k8s集群

一、准备条件：准备三台服务器，一个master,两个nodes。三台服务器都是双核2g的配置

192.168.233.133 master

192.168.233.134 node01

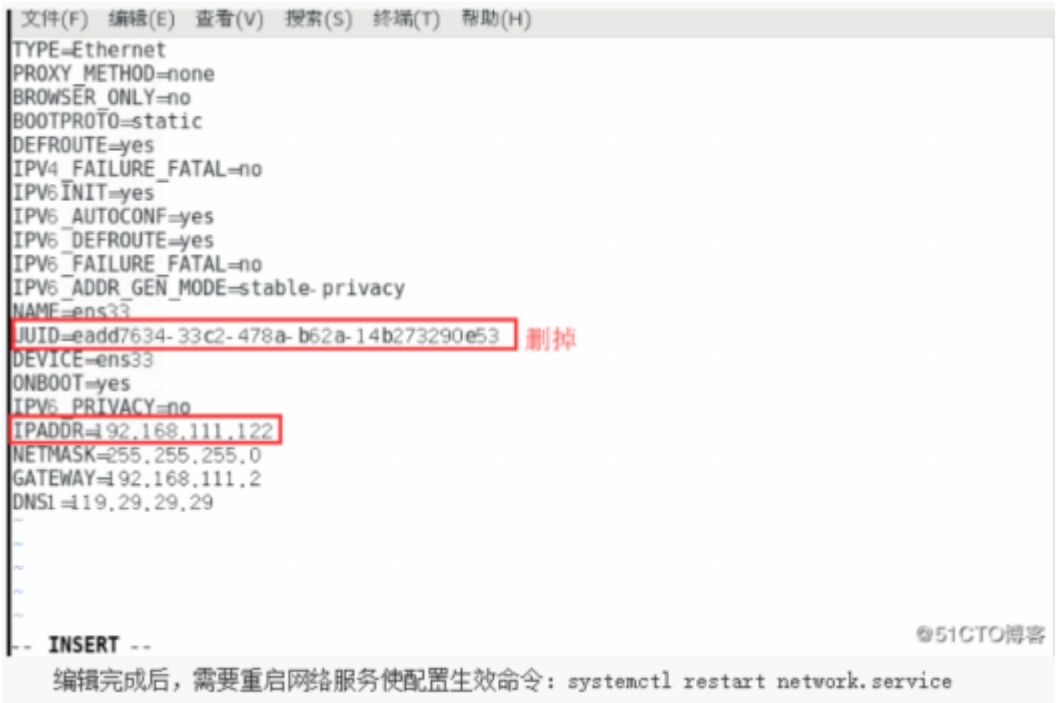
192.168.233.131 node02

如果是用虚拟机链接复制过来的，要修改服务器的名称，

1）进入以下路径，选择ifcfg-en开头的文件，进行修改

```
cd /etc/sysconfig/network-scripts
```


```
vi ifcfg-ens16777736
```




```
systemctl restart network.service
```

2)设置主机名称

```
hostnamectl set-hostname node01
```

查看主机名称：

```
cat /etc/hostname
```

3) 最好在开始之前更新下yum源

```
yum update
```

二、在三个服务器中都需要做的工作，设置yum源，下载docker和kubeadm、kubelet、kubectl

1）编辑 /etc/hosts 文件，添加域名解析。

```
cat <<EOF >>/etc/hosts

192.168.233.133  master

192.168.233.134 node01

192.168.233.131 node02

EOF
```

2）、关闭防火墙、selinux和swap。

```
systemctl stop firewalld

systemctl disable firewalld

setenforce 0

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

swapoff -a

sed -i 's/.*swap.*/#&/' /etc/fstab
```

3）、配置内核参数，将桥接的IPv4流量传递到iptables的链

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

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

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

EOF

sysctl --system
```

4）、配置国内yum源

```
yum install -y wget

mkdir /etc/yum.repos.d/bak && mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/bak

wget -O /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.cloud.tencent.com/repo/centos7_base.repo

wget -O /etc/yum.repos.d/epel.repo http://mirrors.cloud.tencent.com/repo/epel-7.repo

yum clean all && yum makecache
```

配置国内Kubernetes源

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

[kubernetes]

name=Kubernetes

baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64/

enabled=1

gpgcheck=1

repo_gpgcheck=1

gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

EOF
```

