

Raspal

Mise en production de l'alarme sur un raspberry

Dossier d'exploitation

Version 1

Auteur

SIMONIN Cyril

Développeur

TABLE DES MATIÈRES

| | |
|--|-----------|
| 1 - Versions..... | 3 |
| 2 - Introduction..... | 4 |
| 2.1 - Objet du document..... | 4 |
| 2.2 - Références..... | 4 |
| 3 - Pré-requis..... | 5 |
| 3.1 - Système..... | 5 |
| 3.1.1 - Serveur de Web et Base de donnée..... | 5 |
| 3.2 - Bases de données..... | 5 |
| 3.3 - Web-services..... | 5 |
| 4 - Procédure de déploiement..... | 6 |
| 4.1 - Déploiement de l'application..... | 6 |
| 4.1.1 - Configuration du raspberry..... | 6 |
| 4.1.1.1 - Mise à jour du système..... | 6 |
| 4.1.1.2 - Installation des dépendances systemes..... | 6 |
| 4.1.1.3 - Installation de virtualenv..... | 6 |
| 4.1.1.4 - Configuration de la base de donnée..... | 7 |
| 4.1.1.5 - Configuration d'opencv pour environnement virtuel..... | 7 |
| 4.1.2 - Installation et configuration de l'application Raspal..... | 8 |
| 4.1.2.1 - Création de l'environnement virtuel..... | 8 |
| 4.1.2.2 - Téléchargement de l'application..... | 8 |
| 4.1.2.3 - Installation des dépendances de l'application..... | 8 |
| 4.1.2.4 - Création et configuration du fichier production.py..... | 8 |
| 4.1.2.5 - Préparation de la mise en production de l'application..... | 10 |
| 4.1.2.6 - Configuration nginx..... | 10 |
| 4.1.2.7 - Configuration de supervisor..... | 11 |
| 5 - Procédure de démarrage / arrêt..... | 13 |
| 5.1 - Application web..... | 13 |
| 5.1.1 - Lancement de l'application..... | 13 |
| 5.1.2 - Arrêt de l'application..... | 13 |
| 6 - Fonctionnalités..... | 14 |
| 6.1 - Création d'un utilisateur administrateur..... | 14 |
| 6.2 - Suppression d'un utilisateur administrateur..... | 14 |

1 - VERSIONS

| Auteur | Date | Description | Version |
|---------------|------------|----------------------|---------|
| SIMONIN Cyril | 21/12/2019 | Création du document | 1 |

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application RaspAI

Ce document présente la mise en production du système RaspAI développé par SIMONIN Cyril

2.2 - Références

Pour de plus amples informations, se référer :

1. **DCF – RaspAI** : Dossier de conception fonctionnelle de l'application
2. **DCT – RaspAI** : Dossier de conception technique de l'application

3 - PRÉ-REQUIS

3.1 - Système

3.1.1 - Serveur de Web et Base de donnée

La base de données et le serveur web seront hébergés sur le raspberry

OS: Raspbian Buster

3.2 - Bases de données

La base de données et schéma doit être accessible et à jour ;

- **Postgresql** : version 11.5

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **Django** : Version 3.0
- **Python** : Version 3.7
- **Gunicorn**: Version 20.0
- **Nginx**: Version 1.14

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement de l'application

4.1.1 - Configuration du raspberry

4.1.1.1 - Mise à jour du système

Le système doit et mis à jour avant la mise en production

La commande : **sudo apt-get update && sudo apt-get upgrade**

Votre système est maintenant à jour.

4.1.1.2 - Installation des dépendances systemes

Afin que l'application soit complètement fonctionnelle il est nécessaire d'installer les dépendances ci-dessous avec la commande.

```
~# sudo apt-get install python3-pip python3-dev libpq-dev postgresql postgresql-contrib  
python3-numpy libatlas3-base libgfortran5 libpango-1.0-0 libatk1.0-0 libcairo-gobject2  
libpangocairo-1.0-0 libtiff5 libwebp6 libavcodec58 libavutil56 libavformat58 libgdk-  
pixbuf2.0-0 libgtk-3-0 libilmbase23 libjasper1 libcairo2 libswscale5 libopenexr23 libhdf5-  
dev libhdf5-serial-dev libhdf5-103 libatlas-base-dev libjasper-dev ffmpeg nginx supervisor  
git
```

4.1.1.3 - Installation de virtualenv

Afin de déployer l'application dans un environnement virtuel il est nécessaire d'installer virtualenv, l'installation de virtualenv ce fait en tapant la commande.

```
~# sudo pip3 install virtualenv
```

4.1.1.4 - Configuration de la base de donnée

L'application utilise une base de données Postgresql il est donc nécessaire de configurer un utilisateur et une base de données.

