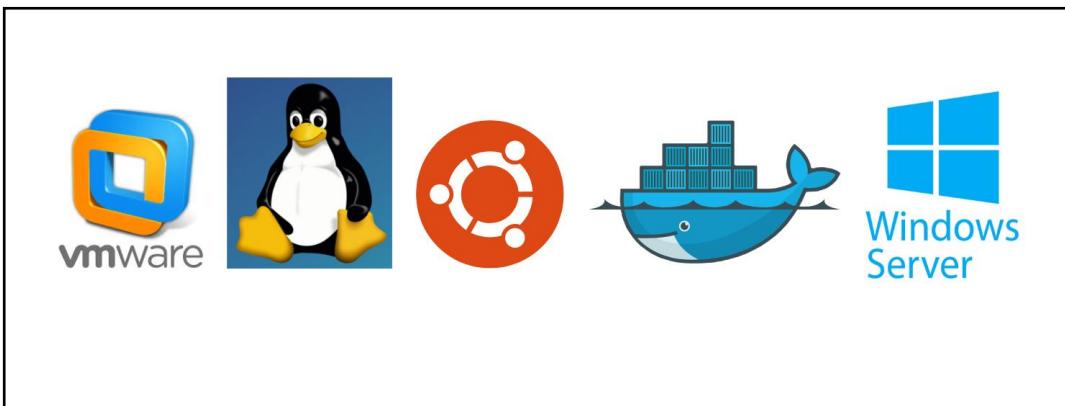




ROYAUME DU MAROC
UNIVERSITE ABDELMALEK ESSAADI
FACULTE DES SCIENCES ET TECHNIQUES
DEPARTEMENT GENIE INFORMATIQUE
Master : Sécurité IT & Big Data



Déploiement d'une Infrastructure Hybride Virtualisée avec VMware ESXi et Docker dans un Environnement de Production



Réalisé par :

Chaimae Bouassab

FatimaEzzahraa Garoud

Encadré par :

Pr.Mohamed BEN AHMED

MASTER: Sécurité IT et Big Data

S2 2024/2025

Introduction du rapport.....	4
Architecture et Environnement	4
Semaine 1 – Planification et Installation.....	5
1. Tâches réalisées :.....	5
2. Choix de l'architecture.....	5
3. Installation des machines virtuelles.....	6
4. Configuration IP statique avec Netplan.....	9
5. Résultat : Connectivité réseau validée.....	10
6. Configuration de l'adaptateur réseau sur Windows Server :.....	11
7. Configuration de l'adresse IP statique sur la VM Windows Server :.....	12
8. Vérification de la connectivité réseau entre les VMs :.....	14
1. Test de connectivité entre les VMs (Ping Ubuntu ↔ Windows Server)	14
2. Test de l'accès à Internet depuis la VM Windows Server	14
Semaine 2 : Déploiement de Docker et Applications.....	14
1. Tâches réalisées sur VM Ubuntu Server (192.168.180.38) :	14
1.1 Installation de Docker.....	14
1.2 Déploiement de la pile applicative WordPress + MySQL.....	15
1.3 Vérification de l'accessibilité.....	15
2. Tâches réalisées sur VM Windows Server (192.168.180.32)	16
2.1 L'installation du rôle File and Storage Services	16
2.1.1 Lancement de l'assistant d'ajout de rôles et fonctionnalités.....	17
2.1.2 Sélectionner le type d'installation	18
2.1.3 Sélectionner le serveur de destination	19
2.1.4 Sélectionner les rôles de serveur.....	20
2.1 Vérification de l'installation	22
2.2 Création du partage de fichiers SMB – TechNovaShare	23
2.2.1 Creation du fichier partagé	24
2.2.4 Partage de donnés	27
2.3 Backup des bases de données.....	29
2.3.1 Déterminer les volumes à sauvegarder	29
2.3.2 Programmer la sauvegarde	29
2.3.2.1 Tester le sauvgarde	30
Semaine 3 : Monitoring et Sécurité	31
1. Objectif général	31
2. Configuration des Snapshots VMware	31
3. Monitoring et Sécurité : Partie Ubuntu Server (192.168.180.38)	33

3.1	Lancement des services.....	34
2.4	Accès à Grafana.....	34
2.5	Importation d'un tableau de bord prédéfini (cAdvisor Dashboard).....	37
1) Sauvegarde automatique des données :	39
	HTTPS avec Certbot	41
	Snapshot:	41
	Sécurisation (pare-feu + HTTPS).....	42
	Partie 1 : Pare-feu avec UFW	42
	Partie 2 : HTTPS avec Certbot (auto-signé, car pas de domaine).....	43
	Semaine 4: Tests et Optimisation.....	46
1.	Architecture :.....	46
2.	Teste des services avec jmeter.....	46
	Conclusion de semaine 4.....	49
	Architecture cible du projet TechNova	50
	Conclusion:.....	52

Introduction du rapport

L'objectif principal de ce projet est de **migrer l'infrastructure physique d'une startup vers une solution virtualisée**, en assurant une meilleure flexibilité, une isolation des services, et une facilité de déploiement et de supervision. Pour cela, nous avons mis en place un environnement composé de **VMs Ubuntu et Windows Server** hébergées sous VMware ESXi, dans lequel nous avons intégré des applications web et des bases de données sous forme de **conteneurs Docker**.

Ce rapport retrace les différentes étapes de conception, de mise en œuvre, de test et d'optimisation de cette infrastructure. Il met également en évidence les choix techniques effectués, les outils utilisés, les défis rencontrés, ainsi que les résultats obtenus.

Nous avons réalisé un projet intégrant une architecture hybride combinant des machines virtuelles (VMware Workstation) et des conteneurs Docker. Le but est de déployer une application web (WordPress + MySQL), de la moniturer, de mettre en place des sauvegardes automatiques, et de protéger l'environnement avec des mécanismes de sécurité (pare-feu et HTTPS).

Architecture et Environnement

Dans le cadre de ce projet, nous avons déployé une architecture hybride composée de deux machines virtuelles (VM) installées sur **VMware ESXi** :

- **VM Ubuntu Server 24.04**

Rôle : serveur Docker avec les conteneurs Prometheus, Grafana et cAdvisor.

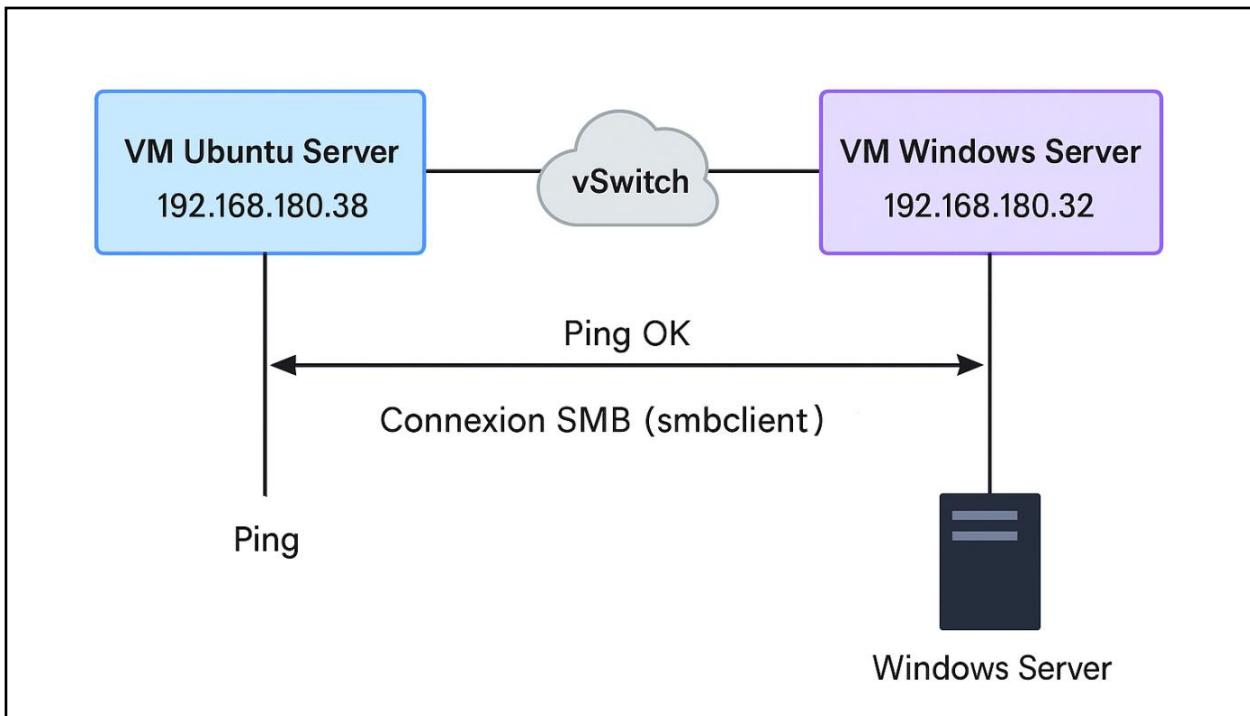
Adresse IP : **192.168.180.38**

- **VM Windows Server 2022**

Rôle : serveur de fichiers (SMB) avec partage de volume et interface distante via RDP.

Adresse IP : **192.168.180.32**

- ✓ Les deux machines sont connectées à un **vSwitch ESXi commun**, sur le même sous-réseau (192.168.180.0/24) avec des **adresses IP statiques** configurées manuellement.



Semaine 1 – Planification et Installation

1. Tâches réalisées :

1.1 Étude des besoins de la startup :

Nous avons défini que l'infrastructure devait permettre :

- La surveillance des performances (CPU, RAM, réseau) via Grafana.
- La haute disponibilité des services.
- Des sauvegardes automatiques (Docker volume, base MySQL).
- La sécurité réseau (pare-feu, HTTPS).

2. Choix de l'architecture

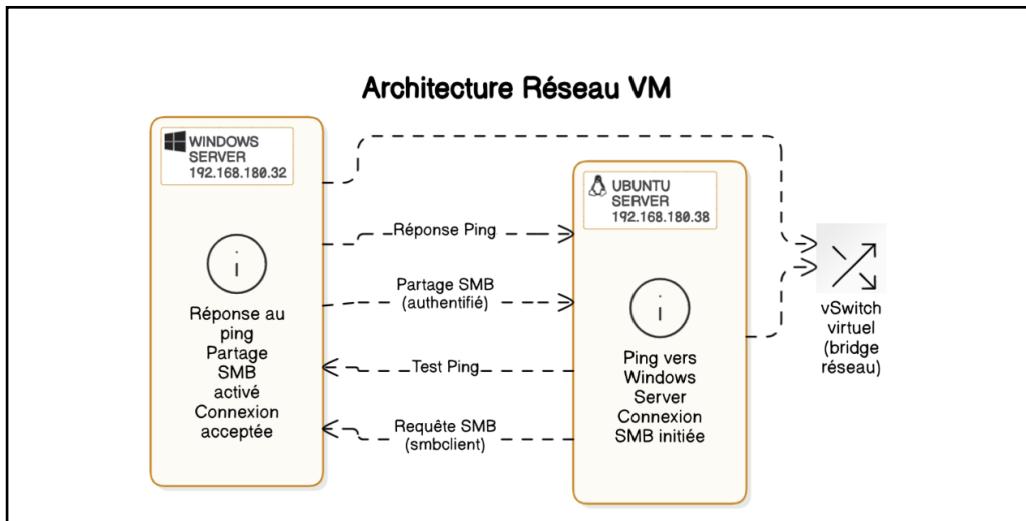
Nous avons décidé d'utiliser une infrastructure **hybride**, basée sur :

- **VMware ESXi** pour la virtualisation.
- Deux machines virtuelles :

- **VM Ubuntu Server** (adresse IP : 192.168.180.38) pour l'hébergement des conteneurs Docker (Grafana, Prometheus, cAdvisor, WordPress...).
- **VM Windows Server** (192.168.180.32) pour les tests de connectivité et le partage SMB.

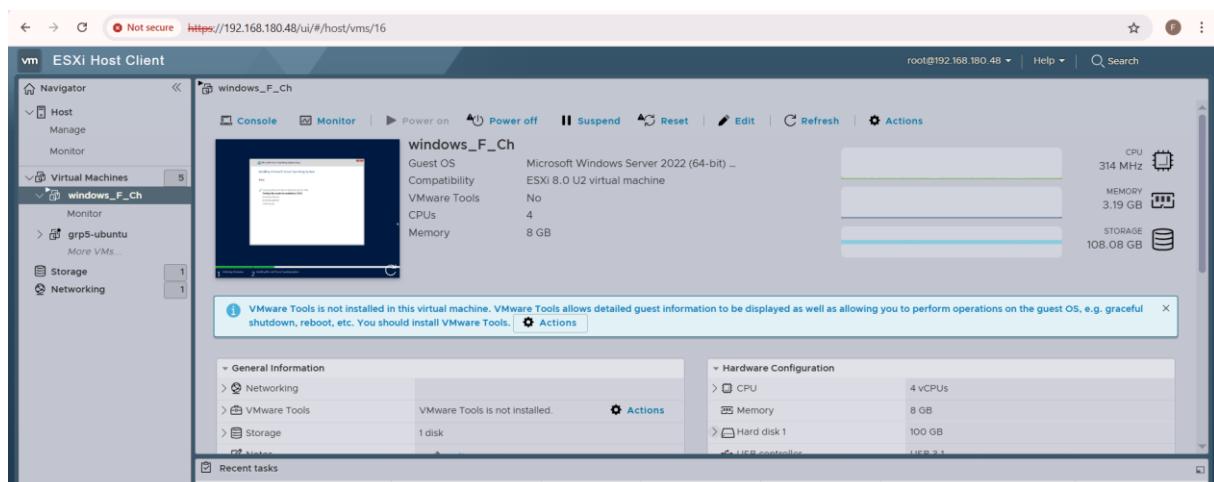
Un schéma réseau a été généré avec [Eraser.io](#), comprenant :

