

FAQ Valoo

Application web - FAQ

Dossier d'exploitation

Version 1.0

Auteur
Kilian Florin

TABLE DES MATIERES

1 - Versions.....	3
2 - Introduction	4
2.1 - Objet du document.....	4
3 - Pré-requis.....	5
3.1 - Système	5
3.1.1 - <i>Serveur Web</i>	5
3.1.1.1 - Caractéristiques techniques	5
3.2 - Bases de données	5
3.3 - Autres ressources	5
4 - Procédure de déploiement.....	6
4.1 - Déploiement de l'Application Web	6
4.1.1 - <i>Environnement de l'application web</i>	6
4.1.1.1 - Environnement virtuel	6
4.1.2 - <i>Clone du repository sur GitHub</i>	6
4.1.3 - <i>Installation des pré-requis</i>	6
4.1.4 - <i>Configuration de l'application</i>	6
4.1.4.1 - Fichier de configuration de l'application Django en production	6
4.1.4.2 - Fichier production.py	7
4.1.5 - <i>Configuration de Nginx</i>	7
4.1.6 - <i>Configuration de Supervisor</i>	8
5 - Procédure de démarrage / arrêt.....	9
5.1 - Base de données	9
5.2 - Application web	9
Procédure de mise à jour	10
5.3 - Base de données	10
5.4 - Application web	10
6 - Procédure de sauvegarde et restauration	11
6.1 - Sauvegarde de la base de données.....	11
6.2 - Restauration de la base de données.....	11

1 - VERSIONS

Auteur	Date	Description	Version
Kilian Florin	16/10/2018	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application FAQ Valoo...

Le document ci-présent a pour but d'être un support à l'installation et au déploiement de l'application web FAQ Valoo.

3 - PRE-REQUIS

3.1 - Système

3.1.1 - Serveur Web

Serveur physique ou virtuel hébergeant l'application web.

3.1.1.1 - Caractéristiques techniques

CPU : 1 vCore à ~3.1 Ghz

RAM : ~2 Go

Stockage : 2 Go disponible.

3.2 - Bases de données

Pour le besoin de notre application, nous utiliserons SQLite. Il n'y a donc aucun prérequis nécessaire.

3.3 - Autres ressources

Les logiciels suivants doivent être accessibles et à jour :

- **Python** : version 3.6.0 ou +
- **Virtualenv** : version 15.0.0
- **Supervisor** : version 3.2.0-2 ou +
- **Nginx** : version 1.9.15 ou +
- **Git** : version 1:2.7.4 ou +

4 - PROCEDURE DE DEPLOIEMENT

4.1 - Déploiement de l'Application Web

4.1.1 - Environnement de l'application web

4.1.1.1 - Environnement virtuel

Afin d'installer l'application proprement, il est nécessaire de lui créer un environnement virtuel :

```
Virtualenv env -p python3
```

```
Source env/bin/activate
```

4.1.2 - Clone du repository sur GitHub

```
Git clone https://github.com/Hideky/FAQ-Valoo.git
```

```
Cd FAQ-Valoo
```

4.1.3 - Installation des pré-requis

```
Pip install -r requirements.txt
```

4.1.4 - Configuration de l'application

4.1.4.1 - Fichier de configuration de l'application Django en production

Le répertoire de configuration de l'application en production doit être créé de la manière suivante :

```
Mkdir faqvaloo/settings
```

Un fichier sera utilisé en substitue du fichier par default de configuration de l'application afin de répondre au besoin de la production, créer le de cette manière :

```
Touch faqvaloo/settings/production.py
```

4.1.4.2 - Fichier *production.py*

Afin de configurer l'application, copier ceci en changeant les valeurs entre chevrons.

```
from . import *
import os

SECRET_KEY = '<Nouvelle clé secrète>'
DEBUG = False
ALLOWED_HOSTS = [<IP du serveur>]
```

4.1.5 - Configuration de Nginx

Afin de permettre à Nginx de distribuer correctement les fichiers statiques, voici comment le configurer :

```
Cd /etc/nginx/
sudo touch sites-available/faqvaloo
sudo ln -s /etc/nginx/sites-available/faqvaloo/etc/nginx/sites-enabled
```

Enfin, changer le contenu de 'sites-available/faqvaloo' pour :

```
server {

    listen 80; server_name 178.62.119.70;
    root /home/<votre utilisateur>/faqvaloo;

    location /static {
        alias /home/<votre utilisateur>/faqvaloo/static;
    }

    location / {
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_redirect off;
        if (!-f $request_filename) {
            proxy_pass http://127.0.0.1:8000;
            break;
        }
    }
}
```

Puis rechargez Nginx avec :

```
sudo service nginx reload
```

4.1.5.1 - Création du dossier 'static' pour Nginx

Afin de servir les fichiers les plus courant de manière plus rapide, nous créons un dossier « static » dans lequel Nginx va directement chercher les ressources sans passer par Django.

```
python manage.py collectstatic
```

4.1.6 - Configuration de Supervisor

Créez le fichier de configuration relatif à notre application avec ceci :

```
sudo touch /etc/supervisor/conf.d/faqvaloo-gunicorn.conf
```

Puis éditez le de la sorte :

```
[program:faqvaloo-gunicorn]
command = /home/<Votre Utilisateur>/env/bin/gunicorn faqvaloo.wsgi:application
user = <Votre utilisateur>
directory = /home/<Votre utilisateur>/faqvaloo
autostart = true
autorestart = true
environment = DJANGO_SETTINGS_MODULE='faqvaloo.settings.production'
stderr_logfile = /var/log/supervisor/faqvaloo_stderr.log
stdout_logfile = /var/log/supervisor/faqvaloo_stdout.log
```


5 - PROCEDURE DE DEMARRAGE / ARRET

5.1 - Base de données

Pour arrêter

Sudo service postgresql stop

Pour démarrer

Sudo service postgresql start

5.2 - Application web

Pour arrêter

Sudo supervisorctl stop faqvaloo-gunicorn

Pour démarrer

Sudo supervisorctl start faqvaloo-gunicorn

PROCEDURE DE MISE A JOUR

5.3 - Base de données

```
# Sauvegarder la base actuelle
Su - postgres
Pg_dumpall > dump.sql
Cp ~postgres/dump.sql ~
# Migrer
Apt-get remove postgresql-x.x.x
Apt-get install postgresql-x.x.x
# Restaurer
Su -postgres
Psql < dump.sql
```

5.4 - Application web

```
Git pull
```

6 - PROCEDURE DE SAUVEGARDE ET RESTAURATION

6.1 - Sauvegarde de la base de données

```
Python manage.py dumpdata > dump.json
```

6.2 - Restauration de la base de données

```
Python manage.py loaddata dump.json
```