Rapport de Virtualisation et Conteneurisation

Prénom et Nom: Felipe Mateus Falcão Barreto

Le projet suivant vise à développer une application web en trois étapes :

- Développement
- Virtualisation
- Contaneurisation

1 - Application

L'application full-stack a été développée à l'aide des outils suivants :

- Front-end : HTML et JavaScript
- Back-end : Node.js, Express.js (création du serveur web), PG (driver du PostgreSQL pour Node.js) et CORS (accès API depuis différentes origines)
- Base de données : PostgreSQL
- Gestionnaire de paquets : NPM

La principale fonctionnalité de cette première étape était de tester les requêtes web entre le front-end et le back-end afin de gérer le flux de données lors des étapes suivantes.

2 - Virtualisation

Cette étape était la plus complexe, car elle a nécessité la création de 4 machines virtuelles :

- Frontend
- Backend
- Base de données
- Tests

La machine virtuelle utilisée était Ubuntu, en raison de la flexibilité offerte par les systèmes Linux. Pour permettre la communication entre les machines virtuelles, les éléments suivants ont été utilisés :

- NAT : facilite l'accès des machines virtuelles à Internet
- Host-Only: permet la communication entre les machines virtuelles et la machine host
- Apache2 : permet aux machines du même hôte d'accéder à un serveur web créé

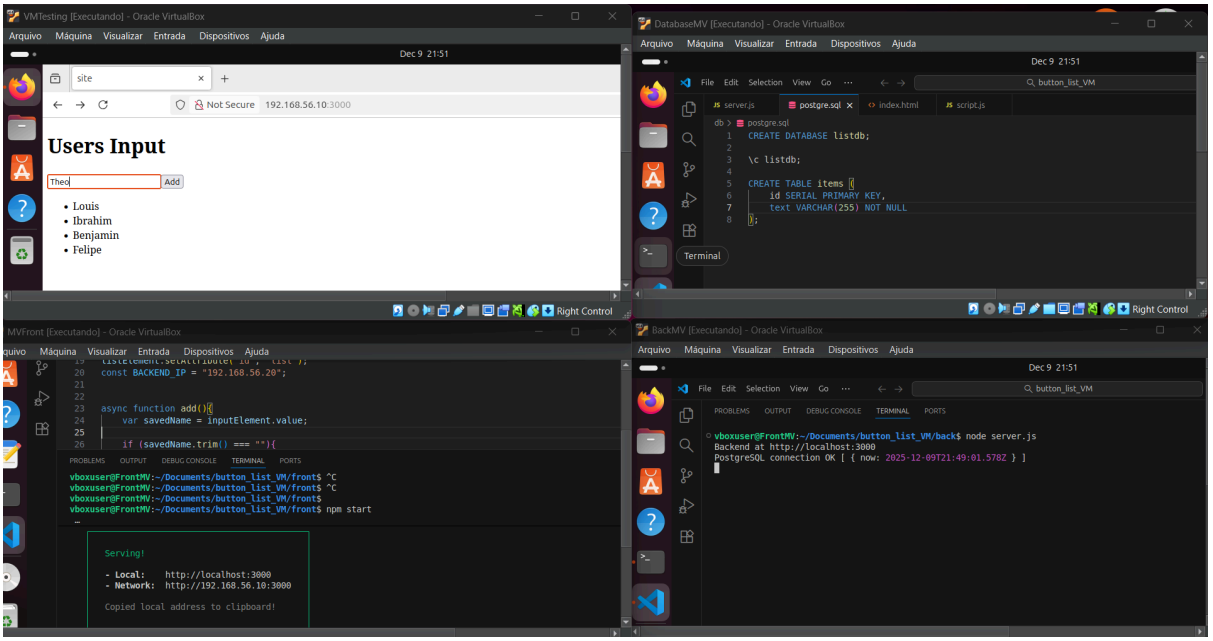
Pour la configuration, le fichier "00-installer-config.yaml" a été utilisé:

