# **详细部署k8s集群，三台机器**



[Dismay十二](https://blog.csdn.net/weixin_37127253" \t "https://blog.csdn.net/weixin_37127253/article/details/_blank) 2020-05-18 14:50:42 IMG_257 1531 IMG_258 收藏 7 IMG_259 原力计划

分类专栏： [kubernetes](https://blog.csdn.net/weixin_37127253/category_10014614.html" \t "https://blog.csdn.net/weixin_37127253/article/details/_blank)

版权

# **一.预先环境搭建**

需要三台主机，我这里选用三台虚拟机，系统centos7，ip：192.168.0.100 101 102

## **1.更改三台主机hostname**

vi /etc/hostname

master01

vi /etc/hostname

node01

vi /etc/hostname

node02

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

## **2.修改三台主机hosts**

vi /etc/hosts

192.168.0.100 master01

192.168.0.101 node01

192.168.0.102 node02

更新yum

yum update

安装依赖包

yum install -y conntrack ipvsadm ipset jq sysstat curl iptables libseccomp

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

## **3.关闭防火墙等**

systemctl stop firewalld && systemctl disable firewalld

重置iptables

iptables -F && iptables -X && iptables -F -t nat && iptables -X -t nat && iptables -P FORWARD ACCEPT

关闭交换

swapoff -a

永久关闭

vi /etc/fstab

# 注释关于swap的那一行

#关闭selinux

setenforce 0

vi /etc/sysconfig/selinux

#将里面的SELINUX配置为disabled SELINUX=disabled

#同步时间

# 安装ntpdate

yum install ntpdate -y

# 添加定时任务

crontab -e

插入内容：

0-59/10 \* \* \* \* /usr/sbin/ntpdate us.pool.ntp.org | logger -t NTP

# 先手动同步一次

ntpdate us.pool.ntp.org

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30

## **4.安装Docker**

若已安装，查看版本是否一致，不一致卸了重装

卸载现有版本

yum remove -y docker\* container-selinux

删除容器镜像：

sudo rm -rf /var/lib/docker

安装依赖包

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

设置阿里云镜像源

sudo yum-config-manager --add-repo https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

安装 Docker-CE

sudo yum install docker-ce

开机自启

sudo systemctl enable docker

启动docker服务

sudo systemctl start docker

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20

## **5.做免密登陆（三台）**

[root@master ~]# ssh-keygen -t rsa

//生成密钥， 连续回车

* 1
* 2

复制密钥到其他主机

ssh-copy-id node01

ssh-copy-id node02

* 1
* 2

把域名解析复制到其他主机，分别改了的就不用了

scp /etc/hosts node01:/etc

scp /etc/hosts node02:/etc

* 1
* 2

## **6. 打开路由转发和iptables桥接功能（三台），两种方法建议二**

[root@master ~]# vim /etc/sysctl.d/k8s.conf

//开启iptables桥接功能

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

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

[root@master ~]# echo net.ipv4.ip\_forward = 1 >> /etc/sysctl.conf

//\*\*打开路由转发

[root@master ~]# sysctl -p /etc/sysctl.d/k8s.conf

[root@master ~]# sysctl -p

//刷新一下

或者（推荐）

#写入配置文件

$ cat <<EOF > /etc/sysctl.d/kubernetes.conf

vm.swappiness=0

vm.overcommit\_memory = 1

vm.panic\_on\_oom=0

net.ipv4.ip\_forward = 1

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

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

fs.inotify.max\_user\_watches=89100

EOF

#生效配置文件

$ sysctl -p /etc/sysctl.d/kubernetes.conf

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25

如果以上命令执行失败可能是缺少模块，可执行以下命令

[root@master ~]# modprobe br\_netfilter

* 1

把路由转发和iptables桥接复制到其他主机

[root@master ~]# scp /etc/sysctl.d/k8s.conf node01:/etc/sysctl.d/

[root@master ~]# scp /etc/sysctl.d/k8s.conf node02:/etc/sysctl.d/

[root@master ~]# scp /etc/sysctl.conf node02:/etc/

[root@master ~]# scp /etc/sysctl.conf node01:/etc/

* 1
* 2
* 3
* 4

记得node01和node02也要执行以下命令

[root@master ~]# sysctl -p /etc/sysctl.d/k8s.conf

[root@master ~]# sysctl -p

* 1
* 2

# **二. 安装部署k8s**

· kubeadm:部署集群用的命令

· kubelet：在集群中每台机器上都要运行的组件，负责管理pod、容器的生命周期

· kubectl：集群管理工具

（1）指定yum安装kubernetes的yum源（三台）

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

[kubernetes]

name=Kubernetes

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

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

EOF

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

下载完成之后，查看一下仓库是否可用

[root@master ~]# yum repolist

* 1

创建本地缓存（三台）

[root@master ~]# yum makecache fast

* 1

（2）各节点安装所需安装包

1.三台主机下载版本号自行更改

master执行

[root@master ~]# yum -y install kubeadm-1.15.0-0 kubelet-1.15.0-0 kubectl-1.15.0-0

node执行

[root@node01 ~]# yum -y install kubeadm-1.15.0-0 kubelet-1.15.0-0

* 1
* 2
* 3
* 4
* 5
* 6

2.三台主机把 kubelet加入开机自启

systemctl enable kubelet

* 1

3.节点创建配置文件夹（所有节点）后面粘贴

mkdir -p /home/glory/working

cd /home/glory/working/

* 1
* 2

4.生成配置文件

kubeadm config print init-defaults ClusterConfiguration > kubeadm.conf

* 1

5.修改 kubeadm.conf 中的如下两项:

imageRepository

kubernetesVersion

vi kubeadm.conf

# 修改 imageRepository: k8s.gcr.io

# 改为 registry.cn-hangzhou.aliyuncs.com/google\_containers

imageRepository: registry.cn-hangzhou.aliyuncs.com/google\_containers

# 修改kubernetes版本kubernetesVersion: v1.13.0

# 改为kubernetesVersion: v1.15.0

kubernetesVersion: v1.15.0

修改 kubeadm.conf 中的API服务器地址，后面会频繁使用这个

地址。serverapi的地址

localAPIEndpoint:

localAPIEndpoint:

advertiseAddress: 192.168.0.100

bindPort: 6443

注意: 192.168.0.100 是master主机的ip地址

配置子网网络

networking:

dnsDomain: cluster.local

podSubnet: 10.244.0.0/16

serviceSubnet: 10.96.0.0/12

scheduler: {}

这里的 10.244.0.0/16 和 10.96.0.0/12 分别是k8s内部pods和services的子网网络，最好使用这个地址，后续flannel网络需要用到。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26

6.查看一下都需要哪些镜像文件需要拉取

kubeadm config images list --config kubeadm.conf

$ kubeadm config images list --config kubeadm.conf

registry.cn-beijing.aliyuncs.com/imcto/kube-

apiserver:v1.15.0

registry.cn-beijing.aliyuncs.com/imcto/kube-

controller-manager:v1.13.1

registry.cn-beijing.aliyuncs.com/imcto/kube-

scheduler:v1.13.1

registry.cn-beijing.aliyuncs.com/imcto/kube-

proxy:v1.13.1

registry.cn-beijing.aliyuncs.com/imcto/pause:3.1

registry.cn-beijing.aliyuncs.com/imcto/etcd:3.2.24

registry.cn-beijing.aliyuncs.com/imcto/coredns:1.2.6

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15

7.拉取镜像

kubeadm config images pull --config ./kubeadm.conf

* 1

8.初始化并且启动

初始化

sudo kubeadm init --config ./kubeadm.conf

若报错根据报错信息对应修改

[error]比如什么cpu个数，swap未关闭，bridge-nf-call-iptables 这个参数，需要设置为 1：

改好重新执行

#iptable设置为一的方法

echo "1" >/proc/sys/net/bridge/bridge-nf-call-iptables

modprobe br\_netfilter

成功后记下最后的token，这个很重要，很重要，很重要。要复制出来。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13

9.更更多kubeadm配置文件参数详见（没啥用）

kubeadm config print-defaults

* 1

10.k8s启动成功输出内容较多，但是记住末尾的内容

末尾token命令复制出来，这种样子的

kubeadm join 172.24.207.115:6443 --token abcdef.0123456789abcdef --discovery-token-ca-cert-hash sha256:cf7ed04f59030a449f2f50c1db6640cbce33770e70aee7af2b7824167a97bad4

kubeadm join 192.168.0.100:6443 --token abcdef.0123456789abcdef \

--discovery-token-ca-cert-hash sha256:89a4e7ac6f2a4722dfe17b70fa6ab06da4a6672ea7017857ef79d50826942b38

