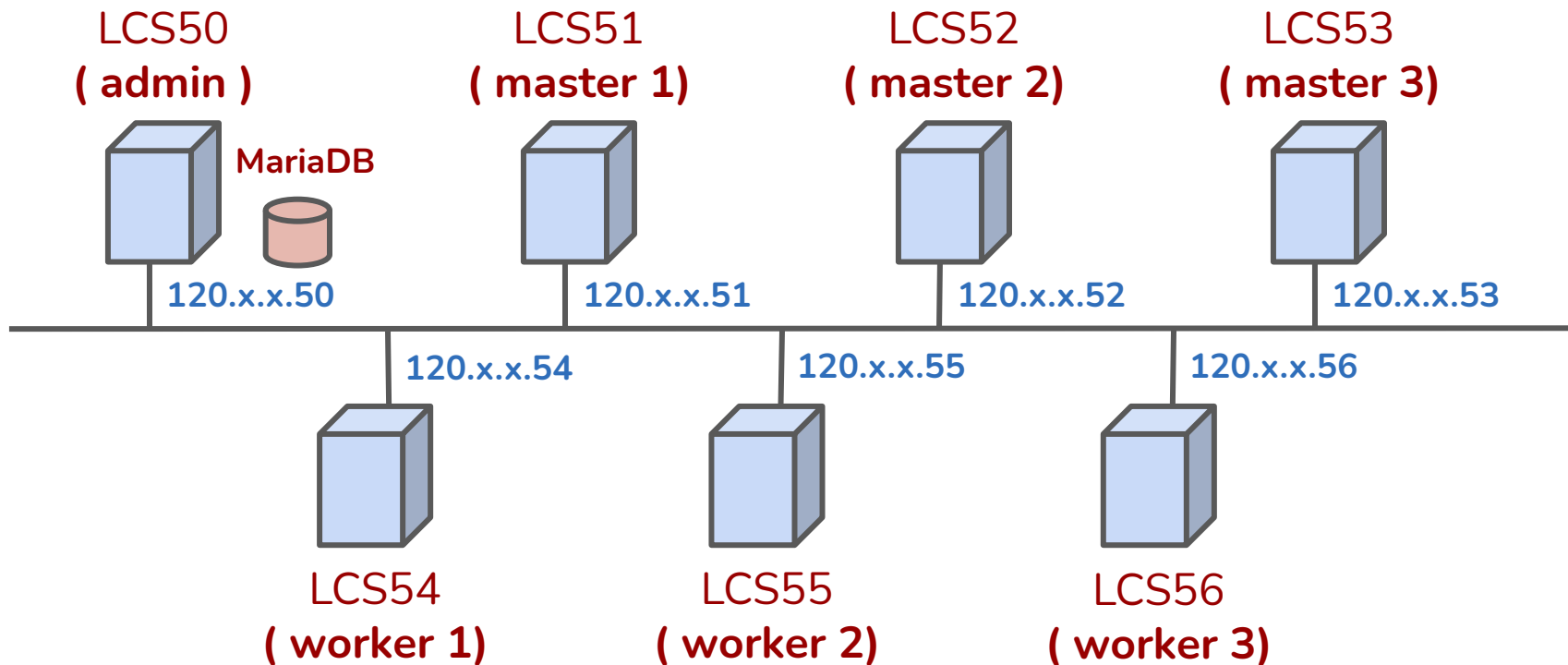


K3S-部署可靠性企業 應用系統建置手冊

指導：陳松林 老師

組員：陳政豪, 莊怡萱, 郭員亨

K3S Cluster Hige Availability



建置 K3S 叢集

安裝 mariaDB

在 LCS50 主機操作

```
[admin]$ sudo apk update
```

```
[admin]$ sudo apk add mariadb mariadb-client
```

```
[admin]$ sudo /etc/init.d/mariadb setup
```

安裝 mariaDB

* Creating a new MySQL database ...

Installing MariaDB/MySQL system tables in '/var/lib/mysql' ...

OK

.....

.....

Two all-privilege accounts were created.

One is root@localhost, it has no password, but you need to be system 'root' user to connect. Use, for example, sudo mysql

The second is mysql@localhost, it has no password either, but you need to be the system 'mysql' user to connect.

