安装kubernetes

1、系统配置

系统准备，做host解析

vim /etc/hosts

禁用防火墙，安装iptables

systemctl stop firewalld

systemctl disable firewalld

yum -y install iptables-services

systemctl enable iptables

systemctl start iptables

iptables -F

iptables -X

iptables -L -n

service iptables save

修改内核参数

vim /etc/sysctl.conf

net.ipv4.ip\_forward=1

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-arptables = 1

vm.swappiness=0

关闭swap

swapoff -a

注释自动挂载

vim /etc/fstab

关闭selinux

vim /etc/selinux/config

保存修改内核参数

sysctl -p

2、安装Docker

curl -fsSL get.docker.com -o get-docker.sh

sh get-docker.sh --mirror Aliyun

systemctl enable docker

systemctl start docker

3安装kubeadm和kubelet

yum install -y ebtables socat

修改yum源

cat >> /etc/yum.repos.d/kubernetes.repo <<EOF

[kubernetes]

name=Kubernetes

baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_64/

enabled=1

gpgcheck=0

EOF

安装kubelet kubeadm kubectl

yum install -y kubelet kubeadm kubectl kubernetes-cni

4、修改docker配置

查看kubelet的 /etc/systemd/system/kubelet.service.d/10-kubeadm.conf文件，其中包含如下内容：

Environment="KUBELET\_CGROUP\_ARGS=--cgroup-driver=systemd"

修改为cgroupfs

systemctl daemon-reload

修改docker配置

vim /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

添加加速器

curl -sSL https://get.daocloud.io/daotools/set\_mirror.sh | sh -s http://5da4382e.m.daocloud.io

或者

vim /etc/docker/daemon.json

{"registry-mirrors": ["http://5da4382e.m.daocloud.io"]}

重启docker

systemctl restart docker && systemctl status docker

开机启动kubelet

systemctl enable kubelet.service

先下载docker镜像

master镜像

docker pull cloudnil/etcd-amd64:3.0.17

docker pull cloudnil/pause-amd64:3.0

docker pull cloudnil/kube-proxy-amd64:v1.7.2

docker pull cloudnil/kube-scheduler-amd64:v1.7.2

docker pull cloudnil/kube-controller-manager-amd64:v1.7.2

docker pull cloudnil/kube-apiserver-amd64:v1.7.2

docker pull cloudnil/kubernetes-dashboard-amd64:v1.6.1

docker pull cloudnil/k8s-dns-sidecar-amd64:1.14.4

docker pull cloudnil/k8s-dns-kube-dns-amd64:1.14.4

docker pull cloudnil/k8s-dns-dnsmasq-nanny-amd64:1.14.4

打标

docker tag cloudnil/etcd-amd64:3.0.17 gcr.io/google\_containers/etcd-amd64:3.0.17

docker tag cloudnil/pause-amd64:3.0 gcr.io/google\_containers/pause-amd64:3.0

docker tag cloudnil/kube-proxy-amd64:v1.7.2 gcr.io/google\_containers/kube-proxy-amd64:v1.7.2

docker tag cloudnil/kube-scheduler-amd64:v1.7.2 gcr.io/google\_containers/kube-scheduler-amd64:v1.7.2

docker tag cloudnil/kube-controller-manager-amd64:v1.7.2 gcr.io/google\_containers/kube-controller-manager-amd64:v1.7.2

docker tag cloudnil/kube-apiserver-amd64:v1.7.2 gcr.io/google\_containers/kube-apiserver-amd64:v1.7.2

docker tag cloudnil/kubernetes-dashboard-amd64:v1.6.1 gcr.io/google\_containers/kubernetes-dashboard-amd64:v1.6.1

docker tag cloudnil/k8s-dns-sidecar-amd64:1.14.4 gcr.io/google\_containers/k8s-dns-sidecar-amd64:1.14.4

docker tag cloudnil/k8s-dns-kube-dns-amd64:1.14.4 gcr.io/google\_containers/k8s-dns-kube-dns-amd64:1.14.4

