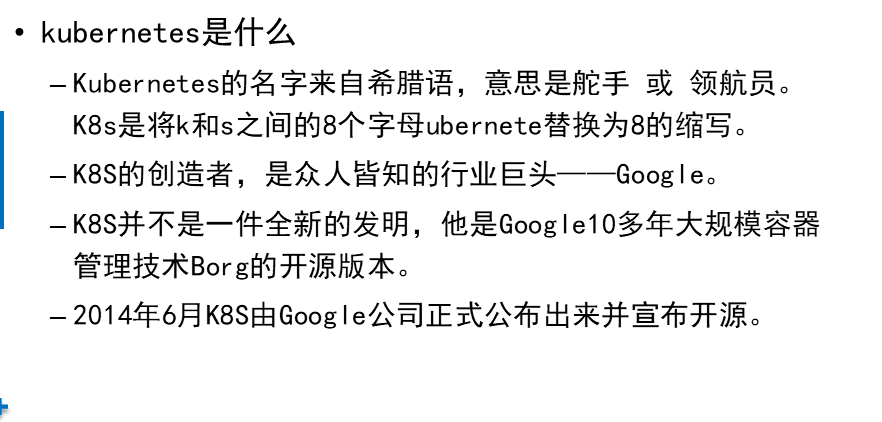
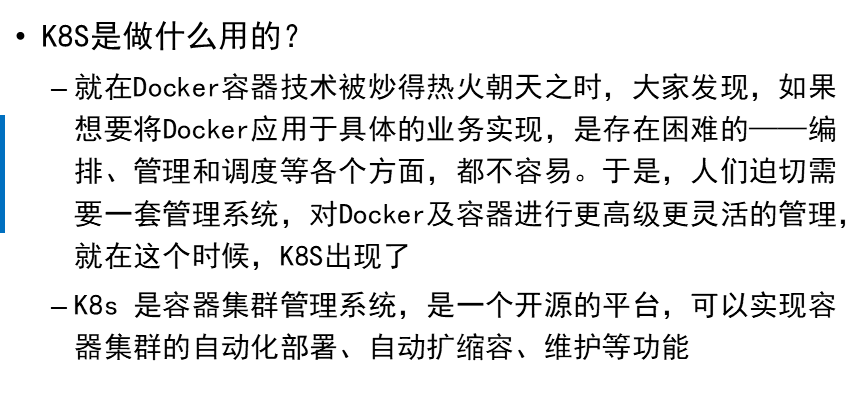
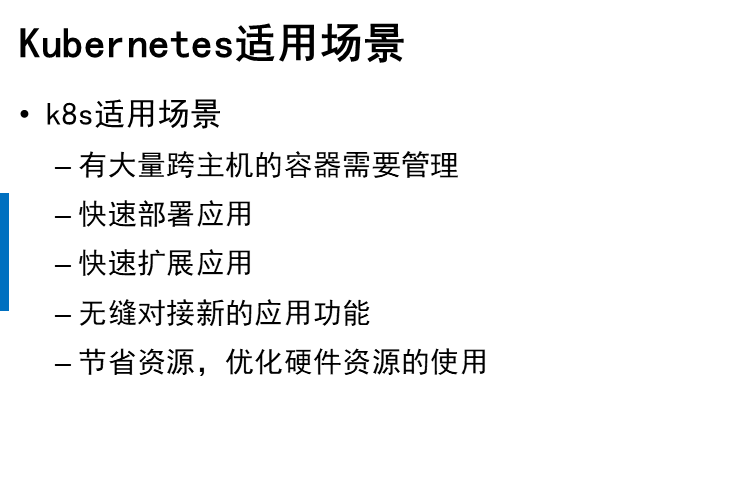
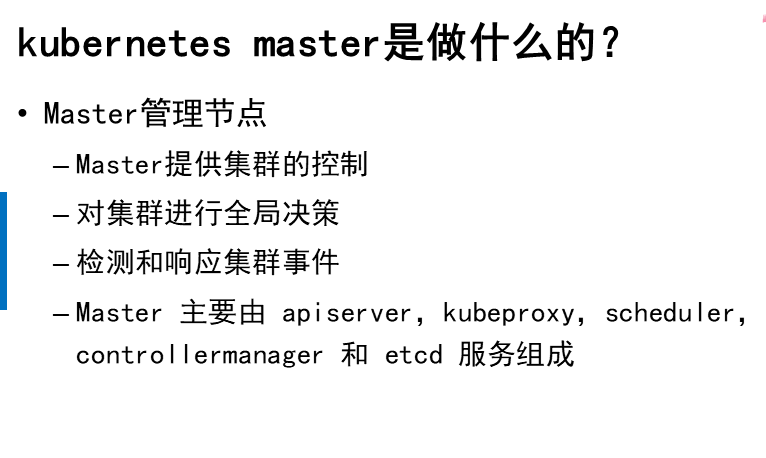
****