配置 docker 源

```
wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O
/etc/yum.repos.d/docker-ce.repo
```

5）、软件安装

5.1 docker的安装,这里安装指定版本18.06

```
yum install -y docker-ce-18.06.1.ce-3.el7
systemctl enable docker && systemctl start docker
docker --version # 查看版本
```

5.2 安装kubeadm、kubelet、kubectl

```
yum install -y kubelet kubeadm kubectl
systemctl enable kubelet
```

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。

四、master节点部署

1）在master进行Kubernetes集群初始化。

```
kubeadm init --kubernetes-version=1.14.2 \

--apiserver-advertise-address=192.168.233.133 \

--image-repository registry.aliyuncs.com/google_containers \

--service-cidr=10.1.0.0/16 \

--pod-network-cidr=10.244.0.0/16
```

注：定义POD的网段为: 10.244.0.0/16，其中的apiserver-advertise-address是主机的ip地址，

–image-repository指定阿里云镜像仓库地址。

执行完成以后，会有这段代码，这段代码是为了后面的节点加入集群准备的。例如：

```
kubeadm join 192.168.233.133:6443 --token kf5egw.qg9dd1u1glyvtkfa \
    --discovery-token-ca-cert-hash sha256:c1a590f1ce5381a92616ddd26eee98299e538ab40e5bde92b603db54243e0507
```

2）配置kubectl工具

```
mkdir -p /root/.kube

cp /etc/kubernetes/admin.conf /root/.kube/config

kubectl get nodes

kubectl get cs
```

3)部署flannel网络

```
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/a70459be0084506e4ec919aa1c114638878db1
1b/Documentation/kube-flannel.yml
```

五、部署node节点：

1)允许kubelet

```
systemctl enable kubelet
```

2)将节点加入集群,一定在join完成之后再看kebelet的状态是否正常，因为再join之前不正常，子节点需要有个指令让kubelet怎么去工作。

```
kubeadm join 192.168.233.133:6443 --token kf5egw.qg9dd1u1glyvtkfa \
    --discovery-token-ca-cert-hash
sha256:c1a590f1ce5381a92616ddd26eee98299e538ab40e5bde92b603db54243e0507
```

3)成功后，开启kubelet，查看状态，是running就是正常的。

```
systemctl start kubelet
```

```
[root@node02 network-scripts]# systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/kubelet.service.d
           └─10-kubeadm.conf
   Active: active (running) since Fri 2019-06-21 12:40:33 CST; 1h 10min ago
     Docs: https://kubernetes.io/docs/
 Main PID: 60177 (kubelet)
   Memory: 44.0M
```

六、集群状态检测

1） master节点的输出,查看节点是否正常，STATUS内容为Ready时，则说明集群状态正常。

```
kubectl get nodes
```

```
[root@master k8s]# kubectl get nodes
NAME     STATUS   ROLES    AGE     VERSION
master   Ready    master   27h     v1.14.3
node01   Ready    <none>   6h59m   v1.14.3
node02   Ready    <none>   73m     v1.14.3
```

七、创建nginx和tomcat查看是否能正常访问。

1）再master服务器上创建文件夹，些replationcontroller和service文件。

```
mkdir /usr/local/k8s
vi /usr/local/k8s/mytomcat.rc.yaml
```

mytomcat.rc.yaml内容如下：

```
apiVersion: v1
kind: ReplicationController
metadata:
 name: mytomcat
spec:
 replicas: 2
 selector:
  app: mytomcat
 template:
  metadata:
   labels:
    app: mytomcat
  spec:
   containers:
    - name: mytomcat
      image: tomcat:7-jre7
      ports:
      - containerPort: 8080
```

