

Docker 簡介

郭忠義

jykuo@ntut.edu.tw

臺北科技大學資訊工程系

Docker 需求

- ❑ 需要在一台伺服器上安裝不同環境測試專案
- ❑ 專案開發時需多種伺服器資源
- ❑ 需要安裝各種環境滿足每一個專案
 - 安裝版本衝突問題

Docker 優點

□ 輕鬆遷移和擴展

- Docker 容器可在任意平台執行，包括實體機器、虛擬機、公有雲、私有雲、個人電腦、伺服器等。這種兼容性讓使用者把應用程式從一個平台直接遷移到另外一個。

□ 更快速的交付和部署

- 對開發和維運（DevOps）人員，希望一次建立或設定，可在任意地方正常執行。
- 開發者使用標準映像檔建立一套開發容器，開發完成後，維運人員可直接使用這個容器部署程式碼。
- Docker 可快速建立容器，快速迭代應用程式，讓整個過程全程可見，使團隊其他成員更易理解應用程式如何建立和工作。
- Docker 容器很輕很快，容器啟動時間是秒級，大量節約開發、測試、部署時間。

Docker 概念

❑ 映像檔 (image)

- 映像檔是 VM 上的 Guest OS，且映像檔本身是唯讀不可修改
- 映像檔包含一個完整的 ubuntu 作業系統環境，裡面僅安裝 Apache 或使用者需要的應用程式。
- 映像檔可用來建立 Docker 容器。

❑ 容器 (container)

- 容器是從映像檔建立的執行實例。可被啟動、開始、停止、刪除。每個容器都是相互隔離、保證安全的平台。
- 可把容器看做一簡易版 Linux 環境和在其中執行的應用程式。
- 一個映像檔可創造多個不同容器，容器像是映像檔的可寫層，與唯讀的映像檔組合成一個環境，便於開發所需應用程式。

Docker 指令

❑ 映像檔 (image)

- docker images : 列出所有的映像檔
- docker rmi “映像檔id”: 刪除映像檔
- docker run “映像檔id”: 將此映像檔啟動為一個獨立的容器

❑ 容器 (container)

- docker ps : 查看執行中的container狀態
- docker rm “容器id”: 刪除container(停止或運行中都行)
- docker exec 外部向執行中的container內部下指令，進入container內部的shell
- docker start“容器id”：啟動停止中的container

安裝redmine docker with mysql

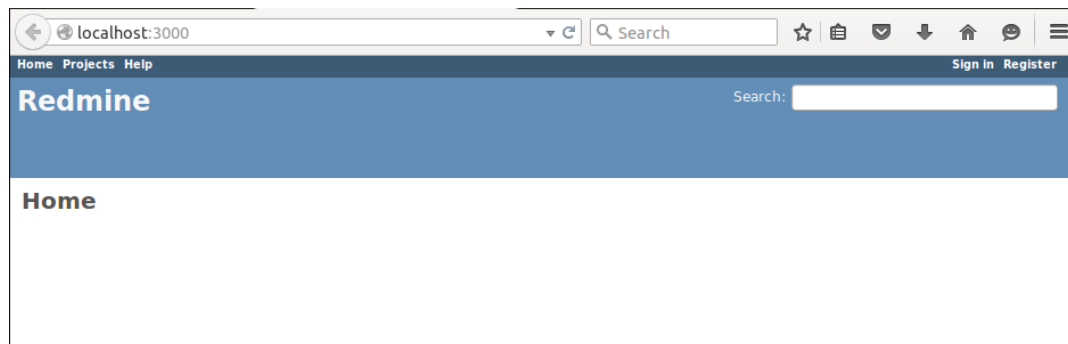
- ❑ `sudo docker pull redmine`
- ❑ `sudo docker run -d --name dockersql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=dockersql -e MYSQL_DATABASE=redmine mysql`

```
sudo docker run -d --name dockersql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=dockersql -e MYSQL_DATABASE=redmine mysql
```

- ❑ `sudo docker run -d --name dockeredmine -p 3000:3000 --link dockersql:mysql redmine`

```
sudo docker run -d --name dockeredmine -p 3000:3000 --link dockersql:mysql redmine
```