安裝 mariaDB

```
[admin]$ sudo rc-service mariadb start
```

```
* Caching service dependencies ...
```

```
* Starting mariadb ...
```

```
210103 15:03:06 mysqld_safe Logging to syslog.
```

```
210103 15:03:06 mysqld_safe Starting mysqld daemon with  
databases from /var/lib/          mysql
```

```
[ ok ]
```

設定 mariaDB

將 mariadb 設為開機時, 自動啟動

[admin]\$ sudo rc-update add mariadb default

* rc-update: mariadb already installed in runlevel `default'; skipping

設定 mariaDB

檢查是否設定成功

```
[admin]$ rc-status default
```

Runlevel: default

cgroups	[started]
sshd	[started]
mariadb	[started]
crond	[started]
acpid	[started]
chronyd	[started]
local	[started]

設定 mariaDB

[admin]\$ sudo nano /etc/my.cnf.d/mariadb-server.cnf

將 skip-networking 加上註解

skip-networking

設定 mariaDB

重新開機

```
[admin]$ sudo reboot
```

```
[admin]$ sudo rc-service mariadb status
```

```
* status: started
```

mariaDB 帳號授權

登入 mariadb

```
[admin]$ mysql -uroot -p
```

mariaDB 帳號授權

MariaDB [(none)]> grant all on *.* to 'k3s'@'120.xx.xx.51' identified by 'k3s' with grant option;

MariaDB [(none)]> grant all on *.* to 'k3s'@'120.xx.xx.52' identified by 'k3s' with grant option;

MariaDB [(none)]> grant all on *.* to 'k3s'@'120.xx.xx.53' identified by 'k3s' with grant option;

MariaDB [(none)]> FLUSH PRIVILEGES;

測試連線 mariaDB

MariaDB [(none)]> select host,user from mysql.user;

Host	User
120.96.143.50	lcs
120.96.143.52	lcs
120.96.143.53	lcs
localhost	mariadb.sys
localhost	mysql
localhost	root

6 rows in set (0.001 sec)

MariaDB [(none)]> quit;

mariaDB 帳號授權

在三台 master

```
[master]$ sudo apk add mariadb-client
```

mariaDB 帳號授權

登入資料庫

```
[master]$ mysql -uk3s -pk3s -h 120.xx.xx.50
```

```
+-----+-----+
| Host      | User      |
+-----+-----+
| 120.xx.xx.51 | lcs      |
| 120.xx.xx..52 | lcs      |
| 120.xx.xx..53 | lcs      |
| localhost   | mariadb.sys |
| localhost   | mysql      |
| localhost   | root       |
+-----+-----+
```

6 rows in set (0.001 sec)

```
MariaDB [(none)]> quit;
```

建立 K3s Master

3 台 master 執行以下命令

```
[master]$ curl -sfL https://get.k3s.io |  
INSTALL_K3S_EXEC="--write-kubeconfig-mode 644 \  
--datastore-endpoint  
mysql://lcs:lcs@tcp(120.xx.xx.56:3306)/kubernetes \  
--cluster-cidr=10.20.0.0/16 \  
--service-cidr=172.30.0.0/24 \  
--cluster-domain=dt.io" sh - && sudo reboot
```


建立 K3s Master

```
[master]$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
lcs51	Ready	master	6d3h	v1.19.5+k3s2
lcs52	Ready	master	6d3h	v1.19.5+k3s2
lcs53	Ready	master	6d3h	v1.19.5+k3s2

加入 k3s node

在 master 執行

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

加入 k3s node

將顯示的指令, 在 **worker** 執行

```
[worker]$ sudo curl -sfL https://get.k3s.io |  
K3S_URL=https://120.xx.xx.52:6443  
K3S_TOKEN=K1087512b0eeb30b8a9ba451215c7a7768d1716d352  
2a519494e1245d2695320cd3::server:8bb8fcf365043724f8b6f9f4b1  
54c0ee K3S_KUBECONFIG_MODE='644' sh - &&sudo reboot
```

設定 k3s worker 標籤

在 master node 執行命令

```
[master]$ kubectl get nodes
```

lcs51	Ready	master	36m	v1.19.5+k3s2
lcs52	Ready	master	40m	v1.19.5+k3s2
lcs53	Ready	master	37m	v1.19.5+k3s2
lcs54	Ready	<none>	30s	v1.19.5+k3s2
lcs55	Ready	<none>	40s	v1.19.5+k3s2
lcs56	Ready	<none>	8s	v1.19.5+k3s2

設定 k3s worker 標籤

