Kubeadm 部署kubenetes 1.9

2017年12月19日 16:09

环境：CentOS Linux release 7.3.1611
内核：Linux gw-k8s-master 3.10.0-514.el7.x86_64
科学上网:shadowsockes+privoxy

1. 准备工作

   a.修改时区和语言

   ```
   [root@gw-k8s-master ~]# localectl  set-locale LANG=zh_CN.utf8
   [root@gw-k8s-master ~]# timedatectl set-timezone Asia/Shanghai
   ```

   优化

   ```
   [root@gw-k8s-master ~]# sed -i "s/^SELINUX\=enforcing/SELINUX\=disabled/g" /etc/selinux/config
   [root@gw-k8s-master ~]# echo alias vi=vim >> /etc/bashrc
   ```

   b.安装privoxy和启动脚本,科学上网用

   ```
   [root@gw-k8s-master ~]# yum -y install privoxy
   ```

   修改配置文件

   ```
   [root@gw-k8s-master ~]# /etc/privoxy/config
   forward-socks5t   /             shadowsockesip:port
   ```

   编写shell翻墙脚本到/root/，每次执行需要断开终端重新连接

   启动shell上网

   ```
   [root@gw-k8s-master ~]# cat vpn-start.sh
   #!/bin/bash
   echo export http_proxy=http://127.0.0.1:8118  >>/etc/profile
   echo export https_proxy=http://127.0.0.1:8118  >>/etc/profile
   echo export ftp_proxy=http://127.0.0.1:8118  >>/etc/profile
   source /etc/profile
   systemctl restart privoxy
   ```

   断开shell科学上网

   ```
   [root@gw-k8s-master ~]# cat vpn-stop.sh
   #!/bin/bash
   sed  -i '/127.0.0.1:8118/d'   /etc/profile
   source /etc/profile
   ```

   c. hosts文件

   ```
    [root@gw-k8s-master ~]#  echo 127.0.0.1 gw-k8s-master >>/etc/hosts
     [root@gw-k8s-master ~]# echo 192.168.2.230 gw-k8s-master >>/etc/hosts
   ```

2. 安装docker

   ```
   [root@gw-k8s-master ~]# yum install -y yum-utils device-mapper-persistent-data lvm2 net-tools
   [root@gw-k8s-master ~]# yum install -y  docker
   [root@gw-k8s-master ~]# systemctl restart docker && systemctl enable docker
   ```

   查看Cgroup Driver

   ```
   [root@gw-k8s-master ~]# docker info |grep "Cgroup Driver"
   Cgroup Driver: systemd
   ```

3. 其他准备工作

   ```
   [root@gw-k8s-master ~]# iptables -P FORWARD ACCEPT
   [root@gw-k8s-master ~]# echo "sleep 30 && /sbin/iptables -P FORWARD ACCEPT" >>  /etc/rc.local
   [root@gw-k8s-master ~]# echo "modprobe br_netfilter" >> /etc/rc.local
   [root@gw-k8s-master ~]# chmod +x /etc/rc.local
   [root@gw-k8s-master ~]# Swapoff -a
   ```

   注释掉/etc/fstab swap 行

   ```
   [root@gw-k8s-master ~]# yum install -y yum-utils device-mapper-persistent-data lvm2 net-tools
   [root@gw-k8s-master ~]# yum install -y docker
   ```

   Docker pull images 科学上网

   修改docker配置文件 pull images通过代理出去

   ```
   [root@gw-k8s-master ~]# mkdir -p /etc/systemd/system/docker.service.d
   [root@gw-k8s-master ~]# cat /etc/systemd/system/docker.service.d/http-proxy.conf
   [Service]
   Environment="HTTP_PROXY=http://127.0.0.1:8118/" "HTTPS_PROXY=http://127.0.0.1:8118/"

   [root@gw-k8s-master ~]# systemctl daemon-reload
   [root@gw-k8s-master ~]# systemctl restart docker
   [root@gw-k8s-master ~]# systemctl show docker --property Environment
   ```

修改网络配置

```
[root@gw-k8s-master ~]# cat  <<EOF >  /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
vm.swappiness=0
EOF
```

增加kubernetes源

```
[root@gw-k8s-master ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

部署kubeadm

先翻墙 /root/vpn-start.sh 执行后断开终端 重新连接

```
[root@gw-k8s-master ~]# yum install -y kubelet kubeadm kubectl

[root@gw-k8s-master ~]# cat  /etc/systemd/system/kubelet.service.d/10-kubeadm.conf  |grep "cgroup-driver"
```

如果不是systemd,进行修改