Pur ce faire il vous suffit d'écrire les commandes suivantes en **remplaçant les parties en gras** par ce que vous désirez mais attention **enregistrer bien ces informations nous en auront besoin pour configurer l'application.**

```
sudo -u postgres psql
CREATE DATABASE Nom_De_La_Base_De_Donnée;
CREATE USER Nom_Utilisateur WITH PASSWORD 'Mot_De_Passe';
ALTER ROLE Nom_Utilisateur SET client_encoding TO 'utf8';
ALTER ROLE Nom_Utilisateur SET default_transaction_isolation TO 'read committed';
ALTER ROLE Nom_Utilisateur SET timezone TO 'Europe/Paris';
GRANT ALL PRIVILEGES ON DATABASE Nom_De_La_Base_De_Donnée TO Nom_Utilisateur;
\q
```

Vous venez de créer une base de données et un utilisateur.

4.1.1.5 - Configuration d'opencv pour environnement virtuel

L'utilisation d'opencv sur raspberry nécessite des modifications particulières sur le fichier bashrc.

Ouvrir le fichier **bashrc** présent sur votre dossier utilisateur

```
~#nano ~/.bashrc
```

à la fin du fichier ajoutez la ligne suivante

```
export LD_PRELOAD=/usr/lib/arm-linux-gnueabi/hf/libatomic.so.1
```

Fermer et enregistrer le fichier et lancer la commande ci-dessous

```
source ~/.bashrc
```

4.1.2 - Installation et configuration de l'application Raspal

L'application fonctionne dans un environnement virtuel pour éviter des conflits avec d'autres applications qui seraient déployées par la suite.

L'utilisation du gestionnaire d'environnement virtuel `virtualenv` est utilisée avec cette application.

Suivez les étapes dans l'ordre et l'installation se déroulera correctement.

4.1.2.1 - Création de l'environnement virtuel

Afin de créer l'environnement virtuel et d'ajouter les dépendances de l'application utiliser les commandes suivantes:

```
~#mkdir raspal
```

```
~#cd raspal
```

```
~#virtualenv -p python3 env
```

L'environnement virtuel a été créé par `virtualenv` maintenant il vous faut activer l'environnement virtuel

```
~#source env/bin/activate
```

4.1.2.2 - Téléchargement de l'application

Il est nécessaire de télécharger l'application via GIT, l'application est téléchargée dans le `"/home/nom_utilisateur"`

```
~# git clone https://github.com/Cyril45/P13-Raspal-.git
```

```
~# mv P13-Raspal-/ raspal/
```

4.1.2.3 - Installation des dépendances de l'application

Afin d'ajouter les dépendances de l'application utiliser la commande.

```
~#cd raspal
```

```
~#pip install -r requirements.txt
```

L'environnement virtuel a été créé par `pipenv` et l'installation des dépendances a été faite.

4.1.2.4 - Création et configuration du fichier `production.py`

Le fichier `production.py` contiendra les configurations pour la production.

Il vous faut créer une clé une clef pour Django, voici les commandes à exécuter

```
~#python3
```


Un terminal python va s'ouvrir exécuter ces commandes:

```
>>> import random, string
>>> "".join([random.choice(string.printable) for _ in range(40)])
```

Une suite de caractere alphanumerique s'affiche sur votre terminal, **sauvegarder la!**

Sortez du terminal python avec la commande

```
>>>exit()
```

Passons à la création et à la configuration de production.py avec la commande

```
~# nano raspal/settings/production.py
```

Et ajouter les informations de l'environnement de production comme ci-dessous en modifiant les champs en gras par vos informations:

```
from . import *

# django appication config
SECRET_KEY = 'CLE_DJANGO_CREER_PRECEDEMMENT'
DEBUG = False
ALLOWED_HOSTS = ['Votre_Adesse_IP_PUBLIC']

# django Databases config
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'NOM_DE_LA_BASE_DE_DONEE',
        'USER': 'NOM_UTILISATEUR_BASE_DE_DONNEES',
        'PASSWORD': 'MOT_DE_PASSE_BASE_DE_DONNEES',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}

# Configuration SMTP
EMAIL_HOST = 'serveur_smtp'
EMAIL_USE_TLS = True # Configuration a verifier chez votre fournisseur de mail
EMAIL_PORT = 587 # Configuration a verifier chez votre fournisseur de mail
EMAIL_HOST_USER = 'Votre_adresse_email'
EMAIL_HOST_PASSWORD = 'Votre_Mo_De_Passe_Email'
```

le fichier de configuration est maintenant fonctionnel

4.1.2.5 - Préparation de la mise en production de l'application

Pour créer la base de données nous allons utiliser le fichier de production que nous venons de créer pour ce faire il suffit de taper la commande ::

```
~#export DJANGO_SETTINGS_MODULE='raspal.settings.production'
```

