Docker

Các khái nim c bn v Docker

Ti sao cn phi s dng Docker?

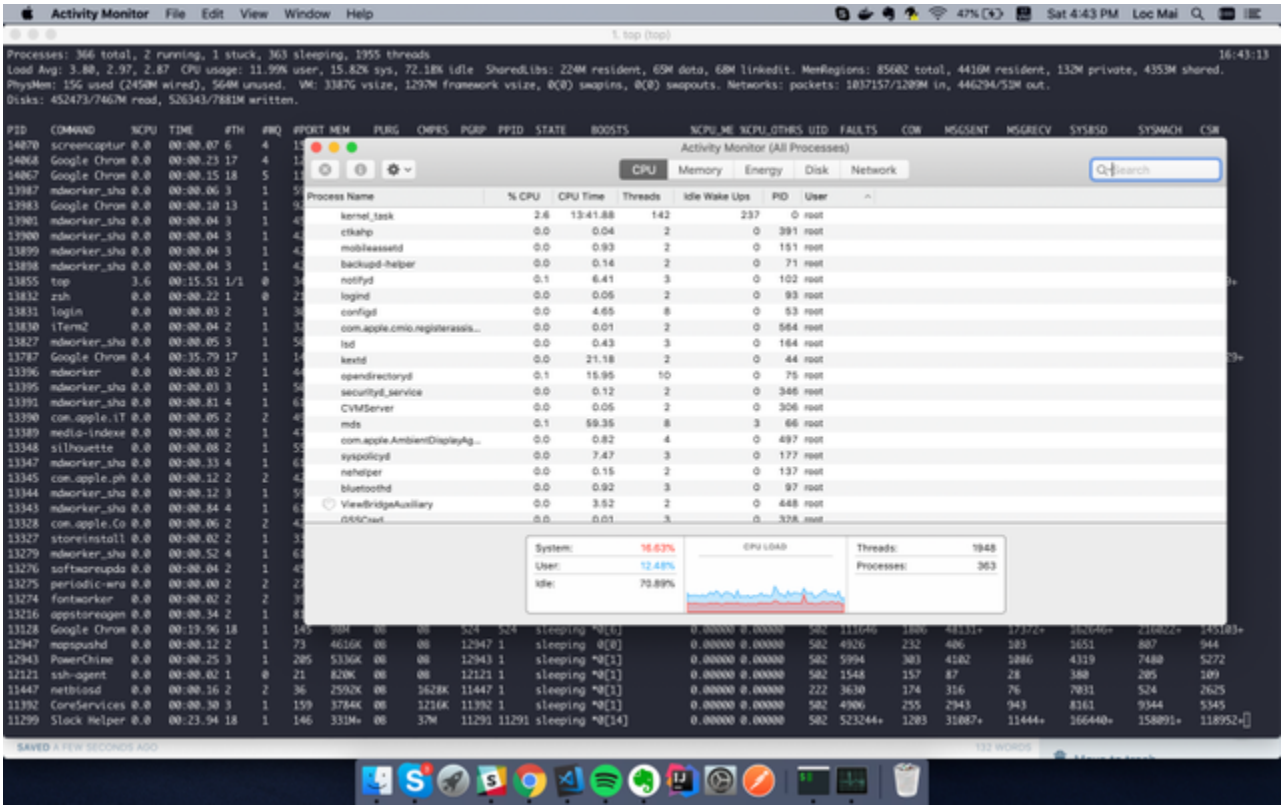
Trc khi i sâu vào ch , chúng ta cn xác nh rõ mt s lý do hay trng hp s cn ti Docker:

- Trong phát trin phn mm, có th s dng Docker nh mt cách óng gói phn mm và m bo ng dng ca bn chy n nh nhiu môi trng (T môi trng Development lên Production)
- Bn cn trin khai mô hình kin trúc Microservices hay mt h thng cn kh nng Scaling tt.
- Trin khai nhanh chóng mt s ng dng Web, công c h tr. Vi các hot ng Testing, mình s dng Docker dng Selenium Grid hay các h thng cluster cho JMeter, Locust.

Vy Docker là gì?

