

kubernetes 集群架构

Kubernetes 集群组件：

etcd 一个高可用的K/V键值对存储和服务发现系统

flannel 实现夸主机的容器网络的通信

kube-apiserver 提供kubernetes集群的API调用

kube-controller-manager 确保集群服务

kube-scheduler 调度容器，分配到Node

kubelet 在Node节点上按照配置文件中定义的容器规格启动容器

kube-proxy 提供网络代理服务

**1.环境准备**

**1.1.服务器环境**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节点 |  | ip 地址 | 操作系统 | CPU | 内存 |
| master |  | **~~192.168.100.246~~** | CentOS 7.x-x86\_64 | >= 4c | >= 4G |
| node1 |  | **~~192.168.100.247~~** | CentOS 7.x-x86\_64 | >= 8c | >= 8G |
| node2 |  | **~~192.168.100.248~~** | CentOS 7.x-x86\_64 | >= 8c | >= 8G |
| harbor |  | **~~192.168.100.241~~** | CentOS 7.x-x86\_64 | >= 8c | >= 8G |

注意：

1. 需要在所有看s集群节点上执行本文档的初始化命令；

2. 需要使用具有 root 权限的账号执行这些命令。

修改主机名

hostnamectl set-hostname master # 将 master 替换为当前主机名

如果 DNS 不支持解析主机名称，则需要修改每台机器的 /etc/hosts 文件，添加主机名和 IP 的对应关系：

cat >> /etc/hosts <<EOF
192.168.100.246 master
192.168.100.247 node1
192.168.100.248 node2
EOF

检查当前 CentOS系统内核版本

uname -sr

**内核最好是大于4.19 （升级内核需要谨慎操作）**

升级内核

安装elrepo

yum update -y
# 导入公钥
rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
# 安装7.x版本的ELRepo
rpm -Uvh https://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm
# 载入elrepo-kernel元数据
yum --disablerepo=\\* --enablerepo=elrepo-kernel repolist

查看可用的包

yum --disablerepo=\\* --enablerepo=elrepo-kernel list kernel\*

安装最新版本的kernel

yum --disablerepo=\\* --enablerepo=elrepo-kernel install kernel-ml.x86\_64 -y

删除旧版本工具包

yum remove kernel-tools-libs.x86\_64 kernel-tools.x86\_64 -y

安装新的工具包

yum --disablerepo=\\* --enablerepo=elrepo-kernel install kernel-ml-tools.x86\_64 -y

修改内核顺序

grub2-set-default 0 #取第一个
grub2-editenv list

重启并检查

reboot
uname -r

安装依赖包

yum install -y epel-release
yum install -y conntrack ntpdate ntp ipvsadm ipset jq iptables curl sysstat libseccomp wget

关闭防火墙

systemctl stop firewalld.service
#禁止开启启动
systemctl disable firewalld.service

**如果是生产环境防火墙开启，开放如下端口**

控制节点（master）

工作节点（work）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议 | 方向 | 端口范围 | 作用 | 使用者 |
| TCP | 入站 | 10250 | Kubelet API | kubelet 自身、控制平面组件 |
| TCP | 入站 | 30000-32767 (此处可以根据 nodeport模式使用的节点开放,如下文中使用了80和443端口作为ingress入口,所以开放这两个端口) | NodePort 服务\*\* | 所有组件 |

关闭 selinux

临时关闭

setenforce 0

永久关闭,可以修改配置文件/etc/selinux/config,将其中SELINUX设置为disabled

vi /etc/selinux/config
...
SELINUX=disabled
...

