参考资料：

<https://kubernetes.io/docs/reference/access-authn-authz/bootstrap-tokens>

<https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet-tls-bootstrapping>

二进制搭建K8s详细步骤：

<https://mp.weixin.qq.com/s/VYtyTU9_Dw9M5oHtvRfseA>

|  |  |
| --- | --- |
| 角色 | IP |
| Master | 192.168.31.61 |
| Node1 | 192.168.31.62 |
| Node2 | 192.168.31.63 |
| Node3 | 192.168.31.64 |

## 1、准备新节点环境

拷贝已部署好的Node相关文件到新节点Node3：

scp -r /opt/kubernetes/ [root@192.168.31.64:/opt/](mailto:root@192.168.3.72:/opt/)

scp -r /usr/lib/systemd/system/{kubelet,kube-proxy}.service [root@192.168.31.64:/usr/lib/systemd/system](mailto:root@192.168.3.72:/usr/lib/systemd/system)

scp -r /opt/cni/ [root@192.168.31.64:/opt](mailto:root@192.168.3.72:/opt)

删除kubelet证书和kubeconfig文件：

cd /opt/kubernetes/ssl/

rm kubelet\* -f

cd /opt/kubernetes/cfg/

rm kubelet.kubeconfig bootstrap.kubeconfig -f

注：这几个文件是证书申请审批后自动生成的，每个Node不同，必须删除重新生成。

## 2、确认启用Bootstrap Token

# cat /opt/kubernetes/cfg/kube-apiserver.conf

…

--enable-bootstrap-token-auth=true

…

## 3、使用Secret存储Bootstrap Token

# vi bootstrap-secret.yaml

apiVersion: v1

kind: Secret

metadata:

# Name MUST be of form "bootstrap-token-<token id>"

name: bootstrap-token-07401b

namespace: kube-system

# Type MUST be 'bootstrap.kubernetes.io/token'

type: bootstrap.kubernetes.io/token

stringData:

# Human readable description. Optional.

description: "The default bootstrap token generated by 'kubeadm init'."

# Token ID and secret. Required.

token-id: 07401b

token-secret: f395accd246ae52d

# Expiration. Optional.

expiration: 2021-03-10T03:22:11Z

# Allowed usages.

usage-bootstrap-authentication: "true"

usage-bootstrap-signing: "true"

# kubectl apply -f bootstrap-secret.yaml

注：expiration 为token过期时间，当前时间向后推几天随意。

## 4、创建RBAC角色绑定，允许 kubelet tls bootstrap 创建 CSR 请求

# vi rbac.yaml

# enable bootstrapping nodes to create CSR

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: create-csrs-for-bootstrapping

subjects:

- kind: Group

name: system:bootstrappers

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: ClusterRole

name: system:node-bootstrapper

apiGroup: rbac.authorization.k8s.io

# kubectl apply -f rbac.yaml

## 5、kubelet配置Bootstrap kubeconfig文件

# vi /opt/kubernetes/cfg/bootstrap.kubeconfig

apiVersion: v1

kind: Config

clusters:

- cluster:

certificate-authority: /opt/kubernetes/ssl/ca.pem

server: https://192.168.31.61:6443

name: bootstrap

contexts:

- context:

cluster: bootstrap

user: kubelet-bootstrap

name: bootstrap

current-context: bootstrap

preferences: {}

users:

- name: kubelet-bootstrap

user:

token: 07401b.f395accd246ae52d

# scp bootstrap.kubeconfig root@192.168.31.64:/opt/kubernetes/cfg/

配置文件指定kubeconfig文件：

# cat /opt/kubernetes/cfg/kubelet.conf

KUBELET\_OPTS="--logtostderr=false \

--v=4 \

--log-dir=/opt/kubernetes/logs \

--hostname-override=k8s-node3 \

--kubeconfig=/opt/kubernetes/cfg/kubelet.kubeconfig \

--bootstrap-kubeconfig=/opt/kubernetes/cfg/bootstrap.kubeconfig \

**启动并设置开机启动：**

systemctl daemon-reload

systemctl start kubelet

systemctl enable kubelet

## 6、在Master节点颁发证书

kubectl get csr

kubectl certifificate approve xxx

**DevOps实战课堂：**[**www.ctnrs.com**](http://www.ctnrs.com)