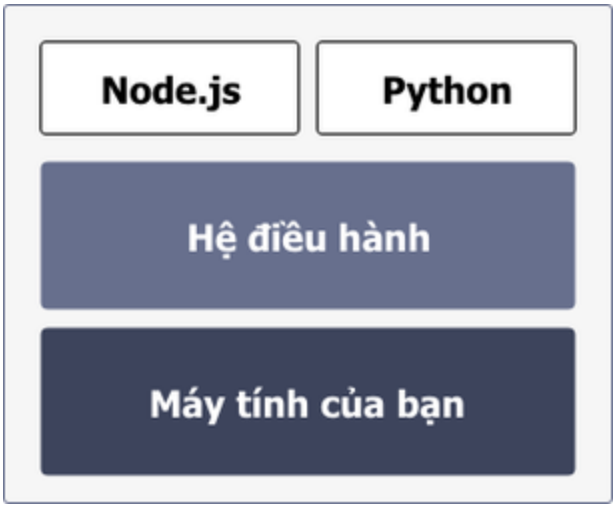
 hiu v Docker, chúng ta cn bit v khái nim “Container” – Nhng máy tính con chy trên mt máy tính cha và Docker Engine.

Gi s, bn ang s dng h iu hành Windows hay MacOS và chy hàng lot các Process khác nhau, bn có th theo dõi nhng Process này bng cách s dng lnh “top” trong Terminal hoc m ca s Tasks Manager ca Windows hay Activity Monitor ca MacOS.



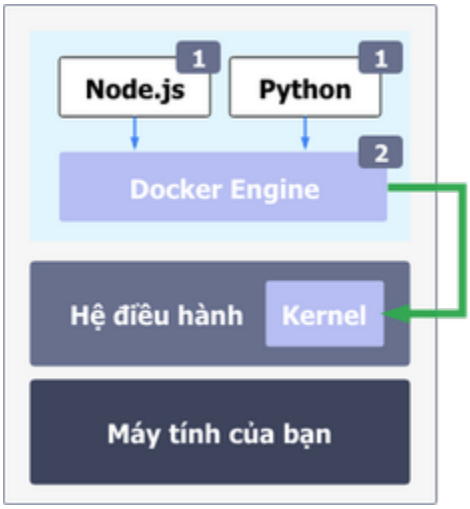
iu cn lu ý ây là nhng Process này cùng s dng mt ngun tài nguyên c nh (CPU, Memory, Disk I/O, Network) và không h có s phân cách (Isolation). Vic này có th dn n trng hp khi mt Process chim dng ht CPU hay Memory sn có, nhng Process khác không th c thc hin.

Và chúng ta có Docker! u tiên chúng ta s cn có Docker Engine chy trên máy tính ch.



Gi s ây là h thng ca chúng ta khi chy hai Process: Mt vi Python và mt vi Node.js

Vi Docker Engine (Hay còn gi tt là Docker), chúng ta có th to ra và thc thi nhng Container (hình dung tng t nh nhng Process trên):



Có 1 vài s thay i hình bên:

1. Trong môi trng Docker, thay cho nhng Process là các Container – Các Process có s tách bit.
2. Container nói chuyn vi Docker Engine và thông qua ó, s gi yêu cu c s dng ngun tài nguyên ca h iu hành trc tip ti Kernel.

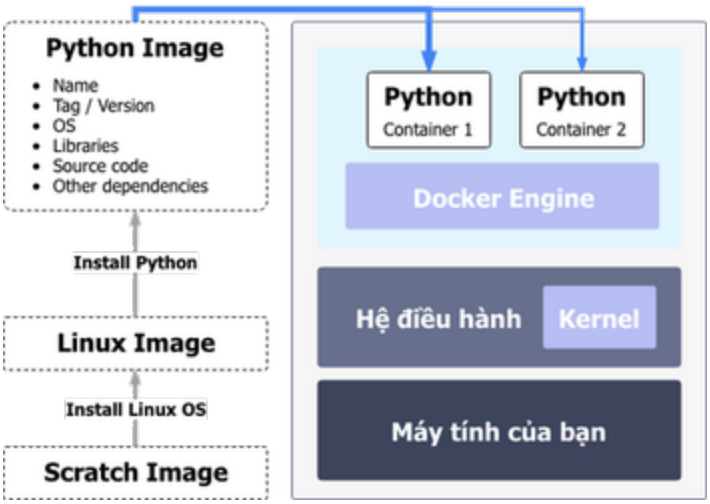
Li ích ca vic “Container hoá” (Containerized) các Process này s giúp chúng ta qun lý s dng ngun tài nguyên máy và tách bit (isolating) môi trng gia các Process.

Tóm li, Container là mt thut ng ch nhng Process c qun lý bi mt Container Enginer (Ngoài Docker Engine ra thì còn mt s công c khác nh Solaris Zones, LXC, etc). Vi bài vit này, mình ch tp trung v Docker Container và nhng li ích ca công ngh này.