* 1
* 2
* 3
* 4

11.按照官方提示，执行以下操作。刚才初始化里面有这个提示

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

* 1
* 2
* 3

12.创建系统服务并启动

# 启动kubelet 设置为开机⾃自启动

$ sudo systemctl enable kubelet

# 启动k8s服务程序

$ sudo systemctl start kubelet

* 1
* 2
* 3
* 4

13.验证输入，注意显示master状态是 NotReady ，证明初始化服务器器成功

kubectl get nodes

NAME STATUS ROLES AGE VERSION

master NotReady master 12m v1.13.1

* 1
* 2
* 3
* 4

14.查看当前k8s集群状态

$ kubectl get cs

NAME STATUS MESSAGE ERROR

scheduler Healthy ok

controller-manager Healthy ok

etcd-0 Healthy {"health": "true"}

* 1
* 2
* 3
* 4
* 5

目前只有一个master，还没有node，而且是NotReady状态，那么我们需要将node加⼊入到master管理理的集群中来。在加入之前，我们需要先配置k8s集群的内部通信网络，这里采用的是flannel网络。

1. 部署集群内部通信flannel网络

cd /home/glory/working

wget https://raw.githubusercontent.com/coreos/flannel/a70459be0084506e4ec919aa1c114638878db11b/Documentation/kube-flannel.yml

#注意：如果下不了，就自己创建一个kube-flannel.yml内容如下

## 因yaml格式要求严格，未确保，先建立txt，保存内容后再改yml

##我建议，不下载，直接复制我的，改名字，不然可能出现镜像源无法下载问题

vi kube-flannel.txt

mv kube-flannel.txt kube-flannel.yml

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

---

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1beta1

metadata:

name: flannel

rules:

- apiGroups:

- ""

resources:

- pods

verbs:

- get

- apiGroups:

- ""

resources:

- nodes

verbs:

- list

- watch

- apiGroups:

- ""

resources:

- nodes/status

verbs:

- patch

---

kind: ClusterRoleBinding

apiVersion: rbac.authorization.k8s.io/v1beta1

metadata:

name: flannel

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: flannel

subjects:

- kind: ServiceAccount

name: flannel

namespace: kube-system

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: flannel

namespace: kube-system

---

kind: ConfigMap

apiVersion: v1

metadata:

name: kube-flannel-cfg

namespace: kube-system

labels:

tier: node

app: flannel

data:

cni-conf.json: |

{

"name": "cbr0",

"plugins": [

{

"type": "flannel",

"delegate": {

"hairpinMode": true,

"isDefaultGateway": true

}

},

{

"type": "portmap",

"capabilities": {

"portMappings": true

}

}

]

}

net-conf.json: |

{

"Network": "10.244.0.0/16",

"Backend": {

"Type": "vxlan"

}

}

---

apiVersion: extensions/v1beta1

kind: DaemonSet

metadata:

name: kube-flannel-ds-amd64

namespace: kube-system

labels:

tier: node

app: flannel

spec:

template:

metadata:

labels:

tier: node

app: flannel

spec:

hostNetwork: true

nodeSelector:

beta.kubernetes.io/arch: amd64

tolerations:

- operator: Exists

effect: NoSchedule

serviceAccountName: flannel

initContainers:

- name: install-cni

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-amd64

command:

- cp

args:

- -f

- /etc/kube-flannel/cni-conf.json

- /etc/cni/net.d/10-flannel.conflist

volumeMounts:

- name: cni

mountPath: /etc/cni/net.d

- name: flannel-cfg

mountPath: /etc/kube-flannel/

containers:

- name: kube-flannel

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-amd64

command:

- /opt/bin/flanneld

args:

- --ip-masq

- --kube-subnet-mgr

resources:

requests:

cpu: "100m"

memory: "50Mi"

limits:

cpu: "100m"

memory: "50Mi"

securityContext:

privileged: true

env:

- name: POD\_NAME

valueFrom:

fieldRef:

fieldPath: metadata.name

- name: POD\_NAMESPACE

valueFrom:

fieldRef:

fieldPath: metadata.namespace

volumeMounts:

- name: run

mountPath: /run

- name: flannel-cfg

mountPath: /etc/kube-flannel/

volumes:

- name: run

hostPath:

path: /run

- name: cni

hostPath:

path: /etc/cni/net.d

- name: flannel-cfg

configMap:

name: kube-flannel-cfg

---

apiVersion: extensions/v1beta1

kind: DaemonSet

metadata:

name: kube-flannel-ds-arm64

namespace: kube-system

labels:

tier: node

app: flannel

spec:

template:

metadata:

labels:

tier: node

app: flannel

spec:

hostNetwork: true

nodeSelector:

beta.kubernetes.io/arch: arm64

tolerations:

- operator: Exists

effect: NoSchedule

serviceAccountName: flannel

initContainers:

- name: install-cni

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-arm64

command:

- cp

args:

- -f

- /etc/kube-flannel/cni-conf.json

- /etc/cni/net.d/10-flannel.conflist

volumeMounts:

- name: cni

mountPath: /etc/cni/net.d

- name: flannel-cfg

mountPath: /etc/kube-flannel/

containers:

- name: kube-flannel

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-arm64

command:

- /opt/bin/flanneld

args:

- --ip-masq

- --kube-subnet-mgr

resources:

requests:

cpu: "100m"

memory: "50Mi"

limits:

cpu: "100m"

memory: "50Mi"

securityContext:

privileged: true

env:

- name: POD\_NAME

valueFrom:

fieldRef:

fieldPath: metadata.name

- name: POD\_NAMESPACE

valueFrom:

fieldRef:

fieldPath: metadata.namespace

volumeMounts:

- name: run

mountPath: /run

- name: flannel-cfg

mountPath: /etc/kube-flannel/

volumes:

- name: run

hostPath:

path: /run

- name: cni

hostPath:

path: /etc/cni/net.d

- name: flannel-cfg

configMap:

name: kube-flannel-cfg

---

apiVersion: extensions/v1beta1

kind: DaemonSet

metadata:

name: kube-flannel-ds-arm

namespace: kube-system

labels:

tier: node

app: flannel

spec:

template:

metadata:

labels:

tier: node

app: flannel

spec:

hostNetwork: true

nodeSelector:

beta.kubernetes.io/arch: arm

tolerations:

- operator: Exists

effect: NoSchedule

serviceAccountName: flannel

initContainers:

- name: install-cni

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-arm

command:

- cp

args:

- -f

- /etc/kube-flannel/cni-conf.json

- /etc/cni/net.d/10-flannel.conflist

volumeMounts:

- name: cni

mountPath: /etc/cni/net.d

- name: flannel-cfg

mountPath: /etc/kube-flannel/

containers:

- name: kube-flannel

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-arm

command:

- /opt/bin/flanneld

args:

- --ip-masq

- --kube-subnet-mgr

resources:

requests:

cpu: "100m"

memory: "50Mi"

limits:

cpu: "100m"

memory: "50Mi"

securityContext:

privileged: true

env:

- name: POD\_NAME

valueFrom:

fieldRef:

fieldPath: metadata.name

- name: POD\_NAMESPACE

valueFrom:

fieldRef:

fieldPath: metadata.namespace

volumeMounts:

- name: run

mountPath: /run

- name: flannel-cfg

mountPath: /etc/kube-flannel/

volumes:

- name: run

hostPath:

path: /run

- name: cni

hostPath:

path: /etc/cni/net.d

- name: flannel-cfg

configMap:

name: kube-flannel-cfg

---

apiVersion: extensions/v1beta1

kind: DaemonSet

metadata:

name: kube-flannel-ds-ppc64le

namespace: kube-system

labels:

tier: node

app: flannel

spec:

template:

metadata:

labels:

tier: node

app: flannel

spec:

hostNetwork: true

nodeSelector:

beta.kubernetes.io/arch: ppc64le

tolerations:

- operator: Exists

effect: NoSchedule

serviceAccountName: flannel

initContainers:

- name: install-cni

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-ppc64le

command:

- cp

args:

- -f

- /etc/kube-flannel/cni-conf.json

- /etc/cni/net.d/10-flannel.conflist

volumeMounts:

- name: cni

mountPath: /etc/cni/net.d

- name: flannel-cfg

mountPath: /etc/kube-flannel/

containers:

- name: kube-flannel

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-ppc64le

command:

- /opt/bin/flanneld

args:

- --ip-masq

- --kube-subnet-mgr

resources:

requests:

cpu: "100m"

memory: "50Mi"

limits:

cpu: "100m"

memory: "50Mi"

securityContext:

privileged: true

env:

- name: POD\_NAME

valueFrom:

fieldRef:

fieldPath: metadata.name

- name: POD\_NAMESPACE

valueFrom:

fieldRef:

fieldPath: metadata.namespace

volumeMounts:

- name: run

mountPath: /run

- name: flannel-cfg

mountPath: /etc/kube-flannel/

volumes:

- name: run

hostPath:

path: /run

- name: cni

hostPath:

path: /etc/cni/net.d

- name: flannel-cfg

configMap:

name: kube-flannel-cfg

---

apiVersion: extensions/v1beta1

kind: DaemonSet

metadata:

name: kube-flannel-ds-s390x

namespace: kube-system

labels:

tier: node

app: flannel

spec:

template:

metadata:

labels:

tier: node

app: flannel

spec:

hostNetwork: true

nodeSelector:

beta.kubernetes.io/arch: s390x

tolerations:

- operator: Exists

effect: NoSchedule

serviceAccountName: flannel

initContainers:

- name: install-cni

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-s390x

command:

- cp

args:

- -f

- /etc/kube-flannel/cni-conf.json

- /etc/cni/net.d/10-flannel.conflist

volumeMounts:

- name: cni

mountPath: /etc/cni/net.d

- name: flannel-cfg

mountPath: /etc/kube-flannel/

containers:

- name: kube-flannel

image: quay-mirror.qiniu.com/coreos/flannel:v0.11.0-s390x

command:

- /opt/bin/flanneld

args:

- --ip-masq

- --kube-subnet-mgr

resources:

requests:

cpu: "100m"

memory: "50Mi"

limits:

cpu: "100m"

memory: "50Mi"

securityContext:

privileged: true

env:

- name: POD\_NAME

valueFrom:

fieldRef:

fieldPath: metadata.name

- name: POD\_NAMESPACE

valueFrom:

fieldRef:

fieldPath: metadata.namespace

volumeMounts:

- name: run

mountPath: /run

- name: flannel-cfg

mountPath: /etc/kube-flannel/

volumes:

- name: run

hostPath:

path: /run

- name: cni

hostPath:

path: /etc/cni/net.d

- name: flannel-cfg

configMap:

name: kube-flannel-cfg

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90
* 91
* 92
* 93
* 94
* 95
* 96
* 97
* 98
* 99
* 100
* 101
* 102
* 103
* 104
* 105
* 106
* 107
* 108
* 109
* 110
* 111
* 112
* 113
* 114
* 115
* 116
* 117
* 118
* 119
* 120
* 121
* 122
* 123
* 124
* 125
* 126
* 127
* 128
* 129
* 130
* 131
* 132
* 133
* 134
* 135
* 136
* 137
* 138
* 139
* 140
* 141
* 142
* 143
* 144
* 145
* 146
* 147
* 148
* 149
* 150
* 151
* 152
* 153
* 154
* 155
* 156
* 157
* 158
* 159
* 160
* 161
* 162
* 163
* 164
* 165
* 166
* 167
* 168
* 169
* 170
* 171
* 172
* 173
* 174
* 175
* 176
* 177
* 178
* 179
* 180
* 181
* 182
* 183
* 184
* 185
* 186
* 187
* 188
* 189
* 190
* 191
* 192
* 193
* 194
* 195
* 196
* 197
* 198
* 199
* 200
* 201
* 202
* 203
* 204
* 205
* 206
* 207
* 208
* 209
* 210
* 211
* 212
* 213
* 214
* 215
* 216
* 217
* 218
* 219
* 220
* 221
* 222
* 223
* 224
* 225
* 226
* 227
* 228
* 229
* 230
* 231
* 232
* 233
* 234
* 235
* 236
* 237
* 238
* 239
* 240
* 241
* 242
* 243
* 244
* 245
* 246
* 247
* 248
* 249
* 250
* 251
* 252
* 253
* 254
* 255
* 256
* 257
* 258
* 259
* 260
* 261
* 262
* 263
* 264
* 265
* 266
* 267
* 268
* 269
* 270
* 271
* 272
* 273
* 274
* 275
* 276
* 277
* 278
* 279
* 280
* 281
* 282
* 283
* 284
* 285
* 286
* 287
* 288
* 289
* 290
* 291
* 292
* 293
* 294
* 295
* 296
* 297
* 298
* 299
* 300
* 301
* 302
* 303
* 304
* 305
* 306
* 307
* 308
* 309
* 310
* 311
* 312
* 313
* 314
* 315
* 316
* 317
* 318
* 319
* 320
* 321
* 322
* 323
* 324
* 325
* 326
* 327
* 328
* 329
* 330
* 331
* 332
* 333
* 334
* 335
* 336
* 337
* 338
* 339
* 340
* 341
* 342
* 343
* 344
* 345
* 346
* 347
* 348
* 349
* 350
* 351
* 352
* 353
* 354
* 355
* 356
* 357
* 358
* 359
* 360
* 361
* 362
* 363
* 364
* 365
* 366
* 367
* 368
* 369
* 370
* 371
* 372
* 373
* 374
* 375
* 376
* 377
* 378
* 379
* 380
* 381
* 382
* 383
* 384
* 385
* 386
* 387
* 388
* 389
* 390
* 391
* 392
* 393
* 394
* 395
* 396
* 397
* 398
* 399
* 400
* 401
* 402
* 403
* 404
* 405
* 406
* 407
* 408
* 409
* 410
* 411
* 412
* 413
* 414
* 415
* 416
* 417
* 418
* 419
* 420
* 421
* 422
* 423
* 424
* 425
* 426
* 427
* 428
* 429
* 430
* 431
* 432
* 433
* 434
* 435
* 436
* 437
* 438
* 439
* 440
* 441
* 442
* 443
* 444
* 445
* 446
* 447
* 448
* 449
* 450
* 451
* 452
* 453
* 454
* 455
* 456
* 457
* 458
* 459
* 460
* 461
* 462
* 463
* 464
* 465
* 466
* 467
* 468
* 469
* 470