- ❑ Open url : localhost:3000



Tomcat + Mysql Docker

❑ 進入 mysql container

○ sudo docker exec -it dockersql /bin/bash

```
root@b2ce9955c757:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

輸入 mysql -u root -p

輸入啟動mysql container
時設定的密碼

○ 建立 database

```
mysql> create database dockerLab;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dockerLab |
| mysql |
| performance_schema |
| redmine |
| sys |
+-----+
6 rows in set (0.02 sec)
```

→ create database “database名稱” ;
(記得加分號)

→ show databases; 確認建立成功

Tomcat + Mysql Docker

❑ 建立table及欄位

```
mysql> use dockerLab;
Database changed
mysql> create table student( id varchar(255), name varchar(255) );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_dockerLab |
+-----+
| student              |
+-----+
1 row in set (0.00 sec)
```

→ use “database名稱” ;

← 建立table

→ show tables; 確認建立成功

❑ 建立使用者及使用者權限

○ 設定完成後，輸入exit兩次離開

```
mysql> create user 'colin'@'%' identified by 'colin';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on dockerLab.* to 'colin'@'%' with grant option;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
```

密碼

帳號

給予使用者權限的資料庫名稱

Tomcat + Mysql Docker

❑ Dockerfile 設定

○ gedit Dockerfile 進入編輯畫面

```
FROM tomcat
MAINTAINER "Colin Chung <colin@c.c>"

RUN apt-get update
RUN apt-get install -y nano
RUN apt-get install -y vim

RUN wget http://140.124.181.9/dockerLab.jsp && mv dockerLab.jsp
/usr/local/tomcat/webapps/examples
RUN wget http://140.124.181.9/dockerlab.html && mv dockerlab.html
/usr/local/tomcat/webapps/examples
RUN wget http://140.124.181.9/mysql-connector-java-5.1.21.jar && mv mysql-connector-
java-5.1.21.jar /usr/local/tomcat/webapps/examples/WEB-INF/lib
RUN apt-get -y install libmysql-java
```

Tomcat + Mysql Docker

❑ 結果如下，存檔離開

```
FROM tomcat
MAINTAINER "Colin Chung <colin@c.c>"

RUN apt-get update
RUN apt-get install -y nano
RUN apt-get install -y vim

RUN wget http://140.124.181.9/dockerLab.jsp && mv dockerLab.jsp /usr/local/tomcat/webapps/examples
RUN wget http://140.124.181.9/dockerlab.html && mv dockerlab.html /usr/local/tomcat/webapps/examples
RUN wget http://140.124.181.9/mysql-connector-java-5.1.21.jar && mv mysql-connector-java-5.1.21.jar /usr/local/tomcat/webapps/examples/WEB-INF/lib
RUN apt-get -y install libmysql-java

~
~
```

❑ Build 此 Dockerfile

○ sudo docker build -t dockertomcat

```
(reducing database ... 19898 files and directories currently installed.)
Preparing to unpack .../libmysql-java_5.1.39-1-deb8u1_all.deb ...
Unpacking libmysql-java (5.1.39-1-deb8u1) ...
Setting up libmysql-java (5.1.39-1-deb8u1) ...
---> 582743b04826
Removing intermediate container 1d1168a86128
Successfully built 582743b04826
```

○ 輸入 sudo docker images，確認此Dockerfile 建置成功

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dockertomcat	latest	582743b04826	16 minutes ago	409 MB

Tomcat + Mysql Docker

❑ 執行 dockertomcat image

○ 將tomcat image 啟動為一個container

- `sudo docker run --name dockertomcat -p 8080:8080 -link dockersql:mysql dockertomcat`

```
sudo docker run --name dockertomcat -p 8080:8080 --link dockersql:mysql dockertomcat
```

○ 確認是否成功執行，輸入 `sudo docker ps`，觀看執行中的container

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e47181aaf95e	dockertomcat	"catalina.sh run"	21 minutes ago	Up 21 minutes	0.0.0.0:8080->8080/tcp	dockertomcat
0c1431b8d379	redmine	"/docker-entrypoint..."	22 hours ago	Up 22 hours	0.0.0.0:3000->3000/tcp	dockeredmine
b2ce9955c757	mysql	"docker-entrypoint..."	22 hours ago	Up 22 hours	0.0.0.0:3306->3306/tcp	dockersql

Tomcat + Mysql Docker

❑ 執行 dockertomcat image

○ 將tomcat image 啟動為一個container

- 因tomcat的port為8080，Jenkins也為8080，若有啟動Jenkins兩者會衝突，先將Jenkins server關閉，`sudo service jenkins stop`
- `sudo docker run --name dockertomcat -p 8080:8080 -link dockersql:mysql dockertomcat`

