

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

1 Dockeriser FastApi en local

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

1.1 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
```

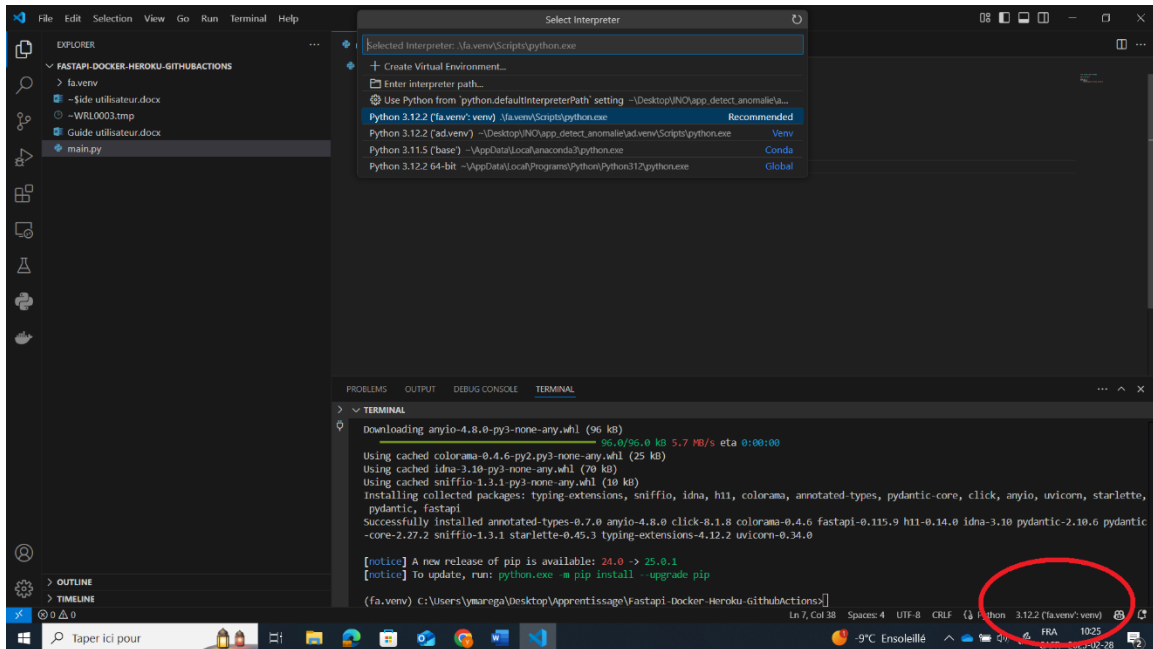
Créer un fichier main.py

```
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



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 tapant dans votre navigateur **127.0.0.1:8000**

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

Pour avoir accès à la documentation (Swagger UI) ajouter **/docs**. Tapper dans le navigateur **127.0.0.1:8000/docs**

Arrêter le serveur : **ctrl+C**

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

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

1.2 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.yml**

```
version: '3'

services:
  web:
    build: .
    command: sh -c "uvicorn main:app --reload --host 0.0.0.0 --port 8000"
    ports:
      - 8000:8000
```

Built le conteneur : **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 navigateur
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.yaml :

```
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
```

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

Probleme : j'arrive pas a écouter 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
```

2 Créer un dépôt github, clona du dépôt et faire des push depuis le local vers github 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

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

1. `ssh-keygen -t rsa -b 4096 -C 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.
10. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>`ssh-keygen -t rsa -b 4096 -C "your_email@example.com"`
11. Generating public/private rsa key pair.
12. Enter file in which to save the key (C:\Users\ymarega\.ssh\id_rsa):
13. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-GithubActions>`ssh-keygen -t rsa -b 4096 -C "FA_D_H_GA_SHH@example.com"`
14. Generating public/private rsa key pair.
15. Enter file in which to save the key (C:\Users\ymarega\.ssh\id_rsa):
`C:\Users\ymarega\.ssh\id_rsa_FA_D_H_GA`
16. Enter passphrase (empty for no passphrase):
17. Enter same passphrase again:
18. Your identification has been saved in
`C:\Users\ymarega\.ssh\id_rsa_FA_D_H_GA`

```

19. Your public key has been saved in
   C:\Users\ymarega\.ssh\id_rsa_FA_D_H_GA.pub
20. The key fingerprint is:
21. SHA256:6e/FlrHG32ypWuACbLGEUmX+PqW3K0+k7snCzh6oA44
   FA_D_H_GA_SHH@example.com
22. The key's randomart image is:
23. +---[RSA 4096]-----+
24. |      ..o      |
25. |      . +      |
26. |      . . +      |
27. |      . o +.      |
28. |      =S. +.      |
29. | .      o.o *o.+      |
30. |o .      ....* +o. .|
31. |E. .. .o+o*+o. +. |
32. |      .. o+o*==o.o.o|
33. +-----[SHA256]-----+
34.
35. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-
   GithubActions>ls C:\Users\ymarega\.ssh\
36. 'ls' n'est pas reconnu en tant que commande interne
37. ou externe, un programme exécutable ou un fichier de commandes.
38.
39. C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-
   GithubActions>dir C:\Users\ymarega\.ssh\
40. Le volume dans le lecteur C s'appelle Windows
41. Le numéro de série du volume est CE36-3077
42.
43. Répertoire de C:\Users\ymarega\.ssh
44.
45. 2025-02-28  12:58    <DIR>          .
46. 2025-02-28  12:58    <DIR>          ..
47. 2025-02-26  10:58                124 config
48. 2024-10-10  00:51                411 id_ed25519
49. 2024-10-10  00:51                104 id_ed25519.pub
50. 2025-02-26  10:46                419 id_ed25519_personnel
51. 2025-02-26  10:46                110 id_ed25519_personnel.pub
52. 2025-02-28  12:58                 3 389 id_rsa_FA_D_H_GA
53. 2025-02-28  12:58                752 id_rsa_FA_D_H_GA.pub
54. 2025-02-26  10:58                 2 328 known_hosts
55. 2025-02-26  10:58                 1 602 known_hosts.old
56.                  9 fichier(s)                9 239 octets
57.                  2 Rép(s)  44 963 508 224 octets libres
58.

```

```

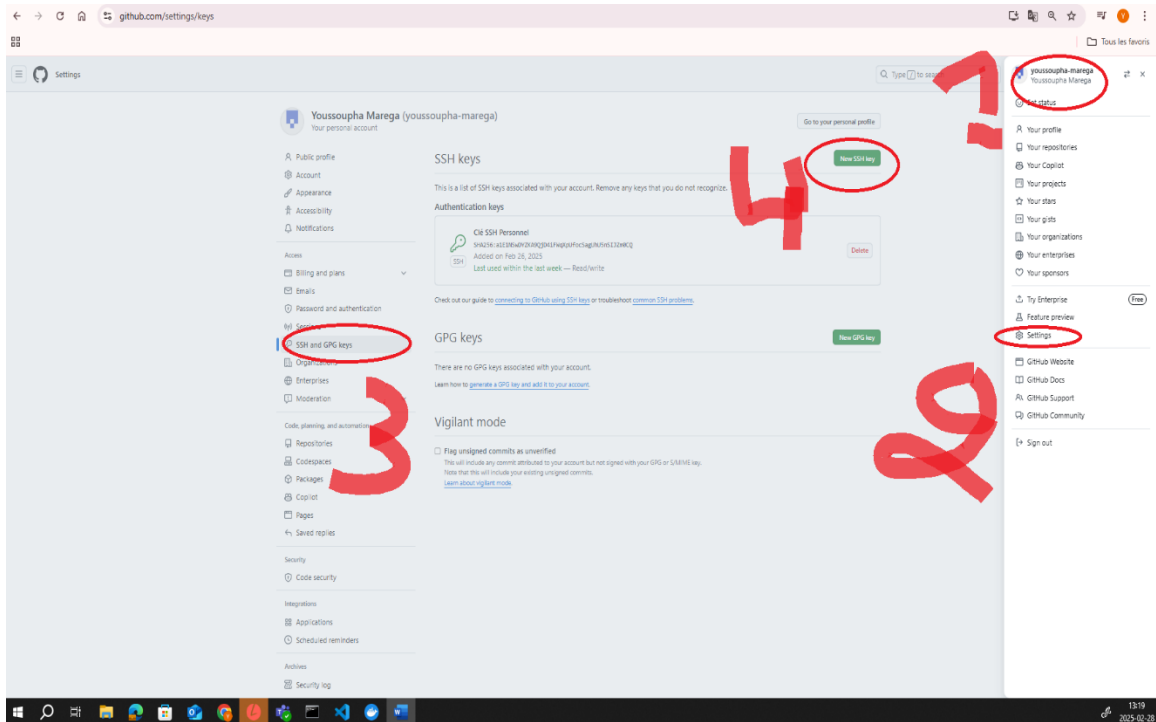
59.C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-
  GithubActions>cat id_rsa_FA_D_H_GA.pub
60.'cat' n'est pas reconnu en tant que commande interne
61.ou externe, un programme exécutable ou un fichier de commandes.
62.
63.C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-
  GithubActions>type id_rsa_FA_D_H_GA.pub
64.Le fichier spécifié est introuvable.
65.
66.C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-
  GithubActions>type C:\Users\ymarega\.ssh\id_rsa_FA_D_H_GA.pub
67.ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACQC6lGSKOJ02kbNndr0k0GD6qjs1tLOIrvMbuTDZ
UYAWYT/BP/sENkc+HbO93+h/ALEKPNJwWjn/0Zcmi5WR/5pWDliXdy4BQUUnlBx0H7nT
ow3GF4Hd3h0Za9Tb4e/JH52LRD/KNSuk5zS2w0y2atLCsiHNBbw1JqqrGFSOCQzTbS1r
/WVxePht+su1kPYtsZwGVH2RSbHmDIzyPD09MSnBfbP6cn6B3DIGkhBeTGNxWPL5Q5C6
TNZCe9zU0iZib2o0hc3qvbaTX8Tn8Itg4imhIX/eLRsBPP+pu2lyhmd08/eX3mHnpJo
5x+pNeJqYDjag2ugFnIBq4XI4xCsu034E3HLCdKQopcgviN8ymsY8iy2leY4y6VKWZOF
6mYENscv66IBesGwyiufC0Du8Vow3r1loDDv+0DNRpWzT7607rNxmd8In2EesE9xYa0i
AxzbMI9b1+UPVxTJtqvqElvKbKCYQmOVxMFryQPowzIGE4dAiES8iQ30y00Hks5LeQF/
1LJRt4Q4nJ9jOiCGnv0fNWV+hWupDuiZ/eTuuiZ+3+ZGkFXCzrGWq3CS5TsQo0BmBOF3
ag/sDuA1r5bjiL/UBvZyfjew+nNuH0wwqr4EeR4jc3WP6s1IuUHs1SMTYUkGDLTkaKLN
3U2i/kHuHpkF9g3EBQm3fUcN06dY3ikxjQ== FA_D_H_GA_SSH@example.com
68.
69.C:\Users\ymarega\Desktop\Apprentissage\Fastapi-Docker-Heroku-
  GithubActions>

```

2.2 Connecter la cle SSH générer avec Guithub

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

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



ssh -T git@github.com

git config --global user.name

git config --global user.email

git init

git remote add origin [git@github.com:yousoupha-marega/MonProjet.git](https://github.com/yousoupha-marega/MonProjet.git)

git remote -v

Problème 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 "yousouphamarega@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-----
```

```
b3B1bnNzaC1rZXktbjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAAtzc2gtZW  
QyNTUxOQAAACDxYTsEawBuUfo8BZ1htZ1MkSXYKJrOyeTowg+DpVRHJwAAKA4iuw40Irs  
OAAAAAtzc2gtZWQyNTUxOQAAACDxYTsEawBuUfo8BZ1htZ1MkSXYKJrOyeTowg+DpVRHJw  
AAAEKUNK0BSi8X8Qd+VTRmVpucAp7fup6N+g02e9t4+m1fvFh0wRrAG5R+jwFnWG1nUyR  
Jdgoms7J50jCD40lVEcnAAAAGnlvdXNzb3VwaGFtYXJlZ2FAZ21haWwY29tAQID
```

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

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

```
$ cat C:/Users/ymarega/.ssh/id_ed25519.pub
```

```
ssh-ed25519
```

```
AAAAC3NzaC1lZDI1NTE5AAAAIPFh0wRrAG5R+jwFnWG1nUyRJdgoms7J50jCD40lVEcn
```

```
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
$
```

3 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 Accountout Setting et copie API Key
- 5) Ajouter la clé API et le nom de l'application dans GitHub
 - Va sur GitHub → Ton repo → Settings → Secrets and variables → 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*).
- 6) Déploiement avec GitHub et automatique
 - Va sur Heroku → Dashboard → nom_application → Deploy
 - Choisis Deployment method Github en specifiant le compte, le repo et la branche principalement c'est main
 - Activer les déploiements automatiques (**Enable Automatic Deploys**)

4 Déployer l'Application avec Guithub Actions via un push

L'Application est fonctionnelle : ci-dessous un exemple pour faire des requêtes vers L'APIs

```
import requests

# ✅ URL de ton API FastAPI sur Heroku
BASE_URL = "https://fastapigithubaction-12b963722dae.herokuapp.com"

# ✅ Paramètres de la requête
params = {
    "nom": "Marega",
    "prenom": "Youssoupha",
    "date_naissance": "19-02-1997",
    "profession": "Data Scientist",
    "salaire": 50000
}

# ✅ Envoyer une requête GET
response = requests.get(f"{BASE_URL}/presente toi", params=params)

# ✅ Vérifier si la requête a réussi (200 OK)
if response.status_code == 200:
    print("Réponse de l'API :", response.json()) # 🐞 Affiche les données en JSON
else:
    print("Erreur :", response.status_code, response.text) # 🐞 Affiche l'erreur si la requête échoue
```