k8s-1、Docker + kubernetes

笔记本:	<Inbox>		
创建时间:	2020/4/2 15:22	**更新时间:**	2020/4/3 13:46
作者:	王鹏		
URL:	https://www.cnblogs.com/weiBlog/p/10503779.html		

本次搭建Docker + kubernetes前提条件是在VMware + CentOS 7 上面搭建的;
我们需要做一下的工作:
（1）安装VMware ，运行CentOS 7系统。
（2）本次是在同一个虚拟机上运行，只搭建一个Master
(3) 安装docker + K8S

一、安装VMware

官网地址：https://my.vmware.com/zh/web/vmware/downloads （网上有很多自带破解的下载，大家也可以找一下）

我这里下载的是VM15（安装和新建虚拟机的步骤就跳过了，我也是百度的）

虚拟机配置：4核4G（也可以减小配置）

CentOs下载地址：https://www.centos.org/download/

具体百度安装VMware。

二、准备工作

VMware启动后，设置登录账号为root ， 密码 （自己设置）

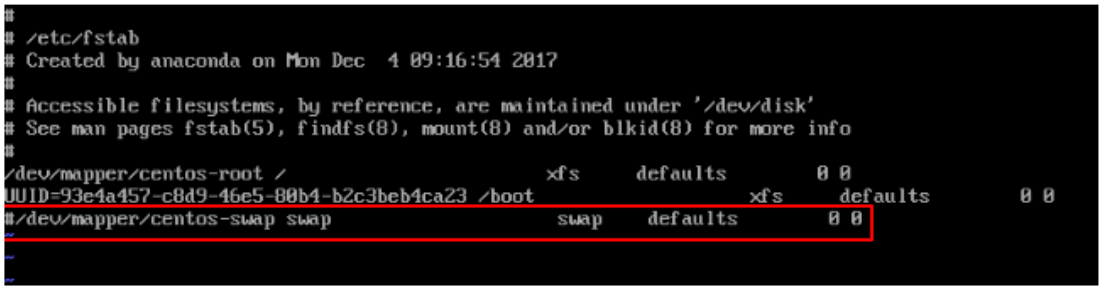
1、关闭防火墙，并且不启动防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

2、关闭swap

```
vi /etc/fstab

在行首加 #，注释该行
```



3、设置主机名称位master

```
hostnamectl set-hostname master
```

4、重新启动虚拟机

二、安装Docker

(1) 下载安装

```
# step 1: 安装必要的一些系统工具

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

# Step 2: 添加软件源信息

sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
如果失败用下方的安装
sudo yum-config-manager --add-repo https://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo

# Step 3: 更新并安装 Docker-CE
sudo yum makecache fast
sudo yum -y install docker-ce
sudo service docker start

systemctl enable docker
systemctl start docker
```

查看docker版本：

```
docker -v
```

（2）配置docker加速器

加速器怎么配置就不说了，这里是我的阿里云加速地址，大家可以自行替换，然后我们还需要指定下k8s的下载地址：

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://bbw0jgk7.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
cat>>/etc/yum.repos.d/kubrenetes.repo<<EOF
[kubernetes]
name=Kubernetes Repo
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64/
gpgcheck=0
gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
EOF
```

（3）安装：

```
yum install -y kubeadm  kubelet kubectl
```

（4）设置开机启动

```
systemctl start docker.service
systemctl enable docker.service
systemctl enable kubelet.service
```

（5）初始化k8s，先试一下，展示一下k8s的版本。

```
kubeadm init --pod-network-cidr=10.244.0.0/16
--service-cidr=10.96.0.0/12
```

```
the config file specified by the Kubelet's --config flag. See https://kubernetes
.io/docs/tasks/administer-cluster/kubelet-config-file/ for more information.)
=0402 04:00:34.282690   47907 server.go:157] invalid argument "ersion" for "-v,
--v" flag: strconv.ParseInt: parsing "ersion": invalid syntax
[root@master arthur]# kubeadm init --pod-network-cidr=10.244.0.0/16 --service-ci
dr=10.96.0.0/12
W0402 04:02:20.151053   47996 configset.go:202] WARNING: kubeadm cannot validate
 component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io

[init] Using Kubernetes version: v1.18.0
[preflight] Running pre-flight checks
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup
 driver. The recommended driver is "systemd". Please follow the guide at https://
/kubernetes.io/docs/setup/cri/
        [WARNING Hostname]: hostname "master" could not be reached
        [WARNING Hostname]: hostname "master": lookup master on 192.168.40.2:53:
 server misbehaving
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your inte
rnet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config
 images pull'
^C
[root@master arthur]#
```

初始化相应的版本：

