# Déployer une **application web Flask** sur **Microsoft Azure App Service**, en utilisant **Git et GitHub** comme système de versionnement et de déploiement

## 1. Contexte du TP

Dans ce TP, vous déployez une application web Flask (Python) sur le cloud Microsoft Azure en utilisant Azure App Service et GitHub.

L'objectif n'est pas d'apprendre à configurer des serveurs, mais de comprendre comment une application passe du code local à une application web accessible sur Internet, comme en entreprise.

## 2. Qu'est-ce qu'Azure App Service ?

**Azure App Service** est un service de type **PaaS (Platform as a Service)**.

Cela signifie que :

- Azure gère les serveurs, le système d'exploitation, le réseau et la sécurité de base
- Vous gérez uniquement le code de l'application

Vous ne créez **pas** de machine virtuelle et vous ne vous connectez **jamais** en SSH.

## 3. À quoi sert Azure App Service ?

Azure App Service permet de :

- héberger une application web (Flask, Django, Node.js, Java, .NET...)
- exposer automatiquement une **URL publique HTTPS**
- gérer le redémarrage, la disponibilité et la montée en charge
- connecter l'application à GitHub pour un déploiement automatique

## 4. Azure App Service vs Machine Virtuelle (VM)

Machine Virtuelle (IaaS) Azure App Service (PaaS)

| | |
|---|---|
| Gestion du serveur | Serveur géré par Azure |
| Installation manuelle de Python | Python prêt à l'emploi |
| Configuration du serveur web | Automatique |
| Maintenance élevée | Maintenance minimale |
| Plus de liberté | Plus simple et rapide |

## 5. Architecture du déploiement

Le flux de travail est le suivant :

1. Vous développez l'application en local
2. Vous versionnez le code avec Git
3. Vous poussez le code sur GitHub
4. Azure App Service récupère le code
5. Azure déploie automatiquement l'application

Ordinateur

↓ git push

GitHub

↓

Azure App Service

↓

Application web en ligne

## 6. Création d'une application Azure App Service

Lors de la création d'une application web sur Azure, vous devez choisir :

- un nom (sera utilisé dans l'URL)
- un runtime (Python 3.x)

- un système d'exploitation (Linux recommandé)
- une région (ex : France Central)
- un App Service Plan (souvent gratuit en TP)

Azure crée automatiquement l'environnement d'exécution.

## 7. Déploiement via GitHub

### *Pourquoi utiliser Git ?*

Git permet de :

- ✓ garder une **trace de toutes les modifications**
- ✓ travailler proprement (même à plusieurs)
- ✓ déployer automatiquement une application sur Azure

Azure App Service peut **récupérer directement le code depuis GitHub**
**chaque git push peut déclencher un déploiement**

### *Étape 1 — Installer Git*

Git est l'outil qui permet :

- ✓ de versionner le code
- ✓ de communiquer avec GitHub
- ✓ d'envoyer le code vers Azure

### Ce que vous faites

- ✓ Télécharger Git depuis le site officiel
- ✓ L'installer sur votre machine

### Pourquoi ajouter Git au PATH ?

Pour pouvoir utiliser la commande git **depuis n'importe quel terminal** (PowerShell, VS Code, etc.).

C:\Program Files\Git\bin

C:\Program Files\Git\cmd

Sans ça, la commande git n'est pas reconnue.

### *Étape 2 — Créer un dépôt GitHub*

GitHub sert de :

- ✓ dépôt central
- ✓ pont entre votre machine et Azure
- ✓ historique du projet

Le dépôt contient :

- ✓ app.py → code Flask
- ✓ requirements.txt → dépendances Python
- ✓ README.md → documentation

### *Étape 3 — Initialiser le dépôt Git en local*

git init

Initialise un dépôt Git dans le dossier du projet.

### *Étape 4 — Ajouter les fichiers au suivi Git*

git add app.py

Dit à Git :“Je veux suivre ce fichier et enregistrer ses modifications.”

### *Étape 5 — Créer un commit*

git commit -m "init"

Un **commit** est :une photo de l’état du projet avec un message explicatif

Objectif :

- ✓ apprendre à **documenter les changements**
- ✓ structurer le travail comme en entreprise

### *Étape 6 — Lier le projet local à GitHub*

git remote add origin https://github.com/username/hello-world.git

git push -u origin master

Cela permet :

- ✓ d'envoyer le code sur GitHub
- ✓ de synchroniser local ↔ distant

### *Étape 7 — Comprendre requirements.txt*

Flask==2.2.2

Werkzeug==2.2.2

gunicorn==20.1.0

#### Pourquoi ce fichier est essentiel ?

Azure **n'installe pas automatiquement** les bibliothèques Python.

requirements.txt dit à Azure :

"Voici exactement les bibliothèques nécessaires pour exécuter l'application."

### *Étape 8 — Installer les dépendances en local*

pip install -r requirements.txt

Permet de :

- ✓ tester l'application localement
- ✓ vérifier que tout fonctionne avant le déploiement

### *Étape 9 — Modifier le code (exemple)*

return "Hello, World! Hello Everyone!"

