

Parcours développeur d'application - Python

Projet 10 : Déployez votre application sur un serveur comme un.e pro !

1 Description du projet

Le projet 10 se place dans la continuité du projet 8 dans lequel nous avons développé une application web pour la startup Purbeurre. Jusqu'alors, cette application était déployée sur la plateforme Heroku. Le but de ce nouveau projet est de redéployer cette application en n'utilisant non plus une « Plateforme as a Service » (PaaS) tel que l'est Heroku, mais via notre propre serveur en utilisant le modèle « Infrastructure as a Service » (IaaS).

2 Configuration du serveur et de l'application

Pour réaliser ce projet, j'ai repris l'application réalisée dans le cadre du projet 8 sans effectuer de modifications majeures.

2.1 Configuration du serveur

Comme suggéré dans le cours associé à ce projet, j'ai choisi de réserver un espace serveur sur le site Digital Ocean. Ayant l'habitude de travailler sous Ubuntu 16.04, j'ai choisi la même configuration pour le serveur. J'ai ensuite créé un nouvel utilisateur et effectué toutes les étapes pour permettre la connexion en ssh avec cet utilisateur sur le serveur.

2.2 Téléchargement de l'application

Une fois le serveur configuré, j'ai téléchargé l'application en clonant son repository GitHub. J'ai ensuite procédé à l'installation des paquets nécessaires au fonctionnement de l'application, ainsi qu'aux différentes modifications des fichiers de configurations préconisés dans le cours associé au projet.

La difficulté rencontrée à cette étape est venue de l'arborescence de mon projet. En effet, lors de la création de mon repository sur GitHub, j'ai choisi de créer un dossier principal contenant à la fois le dossier du projet et des documents annexes tels que les livrables. L'application est donc contenue dans un sous dossier. Après quelques révision sur le fonctionnement du « Python path », j'ai pu effectuer les ajustements de chemin dans les fichiers de configurations (travis, supervisor etc.) afin que l'application fonctionne correctement.

3 Monitoring

Afin de contrôler le bon fonctionnement du serveur, l'utilisation d'outils de monitoring est indispensable. Pour effectuer ces contrôles, j'ai utilisé deux outils ayant un rôle bien distinct :

- Outil de monitoring de Digital Ocean : Cet outil permet de contrôler l'état du serveur en lui même C'est à dire, son statut (allumé/éteint), l'utilisation de la mémoire, du processeur etc.

J'ai configuré des alertes pour être averti dans 3 cas de figures :Lorsque le processeur ou la mémoire vive arrive à saturation. Étant donné le peu de trafic qu'il est censé y avoir sur

l'application, ces alertes signifieraient probablement que l'application crée un dysfonctionnement au niveau du serveur. Enfin, une troisième alerte en cas d'utilisation élevée de la mémoire de stockage, ce qui signifierait qu'il faut passer à un espace plus conséquent sur Digital Océan.

- Outil de contrôle des Logs : J'ai utilisé l'outil Sentry pour récupérer et filtrer les logs issus de Django. Cet outil est paramétré pour m'envoyer une notification lorsque Django a rencontré une erreur. J'ai également ajouté, à titre de test, une notification lorsqu'un utilisateur effectue une recherche sur l'application. Dans le cas où l'application connaîtrait du succès, il serait préférable de désactiver cette notification.

4 Automatisation de la mise à jour de la base de données

Pour finir, la réalisation de ce projet nécessite la mise en place d'une récupération automatique des données de OpenFoodFacts de façon hebdomadaire afin de mettre à jour la base de données de l'application. Pour faire cela, j'ai utilisé CRON, le gestionnaire de tâche présent sur Ubuntu pour programmer le lancement d'un script python tous les mercredi soir à 23h59.

La commande CRON permettant ceci est la suivante :

```
59 23 * * 3 python3 ~/Purbeurre/pur_beurre/substitute/update_db.py
```

La difficulté rencontrée lors de cette étape a été de réaliser un script pouvant être lancé directement via CRON. Jusqu'alors, les scripts interagissant avec la base de données dans mon projet utilisaient tous l'ORM de Django et nécessitaient donc d'être lancés depuis la console de Django. Afin de résoudre ce problème, j'ai dû créer un nouveau script permettant d'interagir directement avec la base de données via des commandes SQL et la librairie psycopg2 de Python.

5 Livrables associés au projet

L'application web est disponible à l'adresse suivante : <http://206.189.119.224/>

Les différentes copies d'écran demandées dans le cadre de ce projet sont stockées à l'adresse suivante : <https://github.com/NanroYahel/P10-Deployez-votre-application-sur-un-serveur-comme-un-e-pro-/tree/master/livrables>