```
kubeadm init --image-repository registry.aliyuncs.com/google_containers --pod-
network-cidr=10.244.0.0/16 --ignore-preflight-errors=cri --kubernetes-
version=1.18.0
```

(6)需要做两个配置：

```
vim /etc/sysconfig/kubelet
```

```
#加入以下参数
KUBELET_EXTRA_ARGS="--fail-swap-on=false"
```

```
第（5）步失败的话，试试执行下方是否会失败。
kubeadm init --pod-network-cidr=10.244.0.0/16 --service-cidr=10.96.0.0/12 --
ignore-preflight-errors=Swap

#在初始化时加入--ignore 选项
```

如何失败，查询kubeadm 配置镜像依赖的版本号：

```
kubeadm config images list
```

拉取镜像命令：共计7个服务

```
docker pull mirrorgooglecontainers/kube-apiserver-amd64:v1.18.0
docker pull mirrorgooglecontainers/kube-controller-manager-amd64:v1.18.0
docker pull mirrorgooglecontainers/kube-scheduler-amd64:v1.18.0
docker pull mirrorgooglecontainers/kube-proxy-amd64:v1.18.0
docker pull mirrorgooglecontainers/pause:3.2
docker pull mirrorgooglecontainers/etcd-amd64:3.4.3-0
docker pull coredns/coredns:1.6.7
```

或者从阿里云拉取镜像：共计7个服务

```
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-apiserver-
amd64:v1.18.0

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-controller-
manager-amd64:v1.18.0

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-scheduler-
amd64:v1.18.0

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/etcd-amd64:3.4.3-
0

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-proxy-
amd64:v1.18.0

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.2
docker pull coredns/coredns:1.6.7
```

查询镜像文件：

```
docker images
```

将下载下来的镜像文件修改成k8s前缀的名字，通过dockers tag 命令来修改镜像的标签，命令如下

```
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-proxy-
amd64:v1.18.0 k8s.gcr.io/kube-proxy:v1.18.0
```

```
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-scheduler-
amd64:v1.18.0 k8s.gcr.io/kube-scheduler:v1.18.0

docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-apiserver-
amd64:v1.18.0 k8s.gcr.io/kube-apiserver:v1.18.0

docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-controller-
manager-amd64:v1.18.0 k8s.gcr.io/kube-controller-manager:v1.18.0

docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/etcd-amd64:3.4.3-0
k8s.gcr.io/etcd:3.4.3-0

docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.2
k8s.gcr.io/pause:3.2

docker tag coredns/coredns:1.6.7 k8s.gcr.io/coredns:1.6.7
```

将之前下载下来的镜像删除

```
docker rmi registry.aliyuncs.com/google_containers/kube-proxy:v1.18.0
docker rmi registry.aliyuncs.com/google_containers/kube-apiserver:v1.18.0
docker rmi registry.aliyuncs.com/google_containers/kube-controller-
manager:v1.18.0
docker rmi registry.aliyuncs.com/google_containers/kube-scheduler:v1.18.0
docker rmi registry.aliyuncs.com/google_containers/etcd:3.4.3-0
docker rmi registry.aliyuncs.com/google_containers/pause:3.2
docker rmi coredns/coredns:1.6.7
```

（7）初始化出错后，先回退初始化，命令：

```
kubeadm reset
```

再进行初始化操作：

```
kubeadm init --pod-network-cidr=10.244.0.0/16 -service-cidr=10.96.0.0/12 --
ignore-preflight-errors=Swap
```

获取主节点的代码。其它节点加入此集群时使用：

```
kubeadm join 192.168.40.131:6443 --token zznd5f.neo3a9h2mycb2gh7 --discovery-
token-ca-cert-hash
sha256:e948084a4eca206c3f1bb74b91acb02a3d17035b5f222b8dba901328fe56d312
```

配置下变量
建议用普通用户运行以下三个命令：

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

建议root用户执行下面命令：

```
export KUBECONFIG=/etc/kubernetes/admin.conf
也可以直接放到~/.bash_profile
echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bash_profile
```

（8）重启

```
docker restart $(docker ps -a -q)
```

到这一步我们已经可以用以下命令查看集群信息：

```
kubectl get node
```

```
master   NotReady   master   4m52s   v1.18.0
[root@master arthur]# kubectl describe master
error: the server doesn't have a resource type "master"
[root@master arthur]# kubectl get node
NAME     STATUS     ROLES    AGE     VERSION
master   NotReady   master   6m45s   v1.18.0
[root@master arthur]# kubectl get node
NAME     STATUS     ROLES    AGE   VERSION
master   NotReady   master   26m   v1.18.0
[root@master arthur]#
```

这个时候只有一个master节点，而且Status是NotReady状态，这是因为没有安装网络组件的原因，我们需要安装下