Toute modification du code :

- ✓ n'est pas automatiquement envoyée
- ✓ doit être commitée puis poussée

### *Étape 10 — Vérifier l'état du dépôt*

git status

Permet de voir :

- ✓ quels fichiers ont changé
- ✓ ce qui est prêt à être envoyé
- ✓ ce qui ne l'est pas encore

### *Étape 11 — Envoyer une modification*

```
git add app.py
```

```
git commit -m "changement texte"
```

```
git push
```

Effet :

- ✓ le code est envoyé sur GitHub
- ✓ Azure détecte le changement
- ✓ Azure redéploie automatiquement l'application

comprendre le lien entre **Git → GitHub → Azure**

### *Étape 12 — Gestion d'erreur classique*

```
git pull origin master --rebase
```

```
git push origin master
```

*Pourquoi cette erreur arrive ?*

- ✓ GitHub a une version plus récente
- ✓ le dépôt local n'est plus synchronisé

pull --rebase :

- ✓ récupère les changements distants
- ✓ rejoue vos commits proprement

## Création de l’App Service

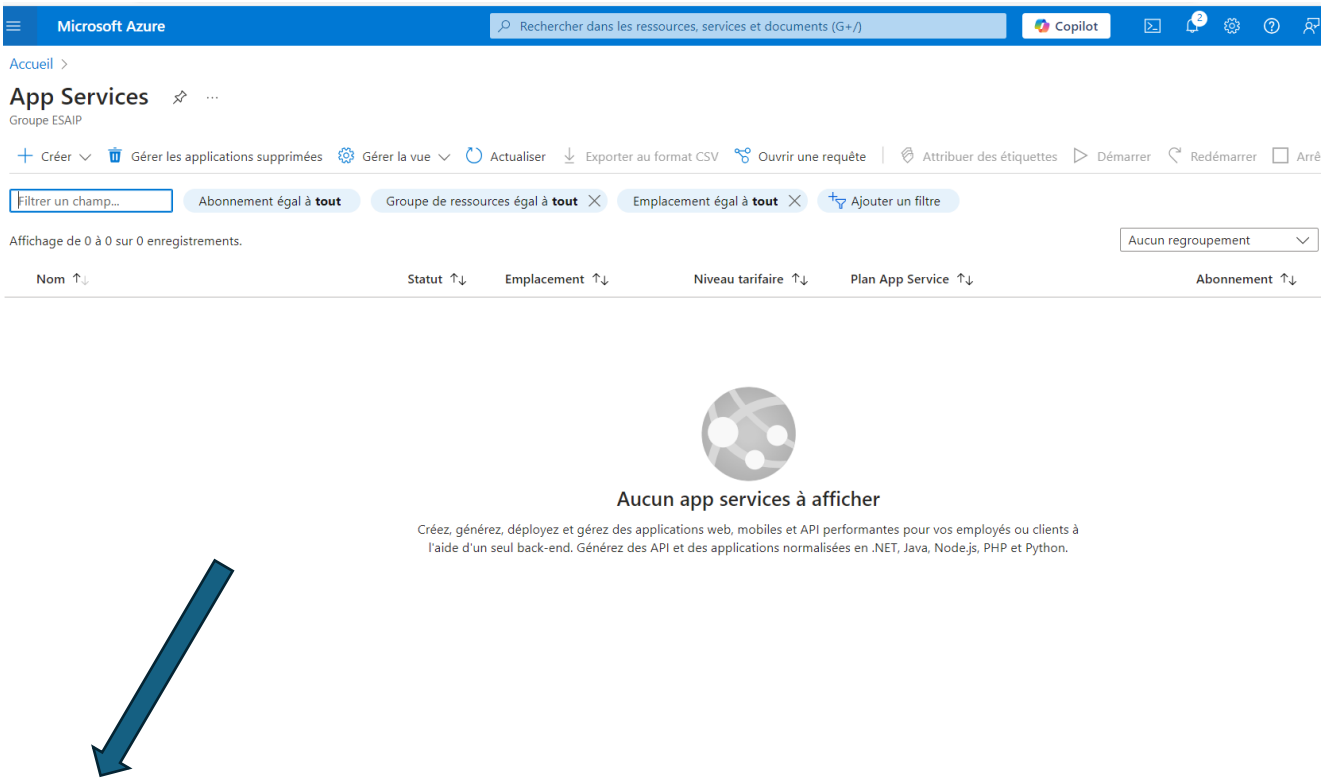
Quand tu fais :Azure → App Services → Créer → Application Web

Tu demandes à Azure :

- un environnement d’exécution Python
- une URL publique
- un App Service Plan (CPU, RAM)
- une région (ex : France Central)

Azure crée :

- un conteneur d’exécution Python
- un serveur web interne
- un point d’accès HTTPS



Accueil >

# App Services

Groupe ESAIP

+ Créer | Gérer les applications supprimées | Gérer la vue

- Application web
- Application web statique
- Application web + base de données
- Wordpress sur App Service

ent égal à **tout** | Groupe de

Statut

---

Microsoft Azure | Rechercher dans les ressources, services

Accueil > App Services >

## Créer une application web

Groupe de ressources * : (Nouveau) PaaS_Class

Créer nouveau

**Détails de l'instance**

Nom : FlaskApp1

.azurewebsites.net

Essayez un nom d'hôte unique par défaut (préversion). Découvrir plus d'informations sur cette mise à jour

Publier * : Code / Conteneur

Pile d'exécution * : Python 3.12

Système d'exploitation * : Linux / Windows

Région * : France Central

Vous ne trouvez pas votre plan App Service ? Essayez une autre région ou sélectionnez votre environnement App Service Environment.

