kubernetes

kubernetes 安装

kube-master安装

按照如下配置准备云主机

主机名	IP地址	最低配置
master	192.168.1.21	8CPU,16G内存
node-0001	192.168.1.31	8CPU,16G内存
node-0002	192.168.1.32	8CPU,16G内存
node-0003	192.168.1.33	8CPU,16G内存
harbor	192.168.1.100	1CPU,1G内存

1、防火墙相关配置

参考前面知识点完成禁用 selinux，禁用 swap，卸载 firewalld-*

上传kubernetes.zip到js跳板机

2、配置yum仓库(跳板机)

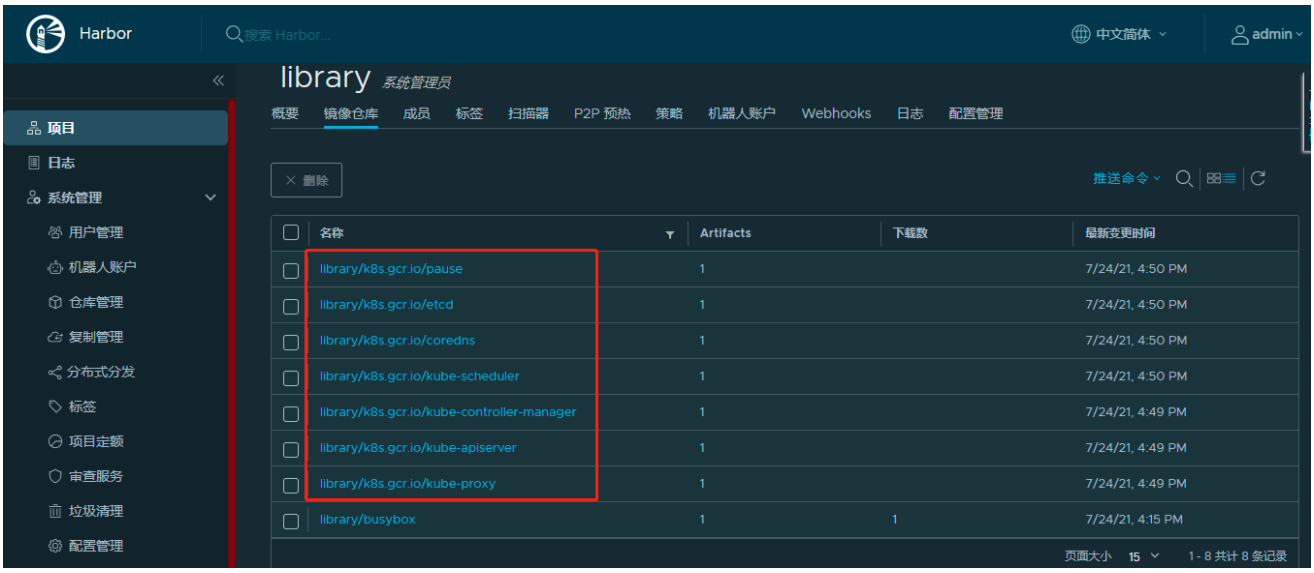
```
跳板机js（1.252）主机配置k8s软件源服务端
[root@js ~]# yum -y install vsftpd
[root@js ~]# mkdir /var/ftp/localrepo
[root@js ~]# systemctl restart vsftpd
[root@js ~]# cd project3/jumpserver/
[root@js kubernetes]# cp -a v1.17.6/k8s-install/ /var/ftp/localrepo/
[root@js kubernetes]# cd /var/ftp/localrepo/
[root@js localrepo]# createrepo .   #如果之前是做好的，可以createrepo --update .更新

master主机更改yum配置文件
[root@master ~]# vim /etc/yum.repos.d/local.repo
[k8s]
name=k8s
baseurl=ftp://192.168.1.252/localrepo
enabled=1
gpgcheck=0

同步到node1,node2,node3
[root@master ~]# scp /etc/yum.repos.d/local.repo 192.168.1.31:/etc/yum.repos.d/
[root@master ~]# scp /etc/yum.repos.d/local.repo 192.168.1.32:/etc/yum.repos.d/
[root@master ~]# scp /etc/yum.repos.d/local.repo 192.168.1.33:/etc/yum.repos.d/
```

```
跳板机js（1.252）主机配置docker的软件源服务端
[root@js localrepo]# cp /root/project3/jumpserver/docker-ce-18.06.3.ce-3.el7.x86_64.rpm  ./k8s-install/
[root@js localrepo]# createrepo --update .

3、安装软件包(master)
安装kubeadm、kubectl、kubelet、docker-ce
[root@master ~]# yum install -y kubeadm kubelet kubectl docker-ce
[root@master ~]# mkdir -p /etc/docker
[root@master ~]# vim /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
[root@master ~]# systemctl enable --now docker kubelet
[root@master ~]# docker info |grep Cgroup
Cgroup Driver: systemd
[root@master ~]# vim /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
[root@master ~]# modprobe br_netfilter
[root@master ~]# sysctl --system
```

