K3S

K3S 彈性 single-server建置

大綱

當使用 single-server 時，比起直接使用 Embedded DB ，我們在 master 透過 podman 起 SQL-pod 先成為 External DB，之後可方便轉移資料庫讓 K3S 維運下從 single-server 變成 HA-server

作法

階段一：企業剛起步，只有三台機器 (Alpine)

Master0: 192.168.1.211

Wocker: 192.168.1.216 , 192.168.1.217

[master]

① 安裝 podman

② 建立 podman 的 yaml

③ 建立 K3S

作法

階段二: 在 K3S 服務不中斷的情況下，變成 3 master, 2 worker, 1 External DB

① 新增 3 台 master

② 讓 master0 退休

完成結構如下:

External DB: 192.168.1.211

Master: 192.168.1.221 , 192.168.1.226 ,
192.168.1.227

Wocker: 192.168.1.216 , 192.168.1.217

作法

階段三：再新增 2 台 worker

階段一

[master0] 安裝 podman

```
$ sudo tee /etc/apk/repositories << EOF
http://dl-cdn.alpinelinux.org/alpine/v3.13/main
http://mirror.yandex.ru/mirrors/alpine/v3.14/main
http://mirror.yandex.ru/mirrors/alpine/v3.14/community
EOF
$ sudo apk update; sudo apk add podman
Executing busybox-1.32.1-r3.trigger
OK: 1053 MiB in 236 packages
$ sudo rc-update add cgroups
$ sudo rc-service cgroups start
```

階段一

[master0] 建立 yaml

```
$ echo $'apiVersion: v1
kind: Pod
Metadata:
  name: sqldb-p
spec:
  containers:
    - name: sqldbc-p
      image: quay.io/cloudwalker/mariadb
      ports:
        - containerPort: 3306
          hostPort: 8888
          protocol: TCP
      env:
        - name: MYSQL_DATABASE
          value: datatest
        - name: MYSQL_ROOT_PASSWORD
          value: mymariadb
      volumeMounts:
        - name: mariadb-init
          mountPath: /docker-entrypoint-initdb.d
        - name: mariadb-db
          mountPath: /var/lib/mysql
  volumes:
    - name: mariadb-init
      hostPath:
        path: /opt/podman/mariadb.init
    - name: mariadb-db
      hostPath:
        path: /opt/podman/mariadb' > podmansql.yaml
```

階段一

[master0] 設定 initdb

```
$ sudo mkdir -p
/opt/podman/{mariadb.init,mariadb}
```

(在 alpine 先創好要 mount 進去 pod 的目錄)

```
$ sudo tee /opt/podman/mariadb.init/sqluser.sql
<< EOF

GRANT ALL PRIVILEGES ON *.* TO
'k3s'@'192.168.1.211' IDENTIFIED BY 'k3s' WITH
GRANT OPTION;

FLUSH PRIVILEGES;

EOF
```

(initdb 要讀的 sql 指令，檔名一定要是 *.sql)

階段一

[master0] 開機時 pod 同時啟動

```
$ sudo tee /etc/local.d/pod.start <<EOF
#!/bin/sh
sudo podman pod start sqldb-p
EOF
$ sudo chmod +x /etc/local.d/pod.start
$ sudo rc-update add local default
$ sudo openrc
```

階段一

[master0] 建立 mariadb

```
$ sudo podman play kube ./podmansql.yaml
$ sudo podman pod ps
POD ID        NAME     STATUS   CREATED        INFRA ID
# OF CONTAINERS
b56287d6142f  sqldb-p  Running  25 hours ago
344f22fa182d  2

$ sudo apk add mariadb-client
Executing busybox-1.32.1-r3.trigger
OK: 1085 MiB in 240 packages

$ mariadb -uk3s -pk3s -h 192.168.1.211 -P 8888
MariaDB [(none)]>_
```

階段一

[master0] 建立 K3S

```
$ curl -sfL https://get.k3s.io |
INSTALL_K3S_EXEC="--write-kubeconfig-mode 644 \
--datastore-endpoint
mysql://k3s:k3s@tcp(192.168.1.211:8888)/kuberne
tes \
--cluster-cidr=10.20.0.0/16 \
--service-cidr=172.30.0.0/24 \
--no-deploy=servicelb \
--no-deploy=traefik \
--cluster-domain=sre" sh - && sudo reboot
```

階段一

[master0] 加入 worker

```
$ clear; echo " sudo curl -sfL
https://get.k3s.io |
K3S_URL=https://192.168.1.211:6443
K3S_TOKEN=`sudo cat
/var/lib/rancher/k3s/server/node-token`
K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot"
```

```
sudo curl -sfL https://get.k3s.io |
K3S_URL=https://192.168.1.211:6443
K3S_TOKEN=K10fee90c14dd16ff5f8fd9ce9365d1038b4774d6065227
ec113b12cf3ccd18075d::server:9bc5c5566423f8a80dfc4ecb17bd
414c K3S_KUBECONFIG_MODE=‘644’ sh - &&sudo reboot
```

(黑色部分為 token 於 worker 執行)

階段一

建立 worker

```
[192.168.1.216] $ sudo curl -sfL
https://get.k3s.io |
K3S_URL=https://192.168.1.211:6443
K3S_TOKEN=K10fee90c14dd16ff5f8fd9ce9365d1038b47
74d6065227ec113b12cf3ccd18075d::server:9bc5c556
6423f8a80dfc4ecb17bd414c
K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot
```

```
[master0] $ sudo kubectl label node 216-w1
node-role.kubernetes.io/worker=216-w1
```

(貼上worker標籤)