2)创建mytomcat.svc.yaml文件

```
vi /usr/local/k8s/mytomcat.svc.yaml
```

内容如下：

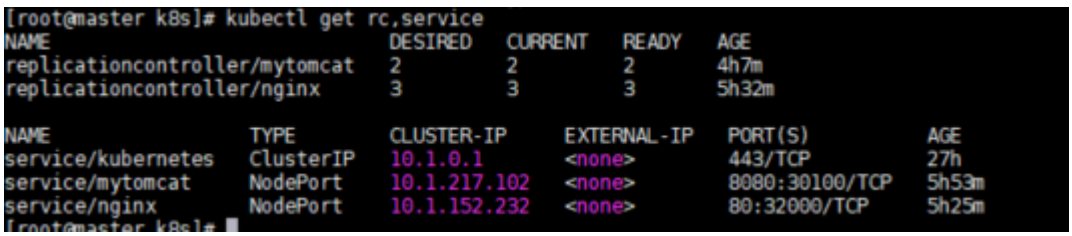
```
apiVersion: v1
kind: Service
metadata:
 name: mytomcat
spec:
 type: NodePort
 ports:
  - port: 8080
    nodePort: 30001
 selector:
  app: mytomcat
```

3）启动命令，创建service

```
kubectl create -f mytomcat.rc.yaml
kubectl create -f mytomcat.svc.yaml
```

再master节点上查看。

```
kubectl get rc,service
```



查看是否能访问：

```
curl 10.1.217.102:8080
```

也可以再浏览器访问，例如：http://192.168.233.133:30100/

或者查看这个service在哪个node上，也可以用该node的ip进行访问。

```
kubectl get pods -o wide
```

```
[root@master k8s]# kubectl get pods -o wide
NAME                READY   STATUS        RESTARTS   AGE    IP           NODE     NOMINATED NODE   READINESS GATES
mytomcat-qzwmb      1/1     Running       0          8m57s  10.244.2.8   node01   <none>           <none>
mytomcat-r52rh      1/1     Running       0          8m57s  10.244.2.7   node01   <none>           <none>
nginx-65f88748fd-pq94w   1/1     Terminating   0          23h    10.244.1.2   node02   <none>           <none>
nginx-65f88748fd-xh272   1/1     Running       0          3h9m   10.244.2.2   node01   <none>           <none>
nginx-955lp         1/1     Running       0          93m    10.244.2.5   node01   <none>           <none>
nginx-hwtnk         1/1     Running       0          93m    10.244.2.6   node01   <none>           <none>
nginx-wgx7q         1/1     Running       0          93m    10.244.2.4   node01   <none>           <none>
```

八、总结有坑的地方：

1）浏览器无法打开tomcat?

node节点上的tomcat访问不了，好像是网关没有打通。

```
iptables -S
```

将FORWARD DROP 改成 ORWARD ACCEPT

命令:

```
iptables -P FORWARD ACCEPT

#机器重启之后，又恢复DROP了，再此加一条防止重启还原DROP的命令
sleep 60 && /sbin/iptables -P FORWARD ACCEPT
#再查看
sudo iptables -S
```

2)加入子节点的时候，token失效，需要重新产生token

```
[join] Reading configuration from the cluster...
[join] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
unable to fetch the kubeadm-config ConfigMap: failed to get config map: Unauthorized
```

2.1

```
kubeadm token create  生成
```

```
kubeadm token list  查看
```

```
[root@master k8s]# kubeadm token list
TOKEN                     TTL         EXPIRES                     USAGES                   DESCRIPTION                                                EXTRA GROUPS
il4nav.mr4uwfh4cahts6lz   <invalid>   2019-06-21T17:38:14+08:00   authentication,signing   The default bootstrap token generated by 'kubeadm init'.   system:bootstrappers:kubeadm:default-node-token
kf5egw.qg9dd1ulglyvtkfa   22h         2019-06-22T19:42:40+08:00   authentication,signing   <none>                                                     system:bootstrappers:kubeadm:default-node-token
```

再添加节点的时候将上面的join后跟着的token换成新生成的就可以了。

3）再节点安装的时候启动kubelet报错，子节点必须再join进master节点后，kubelet的状态才会是正常的。

4）卸载docker https://www.jianshu.com/p/6a749f85b0c0

卸载kubernetes https://www.jianshu.com/p/4b22b5d2f69b

具体详情可以参考：https://www.kubernetes.org.cn/5462.html 安装k8s

https://www.cnblogs.com/spll/p/10075781.html 解决一些踩坑的点

https://blog.csdn.net/nklinsirui/article/details/80583971 子节点kubelet启动失败

https://www.2cto.com/net/201811/787567.html 修改主机名