Docker Image là gì?

Mt khái nim khác trong Docker: Image – Nhng mu (template) cha nhng dòng quy nh mt Container chy lên nh th nào.

Nu hiu theo lp trình hng i tng thì ta có th xem Image nh mt Class và nhng Containers c to t Image này nh mt Object Instance ca Class ó.



mô hình trên, ta có th thy container 1 và 2 c khi to t Image có tên là Python. Mt Docker Image thng c build t mt Base Image khác (Thng s bt ngun t các phân nhánh ca Linux nh Ubuntu, Alpine, Slim, etc). Nhng Base Image này cng có Base Image cho chính nó và gc cui cùng là Scratch Image.

Các thut ng khác

Ngoài **Docker Container** và **Docker Image**, chúng ta cn bit thêm v:

- **Dockerfile:** Là mt tp tin cha nhng mô t build mt Docker Image
- **Docker Registry:** Là mt kho cha các Docker Image, v c bn chúng ta s có sn 2 Docker Registry sn sàng s dng:
  - **Local Registry:** Nhng Image c kéo v máy ch hay c build ti máy ch s nm ti ây. Cng ging nh source code khi bn clone v máy t GitHub.
  - **Docker Hub:** ây là Public Registry và là Registry mc nh ca Docker. a s các Image s uc pull (kéo v) t ây.
- **Volume / Binding Volume:** Khi mt Container c khi to, nó s có mt Volume (cha d liu) tách bit cha các tp tin h thng hay source code, nu Container ó ngng chy, chúng ta s không còn access c nhng tp tin này na. Do ó, vic binding gia Docker Volume và mt cha ca máy ch (Host machine) là cn thit nu chúng ta mun lu gi nhng tp tin sinh ra t Container (ví d nh log files, configuration files, etc)
- **Container Port / Binding Ports:** Tng t nh Volume, Container cng có riêng nhng Network ports, và access nhng Port này t máy ch, chúng ta cng phi bind các port trên máy ch vi Containers.
- **Docker Client:** công c chính chúng ta giao tip vi Docker Engine.
- **Docker Daemon:** óng vai trò nh Listener thc thi các lnh qun lý Docker nh build Image, run Container, get Image list và get Container list, etc.

Bt u vi Docker

Cài t Docker

Các bn có th tìm hng dn cài t trên phn tài liu ca Docker: https://docs.docker.com/install/

Phn này mình s cp mt s lu ý khi cài t Docker:

- **Enable Virtualization cho Windows 10:** Nu bn s dng Windows 10, hãy nh enable Hyper-V và enable Virtualization (trong bios menu), và uninstall tt c virtualization tool khác nh VMWare hay VirtualBox.
- **Docker Desktop cho Windows 10 / MacOS:** ây là phiên bn n nh, d cài t và có nhiu tính nng h tr tin dng nht. Bn có th d dàng iu chnh Docker thông qua Desktop Tool c cung cp. B cài t gói gn y nhng th cn thit: https://www.docker.com/products/docker-desktop

- ng thi khi s dng Docker Desktop, bn có th d dàng chuyn i gia 2 phiên bn Stable và Edge ca Docker, mình hay s dng tính nng này tri nghim các tính nng mi ca bn Edge và quay tr li bn Stable né nhng li t bn build mi :)).
- **Docker Enterprise for Windows Server 2016+:** Vi các bn Windows Server, la chn tt nht mình tng tri nghim là s dng Docker Enterprise. Phiên bn tng thích tt nht vi Windows Server và s tránh làm bn "wtf" nht khi s dng.
- **Vi Linux distributions:** Docker chy tt vi các h iu hành Linux, mt lu ý nh là bn cn phi ý các thành phn cn thit và nhng hng dn cài c th do hin ti cha có mt Installer c th chung cho Linux nh phiên bn Docker Desktop.
- **Restart-máy là tuyt k cui cùng:** Nu bn ã làm theo mi hng dn cài t và vn không th chy c Docker (du hiu là con cá voi s b lè và vài cái ca s vi hình cá voi khóc) . Bn có th th tuyt k Restart-máy. Không phi lúc nào cng thành công, nhng tuyt k này ã giúp mình vt qua vô s ln try trt khi cài t Docker trên Windows. Chúc bn may mn :))

