

# Push docker image to AWS ECR

郭益華

# 目錄

1. [簡介說明](#)
2. [Image 推送至 AWS ECR](#)
3. [Deploy our containerized app](#)

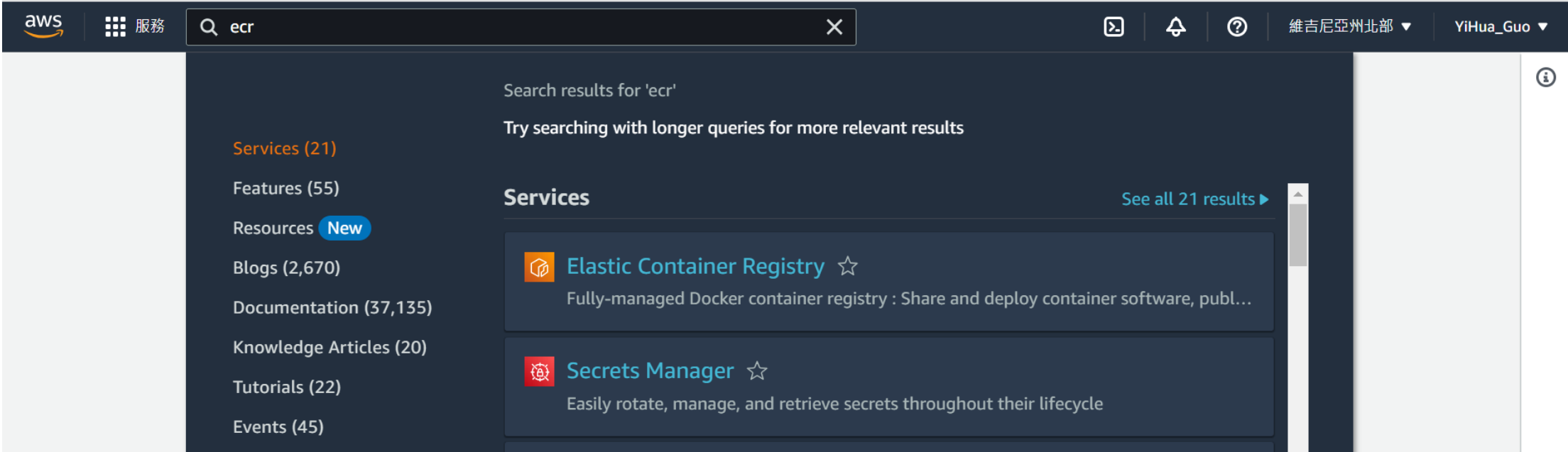
# 1. 簡介說明

# 説明

- Private Docker Repository - Pushing our built Docker Image into a private Registry on AWS
- Deploy our containerized app

## 2. Image 推送至AWS ECR

# 搜尋 ecr



# 點選紅框”入門”選項

容器

## Amazon Elastic Container Registry

以公有或私有方式共享和部署  
容器軟體

Amazon Elastic Container Registry (ECR) 是一個全受管容器登錄檔，可隨時隨地輕鬆存放、管理、共享和部署您的容器映像和成品。

建立儲存庫

入門

定價 (美國)

僅依據公有或私有儲存庫中存放的資料量和傳輸至網際網路的資料量進行計費。

[進一步了解](#)

運作方式

# 建立 私有儲存庫

勾選 私有

命名

Amazon ECR > 儲存庫 > 建立儲存庫

## 建立儲存庫

### 一般設定

可視性設定

資訊

選擇儲存庫的可視性設定。

☒ 私有  
存取透過 IAM 和儲存庫政策許可進行管理。

☐ 公有  
對映像提取公開可見和可存取。

儲存庫名稱  
提供簡明的名稱。開發人員應能夠透過名稱識別儲存庫內容。

288142612271.dkr.ecr.us-east-1.amazonaws.com/ my-app

6/256 個字元上限 (2 下限)。The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

標籤不可變性

資訊

啟用標籤不可變性，以防止映像標籤被使用相同標籤的後續映像推送覆寫。停用標籤不可變性以允許覆寫映像標籤。

☐ 已停用



# 建立成功畫面

☑ 已成功建立儲存庫 my-app

檢視推送命令

×

[Amazon ECR](#) > 儲存庫


PrivatePublic

私有儲存庫 (1)