查看状态

sestatus
# 如果没效果，就重启
# reboot

关闭swap

cat /proc/vmstat | egrep "dirty|writeback"
echo "vm.swappiness = 0">> /etc/sysctl.conf
swapoff -a
free -g
sysctl -p
free -g

**1.2.iptables 放行**

yum install iptables-services -y
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -F
iptables -L -n
service iptables save
systemctl enable iptables.service

**1.3.开启ipvs**

**加载内核模快**

lsmod|grep ip\_vs
nmodprobe -- ip\_vs
modprobe -- ip\_vs\_rr
modprobe -- ip\_vs\_wrr
modprobe -- ip\_vs\_sh
#如果内核版本小于等于4.19，nf\_conntrack替换为nf\_conntrack\_ipv4
modprobe -- nf\_conntrack

**详见** [**torvalds/linux@a0ae256**](https://github.com/torvalds/linux/commit/a0ae2562c6c4b2721d9fddba63b7286c13517d9f)

**1.4.配置yum源**

配置阿里软件源（base和epel）：

mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo\_bak
wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
mv /etc/yum.repos.d/epel.repo /etc/yum.repos.d/epel.repo.backup
mv /etc/yum.repos.d/epel-testing.repo /etc/yum.repos.d/epel-testing.repo.backup
wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

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

执行清理和更新：

yum clean all
yum makecache
yum update -y #第一次需要比较久的时间
yum repolist # 查看正在使用yum源

**2.安装docker**

查看最新的docker版本

yum list docker-ce.x86\_64 --showduplicates | sort -r
yum list docker-ce-cli.x86\_64  --showduplicates |sort -r

安装指定版本的docker

yum install -y docker-ce-19.03.9-3.el7 docker-ce-cli-19.03.9-3.el7
systemctl enable docker #开机启动
systemctl start docker #启动docker
docker --version

修改 /etc/docker/daemon.json

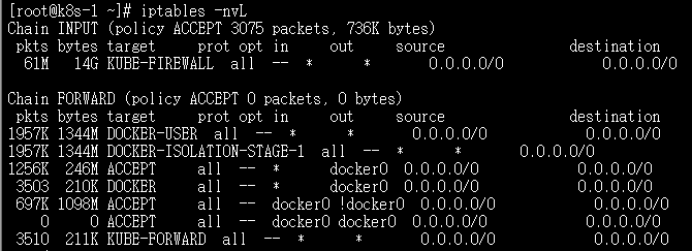
cat > /etc/docker/daemon.json <<EOF
{
   "registry-mirrors": [
      "https://registry.docker-cn.com",
      "https://hub-mirror.c.163.com",
      "https://docker.mirrors.ustc.edu.cn"
   ],
   "exec-opts": ["native.cgroupdriver=systemd"],
   "log-driver": "json-file",
   "log-opts": {
      "max-size": "100m",
      "max-file": "5"
   }
}
EOF

重启docker

systemctl daemon-reload
systemctl restart docker

确认一下iptables filter表中FOWARD链的默认策略(pllicy)为ACCEPT

iptables -nvL



**3.kubeadm安装k8s**

**2.1.安装kubeadm、kubelete、kubectl**

需要在每台机器上都安装以下的软件包：

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。

查看可用版本

yum list --showduplicates | grep 'kubeadm\|kubectl\|kubelet'

安装 kubeadm，kubelet，kubectl1

yum install -y kubelet-1.20.0 kubeadm-1.20.0 kubectl-1.20.0 --disableexcludes=kubernetes
systemctl enable --now kubelet # 此时还不能启动 kubelet，先设置开机启动

**2.2.安装主节点**

**一些 RHEL/CentOS 7 的用户曾经遇到过问题：由于 iptables 被绕过而导致流量无法正确路由的问题。您应该确保 在 sysctl 配置中的 net.bridge.bridge-nf-call-iptables 被设置为 1**

cat <<EOF > /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sysctl --system

