

- Rapport Virtualisation :

Nom et Prénom : BOUKHRIS Mohamed Lyazid

Encadré par : QUAFAFOU Mohamed

Table des matières

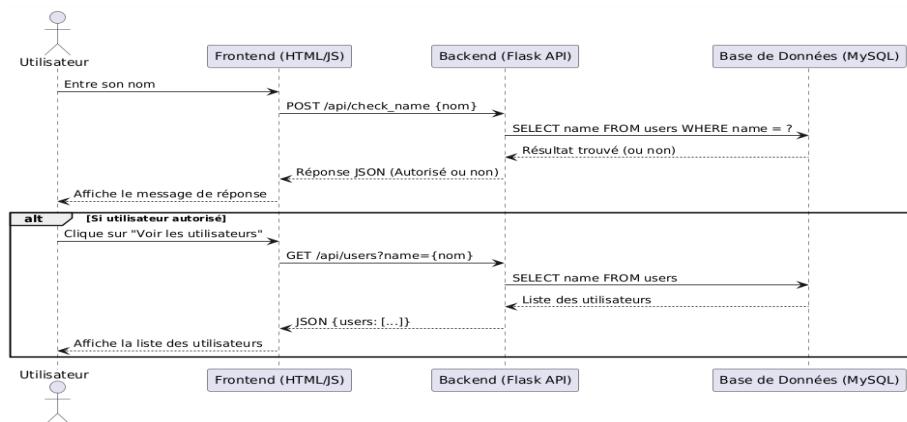
<i>Etapes 1 : Application en local</i>	2
Diagramme de séquence :	2
Installation Effectué :	3
Configuration Effectué :	3
Lancement de l'application :	3
<i>Etapes 2 : Application tourne sur la VM1</i>	4
Installation et configuration effectué :	4
Envoie des fichiers en SSH depuis mon local :	5
<i>Etapes 3 : Application déployer sur 3 VM.....</i>	6
Installation et Configuration de la VM MySQL :	6
Installation et Configuration de la VM Backend :	7
Installation et Configuration de la VM Frontend :	7
Résultat final :	7
<i>Etapes 4 : Tester l'application</i>	8
Instruction pour lancer l'application sur les 3 VM :	8
Instruction pour lancer l'application sur la VM Application python :	9

Etapes 1 : Application en local

J'ai développé une application en local en utilisant Python avec Flask pour le back-end, HTML et JavaScript pour le front-end, et MySQL comme base de données. L'application permet de vérifier si un utilisateur donné existe dans la base de données.

- **Fonctionnalité POST :** L'utilisateur entre son nom, qui est envoyé au serveur via une requête POST. Le serveur vérifie si le nom existe dans la base de données. Si le nom est trouvé, un message "Bonjour + Nom Utilisateur" est affiché, et un bouton apparaît pour permettre à l'utilisateur de consulter les autres utilisateurs autorisés.
- **Fonctionnalité GET :** Après validation, l'utilisateur peut cliquer sur un bouton pour voir la liste des autres utilisateurs qui ont accès. Cela se fait via une requête GET, qui récupère et affiche les noms des utilisateurs autorisés à accéder.
- **Message d'erreur :** Si le nom de l'utilisateur n'existe pas dans la base de données, un message d'erreur "Vous n'avez pas le droit" est affiché.

Diagramme de séquence :



Installation Effectué :

1. Téléchargement de python depuis le site officiel : [Download Python | Python.org](#), puis j'ai lancé son installation.
2. Installation des dépendances nécessaires : pip install flask flask-cors mysql-connector-python
3. Installation et configuration de MySQL : j'ai téléchargé et puis installé MySQL depuis [MySQL :: MySQL Downloads](#)

Configuration Effectué :

1. Création de la base de données et des tables, en se connectant à MySql : mysql -u root -p, (le mot de passe choisi lors de l'installation).
2. Exécution du script SQL : source (suivie du chemin vers le script.sql)
3. Teste de la connexion de MySql dans Python en saisissant l'utilisateur, le mot de passe, la base de données, et le numéro de port

Voici des captures des configurations :

<pre>mysql> use helloSql; Database changed mysql> show tables; +-----+ Tables_in_helloSql +-----+ users +-----+ 1 row in set (0.01 sec) mysql> select * from users; +----+-----+ id name +----+-----+ 1 Achraf 4 Fatima 5 Mehdi 3 Nadia 2 Yassine +----+-----+ 5 rows in set (0.00 sec)</pre>	<pre>from flask import Flask, render_template, request, jsonify from flask_cors import CORS import mysql.connector app = Flask(__name__) CORS(app) 💡 config = { "host": "localhost", "user": "frontend", "password": "frontend", "database": "helloSql", "port": 3306 }</pre>
--	--

Lancement de l'application :

1. Démarrage du serveur Flask, en exécutent la commande suivante : python connexionBD.py
2. Accès à l'application : <http://127.0.0.1:5000>
- 3.

Vérification d'Utilisateur

Achraf

Bonjour Achraf !

Voir les autres utilisateurs

Achraf
Fatima
Mehdi
Nadia
Yassine

Vérification d'Utilisateur

Jean

Vérifier

Vous n'avez pas le droit.

Etapes 2 : Application tourne sur la VM1

Installation et configuration effectué :

1. Installation de virtuel Box.
2. Création d'une nouvelle VM sur VirtualBox, en utilisant l'ISO Ubuntu Server avec cette configuration :

- a) Nom : Ubuntu-Flask
- b) Type : Linux
- c) Version : Ubuntu (64-bit)
- d) Mémoire RAM : 2 Go
- e) Taille du disque : 20 Go

3. Installation de l'interface graphique Ubuntu :

```
sudo apt install ubuntu-desktop -y
sudo apt install gdm3 -y
```

4. Configuration de l'adresse IP Statique : sudo nano /etc/netplan/00-installer-config.yaml

```
GNU nano 4.8                               01-netcfg.yaml
# This file describes the network interfaces available on your system
# For more information, see netplan(5).

network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      dhcp6: no
      addresses:
        - 172.20.10.10/28
      gateway4: 172.20.10.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

Puis : sudo netplan apply

5. Installation de SSH : sudo apt update, et puis : sudo apt install -y openssh-server
6. Configuration du serveur SSH : sudo nano /etc/ssh/sshd_config, et puis j'ai fais ces modifications :


```
PasswordAuthentication yes
AllowUsers utilisateur
PubkeyAuthentication no
```
7. Lancement du serveur SSH :

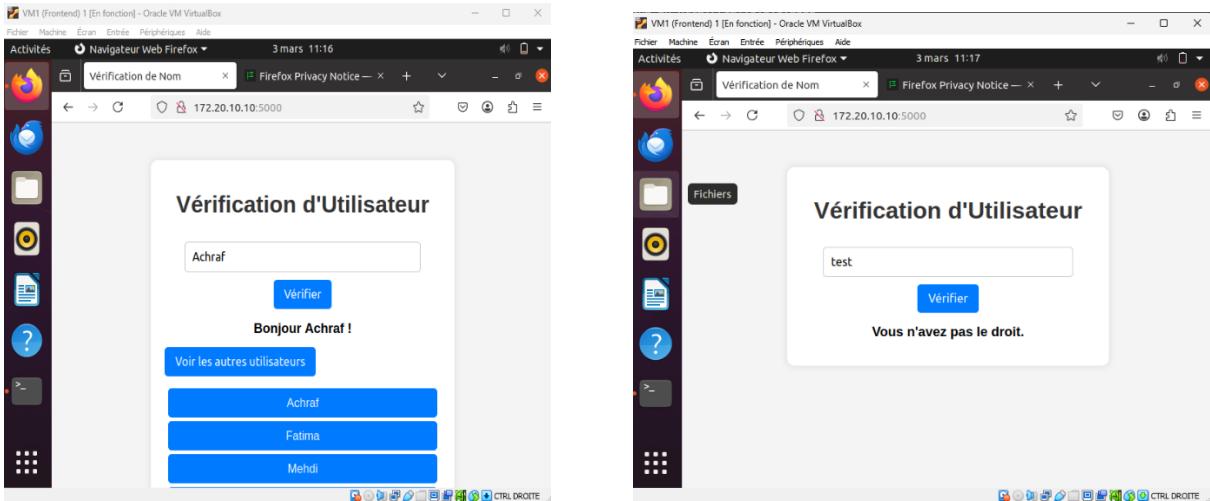