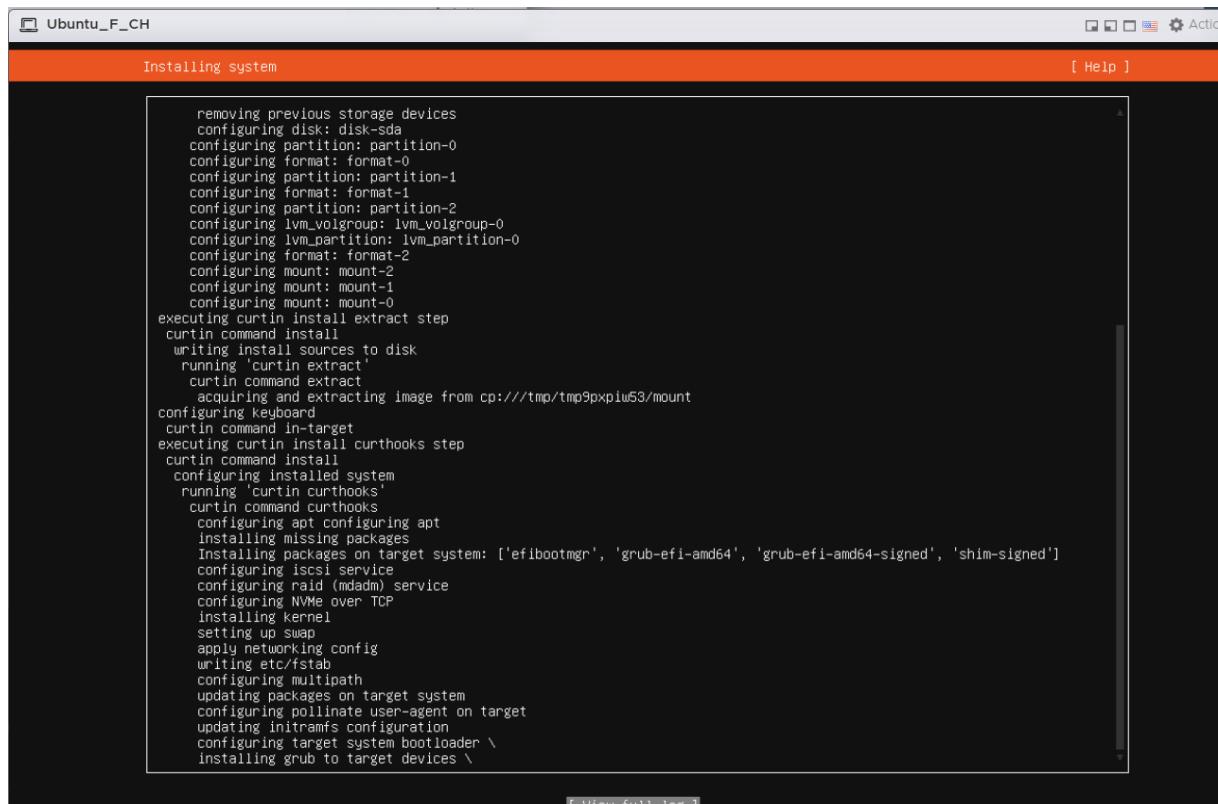
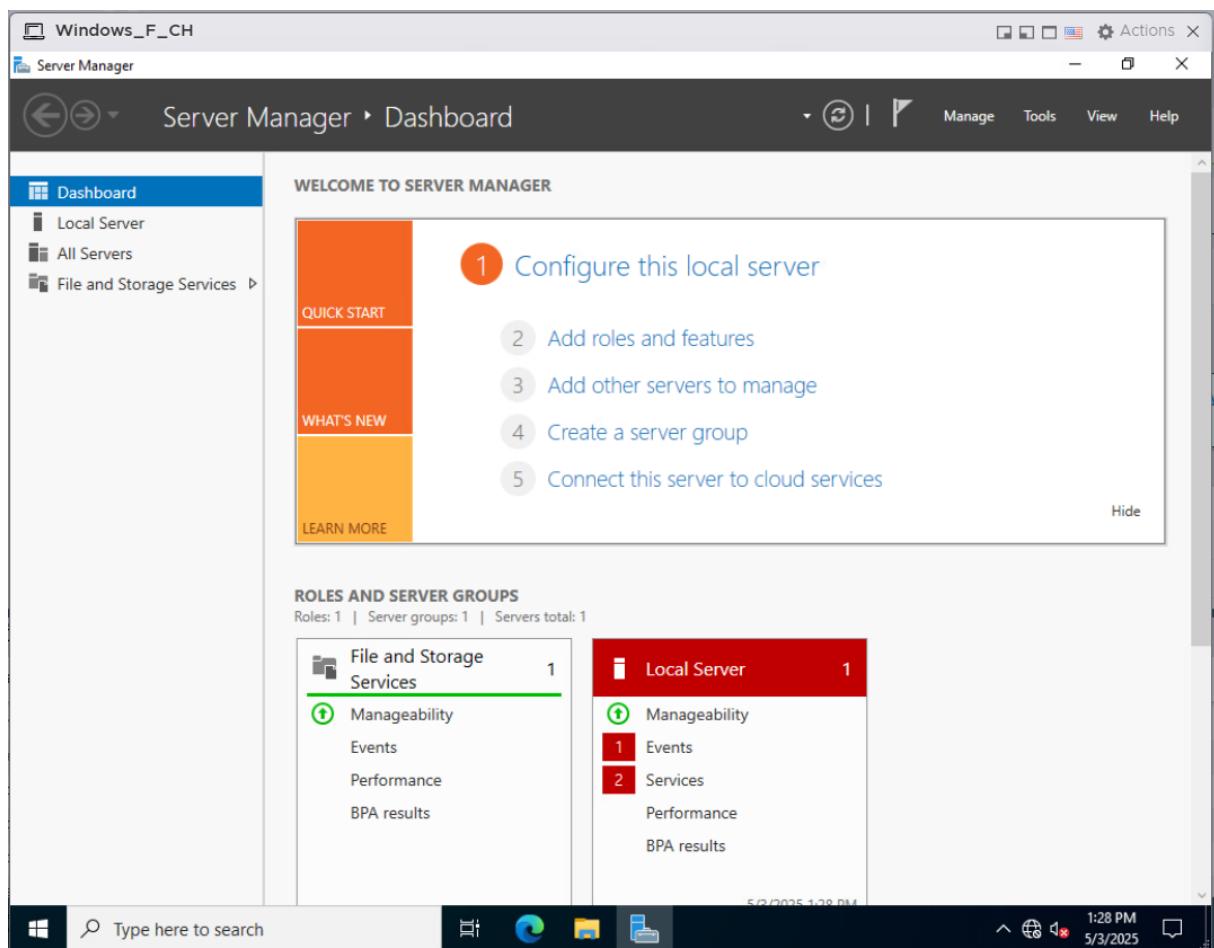
- Un **vSwitch virtuel** entre les deux VMs.
- Des flèches représentant les **tests de ping** et la **connexion SMB** réussie.



3. Installation des machines virtuelles

- Installation de **VMware Workstation** sur nos hôtes.
- Création de la VM Ubuntu et de la VM Windows, avec configuration des ressources (RAM, disque, carte réseau NAT/Bridged).





Not secure <https://192.168.180.48/ui/#/host/vms/15>

vm ESXi Host Client

Ubuntu_F_CH

Console **Monitor** | **Power on** **Shutdown** **Suspend** **Restart** | **Edit** **Refresh** | **Actions**

Ubuntu_F_CH

Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 8.0 U2 virtual machine
VMware Tools: Yes
CPU: 4
Memory: 8 GB
Host name: ubuntu-server

General Information

- Networking:** Host name: ubuntu-server, IP addresses: 1. 192.168.180.23, 2. fe80::20c:29ff:fe1e:97ff
- VMware Tools:** VMware Tools is not managed by vSphere
- Storage:** 1 disk
- Notes:** [Edit notes](#)

Hardware Configuration

- CPU:** 4 vCPUs
- Memory:** 8 GB
- Hard disk 1:** 100 GB
- USB controller:** USB 2.0
- Network adapter 1:** VM Network (Connected)
- Video card:** 16 MB
- CD/DVD drive 1:** ISO [datastore1] ISO_Files/ubuntu-24.04.2-live-server-amd64.iso [Select disc image](#)

Recent tasks

Task	Target	Initiator	Queued	Started	Result	Completed

```
Ubuntu_F_CH

Hint: Num Lock on

ubuntu@ubuntu:~$ 
ubuntu@ubuntu:~$ login: ubuntu
Password: 
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat May  3 12:54:16 PM UTC 2025

System load:  0.46      Processes:           220
Usage of /:   13.3% of 47.41GB   Users logged in:        0
Memory usage: 3%          IPv4 address for ens34: 192.168.180.23
Swap usage:   0%          IPv6 address for ens34: fe80::20c:29ff:fe1e:97ff

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

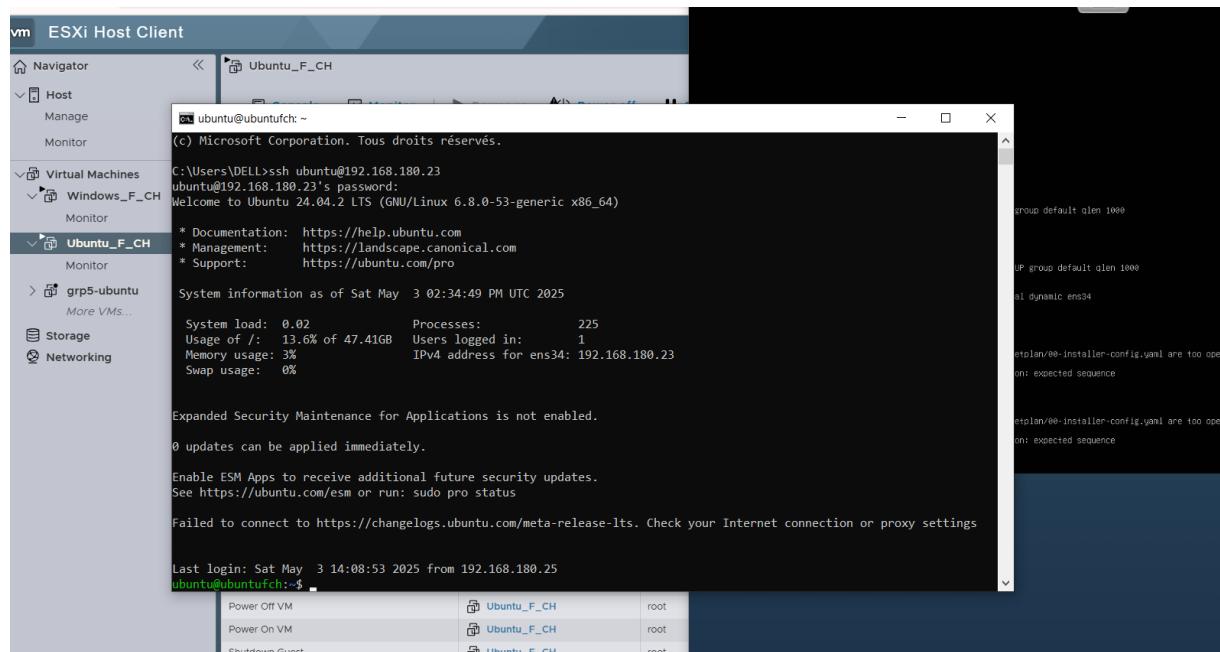
ubuntu@ubuntu:~$ ^C
ubuntu@ubuntu:~$ ^C
ubuntu@ubuntu:~$ ^C
ubuntu@ubuntu:~$ ^C
ubuntu@ubuntu:~$ ^C
```

```

ubuntu@ubuntufch:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openSSH-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntufch:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openSSH-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntufch:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
ubuntu@ubuntufch:~$ sudo systemctl start ssh
ubuntu@ubuntufch:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
    Active: active (running) since Sat 2025-05-03 13:40:19 UTC; 11s ago
TriggeredBy: • ssh.socket
      Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 1517 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1519 (sshd)
   Tasks: 1 (limit: 9441)
  Memory: 2.1M (peak: 2.1M)
    CPU: 31ms
     CGroup: /system.slice/ssh.service
             └─1519 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

May 03 13:40:19 ubuntufch systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 03 13:40:19 ubuntufch sshd[1519]: Server listening on :: port 22.
May 03 13:40:19 ubuntufch systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
ubuntu@ubuntufch:~$
```

Pour permettre la connexion distante sécurisée à notre machine Ubuntu Server (192.168.180.38), nous avons installé et activé le service **OpenSSH**. Cette étape est essentielle pour la gestion à distance, notamment pour l'administration ou les transferts de fichiers (par exemple via scp ou rsync).



4. Configuration IP statique avec Netplan

Le fichier /etc/netplan/00-installer-config.yaml a été modifié pour attribuer une IP statique à Ubuntu, assurant une communication fluide avec la VM Windows.

```
ubuntu@ubuntufch:~$ sudo nano /etc/netplan/00-installer-config.yaml
```



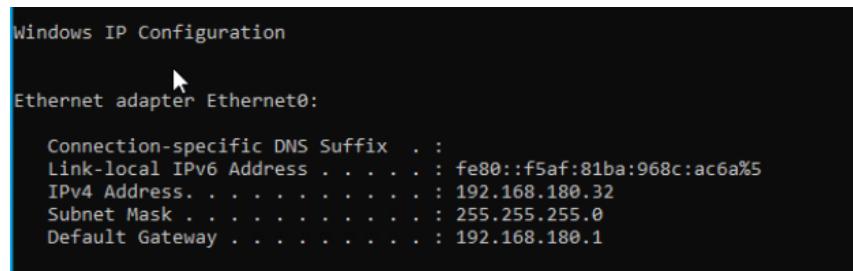
```
GNU nano 7.2
network:
  version: 2
  ethernets:
    ens34:
      dhcp4: no
      addresses:
        - 192.168.180.23/24
      gateway4: 192.168.180.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

5. Résultat : Connectivité réseau validée

Une fois l'IP statique configurée et SSH opérationnel :

- La machine Ubuntu est accessible via SSH à partir d'un autre hôte sur le même réseau (ex : machine physique ou VM Windows).
- Elle peut communiquer avec la VM Windows via un ping, ou établir une connexion SMB.
- Elle a un accès Internet fonctionnel grâce à la passerelle et aux DNS configurés.