**Plans de tarification**

Le niveau tarifaire du plan App Service détermine l'emplacement, les fonctionnalités, le coût et les ressources de calcul associés à votre application. En savoir plus

Vérifier + créer | < Précédent | Suivant : Base de données >

Microsoft Azure

Accueil > App Services >

# Créer une application web

De base | Base de données | Déploiement | Réseau | Superviser + sécuriser | Balises | Vérifier +

Récapitulatif

**Application web**
par Microsoft

**Référence SKU Gratuit**
Prix estimé - Gratuit

L'authentification de base pour cette application est actuellement désactivée et peut avoir un impact sur les déploiements. Cliquez pour en savoir plus.

**Détails**

| | |
|---|---|
| Abonnement | 89c6a329-1ed3-42c1-bb0d-e94637334fe8 |
| Groupe de ressources | PaaS_Class |
| Nom | FlaskApp1 |
| Publier | Code |
| Pile d'exécution | Python 3.12 |

**Plan App Service (nouveauté)**

| | |
|---|---|
| Nom | ASP-PaaSClass-aae3 |
| Système d'exploitation | Linux |
| Région | France Central |
| Référence | Gratuit |
| ACU | Infrastructure partagée |
| Mémoire | 1 Go de mémoire |

**Superviser + sécuriser**

| | |
|---|---|
| Application Insights | Non activé |

Créer | < Précédent | Suivant > | Télécharger un modèle pour automation

Microsoft Azure | Rechercher dans les ressources, services et documents (G+/) | Copilot

Accueil >

# Microsoft.Web-WebApp-Portal-e8779bba-b1b0 | Vue d'ensemble

Déploiement

Rechercher

- Vue d'ensemble
- Entrées
- Sorties
- Modèle

Supprimer | Annuler | Redéployer | Télécharger | Actualiser

## Le déploiement est en cours

Nom du déploiement : Microsoft.Web-WebApp-Portal-e8779bba-b...
Abonnement : Azure for Students
Groupe de ressources : PaaS_Class

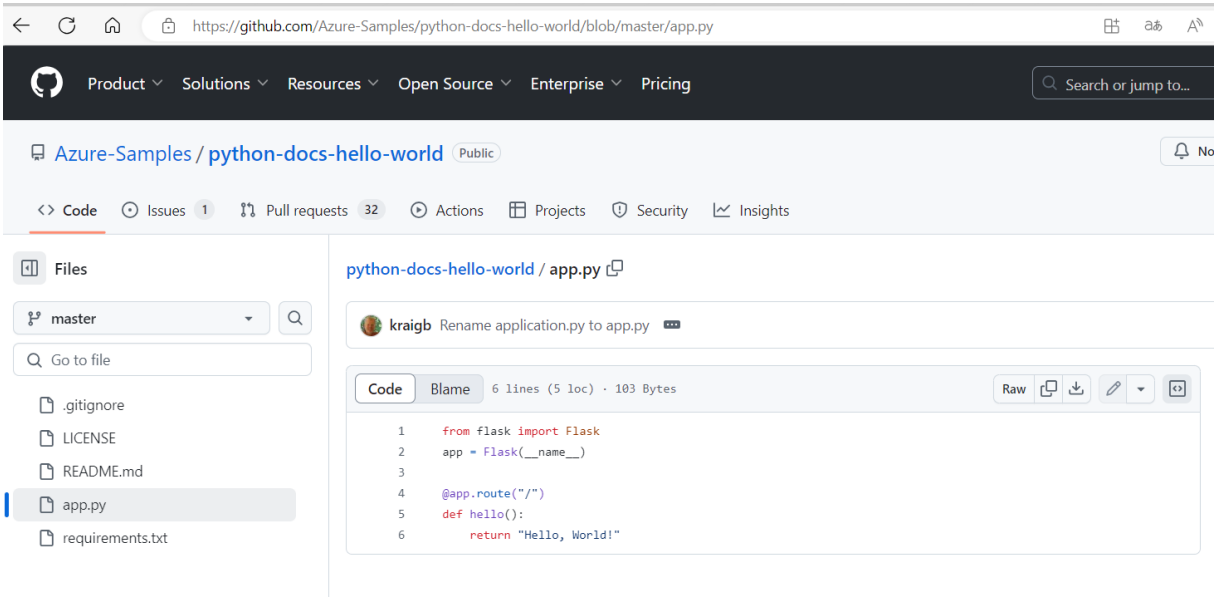
Heure de début : 01/01/2025 11:11:00
ID de corrélation : 2fb47cd2-07f0-4751-8637-ab4dc027f368