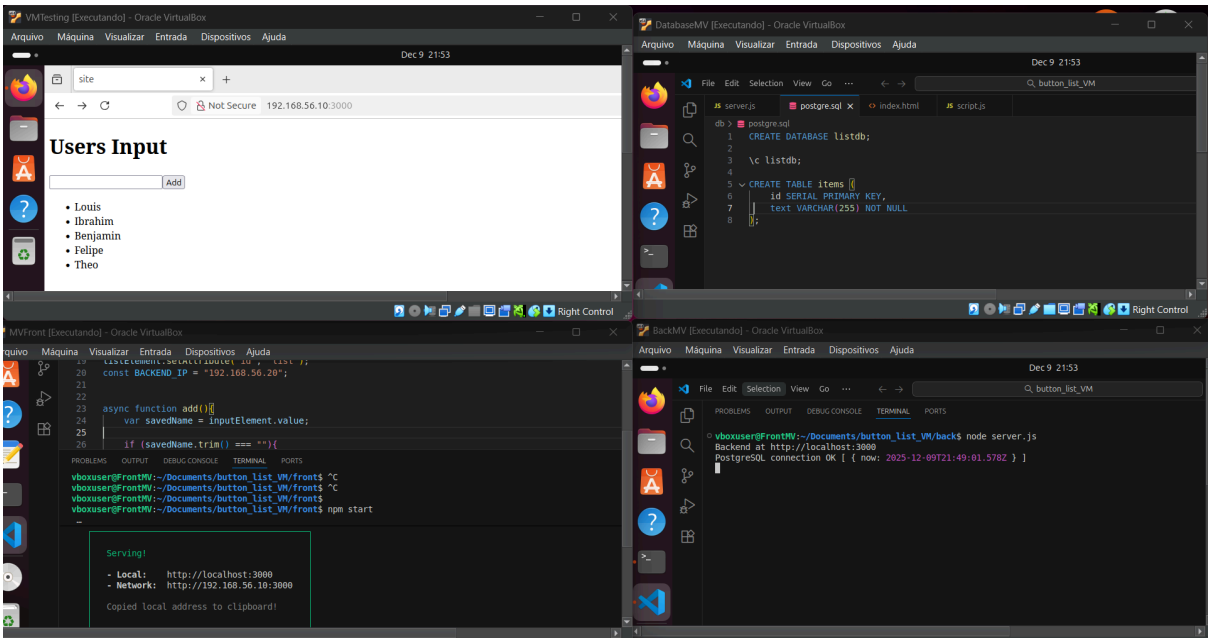
```
network:
        version: 2
        ethernets:
        enp0s3: # NAT (à ne pas mettre pour la VM Data)
                dhcp4: true
        enp0s8: # Host-Only
        dhcp4: false
```

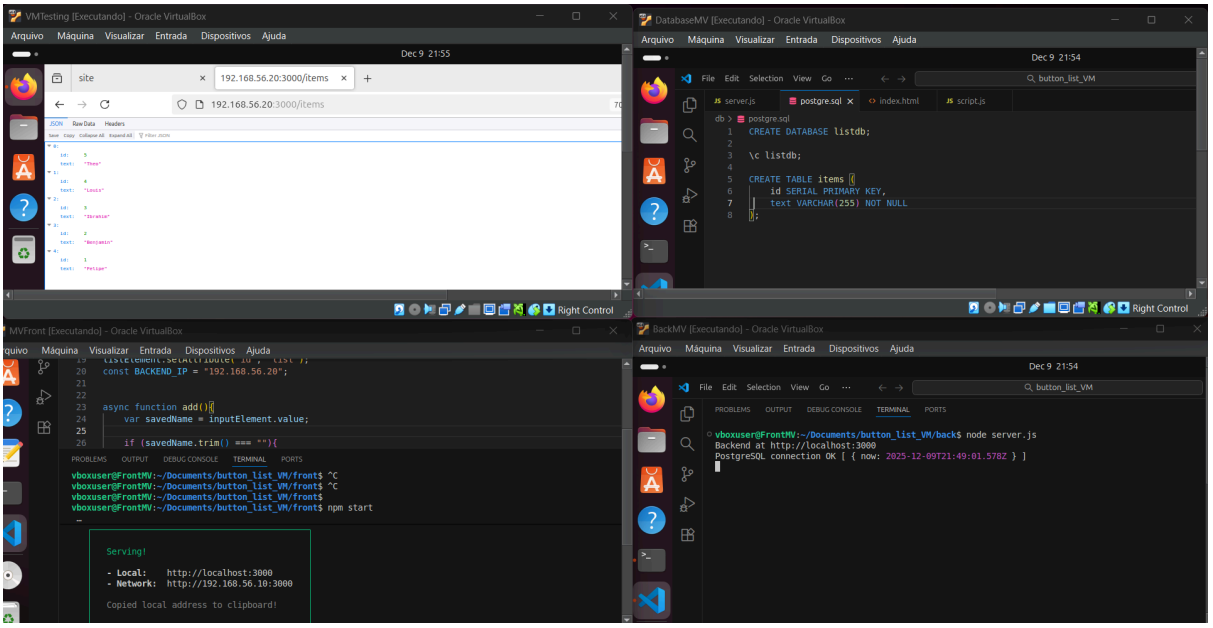
addresses: [192.168.56.X/24] # X égal à {10; 20; 30} selon la VM
nameservers:
addresses: [8.8.8.8, 8.8.4.4] # Serveurs Google

Problème rencontré:
- Certaines machines, malgré l'installation du fichier "00-installer-config.yaml", n'ont pas modifié leur adresse IP, ce qui a perturbé la communication entre elles.

Leçon apprise:
- La virtualisation renforce la sécurité du projet, car un problème isolé n'affecte pas l'ensemble du développement.





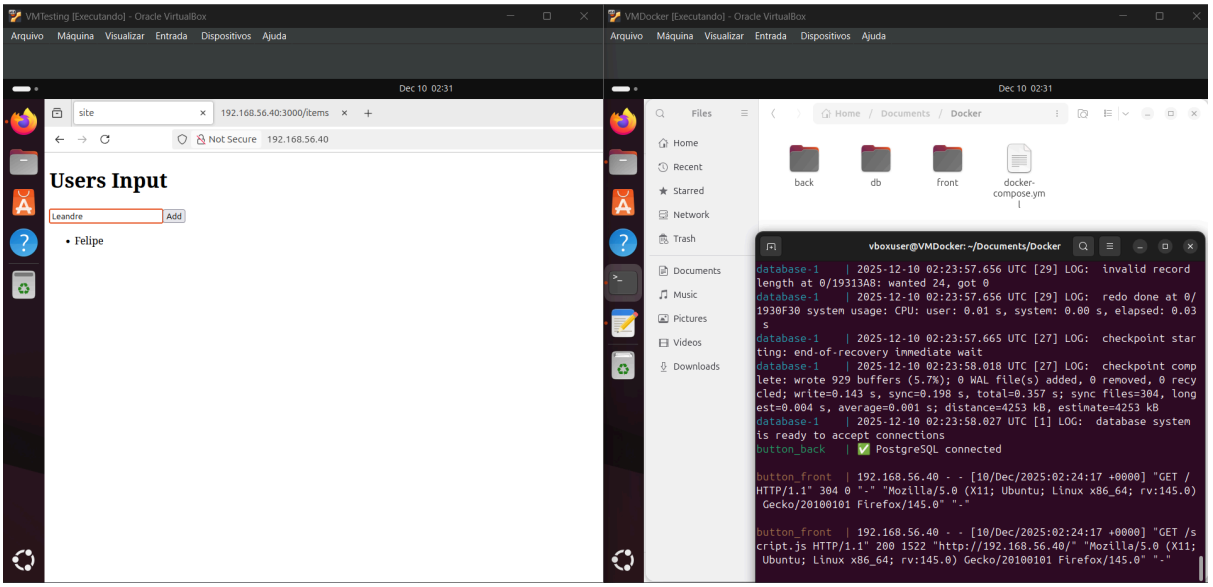


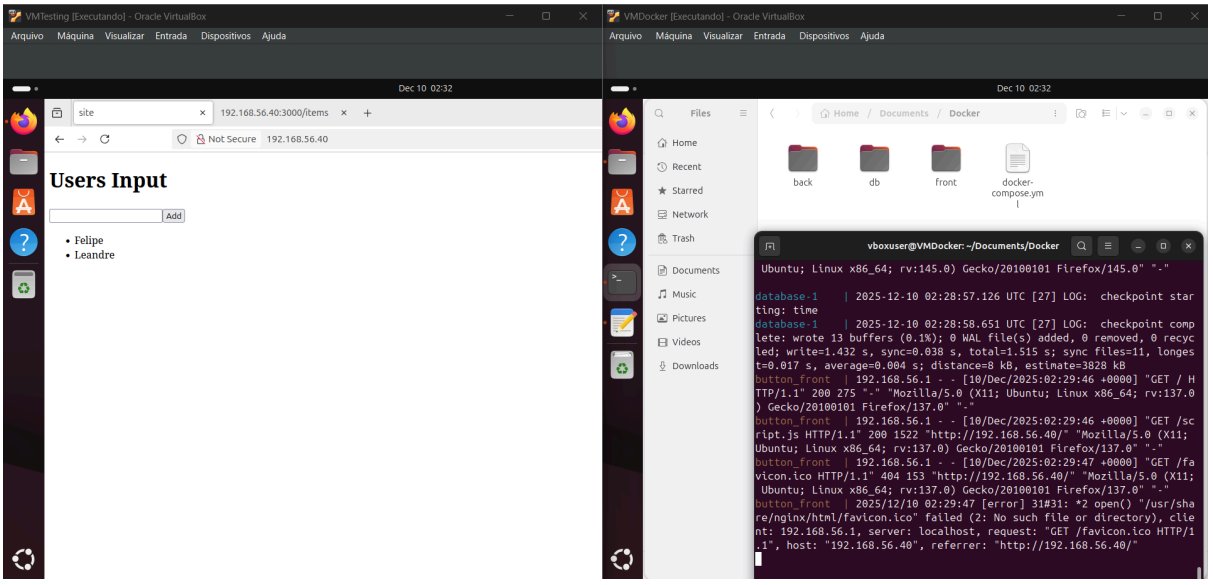
3 - Conteneurisation
Après le développement de la virtualisation du site web, le défi consistait à assembler un système centralisé capable d'exécuter l'ensemble du site de manière sécurisée. Pour ce faire, l'utilisation de Docker a nécessité la conteneurisation des composants du site.

