

Kubernetes 集群部署流程

- `kubeadm`部署集群
- `kubecttl`命令行工具进行管理
- `kubelet`作为后台进程存在

环境说明

- 虚拟机: VMware® Workstation Pro 16
- 操作系统: CentOS Linux release 8.2.2004
- 操作用户: root用户(生产环境根据实际情况来)

资源分配

节点准备

主机节点名称	IP	CPU核心数	内存大小	磁盘大小
k8s-master	192.168.3.13	2	4G	100G
k8s-node1	192.168.3.14	2	8G	150G
k8s-node2	192.168.3.15	2	8G	150G

操作步骤

下面 1-8 步骤所有节点都要执行, 9-10 在master节点执行, 11 在node节点执行

如果 9-11 有报错, 先根据报错信息尝试解决, 无法解决时, 可以`kubeadm reset --force`重置集群重新配置

1.关闭swap

```
# 临时关闭
swapoff -a

# 永久关闭, 注释掉swap分区即可
vim /etc/fstab
#UUID=65c9f92d-4828-4d46-bf19-fb78a38d2fd1 swap
defaults      0 0
```

2.关闭SELinux

```
# 永久生效, 需要重启
vi /etc/selinux/config
selinux=disable
# 临时生效
setenforce 0
```

3. 关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
systemctl status firewalld
```

4. 添加主机信息

```
vi /etc/hosts
192.168.3.13 k8s-master
192.168.3.14 k8s-node1
192.168.3.15 k8s-node2
```

5. 配置流量

将流量传递到iptables链

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

6. 部署Docker

```
# 卸载已存在docker
yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine

# 设置docker仓库
yum install -y yum-utils
yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo

# 安装docker-ce
yum install docker-ce docker-ce-cli containerd.io

# 加入开机启动并启动
systemctl enable docker
systemctl start docker
```

```
# 测试运行并查看版本信息
docker run hello-world
docker version
# 配置阿里docker镜像加速器
mkdir -p /etc/docker
vi /etc/docker/daemon.json
# {your_id} 根据自己申请的阿里镜像加速器id来配置
{
  "registry-mirror": ["https://{your_id}.mirror.aliyuncs.com"]
}
# docker文件驱动改成 systemd
vim /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "registry-mirror": ["https://{your_id}.mirror.aliyuncs.com"]
}
# 重启docker
systemctl restart docker
# 如果启动失败,强制加载再启动试试
systemctl reset-failed docker
systemctl restart docker
# 查看docker配置信息
docker info
docker info | grep Driver
```

7. 配置kubernetes镜像源

```
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
```

8. 安装kubernetes

```
yum install -y kubelet kubeadm kubectl
# 如果gpg检查失败,可以跳过gpg检查进行安装
yum install -y --nogpgcheck kubelet kubeadm kubectl
# 开机启动
systemctl enable kubelet
# 先不启动kubelet,因为会启动失败,提示某文件不存在,需要先执行kubeadm init完成初始化
```

9. 部署k8s-master

在master节点上执行

```
# 初始化集群
kubeadm init --kubernetes-version=1.19.3 \
--apiserver-advertise-address=192.168.3.13 \
--image-repository registry.aliyuncs.com/google_containers \
--service-cidr=192.168.0.0/16 \
--pod-network-cidr=192.168.0.0/16
# 记录好输出信息Your Kubernetes control-plane has initialized successfully!后十几行

# 按照提示执行下面步骤
mkdir -p $HOME/.kube
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
chown $(id -u):$(id -g) $HOME/.kube/config

# 查看docker镜像,可以看到kube..和etcd等镜像
docker images

# kube-apiserver默认只启动安全访问接口6443, 而不启动非安全访问接口8080, kubectl是通过
8080端口访问k8s kubelet的, 所以要修改配置文件, 使其支持8080端口访问
vim /etc/kubernetes/manifests/kube-apiserver.yaml
-insecure-port=8080

# 启动kubelet
systemctl start kubelet

# 查看集群状态信息
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
k8s-master	NotReady	master	28m	v1.19.3

10. 配置calico网络

在master节点上执行

网络方案也可以选择其他(例如: flannel等)

```
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
# 查看calico网络状态, STATUS为Running后查看集群信息
kubectl get pods -n kube-system
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
k8s-master	Ready	master	45m	v1.19.3

11. node节点加入集群

在node节点上执行

```
# 根据kubeadm init最后提示
kubeadm join 192.168.3.13:6443 --token vmbhd0.e4cszyn6ozqv9tet \
  --discovery-token-ca-cert-hash
sha256:c8fd072fd90c9ddbae3b3945ba0120abf5ed6777de3354a6eb4814a93ed1b844
```

测试集群

master节点执行

```
# 查看集群状态
kubectl get nodes
NAME          STATUS    ROLES    AGE    VERSION
k8s-master    Ready     master   52m    v1.19.3
k8s-nnode1    Ready     <none>    2m29s  v1.19.3
k8s-nnode2    Ready     <none>    2m3s   v1.19.3

# namespace
kubectl get namespace
kubectl create namespace test
kubectl get namespace
kubectl delete namespace test

# 创建nginx实例并开放端口
kubectl create deployment nginx --image=nginx
kubectl expose deployment nginx --port=80 --type=NodePort
kubectl get pod,svc
NAME          READY    STATUS    RESTARTS   AGE
pod/nginx-f89759699-9265g  1/1      Running   0           5m30s

NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/kubernetes  ClusterIP     x.x.x.x       <none>         443/TCP          12m
service/nginx      NodePort      x.x.x.x       <none>         80:30086/TCP     1m15s

# 在浏览器输入http://IP:30086/ 访问nginx

# 增加node节点的节点role名称
kubectl label nodes k8s-node1 node-role.kubernetes.io/node=
# 删除node节点的节点role名称
kubectl label nodes k8s-node1 node-role.kubernetes.io/node-
```