```
[master]$ sudo kubectl label node lcs54 node-role.kubernetes.io/worker=lcs54
```

```
[master]$ sudo kubectl label node lcs55 node-role.kubernetes.io/worker=lcs55
```

```
[master]$ sudo kubectl label node lcs56 node-role.kubernetes.io/worker=lcs56
```

查看資料庫

登入

```
[master] $ mysql -ulcs -plcs -h 120.xx.xx.50
```

```
MariaDB [(none)]> use kubernetes;
```

```
MariaDB [kubernetes]> show tables;
```

```
+-----+  
| Tables_in_kubernetes |  
+-----+  
| kine                  |  
+-----+  
1 row in set (0.001 sec)
```

K3s 會自己建一個 kine 資料表, 存放 k3s 的 metadata

建置 Pod

```
[master]$ kubectl run t1 --restart=Never --image=alpine -- sleep 30  
pod/t1 created
```

```
[master] $ 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

建置 Pod

```
[master] $ kubectl delete pods t1  
pod "t1" deleted
```


前期部署

製作 alpine.base images

建造專案資料夾

```
$ mkdir wulin; cd ~/wulin
```

提前部署 CGI 程式

```
$ nano kungfu
```

撰寫 alpine.base Dockerfile

```
$ nano Dockerfile
```

建立 alpine.base image

建立 alpine.base image

```
$ docker build -t alpine.base .
```

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alpine.base	latest	ff61db169094	46 minutes ago	80.5MB
mariadb	10.5.8	3a348a04a815	3 weeks ago	407MB
alpine	3.12.1	d6e46aa2470d	8 weeks ago	5.57MB

測試 alpine.base image

```
$ nano docker-compose.yml
```

```
$ docker-compose -f docker-compose.yml up -d
```

```
Creating httpd ... done
```

```
Creating service ... done
```

建立 database

```
$ docker exec -it service -- bash
root@sqlldb:/# mysql -uroot -proot
MariaDB [(none)]> create database test;
MariaDB [(none)]> use test;
MariaDB [(none)]> show tables;
MariaDB [(test)]> SELECT * FROM test;
MariaDB [(test)]> INSERT INTO test (a int, b int, c int) VALUES
(1,2,3);
MariaDB [(test)]> SELECT * FROM test;
MariaDB [(test)]> exit;
```

測試 httpd

```
$ docker exec httpd hostname -i  
172.23.0.2
```

```
$ curl 'http://172.23.0.2:8888/cgi-bin/kungfu?test&a'  
a 1
```

匯出 alpine.base.tar 檔

```
$ docker save alpine.base > alpine.base.tar
```

部署 images 到 k3s 叢集

將 alpine.base.tar 複製到 k3s 叢集

將 alpine.base.tar 複製到 k3s 叢集(5 台叢集)

```
$ sudo scp alpine.base1.tar bigred@120.xx.xx.51:/home/bigred
```

The authenticity of host '**120.xx.xx.51 (120.xx.xx.51)**' can't be established.

ECDSA key fingerprint is

SHA256:KjiogPkjo+QcsiohQql3/saqCR8JlOjpn6H4DxHV3V3vU.

Are you sure you want to continue connecting (yes/no/[fingerprint])? **yes**

Warning: Permanently added '**120.xx.xx.51**' (ECDSA) to the list of known hosts.

bigred@**120.xx.xx.51**'s password: **bigred**

將 alpine.base.tar 複製到 k3s 叢集

k3s 叢集執行以下命令

製作 image

```
$ sudo ctr images import alpine.base.tar
```

檢查 image

```
$ sudo crictl images
```

IMAGE	TAG	IMAGE ID	SIZE
docker.io/library/alpine.base	latest	ff61db1690949	81.7MB

建造專案資料夾

在 K3S master 執行

```
$ mkdir project; cd project
```

在 k3s 撰寫 yaml 檔

製作 網站應用系統設定檔

\$ nano alpine.base.yml

製作 自動擴充功能設定檔

\$ nano hpa-sp.yml

製作 網站資料庫設定檔

\$ nano mariadb.yml

在 k3s 撰寫 yaml 檔

製作 service 的 yaml 檔案, 讓企業應用系統對外提供服務

\$ nano hpa-svc.yaml

讓資料庫能夠對內提供名稱解析

\$ nano service.yaml

部署應用系統

\$ kubectl apply -f .

