

RAPPORT DE CONFIGURATION ET DE GESTION DES DEUX MACHINES VIRTUELLES

..SOMMAIRE

I-INTRODUCTION

II-INFRASTRUCTURES

III-CONFIGURATION MACHINES VIRTUELLES

III-1.Système d'exploitation

III-2.Ressources allouées

III-3 .Réseau

III-4 .Logiciels installées(sur les deux serveurs)

IV-SECURITE

V-GESTION DES ACCES

VI-CONCLUSION

I-INTRODUCTION

Le projet consistait dans un premier temps en la création deux VM (virtual machine) l'une renfermant le serveur Web pour héberger notre API et une seconde VM renfermant le serveur de notre base de données (MYSQL). Dans un second temps le but était de permettre la communication entre les deux serveurs afin de permettre l'accès et les différentes requêtes sur la base de données par l'API.

II-INFRASTRUCTURES

. **VMWARE** : les VMS ont été créées avec l'hyperviseur VMware-Workstation-pro(version 17 .0.0)

III-CONFIGURATION DES MACHINES VIRTUELLES

III-1. **Système d'exploitation utilisé** : Ubuntu Server(ubuntu-22.04.2-live-server-amd64)

III-2. Ressources allouées : Pour les deux VMS la configuration a été la suivante :

RAM : 4GB

Espace Disque : 20GB

CPU : 2 cores

III -3.**Réseau** :

Les Deux Machines virtuelles sont configurées en Mode 'Network Adapter(NAT),

Avec chacune une adresse IP statique :

Serveur API (192.168.184.132)

Serveur Base de Donnée (192.168.184.133)

III-4. Logiciels installés :

***POUR LE SERVEUR API**

-**NGINX** (serveur qui va prendre en charge la configuration réseau mise en ligne local de l'api) comme suit :

sudo apt update

sudo apt install nginx

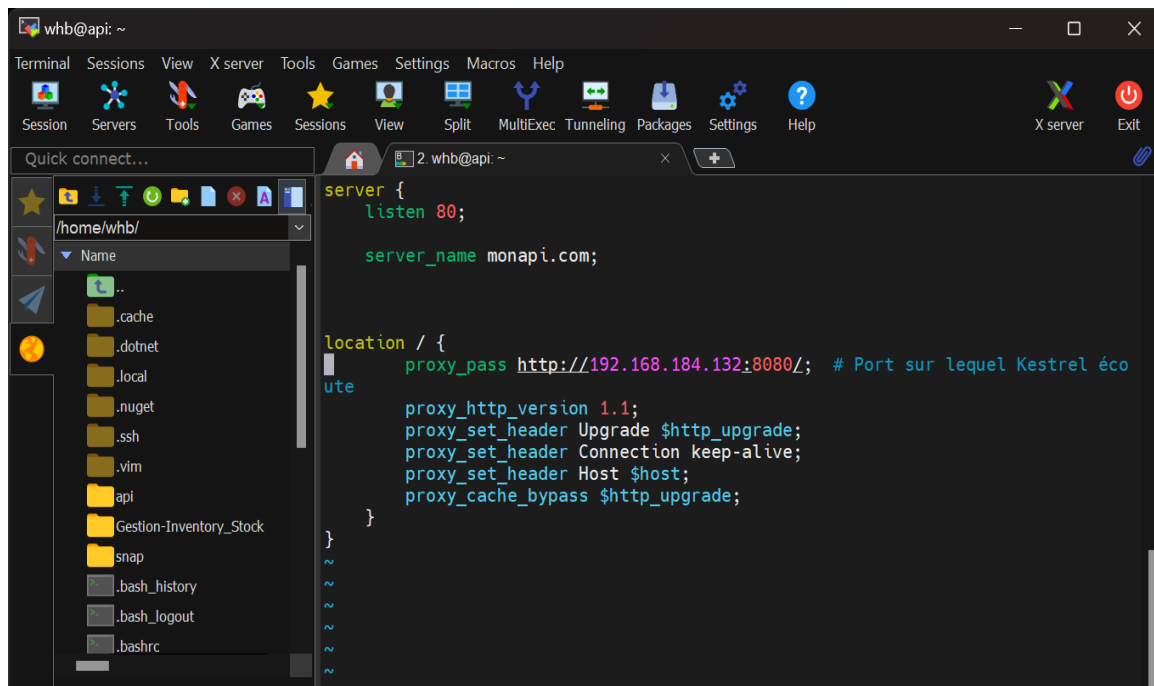
Sudo service nginx start

Sudo service nginx status //voir si le service est bien en cours d'exécution.

Ensuite pour la configuration de notre site

on accède au répertoire **/etc/nginx/sites-available/**(à créer si il n'existe pas)

on crée ensuite un fichier **mo,-api** dans ce répertoire qu'on configure comme suit :



Par la suite on crée un lien symbolique vers le répertoire 'site-enabled' 'pour activer le site' avec la commande : **sudo ln -s /etc/nginx/sites-available/mon-api /etc/nginx/sites-enabled/**

On teste ensuite la configuration de nginx voir d 'éventuelles erreurs :

Sudo nginx -t

Puis on redémarre le service nginx : **sudo systemctl restart nginx**

-DONET :

wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb

sudo dpkg -i packages-microsoft-prod.deb

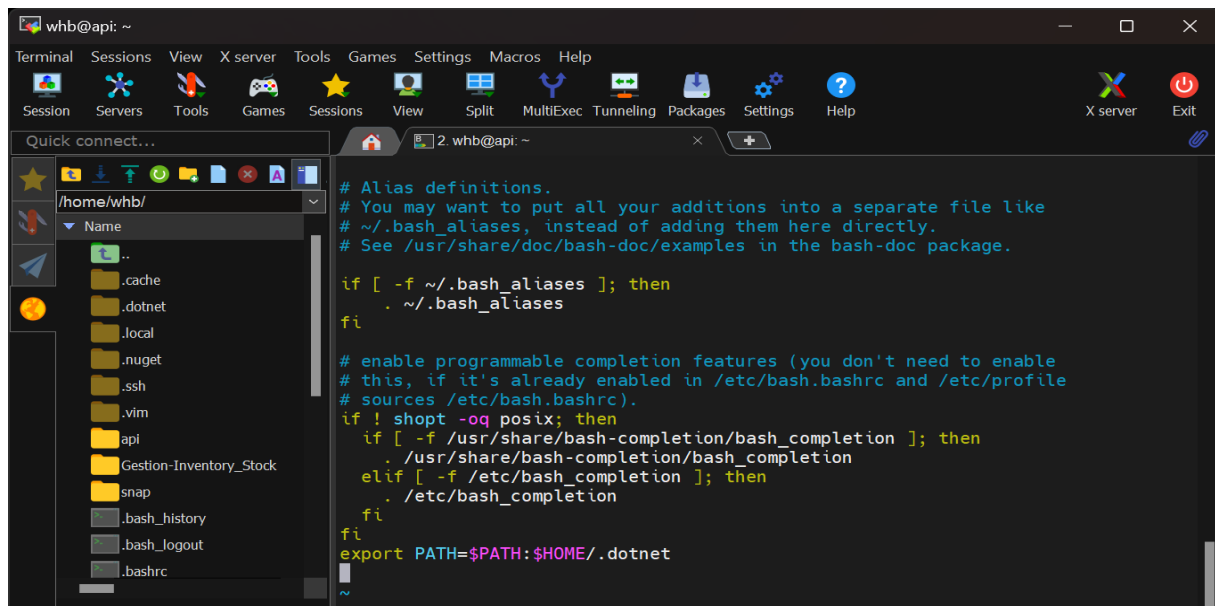
wget https://dot.net/v1/dotnet-install.sh

chmod +x dotnet-install.sh : droit pour pouvoir exécuter le script d'installation

./dotnet-install.sh

Sudo vi ~/.bashrc

export PATH=\$PATH:\$HOME/.dotnet :// ajouter ceci à la fin du script comme ceci :



source ~/.bashrc

dotnet --version : // affiche la version de dotnet installée

-Copie du projet à partir de la machine physique(pc) sur la VM dans le cmd:

C:\Windows\System32>pscp -r C:\Users\HP\Desktop\Gestion-Inventory_Stock.zip
whb@192.168.184.132:/home/whb

On unzip ensuite avec la commande :

Unzip Gestion-Inventory_Stock.zip

Après avoir accéder au répertoire : **/Gestion-Inventory_Stock/InventoryManagement.Api\$**

Mettre les dépendances nécessaires du projet dans un dossier /api avec la commande :

dotnet publish -c Release -o /home/whb/api/

Lancer le projet avec la commande :

