Guide utilisateur pour déployer une application fast api dockeriser avec heroku et Guithub actions

Préambule : Créer un dossier ou on va mettre tous les éléments pour que l’application fonctionne

# Dockeriser FastApi en local

Lien tuto : <https://www.youtube.com/watch?v=CzAyaSolZjY&t=242s>

## Créer l’application en local sans dockeriser

Créer un environnement virtuel, Activation et Installation fastapi et uvicorn

C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>**python -m venv fa.venv**

C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>**fa.venv\Scripts\activate.bat**

(fa.venv) C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>**pip install fastapi uvicorn**

Créer un fichier main.py

from fastapi import FastAPI

app = FastAPI()

@app.get("/")

def index():

    return{"details":"hello, world!"}

Sélectionner le bon interpreter : Cliquer sur le bouton en bas à gauche

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Lancer le serveur

Executer dans le terminal : **uvicorn main:app --reload --host 0.0.0.0 --port 8000**

(fa.venv) C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>uvicorn **main:app --reload --host 0.0.0.0 --port 8000**

INFO: Started server process [44876]

INFO: Waiting for application startup.

INFO: Application startup complete.

INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

INFO: 127.0.0.1:54097 - "GET / HTTP/1.1" 200 OK

INFO: 127.0.0.1:54097 - "GET /favicon.ico HTTP/1.1" 404 Not Found

En tappant dans votre naviguateur **127.0.0.1:8000**

En local l’application devrait marcher et afficher *{"details":"hello, world!"}*

Pour avoi acceés à la documentation (Swagger UI) ajouter **/docs**. Tapper dans le naviguateur **127.0.0.1:8000/docs**

Arrêter le serveur : **ctrl+C**

Creer un requirements.txt : **pip freeze > requirements.txt**

(fa.venv) C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>**pip freeze > requirements.txt**

## Dockeriser l’application

Créer un **Dockerfile**

FROM python:3.12.2-slim

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt

Créer **.dockerignore**

env/

\_\_pycache\_\_/

\*.env

\*.env.\*

Créer **docker-compose.yaml**

version: '3'

services:

  web:

    build: .

    command: sh -c "uvicorn main:app --reload --host 0.0.0.0 --port 8000"

    ports:

      - 8000:8000

Built le contenaire : **docker compose up --build**

(fa.venv) C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>**docker compose up –build**

**NB :** Lorsque je fais des modifications sur le main.py il sont pas disponible dans le naviguateur

Pour pallier ca il faut ajouter ces 2 lignes dans le docker-compose.yaml

    volumes:

      - .:/app

Étapes à faire pour que ca marche :

1. Arrêter le container : **CTRL+C**
2. Modification du docker-compose.yamle :

version: '3'

services:

  web:

    build: .

    command: sh -c "uvicorn main:app --reload --host 0.0.0.0 --port 8000"

    ports:

      - 8000:8000

### Ajouter volume pour voir les modifications apportés dans main.py

    volumes:

      - .:/app

1. Executre dans le terminal : **docker compose up**
2. (fa.venv) C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>**docker compose up**

**Probleme : j’’arrive pas a a ecouter dans mon port 8000 dans ma machine. Du coup j’ai changé avec le port 8001**

**Voile le nouveau compose docker-yml**

version: '3'

services:

  web:

    build: .

    command: sh -c "uvicorn main:app --reload --host 0.0.0.0 --port 8000"

    ports:

      - 8001:8000

    volumes:

      - .:/app

# Créer un dépôt guithub, clona du dépôt et faire des push depuis le local vers guithub en utilisant clé ssh

**Lien utiles :**

* <https://www.youtube.com/watch?v=sjcEruaUOCo>
* <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
* <https://docs.oracle.com/en/learn/generate_ssh_keys/index.html#option-4-prior-windows-versions>

## Étapes pour générer clés ssh, dans le terminal faire :

1. ***ssh-keygen -t rsa -b 4096 -C*** [***your\_email@example.com***](mailto:your_email@example.com)
2. *Enter file in which to save the key (C:\Users\ymarega/.ssh/id\_rsa):* ***C:\Users\ymarega\.ssh\id\_rsa\_FA\_D\_H\_GA***
3. *C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>****dir C:\Users\ymarega\.ssh\***
4. *C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>****type C:\Users\ymarega\.ssh\id\_rsa\_FA\_D\_H\_GA.pub***
5. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"
6. Generating public/private rsa key pair.
7. Enter file in which to save the key (C:\Users\ymarega/.ssh/id\_rsa):
8. Enter passphrase (empty for no passphrase):
9. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"
10. Generating public/private rsa key pair.
11. Enter file in which to save the key (C:\Users\ymarega/.ssh/id\_rsa):
12. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>ssh-keygen -t rsa -b 4096 -C "FA\_D\_H\_GA\_SHH@example.com"
13. Generating public/private rsa key pair.
14. Enter file in which to save the key (C:\Users\ymarega/.ssh/id\_rsa): C:\Users\ymarega\.ssh\id\_rsa\_FA\_D\_H\_GA
15. Enter passphrase (empty for no passphrase):
16. Enter same passphrase again:
17. Your identification has been saved in C:\Users\ymarega\.ssh\id\_rsa\_FA\_D\_H\_GA
18. Your public key has been saved in C:\Users\ymarega\.ssh\id\_rsa\_FA\_D\_H\_GA.pub
19. The key fingerprint is:
20. SHA256:6e/FlrHG32ypWuACbLGEUmX+PqW3K0+k7snCzh6oA44 FA\_D\_H\_GA\_SHH@example.com
21. The key's randomart image is:
22. +---[RSA 4096]----+
23. |     ..o         |
24. |    . +          |
25. |   . . +         |
26. |    . o +.       |
27. |       =S. +.    |
28. | .    o.o \*o.+   |
29. |o .  ....\* +O.  .|
30. |E. .. .o+o\*+o. +.|
31. |   .. o+o\*==o.o.o|
32. +----[SHA256]-----+
33. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>ls C:\Users\ymarega\.ssh\
34. 'ls' n’est pas reconnu en tant que commande interne
35. ou externe, un programme exécutable ou un fichier de commandes.
36. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>dir C:\Users\ymarega\.ssh\
37. Le volume dans le lecteur C s’appelle Windows
38. Le numéro de série du volume est CE36-3077
39. Répertoire de C:\Users\ymarega\.ssh
40. 2025-02-28  12:58    <DIR>          .
41. 2025-02-28  12:58    <DIR>          ..
42. 2025-02-26  10:58               124 config
43. 2024-10-10  00:51               411 id\_ed25519
44. 2024-10-10  00:51               104 id\_ed25519.pub
45. 2025-02-26  10:46               419 id\_ed25519\_personnel
46. 2025-02-26  10:46               110 id\_ed25519\_personnel.pub
47. 2025-02-28  12:58             3 389 id\_rsa\_FA\_D\_H\_GA
48. 2025-02-28  12:58               752 id\_rsa\_FA\_D\_H\_GA.pub
49. 2025-02-26  10:58             2 328 known\_hosts
50. 2025-02-26  10:58             1 602 known\_hosts.old
51. 9 fichier(s)            9 239 octets
52. 2 Rép(s)  44 963 508 224 octets libres
53. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>cat id\_rsa\_FA\_D\_H\_GA.pub
54. 'cat' n’est pas reconnu en tant que commande interne
55. ou externe, un programme exécutable ou un fichier de commandes.
56. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>type id\_rsa\_FA\_D\_H\_GA.pub
57. Le fichier spécifié est introuvable.
58. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>type C:\Users\ymarega\.ssh\id\_rsa\_FA\_D\_H\_GA.pub
59. ssh-rsa  FA\_D\_H\_GA\_SHH@example.com
60. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>

## Connecter la cle SSH générer avec Guithub

Aller dans guithub.com > settings > SSH and GPG Keys > :

Cliquer sur new SSH key et le remplir avec la clé public généré précédemment

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

**ssh -T** [**git@github.com**](mailto:git@github.com)

**git config --global user.name**

**git config --global user.email**

**git init**

**git remote add origin** [**git@github.com:youssoupha-marega/MonProjet.git**](mailto:git@github.com:youssoupha-marega/MonProjet.git)

**git remote -v**

**Probléme avec la methode precedente : Quand je mets en bas quelques choses ca marche pas**

Enter file in which to save the key (/c/Users/ymarega/.ssh/id\_ed25519):

Du coup je mets rien j’appuie enter jusqu’à la fin

ymarega@HPPB650G8-006 MINGW64 ~/Desktop/Apprentissage/Fastapi-Docker-Heroku-GithubActions

$ ssh-keygen -t ed25519 -C "youssouphamarega@gmail.com"

Generating public/private ed25519 key pair.