```
ubuntu@ubuntufch:~$ ping 192.168.180.23
PING 192.168.180.23 (192.168.180.23) 56(84) bytes of data.
64 bytes from 192.168.180.23: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 192.168.180.23: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 192.168.180.23: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 192.168.180.23: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 192.168.180.23: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 192.168.180.23: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 192.168.180.23: icmp_seq=7 ttl=64 time=0.050 ms
64 bytes from 192.168.180.23: icmp_seq=8 ttl=64 time=0.050 ms
64 bytes from 192.168.180.23: icmp_seq=9 ttl=64 time=0.053 ms
64 bytes from 192.168.180.23: icmp_seq=10 ttl=64 time=0.051 ms
64 bytes from 192.168.180.23: icmp_seq=11 ttl=64 time=0.050 ms
64 bytes from 192.168.180.23: icmp_seq=12 ttl=64 time=0.049 ms
64 bytes from 192.168.180.23: icmp_seq=13 ttl=64 time=0.049 ms
64 bytes from 192.168.180.23: icmp_seq=14 ttl=64 time=0.050 ms
64 bytes from 192.168.180.23: icmp_seq=15 ttl=64 time=0.048 ms
64 bytes from 192.168.180.23: icmp_seq=16 ttl=64 time=0.072 ms
64 bytes from 192.168.180.23: icmp_seq=17 ttl=64 time=0.049 ms
64 bytes from 192.168.180.23: icmp_seq=18 ttl=64 time=0.039 ms
64 bytes from 192.168.180.23: icmp_seq=19 ttl=64 time=0.050 ms
64 bytes from 192.168.180.23: icmp_seq=20 ttl=64 time=0.047 ms
64 bytes from 192.168.180.23: icmp_seq=21 ttl=64 time=0.051 ms
```



Attribution des IP statiques via netplan (Ubuntu) et l'interface réseau (Windows).

Configuration réseau (vSwitch)

- Les deux VMs ont été reliées à un même **réseau virtuel privé (vSwitch)**.
- Des tests ont été réalisés :
 - **Ping réciproque** entre Ubuntu et Windows.
 - Connexion **SMB depuis Ubuntu vers Windows** via smbclient.

6. Configuration de l'adaptateur réseau sur Windows Server :

The screenshot displays the VMware ESXi Host Client interface, specifically the Networking section. The top window shows the 'Port groups' tab with the following data:

Name	Active ports	VLAN ID	Type	vSwitch	VMs
VM Network	0	0	Standard port group	vSwitch0	7
Management Network	1	0	Standard port group	vSwitch0	N/A

The bottom window shows the configuration for 'vSwitch0':

- Type: Standard vSwitch
- Port groups: 2
- Uplinks: 3
- vSwitch Details**: MTU 1500, Ports 3840 (3830 available), Link discovery Listen / Cisco discovery protocol (CDP), Attached VMs 9 (0 active), Beacon interval 1.
- NIC teaming policy**: Notify switches yes, Policy Route based on originating port ID, Reverse policy Yes, Fallback Yes.
- Security policy**: Allow promiscuous mode No, Allow forged transmits No, Allow MAC changes No.

A vSwitch topology diagram on the right illustrates the network connections, showing VMs connected to vSwitch0, which is then connected to physical adapters (vmnic0, vmnic3, vmnic1).

Afin d'assurer une communication stable entre les deux machines virtuelles (Ubuntu et Windows Server), l'adaptateur réseau de la VM Windows Server a été configuré manuellement.

Depuis l'interface graphique du **Server Manager**, nous avons accédé aux paramètres réseau et modifié les propriétés de la carte Ethernet en configurant l'**adresse IPv4** statique suivante :

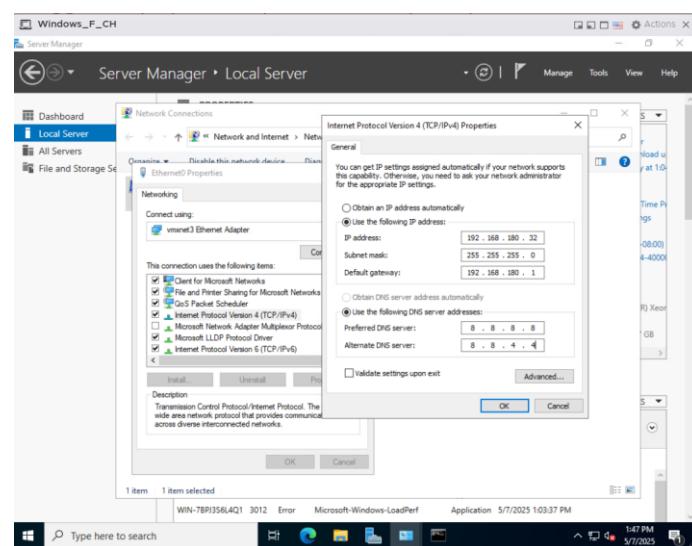
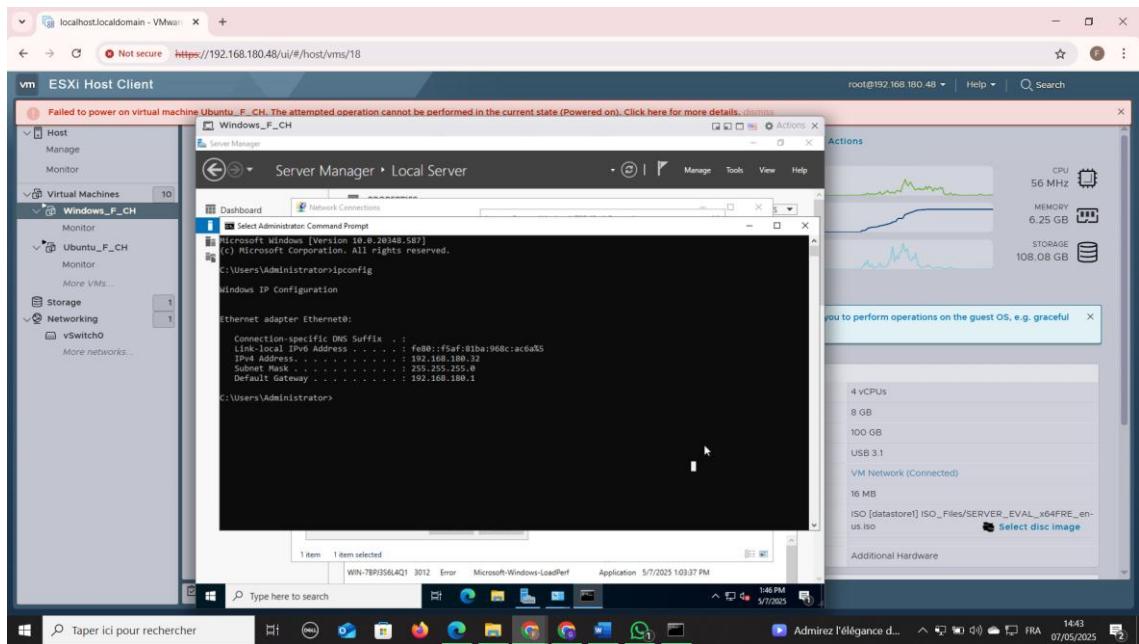
- **Adresse IP** : 192.168.180.32
- **Masque de sous-réseau** : 255.255.255.0
- **Passerelle par défaut** : 192.168.180.1
- **DNS préféré** : 8.8.8.8
- **DNS alternatif** : 1.1.1.1

Cette configuration permet à la VM Windows Server de :

- Communiquer localement avec la VM Ubuntu Server (192.168.180.38)
- Accéder à Internet via la passerelle
- Être joignable pour les connexions SMB et les pings

7. Configuration de l'adresse IP statique sur la VM Windows Server :

Afin d'assurer une communication stable entre les machines virtuelles (VM), nous avons configuré une adresse IP statique sur le **Windows Server**, hébergé sous VMware ESXi. Cette configuration garantit que l'adresse IP de la VM ne changera pas à chaque redémarrage, ce qui est crucial pour les tests réseau (ping, SMB) et les connexions de monitoring (Prometheus/Grafana)



8. Vérification de la connectivité réseau entre les VMs :

1. Test de connectivité entre les VMs (Ping Ubuntu ↔ Windows Server)

Nous avons commencé par un test de ping à partir de la VM Windows Server vers la VM Ubuntu

(192.168.180.38).

Ce test permet de vérifier que les deux machines virtuelles sont bien connectées au même réseau virtuel (vSwitch) et peuvent échanger des paquets IP.

```
C:\Users\Administrator>ping 192.168.180.32
Pinging 192.168.180.32 with 32 bytes of data:
Reply from 192.168.180.32: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.180.32:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. Test de l'accès à Internet depuis la VM Windows Server

Ensuite, un test de connectivité vers une ressource externe (ici google.com) a été effectué.

Ce test est important pour vérifier que la VM peut accéder à Internet, ce qui est nécessaire pour les mises à jour, les téléchargements de paquets et la connexion à des services cloud.

```
Pinging google.com [172.217.168.174] with 32 bytes of data:
Reply from 172.217.168.174: bytes=32 time=18ms TTL=114
Reply from 172.217.168.174: bytes=32 time=38ms TTL=114
Reply from 172.217.168.174: bytes=32 time=18ms TTL=114
Reply from 172.217.168.174: bytes=32 time=18ms TTL=114

Ping statistics for 172.217.168.174:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 18ms, Maximum = 38ms, Average = 23ms

C:\Users\Administrator>
```

Semaine 2 : Déploiement de Docker et Applications

1. Tâches réalisées sur VM Ubuntu Server (192.168.180.38) :

1.1 Installation de Docker

Docker et Docker Compose ont été installés sur la VM Ubuntu Server pour permettre la gestion et l'orchestration de conteneurs

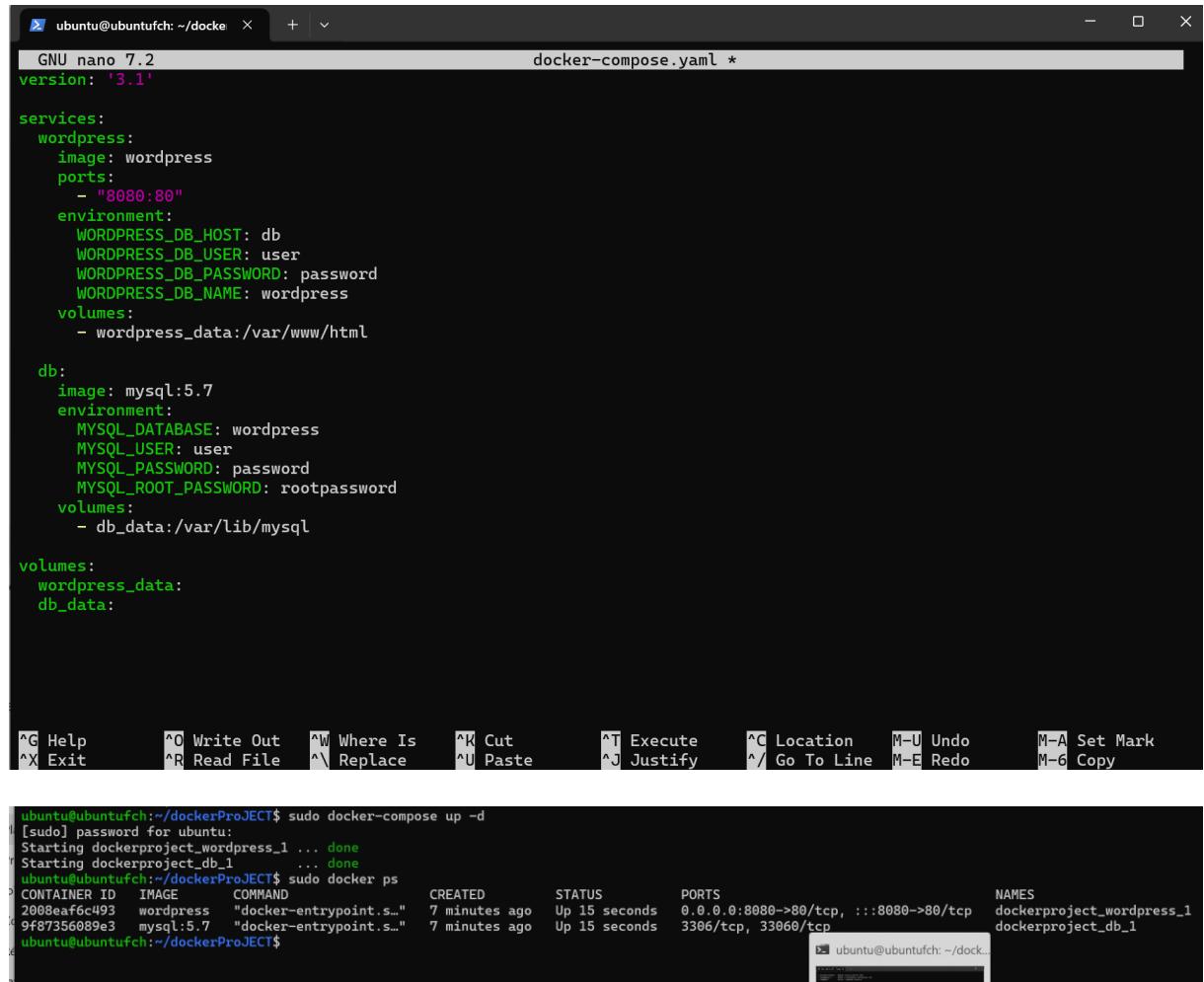
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ubuntufch:~$ sudo systemctl start docker
ubuntu@ubuntufch:~$ sudo systemctl enable docker
ubuntu@ubuntufch:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

```
[root@ubuntufch ~]# docker-compose --version
docker-compose version 1.29.2, build unknown
[root@ubuntufch ~]
```

1.2Déploiement de la pile applicative WordPress + MySQL

Nous avons créé un fichier docker-compose.yml pour déployer simultanément :

- un conteneur WordPress (frontend + PHP),
- un conteneur MySQL (base de données backend).



```
ubuntu@ubuntufch:~/docke ~$ nano docker-compose.yaml
version: '3.1'

services:
  wordpress:
    image: wordpress
    ports:
      - "8080:80"
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: user
      WORDPRESS_DB_PASSWORD: password
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - wordpress_data:/var/www/html

  db:
    image: mysql:5.7
    environment:
      MYSQL_DATABASE: wordpress
      MYSQL_USER: user
      MYSQL_PASSWORD: password
      MYSQL_ROOT_PASSWORD: rootpassword
    volumes:
      - db_data:/var/lib/mysql

volumes:
  wordpress_data:
  db_data:
```

ubuntu@ubuntufch:~/docke ~\$ sudo docker-compose up -d

```
[sudo] password for ubuntu:
Starting dockerproject_wordpress_1 ... done
Starting dockerproject_db_1 ... done
ubuntu@ubuntufch:~/docke ~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2008eaf6c493 wordpress "docker-entrypoint.s..." 7 minutes ago Up 15 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp dockerproject_wordpress_1
9f87356089e3 mysql:5.7 "docker-entrypoint.s..." 7 minutes ago Up 15 seconds 3306/tcp, 33060/tcp dockerproject_db_1
ubuntu@ubuntufch:~/docke ~$
```

1.3Vérification de l'accessibilité

Une fois les conteneurs lancés (docker-compose up -d), l'application WordPress a été testée depuis un navigateur distant en accédant à l'adresse :

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title Virtualisation Hybrid Project

Username admin
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password &0GRpoFXv)r0Vvzor1 Hide

Your Email shaimaabouassab@gmail.com
Double-check your email address before continuing.