配置住主节点

下载该文件直接进行修改

需要修改的地方：

advertiseAddress: 192.168.18.22 #把IP地址改位当前主节点的ip地址

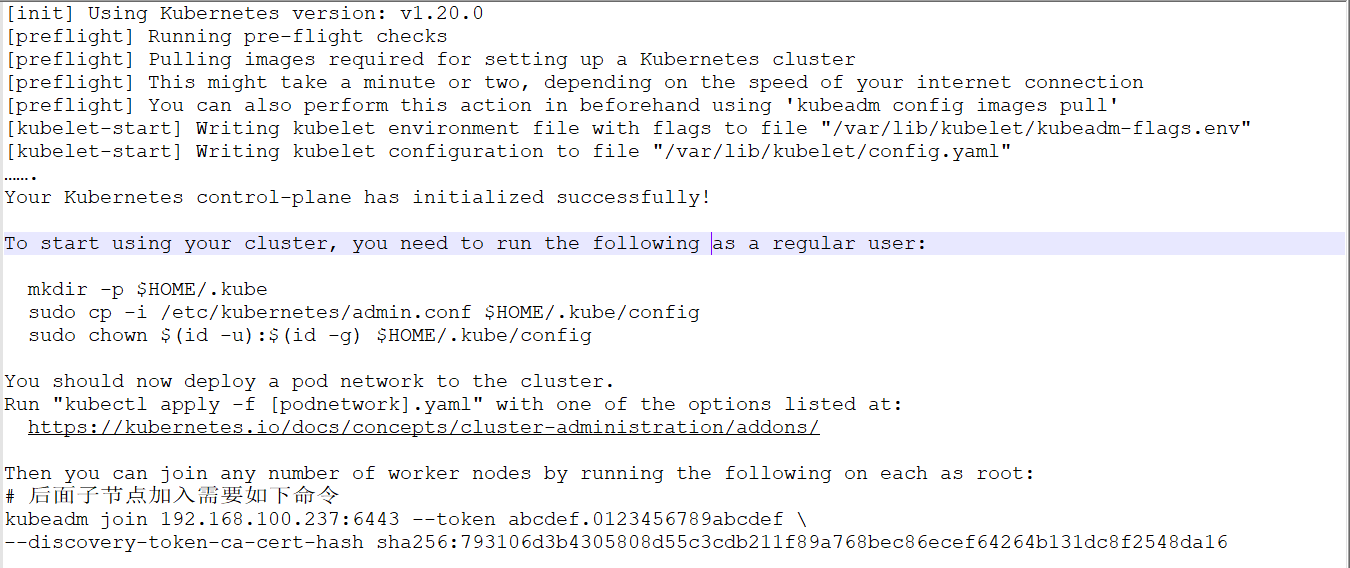
查看和拉取镜像：

kubeadm config images list --config kubeadm.yml
kubeadm config images pull --config kubeadm.yml

初始化主节点

kubeadm init --config=kubeadm.yml --upload-certs | tee kubeadm-init.log

第一次执行需要等待几分钟时间，不要关闭窗口。执行完后窗口最下面会输出 join 命令



请备份好 kubeadm init 输出中的 kubeadm join 命令，因为您会需要这个命令来给集群添加节点。

配置 kubectl

mkdir -p $HOME/.kube && sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config && sudo chown $(id -u):$(id -g) $HOME/.kube/config

验证是否成功

kubectl get node

kubectl自动补全(推荐)

yum install bash-completion -y
echo "source /usr/share/bash-completion/bash\_completion" >> ~/.bashrc
echo 'source <(kubectl completion bash)' >>~/.bashrc
source ~/.bashrc

**2.3.安装网络组件（如果是单节点可以初始化master后安装网络组件，如果是集群模式可以在集群节点都加入后再初始化网络组件）**

下载文件上次到master节点服务器，执行

kubectl apply -f kube-flannel.yml

或者服务器能连接外网，直接执行下面命令（二选一即可）

kubectl apply -f https://raw.githubusercontent.com/flannel-io/flannel/v0.13.1-rc2/Documentation/kube-flannel.yml

**2.5.工作节点**

**2.5.1.单节点（单宿主机，宿主机可以是KVM虚拟机）**

执行以下命令,允许控制节点也部署应用，单节点不要管2.5.2小节的内容

kubectl taint nodes --all node-role.kubernetes.io/master-

**2.5.2.加入其他的节点**

使用上面记录的 join命令把工作节点接入集群

kubeadm join 192.168.18.22:6443 --token abcdef.0123456789abcdef \
    --discovery-token-ca-cert-hash sha256:793106d3b4305808d55c3cdb211f89a768bec86ecef64264b131dc8f2548da16

