



# 軟體開發安全

# Security of Software Development

## 9. Toward to Secure Software Development Life Cycle (SSDLC) (3) – Security Deployment with CI/CD

**Chih-Hung Wang**

2023/10/19





# Outline

- Secure Software Development Life Cycle
  - **Security Deployment**
  - **Security Deployment CD Tools**
  - **Security Deployment for Docker Hub**
- Other CI/CD Deployment Platforms
  - Cloud Run - Google Cloud Platform (GCP)
  - Docker Swarm
  - AWS
  - Kubernetes
  - ...





# Security Deployment

## Automated (Continuous) Deployment





# Security Deployment (1)

- 安全部署實務

- 透過 Ring 進行安全部署

- 隨著平臺的增長，基礎設施的規模和需求也趨於成長。因此提高了對部署模型的需求，該模型平衡了**新部署相關的風險**和其**更新的好處**。
    - 一般的想法是，給定的版本**首先應該只給予在對風險容忍度最高的一小群使用者面前**。然後，如果版本按預期工作，它可以給予給更廣泛的使用者群體。
    - 如果沒有問題，那麼這個過程可以透過更廣泛的使用者群體或環繼續進行，直到每個人都在使用新版本。
    - 可藉助 GitHub Actions 和/或 Azure Pipelines 等**現代持續交付平臺**，任何規模的 DevOps 團隊都可以使用環構建部署流程。



<https://learn.microsoft.com/en/devops/operate/safe-deployment-practices>



# Security Deployment (2)

- 通常同時使用多種做法。例如，一個團隊可能有一個針對非常特定使用的實驗功能。由於風險高，他們將將其部署到第一個環上，**供內部使用者試用**。
- 任何不參與該部署或尚未選擇加入的人都不會接觸到該功能。
- 當一個團隊每年**只部署幾次**時，自動化交付似乎不值得投資。因此，許多部署過程都是手動管理的。這需要大量的時間和精力，並且容易出現人為錯誤。簡單地自動化最常見的構建和部署任務可以大大減少損失的時間和非強制錯誤。

<https://learn.microsoft.com/en/devops/operate/safe-deployment-practices>

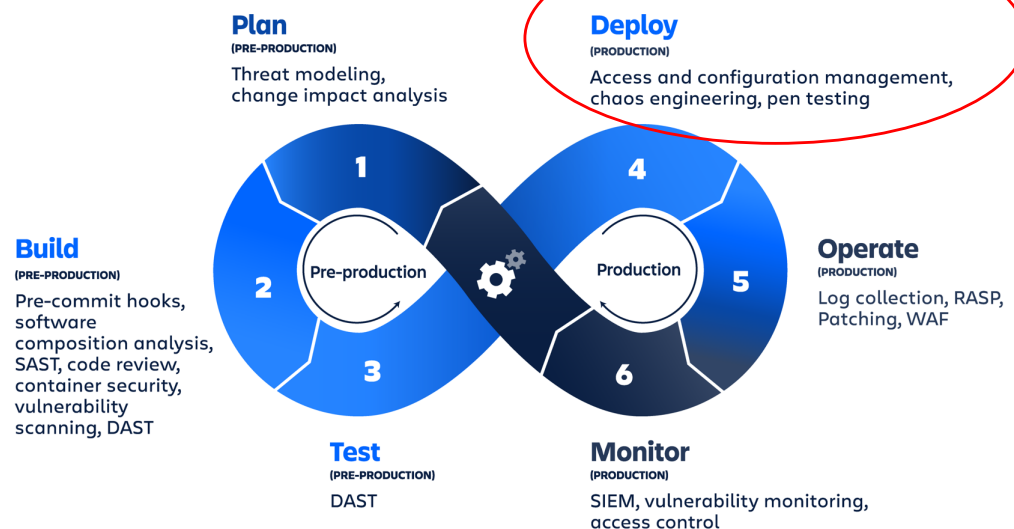


# Security Deployment (3)



- 啟動部署
  - 過去一些組織的政策要求所有部署都由運營人員發起和管理。
  - 當開發團隊可以啟動和控制部署時，敏捷式 DevOps 流程會有許多的優勢。
  - 現代持續交付平臺可以精細控制誰可以啟動哪些部署以及誰可以訪問狀態日誌和其他診斷資訊，以確保正確的人員可以儘快獲得正確的資訊。

## DevSecOps



圖來源：

<https://www.atlassian.com/devops/devops-tools/devsecops-tools>

<https://learn.microsoft.com/en/devops/operate/safe-deployment-practices>





# Security Deployment (4)

- Core principles (自動化部屬的核心原則)
  - Be consistent (一致性)
  - Care about quality signals (注意品質訊號)
  - Deployments should require **zero downtime** (部署應該要能夠零停機)
    - 現代基礎設施和 Pipeline 工具現在足夠先進，幾乎任何團隊都可以實現100%的正常執行時間。
  - Deployments should happen during working hours (部署應該在工作時間進行)
    - 部署特別是在白天早些時候和一週早些時候。因為如果出了問題，應該儘早追蹤，以控制危害範圍。
  - Ring-based deployment (Ring 為主的部署)
  - Allow bake time (允許一些製作時間)
  - Expedite hotfixes (加速緊急修復)

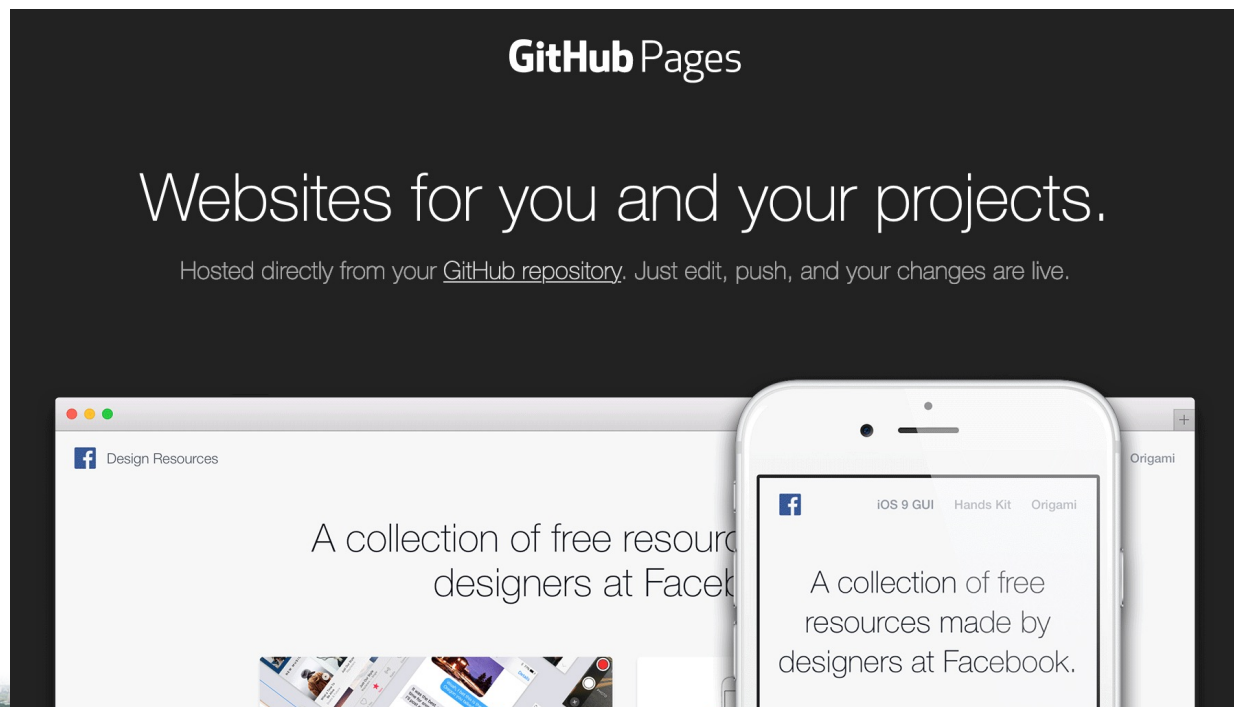






# Security Deployment CD Tools (1)

- GitHub Actions
  - 透過 Actions 進行靜態網頁部署
  - GitHub Pages (<https://pages.github.com/>)







# Security Deployment CD Tools (2)

- 簡單部署方式
  - 創建一個新的 repository: github-pages-test
  - 建立新的 branch: mypages
  - 在 Settings-> Pages 進行修改

Moderation options

Code and automation

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Environments
- Codespaces
- Pages**

Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the mypages branch publishing source for your site.

mypages / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

Custom domain

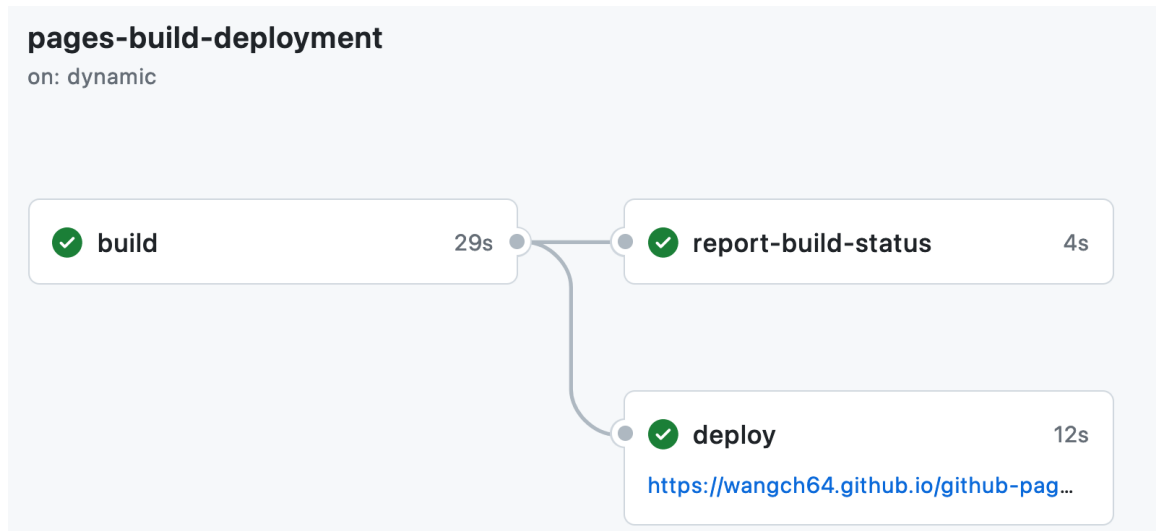


# Security Deployment CD Tools (3)



- Git Clone github-pages-test 到本地端
- 利用 push 到 mypages 進行自動部署
  - 加入靜態網頁資訊，如 index.html
  - git add .
  - git commit -m "commit"
  - git checkout -b mypages
  - git push -u origin mypages

```
(one@kali)-[~/github-pages-test]
$ git push -u origin mypages
枚舉物件：4，完成。
物件計數中：100% (4/4)，完成。
使用 2 個執行緒進行壓縮
壓縮物件中：100% (3/3)，完成。
寫入物件中：100% (3/3)，475 位元組 | 475.00 KiB/s，完成。
總共 3 (差異 0)，復用 0 (差異 0)，重用包 0
To github.com:wangch64/github-pages-test.git
b274f4e..0fd6e0a mypages → mypages
已將 "mypages" 分支設定為追蹤 "origin/mypages"。
```





# Security Deployment CD Tools (4)

- 查看 Actions

```
build
succeeded 5 minutes ago in 29s

> ✓ Set up job
> ✓ Pull ghcr.io/actions/jekyll-build-pages:v1.0.8
> ✓ Checkout
✓ Build with Jekyll

1 ▶ Run actions/jekyll-build-pages@v1
9 /usr/bin/docker run --name ghcr.io/actions/jekyll-build-pages:v1.0.8_510e20
--workdir /github/workspace --rm -e "INPUT_SOURCE" -e "INPUT_DESTIN
"INPUT_TOKEN" -e "INPUT_FUTURE" -e "INPUT_BUILD_REVISION" -e "INPUT
-e "GITHUB_JOB" -e "GITHUB_REF" -e "GITHUB_SHA" -e "GITHUB_REPOSITOI
"GITHUB_REPOSITORY_OWNER" -e "GITHUB_REPOSITORY_OWNER_ID" -e "GITHUI
"GITHUB_RUN_NUMBER" -e "GITHUB_RETENTION_DAYS" -e "GITHUB_RUN_ATTEM
```

```
✓ Deploy to GitHub Pages

1 ▶ Run actions/deploy-pages@v2
9 Artifact exchange URL: https://pipelinesghubeus25.actions
/6p3rxQHQ2Lj8P8CpGFW702BDab0W0yct0qhETKKvIgs160hqzr/_apis
/artifacts?api-version=6.0-preview
10 Creating Pages deployment with payload:
11 {
12     "artifact_url": "https://pipelinesghubeus25.actio
/6p3rxQHQ2Lj8P8CpGFW702BDab0W0yct0qhETKKvIgs160hqzr/_apis
/2/artifacts?artifactName=github-pages&%24expand=SignedCo
13     "pages_build_version": "0fd6e0aa8ac11cab1056aa478
14     "oidc_token": "***"
15 }
```





# Security Deployment CD Tools (5)

- 查看部署情况 <https://wangch64.github.io/github-pages-test/>

**Welcome to my TestSite**

- 嘗試更動網頁，**push** 後可自動重新部署新資料！





# Security Deployment CD Tools (6)

- Using Actions File (repository: github-pages-test2)

The screenshot shows the GitHub Actions interface. On the left is a sidebar with the following menu items: Code and automation, Security, Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages (which is highlighted). The main content area is titled 'Source' and features a dropdown menu set to 'GitHub Actions' with a 'Send feedback' link. Below this, it says 'Use a suggested workflow, [browse all workflows](#), or [create your own](#).' Two workflow cards are displayed: 'GitHub Pages Jekyll' by GitHub Actions, described as 'Package a Jekyll site with GitHub Pages dependencies preinstalled.', and 'Static HTML' by GitHub Actions, described as 'Deploy static files in a repository without a build.' Both cards have a 'Configure' button. A red oval highlights the 'Source' section and the two workflow cards. At the bottom, it says 'Workflow details will appear here once your site has been deployed. [View workflow runs](#)'.



# Security Deployment CD Tools (7)



- 建構 jekyll.yml 或是 static.yml (是否需要 build)
- 當 push 時 (在 main) 即可部署

```
# Sample workflow for building and deploying a Jekyll site to GitHub Pages
name: Deploy Jekyll with GitHub Pages dependencies preinstalled

on:
  # Runs on pushes targeting the default branch
  push:
    branches: ["main"]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# Sets permissions of the GITHUB_TOKEN to allow deployment to GitHub Pages
permissions:
  contents: read
  pages: write
  id-token: write

# Allow only one concurrent deployment, skipping runs queued between the run in-progress
# and latest queued.
# However, do NOT cancel in-progress runs as we want to allow these production deployments to complete.
concurrency:
  group: "pages"
  cancel-in-progress: false
```

jekyll.yml

on: push



build

28s



deploy

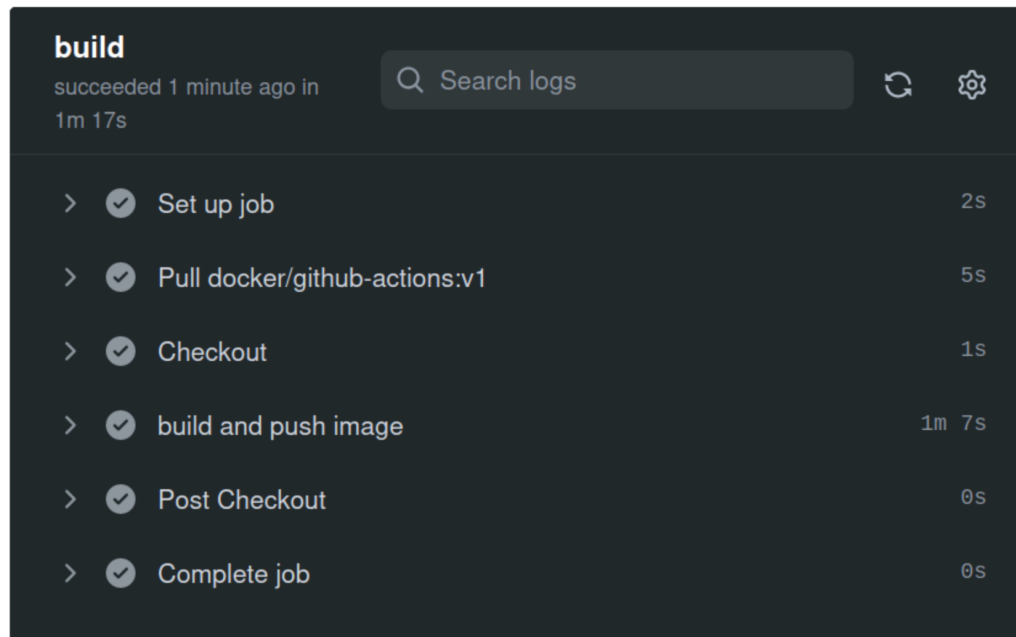
9s

<https://wangch64.github.io/github-pag...>



# Security Deployment for Docker Hub (1)

- Build and push Docker to Docker Hub
  - 自動部署至 Docker Hub



## Security

Code security and analysis

Deploy keys

**Secrets and variables**

Actions

Codespaces

Dependabot

## Repository secrets



DOCKER\_PASSWORD

Updated 12 minutes ago



DOCKER\_USERNAME

Updated 13 minutes ago

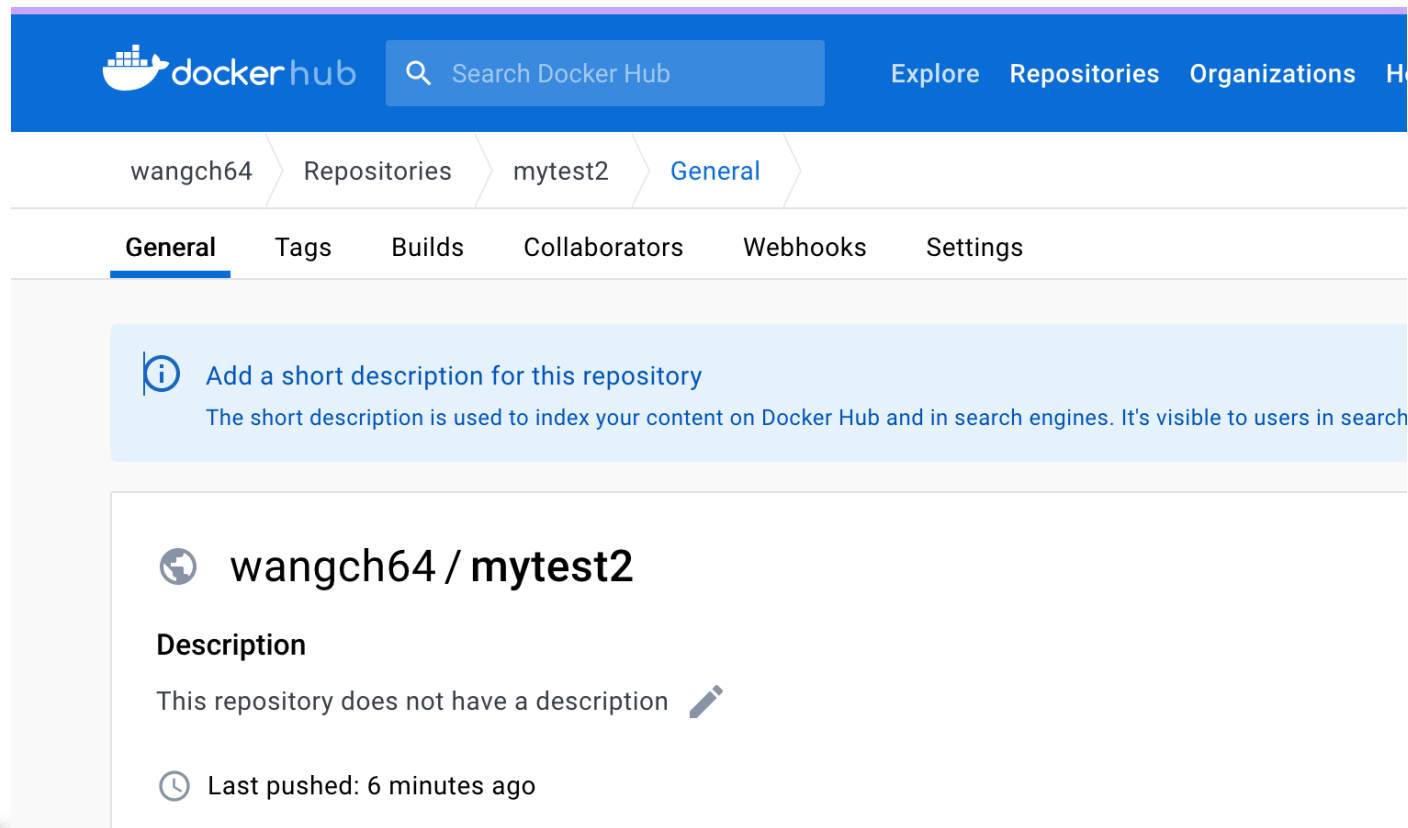






# Security Deployment for Docker Hub (2)

- 必須要在 Docker Hub 上面註冊，建立 repository



# Security Deployment for Docker Hub (3)

• **GitHub** name: CI

## Actions

on:

push:

branches:

- main

tags:

- '\*'

pull\_request:

jobs:

build:

name: build

runs-on: ubuntu-latest

timeout-minutes: 5

steps:

- name: Checkout

uses: actions/checkout@v2

# - name: build and test code

# run: make build

- name: build and push image

uses: docker/build-push-action@v1

with:

username: \${{ secrets.DOCKER\_USERNAME }}

password: \${{ secrets.DOCKER\_PASSWORD }}

repository: wangch64/mytest2

dockerfile: Dockerfile

always\_pull: true

tags: latest

<https://blog.wu-boy.com/2020/03/docker-release-github-actions-plugin/>

(參考資料)



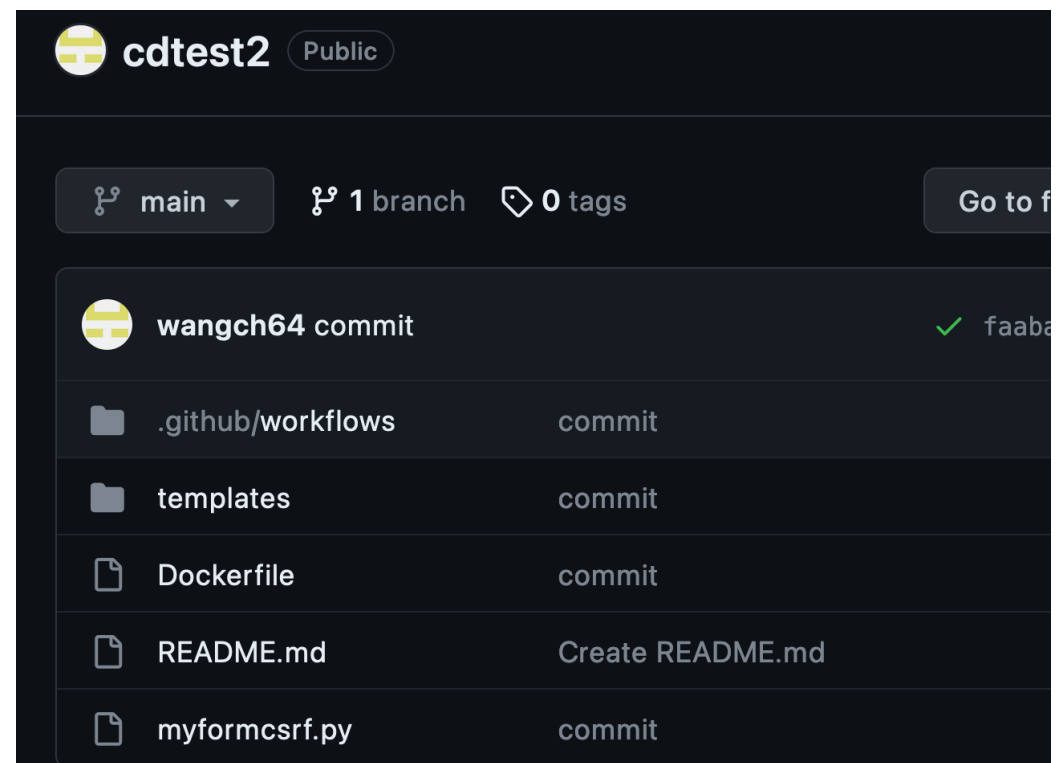


# Security Deployment for Docker Hub (4)

## • Dockerfile


FROM ubuntu  
USER root

RUN apt update  
RUN apt install python3 python3-pip -y  
RUN apt-get install -y python3-pip  
RUN pip install flask  
RUN pip install flask-wtf  
EXPOSE 5000  
COPY myformcsrf.py .  
COPY templates ./templates  
ENTRYPOINT [ "python3" ]  
CMD [ "myformcsrf.py" ]





# Security Deployment for Docker Hub (5)

- 當完成 Build and Push 之後，將 Docker 部署至 Docker Hub 儲存。

 wangch64 / mytest2



## Description

This repository does not have a description 

 Last pushed: 6 minutes ago

## Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	---	7 minutes ago



# Security Deployment for Docker Hub (6)

- 於本機執行 Docker 服務的情形
  - `docker run -p 5000:5000 --name myformapp wangch64/mytest2`

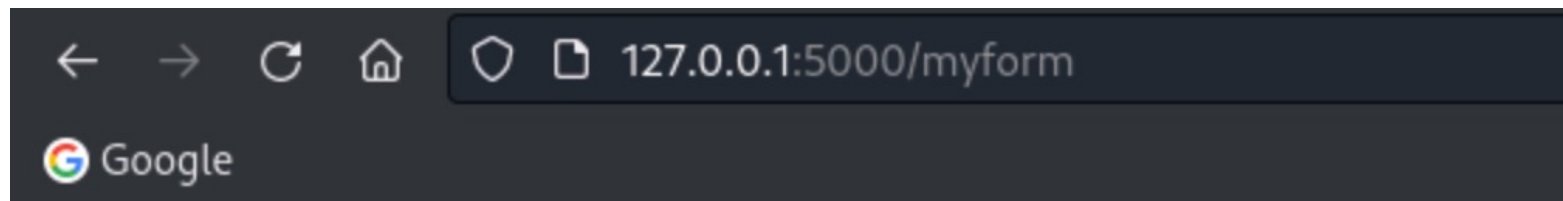
```
(one@kali)-[~]  
$ docker run -p 5000:5000 --name myformapp wangch64/mytest2  
Unable to find image 'wangch64/mytest2:latest' locally  
latest: Pulling from wangch64/mytest2  
aece8493d397: Already exists  
91a2aece80cd: Pull complete  
60e5c003624c: Pull complete  
4f4fb700ef54: Pull complete  
b1f00833a74d: Pull complete  
aadd0ac538a1: Pull complete  
4f8185a583fe: Pull complete  
bffdd8ee8566: Pull complete  
Digest: sha256:ac44b42d97300b19c75bfe2b44f3dba0937f513fedd702c735e48c555faca4c6  
Status: Downloaded newer image for wangch64/mytest2:latest  
* Serving Flask app 'myformcsrf'  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a  
production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:5000  
* Running on http://172.17.0.2:5000  
Press CTRL+C to quit
```





# Security Deployment for Docker Hub (7)

- Service 啟動情形



**Welcome to my TestSite**




# Other CI/CD Deployment Platforms (1)

- Cloud Run - Google Cloud Platform (GCP)
  - <https://cloud.google.com/run?hl=zh-TW>

## Cloud Run

在全代管平台上，以任何語言 (Go、Python、Java、Node.js、.NET 及 Ruby) 建構可擴充的容器化應用程式，並加以部署。

新客戶可以獲得價值 \$300 美元的免費抵免額，盡情體驗 Cloud Run。所有客戶每個月都能免費傳送 200 萬次要求，這些要求不會耗用抵免額。

免費試用 Cloud Run

聯絡銷售人員

- ✓ 查看這個[快速入門導覽課程](#)，瞭解如何部署用於回應傳入網路要求的範例容器。
- ✓ 要使用原始碼從頭開始建構嗎？請參閱這份[指南](#)，瞭解如何使用原始碼將範例應用程式部署至 Cloud Run。
- ✓ 透過 [Cloud Run 工作](#)執行資料庫遷移、每晚報表製作或批次資料轉換等作業







# Other CI/CD Deployment Platforms (2)

- Docker Swarm

- Swarm mode is an advanced feature for managing a cluster of Docker daemons. (管理 Docker 叢集的工具，Docker公司推出之原生容器調度管理平台)
- Use Swarm mode if you intend to use Swarm as a production runtime environment.

[Manuals](#) / [Docker Engine](#) / [Swarm mode](#) / Swarm mode overview

## Swarm mode overview

### Note

Swarm mode is an advanced feature for managing a cluster of Docker daemons.

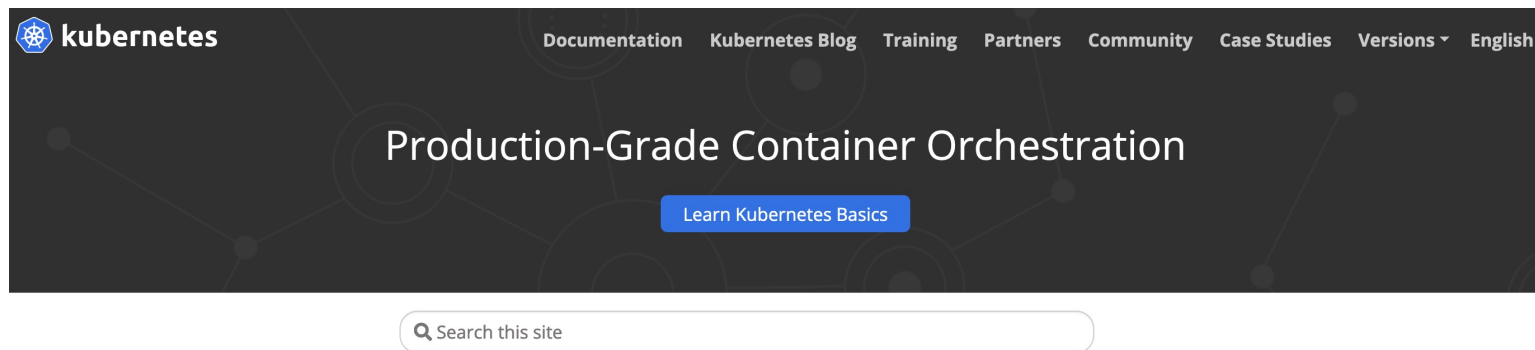
Use Swarm mode if you intend to use Swarm as a production runtime environment.

If you're not planning on deploying with Swarm, use [Docker Compose](#) instead. If you're developing for a Kubernetes deployment, consider using the [integrated Kubernetes feature](#) in Docker Desktop.



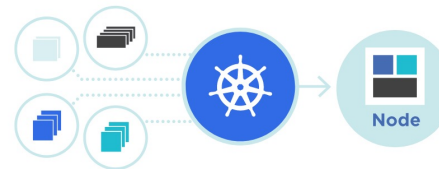
# Other CI/CD Deployment Platforms (3)

- Kubernetes (<https://kubernetes.io/>)
  - Kubernetes, 也稱為 K8s，是一個用於自動部署、擴充套件和管理容器化應用程式的開源系統。



Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.





## 參考文獻

- B. Chess and J. West, Secure Programming with Static Analysis, Addison-Wesley, 2007.
- R. C. Seacord, Secure Coding in C and C++, Addison-Wesley, Second Edition 2013.
- M. Paul, Official (ISC)<sup>2</sup> Guide to the CSSLP CBK, 2nd Edition, Auerbach Publications, 2013.
- 安全程式設計教材 - 109年教育部補助大學校院辦理新型態資安實務示範課程發展計畫