```
sudo docker run --name dockertomcat -p 8080:8080 --link dockersql:mysql dockertomcat
```

- 確認是否成功執行，輸入 `sudo docker ps`，觀看執行中的container

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e47181aaf95e	dockertomcat	"catalina.sh run"	21 minutes ago	Up 21 minutes	0.0.0.0:8080->8080/tcp	dockertomcat
0c1431b8d379	redmine	"/docker-entrypoint..."	22 hours ago	Up 22 hours	0.0.0.0:3000->3000/tcp	dockeredmine
b2ce9955c757	mysql	"docker-entrypoint..."	22 hours ago	Up 22 hours	0.0.0.0:3306->3306/tcp	dockersql

Tomcat + Mysql Docker

❑ 進入tomcat container shell，修改jsp 及 java 檔

○ 輸入 `sudo docker exec -it dockertomcat /bin/bash`

○ 到 `webapps/examples` 的位置

```
root@e47181aaf95e:/usr/local/tomcat# cd webapps/examples/
root@e47181aaf95e:/usr/local/tomcat/webapps/examples# ls
WEB-INF  dockerLab.jsp  dockerlab.html  index.html  jsp  servlets  websocket
root@e47181aaf95e:/usr/local/tomcat/webapps/examples#
```

○ 修改 `dockerLab.jsp` 檔，`gedit dockerLab.jsp`

```
<%
try{

    id = request.getParameter("id");
    name = request.getParameter("name");
    Class.forName("com.mysql.jdbc.Driver");
    con = DriverManager.getConnection("jdbc:mysql://192.168.200.134:3306/dockerLab?user=colin&password=colin");

    stmt = con.createStatement();
    String upd = "INSERT INTO student(id, name) VALUES (" + id + ", ' " + name + "')";
    stmt.executeUpdate(upd);
    stmt1 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
    String query = "SELECT * FROM student";
    rs = stmt1.executeQuery(query);
    rs.beforeFirst();

    while(rs.next()) {

        out.print("<tr>");
        id=rs.getString("id");
        out.print("<td>"+id+"<td>");
        name=rs.getString("name");
        out.print("<td>"+name+"<td>");
        out.print("<tr>");

    }

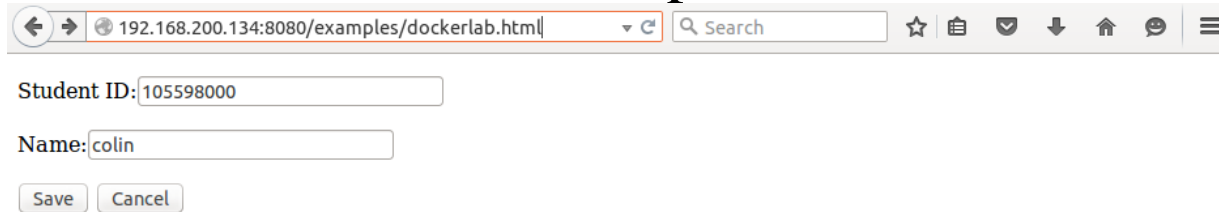
    stmt.close();
    stmt1.close();
    con.close();
}
```

換成自己ip，
設定table名稱
及user帳號密碼

如果table裡欄位和設
定不一樣也需修改

Tomcat + Mysql Docker

- ❑ 輸入 localhost:8080/examples/dockerlab.html 觀看結果



A screenshot of a web browser window. The address bar shows the URL "192.168.200.134:8080/examples/dockerlab.html". The page content includes a form with two input fields: "Student ID:" with the value "105598000" and "Name:" with the value "colin". Below the fields are "Save" and "Cancel" buttons.

- ❑ 截圖，輸出有學號和姓名背景的終端機