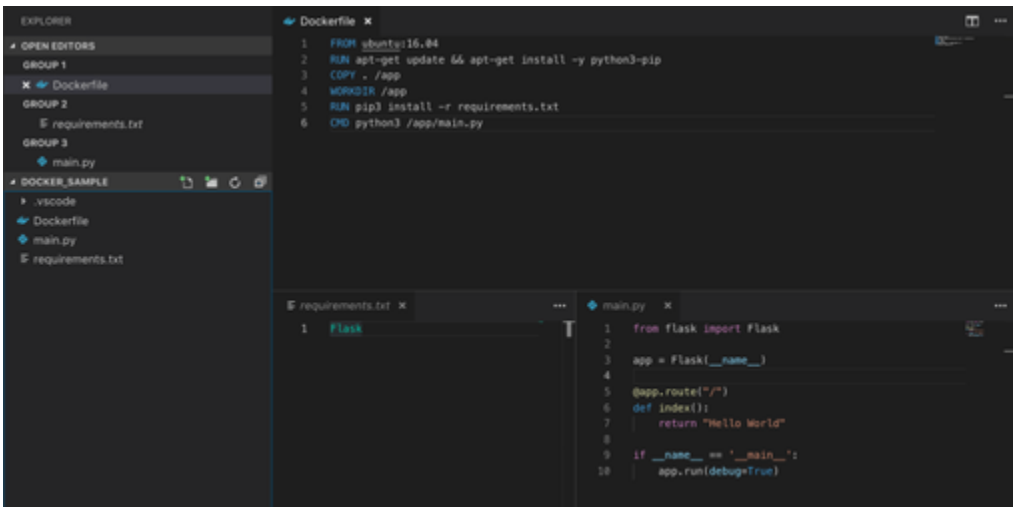
Sau khi cài t xong, bn có th m Terminal (Linux / MacOS) hay Command Prompt / Powershell (Windows) và gõ lnh "docker ps", nu màn hình in ra mt chui "CONTAINER ID IMAGE …" thì chúng ta ã cài t thành công!

Ngoài ra, nu cha quen s dng câu lnh (command line), bn có th s dng th Kitematic – UI tool cho Docker.

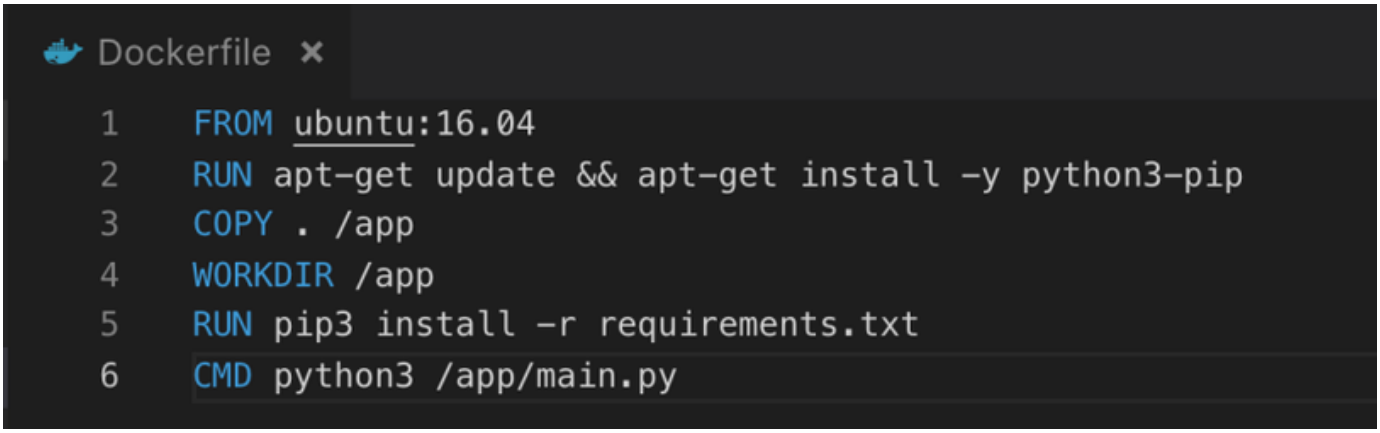
Vit Dockerfile

Cho nhng ví d v sau, chúng ta s build trc mt Docker Image.

 build mt Docker Image, chúng ta cn có Dockerfile – Mt tp nhng ch dn mô t Image c build. Mt ví d nh, vi source code và mt file requirements.txt i kèm cho ng dng Python "Hello World" ca mình – repo trên GitHub:



Gii thích chi tit hn v Dockerfile ca chúng ta:



1. Trc khi build mt Image, chúng ta cn build t mt Base Image khác, ây mình s dng keyword **"FROM"** và chn Ubuntu vi tag (sau ":") 16.04 ch th bn build ca Image cng là phiên bn ca Ubuntu mà mình mun. Tu thuc vào nhu cu khi build cng nh vic la chn Base Image nh th nào cho phù hp, mình s vit thêm v phn này mt bài vit khác v cách optimize cho vic build mt Docker Image.
2. Do s dng h iu hành Ubuntu 16.04, mình dùng **"RUN"** s dng câu lnh chy Package manager ca h iu hành này và cài t 'pip' – (Package manager dành cho Python)
3. Mình dùng **"COPY"** to bn sao ca toàn b source code (". " biu th th mc hin ti) vào folder "app" trong Image.
4. S dng **"WORKDIR"** ch th th mc hot ng trong Image, do mình ã copy source code vào th mc "app", mình s mun chy t ó.
5. Dùng **"RUN"** chy lnh "pip" và cài t nhng Python package mình ã ch nh trong file requirements.txt
6. t câu lnh bt u mt Container bng keyword **"CMD"** (Mt keyword tng t khác là **"ENTRYPOINT"**). Ch th rng khi khi chy Container ca Image này, s chy vi lnh "python3 /app/main.py"

S dng Docker Client

Ti lúc ng tay vào máu me ri. Sau ây chúng ta s s dng Terminal / Command Prompt là ch yu. Trong trng hp bn không nh nhng dòng lnh k tip, bn có th gõ "docker" vào Terminal (mình s gi chung cho c Command Prompt) và nó s in ra danh sách các câu lnh cn thit.

u tiên chúng ta cn kim tra nhng Container ang chy m bo Docker Daemon ã chy:

> *docker ps*

Nu Terminal ch hin ra mt dòng nhng header nh CONTAINER ID, NAME, etc. thì Docker Daemon ã chy ri ó! Trong trng hp Terminal báo:

> *Cannot connect to the Docker daemon. Is the docker daemon running on this host?*

Check li con cá voi Menu Bar, hoc nu bn ang s dng Linux OS, hãy th thêm "sudo" vào trc câu lnh trên.

Tip theo, chúng ta s build image t Dockerfile ã vit phn trc:

> *docker build -t <tên ca image> -f <a ch docker file> <a ch th mc gc>*

- -t: S dng flag này t tên cho Image cng nh ánh tag <tên image>:<tag>
- -f: Ch nh Dockerfile, nu bn không có flag này, Docker s t tìm Dockerfile vi tên default (Dockerfile)
- Phn cui cùng là a ch ti th mc gc.

Mình có th build image u tiên sau khi chuyn Terminal ti th mc cha Dockerfile (s dng lnh cd, nu cha quen vi command line, bn có th s dng lnh sau):

> *git clone https://github.com/locmai/docker-sample.git*
>
> *cd docker-sample*
>
> ***docker build -t docker-test:1.0 .***

Gii thích:

- ây có th thy mình ã b qua flag -f , do mình t tên Dockerfile nh mc nh.
- Phn t cui cùng ca dòng lnh build s là ng dn chính khi build, do mình ang a ch ó nên có th dùng tt bng du "." ch th ti ch

Khi chy, màn hình Terminal s hin ra nhng bc khi build image, ging vi bên di:

> *docker_sample git:(master) docker build -t docker-test:1.0 .*
> *Sending build context to Docker daemon 69.63kB*
> *Step 1/6 : FROM ubuntu:16.04*
> *—> 4a689991aa24*
> *Step 2/6 : RUN apt-get update && apt-get install -y python3-pip*
> *—> Running in 42bc3ec0a6bd*

> *Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]*
> *Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [107 kB]*
> *Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]*
>
> *…*
>
> *Running hooks in /etc/ca-certificates/update.d…*
> *done.*
> *Removing intermediate container 42bc3ec0a6bd*
> *—> cd63958b9606*
> *Step 3/6 : COPY . /app*
> *—> ebcf013e4e77*
> *Step 4/6 : WORKDIR /app*
> *—> Running in aa366a88e2ed*
> *Removing intermediate container aa366a88e2ed*
> *—> 5d427440dc7f*
> *Step 5/6 : RUN pip3 install -r requirements.txt*
> *—> 6037dfec5df8*
> *Step 6/6 : CMD python3 /app/main.py*
> *—> Running in 22333acd1ff8*
> *Removing intermediate container 22333acd1ff8*
> *—> 0312aa41b74f*
> *Successfully built 0312aa41b74f*
> *Successfully tagged docker-test:1.0*

