

k3s安装&ks8安装&rancher 安装

• 安装

- o https://mp.weixin.qq.com/s?__biz=MzkyNzM4Nzk1NQ==&mid=2247500723&idx=1&sn=64def8cd6ec7fd874440690f1f589a3a
- o <http://kingsd.top/2020/07/01/k3s-cert-rotary/>
- o <https://www.cnblogs.com/databank/p/16334715.html>

安装环境要求

安装10个节点

安装kubeadm

<https://github.com/yuyicai/update-kube-cert>

```
# 安装kubeadm
kubeadm certs check-expiration

# 安装update-kubeadm-cert.sh
chmod 777 update-kubeadm-cert.sh

# 安装update-kubeadm-cert.sh
./update-kubeadm-cert.sh all

# 安装master etcd
./update-kubeadm-cert.sh master
```

rancher安装

```
# 安装rancher
curl -vvv https://ip:port

# 安装rancher
/data/vpclub/rancher-2.0/data/k3s/server/tls

# 安装rancher
```

```
# rancher-agent
kubectl get po -n cattle-system

# rancher-agent, rancher
docker ps -a | grep cattle

# cattle-agent
docker logs -f <cattle-agent-container-name>
```

-

```
# agent
kubectl get ns
kubectl -n cattle-system delete daemonset.apps/cattle-node-agent deployment.apps/cattle-cluster-agent

# server-url

# 
RANCHERURL="https://192.168.0.10:8000"

# ID
CLUSTERID="c-8d1c7"

# token, apiKeys Bearer Token
# Token
TOKEN="token-8kdfz:f4kgn4ptrq92wxkmlzjh9wntbxc4jpnnwq6spnv6mlhm76259wl6"

# jq
yum install -y epel-release
yum install -y jq

# 
curl -s -H "Authorization: Bearer ${TOKEN}"
"${RANCHERURL}/v3/clusterregistrationtokens?clusterId=${CLUSTERID}" | jq -r '.data[] |
select(.name != "system") | .command'
```

```
# 创建token

curl -s -k -H "Authorization: Bearer ${TOKEN}"
"${RANCHERURL}/v3/clusterregistrationtokens?clusterId=${CLUSTERID}" | jq -r '.data[] |
select(.name != "system") | .insecureCommand'

# 导入yaml
curl --insecure -sL
https://192.168.0.10:8000/v3/import/sztcbkgpcffdvmd8rn9qvd9flp595np5zc5s6xljqk89jqkdx8rhf.yaml
l | k3s kubectl apply -f -
```

• k3s 安装

安装k8s和k3s

```
# 安装k3s
# 安装k8s

k3s kubectl --insecure-skip-tls-verify=true delete secret k3s-serving -n kube-system

# 安装k3s

kubectl delete secret k3s-serving -n kube-system

# 安装k8s

mkdir -p tl5bak
cp /var/lib/rancher/k3s/server/tls/** ./tl5bak -rf
rm /var/lib/rancher/k3s/server/tls/dynamic-cert.json

# 安装k8s

openssl x509 -noout -dates -in server-ca.crt
```

k8s

--	--	--	--	--	--

```
mkdir -p /data/vpcub/kubernetes-bak/kubernetes-20240620
```

```
cp /etc/kubernetes/** /data/vpclub/kubernetes-bak/kubernetes-20240620 -r
```

```
# master
```

```
kubeadm certs check-expiration
```

[illegible][illegible][illegible]

```
kubeadm certs renew all
```

[] [] [] [] [] [] [] [] [] [] [] [] [] [] ETCD [] [] [] []

```
kubeadm certs renew apiserver
```

```
kubeadm certs renew apiserver-etcd-client
```

```
kubeadm certs renew apiserver-kubelet-client
```

```
kubeadm certs renew etcd-healthcheck-client
```

```
kubeadm certs renew etcd-peer
```

```
kubeadm certs renew etcd-server
```

```
kubeadm certs renew front-proxy-client
```

```
kubeadm certs renew scheduler.conf
```

```
kubeadm certs renew admin.conf
```

```
# master
```

```
# kubelet
```

```
systemctl restart kubelet
```

```
# docker
```

```
systemctl restart docker
```

4 POD

```
# etcd | kube-apiserver | kube-scheduler | kube-controller
```

```
master pki etcd tls etcd etcd tls pki,
```

```
# kubectl exec -i $(kubectl get pod -l component=kube-apiserver --output=NAME) -- /bin/sh

kubeadm init phase certs all

# kubectl exec -i $(kubectl get pod -l component=kube-apiserver --output=NAME) -- /bin/sh

mv etcd/ etcd2/

# kubectl exec -i $(kubectl get pod -l component=kube-apiserver --output=NAME) -- /bin/sh

scp -r /etc/kubernetes/pki/** 172.18.41.4:/etc/kubernetes/pki/

# kubectl exec -i $(kubectl get pod -l component=kube-apiserver --output=NAME) -- /bin/sh

cp etcd2/*peer.crt ./etcd/
cp etcd2/*server.crt ./etcd/

# kubectl exec -i $(kubectl get pod -l component=kube-apiserver --output=NAME) -- /bin/sh
```

• K3S

- [illegible]

```
# node[]

cat /etc/rancher/node/password

# control panel master[]

cat /var/lib/rancher/k3s/server/cred/node-passwd

# []

# []k3s.services[]token

# []token

cat /var/lib/rancher/k3s/server/node-token

# []noded[]token

sed -i "s/<old-token>/<new-token>/g" /usr/lib/systemd/system/k3s.service
```

```
# 00000000
```

```
systemctl daemon-reload
```

```
# 0000
```

```
systemctl restart k3s
```

```
000 #27
```

```
0 000 000 16 00 2023 08:51:27
```

```
0 000 000 3 00 2025 03:23:54
```