**修改主机名**

[root@localhost ~]# hostnamectl set-hostname k8s-master #192.168.102.130

[root@localhost ~]# hostnamectl set-hostname k8s-node1 #192.168.102.131

[root@localhost ~]# hostnamectl set-hostname k8s-node2 #192.168.102.132

注：不修改也可以

**关闭swap**

为什么关闭swap？因为swap会降低速度

编辑文件vim /etc/fstab，把文件中带有swap的那行注释掉

注：所有节点都要执行

**安装docker**

安装请查看docker教程

kubelet的配置项cgroup默认为systemd，而docker的配置项默认为cgroupfs，这会使kubelet启动失败，我们需要修改docker的cgroup配置

1. 编辑vim /etc/docker/daemon.json，没有就新建，添加exec-opts

{

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

}

1. 重启docker，systemctl restart docker

注：所有节点都要执行

**安装kubeadm、kubelet、kubectl**

1. 安装前置

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

1. 使用阿里库（google库国内访问不了）

- 添加key

curl https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -

- 添加镜像源库，在vim /etc/apt/sources.list.d/kubernetes.list中添加如下行

deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main

1. 安装

apt-get update

apt-get install -y kubelet kubeadm kubectl

笔者使用的版本是1.23.5，即如下

apt-get install -y kubelet=1.23.5-00 kubeadm=1.23.5-00 kubectl=1.23.5-00

1. 设置kubelet开机自启

systemctl enable kubelet

注：所有节点都要执行

**在master部署kubernetes**

1. 部署

kubeadm init \

--apiserver-advertise-address=192.168.102.130 \

--image-repository registry.aliyuncs.com/google\_containers \

--kubernetes-version v1.23.5 \

--service-cidr=10.1.0.0/16 \

--pod-network-cidr=10.244.0.0/16

参数说明

--apiserver-advertise-address：这个参数就是master主机的IP地址

--image-repository：这个是镜像仓库地址，这里使用阿里云仓库地址

--kubernetes-version：这是部署的k8s软件版本号，其应该和Kubelet的版本相同

--service-cidr：这个参数后的IP地址直接就套用10.1.0.0/16就可以了

--pod-network-cidr：k8s内部的pod节点之间可以使用的IP段，不能和service-cidr写一样，如果不知道怎么配，就先用这个10.244.0.0/16

1. 部署成功信息

注意，部署成功后会有如下的信息反馈

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube

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

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

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

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

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.102.130:6443 --token qc093z.xr3ltsui1loy87cs \

  --discovery-token-ca-cert-hash sha256:902969f599d324185126f53f0b07c02126063fea73f14e05297b65f102e842a0

其中最后一行是在node节点加入集群的命令

kubeadm join 192.168.102.130:6443 --token qc093z.xr3ltsui1loy87cs \

--discovery-token-ca-cert-hash sha256:902969f599d324185126f53f0b07c02126063fea73f14e05297b65f102e842a0

**在master安装pod网络插件**

在master上执行如下命令

# kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

查看插件是否运行

root@k8s-master:~# kubectl get pod --all-namespaces -o wide

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system kube-flannel-ds-p7pf7 1/1 Running 0 61m

**在nodes上执行加入集群命令**

1. Node节点加入集群

如下，我们在node节点执行kubeadm init执行成功后的反馈命令

# kubeadm join 192.168.102.130:6443 --token qc093z.xr3ltsui1loy87cs \

--discovery-token-ca-cert-hash sha256:902969f599d324185126f53f0b07c02126063fea73f14e05297b65f102e842a0

执行该命令后node节点会加入到集群中

如果你忘了该命令，或者该命令过期了（24小时内有效），可以在master上执行如下命令生成新的加入集群命令

kubeadm token create --print-join-command

1. 查看节点是否加入

在master上运行如下名

root@k8s-master:~# kubectl get nodes

NAME STATUS ROLES AGE VERSION

k8s-master Ready control-plane,master 138m v1.23.5

k8s-node1 Ready <none> 106s v1.23.5

可以看到k8s-node1已加入集群

**还原kubernetes**

执行命令

kubeadm reset

该命令尽力还原由 kubeadm init 或 kubeadm join 所做的更改，如果你在执行这2行命令出错了时，你可以使用kubeadm reset还原