Search engine visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

Install WordPress

Howdy, admin

Dashboard

Welcome to WordPress!

Learn more about the 6.8.1 version.

Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
[Add a new page](#)

Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
[Open site editor](#)

Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
[Edit styles](#)

Site Health Status
No information yet...
Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.

At a Glance

Quick Draft
Title
Content
What's on your mind?

Drag boxes here

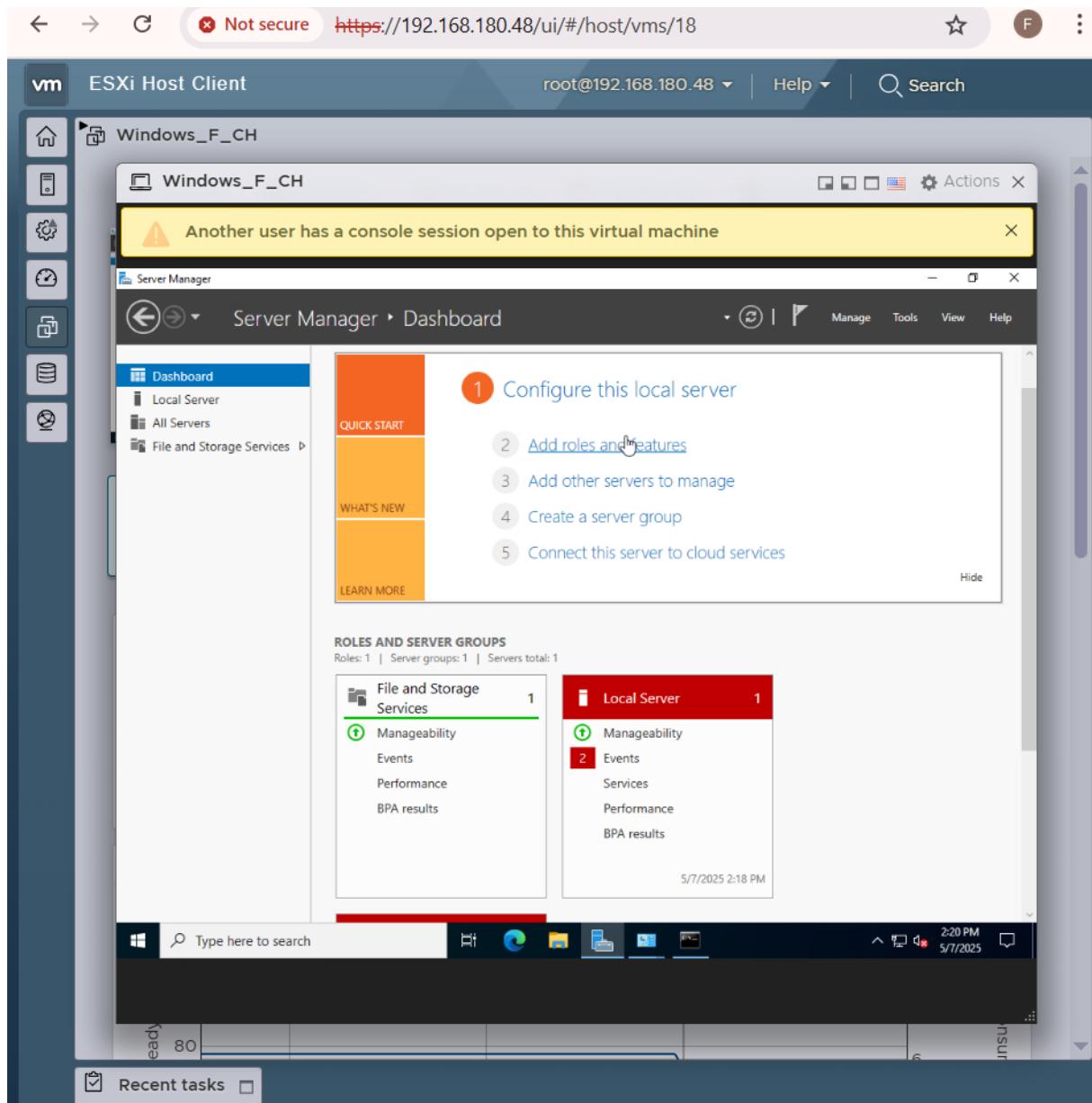
2. Tâches réalisées sur VM Windows Server (192.168.180.32)

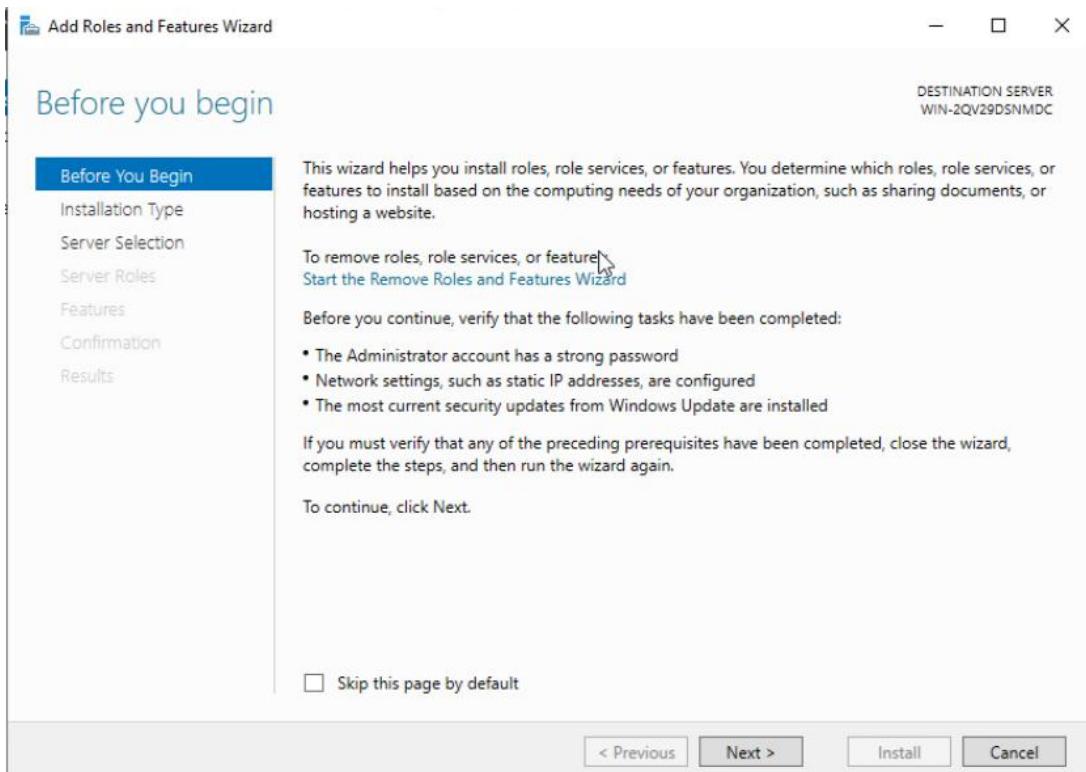
Objectif : Mettre en place un service de partage de fichiers sécurisé (SMB) pour permettre aux utilisateurs du réseau local et à la VM Ubuntu d'accéder facilement à des fichiers partagés, tout en préparant la machine à l'intégration de services supplémentaires (HTTPS, sauvegardes, etc.).

2.1 L'installation du rôle File and Storage Services

L'installation du rôle « File and Storage Services » a été réalisée via le Gestionnaire de serveur pour activer les fonctionnalités nécessaires au partage SMB. Cette étape a permis d'ajouter les composants essentiels pour gérer les partages de fichiers de manière efficace et sécurisée.

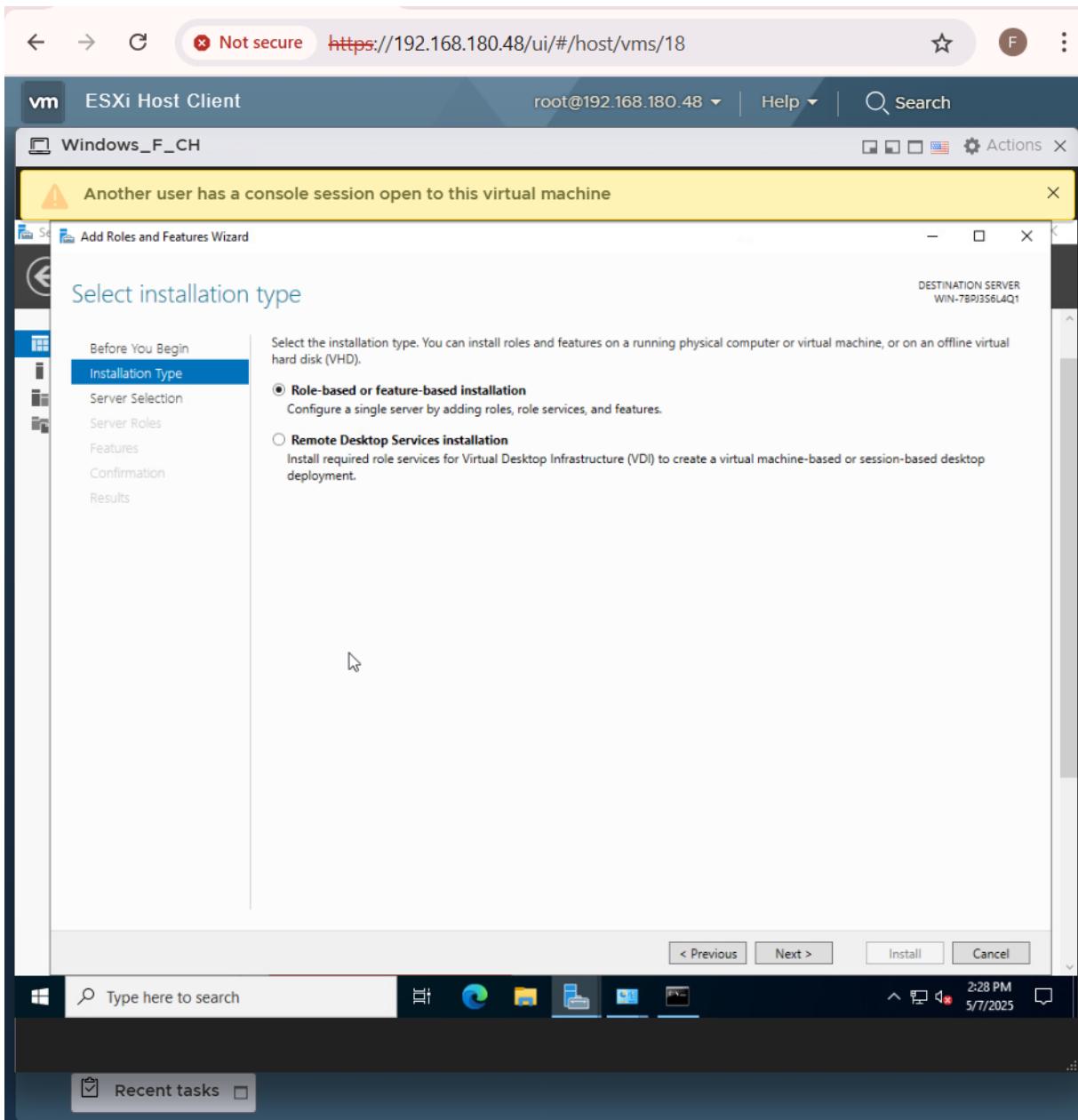
2.1.1 Lancement de l'assistant d'ajout de rôles et fonctionnalités