Enter file in which to save the key (/c/Users/ymarega/.ssh/id\_ed25519):

Enter passphrase for "/c/Users/ymarega/.ssh/id\_ed25519" (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/ymarega/.ssh/id\_ed25519

Your public key has been saved in /c/Users/ymarega/.ssh/id\_ed25519.pub

The key fingerprint is:

SHA256:+j4XeGbYwM3QKho5ml2+x1z16bfsusu/nrpHhYEaF1E youssouphamarega@gmail.com

The key's randomart image is:

+--[ED25519 256]--+

|         .  o=E  |

|        . o o .  |

|     . . = +   o |

|    + o + + . . .|

|   + \* .S= . . ..|

|  o o ..o B   o. |

|      .+ = . ..  |

|      ..= . . o.o|

|       ooo   B@Xo|

+----[SHA256]-----+

ymarega@HPPB650G8-006 MINGW64 ~/Desktop/Apprentissage/Fastapi-Docker-Heroku-GithubActions

$ ls C:/Users/ymarega/.ssh

config  id\_ed25519  id\_ed25519.pub  known\_hosts  known\_hosts.old

ymarega@HPPB650G8-006 MINGW64 ~/Desktop/Apprentissage/Fastapi-Docker-Heroku-GithubActions

$ cat C:/Users/ymarega/.ssh/id\_ed25519

-----BEGIN OPENSSH PRIVATE KEY-----

b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW

QyNTUxOQAAACDxYTsEawBuUfo8BZ1htZ1MkSXYKJrOyeTowg+DpVRHJwAAAKA4iuw4OIrs

OAAAAAtzc2gtZWQyNTUxOQAAACDxYTsEawBuUfo8BZ1htZ1MkSXYKJrOyeTowg+DpVRHJw

AAAEDkUNK0BSi8X8Qd+VTRmVpucAp7fup6N+g02e9t4+m1fvFhOwRrAG5R+jwFnWG1nUyR

Jdgoms7J5OjCD4OlVEcnAAAAGnlvdXNzb3VwaGFtYXJlZ2FAZ21haWwuY29tAQID

-----END OPENSSH PRIVATE KEY-----

ymarega@HPPB650G8-006 MINGW64 ~/Desktop/Apprentissage/Fastapi-Docker-Heroku-GithubActions

$ cat C:/Users/ymarega/.ssh/id\_ed25519.pub

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPFhOwRrAG5R+jwFnWG1nUyRJdgoms7J5OjCD4OlVEcn youssouphamarega@gmail.com

ymarega@HPPB650G8-006 MINGW64 ~/Desktop/Apprentissage/Fastapi-Docker-Heroku-GithubActions

$ ssh -T git@github.com

Hi youssoupha-marega! You've successfully authenticated, but GitHub does not provide shell access.

ymarega@HPPB650G8-006 MINGW64 ~/Desktop/Apprentissage/Fastapi-Docker-Heroku-GithubActions

$

# Créer une application avec heroku et le mettre sous forme de container

1. Se connecter à Heroku : **heroku login**
2. Créer une nouvelle application sur Heroku (Si c’est pas encore fait) : **heroku create nom\_application**
3. Activer le mode Docker sur Heroku : **heroku stack:set container -a nom\_application**
4. Récupérer la clé API Heroku : **heroku auth:token**

To generate a token that expires in one year, use **heroku authorizations:create**

**Attention il est preferable d’aller sur l’interface heroku en ligne pour récupérer cle d’api**

**Va dans Accountount Setting et copie** API Key

1. Ajouter la clé API et le nom de l’application dans GitHub

* **Va sur** [GitHub → Ton repo → Settings → Secrets and variables → Actions](https://github.com/youssoupha-marega/ml-deploy-flask-github-actions/settings/secrets/actions).
* **Clique sur "New repository secret"**.
* **Ajoute ces secrets :**
  + - HEROKU\_API\_KEY → **Colle la clé API récupérée avec heroku auth:token**.
    - HEROKU\_APP\_NAME → **Le nom de ton application (ex:** nom\_application**)**.
    - HEROKU\_EMAIL → **Ton email lié à Heroku** (optionnel mais recommandé).

1. Déploiement avec GitHub et automatique

* **Va sur** [Heroku → Dashboard → nom\_application → Deploy](https://github.com/youssoupha-marega/ml-deploy-flask-github-actions/settings/secrets/actions)
* Choisis Deployment method  Github en specifiant le compte, le repo et la branche principalement c’est main
* Activer les déploiements automatiques (**Enable Automatic Deploys)**

# Déployer l’Application avec Guithub Actions via un push