deployment.apps/httpd created

horizontalpodautoscaler.autoscaling/hpa-sp created

service/svc-sp created

pod/sqlldb created

service/service created

檢查是否成功啟動

\$ kubectl get all

NAME	READY	STATUS	RESTARTS	AGE
pod/ sqldb	1/1	Running	0	7m29s
pod/httpd-5fd6d6d694-8rdvx	1/1	Running	0	7m29s
pod/httpd-5fd6d6d694-fjp8l	1/1	Running	0	7m29s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
SELECTOR					
service/kubernetes	ClusterIP	172.30.0.1	<none>	443/TCP	2d9h
<none>					
service/service	ClusterIP	172.30.0.181	<none>	3306/TCP	6m44s
app=sql					
service/svc-sp	ClusterIP	172.30.0.180	120.96.143.50	8080/TCP	7m29s

檢查是否成功啟動

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/httpd	2/2	2	2	7m29s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/httpd-5fd6d6d694	2	2	2	7m29s

NAME	REFERENCE	TARGETS	MINPODS
horizontalpodautoscaler.autoscaling/hpa-sp	Deployment/httpd	3%/30%	2
MAXPODS	REPLICAS	AGE	
6	2	7m29s	

高效能

兩個一樣的應用系統可以自動提供平衡負載

\$ kubectl get pod

NAME	READY	STATUS	RESTARTS	AGE
pod/httpd-5fd6d6d694-8rdvx	1/1	Running	0	7m29s
pod/httpd-5fd6d6d694-fjp8l	1/1	Running	0	7m29s

高效能

會發現連到不同的應用系統

```
$ curl '120.96.143.59:8080/hostname'
```

8rdvx

```
$ curl '120.96.143.59:8080/hostname'
```

8rdvx

```
$ curl '120.96.143.59:8080/hostname'
```

fjp8l

一鍵部署 - 進版

進版, 把 image 換成 alpine.base

```
$ kubectl set image deployment.v1.apps/httpd httpd=alpine.base  
--record
```

```
$ kubectl describe deployments httpd | grep "Image"  
Image:      alpine.base
```

一鍵部署 - 退版

退版

```
$ kubectl rollout undo deployment.v1.apps/dep1 --to-revision=1  
kubectl describe deployments httpd | grep "Image"
```

自動維護

查看應用系統名稱

```
$ echo $(kubectl get pod --selector=app=httpd  
--output=jsonpath={.items..metadata.name})
```

```
httpd-5fd6d6d694-qzns8 httpd-5fd6d6d694-pncvb
```

自動維護

刪除其中一個應用系統的名稱

```
$ kd pod httpd-5fd6d6d694-pncvb  
pod "httpd-5fd6d6d694-pncvb" deleted
```

自動維護

可以看到 應用系統即使被刪除也會馬上生出來

```
$ echo $(kubectl get pod --selector=app=httpd  
--output=jsonpath={.items..metadata.name})  
httpd-5fd6d6d694-qzns8 httpd-5fd6d6d694-f2swl
```

自動維護

開啟 **readiness** 探測功能

\$ **nano alpine.base.yml**

...

readinessProbe: #提供readiness探測功能

exec:

command:

- /bin/bash

- -c

- ls /

initialDelaySeconds: 20 #系統啟動20秒後開始探測

periodSeconds: 5

自動維護

\$ kubectl get pod

httpd-6495df464c-zwjpt	0/1	Running	0	16s
httpd-6495df464c-ntds2	0/1	Running	0	16s
httpd-6495df464c-ntds2	1/1	Running	0	23s
httpd-6495df464c-zwjpt	1/1	Running	0	26s

隨需擴充

此為模仿大量需求湧進系統裡

\$./testhpa.sh

成果分享

GITHUB 網址

<https://github.com/xuan103/k3s-Enterprise-Application-System>

歡迎大家的指導

歡迎來信指導：

陳政豪: pk821108@gmail.com

莊怡萱: xuanzhuang103@gmail.com

郭員亨: coco8802231225@gmail.com