

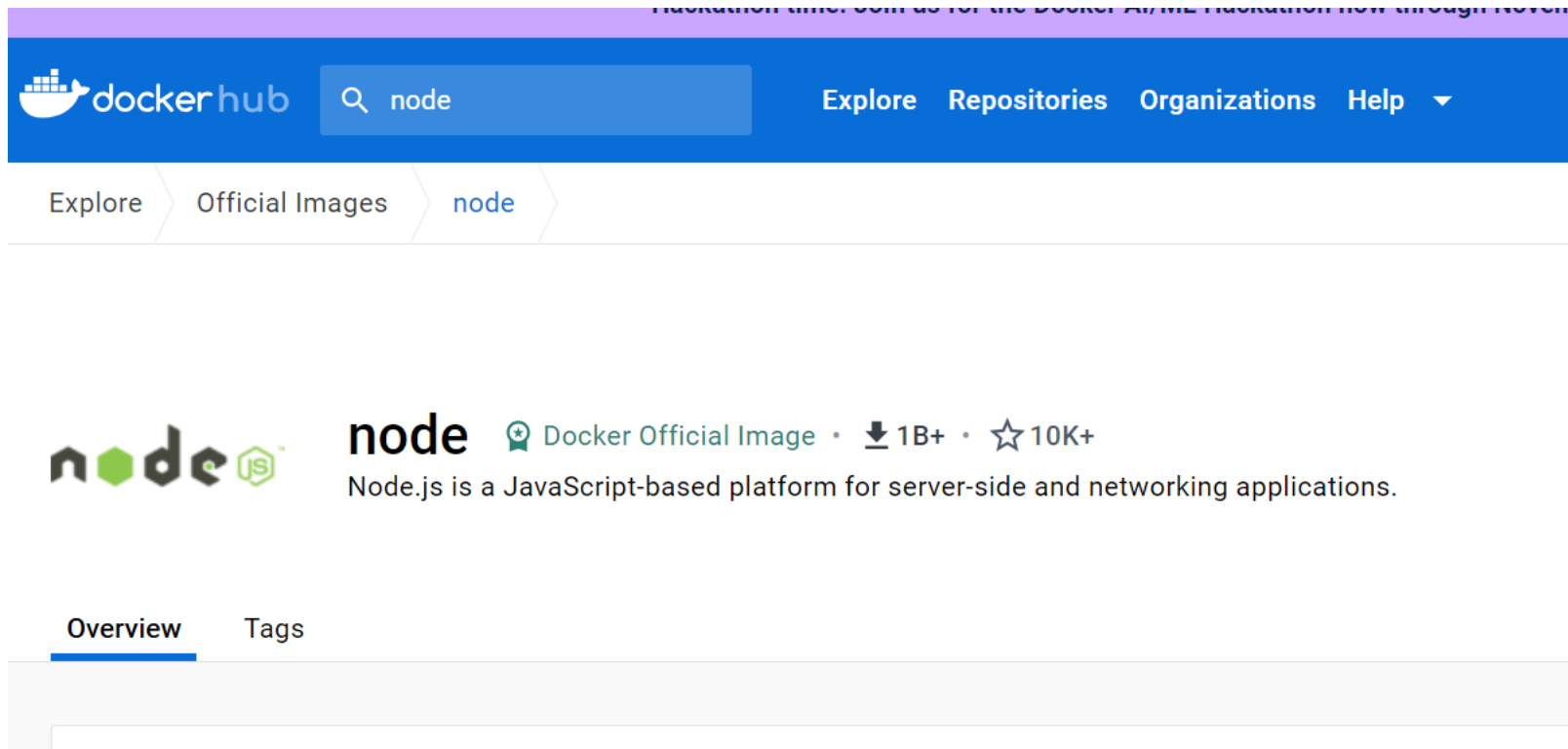
Dockerfile - Building Docker Image

郭益華

說明

- Dockerfile 撰寫
- 建立自己的 image

至 dockerhub 搜尋 nodejs



Dockerfile撰寫

```
1 FROM node:13-alpine
2
3 ENV MONGO_DB_USERNAME=admin \
4     MONGO_DB_PWD=password
5
6 RUN mkdir -p /home/app
7
8 COPY . /home/app
9
10 CMD ["node", "server.js"]
```

