CubeUniverse文档-使用kubeadm安装集群

> kubeadm是一种方便快速在非单机部署集群的工具，是部署集群的常用选择。
>
> 本文档将以CentOS7.9为基础环境，介绍使用kubeadm部署可以运行CubeUniverse的k8s集群的步骤。
>
> 注意：要运行k8s，节点机器必须**开启cpu虚拟化**，cpu不少于**2核**，内存不少于**2G**。安装之前先**关闭SELinux**和swap。建议同时**关闭linux防火墙**。
>
> 注意，CubeUniverse非必须使用本文的安装方法安装集群环境。其他集群环境（版本不宜太低）也可以正常安装。为运行CubeUniverse并保证高可用特性，应当至少准备一个三主三从的集群。

1. 更换yum源

在国内需要修改yum才能正常下载。

先切换到 `root` 用户再执行下列操作。在所有节点都要执行。

```
cd  /etc/yum.repos.d/
mv CentOS-Base.repo CentOS-Base.repo.bak
curl -Lo CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
yum clean all
yum makecache
yum update
yum upgrade
```

> 由于CentOS7的内核版本较老，实践发现容易触发一些bug，建议升级内核版本至4.9以上后继续。

2. 安装Docker

k8s和CubeUniverse基于Docker运行，需要下载Docker。在所有节点都需要执行。

CentOS7源自带的Docker版本较低，不能直接使用。故安装docker-ce 20.10.6版本。

```
yum install -y yum-utils   device-mapper-persistent-data   lvm2
yum-config-manager     --add-repo     http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
yum list docker-ce --showduplicates | sort -r
yum install docker-ce-20.10.6
systemctl start docker
systemctl enable docker
```

此时输入 `docker --version` ，应该可以看到正确的版本提示。

Docker在下载镜像时同样需要国内源。设置Docker国内镜像：

```
cat <<EOF > /etc/docker/daemon.json
{
  "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn/"]
}
EOF
```

3. 安装k8s组件

centos7源自带的k8s版本也较低，需要另外添加源以方便下载：

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=http://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=0
repo_gpgcheck=0
gpgkey=http://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
EOF
```

之后即可使用yum直接安装：

```
yum install -y kubelet-1.23.0 kubeadm-1.23.0 kubectl-1.23.0
```

设置服务：

```
systemctl enable kubelet.service
```

注意在所有节点都需要执行，且使用root用户。

4. 部署master节点

在master机器编写kubeadm.yaml配置文件：

```
cat >> kubeadm.yaml <<EOF
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
kubernetesVersion: v1.23.0
imageRepository: registry.cn-hangzhou.aliyuncs.com/google_containers
networking:
  dnsDomain: cluster.local
  podSubnet: 10.244.0.0/16
  serviceSubnet: 10.96.0.0/12
---
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
EOF
```

下载镜像：

```
kubeadm config images pull --config kubeadm.yaml
```

进行集群初始化：

```
kubeadm init --config kubeadm.yaml --upload-certs
```

> 如果出现control-panel相关报错，可以检查kubelet是否成功启动：
>
> ```
> systemctl status kubelet
> journalctl -xeu kubelet
> ```
>
> 如果提示docker与kubelet的cgoup驱动不一致，可以修改/etc/docker/daemon.json:
>
> ```
> {
> "exec-opts": ["native.cgroupdriver=systemd"]
> }
> ```
>
> 然后重启docker：
>
> ```
> systemctl daemon-reload
> systemctl restart docker
> ```

正常的启动应该在20秒内完成。如果出现`Your Kubernetes control-plane has initialized successfully!`说明启动成功。

接下来会弹出本机的相关信息和加入集群所需的token和指令，复制下来。例如：

```
kubeadm join 192.168.79.11:6443 --token j53hrl.kvy4ruyautc3j9mf --discovery-token-ca-cert-hash
sha256:464a2f1f4d712454a0eab21932587d5319766a3406ea1aa4fc71ae08ca823adb
```

5. 部署其他节点

其他节点的部署可以使用刚刚得到的命令来完成。注意这里还是要确保docker和k8s使用同一个cgroup驱动，并且要使用root用户。