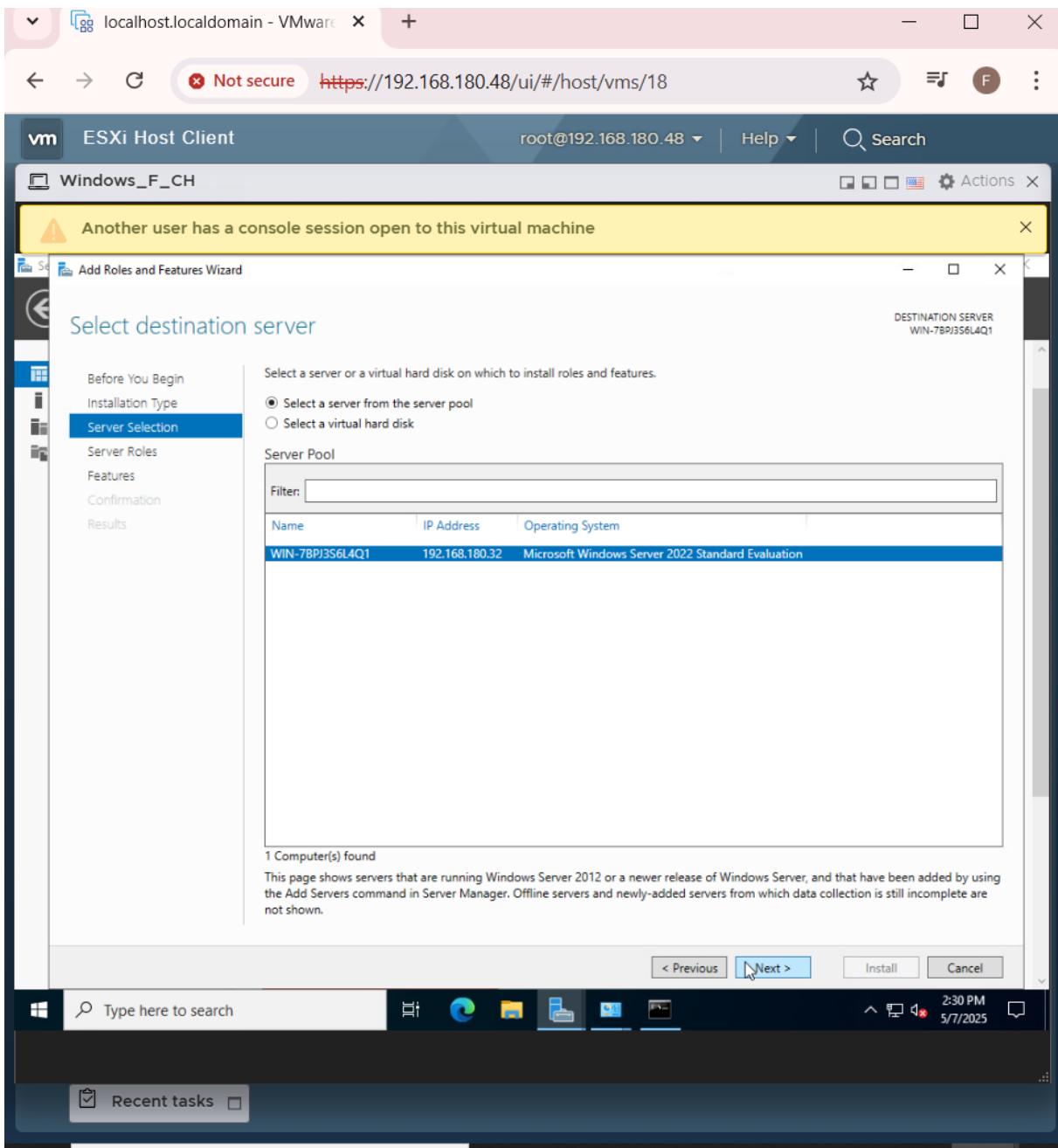
Depuis le tableau de bord ou la section "Servers", sélectionnez "Add roles and features" pour ouvrir l'assistant "Add Roles and Features Wizard". La première fenêtre ("Before You Begin") informe l'utilisateur sur le processus d'installation des rôles, services ou fonctionnalités, et liste les prérequis : un mot de passe administrateur fort, des adresses IP statiques, et des mises à jour de sécurité installées.

2.1.2 Sélectionner le type d'installation



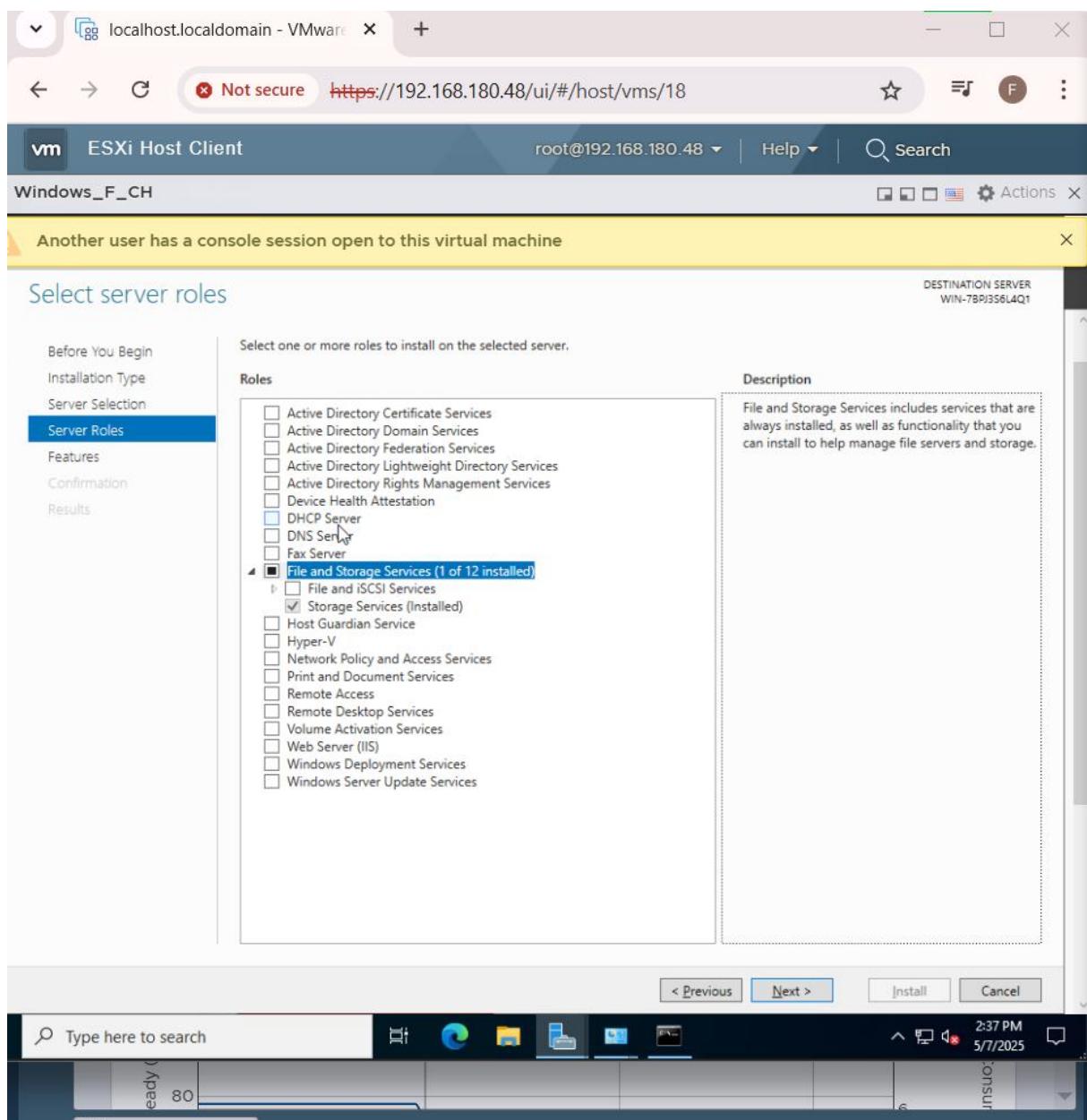
L'assistant demande de choisir le type d'installation : "Role-based or feature-based installation" (installation de rôles ou fonctionnalités sur un serveur unique) ou "Remote Desktop Services installation" (pour une infrastructure VDI). Ici, l'option "Role-based or feature-based installation" est sélectionnée pour configurer des rôles spécifiques sur le serveur local.

2.1.3 Sélectionner le serveur de destination

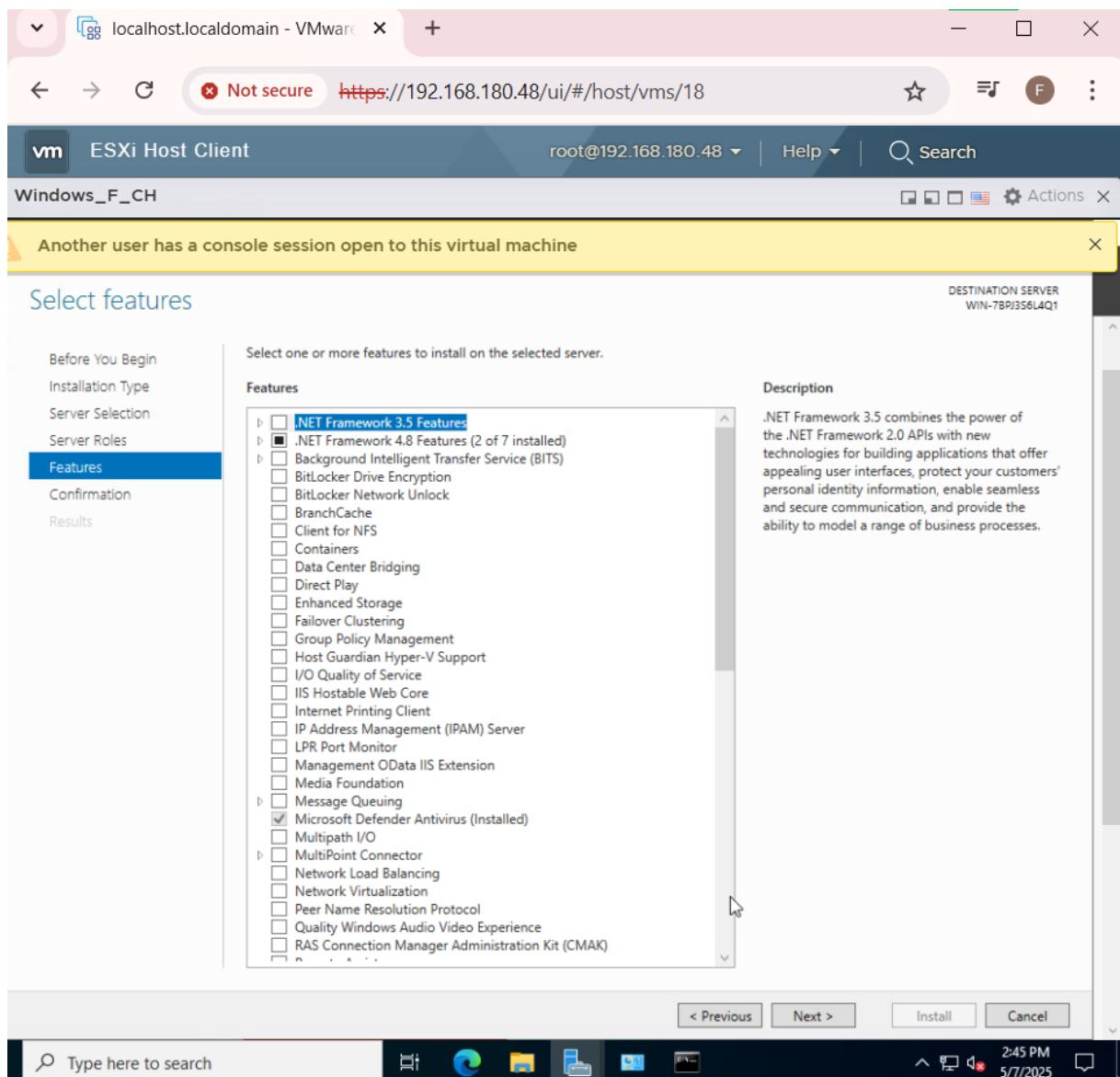


L'assistant affiche la liste des serveurs disponibles dans le pool de serveurs (ici, WIN-2QV29D5NMDC). Cette étape confirme que les rôles et fonctionnalités seront installés sur le serveur local, qui exécute Windows Server 2022 Datacenter Evaluation. Seuls les serveurs en ligne sont affichés.

2.1.4 Sélectionner les rôles de serveur

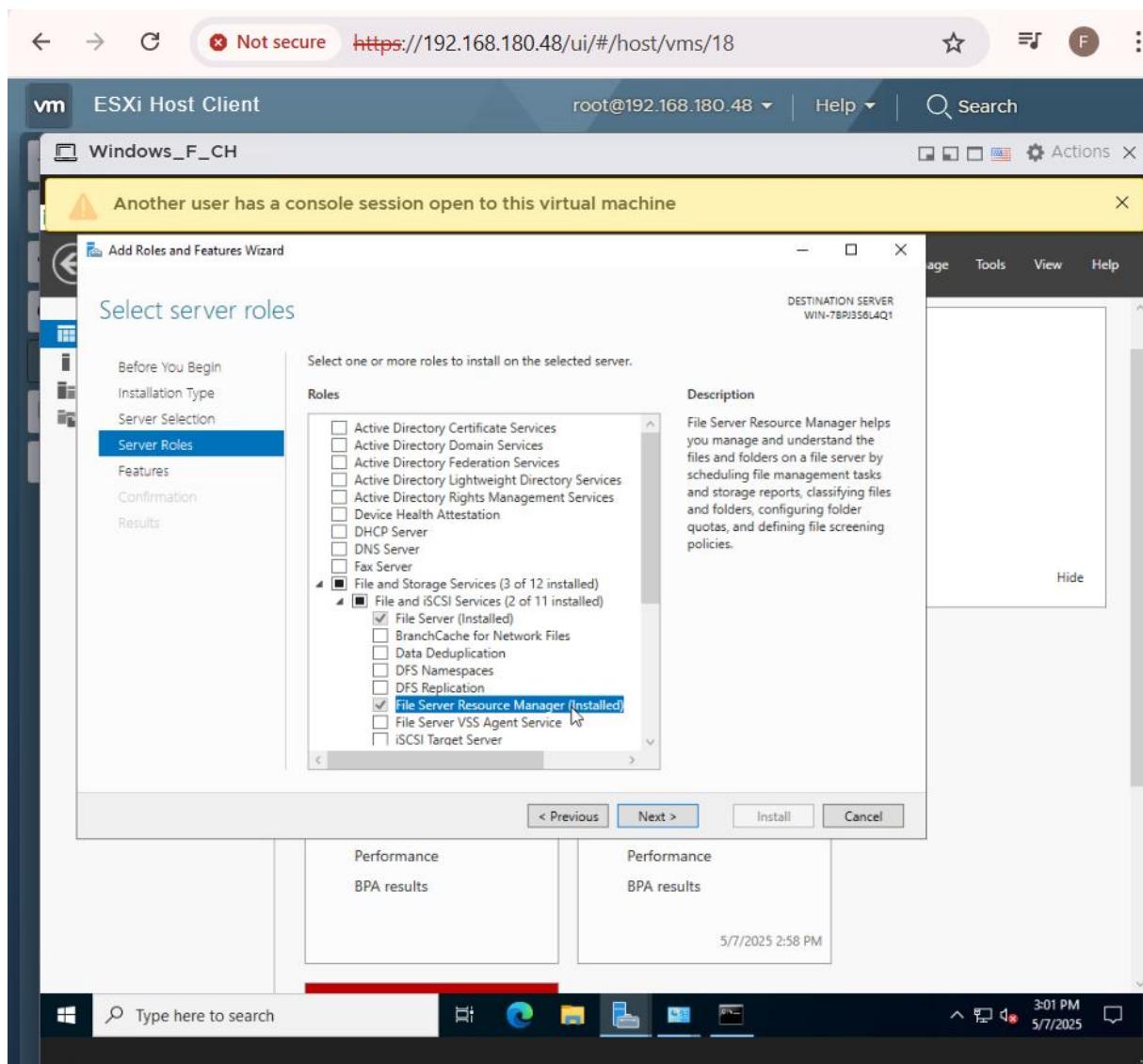


L'assistant présente une liste de rôles disponibles à installer. Ici, "File and Storage Services" est déjà installé, et l'utilisateur coche "Fax Server" et "iSCSI Target Server" pour les ajouter. Ces rôles permettent respectivement la gestion des fax et la création de cibles iSCSI pour le stockage en réseau. D'autres options comme "BranchCache" ou "Data Deduplication" peuvent être sélectionnées selon les besoins.