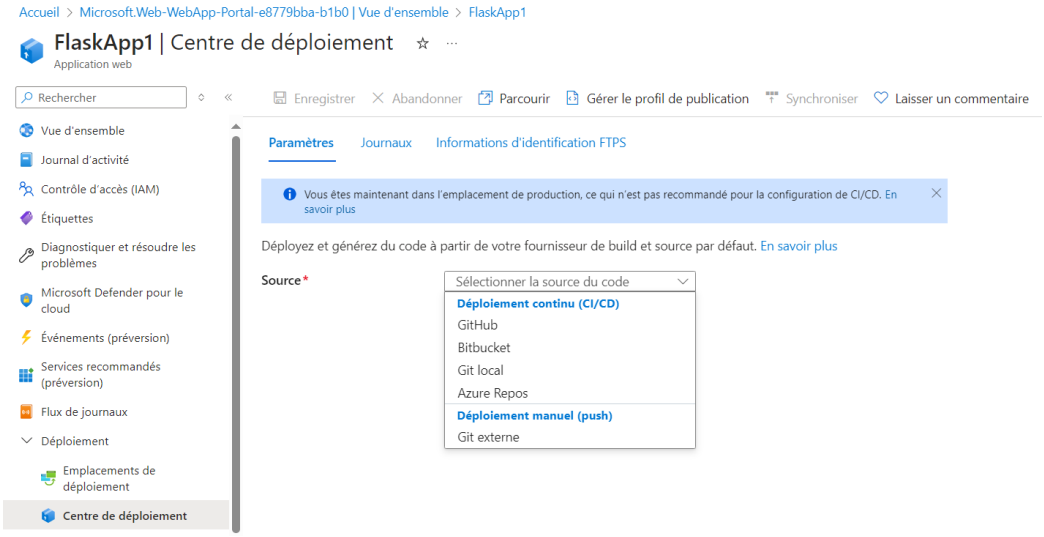
**Détails du déploiement**

| Ressource | Type | Statut | Détails de l'opération |
|---|---|---|---|
| ASP-PaaSClass-aae3 | Microsoft.Web/serverfarms | OK | Détails de l'opération |

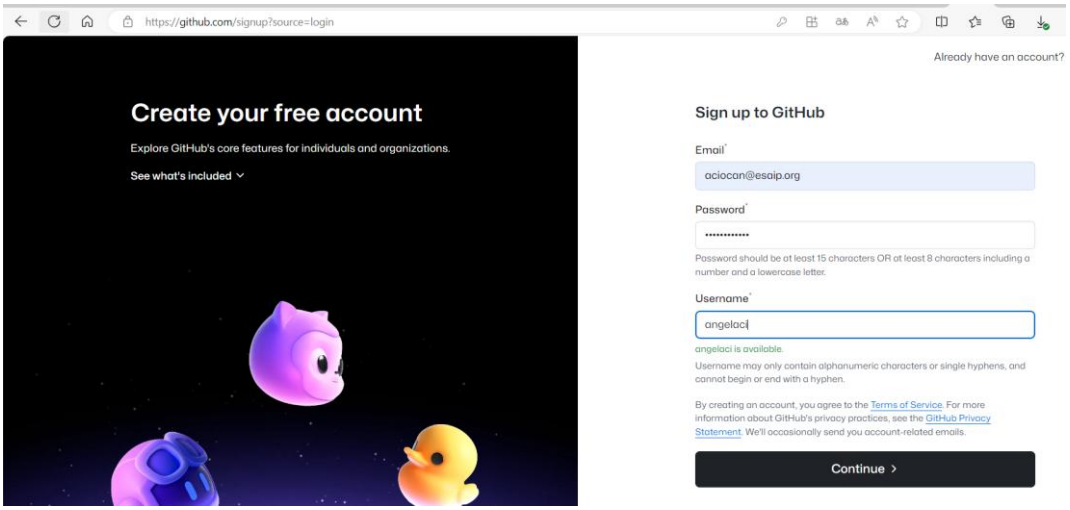
Envoyer des commentaires

Partagez votre expérience avec le déploiement




Create compte GitHub

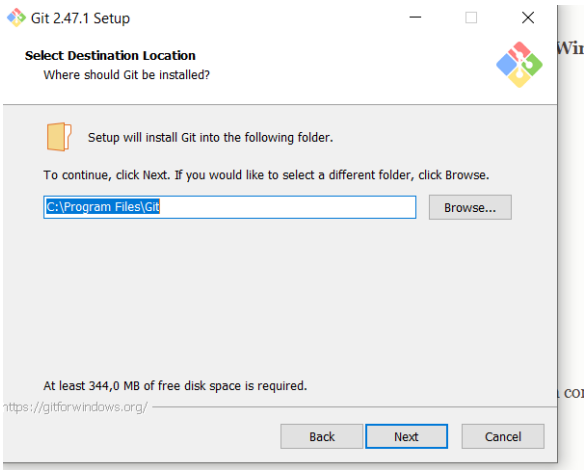


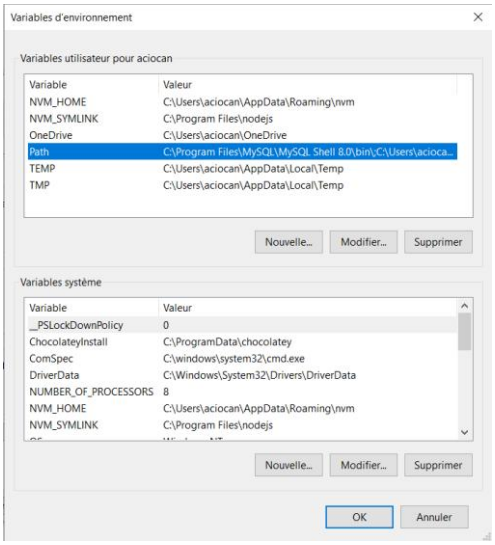

## ...or create a new repository on the command line

```
echo "# CloudComputing" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/angelaci/CloudComputing.git
git push -u origin main
```

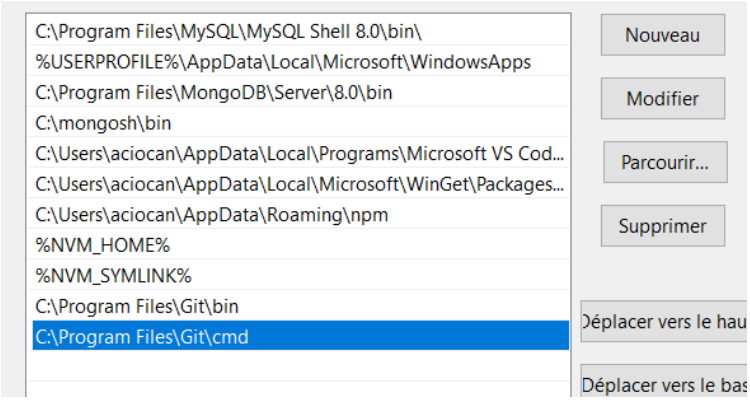
Install Git

[Git](https://git-scm.com/downloads/win)