docker tag cloudnil/k8s-dns-dnsmasq-nanny-amd64:1.14.4 gcr.io/google\_containers/k8s-dns-dnsmasq-nanny-amd64:1.14.4

node镜像

docker pull cloudnil/etcd-amd64:3.0.17

docker pull cloudnil/pause-amd64:3.0

docker pull cloudnil/kube-proxy-amd64:v1.7.2

docker pull cloudnil/kubernetes-dashboard-amd64:v1.6.1

docker pull cloudnil/k8s-dns-sidecar-amd64:1.14.4

docker pull cloudnil/k8s-dns-kube-dns-amd64:1.14.4

docker pull cloudnil/k8s-dns-dnsmasq-nanny-amd64:1.14.4

images=(pause-amd64:3.0 kube-proxy-amd64:v1.7.2)

for imageName in ${images[@]} ; do

docker pull cloudnil/$imageName

docker tag cloudnil/$imageName gcr.io/google\_containers/$imageName

docker rmi cloudnil/$imageName

done

打标

docker tag cloudnil/etcd-amd64:3.0.17 gcr.io/google\_containers/etcd-amd64:3.0.17

docker tag cloudnil/pause-amd64:3.0 gcr.io/google\_containers/pause-amd64:3.0

docker tag cloudnil/kube-proxy-amd64:v1.7.2 gcr.io/google\_containers/kube-proxy-amd64:v1.7.2

docker tag cloudnil/kubernetes-dashboard-amd64:v1.6.1 gcr.io/google\_containers/kubernetes-dashboard-amd64:v1.6.1

docker tag cloudnil/k8s-dns-sidecar-amd64:1.14.4 gcr.io/google\_containers/k8s-dns-sidecar-amd64:1.14.4

docker tag cloudnil/k8s-dns-kube-dns-amd64:1.14.4 gcr.io/google\_containers/k8s-dns-kube-dns-amd64:1.14.4

docker tag cloudnil/k8s-dns-dnsmasq-nanny-amd64:1.14.4

docker pull mirrorgooglecontainers/kubernetes-dashboard-amd64:v1.7.1

docker pull kubernetesdashboarddev/kubernetes-dashboard-init-amd64:v1.0.1

docker tag mirrorgooglecontainers/kubernetes-dashboard-amd64:v1.7.1 gcr.io/google\_containers/kubernetes-dashboard-amd64:v1.7.1

docker tag kubernetesdashboarddev/kubernetes-dashboard-init-amd64:v1.0.1 gcr.io/google\_containers/kubernetes-dashboard-init-amd64:v1.0.1

5、初始化集群

kubeadm init \

--kubernetes-version=v1.7.2 \

--pod-network-cidr=10.244.0.0/16 \

--apiserver-advertise-address=192.168.1.55

如果安装失败，需要重装时。可以使用如下命令来清理环境

kubeadm reset

因为我们选择flannel作为Pod网络插件，所以上面的命令指定–pod-network-cidr=10.244.0.0/16。

6、安装成功后安装网络

export KUBECONFIG=/etc/kubernetes/admin.conf

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/v0.8.0/Documentation/kube-flannel-rbac.yml

kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml

kubectl get pods --all-namespaces

kubeadm join --token 159f16.e32c9f7e14b39d0e 192.168.1.55:6443

kubectl get pods -n kube-system -o wide

kubectl proxy --address='192.168.1.55' --port=8888 --accept-hosts='^\*$'

创建redis

kubectl create -f <https://k8s.io/docs/tasks/access-application-cluster/redis-master.yaml>

minikube安装

检查主机

cat /proc/cpuinfo | grep 'vmx|svm'

安装VritualBOX

yum -y install vritualbox

安装kubelet

curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

chmod +x ./kubectl && mv ./kubectl /usr/local/bin/kubectl

安装minikube

curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.23.0/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/

启动minkube

minikube start --registry-mirror=https://registry.docker-cn.com