2.1 Vérification de l'installation

Après l'installation, une vérification dans le Gestionnaire de serveur a confirmé que « File and Storage Services » était actif, garantissant que les outils nécessaires pour configurer les partages étaient disponibles.



➤ File and Storage Services est bien installé, prêt à être utilisé.

2.2 Création du partage de fichiers SMB – TechNovaShare

Dans le cadre des besoins de collaboration de l’entreprise TechNova, un dossier dédié nommé **TechNovaShare** a été créé sur la VM **Windows Server (192.168.180.32)**. Cette ressource constitue la base du **partage de fichiers via le protocole SMB** (Server Message Block), assurant un accès simple et sécurisé aux documents depuis d’autres machines du réseau, notamment la VM Ubuntu.

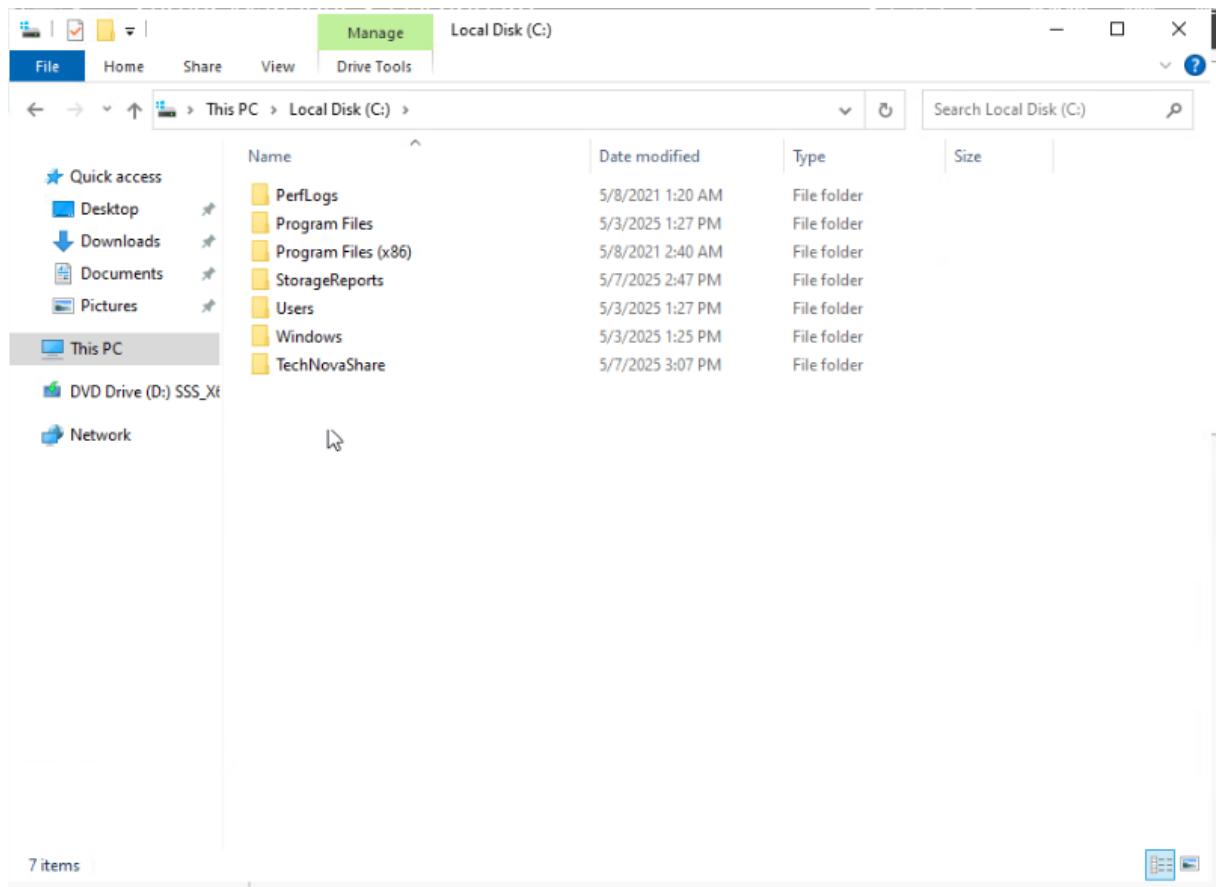
Étapes réalisées :

- **Création de l’utilisateur local TechNovaUser** via la console de gestion des utilisateurs Windows.
- **Attribution des permissions** :

- Le dossier TechNovaShare a été configuré pour accorder à TechNovaUser les droits **Read/Write** (lecture et écriture).
- D'autres utilisateurs du système n'ont **pas** accès par défaut, garantissant que seul TechNovaUser peut interagir avec les fichiers partagés

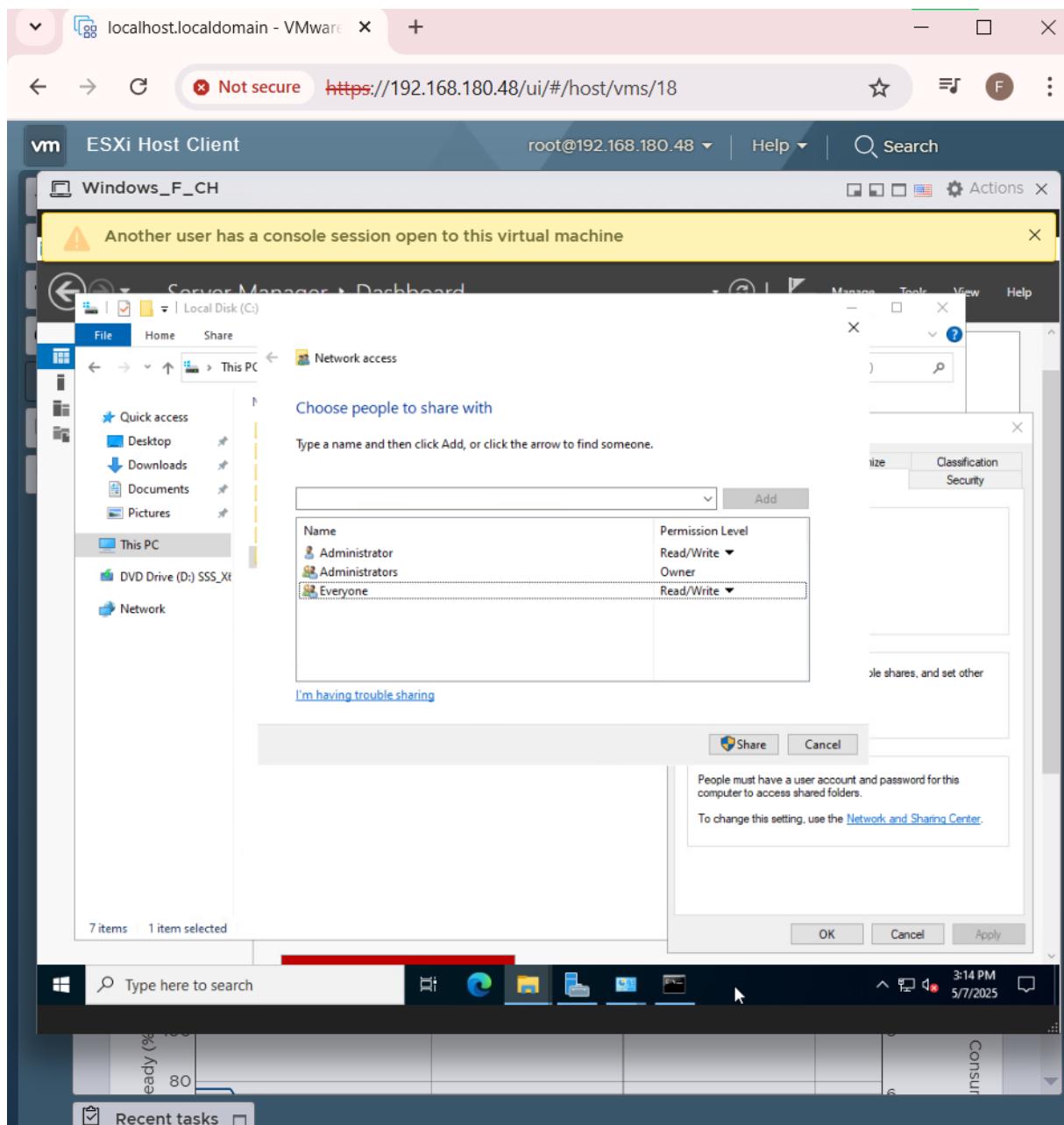
2.2.1 Creation du fichier partagé

Au niveau du serveur Windows, nous avons créé un dossier nommé Files2 sur le disque local C, et c'est ce dossier que nous souhaitons partager avec le serveur web via un protocole SMB pour héberger les sauvegardes.



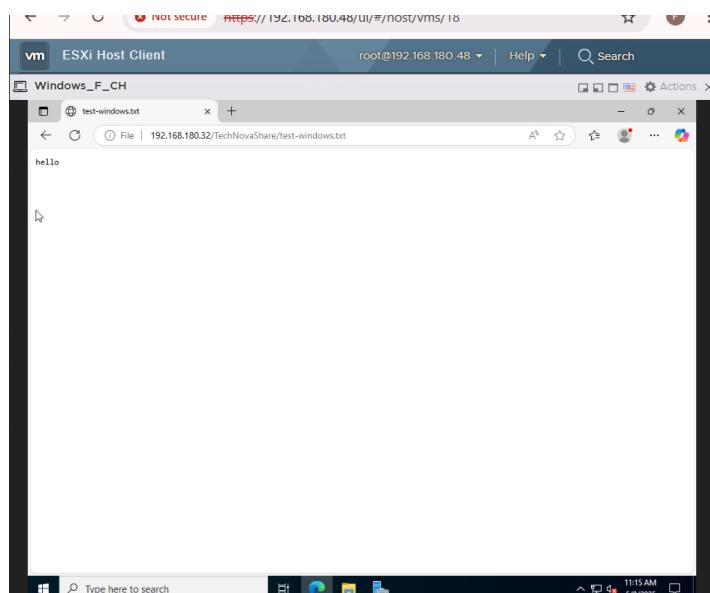
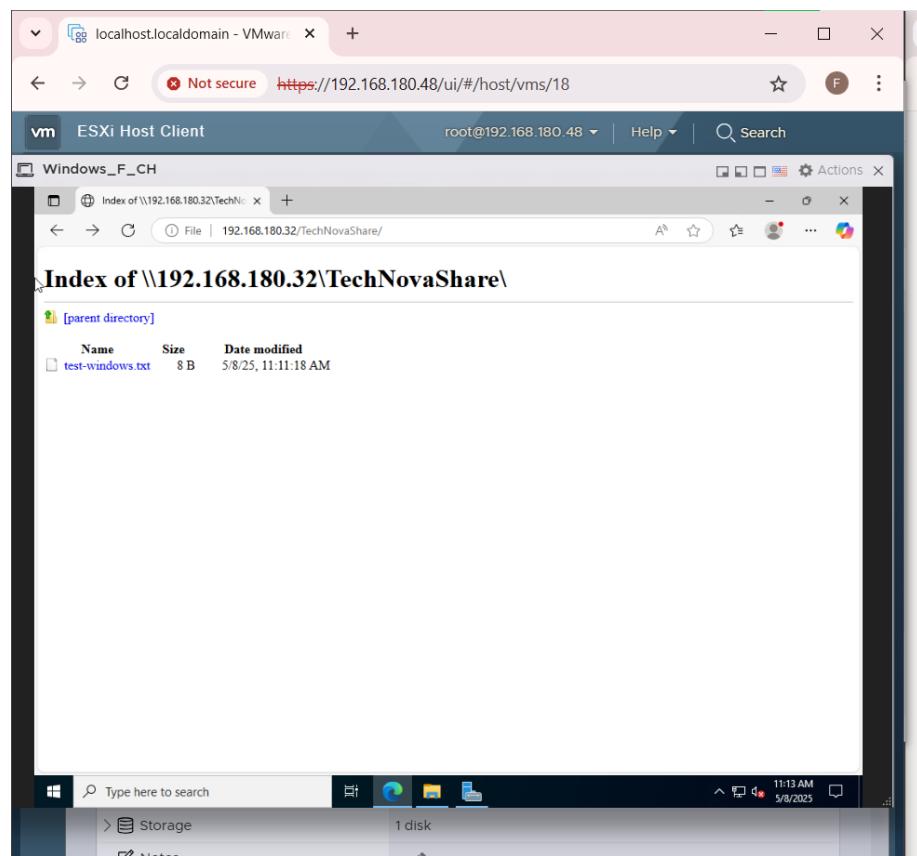
Cette étape permet désormais à la VM Ubuntu d'accéder au dossier partagé pour :

- déployer des sauvegardes,
- échanger des fichiers journaliers,
- ou connecter automatiquement certains conteneurs Docker pour des opérations de lecture/écriture.



Pour renforcer la sécurité et le contrôle d'accès au dossier partagé TechNovaShare, un utilisateur dédié nommé **TechNovaUser** a été créé au sein de la VM **Windows Server (192.168.180.32)**.

2.2.2 Vérification de l'accès au partage SMB



Le fichier test-windows.txt est apparu correctement dans le répertoire partagé.

L'accès s'est effectué sans erreur, validant ainsi la **connectivité SMB**, les **permissions d'écriture** et la **visibilité réseau** entre la machine Windows Server et les autres clients du réseau local (ex : Ubuntu Server).