Note: Bn ý s thy nhng dòng **"—> xxxxxxxxxx"** c in ra trên. ó là nhng on ID ca các Images c sinh ra qua tng Step c thc hin. Ví d sau khi **COPY** s ource code vào Image, Docker s sinh ra mt Image tm thi khác và ánh ID cho nó. vic ánh ID này và cache li trng thái build ca Image s giúp tng tc build cho nhng ln build sau. Gi s bn build li mt image khác vi dòng lnh:

> *docker build -t docker-test:2.0 .*

Bn s thy thi gian build nhanh hn và s dòng log cng ít hn rt nhiu.

kim tra Image ã c build, bn dùng lnh:

> *docker images*
>
> *REPOSITORY TAG IMAGE ID CREATED SIZE*
> *docker-test 1.0 2451fc6aaa3b 9 seconds ago 429MB*

Mt danh sách Image trên Local Registry ca chúng ta s hin ra. Bao gm Repository (tên Image), Tag, Image ID, Create time, Size.

Okie, gi ã có cái chúng ta có th chy th:

> *docker run < ID ca Image hoc tên ca Image kèm theo Tag>*

chy docker-test image va build:

> *docker run 0312aa41b74f*
>
> *# hoc*
>
> *docker run docker-test:1.0*

C 2 dòng lnh trên u in ra kt qu nh sau:

> *docker_sample git:(master) docker run docker-test:1.0*
> ** Serving Flask app "main" (lazy loading)*
> ** Environment: production*
> *WARNING: Do not use the development server in a production environment.*
> *Use a production WSGI server instead.*
> ** Debug mode: on*
> ** Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)*
> ** Restarting with stat*
> ** Debugger is active!*
> ** Debugger PIN: 265-615-895*

Nu ý s thy dòng sau: Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

ây là on log khi chy ng dng ca chúng ta. Dù bn thy rng hin ng dng ã chy a ch 127.0.0.1 và port 5000, thì khi vào browser trên máy vn không th access c. Lý do là vì a ch c in ra thuc network ca Docker Container, hãy th dùng mt dòng lnh khác chy Container khác nhé (thoát khi log ca Container ang chy, nhn Ctrl+C, sau ó dùng lnh k:

> *docker run -d –name binded_port_container -p 8000:5000 docker-test:1.0*

Gii thích:

- Vi -d, Docker s chy Container vi Detached Mode, có ngha là Container s chy nn và log ca Container s không hin ra nh dòng lnh trc
- Dùng c –name t tên cho Container, mt xí na bn s thy s khác bit
- Dùng c -p binding port li, s phía trc ":" là port ca máy ch (8000), sau là port ca Container (5000).

Sau khi chy xong bn s thy mt dòng báo hiu Container ã chy, m browser và access th localhost:8000 bn s thy dòng Hello World

kim tra các Container ang chy, chúng ta s dùng lnh "docker ps" và mt danh sách s c in ra:

> *docker_sample git:(master) docker ps*
> *CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES*
> *032dbf3da035 docker-test:2.0 "/bin/sh -c 'python3…" About a minute ago Up About a minute 0.0.0.0:5000->5000/tcp trusting_thompson*
> *b07485e03132 docker-test:1.0 "/bin/sh -c 'python3…" 9 minutes ago Up 9 minutes 0.0.0.0:8000->5000/tcp* ***binded_port_container***
> *43306dfd95b1 docker-test:1.0 "/bin/sh -c 'python3…" 18 minutes ago Up 18 minutes tender_murdock*
> *d3b52a7b4a0c 0312aa41b74f "/bin/sh -c 'python3…" 19 minutes ago Up 19 minutes frosty_murdock*

Chúng ta có th thy các thông tin nh tên ca Container (s random generate nu không c ch nh) và các Port c bind vi nhau, mi Container cng có mt ID khác nhau, ví d thông tin ca Container va c run:

- CONTAINER ID: b07485e03132
- IMAGE: docker-test:1.0
- PORTS: 0.0.0.0:8000->5000/tcp
- NAME: binded_port_container
- CREATED: 9 minutes ago

Okay, gi chúng ta s th mt s thao tác khác:

Dng mt Container:

> *# docker stop < Tên ca Container hoc ID ca Container>*
>
> *docker stop binded_port_containter*
>
> *# docker stop b07485e03132*

Nu gõ "docker ps" , bn s không thy Container ó na. Container ó thc cht vn còn tn ti vi trng thái Existed (Stopped). Vi "docker ps -a" , bn s thy c Container này. restart li Container ó, dùng lnh "docker start":

> *docker start binded_port_container*
>
> *# docker start b07485e03132*

Gõ "docker ps" kim tra Container ã c restart li.

