Lab: Docker 實做練習 sshd

參考: https://github.com/sakanamax/SA_dockerReading/blob/master/Dockerfile/sshd/openSUSE423sshd_Dockerfile

建立工作目錄

$ mkdir sshd

進入工作目錄
$ cd sshd

取得練習 Dockerfile

$ wget https://raw.githubusercontent.com/sakanamax/SA_dockerReading/master/Dockerfile/sshd/openSUSE423sshd_Dockerfile

將檔案重新命名
$ mv openSUSE423sshd_Dockerfile Dockerfile

請觀察 Dockerfile 內容

$ cat Dockerfile

觀察 docker images
$ docker images

進行 docker build

$ docker build -t testrepo/sshd_opensuse423 .

觀察 docker images
$ docker images

```
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
testrepo/sshd_opensuse423 latest    f575e03b0440   9 seconds ago    213MB
busybox                   latest    8c811b4aec35   6 weeks ago      1.15MB
opensuse                  42.3      35057ab4ef08   2 months ago     110MB
opensuse                  latest    35057ab4ef08   2 months ago     110MB
```

以背景方式執行 docker 隨機指定 port publish
$ docker run -d -P testrepo/sshd_opensuse423

觀察相關資訊

$ docker ps

```
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                     NAMES
f3ddd5146418   testrepo/sshd_opensuse423 "/usr/sbin/sshd -D"   3 minutes ago    Up 3 minutes    0.0.0.0:32768->22/tcp     objective_mccarthy
```

使用 ssh 指令進行連接測試, 例如 ssh -l root SERVER_IP -p XXXX

指定 port 方式執行 sshd

$ docker run -d -p 8022:22 testrepo/sshd_opensuse423

以相同方式練習

https://github.com/sakanamax/SA_dockerReading/tree/master/Dockerfile/jupyterhub

https://github.com/sakanamax/SA_dockerReading/tree/master/Dockerfile/R_shiny

Lab: docker exec

觀察資訊
$ docker ps

```
CONTAINER ID        IMAGE                        COMMAND                  CREATED             STATUS              PORTS                     NAMES
8a38d6ab1bd6        testrepo/sshd_opensuse423    "/usr/sbin/sshd -D"      9 minutes ago       Up 9 minutes        0.0.0.0:8022->22/tcp      eloquent_neumann
f3ddd5146418        testrepo/sshd_opensuse423    "/usr/sbin/sshd -D"      17 minutes ago      Up 17 minutes       0.0.0.0:32768->22/tcp
objective_mccarthy
```

docker exec
- Run a command in a running container
- 會在 container 啓動 2nd process 執行

直接執行指令方式

$ docker exec 8a38d6ab1bd6 cat /etc/hosts

```
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.3      8a38d6ab1bd6
```

拿來執行 shell
$ docker exec -it 8a38d6ab1bd6 /bin/bash

docker attach
- attach - Attach to a running container 針對已經執行的 container 再次進入到容器內
- 如果使用 --sig-proxy=false 的方式（預設 --sig-proxy=true）那 Ctrl + C 之後 container 不會停止