# Kubernetes安装文档

## 安装环境

我们使用4台centos服务器组成kubernetes集群，配置如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主机名 | ip | 配置 |  |  |
| master | 172.16.94.129 | 2核CPU 4G RAM |  |  |
| node-a | 172.16.94.130 | 2核CPU 4G RAM |  |  |
| node-b | 172.16.94.131 | 2核CPU 4G RAM |  |  |
| node-c | 172.16.94.132 | 2核CPU 4G RAM |  |  |

## 下载kubernetes

### 下载安装包

在github上下载最新的kubernetes安装包

https://github.com/kubernetes/kubernetes/releases

下载后解压，解压后运行kubernetes\cluster\get-kube-binaries.sh下载安装文件

这个脚本执行后，server目录中包含了kubernetes-server-linux-amd64.tar.gz压缩文件，解压缩这个文件，解压后得到的文件列表如下所示

|  |  |
| --- | --- |
| kube-apiserver | apiserver主程序 |
| kube-apiserver.docker\_tag | apiserver的docker tag |
| kube-apiserver.tar | apiserver的docker镜像 |
| kube-controller-manager | controller-manager主程序 |
| kube-controller-manager.docker\_tag | controller-manager的docker镜像 |
| kube-controller-manager.tar | controller-manager的镜像文件 |
| kubectl | 客户端命令行工具 |

## 安装master

kubernetes的服务可以通过直接运行二进制文件加上启动参数来完成，为了便于管理，通常我们将kubernetes配置为开机启动的服务。

关闭防火墙

systemctl stop firewalld

systemctl disable firewalld

关闭selinux

修改/etc/selinux/config

将SELINUX设置为disabled

### 安装etcd

etcd做为kubernetes集群的主数据库，安装kubernetes服务之前需要首先安装和启动。

centos7可以使用yum安装etcd

yum install etcd

或者通过二进制文件安装：

下载并解压etcd: <https://github.com/etcd-io/etcd/releases>

创建etc.service文件内容如下

[Unit]

Description=etcd

Documentation=https://github.com/coreos/etcd

[Service]

Type=notify

Restart=always

RestartSec=5s

LimitNOFILE=40000

TimeoutStartSec=0

ExecStart=/tmp/test-etcd/etcd --name s1 \

--data-dir /tmp/etcd/s1 \

--listen-client-urls http://localhost:2379 \

--advertise-client-urls http://localhost:2379 \

--listen-peer-urls http://localhost:2380 \

--initial-advertise-peer-urls http://localhost:2380 \

--initial-cluster s1=http://localhost:2380 \

--initial-cluster-token tkn \

--initial-cluster-state new

[Install]

WantedBy=multi-user.target

### 3.2 创建service account key

执行openssl命令创建所需的key

openssl genrsa -out serviceaccount.key

### 安装kube-apiserver

apiserver是kubernetes的所有API 命令入口，它为其他组件提供服务，接收api对象，并为其他组件提供基于REST的交互功能入口。

#### 3.3.1 创建配置文件

KUBE\_API\_ARGS="--etcd-servers=http://127.0.0.1:2379 \

--enable-admission-plugins=NamespaceLifecycle,LimitRanger,SecurityContextDeny,ServiceAccount,ResourceQuota \

--logtostderr=false \

--log-dir=/var/log/kubernetes \

--service-cluster-ip-range=169.169.0.0/16 \

--bind-address=0.0.0.0 \

--insecure-bind-address=0.0.0.0 \

--service-account-key-file=/home/yaoyuquan/serviceaccount.key \

--v=2"

|  |  |
| --- | --- |
| etcd\_servers | etcd的地址 |
| logtostderr | 向standard error中输出错误，默认为true |
| enable-admission-plugins | 默认启动的安全访问插件 |
| log-dir | 日志的目录 |
| insecure-port | http端口，默认是8080 |
| insecure-bind-address | Apiserver默认会开启两个端口一个安全的和一个非安全的端口，非安全的端口主要用于http访问apiserver |
| service-cluster-ip-range | 为service 分配集群ip的ip地址段，这个地址段不能和nodes和pods的ip地址发生重合 |
| v | 日志级别 |
| service-account-key-file | 这里配置3.2中生成的证书文件路径 |

关于kube-apiserver的详细配置可以参考https://kubernetes.io/docs/reference/command-line-tools-reference/kube-apiserver/

创建service文件kube-apiserver.service

[Unit]

Description=Kubernetes API Server

Documentation=https://github.com/kubernetes/kubernetes

After=etcd.service

Wants=etcd.service

[Service]

EnvironmentFile=/home/yaoyuquan/kubernetes-config/apiserver

ExecStart=/home/yaoyuquan/kubernetes-exec/kube-apiserver $KUBE\_API\_ARGS

Restart=on-failure

Type=notify

LimitNOFILE=65536

[Install]

WantedBy=multi-user.target

将这个文件拷贝到/etc/systemd/system下面，执行下面的命令

sudo systemctl daemon-reload

sudo systemctl start kube-apiserver.service

sudo systemctl enable kube-apiserver.service

### 3.4 安装kube-controller-manager

controller-manager是kubernetes中的控制单元，它通过apiserver获取集群当前状态，并通过一个循环调用控制集群状态。

目前kubernetes中常用的controller包括replication controller, endpoints controller, namespace controller, 和serviceaccounts controller。

