

## Projet n°10 Déploiement d'une application django

---

Dans le cadre de ce projet j'ai réalisé le déploiement de mon application sur un serveur distant et pour ce faire j'ai réalisé l'opération sur mon serveur dédié.

Afin de respecter les recommandations j'ai reconfiguré mon serveur, pour basculer en serveur Http Nginx.

La procédure m'a pris un peu de temps entre les sauvegardes et les uploads des projets déjà présents.

Une fois l'opération terminée j'ai créé un espace pour mon application et je l'ai associé à un sous-domaine de l'un de mes domaines "pb.yoso.fr" et y ai associé une base de données postgresql pour lequel j'ai créé un utilisateur et un mot de passe .

J'ai créé un utilisateur pour un accès ftp pour l'envoi de document directement sur le serveur sans passer par git, notamment dans le cadre du fichier de configuration.

Cette étape terminée j'ai suivi des tutoriels et consulté la documentation de django pour connaître la méthodologie à respecter pour réaliser un déploiement sur serveur de django. Cette étape avait pour objectif de réduire les erreurs lors de ça mise en place.

L'étape suivante fut l'installation des dépendances pour la gestion du projet mise en place de git et supervisor, python étant déjà présent sur mon serveur je n'avais pas besoin de l'installer.

Avant le téléchargement de mes sources j'ai configuré le projet en conséquence en séparant la configuration en 2 modules une de développement et une de production et nettoyé le projet en retirant par exemple whitenoise.

Ma préparation finie j'ai récupéré le projet depuis github vers mon serveur et mis en place les fichiers et dossiers aux bons emplacements.

J'ai renseigné mon fichier de configuration de production pour la connexion à la base de données et déposé le tout sur le serveur par ftp.

Ses différentes étapes m'ont fait reprendre mon code pour qu'il puisse avoir un peu plus de clarté. Ce qui m'a pris le plus de temps c'est la recherche d'informations et l'analyse de la procédure. Néanmoins la récupération de mes sources m'ont posé une légère contrainte dans ma façon de faire, étant connecté en root, je ne pouvais modifier le dossier en déposant mon fichier de configuration depuis le ftp, le temps que je réalise cela j'avais commencé à appliquer des chmod sur l'ensemble de ma structure.

Le système de droit en écriture étant bien défini sous linux, mon erreur était de créer des

dossiers en root et d'essayer d'y accéder par le ftp en tant que utilisateur. La méthode était de me connecter sous l'identifiant de l'utilisateur sur le serveur puis de mettre en place les sources pour avoir accès depuis le ftp en tant que administrateur de l'espace et avoir les droits d'écriture.

Pour le fonctionnement du site j'ai configuré NginX et tester les redirections vers mes fichiers l'ayant testé en amont il me restait plus cas le mettre en application.

Pour pallier aux éventuels erreurs du serveur, j'ai écrit une configuration pour supervisor afin qu'il suive l'activité de mon application et de le redémarrer en cas de panne lors du fonctionnement ou de redémarrage du serveur.

Sur ses étapes celui qui m'a posé quelques difficultés c'est supervisor, il ne me créait pas mes variables d' environnement, je pensais avoir fait une erreur de configuration dans la mise en place de mon serveur ou une non prise en charge de supervisor depuis mon serveur. Mais dans mon cas s'était encore une erreur d' inattention de ma part lors de l'écriture des commandes, voulant faire un code propre j'ai ajouté des espaces là où il ne fallait pas. La commande n' était donc non prise en compte par supervisor, me poussant ainsi à entamer une procédure de recherche de bugs sur mon déploiement.

Une fois cet ensemble opérationnel, j'ai mis en place des outils de surveillance et de contrôle de versionning avec travis.

La prise en main de l'outil n'a pas été complexe, en suivant la documentation et une vidéo, il a été assez simple de mettre le système en place. L'outil permet de surveiller le code qui est transmis avant validation pour être sur d'avoir un code opérationnel avant une mise en production évitant ainsi des bugs lorsque l'ont apport des modifications ou que l'on ajoute des fonctionnalités.

Pour la surveillance du code j'ai utilisé sentry, qui se révèle très efficace pour la détection d'erreurs dans la saisie du code, un message est transmis à la personne ainsi désigner ce qui permet une très grande réactivité, là encore ça mise en place ne m'a pas pris beaucoup de temps en respectant la procédure d'utilisation.

Pour finir ma mise en place j'ai effectué des tests de fonctionnement puis créer une tâche cron pour la mise à jour de ma base de données ayant écrit en amont une commande django il me suffisait de faire appel à elle pour exécuter la mise à jour.

Ce projet a été une très belle expérience me permettant de découvrir de nouveaux outils et surtout une nouvelle méthodologie de travail, pour la quel je n'avais jamais opté pour mes divers projets.

Il est important à ce jour de se tenir à jour sur les avancer technologique et les nouvelles pratiques de programmation facilitant ainsi le déploiement de ses applications.