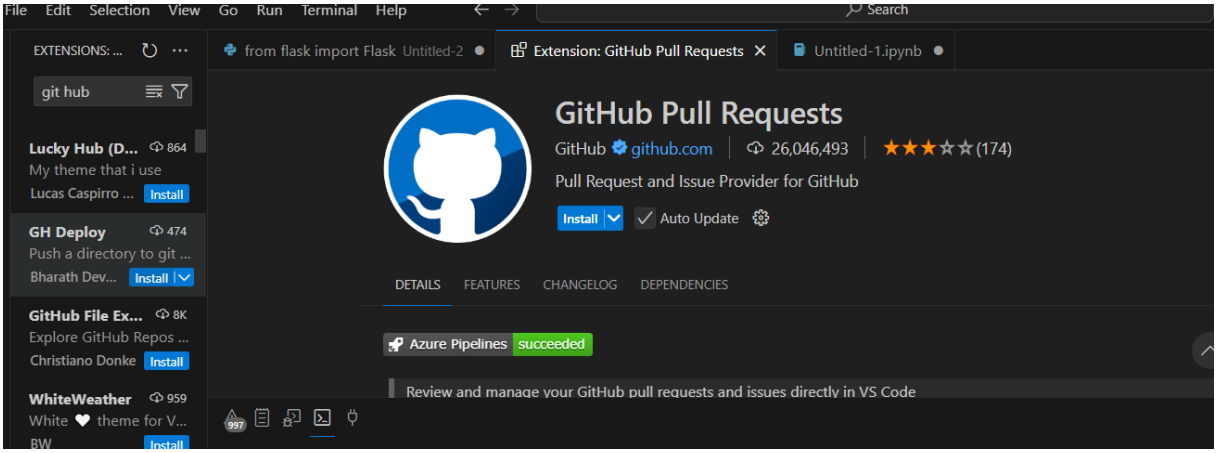


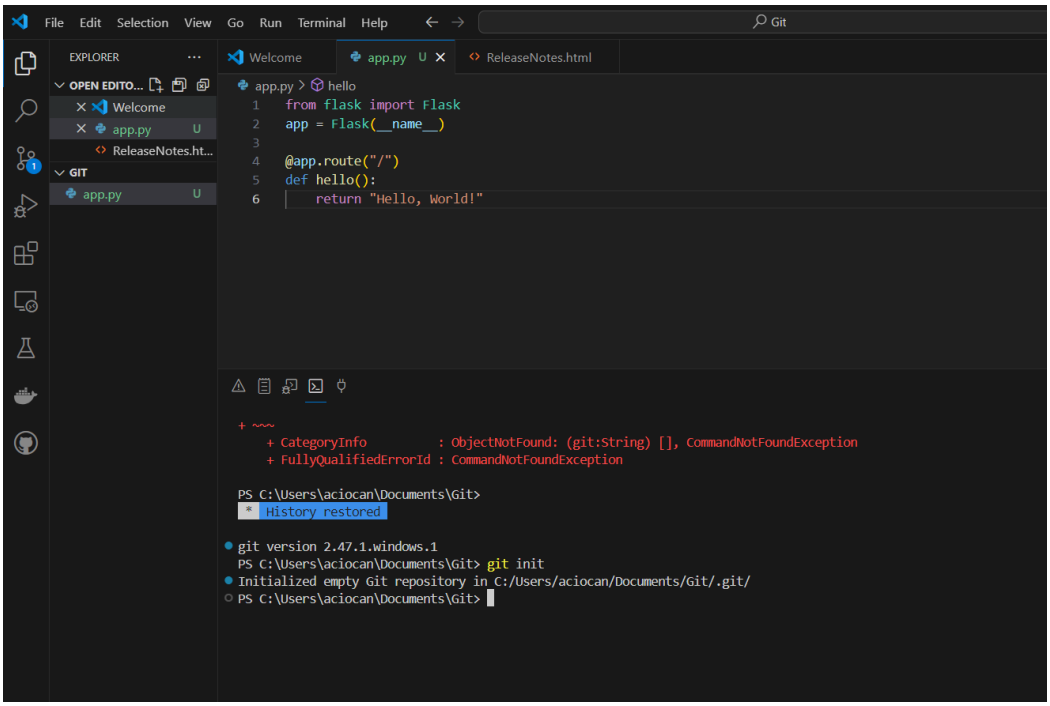



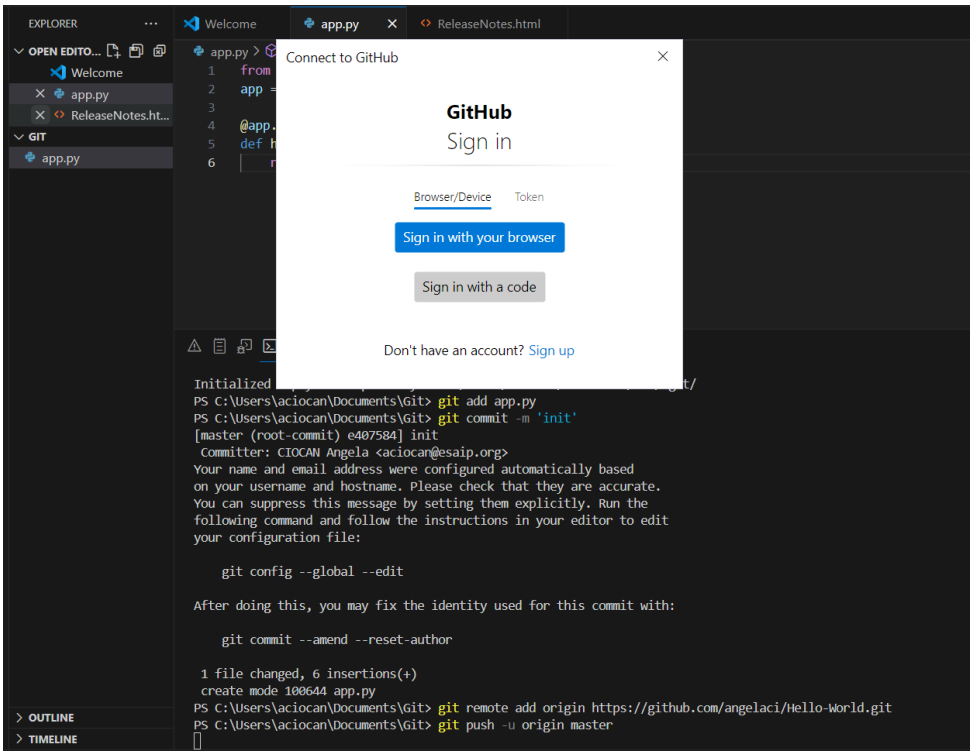

Add the path git/bin and git/cmd in