Docker container為一個微型的Linux，設定均和Linux操作一樣使用Linux 指令

FROM:

使用的後端框架及版本

ENV:

設定環境變數

RUN:

在home裡面建立一個app資料夾

COPY:

將當前路徑的程式檔案複製進此container中的app資料夾。
.(當前目錄)

CMD:

在此container執行命令，啟動後端伺服器

Docker build 建立 images

Command:

Docker build -t < named: tag version > : 建立自己的images

-t: 命名

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker build -t my-app:1.0 .
[+] Building 17.5s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 192B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/node:13-alpine                4.2s
=> [1/3] FROM docker.io/library/node:13-alpine@sha256:527c70f74817f6f6b5853588c28de33459178ab72421f1fb7b63a281a 12.5s
=> => resolve docker.io/library/node:13-alpine@sha256:527c70f74817f6f6b5853588c28de33459178ab72421f1fb7b63a281ab 0.0s
=> => sha256:cbdbe7a5bc2a134ca8ec91be58565ec07d037386d1f1d8385412d224deafca08 2.81MB / 2.81MB 1.2s
=> => sha256:780514bed1adda642a7f7008ea5b1477bb78a23e3f76eabbdde21031b2185a2f 35.30MB / 35.30MB 10.5s
=> => sha256:5d74fb112a7d313344f9f679ccb975062377464235fac36e1b952aa2b125d5e2 2.24MB / 2.24MB 3.2s
=> => sha256:527c70f74817f6f6b5853588c28de33459178ab72421f1fb7b63a281ab670258 1.65kB / 1.65kB 0.0s
=> => sha256:4d6b35f8d14b2f476005883f9dec0a93c83213838fd3def3db0d130b1b148653 1.16kB / 1.16kB 0.0s
=> => sha256:8216bf4583a5d422edd579a3b77f66a8fe127c4acb5eabecf085d911786f3e68 6.77kB / 6.77kB 0.0s
=> => extracting sha256:cbdbe7a5bc2a134ca8ec91be58565ec07d037386d1f1d8385412d224deafca08 0.2s
=> => sha256:4b9536424fa1675675e7edd59ae9a92573f0c010c5458733a493e2c3e1daf1ae 283B / 283B 2.2s
=> => extracting sha256:780514bed1adda642a7f7008ea5b1477bb78a23e3f76eabbdde21031b2185a2f 1.5s
=> => extracting sha256:5d74fb112a7d313344f9f679ccb975062377464235fac36e1b952aa2b125d5e2 0.1s
=> => extracting sha256:4b9536424fa1675675e7edd59ae9a92573f0c010c5458733a493e2c3e1daf1ae 0.0s
=> [internal] load build context                                                  0.1s
=> => transferring context: 6.09MB                                              0.1s
=> [2/3] RUN mkdir -p /home/app                                                  0.5s
=> [3/3] COPY . /home/app                                                        0.0s
=> exporting to image                                                            0.1s
=> => exporting layers                                                          0.1s
=> => writing image sha256:459f1d551aae55e9dec54051bf0afdb64ad36a4263ee9657362ced56ea7b921c 0.0s
=> => naming to docker.io/library/my-app:1.0 0.0s
```

可看到已成功建立image

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-app	1.0	459f1d551aae	24 seconds ago	121MB

docker run發現有error

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker run my-app:1.0
internal/modules/cjs/loader.js:965
  throw err;
  ^

Error: Cannot find module '/home/app/server.js'
    at Function.Module._resolveFilename (internal/modules/cjs/loader.js:962:15)
    at Function.Module._load (internal/modules/cjs/loader.js:838:27)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)
    at internal/main/run_main_module.js:17:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
```

刪除image重來會發現無法刪除

原因:

雖然前面docker run有error(程式碼的error)，但實際也會自動建立一個container，所以要刪除images必須先停止container並刪除container才能成功刪除image

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker rmi my-app:1.0
Error response from daemon: conflict: unable to remove repository reference "my-app:1.0" (must force) - container 3a63eabc0c07 is using its referenced image 011d7ea2eb92
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	P
3a63eabc0c07	my-app:1.0	"docker-entrypoint.s..."	6 minutes ago	Exited (1) 6 minutes ago	

可看到確實有container再運行

Remove container

Command:

`docker rm <container id>`: 移除container

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker rm 3a63eabc0c07
3a63eabc0c07
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PO
RTS	NAMES				
209da0e537be	mongo-express	"/sbin/tini -- /dock..."	24 hours ago	Exited (143) 23 hours ago	
	mongo-express				
f5fde2a69934	mongo	"docker-entrypoint.s..."	24 hours ago	Exited (0) 23 hours ago	

Remove image

Command:

`docker rmi <image name>: 移除image`

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-app	1.0	011d7ea2eb92	10 minutes ago	121MB
mongo-express	latest	a776ee465303	28 hours ago	247MB
mongo	latest	3be86e9501b0	10 days ago	748MB
jupyter/pyspark-notebook	latest	b4062e157cea	3 weeks ago	4.91GB
my-flask-app	latest	3af02193fbc7	3 weeks ago	1.09GB
redis	7.0.13	7cf5a2b10347	5 weeks ago	130MB
redis	latest	5b0542ad1e77	5 weeks ago	138MB
sqlc/sqlc	latest	e5c3b6c57c08	5 weeks ago	98.2MB
confluentinc/cp-zookeeper	latest	04bd40128a4e	7 weeks ago	849MB
postgres	12-alpine	57609cf5d1df	2 months ago	230MB
postgres	14-alpine	955d825eef13	2 months ago	235MB
postgres	9.6	027ccf656dc1	20 months ago	200MB
confluentinc/cp-kafka	5.3.1	3ce94ccaeaea	24 months ago	590MB

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker rmi my-app:1.0
Untagged: my-app:1.0
Deleted: sha256:011d7ea2eb922694cb78dcba52358e7118ac1f3204ae2ae7f604cdc32d7c9897
```

可看到成功移除image

成功移除my-app image

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker rmi my-app:1.0
Untagged: my-app:1.0
Deleted: sha256:011d7ea2eb922694cb78dcba52358e7118ac1f3204ae2ae7f604cdc32d7c9897
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mongo-express	latest	a776ee465303	28 hours ago	247MB
mongo	latest	3be86e9501b0	10 days ago	748MB
jupyter/pyspark-notebook	latest	b4062e157cea	3 weeks ago	4.91GB
my-flask-app	latest	3af02193fbc7	3 weeks ago	1.09GB
redis	7.0.13	7cf5a2b10347	5 weeks ago	130MB
redis	latest	5b0542ad1e77	5 weeks ago	138MB

重新build

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\techworld-js-docker-demo-app-master> docker build -t my-app:1.0 .
[+] Building 3.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 404B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/node:13-alpine 1.1s
=> [1/5] FROM docker.io/library/node:13-alpine@sha256:527c70f74817f6f6b5853588c28de33459178ab72421f1fb7b63a281ab 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 2.53kB 0.0s
=> CACHED [2/5] RUN mkdir -p /home/app 0.0s
=> [3/5] COPY ./app /home/app 0.0s
=> [4/5] WORKDIR /home/app 0.0s
=> [5/5] RUN npm install 2.3s
=> exporting to image 0.1s
=> => exporting layers 0.1s
=> => writing image sha256:2de504f8fd65ecd3b0ed68047c544bf6ede6ed1603e9ddebd0cd061fc7ffa9c6 0.0s
=> => naming to docker.io/library/my-app:1.0 0.0s
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\techworld-js-docker-demo-app-master> docker run my-app:1.0
app listening on port 3000!
```

成功啟動

查看 my-app container 狀態

```
PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
4edc6a41ea90   my-app:1.0    "docker-entrypoint.s..." 47 seconds ago Up 46 seconds          fervent_mirzakhani
PS C:\Users\jerry> docker logs 4edc6a41ea90
app listening on port 3000!
```

進入container內部查看

可以看到我們在local撰寫的相關程式皆已在docker container內

```
PS C:\Users\jerry> docker exec -it 4edc6a41ea90 /bin/sh
/home/app # ls
images          index.html      node_modules    package-lock.json  package.json      server.js
/home/app # env
NODE_VERSION=13.14.0
HOSTNAME=4edc6a41ea90
YARN_VERSION=1.22.4
SHLVL=1
HOME=/root
MONGO_DB_USERNAME=admin
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MONGO_DB_PWD=password
PWD=/home/app
/home/app # exit
PS C:\Users\jerry>
```

End