

# Mettre à disposition un service informatique

## Déployer un service

### Introduction

À la fin de l'année, je vais devoir déployer le frontend de mon projet (interface utilisateur en React) ainsi que le backend (API node.js). Le backend sera déployé sur une infrastructure Linux (par exemple, sur les machines utilisées par les étudiants de deuxième année). Cependant, imaginons que lors de l'examen, je n'aie pas accès à leur infrastructure (que ce soit en raison d'une absence de connexion à internet ou d'un problème technique tel qu'un serveur hors service). Dans ce cas, il est essentiel de trouver des alternatives. C'est pourquoi, lors de l'examen, je dois prévoir des solutions de secours :

- Avoir une copie du backend sur mon propre ordinateur avec le frontend (notamment pour l'épreuve E5).
- Mettre en place le backend sur une machine virtuelle Linux sur mon ordinateur et fournir une documentation expliquant comment j'ai déployé le service sur une infrastructure Linux.

### Objectifs pédagogiques

- Mettre votre backend NodeJS sur une machine virtuelle Linux
- Réaliser les documentations pour l'installation de la machine virtuelle et l'installation d'Ubuntu

Outils nécessaires

- Proxmox
- Visual Studio Code

## Mise en place serveur NodeJS et bdd MySQL

1. Mettre à jour Ubuntu  
`apt update`
2. Installer Git, NodeJS, npm et mariadb-server  
`apt install git nodejs npm mariadb-server`
3. Configurer votre mariadb-server avec la commande 'mysql\_secure\_installation'  
`mysql_secure_installation`

4. Avec Git, récupérer votre backend se trouvant sur GitLab ou GitHub et mettez le dans le dossier /opt.

```
root@G8-back:~# cd /opt
root@G8-back:/opt# git clone https://github.com/51237/M2L.git
Cloning into 'M2L'...
```

5. Déployer votre BDD sur le serveur mysql d'ubuntu, puis créer un utilisateur qui aura accès à votre Base de données.

```
root@G8-back:/opt# mysql -u root -p
MariaDB [(none)]> CREATE DATABASE mligue
-> ;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use mligue;
Database changed
root@G8-back:/opt/M2L# mysql -u root -p mligue < /opt/M2L/mligue.sql
Enter password:
```

On peut voir que les tables sont bien présentes :

```
Database changed
MariaDB [mligue]> show tables;
+-----+
| Tables_in_mligue |
+-----+
| panier            |
| products          |
| users             |
+-----+
3 rows in set (0.001 sec)
```

```
GRANT ALL PRIVILEGES ON votreDatabase.* TO user@localhost identified by
«votreMotDePasse»;
FLUSH PRIVILEGES;
```

6. Créer le fichier .env dans le dossier de votre projet et donner la configuration nécessaire.

```
sudo nano /opt/votreServeurNode/.env
```

```
GNU nano 7.2 /opt/M2L/back/.env *
DATABASE =mligue
DATABASE_HOST =localhost
DATABASE_USER =root
DATABASE_PASSWORD =Test123
API_KEY = mdpapimdp
```

7. Essayer de lancer votre serveur et d'y accéder via votre ordinateur (en tapant quelque chose du genre : <http://adresseIpVm:port/path>)  
Pour voir l'adresse IP de la machine : (ip a). De + il faut s'être mis en connexion par pont sur la config de la VM. Pour lancer votre serveur : npm start à l'emplacement de votre server.js

8. Maintenant, il faut mettre en place le fichier permettant de lancer le serveur au démarrage. Créer un fichier serveurnode.service dans le dossier etc/systemd/system/

```
sudo nano /etc/systemd/system/serveurnode.service
```

9. Dans ce fichier, modifier la config suivante pour qu'elle soit conforme pour votre machine :

```
GNU nano 7.2 /etc/systemd/system/serveurnode.service
[Unit]
Description= back Node.js AP3
After=network.target

[Service]
User=root
WorkingDirectory=/opt/M2L/back
ExecStart=/usr/bin/node /opt/M2L/back/server.js
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

10. Si le fichier est bien mis en place, actualisez les unités systemd puis activez le service que vous venez de créer.

```
root@G8-back:/opt/M2L/back# systemctl daemon-reload
root@G8-back:/opt/M2L/back# systemctl enable serveurnode.service
root@G8-back:/opt/M2L/back# systemctl start serveurnode.service
```

11. Démarrez le service puis redémarrer votre vm, si tout est bien réalisez, votre api se lancera au démarrage.

# Mise en place du Front en ReactJS

1. Mettre à jour Ubuntu

`apt update`

2. Installer Git, NodeJS, npm

`apt install git nodejs npm`

3. Avec Git, récupérer votre backend se trouvant sur GitLab ou GitHub et mettez le dans le dossier /opt.

```
root@G8-back:~# cd /opt
root@G8-back:/opt# git clone https://github.com/51237/M2L.git
Cloning into 'M2L'...
```

4. Essayer de lancer votre serveur et d'y accéder via votre ordinateur (en tapant quelque chose du genre : <http://adresseIpVm:port/path>)

Pour voir l'adresse IP de la machine : (ip a). De + il faut s'être mis en connexion par pont sur la config de la VM. Pour lancer votre serveur : `npm start` à l'emplacement de votre `server.js`

5. Pour que le front et le back soit liés il faut que créer une branche serveur à notre projet et remplacer dans tous les liens, le localhost par l'ip de notre serveur
6. Démarrez le service puis redémarrer votre vm, si tout est bien réalisé, votre api se lancera au démarrage.