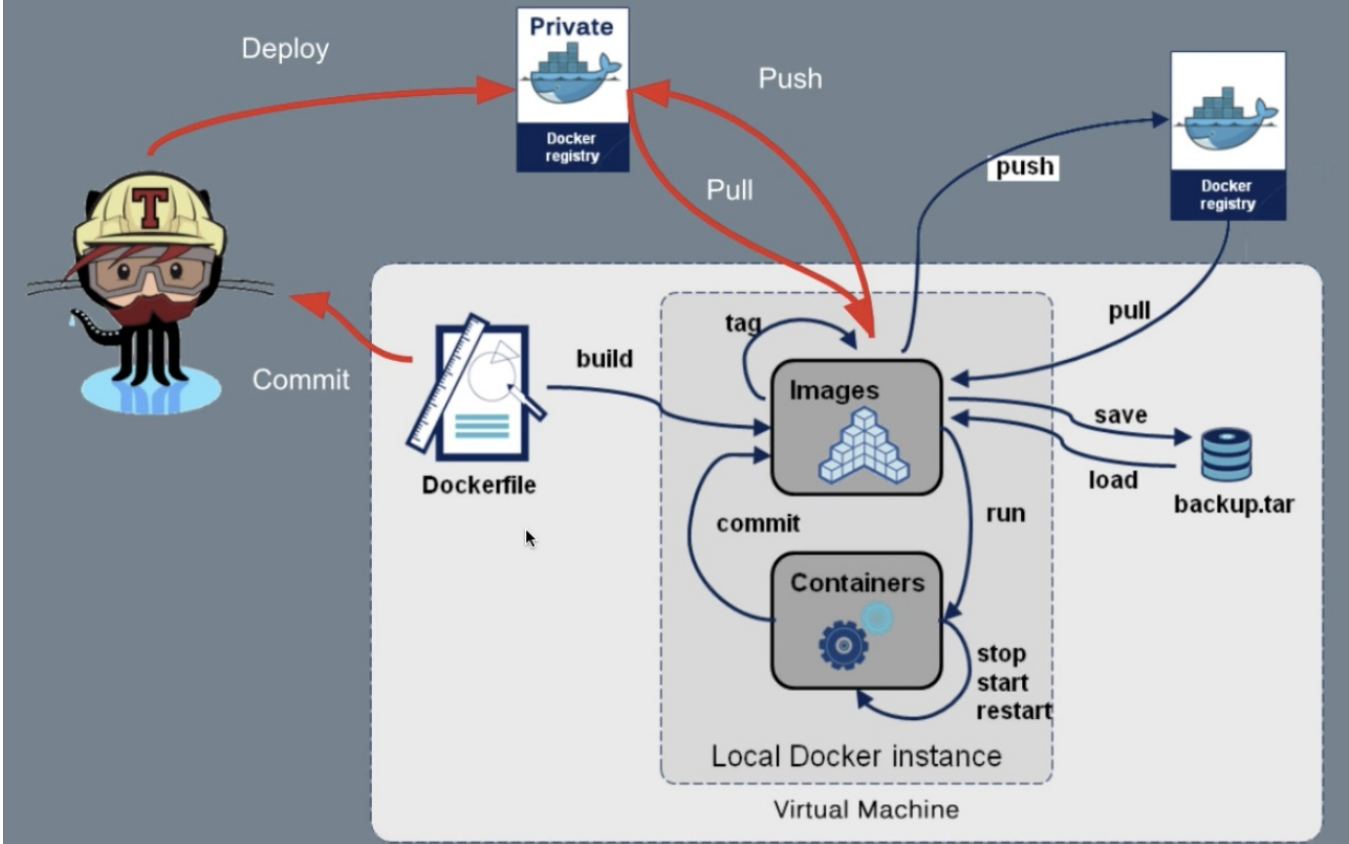


Docker常用指令

Use Scenario



*container 運行

```
#run:
#-d 背景執行
#nginx 執行什麼服務
#ng 自訂的tagname
#-name 自訂名稱
#-e 設定環境變數, TZ=Asia/Taipei => 設定時區
docker container run -d nginx
docker container run -d -name ngx nginx
docker container run -d nginx:ng
docker container run -d -e "TZ=Asia/Taipei" nginx
```