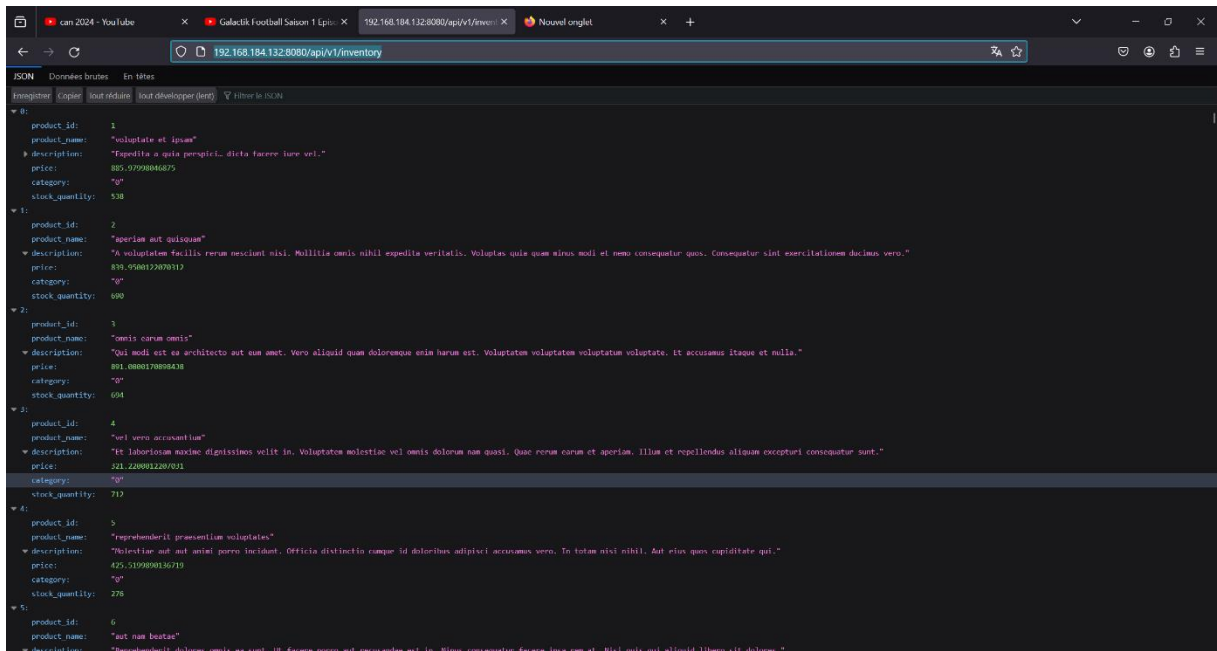
dotnet ./api/InventoryManagement.Api.dll

aller ensuite dans un navigateur et entrée la :

<http://192.168.184.132:8080/api/v1/inventory>

on voit alors le contenu de notre API :

avec FIREFOX :



AVEC CHROME :



*POUR LE SERVEUR de la DB

-MYSQL installation comme suit :

sudo apt update

sudo apt install mysql-server//intall mysql

sudo mysql_secure_installation // securise l'installation

après l'avoir placer sur la VM (sur MOBAXTERM juste faire glisser le fichier) :

on copie le fichier de la db ainsi transférer sur la vm vers le fichier contenant les db sur mysql par défaut **/var/lib/mysql** :

```
sudo cp e-commerce.sql /var/lib/mysql
```

sudo chown -R mysql:mysql /var/lib/e-commerce.sql //permettre au serveur mysql de pouvoir lire le fichier contenant la database.

Ensuite on crée une base de donnée vide et une utilisateur ayant tout les sur cette dernière afin de pouvoir utiliser notre fichier sql la db qu’il contient :

```
sudo mysql -u root -p
```

```
mysql>CREATE DATABASE ‘your_database’
```

```
mysql> CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';
```

```
mysql> GRANT ALL PRIVILEGES ON your_database.* TO 'your_user'@'%';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql>USE your_database //pour utiliser la db créée;
```

mysql>SOURCE /chemin/vers//ecommerce.sql;// cela Cela exécute toutes les commandes SQL du fichier, créant ainsi la structure de la base de données et insérant les données.

On vérifie ensuite comme suit :

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| your_database |
+-----+
```

```
mysql> use your_database
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql> show tables;
+-----+
| Tables_in_your_database |
+-----+
| address                  |
| cart                     |
| category                 |
| command                  |
| contactus                |
| currency                 |
| inventory                 |
| invoices                 |
| language                 |
| notification              |
| paiement                 |
| photo                    |
| product                  |
| promotion                |
| rate                     |
| retour                   |
| review                   |
| shipping                 |
| user                     |
| wishlist                 |
+-----+
```

On voit alors bien nos différentes tables.

.Configuration pour permettre l'accès de la **Database par le serveur API** :

On entre dans le répertoire : /etc/mysql/mysql.conf.d ,puis on ouvre le fichier

mysqld.cnf : sudo vi mysqld.cnf

on le configure de cette manière :

```

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
port                = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.
#html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 0.0.0.0
mysqlx-bind-address = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack        = 256K

# thread_cache_size   = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP

```

Ensuite on redémarre le service mysql : **sudo systemctl restart mysql** , puis on test les pings entres les deux serveurs :

```

whb@db:/etc/mysql/mysql.conf.d$ ping 192.168.184.132
PING 192.168.184.132 (192.168.184.132) 56(84) bytes of data.
64 bytes from 192.168.184.132: icmp_seq=1 ttl=64 time=0.977 ms
64 bytes from 192.168.184.132: icmp_seq=2 ttl=64 time=0.391 ms
64 bytes from 192.168.184.132: icmp_seq=3 ttl=64 time=0.406 ms

--- 192.168.184.132 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2013ms
rtt min/avg/max/mdev = 0.391/0.591/0.977/0.272 ms

```



```

whb@api:~$ ping 192.168.184.133
PING 192.168.184.133 (192.168.184.133) 56(84) bytes of data.
64 bytes from 192.168.184.133: icmp_seq=1 ttl=64 time=0.419 ms
64 bytes from 192.168.184.133: icmp_seq=2 ttl=64 time=0.358 ms
64 bytes from 192.168.184.133: icmp_seq=3 ttl=64 time=0.357 ms

--- 192.168.184.133 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.357/0.378/0.419/0.029 ms

```

On test ensuite la connexion à la **database** depuis le serveur de **l'API** :

```

whb@api:~$ sudo mysql -u your_user -h 192.168.184.133 -p
[sudo] password for whb:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| your_database |
+-----+
3 rows in set (0,01 sec)

```

On voir bien notre database '**your_database**'

Nos deux pourront donc bien communiquer et l'Api fairr toute type requête .

IV-SECURITE

-sudo ufw allow 3306: pour le serveur «database » permet ainsi le trafic des données

-sudo ufw allow 8080 : pour le serveur API qui sert sur le 8080 (permission de servir)

Sudo ufw allow 22: pour assurer la connexion en **ssh** des deux serveurs

-sudo ufw enable : pour active le service **ufw**

V-GESTION ACCES :

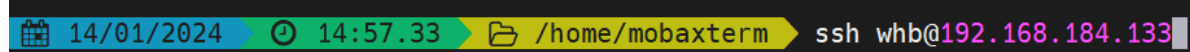
L'accès aux VM se fait en local en se connectant directement aux interfaces de ces dernières ou en connexion su MobaXTERM ssh comme suit :

Serveur API :

A terminal window from MobaXterm showing the command 'ssh whb@192.168.184.132'. The window header includes a calendar icon, the date '14/01/2024', a clock icon, the time '14:57.33', a folder icon, and the path '/home/mobaxterm'.

```
14/01/2024 14:57.33 /home/mobaxterm ssh whb@192.168.184.132
```

Serveur Database :

A terminal window from MobaXterm showing the command 'ssh whb@192.168.184.133'. The window header includes a calendar icon, the date '14/01/2024', a clock icon, the time '14:57.33', a folder icon, and the path '/home/mobaxterm'.

```
14/01/2024 14:57.33 /home/mobaxterm ssh whb@192.168.184.133
```

VI-CONCLUSION

En résumé la configuration des machines c'est fait globalement suivant deux

Axes :

- la configuration du logiciel nginx pour la connectivité en http et la connexion ,de ce serveur à la database

- l'hébergement de la base de données local sur le serveur « database », suivie ,

De la configuration des fichiers « sudo vi **/etc/mysql/mysql.conf.d** » permettant la connexion à distance depuis

Le serveur API à cette base de données.