4、镜像导入私有仓库

```
[root@master ~]# vim /usr/lib/systemd/system/docker.service
12 ExecStart=/usr/bin/dockerd --insecure-registry 192.168.1.100:80
[root@master ~]# systemctl daemon-reload && systemctl enable docker && systemctl restart docker
登录harbor，如果harbor没有启动，需要去harbor主机启动
[root@master ~]# docker login http://192.168.1.100:80
Username: admin
Password:        #密码：Harbor12345
Login Succeeded

在js主机把project3/kubernetes/v1.17.6/base-images 中的镜像拷贝到 master主机
[root@js ~]# scp -r project3/kubernetes/v1.17.6/base-images 192.168.1.21:/root/

master主机操作
[root@master ~]# cd base-images/
[root@master base-image]# for i in *.tar.gz;do docker load -i ${i};done
[root@master base-image]# docker tag k8s.gcr.io/kube-proxy:v1.17.6
192.168.1.100:80/library/k8s.gcr.io/kube-proxy:v1.17.6

[root@master base-image]# docker tag k8s.gcr.io/kube-apiserver:v1.17.6
192.168.1.100:80/library/k8s.gcr.io/kube-apiserver:v1.17.6

[root@master base-image]# docker tag k8s.gcr.io/kube-controller-manager:v1.17.6
192.168.1.100:80/library/k8s.gcr.io/kube-controller-manager:v1.17.6

[root@master base-image]# docker tag k8s.gcr.io/kube-scheduler:v1.17.6
192.168.1.100:80/library/k8s.gcr.io/kube-scheduler:v1.17.6

[root@master base-image]# docker tag k8s.gcr.io/coredns:1.6.5
192.168.1.100:80/library/k8s.gcr.io/coredns:1.6.5

[root@master base-image]# docker tag k8s.gcr.io/etcd:3.4.3-0
192.168.1.100:80/library/k8s.gcr.io/etcd:3.4.3-0

[root@master base-image]# docker tag k8s.gcr.io/pause:3.1
192.168.1.100:80/library/k8s.gcr.io/pause:3.1
```

```
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/kube-proxy:v1.17.6
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/kube-apiserver:v1.17.6
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/kube-controller-
manager:v1.17.6
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/kube-scheduler:v1.17.6
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/coredns:1.6.5
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/etcd:3.4.3-0
[root@master base-image]# docker push 192.168.1.100:80/library/k8s.gcr.io/pause:3.1

测试私有仓库是否可以正常使用，node-0001，node-0002，node-0003同样操作
[root@node-0001 ~]# yum -y install docker-ce
[root@node-0001 ~]# mkdir -p /etc/docker

master主机拷贝配置文件到node-0001-0003
[root@master ~]# for i in  192.168.1.{31..33}; do scp /usr/lib/systemd/system/docker.service
$i:/usr/lib/systemd/system/; done

node-0001--0003重启docker，并登录harbor
[root@node-0001 ~]# systemctl daemon-reload && systemctl enable docker && systemctl restart
docker
[root@node-0001 ~]# docker login http://192.168.1.100:80
Username: admin
Password:
Login Succeeded
```



5、Tab键设置

```
master主机设置tab键
[root@master ~]# kubectl completion bash >/etc/bash_completion.d/kubectl
[root@master ~]# kubeadm completion bash >/etc/bash_completion.d/kubeadm
[root@master ~]# exit
```

6、安装IPVS代理软件包

```
[root@master ~]# yum install -y ipvsadm ipset
```

7、配置主机名

```
[root@master ~]# vim /etc/hosts
192.168.1.21    master
192.168.1.31    node-0001
192.168.1.32    node-0002
192.168.1.33    node-0003
192.168.1.100   harbor
```