创建配置文件

KUBE\_CONTROLLER\_MANAGER\_ARGS="--master=http://127.0.0.1:8080 \

--log-dir=/var/log/kubernetes \

--service-account-private-key-file=/home/yaoyuquan/serviceaccount.key \

--v=2"

|  |  |
| --- | --- |
| master | apiserver的地址 |
| service-account-private-key-file | 这里配置3.2中生成的证书文件路径 |

关于kube-controller-manager的详细配置可以参考

https://kubernetes.io/docs/reference/command-line-tools-reference/kube-controller-manager/

创建service文件kube-controller-manager.service

sudo systemctl daemon-reload

sudo systemctl start kube-controller-manager

sudo systemctl enable kube-controller-manager

### 3.5 安装kube-scheduler

创建配置文件

KUBE\_SCHEDULER\_ARGS="--master=http://127.0.0.1:8080 --log-dir=/var/log/kubernetes --v=2"

|  |  |
| --- | --- |
| master | apiserver的地址 |

关于kube-scheduler的详细配置可以参考

<https://kubernetes.io/docs/reference/command-line-tools-reference/kube-scheduler/>

创建service文件kube-scheduler.service

[Unit]

Description=Kubernetes Scheduler

Documentation=https://github.com/kubernetes/kubernetes

After=kube-apiserver.service

Requires=kube-apiserver.service

[Service]

EnvironmentFile=/home/yaoyuquan/kubernetes-config/scheduler

ExecStart=/home/yaoyuquan/kubernetes-exec/kube-scheduler $KUBE\_SCHEDULER\_ARGS

Restart=on-failure

LimitNOFILE=65536

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl start kube-scheduler

sudo systemctl enable kube-scheduler

### 3.6 创建kubeconfig文件

这个步骤所用到的命令请参考

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#config>

创建用户，用户名是admin，密码是1234

kubectl config set-credentials cluster-admin --username=admin --password=1234

创建集群，名称为test-cluster，server的地址是本机的apiserver的地址

kubectl config set-cluster test-cluster --server=http://172.16.94.129:8080 --insecure-skip-tls-verify=true

创建对应的context，名称为test-context

kubectl config set-context test-context --cluster=test-cluster --user=cluster-admin

应用当前context为test-context

kubectl config use-context test-context

下载/root/.kube/config文件，将这个文件拷贝到node服务器的对应目录中

至此，master已经安装完毕

## 安装node

### 修改hostname

sudo hostnamectl set-hostname hostname

### 安装docker

#### 5.1.1 配置yum仓库

sudo yum install -y yum-utils

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

#### 5.1.2 安装docker

sudo yum install docker-ce docker-ce-cli containerd.io

#### 5.1.3 启动docker

sudo systemctl start docker

配合docker开机启动

sudo systemctl enable docker

#### 5.1.4 将linux用户添加进docker用户组

docker程序需要监听linux socket端口，在Linux系统中，只有root用户才有这个权限，所以对于docker所有的命令都需要sudo。

我们可以将当前用户加入docker用户组来完成授权。

sudo groupadd docker

sudo usermod -aG docker $USER

#### 5.1.5 配置docker开机启动

sudo systemctl enable docker

### 4.2 关闭swap

kubernates环境需要关闭linux swap，否则kubelet将无法和apiserver通信。

swapoff -a

然后修改/etc/fstab 注释掉swap一行，然后重启

### 4.3 安装kubelet

创建service文件

[Unit]

Description=Kubernetes Kubelet Server

Documentation=https://github.com/kubernetes/kubernetes

After=docker.service

Wants=docker.service

[Service]

WorkingDirectory=/home/yaoyuquan/kubelet

EnvironmentFile=/home/yaoyuquan/kubernetes-config/kubelet

ExecStart=/home/yaoyuquan/kubernetes-exec/kubelet $KUBELET\_ARGS

Restart=on-failure

[Install]

WantedBy=multi-user.target

配置文件

KUBELET\_ARGS="--pod-infra-container-image=kubernetes/pause --kubeconfig=/home/yaoyuquan/.kube/config --logtostderr=false --log-dir=/var/log/kubernetes --v=2"

sudo systemctl daemon-reload

sudo systemctl start kubelet.service

sudo systemctl enable kubelet.service

### 安装kube-proxy

创建service文件

[Unit]

Description=Kubernetes Proxy-Server

Documentation=https://github.com/kubernetes/kubernetes

After=network.target

Requires=network.service

[Service]

EnvironmentFile=/home/yaoyuquan/kubernetes-config/kube-proxy

ExecStart=/home/yaoyuquan/kubernetes-exec/kube-proxy $KUBE\_PROXY\_ARGS

Restart=on-failure

LimitNOFILE=65536

[Install]

WantedBy=multi-user.target

配置文件

KUBE\_PROXY\_ARGS="--kubeconfig=/home/yaoyuquan/.kube/config --logtostderr=false --log-dir=/var/log/kubernetes --v=2"

sudo systemctl daemon-reload

sudo systemctl start kube-proxy

sudo systemctl enable kube-proxy