



链滴

kubernetes1.8 kubeadm 安装

作者: [liuxin](#)

原文链接: <https://ld246.com/article/1511751833956>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

准备四台centos7.0 物理机/虚拟机

角色(hostname) PU	IP	系统名称	
k8s-master VIDIA TESLA M10 X 1	100.64.0.114	centos 7.0	
k8s-node1 VIDIA TESLA M10 X 1	100.64.0.115	centos 7.0	
k8s-node2 VIDIA TESLA M10 X 1	100.64.0.116	centos 7.0	
翻墙	----	centos 7.0	无

每个节点需要设置hosts文件

```
[root@k8s-master ~]# cat /etc/hosts
100.64.0.114 k8s-master
100.64.0.115 k8s-node1
100.64.0.116 k8s-node2
```

每个节点需要关闭swap。

Kubernetes 1.8开始要求关闭系统的Swap，如果不关闭，默认配置下kubelet将无法启动。可以通过ubelet的启动参数`-fail-swap-on=false`更改这个限制。我们这里关闭系统的Swap:

```
[root@k8s-master ~]# swapoff -a
[root@k8s-master ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Nov 23 05:59:19 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=cfc379a-dfd9-49cb-b003-dcdc8eb8d674 /boot xfs defaults 0 0
#/dev/mapper/centos-swap swap swap defaults 0 0
/dev/xvdb1 /var/lib/docker ext3 defaults 1 2
```

每个节点需要关闭防火墙

```
[root@k8s-master ~]# ufw dsiable
```

翻墙机器下载kubernetes1相关安装包(4个)

```
[root@i-tp9112lc ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
```

```

name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
[root@i-tp9112lc ~]# yumdownloader kubelet kubeadm kubectl kubernetes
[root@i-tp9112lc ~]# ls
1acca81eb5cf99453f30466876ff03146112b7f12c625cb48f12508684e02665-kubelet-1.8.4-0.x86_64.rpm
a9db28728641ddb7f025b8b496804d82a396d0ccb178fffd124623fb2f999ea-kubectl-1.8.4-0.x86_64.rpm
79f9ba89dbe7000e7dfeda9b119f711bb626fe2c2d56abeb35141142cda00342-kubernetes-cni-0.5.1-1.x86_64.rpm
aeaad1e283c54876b759a089f152228d7cd4c049f271125c23623995b8e7696-kubeadm-1.8.4-0.x86_64.rpm
[root@i-tp9112lc ~]#

```

将以上4个rpm包上传到各node上，执行如下命令(每个节点都需要执行)

```

#IPv4 iptables 链设置 CNI插件需要
[root@k8s-master ~]# systemctl net.bridge.bridge-nf-call-iptables=1
[root@k8s-master ~]# yum install -y ebtables socat
#装kubeadm kubectl kubelet
[root@k8s-master ~]# rpm -ivh *.rpm

```

下载必要镜像（每个节点需要保证本地有以下9个必要镜像）

默认kubernetes1安装它会需要到Google去下载很多组件，墙内无法直接下载，需要借助翻墙的机或者docker hub+github来做一次中转，具体的镜像列表如下：

来自官网的表格

Image Name	v1.7 release branch version
gcr.io/google_containers/kube-apiserver-\${ARCH}	v1.7.x
gcr.io/google_containers/kube-controller-manager-\${ARCH}	v1.8.x
gcr.io/google_containers/kube-scheduler-\${ARCH}	v1.7.x
gcr.io/google_containers/kube-proxy-\${ARCH}	v1.7.x
gcr.io/google_containers/etcd-\${ARCH}	3.0.17
gcr.io/google_containers/pause-\${ARCH}	3.0
gcr.io/google_containers/k8s-dns-sidecar-\${ARCH}	1.14.4
gcr.io/google_containers/k8s-dns-kube-dns-\${ARCH}	1.14.4

.14.4

gcr.io/google_containers/k8s-dns-dnsmasq-nanny-\${ARCH}
14.4 1.14.4

1

Here **v1.7.x** means the “latest patch release of the v1.7 branch” .

\${ARCH} can be one of: **amd64, arm, arm64, ppc64le** or **s390x**

经过测试发现1.8.4的版本对应的k8s-dns-dnsmasq-nanny、k8s-dns-kube-dns、k8s-dns-sidecar版本是1.4.5 (官网未来得及更新)

下载脚本如下:

我是将Google的镜像中转到docker hub

```
[root@i-tp9112lc ~]# cat getimg.sh
#!/bin/bash
set -o errexit
set -o nounset
set -o pipefail
```

```
KUBE_VERSION=v1.8.4
KUBE_PAUSE_VERSION=3.0
ETCD_VERSION=3.0.17
DNS_VERSION=1.14.5
```

```
GCR_URL=gcr.io/google_containers
ALIYUN_URL=lcasi
```

```
images=(kube-proxy-amd64:${KUBE_VERSION}
kube-scheduler-amd64:${KUBE_VERSION}
kube-controller-manager-amd64:${KUBE_VERSION}
#kube-apiserver-amd64:${KUBE_VERSION}
pause-amd64:${KUBE_PAUSE_VERSION}
etcd-amd64:${ETCD_VERSION}
k8s-dns-sidecar-amd64:${DNS_VERSION}
k8s-dns-kube-dns-amd64:${DNS_VERSION}
k8s-dns-dnsmasq-nanny-amd64:${DNS_VERSION})
```

```
for imageName in ${images[@]} ; do
  docker pull $GCR_URL/$imageName
  docker tag $GCR_URL/$imageName $ALIYUN_URL/$imageName
  docker push $ALIYUN_URL/$imageName
  docker rmi $ALIYUN_URL/$imageName
  docker rmi $GCR_URL/$imageName
done
[root@i-tp9112lc ~]#
```

初始化master(在master节点执行)

```
[root@k8s-master ~]# kubeadm init --kubernetes-version=v1.8.4 --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=100.64.0.114
```

记录下输出的一行关键命令，在其他node节点执行：

```
kubeadm join --token xxxxxxxxxxxxxxxx 100.64.0.114:6443 --discovery-token-ca-cert-hash sha56:xxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

顺利的话你将会看到：

```
[root@k8s-master ~]# kubectl get nodes
NAME          STATUS    ROLES    AGE    VERSION
k8s-master    Ready     master   2d     v1.8.4
k8s-node1     Ready     <none>   2d     v1.8.4
k8s-node2     Ready     <none>   2d     v1.8.4
[root@k8s-master ~]#
```