path (environment variables)

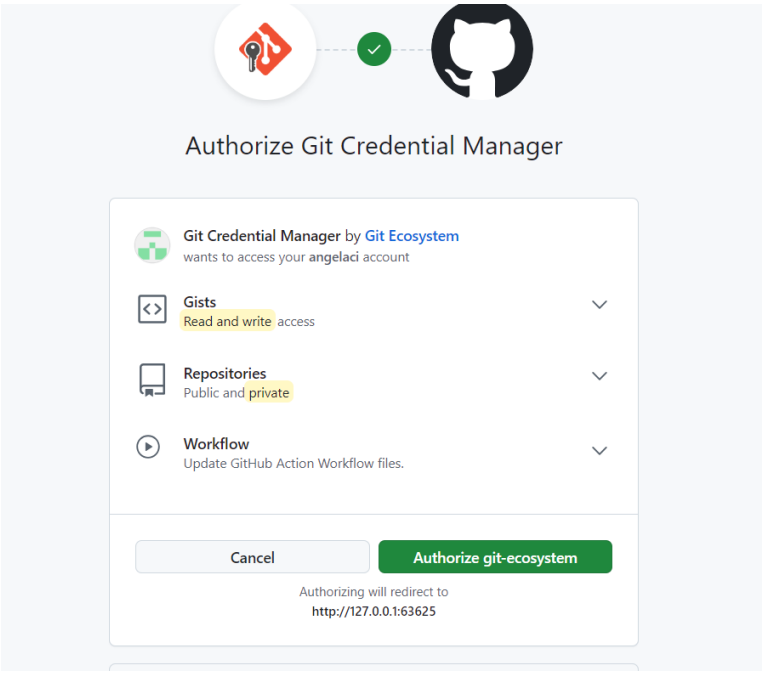


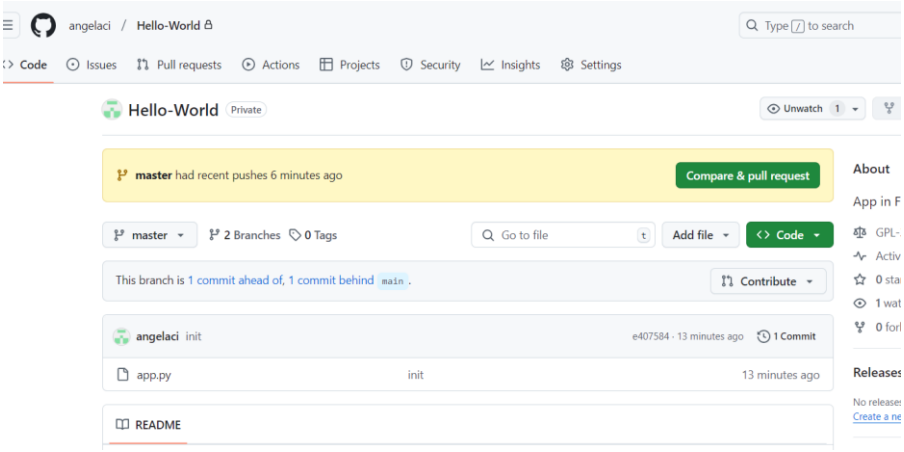
After installation, close and reopen PowerShell.