8、使用kubeadm部署

应答文件在js（1.252）主机的 project3/kubernetes/v1.17.6/config 目录下

```
[root@master ~]# mkdir init;cd init
# 拷贝 kubeadm-init.yaml 到 master 云主机 init 目录下
[root@js ~]# scp  project3/kubernetes/v1.17.6/config/kubeadm-init.yaml 192.168.1.21:/root/init

[root@master init]# vim kubeadm-init.yaml
32 imageRepository: 192.168.1.100:80/library/k8s.gcr.io

[root@master init]# kubeadm init --config=kubeadm-init.yaml |tee master-init.log
# 根据提示执行命令
[root@master init]# mkdir -p $HOME/.kube
[root@master init]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[root@master init]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

9、验证安装结果

```
[root@master ~]# kubectl version
[root@master ~]# kubectl get cs
NAME                 STATUS    MESSAGE             ERROR
controller-manager   Healthy   ok
scheduler            Healthy   ok
etcd-0               Healthy   {"health":"true"}
```

计算节点安装

1、获取token

```
# 创建token
[root@master ~]# kubeadm token create --ttl=0 --print-join-command
[root@master ~]# kubeadm token list
# 获取token_hash
[root@master ~]# openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt |openssl rsa -pubin -
outform der |openssl dgst -sha256 -hex
```

2、node安装，在js主机，使用ansible执行node节点的安装

```
[root@js ~]# cd  project3/kubernetes/
[root@js kubernetes]# unzip ansible.zip
[root@js kubernetes]# cd ansible/
[root@js ansible]# yum -y install ansible-2.4.2.0-2.el7.noarch.rpm
[root@js ~]# ssh-keygen
[root@js ~]# ssh-copy-id 192.168.1.31
[root@js ~]# ssh-copy-id 192.168.1.32
[root@js ~]# ssh-copy-id 192.168.1.33
[root@js ~]# cd  /root/project3/kubernetes/v1.17.6/node-install/
```

```
[root@js node-install]# vim files/hosts
::1             localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1       localhost localhost.localdomain localhost4 localhost4.localdomain4
192.168.1.21    master
192.168.1.31    node-0001
192.168.1.32    node-0002
192.168.1.33    node-0003
[root@js node-install]# vim files/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
[root@js node-install]# vim node_install.yaml
... ...
  vars:
    master: '192.168.1.21:6443'
    token: 'fm6kui.mp8rr3akn74a3nyn'
    token_hash: 'sha256:f46dd7ee29faa3c096cad189b0f9aedf59421d8a881f7623a543065fa6b0088c'
... ...
[root@js node-install]# ansible-playbook node_install.yaml
```

3、master主机验证安装

```
[root@master ~]# kubectl get nodes
NAME        STATUS     ROLES    AGE     VERSION
master      NotReady   master   130m    v1.17.6
node-0001   NotReady   <none>   2m14s   v1.17.6
node-0002   NotReady   <none>   2m15s   v1.17.6
node-0003   NotReady   <none>   2m9s    v1.17.6
```

网络插件安装配置

1、上传镜像到私有仓库

```
拷贝js(1.252)主机project3/kubernetes/v1.17.6/flannel 目录到 master 上
[root@js ~]# scp -r /root/project3/kubernetes/v1.17.6/flannel 192.168.1.21:/root/

master主机操作
[root@master ~]# cd /root/flannel/
[root@master flannel]# docker load -i flannel.tar.gz
[root@master flannel]# docker tag quay.io/coreos/flannel:v0.12.0-amd64
192.168.1.100:80/library/flannel:v0.12.0-amd64
[root@master flannel]# docker push 192.168.1.100:80/library/flannel:v0.12.0-amd64
```

2、修改配置文件并安装

```
[root@master flannel]# vim kube-flannel.yml
128: "Network": "10.244.0.0/16",
172: image: 192.168.1.100:80/library/flannel:v0.12.0-amd64
186: image: 192.168.1.100:80/library/flannel:v0.12.0-amd64
227-结尾: 删除
[root@master flannel]# kubectl apply -f kube-flannel.yml
```

3、验证结果

```
[root@master flannel]# kubectl get nodes
NAME        STATUS   ROLES    AGE     VERSION
master      Ready    master   26h     v1.17.6
node-0001   Ready    <none>   151m    v1.17.6
node-0002   Ready    <none>   152m    v1.17.6
node-0003   Ready    <none>   153m    v1.17.6
```