```
[root@gw-k8s-master ~]# systemctl enable kubelet.service
```

关闭翻墙

```
/root/vpn-stop.sh
```

重启终端

```
[root@gw-k8s-master ~]# systemctl restart privoxy

[root@gw-k8s-master ~]# kubeadm init   --kubernetes-version=v1.9.0   --pod-network-cidr=10.244.0.0/16  --apiserver-advertise-address=192.168.2.230
[init] Using Kubernetes version: v1.9.0
[init] Using Authorization modes: [Node RBAC]
[preflight] Running pre-flight checks.
	[WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
	[WARNING FileExisting-crictl]: crictl not found in system path
[preflight] Starting the kubelet service
[certificates] Generated ca certificate and key.
[certificates] Generated apiserver certificate and key.
[certificates] apiserver serving cert is signed for DNS names [gw-k8s-master kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 192.168.2.230]
[certificates] Generated apiserver-kubelet-client certificate and key.
[certificates] Generated sa key and public key.
[certificates] Generated front-proxy-ca certificate and key.
[certificates] Generated front-proxy-client certificate and key.
[certificates] Valid certificates and keys now exist in "/etc/kubernetes/pki"
[kubeconfig] Wrote KubeConfig file to disk: "admin.conf"
[kubeconfig] Wrote KubeConfig file to disk: "kubelet.conf"
[kubeconfig] Wrote KubeConfig file to disk: "controller-manager.conf"
[kubeconfig] Wrote KubeConfig file to disk: "scheduler.conf"
[controlplane] Wrote Static Pod manifest for component kube-apiserver to "/etc/kubernetes/manifests/kube-apiserver.yaml"
[controlplane] Wrote Static Pod manifest for component kube-controller-manager to "/etc/kubernetes/manifests/kube-controller-manager.yaml"
[controlplane] Wrote Static Pod manifest for component kube-scheduler to "/etc/kubernetes/manifests/kube-scheduler.yaml"
[etcd] Wrote Static Pod manifest for a local etcd instance to "/etc/kubernetes/manifests/etcd.yaml"
[init] Waiting for the kubelet to boot up the control plane as Static Pods from directory "/etc/kubernetes/manifests".
[init] This might take a minute or longer if the control plane images have to be pulled.
[apiclient] All control plane components are healthy after 757.507669 seconds
[uploadconfig]?Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[markmaster] Will mark node gw-k8s-master as master by adding a label and a taint
[markmaster] Master gw-k8s-master tainted and labelled with key/value: node-role.kubernetes.io/master=""
[bootstraptoken] Using token: 43083c.e1942cbccb677789
[bootstraptoken] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certif icate credentials
[bootstraptoken] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap T oken
[bootstraptoken] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstraptoken] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: kube-dns
[addons] Applied essential addon: kube-proxy

Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

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

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

You can now join any number of machines by running the following on each node
```

as root:

```
kubeadm join --token 43083c.e1942cbccb677789 192.168.2.230:6443 --discovery-token-ca-cert-hash
sha256:d894cf972ba16af964c6bd5d148766a72c0eb33169a24fb6f03eb5d8d2979ef1
```

```
 [root@gw-k8s-master ~]# mkdir -p $HOME/.kube
 [root@gw-k8s-master ~]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
 [root@gw-k8s-master ~]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

安装flannel 网络

```
[root@gw-k8s-master ~]# mkdir ~/k8s
[root@gw-k8s-master ~]# cd ~/k8s
[root@gw-k8s-master ~]# wget https://raw.githubusercontent.com/coreos/flannel/v0.9.1/Documentation/kube-flannel.yml
[root@gw-k8s-master ~]#  vi kube-flannel.yml
command: [ "/opt/bin/flanneld", "--ip-masq", "--kube-subnet-mgr","--ifcfe=ens160"]  #填写物理网卡的name
[root@gw-k8s-master ~]# kubectl apply -f kube-flannel.yml
[root@gw-k8s-master ~]# echo "1" > /proc/sys/net/bridge/bridge-nf-call-iptables
```

Node 执行

添加node到群集

```
kubeadm join --token 43083c.e1942cbccb677789 192.168.2.230:6443 --discovery-token-ca-cert-hash
sha256:d894cf972ba16af964c6bd5d148766a72c0eb33169a24fb6f03eb5d8d2979ef1
scp root@<master ip>:/etc/kubernetes/admin.conf .
kubectl --kubeconfig ./admin.conf get nodes
```

移出群集

```
kubectl drain gw-k8s-node1 --delete-local-data --force --ignore-daemonsets
kubectl delete node gw-k8s-node1
```

```
[root@gw-k8s-master k8s]# kubectl get node
NAME           STATUS    ROLES     AGE       VERSION
gw-k8s-master   Ready     master    54m       v1.9.0
gw-k8s-node1    Ready     <none>    31m       v1.9.0
```