階段一

建立 worker

```
[192.168.1.217] $ sudo curl -sfL
https://get.k3s.io |
K3S_URL=https://192.168.1.211:6443
K3S_TOKEN=K10fee90c14dd16ff5f8fd9ce9365d1038b47
74d6065227ec113b12cf3ccd18075d::server:9bc5c556
6423f8a80dfc4ecb17bd414c
K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot
```

```
[master0] $ sudo kubectl label node 217-w2
node-role.kubernetes.io/worker=217-w2
```

(貼上worker標籤)

測試 K3S

[master0] 檢查 nodes

```
$ kubectl get nodes
NAME      STATUS   ROLES                  AGE    VERSION
211-m1    Ready    control-plane,master   4h5m
v1.21.1+k3s1
217-w2    Ready    worker                 24h
v1.21.1+k3s1
216-w1    Ready    worker                 24h
v1.21.1+k3s1
```

測試 K3S

[master0] 檢查 mariadb

```
$ mysql -uk3s -pk3s -h 192.168.1.211 -P 8888
MariaDB [(none)]> use kubernetes;
MariaDB [kubernetes]> show tables;
(K3S 會自己建一個 kine 資料表，存放 k3s 的 metadata)
+----------------------+
| Tables_in_kubernetes |
+----------------------+
| kine                 |
+----------------------+

1 row in set (0.002 sec)
```

測試 K3S

[master0] 測試 pod 能否正常運作

```
$ kubectl run t1 --restart=Never --image=alpine
-- sleep 30
pod/t1 created
$ kubectl get pods --watch
NAME READY    STATUS            RESTARTS    AGE
t1    0/1    ContainerCreating   0         2s
t1    1/1     Running            0         10s
t1    0/1    Completed           0         40s
^C
$ kubectl delete pods t1
pod "t1" deleted
```

階段二

新增 3 台 master

```
[master] $ curl -sfL https://get.k3s.io |
INSTALL_K3S_EXEC="--write-kubeconfig-mode 644 \
--datastore-endpoint
mysql://k3s:k3s@tcp(192.168.1.211:8888)/kuberne
tes \
--cluster-cidr=10.20.0.0/16 \
--service-cidr=172.30.0.0/24 \
--no-deploy=servicelb \
--no-deploy=traefik \
--cluster-domain=sre" sh - && sudo reboot
```

(master: 192.168.1.221 , 192.168.1.226 , 192.168.1.227)

階段二

[master0] 讓 master0 退休

```
$ /usr/local/bin/k3s-uninstall.sh
```

```
+ id -u
+ '[' 1000 -eq 0 ]
+ exec sudo /usr/local/bin/k3s-uninstall.sh
.
.
.

+ type yum
+ remove_uninstall
+ rm -f /usr/local/bin/k3s-uninstall.sh
```

階段二

[master] (192.168.1.221 , 226 , 227)

```
$ kubectl delete nodes 211-m1
$ kubectl get nodes
NAME    STATUS    ROLES                    AGE    VERSION
221-m1  Ready     control-plane,master     23h    v1.21.1+k3s1
226-m2  Ready     control-plane,master     23h    v1.21.1+k3s1
227-m3  Ready     control-plane,master     23h    v1.21.1+k3s1
216-w1  Ready     worker                   44h    v1.21.1+k3s1
217-w2  Ready     worker                   44h    v1.21.1+k3s1
```

階段三

[master] 加入 worker

```
$ clear; echo " sudo curl -sfL
https://get.k3s.io |
K3S_URL=https://192.168.1.221:6443
K3S_TOKEN=`sudo cat
/var/lib/rancher/k3s/server/node-token`
K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot"
```

```
sudo curl -sfL https://get.k3s.io |
K3S_URL=https://192.168.1.221:6443
K3S_TOKEN=K10fee90c14dd16ff5f8fd9ce9365d1038b4774d6065227
ec113b12cf3ccd18075d::server:9bc5c5566423f8a80dfc4ecb17bd
414c K3S_KUBECONFIG_MODE=‘644’ sh - &&sudo reboot
```

(黑色部分為 token 於 worker 執行)

階段三

建立 worker

```
[192.168.1.XXX] $ sudo curl -sfL
https://get.k3s.io |
K3S_URL=https://192.168.1.221:6443
K3S_TOKEN=K10fee90c14dd16ff5f8fd9ce9365d1038b47
74d6065227ec113b12cf3ccd18075d::server:9bc5c556
6423f8a80dfc4ecb17bd414c
K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot

[master] $ sudo kubectl label node XXX-w3 node-
role.kubernetes.io/worker=XXX-w3
```

(貼上worker標籤)

階段三

建立 worker

```
[192.168.1.XXX] $ sudo curl -sfL
https://get.k3s.io |
K3S_URL=https://192.168.1.221:6443
K3S_TOKEN=K10fee90c14dd16ff5f8fd9ce9365d1038b47
74d6065227ec113b12cf3ccd18075d::server:9bc5c556
6423f8a80dfc4ecb17bd414c
K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot

[master] $ sudo kubectl label node XXX-w4 node-
role.kubernetes.io/worker=XXX-w4
```

(貼上worker標籤)

階段三

[master] (192.168.1.221 , 226 , 227)

```
$ kubectl get nodes
```

```
NAME    STATUS    ROLES                       AGE    VERSION
221-m1  Ready     control-plane,master    23h    v1.21.1+k3s1
226-m2  Ready     control-plane,master    23h    v1.21.1+k3s1
227-m3  Ready     control-plane,master    23h    v1.21.1+k3s1
216-w1  Ready     worker                  44h    v1.21.1+k3s1
217-w2  Ready     worker                  44h    v1.21.1+k3s1
XXX-w3  Ready     worker                   2h    v1.21.1+k3s1
XXX-w4  Ready     worker                   2h    v1.21.1+k3s1
```