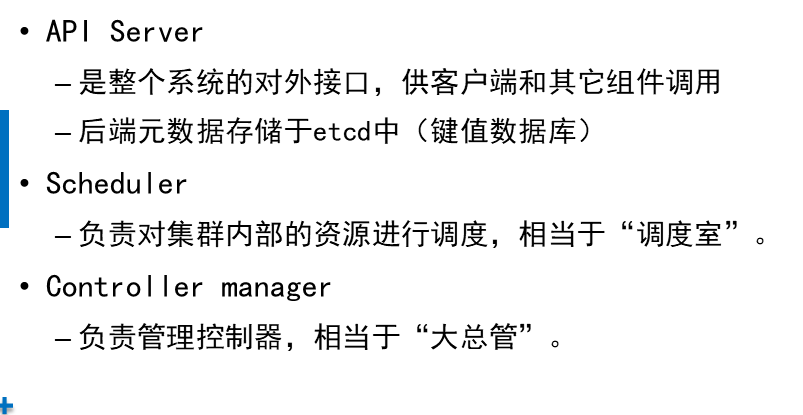
****

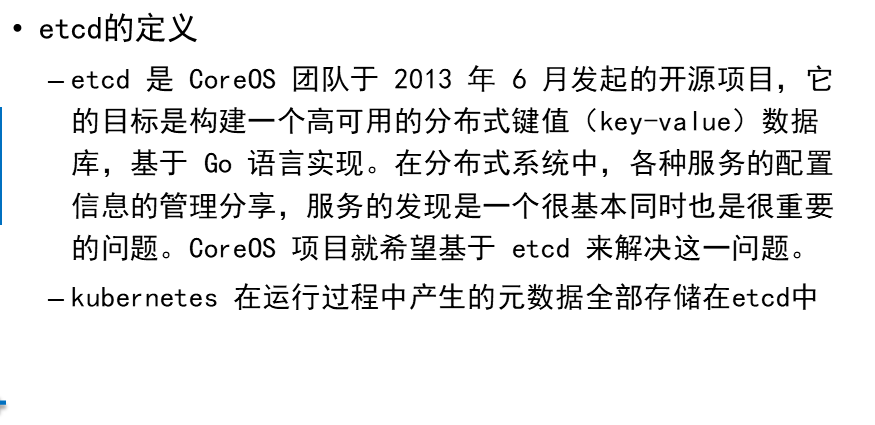
****

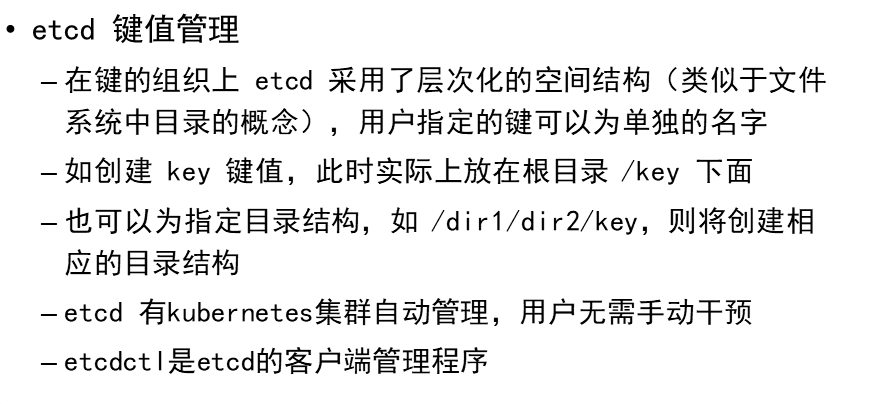
****

****

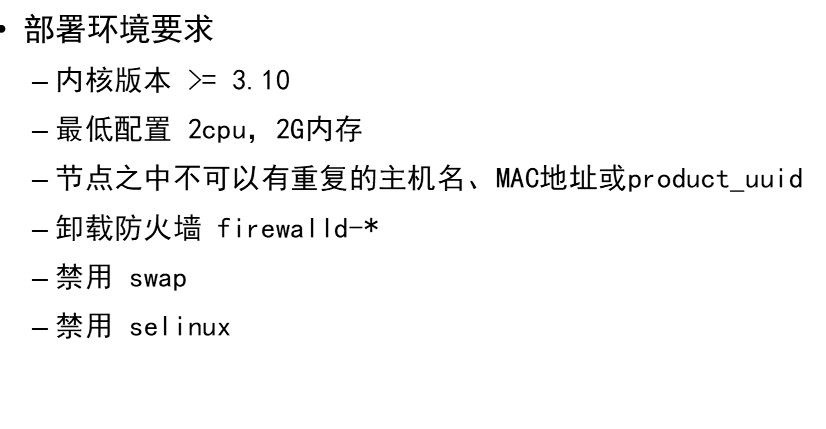
****

****

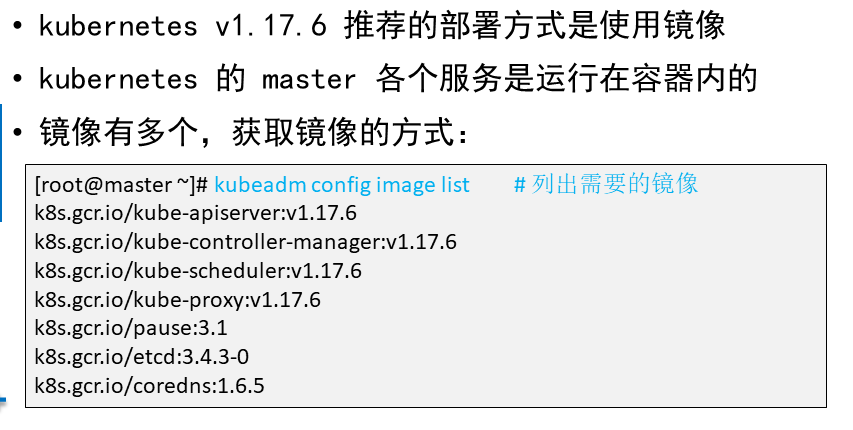