↺檢視推送命令刪除動作▼建立儲存庫

🔍 尋找儲存庫

< 1 > ⚙

<input type="checkbox"/>	儲存庫名稱 ▲	URI	建立時間 ▼	標籤不可變性	掃描頻率	加密類型	提取式快取
<input type="checkbox"/>	my-app	 288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app	2023年10月14日, 09:44:41 (UTC+08)	已停用	手動	AES-256	非作用中

點選進入

# 點選 檢視推送命令

Amazon ECR > 儲存庫 > my-app

## my-app

檢視推送命令

編輯

映像 (0)

🔄

刪除

詳細資訊

掃描

🔍 尋找映像

< 1 > ⚙️

	映像標籤 ▼	成品類型	推送時間 ▼	大小 (MB) ▼	映像 URI	摘要	掃描狀態	漏洞
No images No images to display								

# 依照作業系統不同使用相對應的命令

## 推送命令

推送 my-app 的命令

macOS / Linux

Windows

確認您已安裝最新版 **AWS CLI** 和 **Docker**。如需詳細資訊，請參閱 [Amazon ECR 入門](#)。

使用以下步驟來驗證映像並將映像推送至您的儲存庫。如需了解其他登錄檔驗證方法 (包括 Amazon ECR 憑證協助程式)，請參閱 [登錄檔驗證](#)。

1. 擷取驗證字符並將 Docker 用戶端驗證至您的登錄檔。  
使用 AWS CLI：  

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 288142612271.dkr.ecr.us-east-1.amazonaws.com
```

注意：如果您在使用 AWS CLI 時收到錯誤，請確認您已安裝最新版 AWS CLI 和 Docker。
2. 使用下列命令建置您的 Docker 映像。如需有關從頭開始建置 Docker 檔案的資訊，請參閱 [這裡](#) 說明。如果您的映像已建置，則可以跳過此步驟：  

```
docker build -t my-app .
```
3. 建置完成後，將您的映像加上標籤，以便將映像推送至此儲存庫：  

```
docker tag my-app:latest 288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app:latest
```
4. 執行下列命令將此映像推送至新建立的 AWS 儲存庫：

# 查看當前存在的 image

```
PS C:\Users\jerry> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-app	1.0	2de504f8fd65	14 hours ago	126MB
mongo-express	latest	a776ee465303	43 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

要推送的 image

# 依照推送命令執行

```
PS C:\Users\jerry> (Get-ECRLLoginCommand).Password | docker login --username AWS --password-stdin 288142612271.dkr.ecr.us-east-1.amazonaws.com  
Login Succeeded
```

```
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
```

# 複製一份my-app 的 image

```
PS C:\Users\jerry> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
my-app              1.0                2de504f8fd65       15 hours ago       126MB
mongo-express        latest             a776ee465303       44 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       8 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> docker tag my-app:1.0 288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app:1.0
```

```
PS C:\Users\jerry> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app	1.0	2de504f8fd65	15 hours ago	126MB
my-app	1.0	2de504f8fd65	15 hours ago	126MB

建立成功

# Push 至 AWS ECR

```
PS C:\Users\jerry> docker push 288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app:1.0
The push refers to repository [288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app]
1c5bd2d52c4c: Pushed
5f70bf18a086: Pushed
a5d5e42ffe64: Pushed
c4389ef970ca: Pushed
629960860aca: Pushed
f019278bad8b: Pushed
8ca4f4055a70: Pushed
3e207b409db3: Pushed
1.0: digest: sha256:8358fd0db3b221f869d6baa477362a65577ba1a43105fa5761a5737ce313cae6 size: 1993
```

# 回到AWS ECR儲存庫畫面

點選

Amazon ECR > 儲存庫

Private Public

私有儲存庫 (1)

檢視推送命令 刪除 動作 ▼ 建立儲存庫

尋找儲存庫

<input type="checkbox"/>	儲存庫名稱 ▲	URI	建立時間 ▼	標籤不可變性	掃描頻率	加密類型	提取式快取
<input type="checkbox"/>	my-app	288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app	2023年10月14日, 10:35:29 (UTC+08)	已停用	手動	AES-256	非作用中

成功推送至 AWS ECR



# 進入後可看到tag 1.0

Amazon ECR > 儲存庫 > my-app

my-app 檢視推送命令 編輯

映像 (1) 刷新 刪除 詳細資訊 掃描

🔍 尋找映像 < 1 > ⚙️

<input type="checkbox"/>	映像標籤 ▼	成品類型	推送時間 ▼	大小 (MB) ▼	映像 URI	摘要	掃描狀態	漏洞
<input type="checkbox"/>	1.0	Image	2023年 10月14 日, 11:05:38 (UTC+08)	48.53	複製 URI	sha256:8358fd0...	-	-

# 3. Deploy our containerized app

# 修改 .yaml