*查看運行container

```
docker container ls -a
```

```
[tomcat@docker-qc ~]$ docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
df6ab3df4b14   nginx    "nginx -g 'daemon of..." 9 seconds ago  Up 8 seconds  80/tcp      modest_cray
940d3d804bcd   nginx    "nginx -g 'daemon of..." 7 minutes ago  Up 7 minutes  80/tcp      elastic_raman
```

*運行container 並進入shell

```
docker container run -it {container name} sh
```

```
[tomcat@docker-qc ~]$ docker container run -it busybox sh
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
e685c5c858e3: Pull complete
Digest: sha256:e7157b6d7ebbe2cce5eaa8cfe8aa4fa82d173999b9f90a9ec42e57323546c353
Status: Downloaded newer image for busybox:latest
/ #
```

*進入已運行container

```
# 進入名為apache的容器執行sh
docker exec -it apache sh
```

*複製檔案

```
# 建立一個hello.html
echo "hello" > hello.html
# 建立apache
docker run -d --name apache httpd:alpine
# 複製檔案至apache容器中
docker cp hello.html apache:/usr/local/apache2/htdocs/
# 複製容器中的檔案至本機
docker cp apache:/usr/local/apache2/htdocs/index.html ./index.html
```

*掛載目錄

```
# 把hello.html掛成index.html
docker run -d --name apache2 -v $PWD/hello.html:/usr/local/apache2/htdocs/index.html httpd:alpine
```

*建立外部連線

```
# -p port 對應 {本機}:{容器}
docker run -d -v $PWD.hell.html:/url/local/apache2/htdocs/index.html -p 80:80 httpd:alpine
```

*刪除container

```
[tomcat@docker-qc ~]$ docker rm -f df6
df6
[tomcat@docker-qc ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
940d3d804bcd       nginx              "nginx -g 'daemon of..." 25 hours ago       Up 25 hours        80/tcp            elastic_raman
```

```
# -f 強制
docker rm -f {container id}

# 刪除所有容器
docker rm -f $(docker ps -a -q)
# 刪除所有名字有demo的容器
docker rm -f $(docker ps -a -q -f name=demo)

# 停用所有container
docker stop $(docker ps -a -q)
# 刪除沒已使用的container 與 image
docker system prune
# 刪除所有映像檔
docker rmi $(docker images -a -q)
# 刪除所有name = <none> 的映像檔
docker rmi $(docker images -f "dangling=true" -q)
# 刪除所有名稱叫shopcar的映像檔
docker rmi $(docker images shopcar* -q)

# 清除volume
docker volume rm $(docker volume ls)

# 刪除所有沒有使用的container image volume
docker system prune -a -f --volumes

#起一個不會停止的centos
docker run -d --name centos7 centos:7 sleep infinity
```

查看要效能

*從主機查看container process

top Display the running processes of a container

```
[tomcat@docker-qc ~]$ docker container ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
df6ab3df4b14   nginx    "nginx -g 'daemon of..." 24 hours ago  Up 24 hours   80/tcp              modest_cray
940d3d804bcd   nginx    "nginx -g 'daemon of..." 25 hours ago  Up 25 hours   80/tcp              elastic_raman
[tomcat@docker-qc ~]$ docker container top df6ab
UID            PID          PPID          C              STIME         TTY           TIME          CMD
root           5355        5336         0              Nov16         ?             00:00:00     nginx: r
aemon off;    101         5395        0              Nov16         ?             00:00:00     nginx: v
```

```
docker container top {container id}
```

*查看狀態

```
# 容器狀態
docker stats {container id | name }
# cpu ram
docker top {container id | name }
# log
docker logs {container id | name }
docker logs -f {container id | name }
```

🕒 修訂版本 #16

★ 由 treeman 建立於 17 🕒 @🕒 2021 01:03:15

🔧 由 treeman 更新於 5 🕒 2023 10:14:58