【運算雲 K8S 範本】新增 Worker 節點

2024 年 11 月 7 日
下午 04:06

新增 Worker 節點

MAC 要重設!

更新/etc/hosts

```
192.168.0.81    K8S-Master01

192.168.0.92    K8S-Worker02
192.168.0.91    K8S-Worker01
[root@K8S-Worker02 ~]# vi /etc/hosts
```

▼▼▼▼ Master 節點 ▼▼▼▼

檢查憑證有效期

kubeadm certs check-expiration

```
[root@K8S-Master01 ~]# kubeadm certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm
-config -o yaml'

CERTIFICATE                EXPIRES                  RESIDUAL TIME   CERTIFICATE AUTHORITY   EXTERNAL
LY MANAGED
admin.conf                 Nov 07, 2025 06:34 UTC   360d            ca                      no
apiserver                  Nov 07, 2025 06:34 UTC   360d            ca                      no
apiserver-etcd-client      Nov 07, 2025 06:34 UTC   360d            etcd-ca                 no
apiserver-kubelet-client   Nov 07, 2025 06:34 UTC   360d            ca                      no
controller-manager.conf    Nov 07, 2025 06:34 UTC   360d            ca                      no
etcd-healthcheck-client    Nov 07, 2025 06:34 UTC   360d            etcd-ca                 no
etcd-peer                  Nov 07, 2025 06:34 UTC   360d            etcd-ca                 no
etcd-server                Nov 07, 2025 06:34 UTC   360d            etcd-ca                 no
front-proxy-client         Nov 07, 2025 06:34 UTC   360d            front-proxy-ca          no
scheduler.conf             Nov 07, 2025 06:34 UTC   360d            ca                      no
super-admin.conf           Nov 07, 2025 06:34 UTC   360d            ca                      no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
ca                      Nov 05, 2034 06:34 UTC   9y              no
etcd-ca                 Nov 05, 2034 06:34 UTC   9y              no
front-proxy-ca          Nov 05, 2034 06:34 UTC   9y              no
[root@K8S-Master01 ~]# _
```

產生 TOKEN 後，傳輸給要新增的 Worker 節點

kubeadm token create --print-join-command

```
[root@K8S-Master01 ~]# kubeadm token create --print-join-command
kubeadm join 192.168.0.81:6443 --token cbrtna.40kvxt4o4mrfpm45 --discovery-token-ca-cert-hash sha256
:f79f728413f11bcbd5b39a7ec2d14254b92597b2cb7c6468b7d665343411ca22
[root@K8S-Master01 ~]# kubeadm token create --print-join-command > /tmp/joinWorker
```

▼▼▼▼ 要新增的 Worker 節點 ▼▼▼▼

rm -rf /var/lib/kubelet ; rm -rf /etc/kubernetes
systemctl restart kubelet
modprobe br_netfilter
echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
echo 1 > /proc/sys/net/ipv4/ip_forward

```
[root@K8S-Worker02 ~]# rm -rf /var/lib/kubelet
rm: cannot remove '/var/lib/kubelet/pods/f2e36dd1-b450-4b1a-8716-d951351e467c/volumes/kubernetes.io~
secret/typha-certs': Device or resource busy
rm: cannot remove '/var/lib/kubelet/pods/bd37afc7-504b-4c7d-a1ab-de01dfbd9ffc/volumes/kubernetes.io~
secret/node-certs': Device or resource busy
[root@K8S-Worker02 ~]# rm -rf /etc/kubernetes
[root@K8S-Worker02 ~]# systemctl restart kubelet
[root@K8S-Worker02 ~]# modprobe br_netfilter
[root@K8S-Worker02 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@K8S-Worker02 ~]# cat /proc/sys/net/bridge/bridge-nf-call-iptables
1
[root@K8S-Worker02 ~]# echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
[root@K8S-Worker02 ~]#
```

執行 join 指令

```
[root@K8S-Worker02 ~]# chmod +x /tmp/joinWorker
[root@K8S-Worker02 ~]# /tmp/joinWorker
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config
 -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.
env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

[root@K8S-Worker02 ~]# _
```

▼▼▼▼ Master 節點 ▼▼▼▼

kubectl get nodes
kubectl get nodes -o wide

```
[root@K8S-Master01 ~]# kubectl get nodes
NAME           STATUS   ROLES           AGE     VERSION
k8s-master01   Ready    control-plane   4d18h   v1.29.7
k8s-worker01   Ready    <none>          4d18h   v1.29.7
k8s-worker02   Ready    <none>          22m     v1.29.7
[root@K8S-Master01 ~]# kubectl get nodes -o wide
NAME           STATUS   ROLES           AGE     VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE
              KERNEL-VERSION                      CONTAINER-RUNTIME
k8s-master01   Ready    control-plane   4d18h   v1.29.7   192.168.0.81   <none>        Oracle Linux
Server 8.10   5.15.0-301.163.5.2.el8uek.x86_64   containerd://1.6.32
k8s-worker01   Ready    <none>          4d18h   v1.29.7   192.168.0.91   <none>        Oracle Linux
Server 8.10   5.15.0-301.163.5.2.el8uek.x86_64   containerd://1.6.32
k8s-worker02   Ready    <none>          22m     v1.29.7   192.168.0.92   <none>        Oracle Linux
Server 8.10   5.15.0-301.163.5.2.el8uek.x86_64   containerd://1.6.32
[root@K8S-Master01 ~]# _
```