YAHIA-BENATTIA Mehdi

Compte-rendu

Serveur WEB

# 1. Contexte

Nous avons mis en place un serveur web sur une **machine virtuelle Ubuntu Desktop**.

# 2. Objectif

Notre objectif est l’installation d’un serveur Web fonctionnel sous Ubuntu et la réussite de l’hébergement de deux sites web des groupes SLAM Loris L. et Loris M.

# 3. Architecture

**- OS** : Ubuntu Desktop (VM sur vSphere)  
**- Hébergeur** : Nginx (service gunicorn)   
**- Conteneurisation** : Docker  
- **Application** : Gunicorn

## 3.1 Pourquoi ? :

### Ubuntu :

Ubuntu est une bonne alternative à Windows serveur car premièrement est open-source, il est donc libre d’utilisation et gratuit mais pas qu’il est aussi facile à utiliser, modulable et utilisable pour tout le monde particulier ou entreprise.

### Nginx :

Nginx est plus léger et moins gourmand en ressource, Nginx utilise aussi une architecture asynchrone, il peut avec un seul processus géré des milliers de connexions. Apache était tout aussi valable mais nous avons préférer utiliser Nginx car plus simple à mettre en place.

Docker :

Docker permet également de configurer un système d'exploitation et des versions de logiciels différentes sans avoir à gérer les problèmes de dépendance

Gunicorn :

Gunicorn sert à exécuter un site Python (Django dans notre cas) en production. Il reçoit les requêtes, les envoie à l’application, puis renvoie la réponse. Il est plus rapide et plus stable que le serveur de développement, et permet de gérer plusieurs utilisateurs en même temps.

# 4. Étapes réalisées

* Installation de Ubuntu Desktop
* Installation des paquets essentiels (Docker, Nginx…)
* Création du Docker file avec les dépendances nécessaires (Apache, MySQL, ADMiner…)
* Lancement du conteneur avec port exposé

# 5. Problème rencontré

Un conflit de port persistant avec Apache que nous avions précédemment installé pour héberger le site de Loris.M. Cela a nécessité de vérifier qu’Apache ne redémarre pas automatiquement après avoir éteint la machine, Vérifier les processus (sudo lsof -i :PORT) pour voir si il n’y a pas de conflits de port.

Nous avons pris comme solution de supprimer Apache car il sera présent dans le docker.

# 6. Documentation

Un **script d’installation documenté étape par étape** a été rédigé, contenant toutes les commandes utilisées (Docker, Apache, configuration réseau).

## Création d’un utilisateur root :



Création de mot de passe :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

MP actuelle : Di@blo\_26500

MP ROOT : ROOTDi@blo\_26500 (MP à changer car problème de sécurité)

Activation compte ROOT :



Connection au compte :



## Installation Ningx (pour Djnago) :



Installation PIP :



Instalation de pip, le gestionnaire de paquets Python.

Création environement virtuel : 

La commande python3 -m venv env initialise un environnement local appelé env.

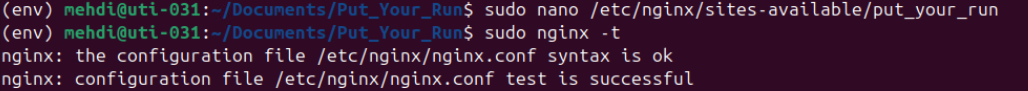
Installation environnement virtuel Python :



source env/bin/activate active l’environnement.

pip install -r requirements.txt installe toutes les bibliothèques nécessaires listées dans le fichier requirements.txt.

Configuration de Nginx pour le site avec un fichier répertoire.



Préparation de la mise en place Django :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Avant de lancer le site, j’ai exécuté la commande python manage.py collectstatic afin de rassembler tous les fichiers statiques (CSS, JS, images…) dans un seul répertoire.

## Installation Gunicorn :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Gunircorn permet la communication entre le serveur web et l’application web Python.

