Create Docker Network & Run Multiple Containers

郭益華

目錄

1. 簡介說明
2. Developing with Containers
3. 撰寫.yaml 使用docker-compose整合

1. 簡介說明

說明

- Docker container 開發
- 以MongoDB & Mongo – Express 為例進行實際操作
- 撰寫.yaml
- 使用 docker-compose 整合 啟動多個 container

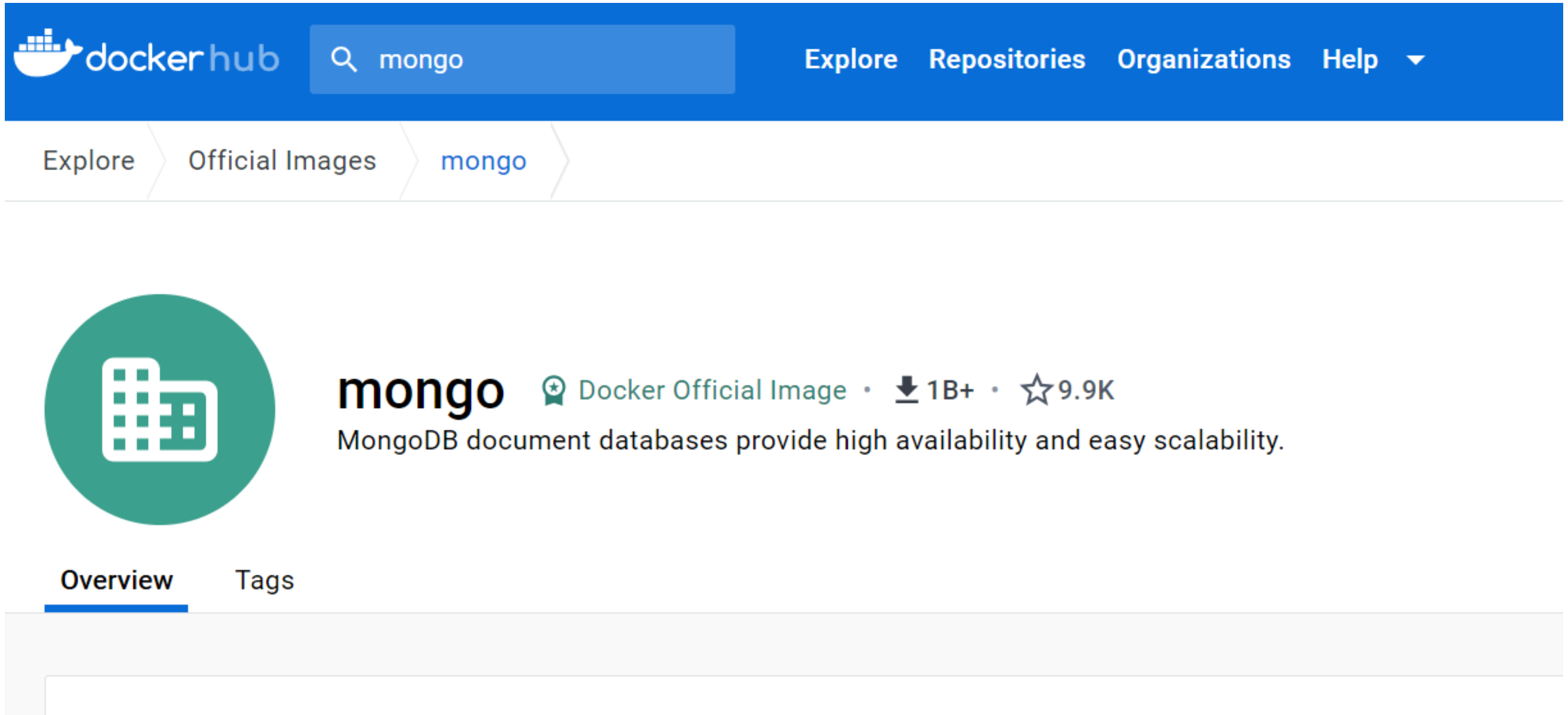
新學到的指令

- docker network create <named network>: 建立一個新網路
- docker compose –f <.yaml> up: 同時啟動多個containe並建立network
- docker compose –f <.yaml> down: 同時關閉刪除containe & network
- -f: 指定要執行的檔案

2. Developing with Containers

至 dockerhub 搜尋 mongo

MongoDB - Database

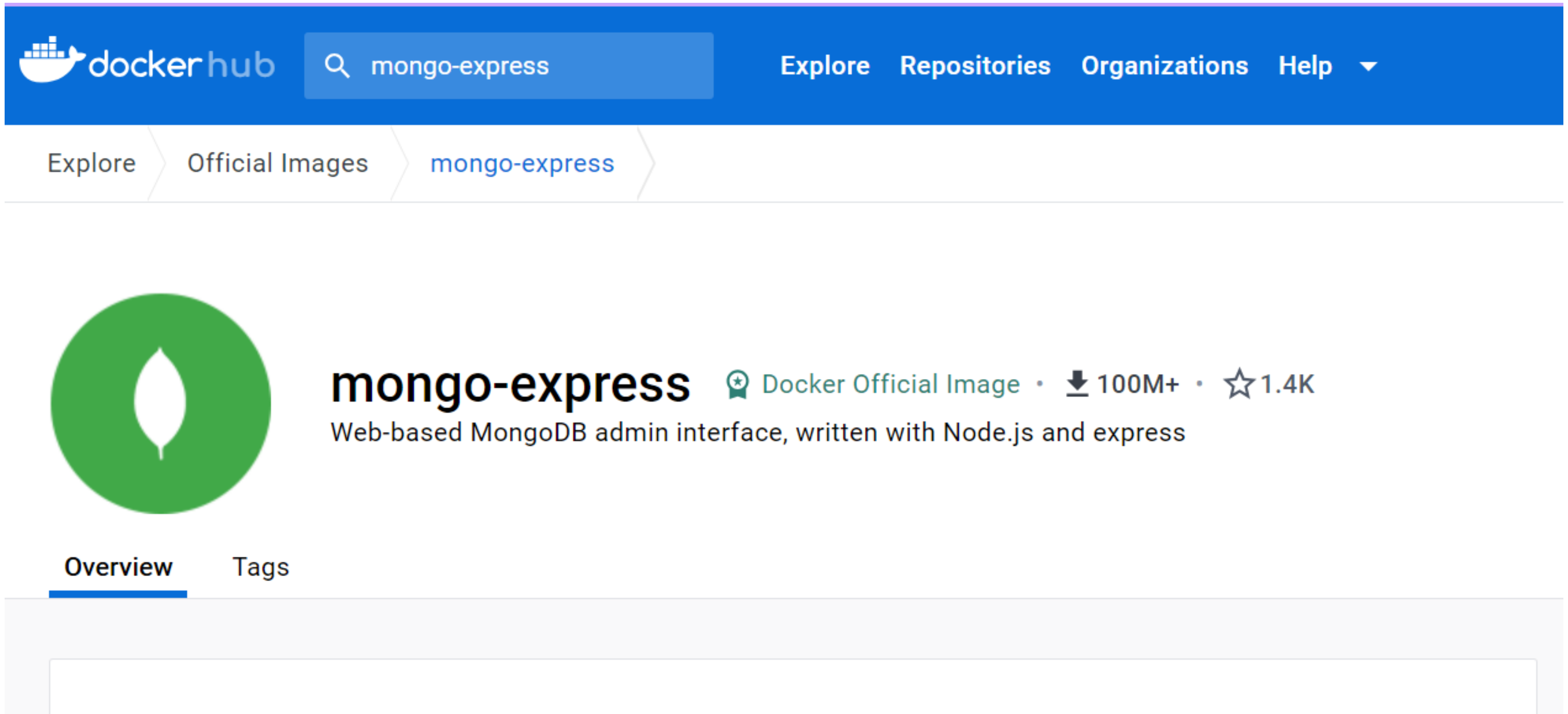


mongo Docker Official Image • 1B+ • 9.9K

MongoDB document databases provide high availability and easy scalability.