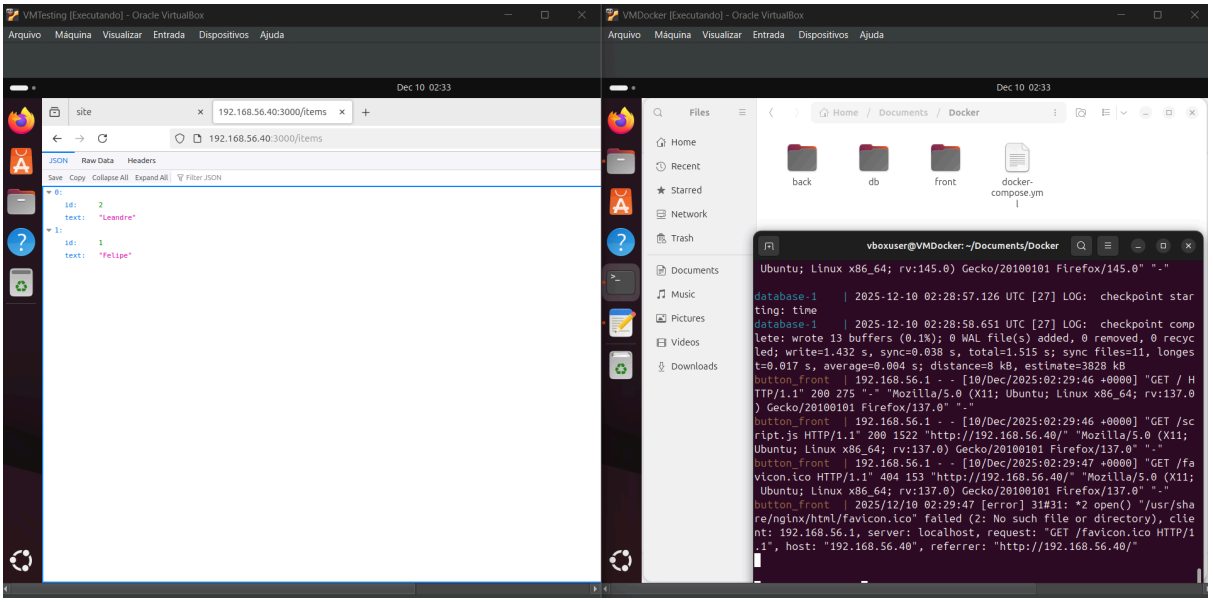
Outils utilisés:
- Docker (packaging d'applications, plus léger qu'une machine virtuelle)
- Docker Compose (gestion des conteneurs)
- Dockerfile (fichier décrivant la création d'une image Docker)

Défi:
- Restructuration générale du projet suite à l'utilisation de Docker, impliquant des Dockerfiles et des modifications du code pour un fonctionnement optimal.

Leçons apprises:
- Le Docker optimise cette phase de développement sécurisé, car elle centralise tout dans une seule application sans créer d'environnement à risque pour le logiciel.