Déploiement du serveur applicatif :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Gunicorn exécute Put\_Your\_Run.wsgi, le point d’entrée de l’application. Le message *"Listening at http://0.0.0.0:8000"* confirme que l’application est active et prête à être servie via Nginx

Configuration site Nginx :

Création du fichier :



Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.



Amélioration des proxys (fait passer par Nginx au lieu de Gunicorn car problème de chargement de CSS)



Mise en place service gunicorn :

Création du fichier service.

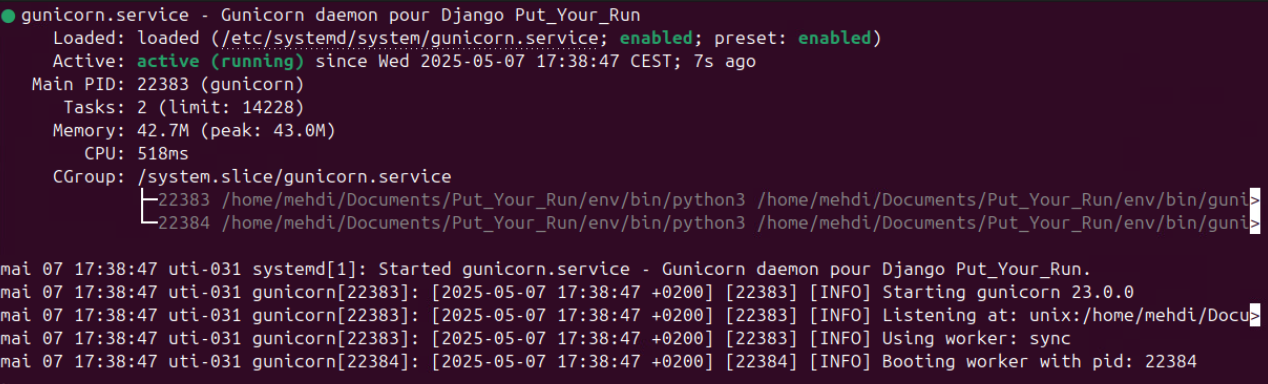


Script pour automatiser le service gunicorn au lancement de la machine

Une image contenant texte, capture d’écran, Page web, logiciel

Le contenu généré par l’IA peut être incorrect.

Teste fonctionnement du service.



## Installation Docker :

Installation des prérequis via APT



Clé GPG de Docker pour garantir ses paquets Docker téléchargés



Ajout du dépôt Docker stable



Installation de Docker Community Edition



Vérification du service Docker



Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Verification de la version de docker



Installation manuelle de Docker Compose



Lancement de docker

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## Site Loris.M

Site du groupe de Loris. M héberger en locale.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Serveur FTP dans le bash (problème d’installation de Filezilla)

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## Image du docker du groupe Loris. M

services:

db:

image: mysql:latest

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: root\_password

MYSQL\_DATABASE: chat

volumes:

- ./mysql\_data:/var/lib/mysql

ports:

- "8014:8013"

healthcheck:

test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]

timeout: 20s

retries: 100

adminer:

image: adminer

restart: always

ports:

- "8082:8080"

depends\_on:

- db

myapp:

build: .

restart: always

ports:

- "8015:80"

environment:

NODE\_ENV: development

volumes:

- .:/app

depends\_on:

db:

condition: service\_healthy

ftp:

image: fauria/vsftpd

container\_name: ftp\_server

restart: always

ports:

- "21:21"

- "21000-21010:21000-21010"

environment:

- FTP\_USER=UTI031user

- FTP\_PASS=UTP031password

- PASV\_ADDRESS=127.0.0.1

- PASV\_MIN\_PORT=21000

- PASV\_MAX\_PORT=21010

- FILE\_OPEN\_MODE=0666

- LOCAL\_UMASK=022

- LOG\_STDOUT=YES

volumes:

- ./ftp\_data:/home/vsftpd

volumes:

mysql\_data: