**Kylinux银河麒麟v10操作系统扩容node节点操作文档**

## **目的**

用于在产品标准k8s三节点集群后，添加k8s的worker节点。

## **适用范围**

ECM V6.3~V6.6版本，标准K8S集群部署。

## **环境说明**

1、三台服务器IP为：192.168.251.34 192.168.251.36 192.168.251.247；

2、三台k8s的master节点+复用的worker节点，即这三台服务器同时为master与worker节点，同时需要额外添加一台worker节点；

/etc/hosts文件如下：

192.168.251.243   k8s-master-lb

192.168.251.34   k8s-master01

192.168.251.36   k8s-master02

192.168.251.247   k8s-master03

192.168.251.34   k8s-node01

192.168.251.36   k8s-node02

192.168.251.247   k8s-node03

192.168.251.175   k8s-node04    #新增的节点

192.168.251.176   registry.edoc2.com

192.168.251.243   edoc2v5.rancher.pre.com

​

## **操作步骤**

本文除非特殊说明，操作全部操作在 k8s-master01（k8s-node01复用）节点操作。

#### **1、新增节点设置主机名：**

在新增服务器主机节点操作执行：

hostnamectl set-hostname k8s-node04

​

#### **2、在k8s-master01（k8s-node01复用）节点操作：**

注意：所有操作均在k8s-master01节点操作，master节点主机是k8s-master01、k8s-master02、k8s-master03，master和node复用是k8s-node01、k8s-node02、k8s-node03。

**（1）、修改主机名**

使用vi/vim编辑/etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost localhost.

localdomain localhost6 localhost6.localdomain6 192.168.11.254 k8s-master-lb

192.168.251.243   k8s-master-lb

192.168.251.34    k8s-master01

192.168.251.36    k8s-master02

192.168.251.247   k8s-master03

192.168.251.34    k8s-node01

192.168.251.36    k8s-node02

192.168.251.247   k8s-node03

192.168.251.175   k8s-node04   #新增节点 需要扩容 k8s-node05 k8s-node06 类推追加

192.168.251.176   registry.edoc2.com

192.168.251.243   edoc2v5.rancher.pre.com

​

**（2）、设置node04免密登录并发分发同步 /etc/hosts（k8s-master01节点、复用即k8s-node01节点操作）：**

ssh-copy-id root@k8s-node04 # （此步在node01 node02 node03节点上已经一键安装做完免密）系统会提示输入密码，此时输入目标服务器的密码。

scp -rp /etc/hosts root@k8s-node01:/etc/hosts

scp -rp /etc/hosts root@k8s-node02:/etc/hosts

scp -rp /etc/hosts root@k8s-node03:/etc/hosts

scp -rp /etc/hosts root@k8s-node04:/etc/hosts

**（3）、初始化04节点操作系统及安装04节点docker（颜色区分整条命令作用）**

**# k8s安装包路径（k8s-master01节点、复用即k8s-node01节点操作）：**

cd kylinux10-cluster-k8s-1.23-middleware-v6.6

scp -rp /etc/sysctl.conf root@k8s-node04:/etc/

scp -rp /etc/profile.d/base.sh root@k8s-node04:/etc/profile.d/

scp -rp /etc/security/limits.conf root@k8s-node04:/etc/security/

scp -rp /etc/systemd/system/docker.service root@k8s-node04:/opt/

scp -rp /etc/docker/daemon.json root@k8s-node04:/opt/

scp -r ./soft/k8s1.23/docker-20.10.8.tgz root@k8s-node04:/opt/

**（4）、k8s-node04扩容节点操作以下```内```整个片段命令（颜色标记为整行执行标记）：**

```

cat /etc/profile | grep ulimit

if grep -q "ulimit" /etc/profile; then echo "ulimit have been exist"; else echo "ulimit not exist"; echo "ulimit -HSn 655350" >> /etc/profile; fi

```

cat /etc/resolv.conf # 查看用户环境 nameserver地址是否存在且不能为127.0.0.53（此地址不可用，会发生dns解析环回故障）如果异常执行下一条命令，正确跳过下一条命令

echo "nameserver 114.114.114.114" >> /etc/resolv.conf

systemctl disable --now firewalld

systemctl stop firewalld.service

setenforce 0

sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/sysconfig/selinux

sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config

swapoff -a && sysctl -w vm.swappiness=0

sed -ri '/^[^#]\*swap/s@^@#@' /etc/fstab

sysctl -p

source /etc/profile

‍

**# 登录扩容节点k8s-node04，部署k8s扩容节点系统优化操作:**

**执行如下命令（逐条执行）：**

```bash

modprobe overlay

modprobe br\_netfilter

echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables

echo '1' > /proc/sys/net/bridge/bridge-nf-call-ip6tables

echo '1' > /proc/sys/net/ipv4/ip\_forward

cd /opt/docker-20.10.8.tgz

\cp docker/\* /usr/bin/

sysctl -p

```

**# 登录 k8s扩容节点操作，登录k8s-node04节点操作；**

执行如下命令（逐条执行）：

```bash

yum remove -y podman

mkdir /etc/docker/

\cp /opt/docker.service /etc/systemd/system/

\cp /opt/daemon.json /etc/docker/

chmod +x /etc/systemd/system/docker.service

systemctl daemon-reload

systemctl enable docker.service

systemctl restart docker

systemctl status docker | grep "" # 查看docker启动运行状况

```

**（5）、拷贝k8s组件，在主节点k8s-node01上执行（k8s-master01节点、复用即k8s-node01节点）**

```bash

ssh root@k8s-node04 "mkdir -p /etc/kubernetes/pki"

cd /etc/kubernetes/

scp -rp pki/ca.pem root@k8s-node04:/etc/kubernetes/pki

scp -rp pki/ca-key.pem root@k8s-node04:/etc/kubernetes/pki

scp -rp pki/front-proxy-ca.pem root@k8s-node04:/etc/kubernetes/pki

scp -rp bootstrap-kubelet.kubeconfig root@k8s-node04:/etc/kubernetes/

scp -rp /etc/docker/daemon.json root@k8s-node04:/etc/docker/

ssh root@k8s-node04 "ls /etc/kubernetes/pki"

```

**（6）、同步k8s node二进制组件kubelet、kube-proxy、cni（k8s-master01节点、复用即k8s-node01节点）：**

```bash

# 同步service文件（用于systemctl启动）

# systemctl启动k8s组件并加入启动项

scp -rp /usr/local/bin/{kube-proxy,kubelet} root@k8s-node04:/usr/local/bin/

ssh root@k8s-node04 "mkdir -p /etc/cni/net.d /opt/cni/bin && mkdir -p /var/lib/kubelet /var/log/kubernetes /lib/systemd/system/kubelet.service.d /etc/kubernetes/manifests && docker login -u admin -p edoc2edoc2 registry.edoc2.com:5000 && docker pull registry.edoc2.com:5000/library/pause:3.6"

```

添加kubelet

```bash

cd /home/kylinux10-cluster-k8s-1.23-middleware-v6.6

scp -rp /opt/cni/bin/\* root@k8s-node04:/opt/cni/bin/

scp -rp scripts/k8s\_conf\_file/kubelet.service root@k8s-node04:/lib/systemd/system/kubelet.service

scp -rp scripts/k8s\_conf\_file/10-kubelet.conf root@k8s-node04:/lib/systemd/system/kubelet.service.d/10-kubelet.conf

scp -rp scripts/k8s\_conf\_file/kubelet-conf.yml root@k8s-node04:/etc/kubernetes/kubelet-conf.yml

```

添加kube-proxy

```sh

cd /home/kylinux10-cluster-k8s-1.23-middleware-v6.6

scp -rp /etc/kubernetes/kube-proxy.kubeconfig root@k8s-node04:/etc/kubernetes/kube-proxy.kubeconfig

scp -rp scripts/k8s\_conf\_file/kube-proxy.yaml root@k8s-node04:/etc/kubernetes/kube-proxy.yaml

scp -rp scripts/k8s\_conf\_file/kube-proxy.service root@k8s-node04:/lib/systemd/system/kube-proxy.service

```

**（7）、 登录k8s-node04扩容节点操作，启动kube-proxy：**

```sh

systemctl daemon-reload

systemctl start kube-proxy && systemctl enable kube-proxy

```

**（8）、 登录k8s-node04扩容节点操作，启动kubelet：**

```bash

rm -rf /var/lib/kubelet/pki/kubelet\*

systemctl daemon-reload && systemctl start kubelet && systemctl enable kubelet

```

‍

**（9）、 时间同步：**

登录k8s-node04扩容节点服务器，进行如下操作(kylinux标准安装自带chronyd)；

```bash

date

timedatectl show

timedatectl set-timezone Asia/Shanghai

service chronyd restart

systemctl enable chronyd.service

systemctl status chronyd.service | grep " "

chronyc makestep

chronyc sources

chronyc tracking

chronyc sourcestats

chronyc ntpdata

```

执行如下命令：

REGISTRY 为 k8s-node01 部署k8s集群核心服务器ip地址

```bash

REGISTRY=xxx.xxx.xxx.xxx # ip地址

```

执行如下命令（以下一段复制粘贴执行）：

```bash

cat << EOF > /etc/chrony.conf

server ${REGISTRY} iburst

rtcsync

makestep 30 3

EOF

```

执行如下命令（逐执行）：

```bash

systemctl restart chronyd

systemctl enable chronyd

systemctl status chronyd | grep " " # 所有status需要加处理

sleep 3

chronyc makestep

chronyc sources

chronyc tracking

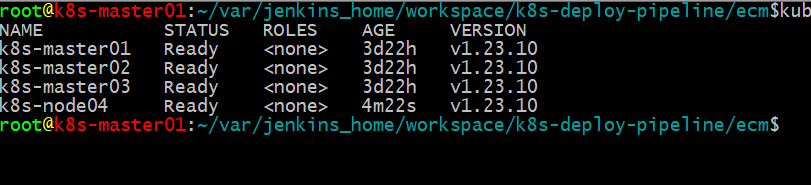
sed -i.backup 's/defaults/defaults,noatime/g' /etc/fstab

ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

echo 'Asia/Shanghai' >/etc/timezone

#### **3、查看k8s节点是否添加成功：**

kubectl get nodes



#### **4、查看新添加节点calico状态：**

kubectl -n kube-system get pods -o wide

​