4.安装dashboard

```
[root@gw-k8s-master k8s]# wget https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml
[root@gw-k8s-master k8s]# kubectl create -f kubernetes-dashboard.yaml

[root@gw-k8s-master k8s]# kubectl -n kube-system get service kubernetes-dashboard
NAME                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes-dashboard   ClusterIP   10.96.188.202   <none>        443/TCP   1m

[root@gw-k8s-master k8s]#  vi kubectl -n kube-system edit service kubernetes-dashboard
      type: NodePort

[root@gw-k8s-master k8s]# kubectl -n kube-system get service kubernetes-dashboard
NAME                   TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)         AGE
kubernetes-dashboard   NodePort   10.96.188.202   <none>        443:32497/TCP   1m

[root@gw-k8s-master k8s]#  kubectl get pods --all-namespaces
NAMESPACE     NAME                                    READY     STATUS    RESTARTS   AGE
default       curl-545bbf5f9c-nqsrt                   1/1       Running   0          1d
kube-system   etcd-gw-k8s-master                      1/1       Running   3          12h
kube-system   kube-apiserver-gw-k8s-master            1/1       Running   9          12h
kube-system   kube-controller-manager-gw-k8s-master   1/1       Running   3          12h
kube-system   kube-dns-6f4fd4bdf-qs66s                3/3       Running   0          1d
kube-system   kube-flannel-ds-c6dkp                   1/1       Running   0          26m
kube-system   kube-flannel-ds-ptwcm                   1/1       Running   0          26m
kube-system   kube-proxy-mjbkx                        1/1       Running   3          1d
kube-system   kube-proxy-mn7fp                        1/1       Running   0          1d
kube-system   kube-scheduler-gw-k8s-master            1/1       Running   3          12h
kube-system   kubernetes-dashboard-7b7b5cd79b-nf9gx   1/1       Running   0          1m
```

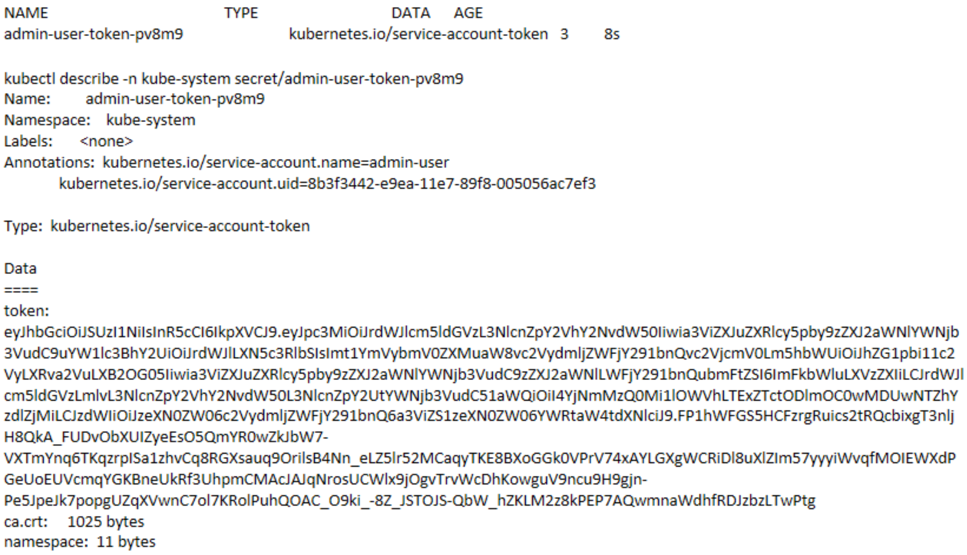
增加admin-user secret 使用admin-user token登录并有管理员权限

```
[root@gw-k8s-master k8s]#  cat kubernetes-dashboard-admin.rbac.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: admin-user
  namespace: kube-system

---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: admin-user
  labels:
    k8s-app: kubernetes-dashboard
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kube-system
```

```
[root@gw-k8s-master k8s]# kubectl create -f  kubernetes-dashboard-admin.rbac.yaml

[root@gw-k8s-master k8s]#  kubectl -n kube-system get secret|grep admin-user
```



```
NAME                              TYPE                          DATA     AGE
admin-user-token-pv8m9                    kubernetes.io/service-account-token   3        8s

kubectl describe -n kube-system secret/admin-user-token-pv8m9
Name:        admin-user-token-pv8m9
Namespace:   kube-system
Labels:      <none>
Annotations:  kubernetes.io/service-account.name=admin-user
          kubernetes.io/service-account.uid=8b3f3442-e9ea-11e7-89f8-005056ac7ef3

Type:  kubernetes.io/service-account-token

Data
====
token:
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb
3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2
VyLXRva2VuLXB2OG05Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJl
cm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiI4YjNmMzQ0Mi1lOWVhLTExZTctODlmOC0wMDUwNTZhY
zdlZjMiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.FP1hWFGS5HCFzrgRuics2tRQcbixgT3nlj
H8QkA_FUDvObXUIZyeEsO5QmYR0wZkJbW7-
VXTmYnq6TKqzrpISa1zhvCq8RGXsauq9OrilsB4Nn_eLZ5lr52MCaqyTKE8BXoGGk0VPrV74xAYLGXgWCRiDl8uXlZIm57yyyiWvqfMOIEWXdP
GeUoEUVcmqYGKBneUkRf3UhpmCMAcJAJqNrosUCWlx9jOgvTrvWcDhKowguV9ncu9H9gjn-
Pe5JpeJk7popgUZqXVwnC7ol7KRolPuhQOAC_O9ki_-8Z_JSTOJS-QbW_hZKLM2z8kPEP7AQwmnaWdhfRDJzbzLTwPtg
ca.crt:     1025 bytes
namespace:  11 bytes
```

使用火狐浏览器：

https://master-ip:32497