NASA hw7

孫凡耘 b04902045

System Administration

1. Server status

使用環境： **RASPBIAN JESSIE LITE**

a. 確定 mod_status 模組有啟用

$ sudo a2enmod status

b. 控制可以開啟狀態報告的網域

Add the following lines to /etc/apache2/apache2.conf

```
<Location server-status>
  SetHandler server-status
  Require ip 127.0.0.1
</Location>


~
~
~
"apache2.conf" 228L, 7206C written
```

c. 重新啟動 Apache Server

$ sudo service apache2 restart

Server Version: Apache/2.4.10 (Raspbian)
Server MPM: prefork
Server Built: Apr 14 2017 13:49:41

Current Time: Saturday, 10-Jun-2017 08:41:53 UTC
Restart Time: Saturday, 10-Jun-2017 08:41:34 UTC
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 18 seconds
Server load: 0.06 0.02 0.00
Total accesses: 3 - Total Traffic: 4 kB
CPU Usage: u0 s0 cu0 cs0
.167 requests/sec - 227 B/second - 1365 B/request
1 requests currently being processed, 5 idle workers

_W____.....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	19135	0/1/1	_	0.00	7	3	0.0	0.00	0.00	192.168.137.1	127.0.1.1:80	NULL
1-0	19136	2/2/2	W	0.00	0	0	2.7	0.00	0.00	192.168.137.1	127.0.1.1:80	GET /server-status HTTP/1.1

Srv	Child Server number - generation
PID	OS process ID
Acc	Number of accesses this connection / this child / this slot
M	Mode of operation
CPU	CPU usage, number of seconds
SS	Seconds since beginning of most recent request
Req	Milliseconds required to process most recent request
Conn	Kilobytes transferred this connection
Child	Megabytes transferred this child
Slot	Total megabytes transferred this slot

Reference:
http://www.arthurtoday.com/2012/12/ubuntu-enable-apache-2-mod-status.html#ixzz4jaIButHF
https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

2. A small taste of PHP

Code in /var/www/html/index.html

```
<html>
<?php
function get_client_ip() {
    $ipaddress = '';
    if (getenv('HTTP_CLIENT_IP'))
        $ipaddress = getenv('HTTP_CLIENT_IP');
    else if(getenv('HTTP_X_FORWARDED_FOR'))
        $ipaddress = getenv('HTTP_X_FORWARDED_FOR');
    else if(getenv('HTTP_X_FORWARDED'))
        $ipaddress = getenv('HTTP_X_FORWARDED');
    else if(getenv('HTTP_FORWARDED_FOR'))
        $ipaddress = getenv('HTTP_FORWARDED_FOR');
    else if(getenv('HTTP_FORWARDED'))
       $ipaddress = getenv('HTTP_FORWARDED');
    else if(getenv('REMOTE_ADDR'))
        $ipaddress = getenv('REMOTE_ADDR');
    else
        $ipaddress = 'UNKNOWN';
    return $ipaddress;
}
echo get_client_ip();
?>

</html>
```

Result:

Reference:https://stackoverflow.com/questions/15699101/get-the-client-ip-address-using-php

3. Docker

Environment: Ubuntu 16.04

a. Install docker

$ curl -sSL https://get.docker.com/ubuntu/ | sudo sh

b. Get the Wordpress Docker Image

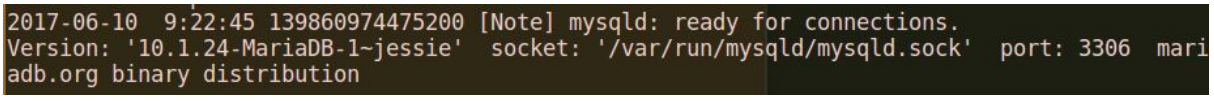
$ sudo docker pull wordpress

c. Install mariadb

$ sudo apt-get install mariadb-server

d. Run container for mariadb(fetches latest image for me)