注意：国内环境应将image 中quay.io 替换为 quay-mirror.qiniu.com，这是七牛的。若挂了

参照这篇镜像源声明：https://www.cnblogs.com/ants/archive/2020/04/09/12663724.html

若是之前忘了改，直接执行了下面，发现又下不来，只能手动下载，复制的我的，直接16.

docker pull quay-mirror.qiniu.com/coreos/flannel:v0.11.0-amd64

#你上面文件没改，就要把这镜像标签改回quay.io

docker tag quay-mirror.qiniu.com/coreos/flannel:v0.11.0-amd64 quay.io/coreos/flannel:v0.11.0-amd64

* 1
* 2
* 3

16.开启flannel服务

kubectl apply -f kube-flannel.yml

过程比较久需要下镜像

* 1
* 2
* 3
* 4

17.master准备成功

notready变为ready就好了

查看镜像下载情况

kubectl get pods --all-namespaces -o wide

root@master01:/home/itcast/working# kubectl get nodes

NAME STATUS ROLES AGE VERSION

master01 Ready master 6m58s v1.13.1

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

18.配置k8s集群的Node主机环境

19.启动k8s后台服务

*#启动kubelet 设置为开机⾃自启动*

$ sudo systemctl enable kubelet*#启动k8s服务程序*

$ sudo systemctl start kubelet

* 1
* 2
* 3
* 4

20.将master机器器的 /etc/kubernetes/admin.conf 传到到node1和node2

#将admin.conf传递给node1

sudo scp /etc/kubernetes/admin.conf root@192.168.0.101:/home/glory/

#将admin.conf传递给node2

sudo scp /etc/kubernetes/admin.conf root@192.168.0.102:/home/glory/

* 1
* 2
* 3
* 4

21.node执行以下命令

mkdir -p $HOME/.kube

sudo cp -i admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

* 1
* 2
* 3

22.node节点执行

之前复制的token那条命令

kubeadm join 172.17.93.196:6443 --token abcdef.0123456789abcdef --discovery-token-ca-cert-hash sha256:4c38f97d88b8ab34fae182b3387e260b12807511094a13d9884337169a750eba

* 1

23.应用两个node主机分别应用flannel网络

将 master 中的 kube-flannel.yml 分别传递给两个 node 节点.

0#将kube-flannel.yml传递给node1

sudo scp $HOME/working/kube-flannel.yml root@192.168.0.101:/home/glory/

#将kube-flannel.yml传递给node2

sudo scp $HOME/working/kube-flannel.yml root@192.168.0.102:/home/glory/

分别启动 flannel ⽹网络

root@node1:~$ kubectl apply -f kube-flannel.yml

root@node2:~$ kubectl apply -f kube-flannel.yml

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

24.查看node是否已经加入到k8s集群中(需要等一段时间才能ready)

kubectl get pods --all-namespaces -o wide

glory@node2:~$ kubectl get nodes

NAME STATUS ROLES AGE VERSION

master Ready master 35m v1.13.1

node1 Ready <none> 2m23s v1.13.1

node2 Ready <none> 40s v1.13.1

* 1
* 2
* 3
* 4
* 5
* 6

参考文献：  
https://blog.51cto.com/14320361/2463790?source=drh