至 dockerhub 搜尋 mongo-express

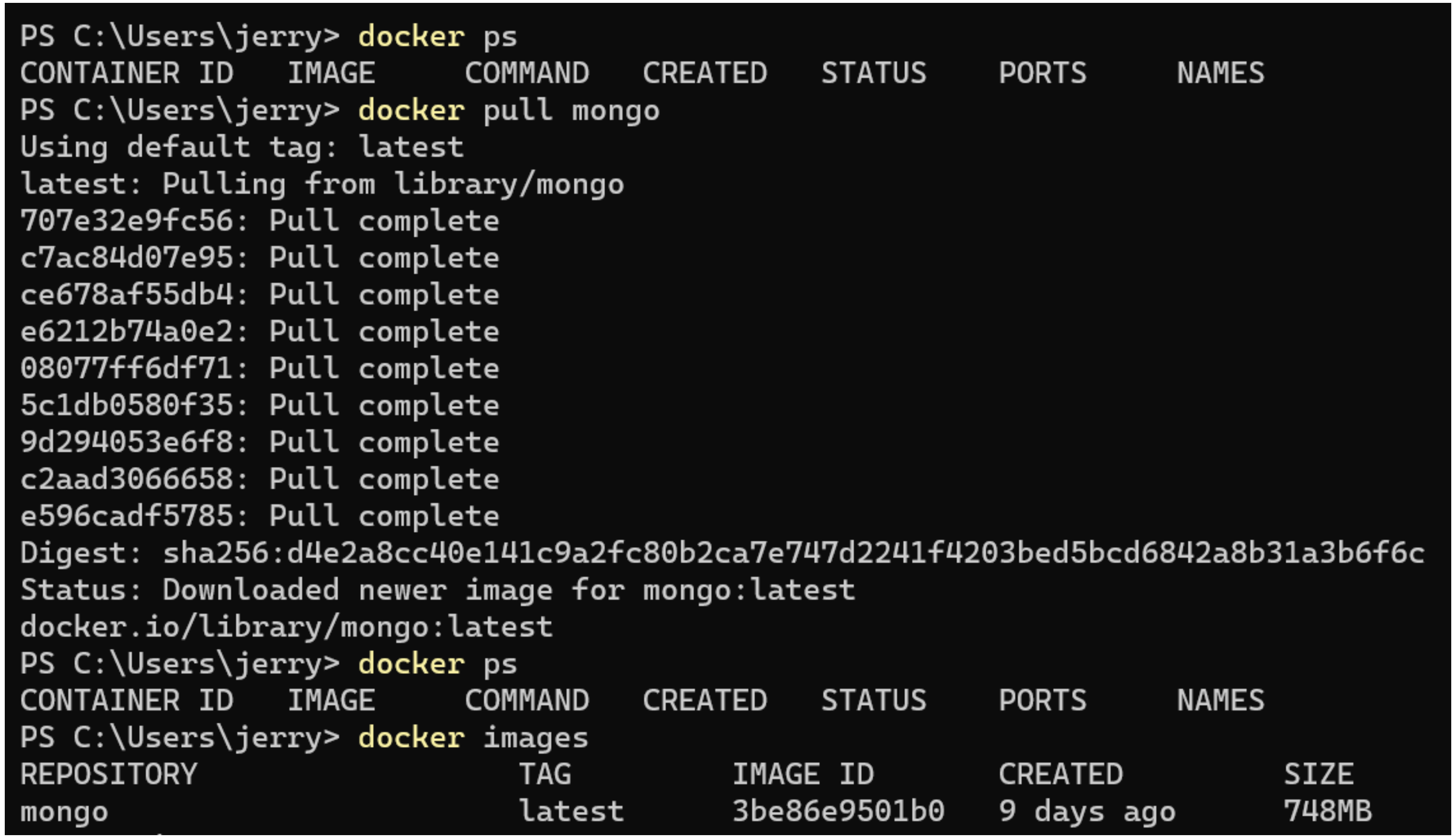
Mongo-Express – 操作 MongoDB 的視覺化界面



mongo-express Docker Official Image • 100M+ • ☆1.4K

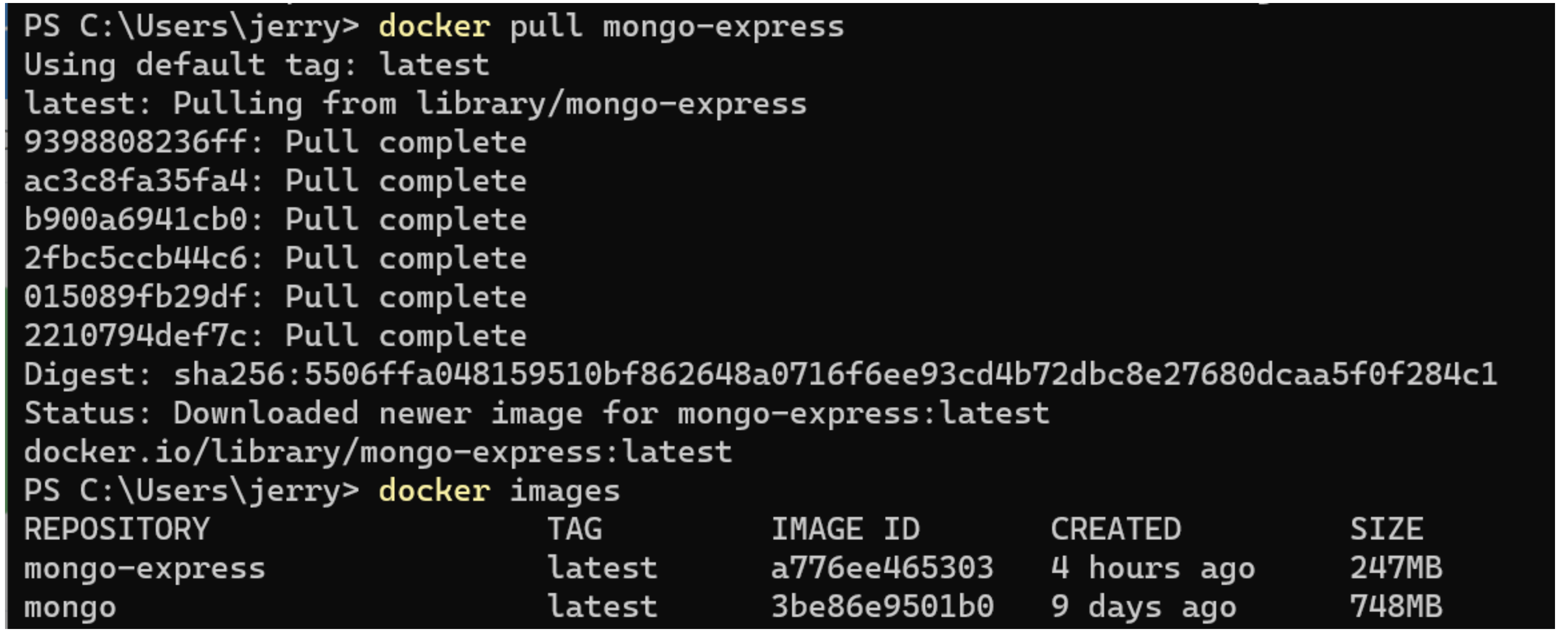
Web-based MongoDB admin interface, written with Node.js and express

pull mongo images to local docker



```
PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\jerry> docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
707e32e9fc56: Pull complete
c7ac84d07e95: Pull complete
ce678af55db4: Pull complete
e6212b74a0e2: Pull complete
08077ff6df71: Pull complete
5c1db0580f35: Pull complete
9d294053e6f8: Pull complete
c2aad3066658: Pull complete
e596cadf5785: Pull complete
Digest: sha256:d4e2a8cc40e141c9a2fc80b2ca7e747d2241f4203bed5bcd6842a8b31a3b6f6c
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\jerry> docker images
REPOSITORY                  TAG         IMAGE ID       CREATED         SIZE
mongo                       latest      3be86e9501b0   9 days ago      748MB
```

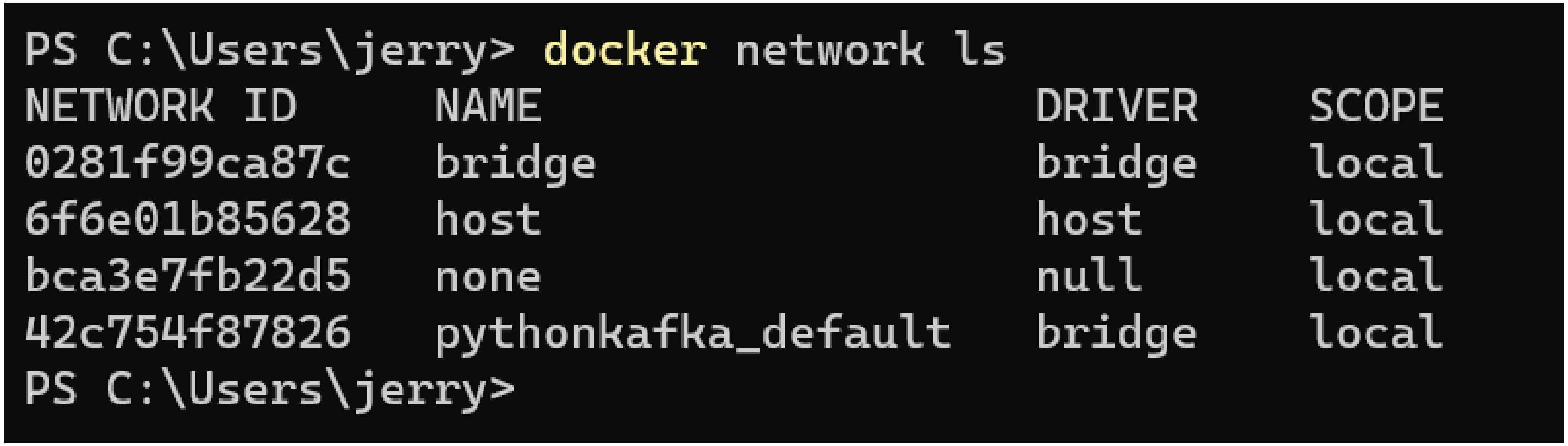
pull mongo-express image to local docker



```
PS C:\Users\jerry> docker pull mongo-express
Using default tag: latest
latest: Pulling from library/mongo-express
9398808236ff: Pull complete
ac3c8fa35fa4: Pull complete
b900a6941cb0: Pull complete
2fbc5ccb44c6: Pull complete
015089fb29df: Pull complete
2210794def7c: Pull complete
Digest: sha256:5506ffa048159510bf862648a0716f6ee93cd4b72dbc8e27680dcaa5f0f284c1
Status: Downloaded newer image for mongo-express:latest
docker.io/library/mongo-express:latest
PS C:\Users\jerry> docker images
REPOSITORY                  TAG         IMAGE ID       CREATED         SIZE
mongo-express               latest      a776ee465303   4 hours ago     247MB
mongo                       latest      3be86e9501b0   9 days ago      748MB
```

MongoDB & Mongo-Express 兩者互動

- 因為 MongoDB & Mongo-Express 個自皆為獨立的 images & container
- 如果兩者要互動連接，必須在同一個網路內才可互相串接

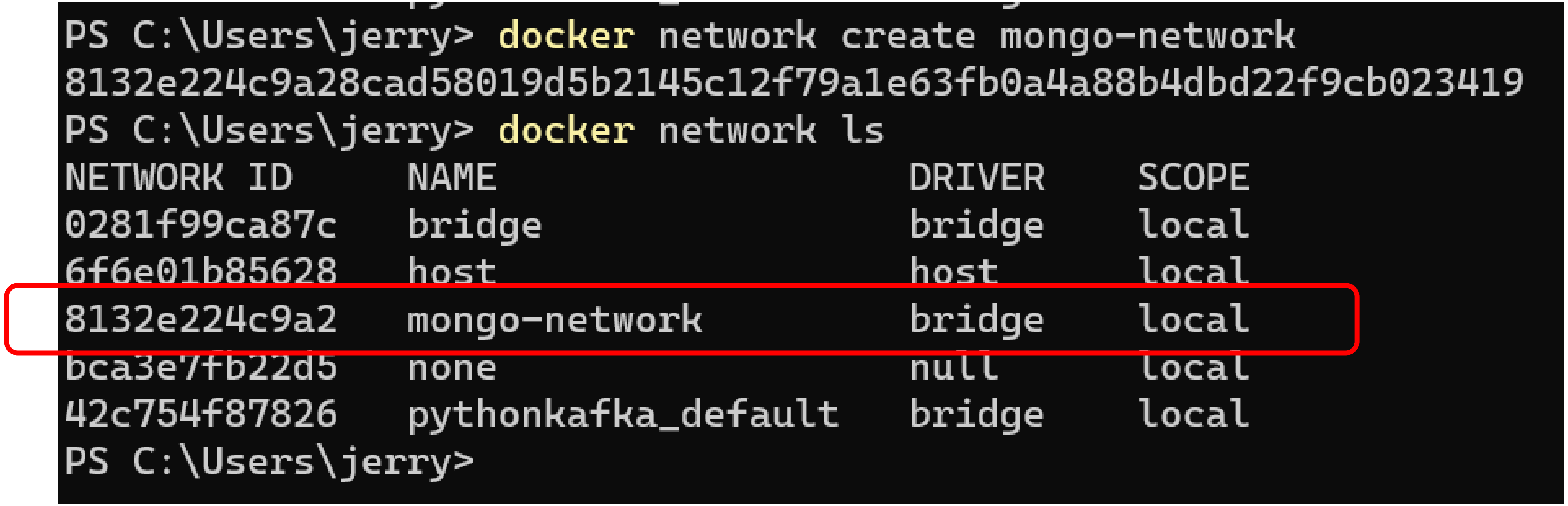


```
PS C:\Users\jerry> docker network ls
NETWORK ID     NAME                   DRIVER    SCOPE
0281f99ca87c   bridge                 bridge    local
6f6e01b85628   host                   host      local
bca3e7fb22d5   none                   null      local
42c754f87826   pythonkafka_default    bridge    local
PS C:\Users\jerry>
```

Command:
docker network ls: 查看當前存在的網路

建立一個網路

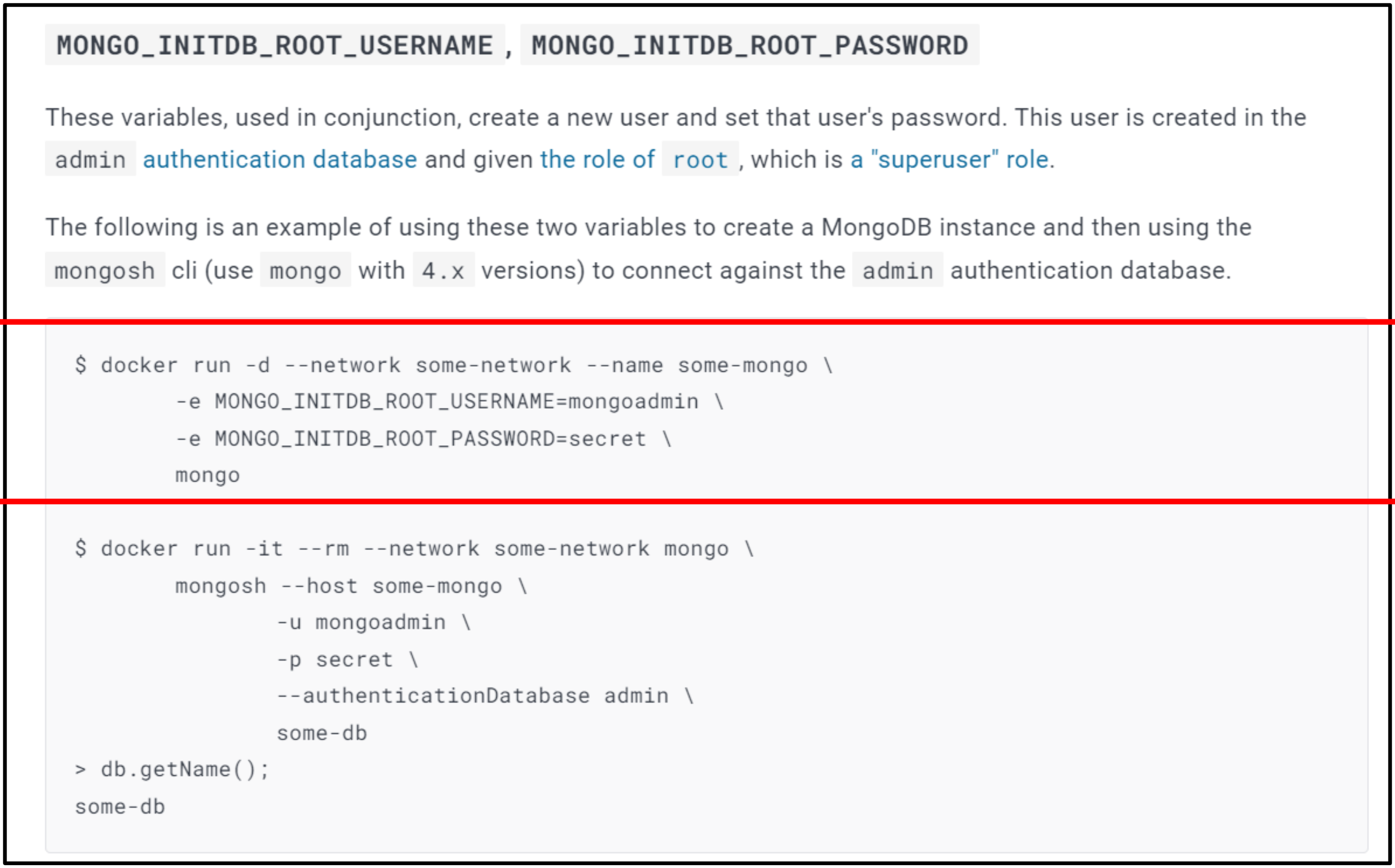
Command:
docker network create <named network>: 建立一個新網路



```
PS C:\Users\jerry> docker network create mongo-network
8132e224c9a28cad58019d5b2145c12f79a1e63fb0a4a88b4dbd22f9cb023419
PS C:\Users\jerry> docker network ls
NETWORK ID     NAME                  DRIVER    SCOPE
0281f99ca87c   bridge                bridge    local
6f6e01b85628   host                  host      local
8132e224c9a2   mongo-network         bridge    local
bca3e7fb22d5   none                  null      local
42c754f87826   pythonkafka_default   bridge    local
PS C:\Users\jerry>
```

可看到已經成功建立一個名為 mongo-network 的網路

至dockerhub的mongoDB查看串接網路範例



`MONGO_INITDB_ROOT_USERNAME` , `MONGO_INITDB_ROOT_PASSWORD`

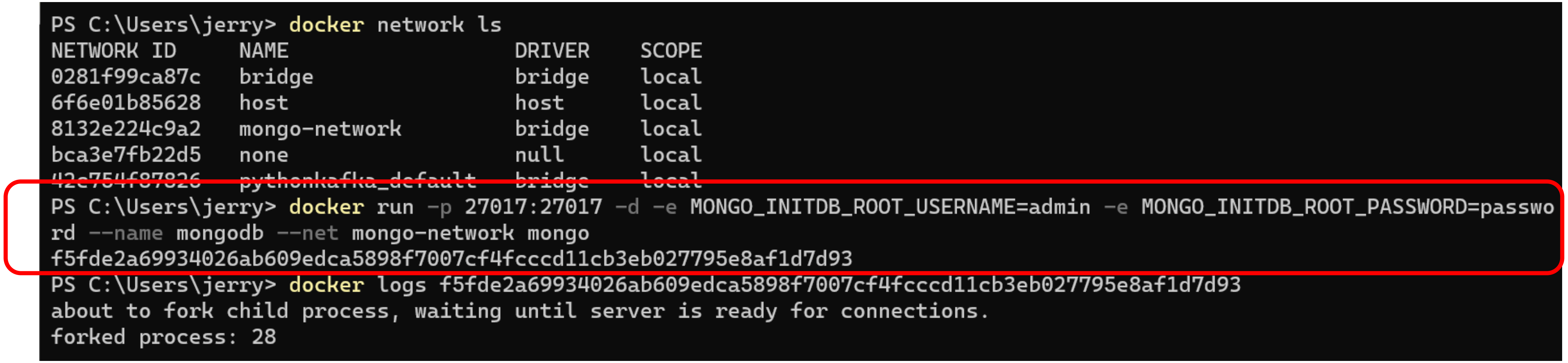
These variables, used in conjunction, create a new user and set that user's password. This user is created in the `admin` authentication database and given the role of `root` , which is a "superuser" role.

The following is an example of using these two variables to create a MongoDB instance and then using the `mongosh` cli (use `mongo` with `4.x` versions) to connect against the `admin` authentication database.

```
$ docker run -d --network some-network --name some-mongo \
	-e MONGO_INITDB_ROOT_USERNAME=mongoadmin \
	-e MONGO_INITDB_ROOT_PASSWORD=secret \
	mongo

$ docker run -it --rm --network some-network mongo \
	mongosh --host some-mongo \
		-u mongoadmin \
		-p secret \
		--authenticationDatabase admin \
		some-db
> db.getName();
some-db
```

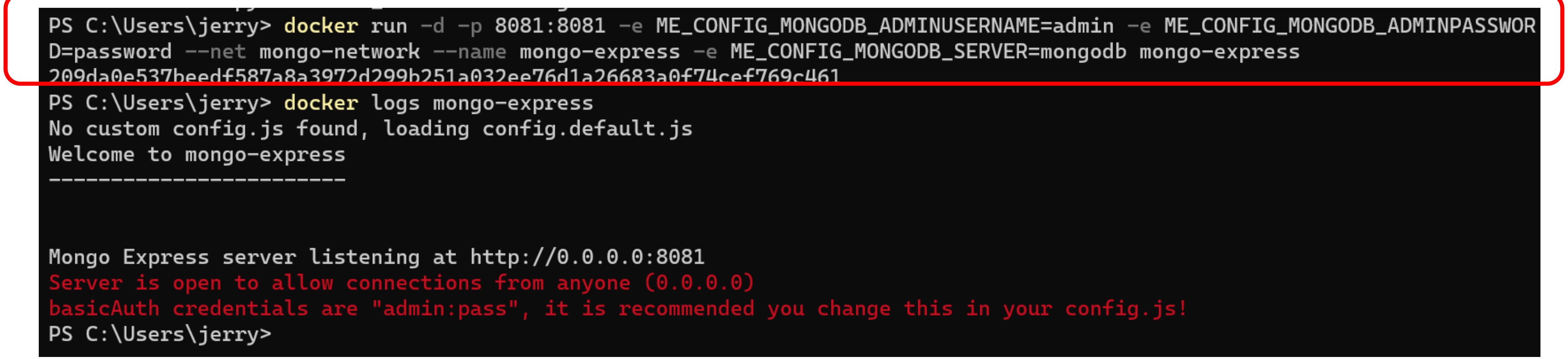
建立MongoDB container並串接至network



```
PS C:\Users\jerry> docker network ls
NETWORK ID     NAME                  DRIVER    SCOPE
0281f99ca87c   bridge                bridge    local
6f6e01b85628   host                  host      local
8132e224c9a2   mongo-network         bridge    local
bca3e7fb22d5   none                  null      local
42c754f87826   pythonkafka_default   bridge    local
PS C:\Users\jerry> docker run -p 27017:27017 -d -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=passwo
rd --name mongodb --net mongo-network mongo
f5fde2a69934026ab609edca5898f7007cf4fcccd11cb3eb027795e8af1d7d93
PS C:\Users\jerry> docker logs f5fde2a69934026ab609edca5898f7007cf4fcccd11cb3eb027795e8af1d7d93
about to fork child process, waiting until server is ready for connections.
forked process: 28
```

- -p: 設定 port
- -d: 建立新的 container
- -e: 設定 MongoDB 環境變數 帳號&密碼
- --name: container name
- --net: 要連線的 network 名稱
- mongo: 要 run 的 image

建立Mongo-Express container並串接至network



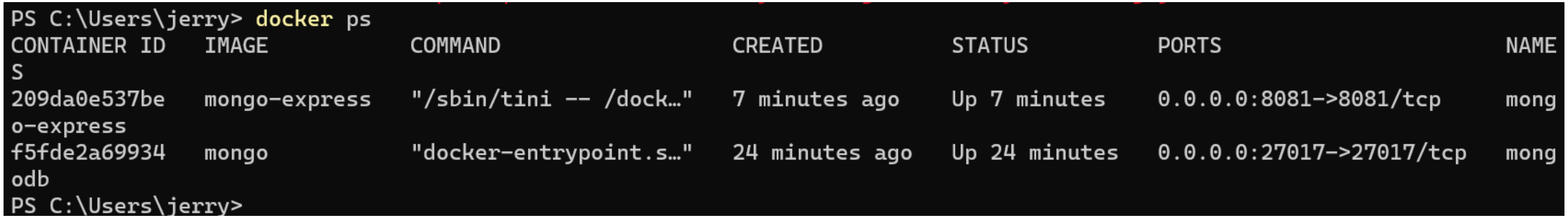
```
PS C:\Users\jerry> docker run -d -p 8081:8081 -e ME_CONFIG_MONGODB_ADMINUSERNAME=admin -e ME_CONFIG_MONGODB_ADMINPASSWOR
D=password --net mongo-network --name mongo-express -e ME_CONFIG_MONGODB_SERVER=mongodb mongo-express
209da0e537beedf587a8a3972d299b251a032ee76d1a26683a0f74cef769c461
PS C:\Users\jerry> docker logs mongo-express
No custom config.js found, loading config.default.js
Welcome to mongo-express
------------------------


Mongo Express server listening at http://0.0.0.0:8081
Server is open to allow connections from anyone (0.0.0.0)
basicAuth credentials are "admin:pass", it is recommended you change this in your config.js!
PS C:\Users\jerry>
```

- -p: 設定 port
- -d: 建立新的 container
- -e: 設定 MongoDB 環境變數 帳號&密碼
- --name: container name
- --net: 要連線的 network 名稱
- mongo: 要 run 的 image

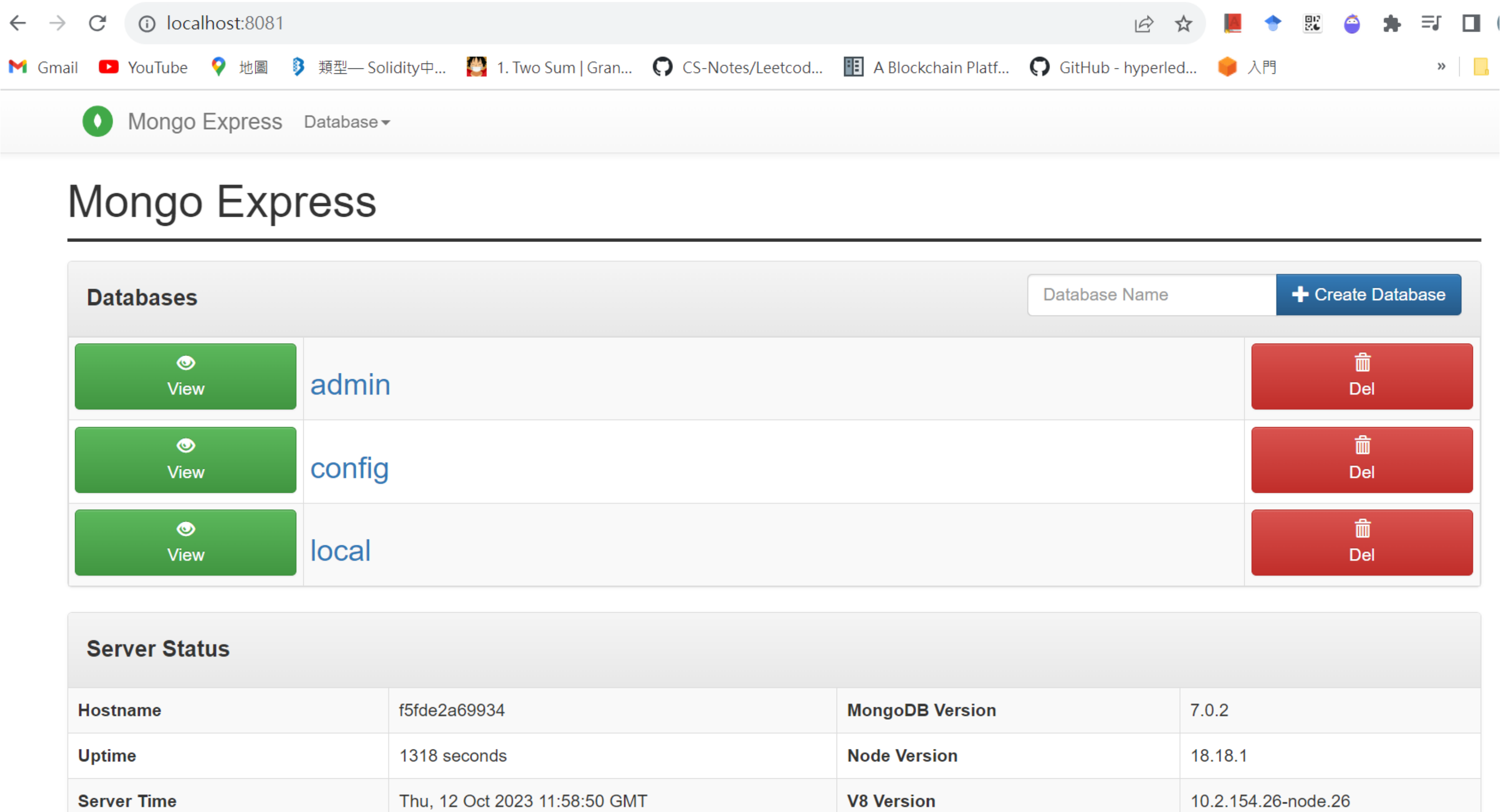
查看當前運行中的container

可看到 mongodb & mongo-express 皆成功執行中



```
PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED          STATUS          PORTS                          NAME
S
209da0e537be   mongo-express   "/sbin/tini -- /dock…"   7 minutes ago    Up 7 minutes    0.0.0.0:8081->8081/tcp         mong
o-express
f5fde2a69934   mongo           "docker-entrypoint.s…"   24 minutes ago   Up 24 minutes   0.0.0.0:27017->27017/tcp       mong
odb
PS C:\Users\jerry>
```

開啟網頁測試



Mongo Express

Databases

View	admin	Del
View	config	Del
View	local	Del

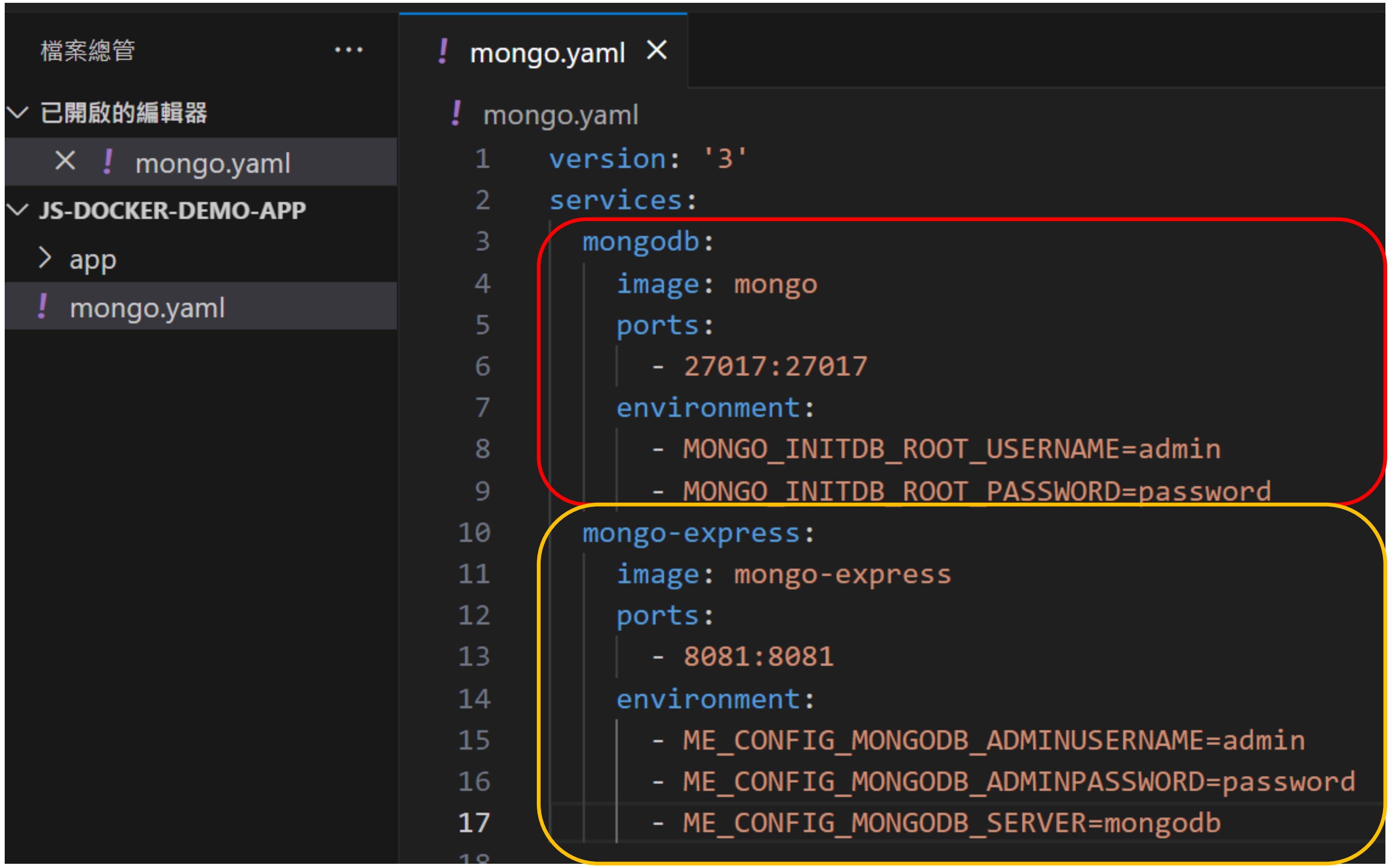
Server Status

Hostname	f5fde2a69934	**MongoDB Version**	7.0.2
Uptime	1318 seconds	**Node Version**	18.18.1
Server Time	Thu, 12 Oct 2023 11:58:50 GMT	**V8 Version**	10.2.154.26-node.26

3. 撰寫.yaml 使用docker-compose整合

將前面指令所包含的參數整合寫進.yaml

使用docker-compose不需要自己建立network，啟動時會自動建立



```yaml
version: '3'
services:
  mongodb:
    image: mongo
    ports:
      - 27017:27017
    environment:
      - MONGO_INITDB_ROOT_USERNAME=admin
      - MONGO_INITDB_ROOT_PASSWORD=password
  mongo-express:
    image: mongo-express
    ports:
      - 8081:8081
    environment:
      - ME_CONFIG_MONGODB_ADMINUSERNAME=admin
      - ME_CONFIG_MONGODB_ADMINPASSWORD=password
      - ME_CONFIG_MONGODB_SERVER=mongodb
```

version: 指定docker-compose的版本

mongodb

```
docker run -d -p 27017:27017 \
-e MONGO_INITDB_ROOT_USERNAME=admin \
-e MONGO_INITDB_ROOT_PASSWORD=password \
--name mongodb \
mongo
```

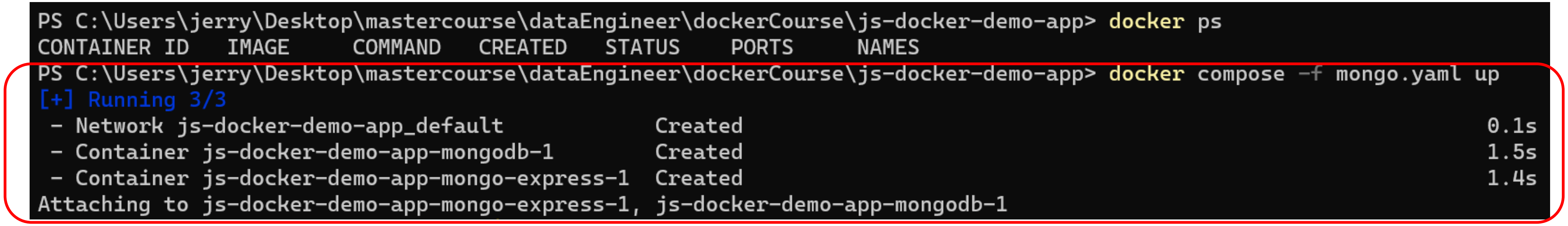
mongo-express

```
docker run -d -p 8081:8081 \
-e ME_CONFIG_MONGODB_ADMINUSERNAME=admin \
-e ME_CONFIG_MONGODB_ADMINPASSWORD=password \
--name mongo-express \
-e ME_CONFIG_MONGODB_SERVER=mongodb \
mongo-express
```

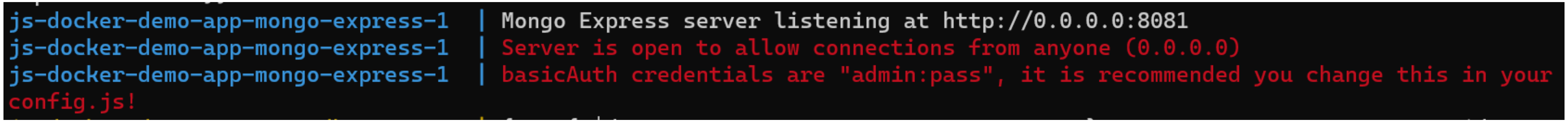
docker-compose up

Command:

- docker compose –f <.yaml> up: 同時啟動多個containe並建立network
- -f: 指定要執行的檔案

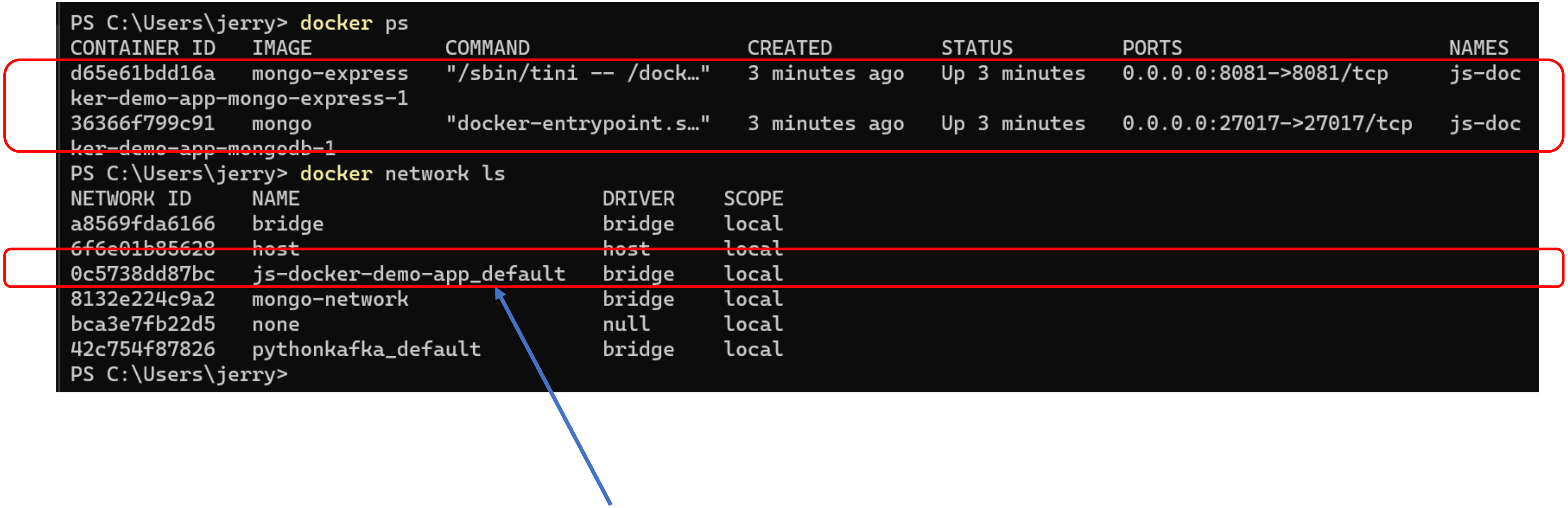


```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker compose -f mongo.yaml up
[+] Running 3/3
 - Network js-docker-demo-app_default              Created                                                        0.1s
 - Container js-docker-demo-app-mongodb-1          Created                                                        1.5s
 - Container js-docker-demo-app-mongo-express-1    Created                                                        1.4s
Attaching to js-docker-demo-app-mongo-express-1, js-docker-demo-app-mongodb-1
```



```
js-docker-demo-app-mongo-express-1  | Mongo Express server listening at http://0.0.0.0:8081
js-docker-demo-app-mongo-express-1  | Server is open to allow connections from anyone (0.0.0.0)
js-docker-demo-app-mongo-express-1  | basicAuth credentials are "admin:pass", it is recommended you change this in your
config.js!
```