$ sudo docker run -e MYSQL_ROOT_PASSWORD=nasa mariadb



e. Setting up database in the mariadb container

```
root@willy-X555LB:~# docker exec -ti pensive_mclean mysql -uroot -pnasa
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.24-MariaDB-1~jessie mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database somedb;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all privileges on somedb.* to user_1 identified by 'nasa_1'
    -> ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

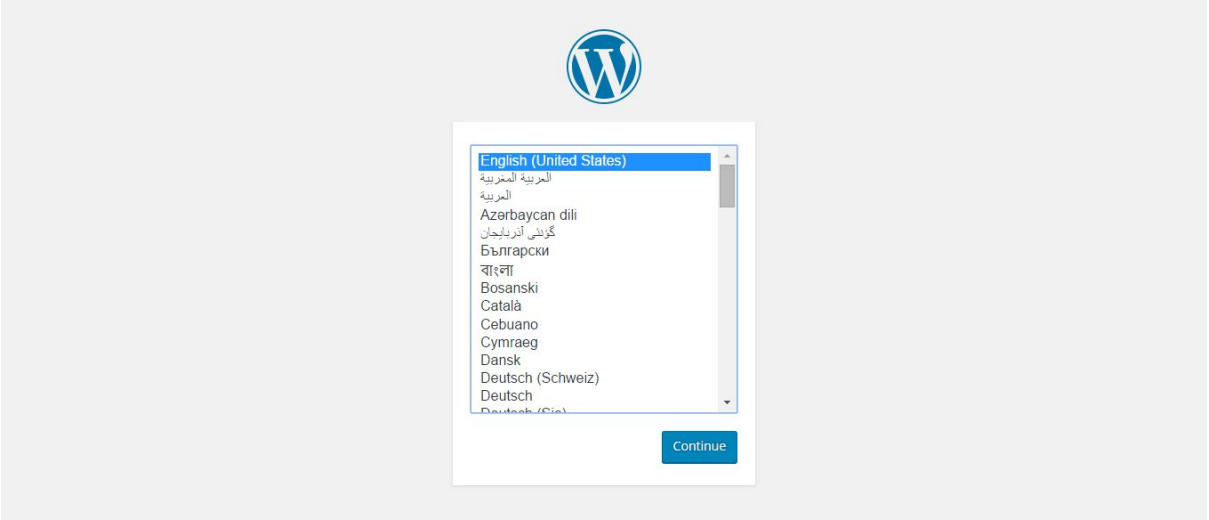
f. Run container for wordpress

$ docker run --name mywordpress --link pensive_mclean:somedb -p 192.168.137.9:80:80 -d wordpress

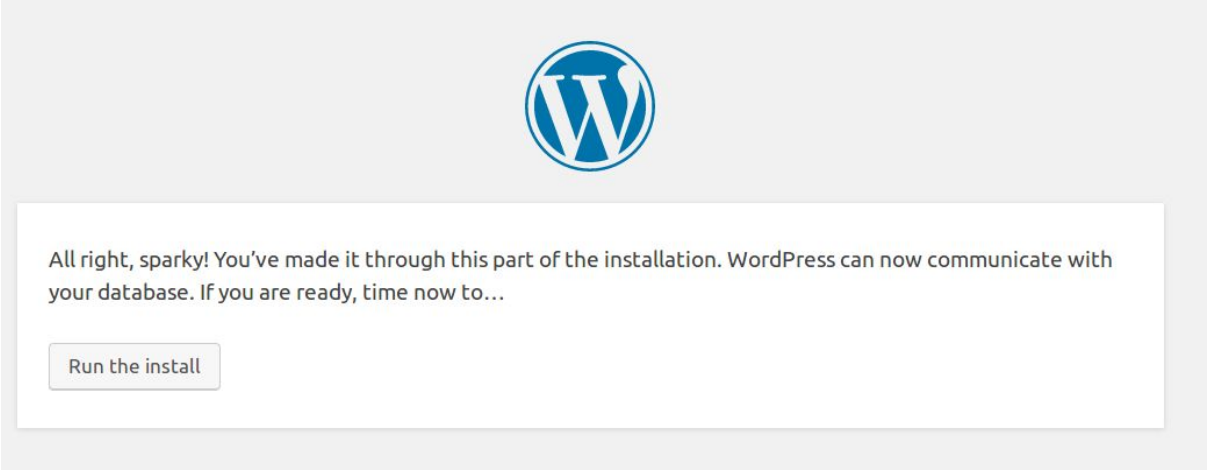
--link **--link pensive_mclean:somedb** – Links the WordPress container with the MariaDB container so that the applications can interact.
-p **192.168.137.9**:80:80 – Tells Docker to pass connections from your server's HTTP port to the containers internal port 80.

```
root@willy-X555LB:~# docker run --name mywordpress --link pensive_mclean:somedb -p 192.168.137.9:80:80 -d wordpress
d63ac7e0b8c602466cc48a5fddb1755ae2a34820a9353231e6dcff109fa24d16
root@willy-X555LB:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                        NAMES
d63ac7e0b8c6        wordpress           "docker-entrypoint..."   7 seconds ago       Up 5 seconds        192.168.137.9:80->80/tcp     mywordpress
2f647c84e7a2        mariadb             "docker-entrypoint..."   15 minutes ago      Up 15 minutes       3306/tcp                     pensive_mclean
root@willy-X555LB:~#
```

g. Test!

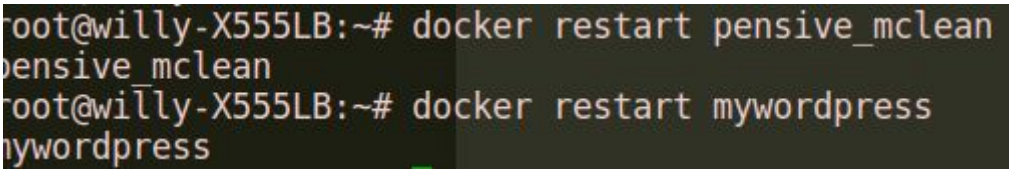


Go through the setup wizard then I can see the following page which says that WordPress can now communicate with your database.



Done!

The content stays the same after



Reference:
https://philipzheng.gitbooks.io/docker_practice/content/basic_concept/
http://doc.nethence.com/docker/mariadb.html
https://www.upcloud.com/support/wordpress-with-docker/