```
... ! mongo.yaml X Dockerfile
! mongo.yaml
1  version: '3'
2  ∨ services:
3  ∨   myapp:
4    image: 288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app
5  ∨   mongodb:
6    image: mongo
7  ∨   ports:
8    - 27017:27017
9  ∨   environment:
10   - MONGO_INITDB_ROOT_USERNAME=admin
11   - MONGO_INITDB_ROOT_PASSWORD=password
12  ∨   mongo-express:
13    image: mongo-express
14  ∨   ports:
15    - 8081:8081
16  ∨   environment:
17    - ME_CONFIG_MONGODB_ADMINUSERNAME=admin
18    - ME_CONFIG_MONGODB_ADMINPASSWORD=password
19    - ME_CONFIG_MONGODB_SERVER=mongodb
20
```

私有儲存庫 (1) 檢視推送命令 刪除 動作 ▼ 建立儲存庫

🔍 尋找儲存庫 < 1 > ⚙️

<input type="checkbox"/>	儲存庫名稱 ▲	URI	建立時間 ▼	標籤不可變性	掃描頻率	加密類型	提取式快取
<input type="checkbox"/>	my-app	288142612271.dkr.ecr.us-east-1.amazonaws.com/my-app	2023年10月14日, 10:35:29 (UTC+08)	已停用	手動	AES-256	非作用中

新增此image的位置

# 查看相關命令

推送 my-app 的命令



macOS / Linux

Windows

確認您已安裝最新版 **AWS Tools for PowerShell** 和 **Docker**。如需詳細資訊，請參閱 [Amazon ECR 入門](#)。

使用以下步驟來驗證映像並將映像推送至您的儲存庫。如需了解其他登錄檔驗證方法 (包括 Amazon ECR 憑證協助程式)，請參閱 [登錄檔驗證](#)。

1. 擷取驗證字符並將 Docker 用戶端驗證至您的登錄檔。

使用 AWS Tools for PowerShell：

```
(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin 288142612271.dkr.ecr.us-east-1.amazonaws.com
```

登入指令

# 成功運行AWS ECR上的image並串聯DB

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> (Get-ECRLoginCommand).Password | do  
cker login --username AWS --password-stdin 288142612271.dkr.ecr.us-east-1.amazonaws.com  
Login Succeeded
```

登入指令

```
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/  
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app>
```

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> (Get-ECRLoginCommand).Password | do  
cker login --username AWS --password-stdin 288142612271.dkr.ecr.us-east-1.amazonaws.com  
Login Succeeded
```

```
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
```

```
PS C:\Users\jerry\Desktop\mastercourse\dataEngineer\dockerCourse\js-docker-demo-app> docker-compose -f mongo.yaml up
```

```
Creating js-docker-demo-app_myapp_1 ... done
```

```
Creating js-docker-demo-app_mongodb_1 ... done
```

```
Creating js-docker-demo-app_mongo-express_1 ... done
```

```
Attaching to js-docker-demo-app_mongo-express_1, js-docker-demo-app_mongodb_1, js-docker-demo-app_myapp_1
```

Docekr-compose 執行 .yaml

# 開啟測試

Home - Mongo Express x +

localhost:8081







Gmail YouTube 地圖 類型— Solidity中... 1. Two Sum | Gran... CS-Notes/Leetcod... A Blockchain Platf... GitHub - hyperled... 入門 所有書

Mongo Express Database ▾

## Mongo Express

Databases

Database Name  + Create Database

 View	admin	 Del
 View	config	 Del
 View	local	 Del

Server Status

**End**