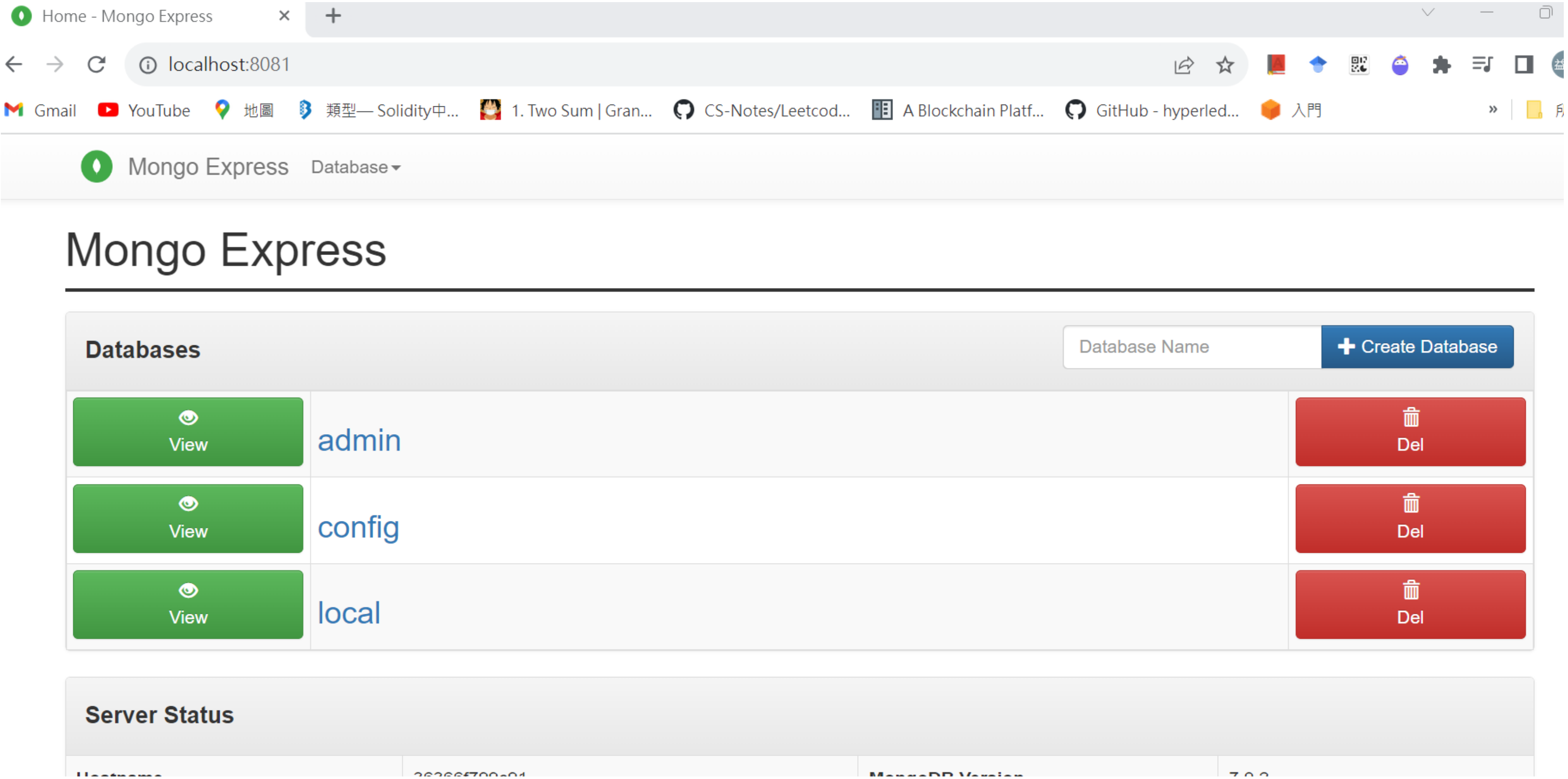
查看當前運行中 container & network



```
PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS                      NAMES
d65e61bdd16a   mongo-express   "/sbin/tini -- /dock…"   3 minutes ago   Up 3 minutes   0.0.0.0:8081->8081/tcp     js-doc
ker-demo-app-mongo-express-1
36366f799c91   mongo           "docker-entrypoint.s…"   3 minutes ago   Up 3 minutes   0.0.0.0:27017->27017/tcp   js-doc
ker-demo-app-mongodb-1
PS C:\Users\jerry> docker network ls
NETWORK ID     NAME                         DRIVER    SCOPE
a8569fda6166   bridge                       bridge    local
6f6e01b85628   host                         host      local
0c5738dd87bc   js-docker-demo-app_default   bridge    local
8132e224c9a2   mongo-network                bridge    local
bca3e7fb22d5   none                         null      local
42c754f87826   pythonkafka_default          bridge    local
PS C:\Users\jerry>
```

network

開啟網頁測試



Mongo Express

Databases

admin

config

local

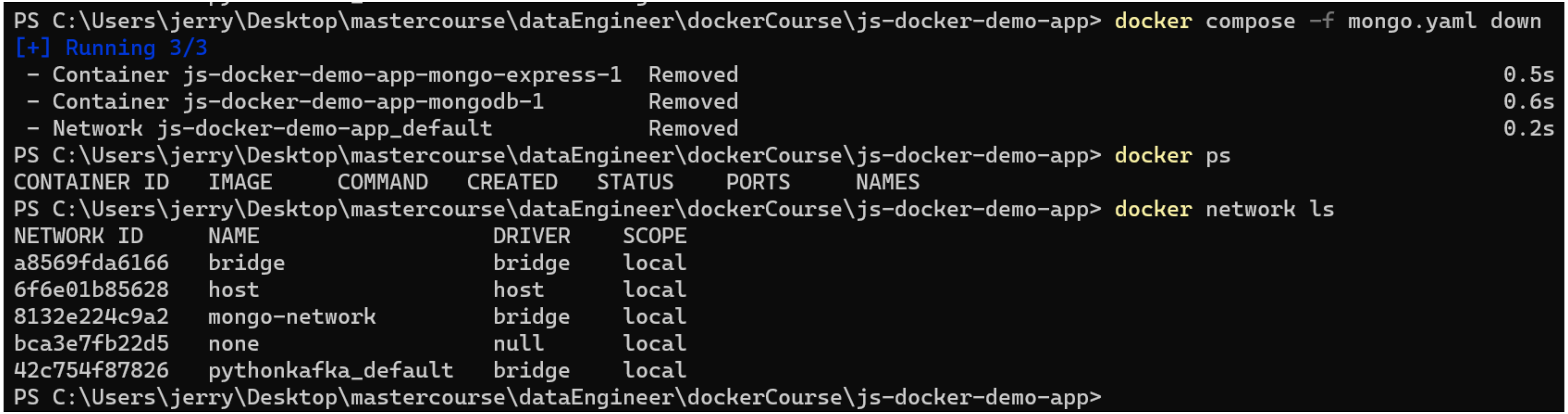
Server Status

Hostname	36366f799c91	MongoDB Version	7.0.2

docker compose down

Command:

- docker compose –f <.yaml> down: 同時關閉刪除containe & network
- -f: 指定要執行的檔案



```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker compose -f mongo.yaml down
[+] Running 3/3
 - Container js-docker-demo-app-mongo-express-1  Removed                                                              0.5s
 - Container js-docker-demo-app-mongodb-1        Removed                                                              0.6s
 - Network js-docker-demo-app_default            Removed                                                              0.2s
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker network ls
NETWORK ID     NAME                 DRIVER    SCOPE
a8569fda6166   bridge               bridge    local
6f6e01b85628   host                 host      local
8132e224c9a2   mongo-network        bridge    local
bca3e7fb22d5   none                 null      local
42c754f87826   pythonkafka_default  bridge    local
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app>
```

End