链滴

CentOS7.x 安装 k8s 集群

作者：justin177

原文链接：https://ld246.com/article/1549102114183

来源网站：链滴

CentOS7.x安装k8s集群

环境：CentOS7.x

一、配置hosts

```
vi /etc/hosts
1.2.3.4 etcd-single
```

二、安装单节点etcd

1. 下载

```
wget https://github.com/etcd-io/etcd/releases/download/v3.3.11/etcd-v3.3.11-linux-amd64.tar.gz
```

2. 安装

```
tar xzf etcd-v3.3.11-linux-amd64.tar.gz
cd etcd-v3.3.11-linux-amd64
cp etcd* /usr/local/bin/
```

3. 配置

```
vi /usr/lib/systemd/system/etcd.service
[Unit]
Description=etcd
[Service]
Environment=ETCD_NAME=etcd-single
Environment=ETCD_DATA_DIR=/web/etcd/data/etcd-single
Environment=ETCD_LISTEN_CLIENT_URLS=http://0.0.0.0:2379
Environment=ETCD_LISTEN_PEER_URLS=http://0.0.0.0:2380
Environment=ETCD_INITIAL_ADVERTISE_PEER_URLS=http://etcd-single:2380
Environment=ETCD_ADVERTISE_CLIENT_URLS=http://etcd-single:2379
Environment=ETCD_INITIAL_CLUSTER_STATE=new
Environment=ETCD_INITIAL_CLUSTER_TOKEN=etcd-single
Environment=ETCD_INITIAL_CLUSTER=etcd-single=http://etcd-single:2380
ExecStart=/usr/local/bin/etcd
[Install]
WantedBy=multi-user.target
```

4. 启动

```
systemctl daemon-reload
systemctl restart etcd
```

5. 创建网络

```
etcdctl --endpoints http://etcd-single:2379  set /coreos.com/network/config '{"NetWork":"10.
.0.0/16","SubnetLen":24,"Backend":{"Type":"vxlan"}}'
```

三、安装flannel

使用yum安装

1. 安装

```
yum install flannel -y
```

2. 配置

```
vi /etc/sysconfig/flanneld
FLANNEL_ETCD_ENDPOINTS="http://etcd-single:2379"
FLANNEL_ETCD_PREFIX="/coreos.com/network"
FLANNEL_OPTIONS="--ip-masq=true --public-ip=$LOCAL_MACHINE_IP"
```

3. 其他步骤见手动安装

手动安装

1. 下载

```
wget https://github.com/coreos/flannel/releases/download/v0.11.0/flannel-v0.11.0-linux-am
64.tar.gz
```

2. 安装

```
mkdir flannel
tar xzf flannel-v0.11.0-linux-amd64.tar.gz -C flannel
cd flannel
cp flannel mk-docker-opts.sh /usr/local/bin/
```

3. 配置

```
vi /usr/lib/systemd/system/flanneld.service
[Unit]
Description=flannel
[Service]
ExecStart=/usr/local/bin/flanneld \
-etcd-endpoints=http://etcd-single:2379 \
-etcd-prefix=/coreos.com/network \
-ip-masq=true \
-public-ip=$LOCAL_MACHINE_IP
ExecStartPost=/usr/local/bin/mk-docker-opts.sh -k DOCKER_NETWORK_OPTIONS -d /run/fla
nel/docker
[Install]
WantedBy=multi-user.target
```

4. 启动

```
systemctl daemon-reload
systemctl restart flanneld
```

5. 验证

- 新网络设备

```
ifconfig
ip a
```

看是否有名称开头为 flannel 的网络设备，查看其 inet 地址，如下：

```
3: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKN
WN
    link/ether 36:41:09:3f:31:59 brd ff:ff:ff:ff:ff:ff
    inet 10.0.33.0/32 scope global flannel.1
       valid_lft forever preferred_lft forever
    inet6 fe80::3441:9ff:fe3f:3159/64 scope link
       valid_lft forever preferred_lft forever
```

- 新子网

```
etcdctl --endpoints http://etcd-single:2379 ls /coreos.com/network/subnets
```

看是否有新增 subnet，以上 flannel 的 inet 地址是否在 subnets 当中，如下：

```
/coreos.com/network/subnets/10.0.33.0-24
```

查看子网信息：

```
etcdctl --endpoints http://etcd-single:2379 get /coreos.com/network/subnets/10.0.33.0-24
```

```
{"PublicIP":"$LOCAL_MACHINE_IP","BackendType":"vxlan","BackendData":{"VtepMAC":"36:41:
9:3f:31:59"}}
```

四、安装docker

1. 安装

参：Get Docker CE for CentOS

SET UP THE REPOSITORY

```
yum install -y yum-utils \
  device-mapper-persistent-data \
  lvm2

yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
```

INSTALL DOCKER CE

```
yum install docker-ce docker-ce-cli containerd.io
```

2. 配置添加

```
vi /usr/lib/systemd/system/docker.service
EnvironmentFile=/run/flannel/docker
ExecStart=/usr/bin/dockerd -H unix:// $DOCKER_NETWORK_OPTIONS
```