Bn có th mt vài câu lnh và c khác liên quan ti vòng i ca mt Container. Và dn dp sau ó. dn dp hoàn toàn các Container, bn có th dùng nhng lnh sau (sau khi stop các Container):

> *# Xoá nhng Container ã Existed / Stopped*
>
> *docker container prune*
>
> *# Xoá toàn b nhng Container ã stopped, nhng Image "o" c cache li trong quá trình build Image, xoá nhng Docker Network (Khi chy mt Container, Docker s ng ký mt Network interface trên máy ch)*
>
> *docker system prune*
>
> *# Xoá nhng local volume không s dng (Mình s demo công dng ca binding volume tí na)*
>
> *docker volume prune*

Khi chy xong thì bn s phi nhp y (Yes) hoc N (No) confirm rng mình mun xoá:

> *~ docker system prune*
> *WARNING! This will remove:*
> *– all stopped containers*
> *– all networks not used by at least one container*
> *– all dangling images*
> *– all dangling build cache*
> *Are you sure you want to continue? [y/N]*

Gõ "docker ps -a" m bo ã dn dp gn gàng ng Container.

Bn cng có th xoá hn luôn c Docker Image c ra:

> *# docker rmi < Tên và tag ca Image hay ID ca Image>*
>
> *docker rmi docker-test:1.0*

```
# Hoc

docker rmi 0312aa41b74f # <=== ID ca Image docker-test:1.0
```

Nu xoá không c Image thì có th do mt Container nào ó vn còn ang chy da trên Image ó.

Chúng ta va i qua mt s lnh c bn sau:

- **docker ps:** lit kê danh sách các Container và trng thái ca chúng
- **docker build:** dùng build mt Docker Image t Dockerfile.
- **docker images:** dùng lit kê danh sách cái Image có sn trong Local Registry
- **docker run:** dùng khi to mt Container instance t mt Docker Image.
- **docker stop:** dùng dng mt Container ang chy.
- **docker start:** dùng chy li mt Container ang dng.
- **docker <gì ó> prune:** dn dp các Container, cached images, volume và network c to bi Docker trong quá trình khi to Container.
- **docker rmi:** xoá mt Image trên Local Registry

Tip n, chúng ta s th chy mt Image ã build sn. Hu ht các dch v ni ting và c nhiu ngi s dng u ã c vit sn Dockerfile và build Image, nhng Image và Dockerfile này c share public trên trang https://hub.docker.com .

Vic u tiên cn làm là truy cp trang Web trên, ng ký mt Docker ID (Mt lát na chúng ta s dùng ti nó ^^).

Gi nh, mình ang cn dng nhanh mt instance Mongo database to Collection, vc linh tinh. Mình có th s dng Docker Image ã build sn và c publish trên Docker Hub:

```
docker pull mongo

# Ý ngh tng t vi git clone
```

Nu không ch nh Tag ca Image, Docker s pull Image vi tag là latest t Docker Hub.

```
 ~ docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
18d680d61657: Already exists
0addb6fece63: Already exists
78e58219b215: Already exists
eb6959a66df2: Already exists
1bb66a4db707: Already exists
b18fa018e44b: Already exists
66804df38cf1: Pull complete
49311c56c330: Pull complete
178125f0d942: Pull complete
26df264cf5ba: Pull complete
d3cc397c5b0b: Downloading [=====> ] 10.2MB/87.07MB
41e1e68a27fe: Download complete
4c9327cacf3d: Download complete
```

C ch ca pull cng gn ging vi build Image, ó là nó s check các bn Image ID có sn và s ch pull nhng phn build khác còn thiu. Vic này s giúp quá trình pull nhanh hn.

Dùng “docker images” check li danh sách các Image ti Local:

```
 docker_sample git:(master) docker images

REPOSITORY TAG IMAGE ID CREATED SIZE
mongo latest f4f3756fa507 40 hours ago 382MB
mongo 4.0.3 05b3651ee24e 4 weeks ago 382MB
```

Vi Mongo image này, chúng ta cng có th chy tng t nh nhng Image khác bng “docker run”, nhng phn ln chúng ta mun lu li và access c nhng system file ca Mongo. Trong trng hp này, ta có s dng binding volume:

u tiên, to mt cha trên máy có th bind vi Docker Volume:

```
# Vi MacOS hoc Linux OS

WORKING_DIR=$(pwd)

docker run -v $WORKING_DIR/data/:/data/db mongo
```

Gii thích:

- Lnh pwd s giúp ly ng dn th mc hin ti. Mình gán vào bin WORKING_DIR s dng sau này
- S dng lnh docker run cùng vi c -v s giúp bind th mc ca máy ch vi th chc ca Docker, ây mình dn thng ti th mc /data/db (tng t vi Network phi không ^^).
- Mt vài lu ý:

- Nu ng dn không tn ti, Docker có th to các th mc và ng dn úng theo lnh nu có quyn.
- Do cn quyn nên nu to không bind c, các bn nên check li permission ca folder hay drive ã c shared hay cha.
- Nên s dng absolute path cho các mc trên host volume.

Check li folder data bn s thy mt c s file ã c ng b t phía Container mongo qua:

```
docker_sample git:(master)  tree data
data
 WiredTiger
 WiredTiger.lock
 WiredTiger.turtle
 WiredTiger.wt
 WiredTigerLAS.wt
 _mdb_catalog.wt
 collection-0-7461225930492241826.wt
 collection-2-7461225930492241826.wt
 collection-4-7461225930492241826.wt
 diagnostic.data
    metrics.2018-11-17T17-40-10Z-00000
 index-1-7461225930492241826.wt
 index-3-7461225930492241826.wt
 index-5-7461225930492241826.wt
 index-6-7461225930492241826.wt
 journal
    WiredTigerLog.0000000001
    WiredTigerPreplog.0000000001
    WiredTigerPreplog.0000000002
 mongod.lock
 sizeStorer.wt
 storage.bson

2 directories, 20 files
```

Okay, vy là chúng ta ã hiu thêm v binding volumes vi Docker Container. K n s là v vic Push / Published mt Image c build lên Docker Hub hay mt Remote Registry khác, trc hn ht chúng ta s build li Image phn u:

```
docker build -t locmai/docker-test:1.0 .
```

S khác bit: ây mình ã thêm mt phn trc tên Image: Docker ID – locmai.

Kim tra li images c build:

```
 docker_sample git:(master) docker images “locmai/*”
REPOSITORY TAG IMAGE ID CREATED SIZE
locmai/docker-test 1.0 65da2c265694 46 seconds ago 429MB
```

push lên Docker Hub:

```
docker push locmai/docker-test:1.0
```

Docker có th s yêu cu bn phi login vi Docker ID ã ng ký. ây s dng Docker ID mà bn ã ng ký t tên cho Image Docker có th tìm repo chính ch trên Docker Hub và push lên ó.

ây là Image ca mình ã push lên sau khi build: https://hub.docker.com/r/locmai/docker-test/

Tóm gn v mt s lnh mi:

- **docker pull:** dùng pull Image có sn t Docker Hub hay mt Remote Registry khác
- **docker push:** dùng push mt Image t Local Registry lên Docker Hub hay mt Remote Registry khác.
- **docker run -v:** binding volumes gia máy ch và Container

, ht ri!!!

Trc khi kt thúc mình xin rút kt li mt s ý:

- S dng Docker nu bn cn trin khai mt ng dng Web hay công c h tr nhanh nht có th.
- Container là khái nim v phân tách các Process bi mt Container Engine, giúp các Process mang tính óng gói hn.
- Docker Image là mt mu mô t v Container. Xem Image nh Class và Container nh Instance ca Class ó.
- Các lnh qun lý vòng i ca Container bao gm: run, start, stop, ps.
- Các lnh qun lý Image bao gm: images, rmi, build, pull, push.
- dn dp: docker system prune

Bài vit cng ã khá dài, hi vng nhng khái nim v Docker s tr nên d hiu hn vi mi ngi và có th th áp dng trong công vic hàng ngày. Good luck guys! ^^

Ph lc

S khác bit gia Container và Virtual Machine (Máy o)

Container khác bit vi Virtual Machine nh VMWare hay VirtualBox im công ngh Container không gi lp toàn b phn cng nh các Virtualization tool mà thay vào ó các Container nói chuyn trc tip vi Kernel ca máy ch thông qua Container Engine. Do ó các “máy-tính-con” container s vô cùng nh ký (light-weight). Mt hình nh so sánh c s dng khá nhiu: