## kubeadm 快速搭建k8s集群

### 第1节 环境设置

### 1.1 固定虚拟机系统IP

参考: <https://www.cnblogs.com/lfhappy/p/10798400.html>

### 1.2 配置hosts解析各主机

vi /etc/hosts：

**192.168.116.129 master**

**192.168.116.130 node1**

**192.168.116.131 node2**

### **1.3同步各主机时间**

### yum install -y ntpdate

### ntpdate time.windows.com

**1.4关闭 SWAP，关闭selinux**

**1：**swapoff -a

**2：**vim /etc/selinux/config

--------> SELINUX=disabled

改完以后执行 setenforce 0

**1.5 更换yum源**

**1) 备份源yum源**

cd /etc/yum.repos.d/

cp CentOS-Base.repo CentOS-Base-repo.bak

1. 使用wget下载163yum源repo文件

wget <http://mirrors.163.com/.help/CentOS7-Base-163.repo>

1. 清理旧包

yum clean all

1. 把下载下来163repo文件设置成为默认源

mv CentOS7-Base-163.repo CentOS-Base.repo

1. 生成163yum源缓存并更新yum源

yum makecache

yum update

**内核升级-----CentOS7**

#下载内核rpm包

rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm

#指定包名安装新版

kernelyum --enablerepo=elrepo-kernel install -y kernel-lt

#设置默认启动的内核

grub2-set-default 'CentOS Linux (4.4.218-1.el7.elrepo.x86\_64) 7 (Core)'

#重启生效

systemctl reboot

#重新查看内核版本

[root@Centos8 ~]# uname -r

**设置kube-proxy开启ipvs的前置条件**

modprobe br\_netfilter

cat > /etc/sysconfig/modules/ipvs.modules <<EOF

#!/bin/bash

modprobe -- ip\_vs

modprobe -- ip\_vs\_rr

modprobe -- ip\_vs\_wrr

modprobe -- ip\_vs\_sh

modprobe -- nf\_conntrack

EOF

chmod 755 /etc/sysconfig/modules/ipvs.modules && bash /etc/sysconfig/modules/ipvs.modules

**# 查看是否加载成功ipvs模块**

lsmod | grep -e ip\_vs -e nf\_conntrack\_ipv4

**1.6 修改内核参数 ？？？？**

# 编辑 /etc/sysctl.d/k8s.conf

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

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

**1.7 关闭防火墙**

systemctl disable firewalld && systemctl stop firewalld

**1.8 docker安装**

**1.** curl -fsSL https://get.docker.com/ | sh（最新版）

### **修改docker cgroup driver为systemd**

创建或修改/etc/docker/daemon.json

**{**

**"registry-mirrors": ["http://f1361db2.m.daocloud.io"],#使用国内镜像**

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

**}**

**重启docker: systemctl restart docker**

2：设置docker自动启动

systemctl enable docker && systemctl start docker

1. **kubeadm安装**

1：添加 repo

**1）vi /etc/yum.repos.d/kubenetes.repo**

**[kubernetes]**

**name=Kubernetes Repo**

**baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_4/**

**gpgcheck=1 #开启gpg校验**

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

**enabled=1**

**2）: 下载校验文件**

**wget https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg**

**wget** <https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg>

**3）：导入校验文件**

**rpm --import rpm-package-key.gpg**

**rpm --import yum-key.gpg**

**2 开始安装kubeadm**

**1.查看 kubeadm目前支持的版本**

yum list kubelet kubeadm kubectl --showduplicates|sort -r

**2.安装指定版本 kubelet kubeadm kubectl**

yum -y install kubeadm-1.15.1 kubectl-1.15.1 kubelet-1.15.1

**问题**

错误：软件包：kubelet-1.8.9-0.x86\_64 (kubernetes)

需要：kubernetes-cni = 0.5.1

**解决：**yum install kubernetes-cni = **0.5.1**

****3.查看安装的版本****

**kubeadm version**

**kubectl version --client**

**kubelet --version**

1. ****启动kubelet(所有节点)****

systemctl enable kubelet && systemctl start kubelet

1. **拉取初始化镜像**

**1: 构建拉取镜像脚本**

**v**i initimage.sh

**2： 脚本内容**

#!/usr/bin/env bash

K8S\_VERSION=v1.8.9

ETCD\_VERSION=3.3.10

DASHBOARD\_VERSION=v1.8.3

FLANNEL\_VERSION=v0.10.0-amd64

DNS\_VERSION=1.3.1

PAUSE\_VERSION=3.1

# 基本组件

docker pull mirrorgooglecontainers/kube-apiserver-amd64:$K8S\_VERSION

docker pull mirrorgooglecontainers/kube-controller-manager-amd64:$K8S\_VERSION

docker pull mirrorgooglecontainers/kube-scheduler-amd64:$K8S\_VERSION

docker pull mirrorgooglecontainers/kube-proxy-amd64:$K8S\_VERSION

docker pull mirrorgooglecontainers/etcd-amd64:$ETCD\_VERSION

docker pull mirrorgooglecontainers/pause:$PAUSE\_VERSION

docker pull coredns/coredns:$DNS\_VERSION

# 修改tag

docker tag mirrorgooglecontainers/kube-apiserver-amd64:$K8S\_VERSION k8s.gcr.io/kube-apiserver:$K8S\_VERSION

docker tag mirrorgooglecontainers/kube-controller-manager-amd64:$K8S\_VERSION k8s.gcr.io/kube-controller-manager:$K8S\_VERSION

docker tag mirrorgooglecontainers/kube-scheduler-amd64:$K8S\_VERSION k8s.gcr.io/kube-scheduler:$K8S\_VERSION

docker tag mirrorgooglecontainers/kube-proxy-amd64:$K8S\_VERSION k8s.gcr.io/kube-proxy:$K8S\_VERSION

docker tag mirrorgooglecontainers/etcd-amd64:$ETCD\_VERSION k8s.gcr.io/etcd:$ETCD\_VERSION

docker tag mirrorgooglecontainers/pause:$PAUSE\_VERSION k8s.gcr.io/pause:$PAUSE\_VERSION

docker tag coredns/coredns:$DNS\_VERSION k8s.gcr.io/coredns:$DNS\_VERSION

#删除冗余的images

docker rmi mirrorgooglecontainers/kube-apiserver-amd64:$K8S\_VERSION

docker rmi mirrorgooglecontainers/kube-controller-manager-amd64:$K8S\_VERSION

docker rmi mirrorgooglecontainers/kube-scheduler-amd64:$K8S\_VERSION

docker rmi mirrorgooglecontainers/kube-proxy-amd64:$K8S\_VERSION

docker rmi mirrorgooglecontainers/etcd-amd64:$ETCD\_VERSION

docker rmi mirrorgooglecontainers/pause:$PAUSE\_VERSION

docker rmi coredns/coredns:$DNS\_VERSION

...chmod +x initimage.sh

**3.执行此脚本，开始导入镜像...**

./initimage.sh

1. 查看存在的镜像

[root@Centos8 ]# docker images

**初始化主节点**

kubeadm init --kubernetes-version=v1.15.1 --image-repository registry.aliyuncs.com/google\_containers --pod-network-cidr=192.168.116.0/16 --ignore-preflight-errors=...

成功后提示：

mkdir -p $HOME/.kube

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

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

**添加flannel网络**

mkdir -p install-k8s/plugin/flannel

mkdir -p install-k8s/core

cd install-k8s/core

mv /etc/kubernetes/kubeadm-init.log /etc/kubernetes/kubeadm-config.yaml ./

cd ../plugin/flannel

**#下载flannel.yml**

/etc/hosts 添加 199.232.68.133 raw.githubusercontent.com

wget <https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

**#执行yaml文件**

kubectl create -f kube-flannel.yml

**#构建完成，查看当前名称空间为kube-system的pod状态，-n 指定名称空间**

[root@Centos8 core]# kubectl get pod -n kube-system

**#构建完成后，网卡界面会显示flannel信息**

[root@Centos8 core]# ifconfig

**#网络加载成功，状态变为Ready**

[root@Centos8 ~]# kubectl get node

**配置从节点与主节点关联**

**从节点加入主节点：**

kubeadm join 192.168.116.129:6443 --token 74dm27.jyvm497iwaqg5s5c --discovery-token-ca-cert-hash sha256:f5d21085f7ad175b9495c3cdfde0918e8ac0a8cbb00b3b2bc49b7a9684aed9db (该指令在初始化主节点的时候会在控制台回显)

**可能存在问题:Initial timeout of 40s passed. error execution phase kubelet-start: error uploading crisocket: timed out waiting for the condition**

**解决办法:**

$ swapoff -a

$ kubeadm reset

$ systemctl daemon-reload

$ systemctl restart kubelet

$ iptables -F && iptables -t nat -F && iptables -t mangle -F && iptables -X

**Master上的etcd、kube-apiserver、kubecontroller-manager、kube-scheduler服务**

**安装MASTER及各节点 etcd服务(TMD坑爹啊！！！以后能用yum安装的东西,绝不自己手动安装！！！)**

1.**各个节点分别执行指令** yum install etcd

2**.准备目录**

mkdir -p /work/etcd #生成工作目录

chown etcd:etcd /work/etcd #添加权限

1. **备份自动安装产生的配置文件**

cp /etc/etcd/etcd.conf /etc/etcd/etcd.conf.template

1. **编辑配置文件**

vim /etc/etcd/etcd.conf

**5 生成TOKEN集群配置需要**

echo k8s-cluster|md5sum

m**aster节点配置：**

#[Member]

ETCD\_DATA\_DIR="/work/etcd/"

ETCD\_LISTEN\_PEER\_URLS="http://192.168.116.129:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://127.0.0.1:2379"

ETCD\_NAME="master"

#[Clustering]

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.116.129:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.116.129:2379,http://192.168.116.130:2379,http://192.168.116.131:2379"

ETCD\_INITIAL\_CLUSTER="master=http://192.168.116.129:2380,node1=http://192.168.116.130:2380,node2=http://192.168.116.131:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="ea8cfe2bfe85b7e6c66fe190f9225838"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

**node1 节点etcd.conf配置**

#[Member]

ETCD\_LISTEN\_PEER\_URLS="http://192.168.116.130:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://127.0.0.1:2379"

ETCD\_NAME="node1"

#[Clustering]

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.116.130:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.116.130:2379,http://192.168.116.131:2379,http://192.168.116.129:2379"

ETCD\_INITIAL\_CLUSTER="master=http://192.168.116.129:2380,node1=http://192.168.116.130:2380,node2=http://192.168.116.131:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="ea8cfe2bfe85b7e6c66fe190f9225838"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

n**ode2 节点配置文件设置**

#[Member]

ETCD\_DATA\_DIR="/work/etcd"

ETCD\_LISTEN\_PEER\_URLS="http://192.168.116.131:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://127.0.0.1:2379"

ETCD\_NAME="node2"

#[Clustering]

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.116.131:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.116.131:2379,http://192.168.116.130:2379,http://192.168.116.129:2379"

ETCD\_INITIAL\_CLUSTER="master=http://192.168.116.129:2380,node1=http://192.168.116.130:2380,node2=http://192.168.116.131:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="ea8cfe2bfe85b7e6c66fe190f9225838"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

**6.启动etcd服务**

systemctl daemon-reload  
 systemctl enable etcd.service  
 systemctl start etcd.service

**7 设置环境变量：export ETCDCTL\_API=3 (etcdctl api设置为v3版本,此时下边指令可用)**

etcdctl endpoint health

======安装不成功：版本过高问题较多，卸载kubelet,kubectl,kubeadm ==============

remove  -y kubelet kubeadm kubectl

**安装指定版本**

yum -y install kubeadm-1.15.1 kubectl-1.15.1 kubelet-1.15.1.