

Site internet déployé.

Choix technologiques et étapes d'installation

- Création d'une Virtual Machine (ec2) avec aws d'amazon.
- Installation d'Ubuntu 18.04 sur la VM.
- Mise à jour des paquets , installation de python et postgresSQL.

Création de la configuration de production

Installation du serveur web : nginx

NGINX est un [système asynchrone](#) par opposition aux serveurs synchrones où chaque requête est traitée par un [processus](#) dédié.

Installation de supervisor

Supervisor est un système qui permet de contrôler des processus dans un environnement Linux. Ici nous l'utilisons pour contrôler et redémarrer au besoin Gunicorn (serveur HTTP Python qui utilise les spécifications WSGI, utilisé par Django).

Installation de Travis

Travis est un service d'automatisation. Travis crée un environnement équivalent à celui de notre application et y fait tourner les tests. Nous l'interfaçons avec notre dépôt Github. Il surveille dans notre cas la branche `staging` où désormais toutes les modifications de l'application seront faites.

Configuration de Travis

Création d'un [compte](#). Ajout sur la dashboard de Travis de notre dépôt Github:

Monitorer le serveur : mise en place de Newrelic

[Newrelic Infrastructure](#) offre la possibilité de monitorer notre serveur : Etat de la CPU, de la mémoire vive, le load average etc.

Surveillance de l'application Django oc_projet_10 : Sentry

Les erreurs de l'application seront reportées sur sentry.

Tâche planifiée

Notre serveur virtuel oc_projet_10 peut vouloir mettre à jour sa base de données régulièrement avec des nouvelles données importées d'OpenFoodFacts.

Nous récupérons la liste de tous les produits enregistrer dans la base de données et la comparons avec le produit correspondant dans openfoodfacts. Puis nous mettons à jour les produits dans notre propre base de données.

Script shell et Cron job

La tâche s'exécutera tous les dimanches à minuit.