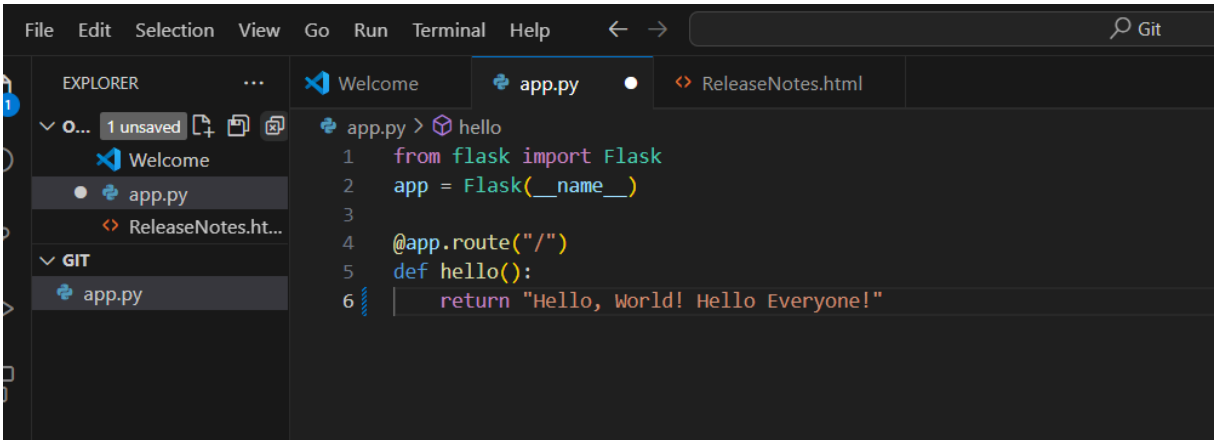




EXPLORER
OPEN EDITO...
Welcome
app.py
ReleaseNotes.ht...
GIT
app.py
OUTLINE
TIMELINE
Welcome
app.py
ReleaseNotes.html
app.py
Connect to GitHub
GitHub
Sign in
Browser/Device
Token
Sign in with your browser
Sign in with a code
Don't have an account? Sign up
Initialized
PS C:\Users\aciocan\Documents\Git> git add app.py
PS C:\Users\aciocan\Documents\Git> git commit -m 'init'
[master (root-commit) e407584] init
Committer: CIOCAN Angela <aciocan@esaip.org>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
1 file changed, 6 insertions(+)
create mode 100644 app.py
PS C:\Users\aciocan\Documents\Git> git remote add origin https://github.com/angelaci/Hello-World.git
PS C:\Users\aciocan\Documents\Git> git push -u origin master


Authorize Git Credential Manager
Git Credential Manager by Git Ecosystem
wants to access your angelaci account
Gists
Read and write access
Repositories
Public and private
Workflow
Update GitHub Action Workflow files.
Cancel
Authorize git-ecosystem
Authorizing will redirect to
http://127.0.0.1:63625




If changement :



git status

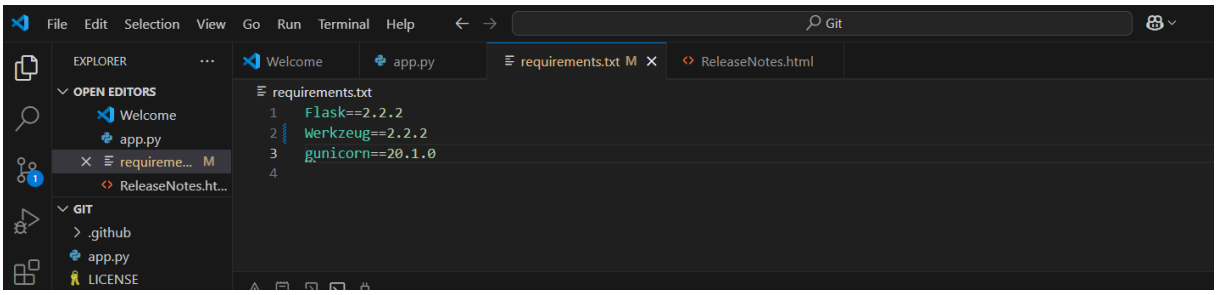
```
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\aciocan\Documents\Git>
```

We see the app was changed

git add app.py

git commit -m ‘changement texte’

git push




*Que fait Azure quand on pousse du code ?*

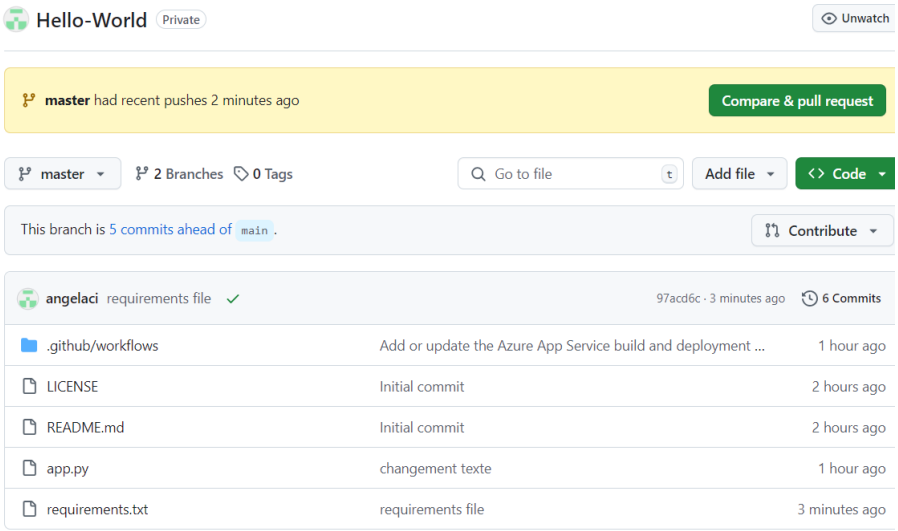
Azure App Service :

1. Télécharge le code depuis GitHub
2. Lit requirements.txt
3. Installe les dépendances Python
4. Démarre l’application Flask avec Gunicorn
5. Expose l’application via HTTPS

Sans qu’on ne touche à un serveur.