回到 master 服务器验证是否成功，确认个组件都处于healthy状态。

kubectl get cs,nodes

**2.6.安装ingress-controller**

使用nginx-ingress

文档地址 <https://docs.nginx.com/nginx-ingress-controller/installation/installation-with-manifests/>

**2.6.1.下载镜像文件**

由于谷歌镜像仓库国内无法访问,所以使用docker-hub上的镜像

docker pull acicn/ingress-nginx-controller:v0.41.2
docker tag acicn/ingress-nginx-controller:v0.41.2 k8s.gcr.io/ingress-nginx/controller:v0.41.2

**2.6.2.开始安装**

**将上面的文件拷贝到服务器 根目录下k8s\_config文件夹，按顺序执行**

kubectl apply -f ~/k8s\_config/k8s\_static\_nginx-controller-deploy.yaml
kubectl apply -f ~/k8s\_config/k8s\_static\_nginx-controller-ingressclass.yaml

**2.6.3.检验**

kubectl get all -n ingress-nginx

应该有 pod/ingress-nginx-controller-xxxx和service/ingress-nginx-controller 查看Pod日志是否启动成功

kubectl -n ingress-nginx logs -f ingress-nginx-controller- 【tab补全】

**2.7.安装metrics-server**

**2.7.1.镜像准备**

将上面文件拷贝到k8s的集群的每一个服务器中（宿主机）

将镜像载入服务器本地仓库，在每一个k8s的宿主机上执行

docker load -i k8s\_images\_metrics-server.tar

**2.7.2.安装**

**将文件拷贝到服务器根目录下k8s\_config文件夹**

kubectl apply -f ~/k8s\_config/k8s\_static\_metrics-server-components.yaml

**2.8.安装dashboard**

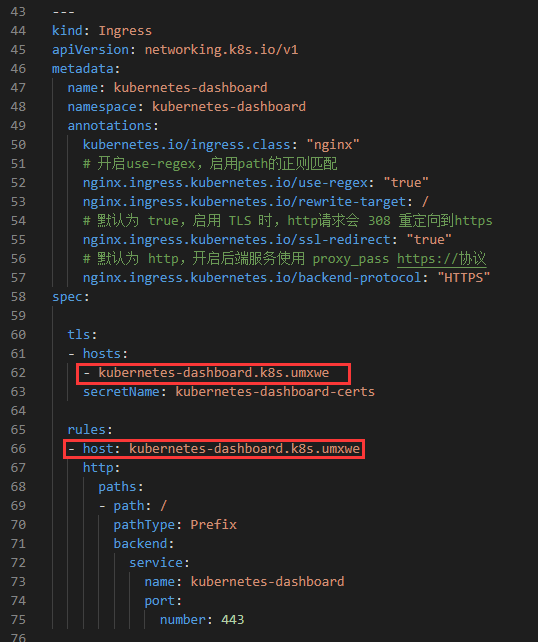
**将上面的文件拷贝到服务器根目录下k8s\_config文件夹**

kubectl apply -f ~/k8s\_config/k8s\_static\_dashboard-recommended.yaml

访问<http://kubernetes-dashboard.k8s.umxwe>

**上面的地址是**k8s\_static\_dashboard-recommended.yaml**中Ingress中host的值** **host的值必须是能够访问k8s集群的合法域名**

**如果需要改域名，修改下图中位置，两个必须一致，tls是HTTPS，rules里面是真实规则**



**ingress详细配置见官方文档**[**https://kubernetes.io/zh/docs/concepts/services-networking/ingress/**](https://kubernetes.io/zh/docs/concepts/services-networking/ingress/) **获取token**

kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep admin-user | awk '{print $1}')