

Déploiement Heroku

SOMMAIRE

1. Fichiers nécessaires
2. Création d'un référentiel Git (si vous ne l'avez pas encore fait)
3. Créez un compte Heroku
4. Installation de l'interface de ligne de commande Heroku (CLI)
5. Connectez-vous à Heroku
7. Vérifiez s'il fonctionne6. Déployez l'application

Maintenant que nous avons notre application, nous sommes prêts à commencer à déployer celle-ci sur Heroku.

1. Fichiers nécessaires

Tout d'abord, vous devez ajouter des fichiers qui permettent à Heroku d'installer les exigences nécessaires et d'exécuter l'application.

Create a Git Repo

Note: Heroku allows deployment using Git or Docker.

```
git init
```

Build your App

```
import streamlit as st
import pandas as pd
import numpy as np

def main():
    # Your Code
    if __name__ == "__main__":
        main()
```

Test your App (Local Environment)

```
streamlit run app.py
```

requirements.txt

Le fichier requirements.txt contient toutes les bibliothèques qui doivent être installées pour que le projet fonctionne. Ce fichier peut être créé manuellement en parcourant tous les fichiers et en recherchant les bibliothèques utilisées ou en utilisant automatiquement quelque chose comme [pipreqs](#) .

```
pipreqs <directory-path>
```

Le fichier des exigences doit ressembler à ceci:

```
streamlit==0.79.0
pandas==1.2.3
numpy==1.18.5
matplotlib==3.3.2
seaborn==0.11.0
scikit_learn==0.24.1
```

setup.sh et Procfile

En utilisant les fichiers **setup.sh** et **Procfile**, vous pouvez indiquer à Heroku les commandes nécessaires pour démarrer l'application.

Dans le fichier **setup.sh**, nous allons créer un dossier **streamlit** avec un **credentials.toml** et un fichier **config.toml**.

```
mkdir -p ~/.streamlit/

echo "\
[general]\n\
email = \"your-email@domain.com\"\n\
" > ~/.streamlit/credentials.toml

echo "\
[server]\n\
headless = true\n\
enableCORS=false\n\
port = $PORT\n\
" > ~/.streamlit/config.toml
```

Le Procfile est utilisé pour exécuter d'abord setup.sh, puis pour appeler *streamlit run* pour exécuter l'application.

```
web: sh setup.sh && streamlit run app.py
```

Pour plus d'informations sur la façon d'écrire un fichier Procfile, consultez la [documentation officielle](#) .

2. Création d'un référentiel Git (si vous ne l'avez pas encore fait)

Heroku permet le [déploiement à l'aide de nombreuses technologies différentes](#), notamment Git et Docker. Dans cet article, nous utiliserons Git car c'est simple.

Un dépôt Git peut être créé avec `git init`.

```
C:\Users\#####\Desktop\dataset>git init
Initialized empty Git repository in C:/Users/#####/Desktop/dataset/.git/
```

3. Créez un compte Heroku

Ensuite, vous devrez créer un compte Heroku [ici](#). En général, l'utilisation d'Heroku est gratuite, mais pour obtenir plus de fonctionnalités telles que les métriques d'application ou le SSL gratuit, vous devrez payer.

Vous pouvez trouver un aperçu détaillé de tous les différents niveaux sur [leur page de tarification](#).

4. Installation de l'interface de ligne de commande Heroku (CLI)

Maintenant que vous avez un compte, vous devrez [télécharger l'interface de ligne de commande Heroku](#).

5. Connectez-vous à Heroku

Après avoir installé la **CLI**, vous pouvez vous connecter à la **CLI** en utilisant votre [compte Heroku](#). Pour cela, ouvrez un terminal, déplacez-vous dans le dossier de l'application puis exécutez la **connexion heroku**. Vous serez invité à appuyer sur un bouton et après avoir fait, vous serez redirigé vers un écran de connexion dans votre navigateur par défaut.

```
C:\Users\#####\Desktop\dataset>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/browser/51ce4b4c-bd75-4fec-
b67e-2ea39e00f36d
Logging in... done
Logged in as <xyz>
```

6. Déployez l'application

Enfin, vous pouvez déployer votre application sur heroku en exécutant **heroku create** pour créer une instance Heroku.

Enfin, transmettez le code à cette instance à l'aide des commandes suivantes.

```
git add .
git commit -m "some message"
git push heroku master
```

Lorsque vous exécutez `git push heroku master`, vous devriez remarquer qu'il devrait détecter automatiquement que vous avez une application Python et qu'il devrait installer tous les packages dans le `requirements.txt`. Une fois la commande terminée, vous devriez voir quelque chose de similaire à ce qui suit:

```
remote: -----> Launching...
remote:          Released v3
remote:          https://glacial-meadow-66539.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/glacial-meadow-66539.git
* [new branch]      master -> master
```

7. Vérifiez son fonctionne

Vous pouvez vérifier si l'application a été déployée avec succès à l'aide de

Vérifiez si l'application est en cours d'exécution

```
heroku ps: scale web =1
```

Enfin, l'application peut être ouverte avec *heroku ouvert* .

Cela ouvrira l'application en utilisant votre navigateur par défaut.

Félicitations: Application déployée