pip install -r requirements.txt

```
git add requirements.txt

git commit -m 'requirements file'

if error(

git pull origin master --rebase

git push origin master)
```



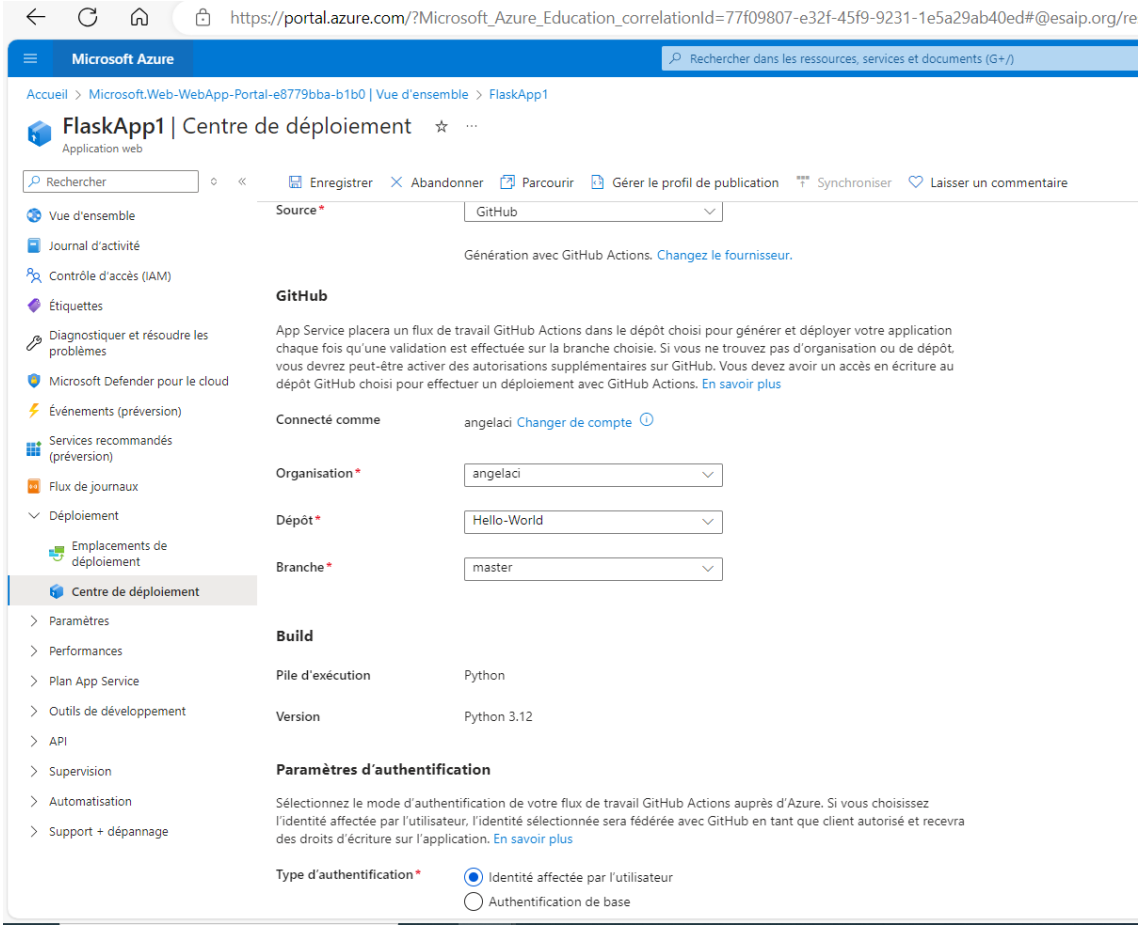

Dans **Centre de déploiement**, tu choisis :

- GitHub
- le dépôt
- la branche (main ou master)

Azure :

- surveille le dépôt GitHub
- détecte les changements
- déclenche un déploiement automatique

C’est un mini CI/CD.




Avec Azure App Service, vous déployez une application web sans gérer de serveur. Le cloud s'occupe de l'infrastructure, vous vous concentrez sur le code.

ccueil > FlaskApp1

# FlaskApp1 | Configuration

Application web

Rechercher

- Vue d'ensemble
- Journal d'activité
- Contrôle d'accès (IAM)
- Étiquettes
- Diagnostiquer et résoudre les problèmes
- Microsoft Defender pour le cloud
- Événements (préversion)
- Services recommandés (préversion)
- Flux de journaux
- Déploiement
  - Emplacements de déploiement
  - Centre de déploiement
- Paramètres
  - Variables d'environnement
  - Configuration

Actualiser | Enregistrer | Abandonner | Laisser un commentaire

Cliquez ici pour effectuer une mise à niveau vers une référence ultérieure et activer des fonctionnalités supplémentaires.

| | |
|---|---|
| Pile | Python |
| Version principale | Python 3 |
| Version secondaire | Python 3.12 |
| Commande de démarrage | gunicorn --bind=0.0.0.0 --timeout 600 app:app |

Indiquez une commande de démarrage facultative à exécuter lors du démarrage du conteneur. En sa

**Paramètres de plateforme**

| | |
|---|---|
| Informations d'identific... | Activé / Désactivé |
| Informations d'identific... | Activé / Désactivé |

Désactivez l'authentification de base pour l'accès FTP et SCM. En savoir plus

| | |
|---|---|
| État FTP | FTPS uniquement |

InPrivate (4) | flaskapp1.azurewebsites.net

https://flaskapp1.azurewebsites.net

Hello, World! Hello Everyone!