2.2.3 Validation Réseau et Intégration des VMs

Afin de garantir une interconnexion fiable entre les machines virtuelles du projet, plusieurs vérifications ont été réalisées :

1. Test de connectivité réseau

- Des tests de **ping mutuels** ont été effectués entre la **VM Ubuntu Server** (192.168.180.38) et la **VM Windows Server** (192.168.180.32).
- Ces tests ont confirmé une **connectivité stable**, essentielle pour le bon fonctionnement des services partagés.

```
ubuntu@ubuntufch:~$ smbclient -L //192.168.180.32/ -U administrator
Password for [WORKGROUP\administrator]:
[...]
Sharename      Type      Comment
-----        ---       -----
ADMIN$         Disk      Remote Admin
C$            Disk      Default share
IPC$          IPC       Remote IPC
TechNovaShare  Disk      [...]
SMB1 disabled -- no workgroup available
```

- Test de ping :**

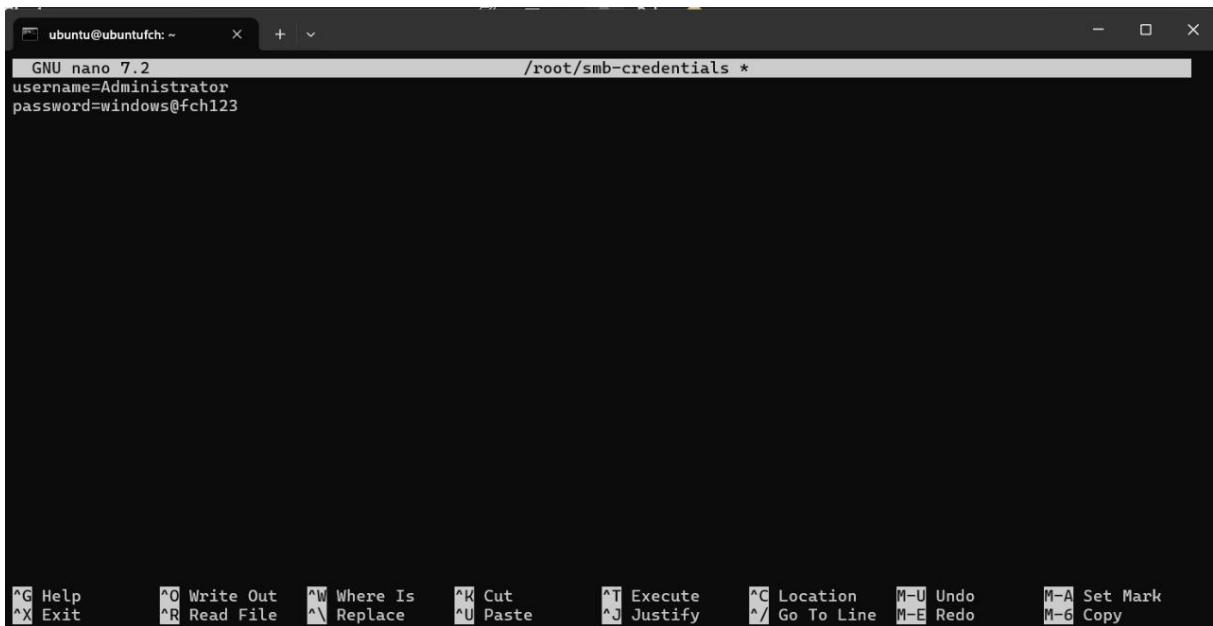
On peut faire le ping sur machine windows depuis la machine ubuntu :

```
ubuntu@ubuntufch:~$ ping 192.168.180.32
PING 192.168.180.32 (192.168.180.32) 56(84) bytes of data.
64 bytes from 192.168.180.32: icmp_seq=1 ttl=128 time=0.546 ms
64 bytes from 192.168.180.32: icmp_seq=2 ttl=128 time=0.621 ms
64 bytes from 192.168.180.32: icmp_seq=3 ttl=128 time=0.470 ms
64 bytes from 192.168.180.32: icmp_seq=4 ttl=128 time=0.604 ms
64 bytes from 192.168.180.32: icmp_seq=5 ttl=128 time=0.651 ms
64 bytes from 192.168.180.32: icmp_seq=6 ttl=128 time=0.687 ms
64 bytes from 192.168.180.32: icmp_seq=7 ttl=128 time=0.719 ms
64 bytes from 192.168.180.32: icmp_seq=8 ttl=128 time=0.592 ms
64 bytes from 192.168.180.32: icmp_seq=9 ttl=128 time=0.786 ms
64 bytes from 192.168.180.32: icmp_seq=10 ttl=128 time=0.685 ms
64 bytes from 192.168.180.32: icmp_seq=11 ttl=128 time=0.640 ms
```

2.2.4 Partage de données

Sur Ubuntu Server, nous avons installé cifs-utils pour permettre le montage de partages réseau Windows (SMB/CIFS), facilitant ainsi l'accès et le stockage de données, comme les sauvegardes de WordPress et MongoDB, via un point de montage tel que /mnt/windows-share.

Nous avons créé un fichier */root/smb_credentials* contenant les informations d'authentification du serveur Windows (nom d'utilisateur et mot de passe), puis sécurisé ses permissions avec chmod 600.



The screenshot shows a terminal window titled "ubuntu@ubuntufch: ~". The command "nano 7.2" was run to edit the file "/root/smb-credentials". The file contains the following text:

```
GNU nano 7.2
username=Administrator
password=windows@fchl23
```

The terminal window includes a menu bar at the top and a keyboard shortcut bar at the bottom.

Ensuite, nous avons créé le répertoire `/mnt/windows-share` à monter, où nous allons placer les fichiers à partager avec le serveur web.

```
ubuntu@ubuntufch:~$ sudo mkdir /mnt/windows-share
[sudo] password for ubuntu:
ubuntu@ubuntufch:~$
```

```
ubuntu@ubuntufch:~$ sudo mount -t cifs //192.168.180.32/TechNovaShare /mnt/windows-share -o credentials=/root/smb-credentials,file_mode=0777,dir_mode=0777
ubuntu@ubuntufch:~$
```

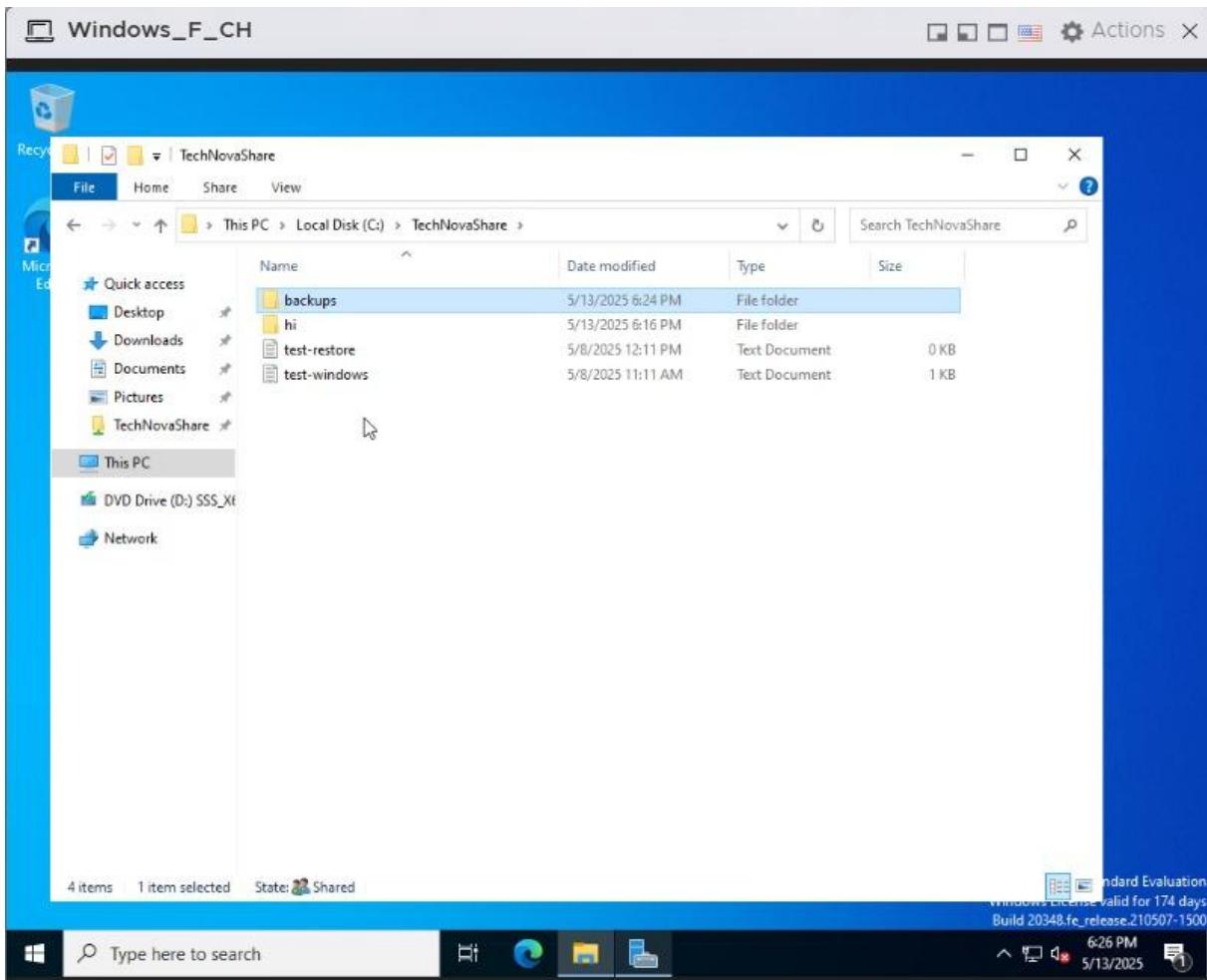
Pour tester, nous avons créé un fichier dans Ubuntu :

```
ubuntu@ubuntufch:/mnt/windows-share$ mkdir hi
```

En observe les fichiers existes dans Windows depuis Ubuntu :

```
ubuntu@ubuntufch:/mnt/windows-share$ ls
hi test-restore.txt test-windows.txt
ubuntu@ubuntufch:/mnt/windows-share$
```

En observe les fichiers existes dans Windows :



2.3 Backup des bases de données

Notre objectif est d'effectuer des sauvegardes des bases de données sur Windows Server, qui héberge notre système de gestion des données.

2.3.1 Déterminer les volumes à sauvegarder

```
ubuntu@ubuntufch:/mnt/windows-share$ docker volume ls
DRIVER      VOLUME NAME
local        8cce89176b87a3ee9872e6ba5ab3556f0a9b17929e3084eda7bc8f6e6f898694
local        80f06bd22a128dcc2ad334f41bf51b0db9ebd6765add5a54dbc579b41d9663f7
local        687cd0b618421dd32ab4a9e4cbce50b86641bd178b48705f9650bc9c46aa337f
local        dockerproject_db_data
local        dockerproject_wordpress_data
local        dockerproject_wp_data
local        monitoring_grafana-storage
local        ubuntu_mysql_data
local        wordpress_data
ubuntu@ubuntufch:/mnt/windows-share$
```

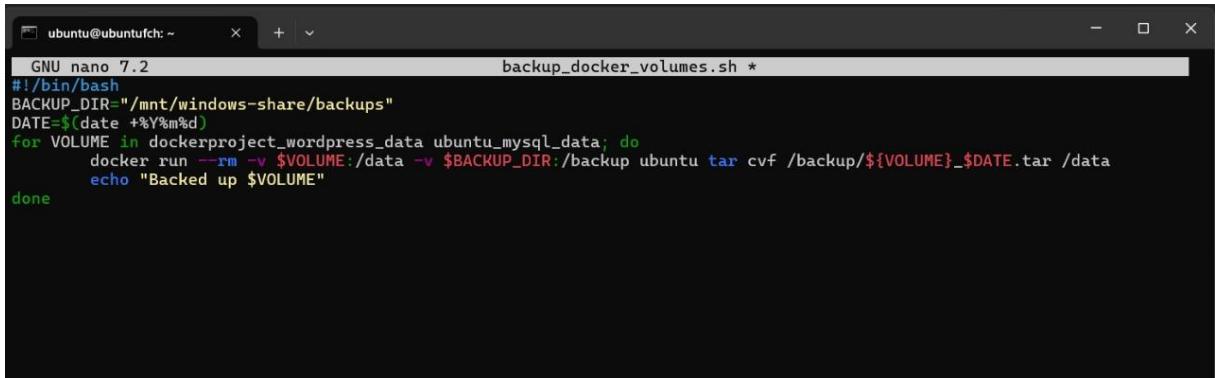
Nous avons sélectionné les volumes `dockerproject_wordpress_data` et `ubuntu_mysql_data` comme cibles de notre sauvegarde

2.3.2 Programmer la sauvegarde

Création du dossier `backup` dans le répertoire partagé

```
ubuntu@ubuntufch:/mnt/windows-share$ mkdir backups
ubuntu@ubuntufch:/mnt/windows-share$
```

Modifier le fichier backup_docker_volumes.sh pour copier les données des volumes vers la destination des sauvegardes.

