**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

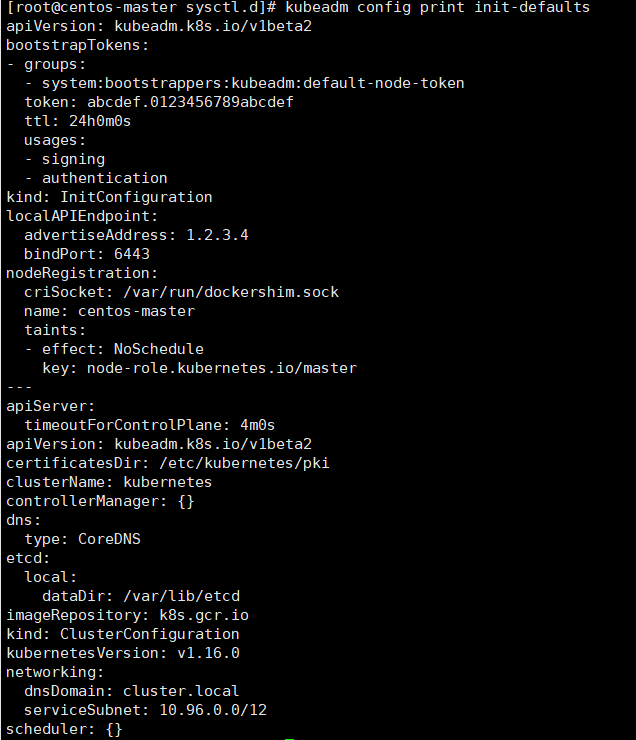
**1、前言**

在这之前已准备好三台集群机器，接下来的操作在centos-master执行

**2、部署master 节点（指定配置文件kubeadm.yaml）**

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 # 主节点IP

  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 # 使用阿里镜像库

kind: ClusterConfiguration

kubernetesVersion: v1.16.0

networking:

  dnsDomain: cluster.local

  serviceSubnet: 10.244.0.0/16 # 子网范围

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

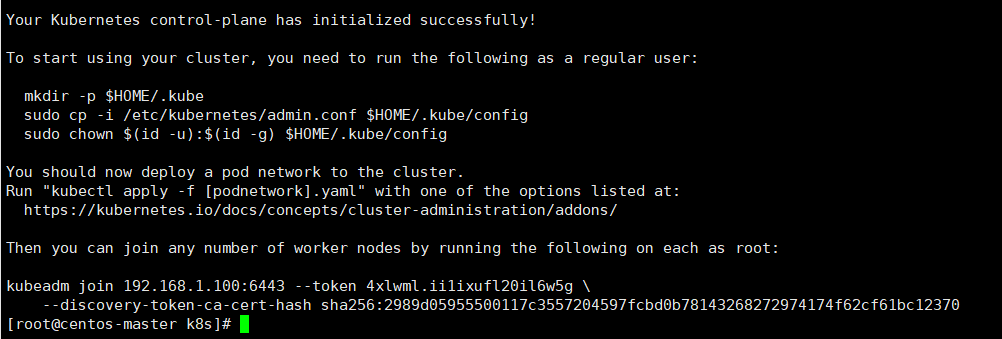
imageRepository:指定阿里云镜像仓库这一步很关键，由于kubeadm 默认从官网k8s.grc.io下载所需镜像，国内无法访问，因此需要通过–image-repository指定阿里云镜像仓库地址，很多新手初次部署都卡在此环节无法进行后续配置。

serviceSubnet：子网范围

**3、如果已提前下载好k8s所需镜像，则执行这一步**

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

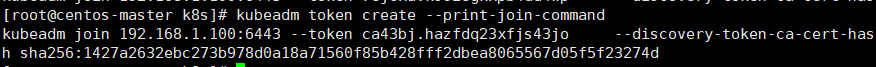
ifconfig flannel.1 down

ip link delete flannel.1

rm -rf /var/lib/cni/

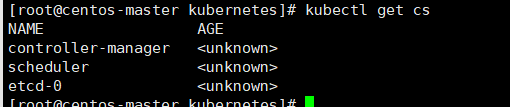
添加节点,通过以下命令获取join command

kubeadm token create --print-join-command



**4、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：

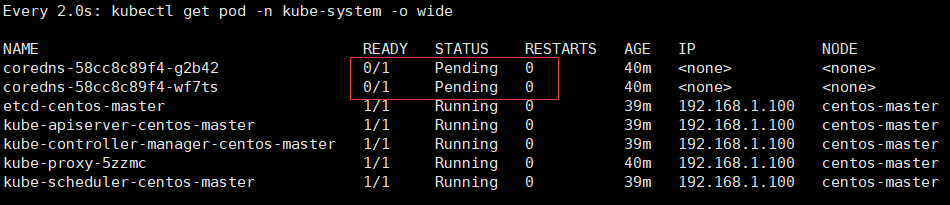
#admin.conf这个文件主要是集群初始化的时候用来传递参数的

export KUBECONFIG=/etc/kubernetes/admin.conf

**5、检查 master 初始化结果**

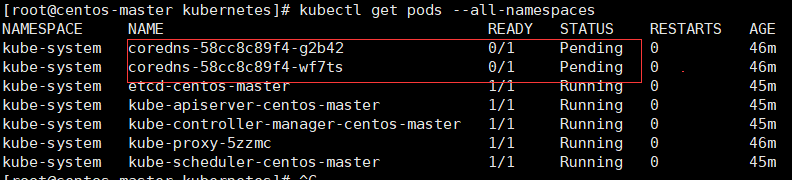
执行如下命令，等待 3-10 分钟，直到所有的容器组处于 Running 状态

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



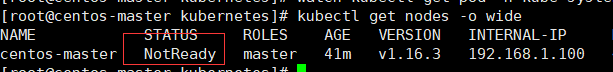
kubectl get pods --all-namespaces #查看所有pod列表

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6步骤

**6、安装Pod Network，选择Flannel**

参考文档：

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

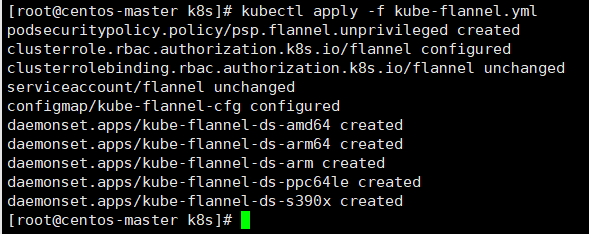
flannel：https://github.com/coreos/flannel/

#下载kube-flannel.yaml文件

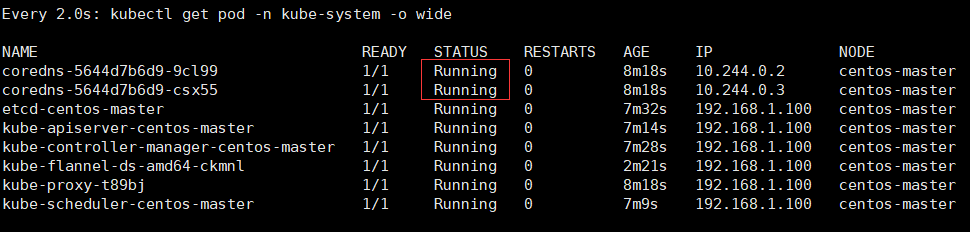
curl -O https://raw.githubusercontent.com/coreos/flannel/master/Documentation//kube-flannel.yml

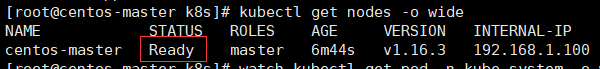
#应用到k8s

kubectl apply -f kube-flannel.yml



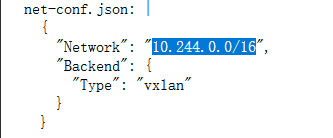
重新执行12步骤，发现coredns已正常，主节点centos-master状态为Ready

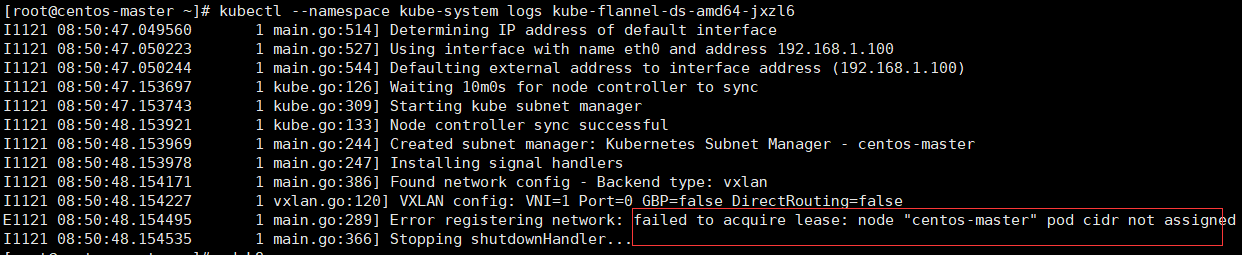




其他问题：

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的保持一致，修改kube-flannel.yml如下：





**7、**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