VMTesting [Executando] - Oracle VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Dec 10 02:31
site
192.168.56.40:3000/items
Not Secure 192.168.56.40
Users Input
Leandre
Add
• Felipe
VMDocker [Executando] - Oracle VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Dec 10 02:31
Files
Home
Recent
Starred
Network
Trash
Documents
Music
Pictures
Videos
Downloads
Home / Documents / Docker
back
db
front
docker-compose.yml
vboxuser@VMDocker: ~/Documents/Docker
database-1 | 2025-12-10 02:23:57.656 UTC [29] LOG: invalid record length at 0/19313A8: wanted 24, got 0
database-1 | 2025-12-10 02:23:57.656 UTC [29] LOG: redo done at 0/1930F30 system usage: CPU: user: 0.01 s, system: 0.00 s, elapsed: 0.03 s
database-1 | 2025-12-10 02:23:57.665 UTC [27] LOG: checkpoint starting: end-of-recovery immediate wait
database-1 | 2025-12-10 02:23:58.018 UTC [27] LOG: checkpoint complete: wrote 929 buffers (5.7%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.143 s, sync=0.198 s, total=0.357 s; sync files=304, longest=0.004 s, average=0.001 s; distance=4253 kB, estimate=4253 kB
database-1 | 2025-12-10 02:23:58.027 UTC [1] LOG: database system is ready to accept connections
button_back | ✅ PostgreSQL connected
button_front | 192.168.56.40 - - [10/Dec/2025:02:24:17 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0" "-"
button_front | 192.168.56.40 - - [10/Dec/2025:02:24:17 +0000] "GET /script.js HTTP/1.1" 200 1522 "http://192.168.56.40/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0" "-"


VMTesting [Executando] - Oracle VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Dec 10 02:32
site
192.168.56.40:3000/items
Not Secure 192.168.56.40
Users Input
Add
• Felipe
• Leandre
VMDocker [Executando] - Oracle VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Dec 10 02:32
Files
Home
Recent
Starred
Network
Trash
Documents
Music
Pictures
Videos
Downloads
Home / Documents / Docker
back
db
front
docker-compose.yml
vboxuser@VMDocker: ~/Documents/Docker
Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0" "-"
database-1 | 2025-12-10 02:28:57.126 UTC [27] LOG: checkpoint starting: time
database-1 | 2025-12-10 02:28:58.651 UTC [27] LOG: checkpoint complete: wrote 13 buffers (0.1%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.432 s, sync=0.038 s, total=1.515 s; sync files=11, longest=0.017 s, average=0.004 s; distance=8 kB, estimate=3828 kB
button_front | 192.168.56.1 - - [10/Dec/2025:02:29:46 +0000] "GET / HTTP/1.1" 200 275 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0" "-"
button_front | 192.168.56.1 - - [10/Dec/2025:02:29:46 +0000] "GET /script.js HTTP/1.1" 200 1522 "http://192.168.56.40/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0" "-"
button_front | 192.168.56.1 - - [10/Dec/2025:02:29:47 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://192.168.56.40/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0" "-"
button_front | 2025/12/10 02:29:47 [error] 31#31: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 192.168.56.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "192.168.56.40", referrer: "http://192.168.56.40/"


VMTesting [Executando] - Oracle VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Dec 10 02:33
site
192.168.56.40:3000/items
192.168.56.40:3000/items
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
0:
id: 2
text: "Leandre"
1:
id: 1
text: "Felipe"
VMDocker [Executando] - Oracle VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Dec 10 02:33
Files
Home
Recent
Starred
Network
Trash
Documents
Music
Pictures
Videos
Downloads
Home / Documents / Docker
back
db
front
docker-compose.yml
vboxuser@VMDocker: ~/Documents/Docker
Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0" "-"
database-1 | 2025-12-10 02:28:57.126 UTC [27] LOG: checkpoint starting: time
database-1 | 2025-12-10 02:28:58.651 UTC [27] LOG: checkpoint complete: wrote 13 buffers (0.1%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.432 s, sync=0.038 s, total=1.515 s; sync files=11, longest=0.017 s, average=0.004 s; distance=8 kB, estimate=3828 kB
button_front | 192.168.56.1 - - [10/Dec/2025:02:29:46 +0000] "GET / HTTP/1.1" 200 275 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0" "-"
button_front | 192.168.56.1 - - [10/Dec/2025:02:29:46 +0000] "GET /script.js HTTP/1.1" 200 1522 "http://192.168.56.40/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0" "-"
button_front | 192.168.56.1 - - [10/Dec/2025:02:29:47 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://192.168.56.40/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0" "-"
button_front | 2025/12/10 02:29:47 [error] 31#31: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 192.168.56.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "192.168.56.40", referrer: "http://192.168.56.40/"

Ce qui manquait:
- Tests de charge

Conclusion :
L'apprentissage des techniques de virtualisation et de conteneurisation a enrichi ma formation. J’ai pu comprendre plus en détail comment ces technologies sont appliquées pour créer des environnements sécurisés de développement logiciel. La compréhension des impacts et des différences entre chaque approche fait de moi un ingénieur plus averti et mieux préparé, capable d’adopter des méthodes efficaces de protection et de stabilité pour les applications que je développerai à l’avenir.