

# 【運算雲 K8S 範本】新增 Master 節點

2024 年 11 月 7 日

下午 04:06

## 新增 Master 節點

### 新增虛擬機器

名稱 \*

電腦名稱 \*

說明

類型

新增

從範本

### 範本

	vApp 範本名稱	虛擬機器名稱	目錄	作業系統	計算	儲存區
<input type="radio"/>	K8S-Worker01	K8S-Worker01	InProgress	Oracle Linux 8 (64-bit)	CPU 2 記憶體 4 GB	原則 -
<input checked="" type="radio"/>	K8S-Master01	K8S-Master01	InProgress	Oracle Linux 8 (64-bit)	CPU 2 記憶體 4 GB	原則 -

## MAC 要重設!

### NIC

主要 NIC	NIC	已連線	網路介面卡類型	網路	IP 模式	IP 位址	IP 類型	MAC 位址	
<input checked="" type="radio"/>	0	<input checked="" type="checkbox"/>	VMXNE	andrews	靜態 - 手	192.168.0.82	IPv4	重設	<input type="button" value="🗑"/>

## 更新/etc/hosts

```
192.168.0.91 K8S-Worker01
192.168.0.92 K8S-Worker02

192.168.0.81 K8S-Master01
192.168.0.82 K8S-Master02
[root@K8S-Master02 ~]# 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 AUTHORITY	EXPIRES	RESIDUAL TIME	CERTIFICATE AUTHORITY	EXTERNALLY 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

kubeadm token create --print-join-command

```
[root@K8S-Master01 ~]# kubeadm token create --print-join-command
kubeadm join 192.168.0.81:6443 --token cbrtna.40kvoxt4o4mrfpm45 --discovery-token-ca-cert-hash sha256
:f79f728413f11bcbd5b39a7ec2d14254b92597b2cb7c6468b7d665343411ca22
[root@K8S-Master01 ~]# kubeadm token create --print-join-command > /tmp/joinWorker
```

取得既有產生憑證

kubeadm init phase upload-certs --upload-certs

```
[root@K8S-Master01 ~]# kubeadm init phase upload-certs --upload-certs ltail -n 1 > /tmp/cert
[1112 10:00:40.951975 3395021 version.go:256] remote version is much newer: v1.31.2; falling back to
: stable-1.29
[root@K8S-Master01 ~]# cat /tmp/cert
229e479641a24468595a029b93fb716f184479882d5fc342a93dbf02811d66c7
[root@K8S-Master01 ~]#
```

將以上 command 合併給新增 Master 節點使用，傳輸到要新增的 Master 節點

```
[root@K8S-Master01 ~]# cat /tmp/joinMaster
kubeadm join 192.168.0.81:6443 --token 4hxnyv.16gzi3vsxa3b7njm --discovery-token-ca-cert-hash sha256
:f79f728413f11bcbd5b39a7ec2d14254b92597b2cb7c6468b7d665343411ca22 --control-plane --certificate-key
229e479641a24468595a029b93fb716f184479882d5fc342a93dbf02811d66c7
[root@K8S-Master01 ~]#
```

## ▼▼▼▼ 要新增的 Master 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
kubeadm reset (中途卡住則 control+c 中斷，接著執行 join 指令)
```

```
[root@K8S-Master03 ~]# rm -rf /var/lib/kubelet ; rm -rf /etc/kubernetes
[root@K8S-Master03 ~]# systemctl restart kubelet
[root@K8S-Master03 ~]# modprobe br_netfilter
[root@K8S-Master03 ~]# echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
[root@K8S-Master03 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

執行 join 指令

```
chmod +x /tmp/joinMaster
/tmp/joinMaster
```

```
mkdir -p $HOME/.kube ; cp -i /etc/kubernetes/admin.conf $HOME/.kube/config ; chown
$(id -u):$(id -g) $HOME/.kube/config
```

```
[root@K8S-Master02 ~]# mkdir /p $HOME/.kube
mkdir: cannot create directory '/root/.kube': File exists
[root@K8S-Master02 ~]# cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
cp: overwrite '/root/.kube/config'? y
[root@K8S-Master02 ~]# chown $(id -u):$(id -g) $HOME/.kube/config
```

kubectl get nodes

```
[root@K8S-Master02 ~]# kubectl get nodes
NAME                STATUS    ROLES    AGE     VERSION
k8s-master01       Ready    control-plane   4d19h   v1.29.7
k8s-master02       Ready    control-plane   3m54s   v1.29.7
k8s-worker01       Ready    <none>         4d19h   v1.29.7
k8s-worker02       Ready    <none>         87m     v1.29.7
[root@K8S-Master02 ~]#
```

## ▼▼▼▼ 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  4d19h   v1.29.7
k8s-master02        Ready    control-plane  4m23s   v1.29.7
k8s-worker01        Ready    <none>     4d19h   v1.29.7
k8s-worker02        Ready    <none>     87m     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  4d19h   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-master02        Ready    control-plane  4m30s   v1.29.7    192.168.0.82    <none>         Oracle Linux
Server 8.10         5.15.0-301.163.5.2.el8uek.x86_64    containerd://1.6.32
k8s-worker01        Ready    <none>     4d19h   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>     87m     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 ~]# _

```