```
sudo systemctl enable ssh
sudo systemctl start ssh
sudo systemctl status ssh
```

Envoie des fichiers en SSH depuis mon local :

J'ai envoyé les fichiers nécessaires pour mon application avec cette commande scp connexion.py frontend@172.20.10.10:/home/frontend/.

Puis j'ai fais les installations comme j'ai fais au niveau de mon local, installation de python, des dépendances de mon app, de MySQL :

1. sudo apt update
2. sudo apt install -y python3 python3-pip : installation de python
3. pip install flask flask-cors mysql-connector-python : installation des dépendances
4. sudo apt install -y mysql-server : installation de MySQL
5. mysql -u root -p < script.sql : création de la base de données et les tables
6. Ajout d'un nouvel utilisateur avec tous les permissions (voir capture), pour se connecter à ma base, avec la modification des informations de la connexion à la BD dans le fichier connexion.py
7. Python3 connexion.py : lancement de mon application
8. <http://172.20.10.10:5000> : puis je teste mon application sur mon navigateur



Etapes 3 : Application déployer sur 3 VM

J'ai créé trois machines virtuelles à partir de l'ISO utilisée à l'étape 2. Chaque machine est configurée pour un rôle spécifique : une pour la base de données, une pour le backend et une pour le frontend. J'ai configuré ces machines en mode accès par pont, désactivé le service DHCP, et attribué des adresses IP statiques, comme à l'étape 2. Chaque machine possède ainsi sa propre adresse IP :

Backend : 172.20.10.12

Frontend : 172.20.10.11

MySQL : 172.20.10.8

Puis j'ai fait les installations nécessaires sur chaque machine :

Installation et Configuration de la VM MySQL :

1. Installation de MySQL : `sudo apt install mysql-server -y`
2. Configuration de MySQL pour autoriser les connexions distantes : `sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf`
Puis j'ai modifié cette ligne : `bind-address = 0.0.0.0`
Puis je redémarre MySQL : `sudo systemctl restart mysql`
3. Création de la base avec le script que j'ai envoyé en ssh comme j'ai fait dans l'étapes 2
`scp script.sql bdd@172.20.10.2:/home/bdd/`, après avoir installé et activer le service ssh sur la VM MySQL.
4. Création d'un utilisateur pour le backend :
`CREATE USER 'backend'@'%' IDENTIFIED BY 'backend';`
`GRANT ALL PRIVILEGES ON helloSql.* TO 'backend'@'%';`

FLUSH PRIVILEGES ;

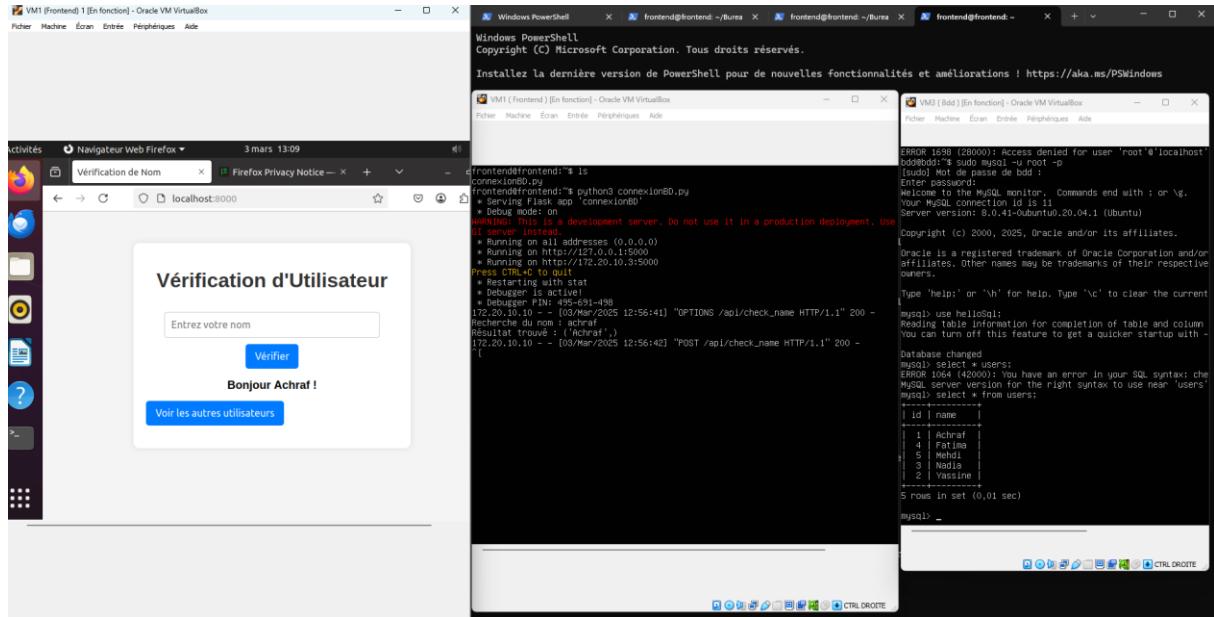
Installation et Configuration de la VM Backend :

1. Installation de python : sudo apt install -y python3 python3-pip
2. pip install flask flask-cors mysql-connector-python : installation des dépendances
3. Je teste la connexion vers la VM MySQL : mysql -u backend -p -h 172.20.10.2 -P 3306
Et ça fonctionne je peux accéder à ma base de données.
4. J'ai envoyé en SSH mon backend python (connexionBD.py) et puis je le lance, mon application fonctionne bien.
5. J'ai fait un test avec curl pour voir si ça fonctionne

Installation et Configuration de la VM Frontend :

1. D'abord envoie de index.html depuis mon local vers la VM en SSH : scp index.html bdd@172.20.10.10:/home/frontend/
2. Modification de l'adresse IP au niveau du fetch :
Je fais maintenant un fetch vers l'adresse IP de la VM de mon backend :
`fetch(http://172.20.10.3:5000/api/check_name)`
3. Je lance un serveur dans le même répertoire que celui où se trouve le fichier index.html : python3 -m http.server 8000 --bind 0.0.0.0
4. J'ouvre le port 8000 dans le pare feu :
`sudo ufw allow 8000/tcp` `sudo ufw reload`
5. Puis je teste au niveau de mon navigateur : <http://172.20.10.10:8000>

Résultat final :



Etapes 4 : Tester l'application

Instruction pour lancer l'application sur les 3 VM :

1. D'abord se mettre dans un réseau privé pour mon cas j'utilise le partage de connexion de mon téléphone.
2. Lancé les 3 VM (Bdd, Backend, Frontend)
3. Se connecter le mot de passe sur les trois c'est : frontend
4. Ouvrir un terminal sur les trois VM et se mettre dans le répertoire Bureau : cd Bureau
5. Sur la VM backend on teste d'abord si on ping la VM BDD pour voir s'il y'a une communication, puis je peux lancer mon application voici les commandes :
 - a. ping 172.20.10.8
 - b. python3 connexion.py
6. Pour la VM BDD rien à faire tous est configuré, il faut juste lancé la VM.
7. Pour la VM Frontend, il faut se mettre d'abord dans Bureau, car c'est le répertoire où se trouve mon fichier index.html, puis il faut lancer un serveur python, voici les commandes :
 - a. cd Bureau
 - b. ls (vérifier si le fichier index.html existe)
 - c. python -m http.server 8000 (lancer le serveur)
 - d. puis sur le navigateur on teste en tapant : <http://localhost:8000/>

Instruction pour lancer l'application sur la VM Application python :

1. Lancer la VM ApplicationPython
2. Se connecter le mot de passe est le même pour les autres : frontend
3. Ouvrir un terminal et se mettre dans le répertoire Bureau
4. Vérifier si 3 fichiers existent : connexionBD.py, script.sql, index.html
5. Lancer l'application : python3 connexionBD.py
6. Ouvrir un navigateur et taper : <http://localhost:5000/>, et l'application est lancé et on peut l'utiliser.

NB : Pour le test vous trouverez un capture tout en haut dans la section où je présente la configuration faite sur mon local, qui contient les noms des personnes qui sont autorisées à entrer, pour pouvoir tester un bon nom.