****

****

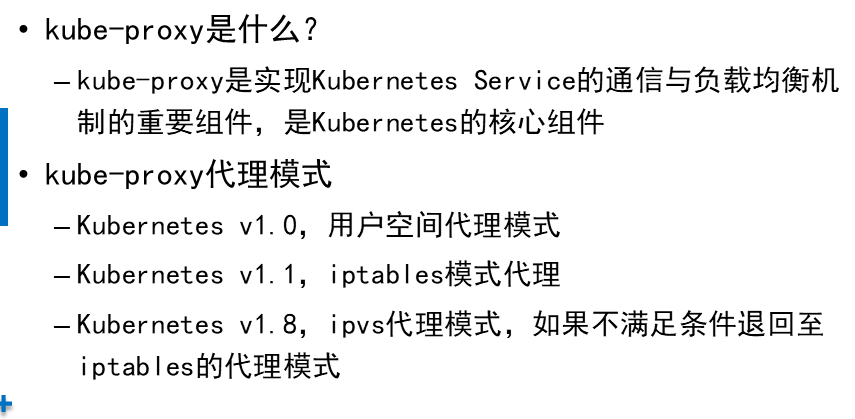
****

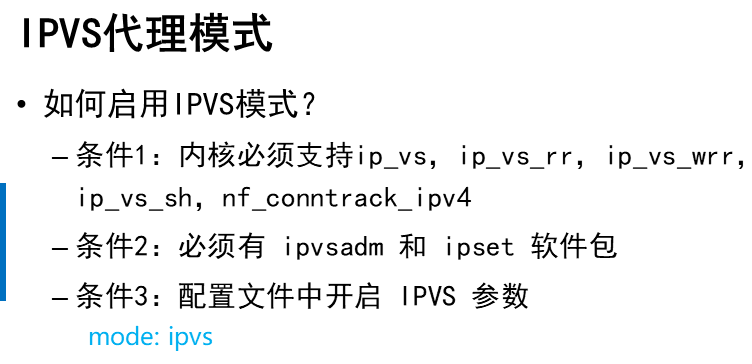
****

****

****

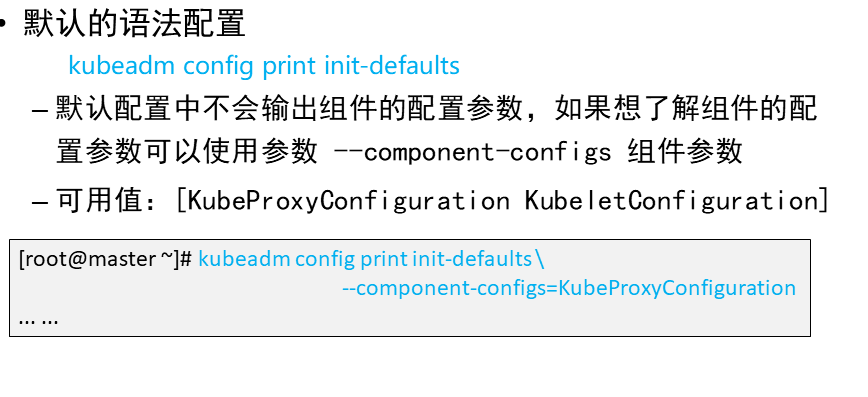
****

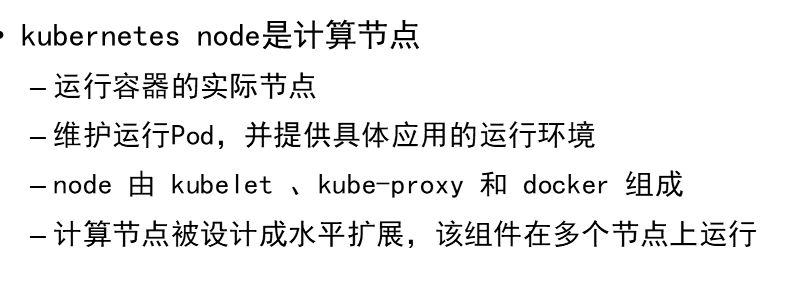
****

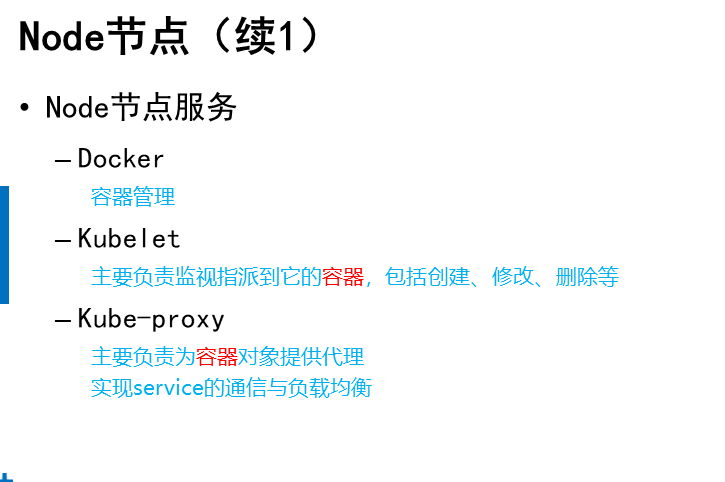
****

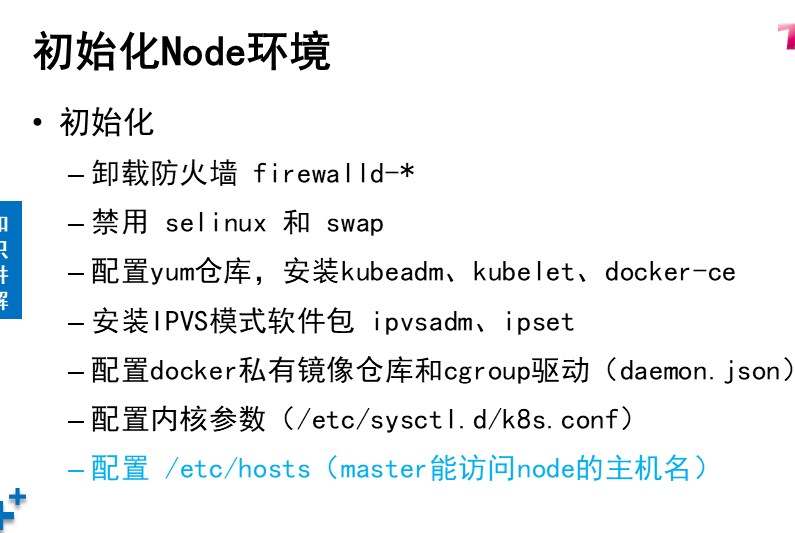
****

****

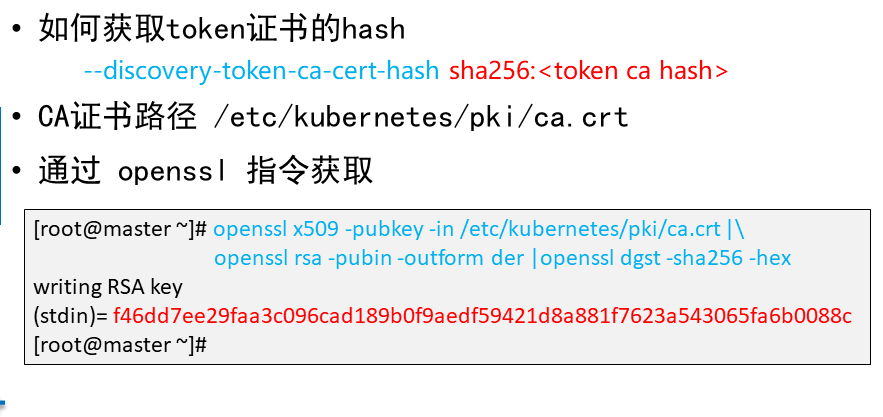
****

****

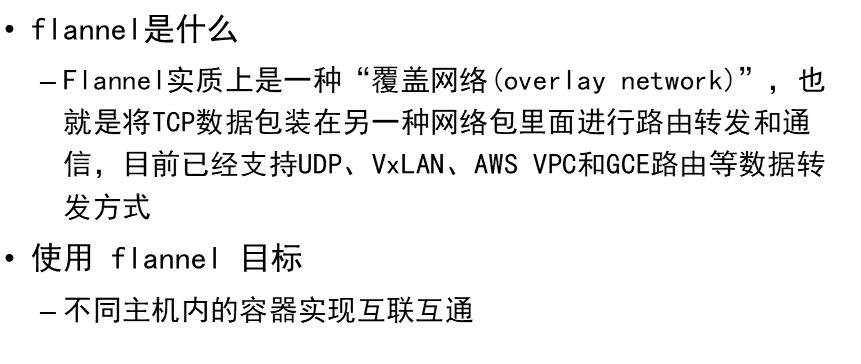
****

****

****

****

****

****

**kubernetes – 01**

**kubernetes 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 | 2CPU,2G内存 |
| node-0001 | 192.168.1.31 | 2CPU,2G内存 |
| node-0002 | 192.168.1.32 | 2CPU,2G内存 |
| node-0003 | 192.168.1.33 | 2CPU,2G内存 |
| registry | 192.168.1.100 | 1CPU,1G内存 |