Afin de créer la base de données il est nécessaire de lancer la commande:

```
~# python manage.py migrate
```

Quand l'interface affiche **"Applying sessions.0001_initial... OK"** appuyer sur CTRL+C

```
~# python manage.py makemigrations
```

Quand l'interface affiche **"- Create model MyUser"** appuyer sur CTRL+C

```
~# python manage.py migrate
```

Quand l'interface affiche **"Applying alarme.0001_initial... OK"** appuyer sur CTRL+C

Normalement ce n'est pas nécessaire de faire des CTRL+C mais Django ne gère pas les threads et n'arrive pas à fermer les Thread liés à l'alarme.

4.1.2.6 - Configuration nginx

Il est maintenant nécessaire de configurer nginx pour votre alarme RaspAl

Créer le fichier raspal à l'aide de la commande:

```
~#sudo nano /etc/nginx/sites-available/raspal
```

Coller le code ci-dessous (et modifier les parties en gras par vos informations) dans le fichier et enregistrer:

```
server {  
    listen 80;
```

```

server_name IP_ADRESS;
root /home/nom_utilisateur/raspal/raspal;
location / {
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_redirect off;
    proxy_pass http://127.0.0.1:8000;
}
}

```

La configuration pour le site web est créée il faut maintenant en créer un lien symbolique vers le dossier sites-enabled avec la commande.

```
~#sudo ln -s /etc/nginx/sites-available/raspal /etc/nginx/sites-enabled
```

Maintenant redémarrer le service à l'aide de la commande:

```
~#sudo service nginx reload
```

La configuration de nginx est terminée.

4.1.2.7 - Configuration de supervisor

Supervisor est une application de surveillance de processus elle permet de lancer une application au démarrage d'un serveur et de relancer une application si celle-ci s'arrête.

Créer le fichier "raspal-gunicorn.conf" dans le dossier de configuration de supervisor.

```
~# sudo nano /etc/supervisor/conf.d/raspal-gunicorn.conf
```

et y ajouter les informations suivantes (les informations en gras sont à modifier par vos informations)

```

[program:raspal-gunicorn]
command = /home/Nom_utilisateur/raspal/env/bin/gunicorn raspal.wsgi:application
user = nom_utilisateur
environment= DJANGO_SETTINGS_MODULE='raspal.settings.production',
LD_PRELOAD="/usr/lib/arm-linux-gnueabi/libatomic.so.1"
directory = /home/Nom_utilisateur/raspal/raspal
autostart = true

```

autorestart = true

Afin de faire prendre en charge la nouvelle configuration par supervisor les commande suivantes doivent être utilisées:

~#sudo supervisorctl reread

~#sudo supervisorctl update

5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 - Application web

5.1.1 - Lancement de l'application

Le lancement de l'application à l'aide de la commande

~#sudo supervisorctl start raspal-gunicorn

5.1.2 - Arrêt de l'application

L'arrêt de l'application à l'aide de la commande

~#sudo supervisorctl stop raspal-gunicorn

6 - FONCTIONNALITÉS

6.1 - Création d'un utilisateur administrateur

Pour des raisons de sécurité la création d'un compte utilisateur se fait en ligne de commande sur le raspberry.

Pour créer un utilisateur il est nécessaire de se placer dans l'environnement virtuel, les commandes sont les suivantes;

```
~#cd home/NOM_UTILISATEUR/raspal/
```

```
~#source env/bin/activate
```

```
~# cd raspal
```

```
~# python manage.py create_superuser
```

Le programme va vous demander les informations de compte quand le message "votre compte vient d'être créé" appuyer sur CTRL+C pour terminer

Le compte admin est créé et fonctionnel

6.2 - Suppression d'un utilisateur administrateur

Pour des raisons de sécurité la suppression d'un compte utilisateur administrateur se fait en ligne de commande sur le raspberry.

Pour créer un utilisateur il est nécessaire de se placer dans l'environnement virtuel, les commandes sont les suivantes;

```
~#cd home/NOM_UTILISATEUR/raspal/
```

```
~#source env/bin/activate
```

```
~# cd raspal
```

```
~# python manage.py delete_superuser
```

Le programme va vous demander les informations de compte quand le message après les avoir renseignés appuyer sur CTRL+C pour terminer

Le compte admin est maintenant supprimé