

在安装 flannel 网络插件后，发现 kube-flannel-ds 一直处于 CrashLoopBackOff 状态，kubectl -n kube-system logs kube-flannel-ds-amd64-jxzl6查看日志(如下图)，则问题是出在默认的网段上，需要判断在 kubeadm init 初始化的时候，是否没有添加 --pod-network-cidr=10.244.0.0/16 参数，这里的 10.244.0.0/16需要与kube-flannel.yml的一致





**14、**将Master作为工作节点(可选)

K8S集群默认不会将Pod调度到Master上，这样Master的资源就浪费了。在Master（即centos-master）上，可以运行以下命令使其作为一个工作节点：  
kubectl taint nodes --all node-role.kubernetes.io/master-

在这里插入图片描述

利用该方法，我们可以不使用minikube而创建一个单节点的K8S集群

15、部署kube-proxy

kube-proxy的作用主要是负责service的实现，具体来说，就是实现了内部从pod到service和外部的从node port向service的访问

新版本目前 kube-proxy 组件全部采用 ipvs 方式负载，所以为了 kube-proxy 能正常工作需要预先处理一下 ipvs 配置以及相关依赖(每台 node 都要处理)

#安装 ansible

yum -y install ansible

# 开启ipvs

ansible k8s-node -m shell -a "yum install -y ipvsadm ipset conntrack"

ansible k8s-node -m shell -a 'modprobe -- ip\_vs'

ansible k8s-node -m shell -a 'modprobe -- ip\_vs\_rr'

ansible k8s-node -m shell -a 'modprobe -- ip\_vs\_wrr'

ansible k8s-node -m shell -a 'modprobe -- ip\_vs\_sh'

ansible k8s-node -m shell -a 'modprobe -- nf\_conntrack\_ipv4'