**kube-master安装**

**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-\*

**2、配置yum仓库**

[root@ecs-proxy ~]# cp -a v1.17.6/k8s-install /var/ftp/localrepo/  
[root@ecs-proxy ~]# cd /var/ftp/localrepo/  
[root@ecs-proxy localrepo]# createrepo --update .

**3、安装工具软件包**

安装kubeadm、kubectl、kubelet、docker-ce

[root@master ~]# yum makecache  
[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"],  
   "registry-mirrors": ["https://hub-mirror.c.163.com"],  
   "insecure-registries":["192.168.1.100:5000", "registry:5000"]  
}  
[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 开启桥设备内核监控  
net.bridge.bridge-nf-call-iptables = 1 开启桥设备内核监控  
net.ipv4.ip\_forward = 1 开启路由转发  
[root@master ~]# modprobe br\_netfilter  
[root@master ~]# sysctl --system 加载上面的配置文件

**4、镜像导入私有仓库**

# 把云盘 kubernetes/v1.17.6/base-images 中的镜像拷贝到 master  
[root@master ~]# cd base-image/  
[root@master base-image]# for i in \*.tar.gz;do docker load -i ${i};done  
[root@master base-image]# docker images  
[root@master base-image]# docker images |awk '$2!="TAG"{print $1,$2}'|while read \_f \_v;do  
  docker tag ${\_f}:${\_v} 192.168.1.100:5000/${\_f##\*/}:${\_v};   
  docker push 192.168.1.100:5000/${\_f##\*/}:${\_v};   
  docker rmi ${\_f}:${\_v};   
done  
# 查看验证  
[root@master base-image]# curl http://192.168.1.100:5000/v2/\_catalog

**5、Tab键设置**

[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、系统初始化，排错**

根据提示排错若干

[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 registry  
[root@master ~]# kubeadm init --dry-run

**8、使用kubeadm部署**

应答文件在云盘的 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@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 componentstatuses  
NAME                       STATUS     MESSAGE             ERROR  
controller-manager       Healthy         ok  
scheduler                   Healthy   ok  
etcd-0                 Healthy   {"health":"true"}

**计算节点安装**

**1、删除实验使用的容器**

node-0001，node-0002 上执行

[root@node-0001 ~]# docker rm -f $(docker ps -aq)  
-----------------------------------------------------------------------------------------  
[root@node-0002 ~]# docker rm -f $(docker ps -aq)

**2、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

**3、node安装**

拷贝云盘上 kubernetes/v1.17.6/node-install 到跳板机

[root@ecs-proxy ~]# cd node-install/  
[root@ecs-proxy 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  
192.168.1.100   registry  
[root@ecs-proxy node-install]# vim node\_install.yaml  
... ...  
vars:  
  master: '192.168.1.21:6443'  
  token: 'fm6kui.mp8rr3akn74a3nyn'  
  token\_hash: 'sha256:f46dd7ee29faa3c096cad189b0f9aedf59421d8a881f7623a543065fa6b0088c'  
... ...  
[root@ecs-proxy node-install]# ansible-playbook node\_install.yaml

**4、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**

拷贝云盘 kubernetes/v1.17.6/flannel 目录到 master 上

**1、上传镜像到私有仓库**

[root@master ~]# cd flannel  
[root@master flannel]# docker load -i flannel.tar.gz  
[root@master flannel]# docker tag quay.io/coreos/flannel:v0.12.0-amd64 192.168.1.100:5000/flannel:v0.12.0-amd64  
[root@master flannel]# docker push 192.168.1.100:5000/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: 192.168.1.100:5000/flannel:v0.12.0-amd64  
186: image: 192.168.1.100:5000/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