```
kubeadm join 192.168.79.11:6443 --token j53hrl.kvy4ruyautc3j9mf --discovery-token-ca-cert-hash
sha256:464a2f1f4d712454a0eab21932587d5319766a3406ea1aa4fc71ae08ca823adb
```

如果看到`This node has joined the cluster`之类的信息，说明节点已经加入成功。

6. 配置环境变量

要访问api-server（比如使用kubectl get），必须获取k8s集群的配置。这里用环境变量的方式来引入：

```
echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bash_profile
source ~/.bash_profile
```

如果不这样设置，使用kubectl时可能会报错。

之后就可以查看我们已经加入的节点情况了：

```
[root@master master]# kubectl get nodes
NAME     STATUS     ROLES                  AGE     VERSION
master   NotReady   control-plane,master   16m     v1.23.0
node1    NotReady   <none>                 4m29s   v1.23.0
```

7. 配置网络插件

可以看到，两个节点都处于Pending状态，这是因为网络插件还未安装。

这里我们下载flannel的配置文件：

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

然后让集群应用：

```
kubectl apply -f kube-flannel.yml
```

再次检查节点，发现已经变为Ready状态：

```
[root@master ~]# kubectl get nodes
NAME     STATUS   ROLES                  AGE   VERSION
master   Ready    control-plane,master   26m   v1.23.0
node1    Ready    <none>                 15m   v1.23.0
```

以此类推，加入其他节点即可。

8. 常见问题

> [preflight] Some fatal errors occurred:
> [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1

```
sysctl -w net.ipv4.ip_forward=1
```

> flannel应用后不能正常运行，保持在error或CrashLoopBackOff状态

一些系统内核模块可能没有启动。

```
cat > /etc/sysconfig/modules/ipvs.modules <<EOF
#!/bin/bash
ipvs_modules="ip_vs ip_vs_lc ip_vs_wlc ip_vs_rr ip_vs_wrr ip_vs_lblc ip_vs_lblcr ip_vs_dh ip_vs_sh ip_vs_fo ip_vs_nq ip_vs_sed ip_vs_ftp nf_conntrack_ipv4"
for kernel_module in \${ipvs_modules}; do
    /sbin/modinfo -F filename \${kernel_module} > /dev/null 2>&1
    if [ $? -eq 0 ]; then
        /sbin/modprobe \${kernel_module}
    fi
done
EOF
chmod 755 /etc/sysconfig/modules/ipvs.modules && bash /etc/sysconfig/modules/ipvs.modules && lsmod | grep ip_vs
```

执行上述脚本。

也有可能是集群未分配pod的cidr。编辑 `/etc/kubernetes/manifests/kube-controller-manager.yaml`，在commands字段添加参数：

```
--allocate-node-cidrs=true
--cluster-cidr=10.244.0.0/16
```

> 在master节点执行kubectl get 时提示：The connection to the server localhost:8080 was refused - did you specify the right host or port?

确保使用的是root用户。执行：

```
echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bash_profile
source ~/.bash_profile
```

> 在非master节点上使用kubectl get 时提示：The connection to the server localhost:8080 was refused - did you specify the right host or port?

需要把master节点的admin信息复制到节点上。

在master节点执行：

```
#复制admin.conf，IP替换成子节点的IP
scp /etc/kubernetes/admin.conf 192.168.79.12:/etc/kubernetes/admin.conf
scp /etc/kubernetes/admin.conf 192.168.79.13:/etc/kubernetes/admin.conf
```

在子节点执行：

```
#设置环境变量
export KUBECONFIG=/etc/kubernetes/admin.conf
echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bash_profile
```

> [ERROR Port-10257]: Port 10257 is in use

```
kubeadm reset
rm -rf $HOME/.kube
```