3. 启动

```
systemctl daemon-reload
systemctl restart docker
```

4. 验证

- 新网络设备

```
ifconfig
ip a
```

看是否有名称开头为 docker 的网络设备，查看其 inet 地址，是否在 flannel 的 subnet 当中，如下：

```
3: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKN
WN
    link/ether 36:41:09:3f:31:59 brd ff:ff:ff:ff:ff:ff
    inet 10.0.33.0/32 scope global flannel.1
       valid_lft forever preferred_lft forever
    inet6 fe80::3441:9ff:fe3f:3159/64 scope link
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DO
N
    link/ether 02:42:e5:e3:22:6c brd ff:ff:ff:ff:ff:ff
    inet 10.0.33.1/24 brd 10.0.33.255 scope global docker0
       valid_lft forever preferred_lft forever
```

五、配置ip forward

1. 配置iptables设置允许forward

```
iptables -P FORWARD ACCEPT
iptables-save
```

2. 配置sysctl文件

```
vi /etc/sysctl.conf
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
sysctl -p
cat /proc/sys/net/ipv4/conf/all/forwarding
cat /proc/sys/net/ipv6/conf/all/forwarding
```

六、重新启动docker

```
systemctl restart docker
```

七、连通性测试

1. 查看子网

```
etcdctl --endpoints http://etcd-single:2379 ls /coreos.com/network/subnets
```

当前所有的 subnets，如下：

```
/coreos.com/network/subnets/10.0.41.0-24
/coreos.com/network/subnets/10.0.33.0-24
```

2. 在各个docker上起一个container

```
docker run -d --name c01 httpd
```

3. ping 测试

```
 ping -c2 10.0.33.2
 ping -c2 10.0.41.2
```

能 ping 通则表示配置成功

```
# ping -c2 10.0.33.2
PING 10.0.33.2 (10.0.33.2) 56(84) bytes of data.
64 bytes from 10.0.33.2: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 10.0.33.2: icmp_seq=2 ttl=64 time=0.068 ms

--- 10.0.33.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.053/0.060/0.068/0.010 ms
# ping -c2 10.0.41.2
PING 10.0.41.2 (10.0.41.2) 56(84) bytes of data.
64 bytes from 10.0.41.2: icmp_seq=1 ttl=63 time=180 ms
64 bytes from 10.0.41.2: icmp_seq=2 ttl=63 time=180 ms

--- 10.0.41.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 180.221/180.284/180.348/0.429 ms
```

八、安装k8s

1. 下载

```
wget https://github.com/kubernetes/kubernetes/releases/download/v1.13.3/kubernetes.tar.g
```

2. 安装

```
tar xzf kubernetes.tar.gz
cd kubernetes
bash cluster/get-kube-binaries.sh
cp ./client/bin/kubectl /usr/local/bin/
cd server
tar xzf kubernetes-server-linux-amd64.tar.gz
cd kubernetes/server/bin
```

只在master上：

```
cp  kube-apiserver kube-controller-manager kube-scheduler kubectl /usr/local/bin/

vi /usr/lib/systemd/system/kube-apiserver.service
[Unit]
Description=kube-apiserver
[Service]
ExecStart=/usr/local/bin/kube-apiserver \
--etcd-servers=http://etcd-single:2379 \
--etcd-prefix=/k8s/registry \
--insecure-bind-address=0.0.0.0 \
--insecure-port=8080
[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl restart kube-apiserver
systemctl status kube-apiserver
netstat -anop | grep 6443
netstat -anop | grep 8080

vi /usr/lib/systemd/system/kube-scheduler.service
[Unit]
Description=kube-scheduler
[Service]
ExecStart=/usr/local/bin/kube-scheduler \
--master=http://etcd-single:8080
[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl restart kube-scheduler
systemctl status kube-scheduler
netstat -anop | grep 10259
netstat -anop | grep 10251

vi /usr/lib/systemd/system/kube-controller-manager.service
[Unit]
Description=kube-controller-manager
[Service]
ExecStart=/usr/local/bin/kube-controller-manager \
--master=http://etcd-single:8080
[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl restart kube-controller-manager
systemctl status kube-controller-manager
netstat -anop | grep 10257
netstat -anop | grep 10252
```

所有node上：

```
cp kube-proxy kubelet /usr/local/bin/
```

```
vi /usr/lib/systemd/system/kube-proxy.service
[Unit]
Description=kube-proxy
[Service]
ExecStart=/usr/local/bin/kube-proxy \
--master=http://etcd-single:8080
[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl restart kube-proxy
systemctl status kube-proxy
netstat -anop | grep 10256
netstat -anop | grep 10249

mkdir -p /opt/kubernetes/cfg
vi /opt/kubernetes/cfg/kubelet.kubeconfig
apiVersion: v1
kind: Config
clusters:
  - cluster:
      server: http://etcd-single:8080/
    name: local
contexts:
  - context:
      cluster: local
    name: local
current-context: local

vi /usr/lib/systemd/system/kubelet.service
[Unit]
Description=kubelet
[Service]
ExecStart=/usr/local/bin/kubelet \
--fail-swap-on=false \
--hostname-override=$NODE_NAME \
--kubeconfig=/opt/kubernetes/cfg/kubelet.kubeconfig
[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl restart kubelet
systemctl status kubelet
netstat -anop | grep 10255
netstat -anop | grep 10250
netstat -anop | grep 10248
```

查看注册的nodes：

```
kubectl get nodes
```