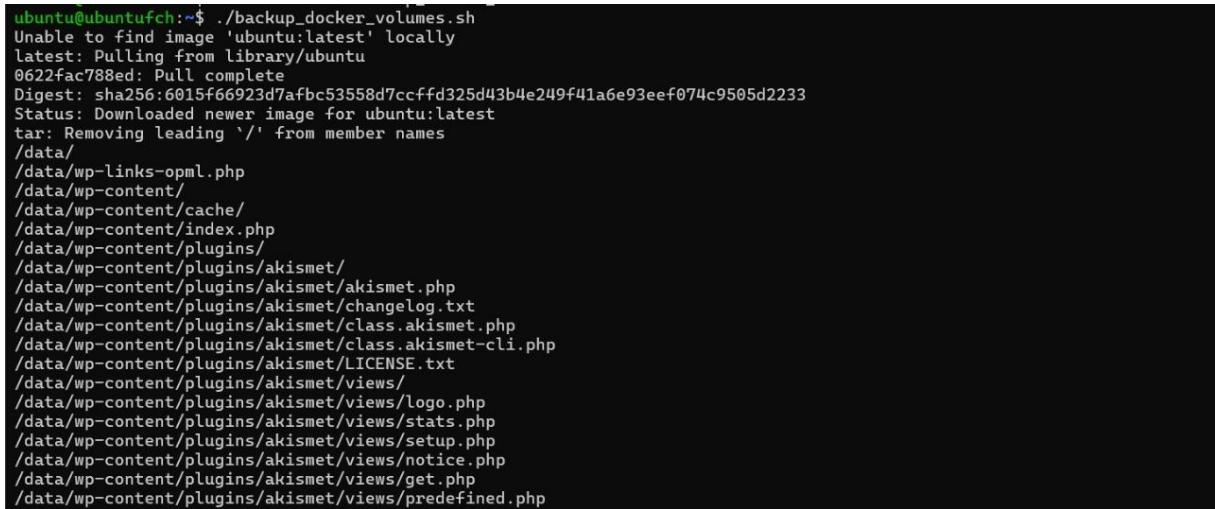


```
GNU nano 7.2                                     backup_docker_volumes.sh *
#!/bin/bash
BACKUP_DIR="/mnt/windows-share/backups"
DATE=$(date +%Y%m%d)
for VOLUME in dockerproject_wordpress_data ubuntu_mysql_data; do
    docker run --rm -v $VOLUME:/data -v $BACKUP_DIR:/backup ubuntu tar cvf /backup/${VOLUME}_${DATE}.tar /data
    echo "Backed up $VOLUME"
done
```

2.3.2.1 Tester le sauvegarde

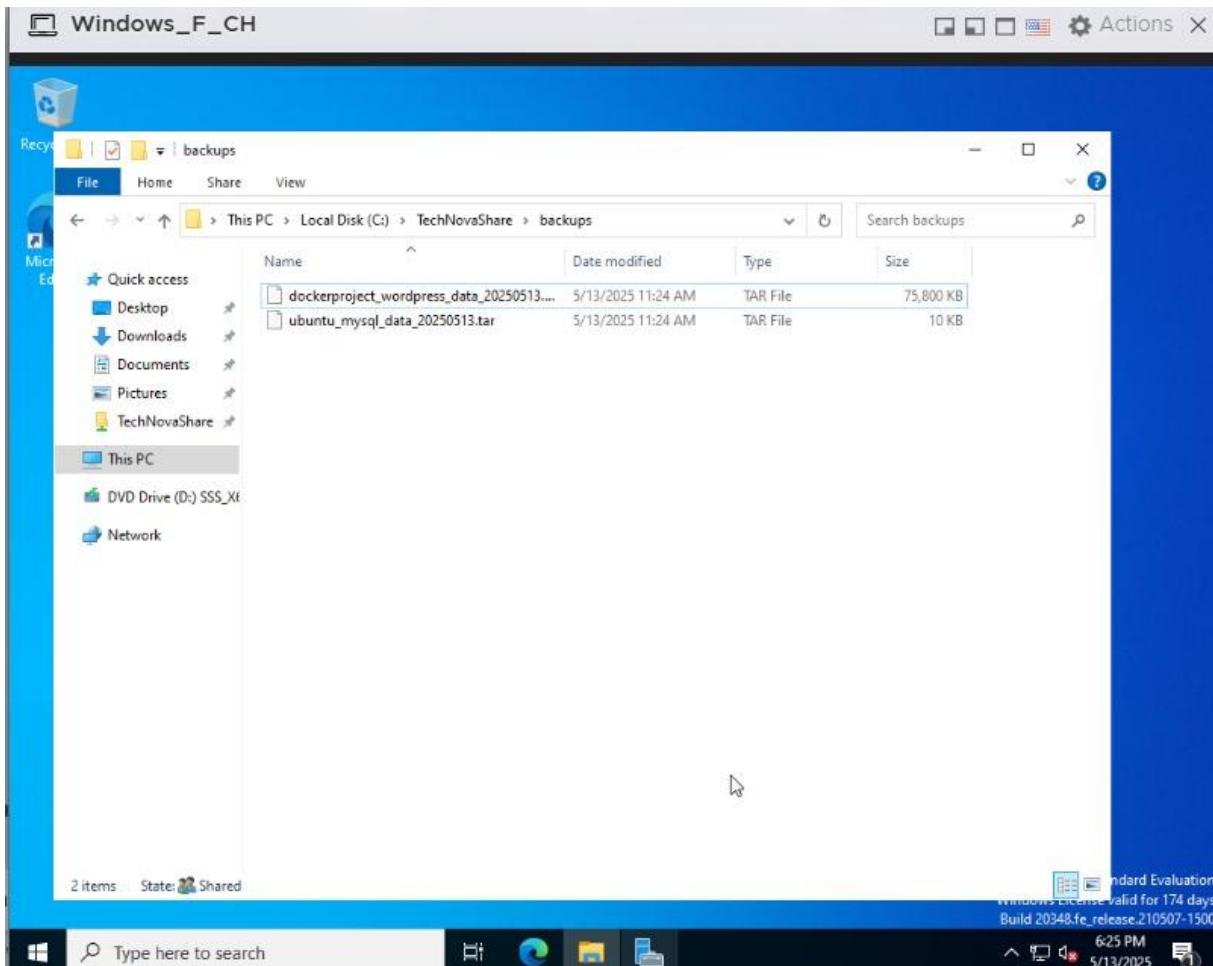
Nous avons modifié les permissions du fichier backup_docker_volumes.sh afin de le rendre exécutable.

```
ubuntu@ubuntufch:~$ sudo chmod +x backup_docker_volumes.sh
ubuntu@ubuntufch:~$
```



```
ubuntu@ubuntufch:~$ ./backup_docker_volumes.sh
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
0622fac788ed: Pull complete
Digest: sha256:6015f66923d7afbc53558d7ccffd325d43b4e249f41a6e93ee074c9505d2233
Status: Downloaded newer image for ubuntu:latest
tar: Removing leading '/' from member names
/data/
/data/wp-links-opml.php
/data/wp-content/
/data/wp-content/cache/
/data/wp-content/index.php
/data/wp-content/plugins/
/data/wp-content/plugins/akismet/
/data/wp-content/plugins/akismet/akismet.php
/data/wp-content/plugins/akismet/changelog.txt
/data/wp-content/plugins/akismet/class.akismet.php
/data/wp-content/plugins/akismet/class.akismet-cli.php
/data/wp-content/plugins/akismet/LICENSE.txt
/data/wp-content/plugins/akismet/views/
/data/wp-content/plugins/akismet/views/logo.php
/data/wp-content/plugins/akismet/views/stats.php
/data/wp-content/plugins/akismet/views/setup.php
/data/wp-content/plugins/akismet/views/notice.php
/data/wp-content/plugins/akismet/views/get.php
/data/wp-content/plugins/akismet/views/predefined.php
```

Sur Windows nous remarquons que les backups sont enregistrés avec succès.



Semaine 3 : Monitoring et Sécurité

1. Objectif général

L'objectif principal de cette semaine était d'assurer la **stabilité, la traçabilité et la sécurité** du système mis en place. Cela comprenait :

- La **mise en place d'un monitoring visuel** (Prometheus + Grafana)
- La **création de snapshots VMware** pour permettre des restaurations rapides
- L'application de **mesures de sécurité réseau** (pare-feu UFW, certificats HTTPS)

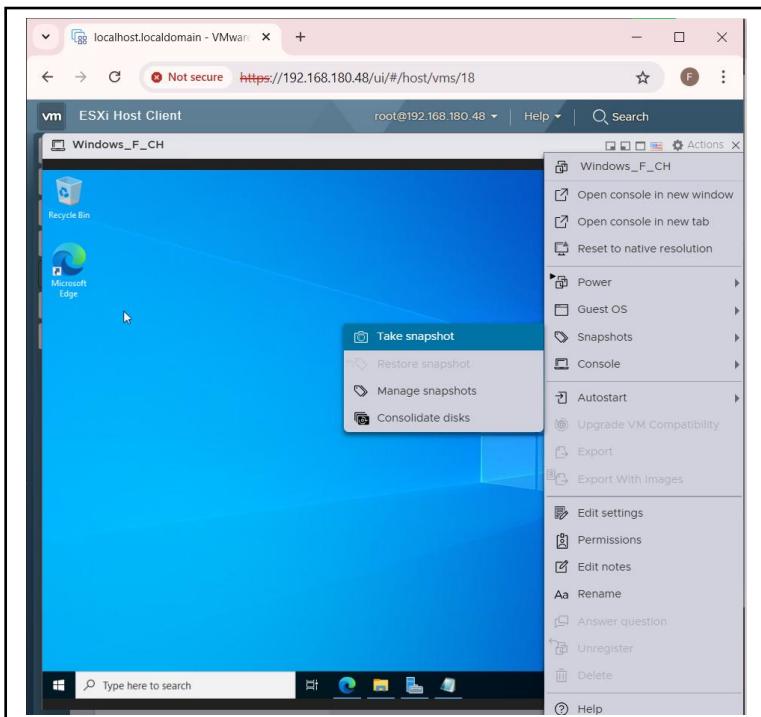
2. Configuration des Snapshots VMware

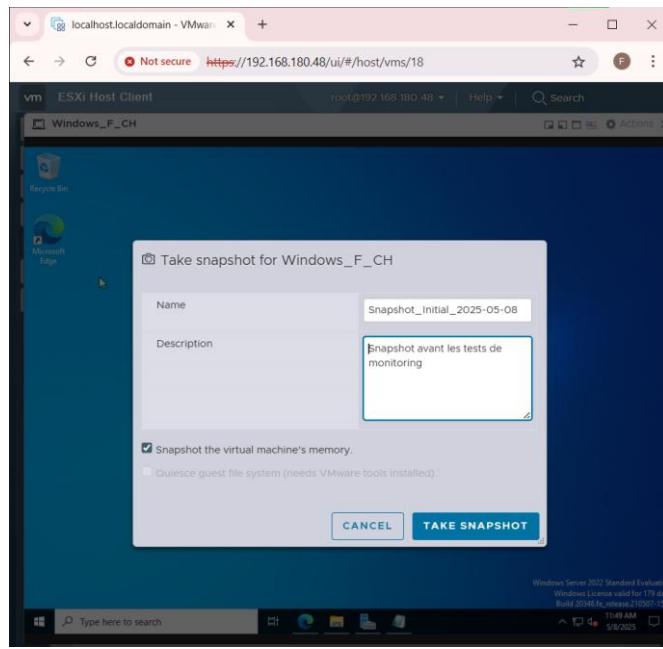
Objectif :

Créer un point de restauration stable de la VM Windows Server afin de préserver l'état fonctionnel du système avant le déploiement de solutions de monitoring et la mise en place des règles de sécurité.

Étapes réalisées :

- Ouverture de la console **VMware ESXi**
- Sélection de la VM Windows Server (192.168.180.32)
- Clic droit > **Snapshots** > *Take Snapshot*
- Nom : Snapshot_Initial_2025-05-08
- Description : "Snapshot avant les tests de monitoring"
- Sauvegarde avec mémoire incluse (option cochée)





3. Monitoring et Sécurité : Partie Ubuntu Server (192.168.180.38)

Installation et configuration de Docker Compose pour le monitoring

- Déploiement de Prometheus, Grafana et cAdvisor sur Ubuntu via un fichier docker-compose.yml.
- Le fichier prometheus.yml a été adapté pour surveiller :
 - localhost:9090 (Prometheus lui-même),
 - cadvisor:8080 (cAdvisor pour les métriques Docker).
- Objectif : surveiller en temps réel les ressources des conteneurs Docker (CPU, mémoire, réseau).

```
ubuntu@ubuntufch: ~/monit ~ + v
GNU nano 7.2                                     docker-compose.yml

version: '3'
services:
  prometheus:
    image: prom/prometheus
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
    ports:
      - "9090:9090"

  grafana:
    image: grafana/grafana
    ports:
      - "3000:3000"
```

```
GNU nano 7.2                                         prometheus.yml
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'docker'
    static_configs:
      - targets: ['localhost:9090']
```

3.1 Lancement des services

```
ubuntu@ubuntufch:~/monitoring$ sudo docker-compose up -d
[sudo] password for ubuntu:
Creating network "monitoring_default" with the default driver
Pulling prometheus (prom/prometheus:)...latest: Pulling from prom/prometheus
9fa9226be034: Pull complete
1617e25568b2: Pull complete
dd1e13a1db5e: Pull complete
85395c032c94: Pull complete
96a99fc8b470: Pull complete
6d243588c032: Pull complete
c66110f5fa59: Pull complete
759fe8dc37ae: Pull complete
d71c3f577a67: Pull complete
4c0194f7eb43: Pull complete
Digest: sha256:e2b8aa62b64855956e3ec1e18b4f9387fb6203174a4471936f4662f437f04405
Status: Downloaded newer image for prom/prometheus:latest
Pulling grafana (grafana/grafana:)...latest: Pulling from grafana/grafana
f18232174bc9: Pull complete
0ece60a185bb: Pull complete
915e4f5b7bd8: Pull complete
23b4b38b1cbe: Pull complete
877a1369766b: Pull complete
665f506d012f: Pull complete
deca3a715210: Pull complete
felff72baa5f5: Pull complete
b0de0e260b82: Pull complete
5b9c995414e7: Pull complete
Digest: sha256:263cbefcd5d9b179893c47c415daab4da5c1f3d6770154741eca4f45c81119884
Status: Downloaded newer image for grafana/grafana:latest
Creating monitoring_prometheus_1 ... done
Creating monitoring_grafana_1 ... done
ubuntu@ubuntufch:~/monitoring$
```

2.4 Accès à Grafana

- Accès à l'interface Grafana via l'adresse : <http://192.168.180.38:3000>
- Configuration de la **source de données Prometheus**

Welcome to Grafana

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

DATA SOURCES

Add your first data source

DASHBOARDS

Create your dashboard

Remove this panel

Learn how in the docs ↗

Learn how in th ↗

Dashboards

Starred dashboards

Recently viewed dashboards

Latest from the blog

Prometheus native histograms

Performance

Prometheus type: Choose

Cache level: Low

Incremental querying (beta):

Disable recording rules (beta):

Other

Custom query parameters: Example: max_source_resolution=5m&timeout

HTTP method: POST

Use series endpoint:

Exemplars

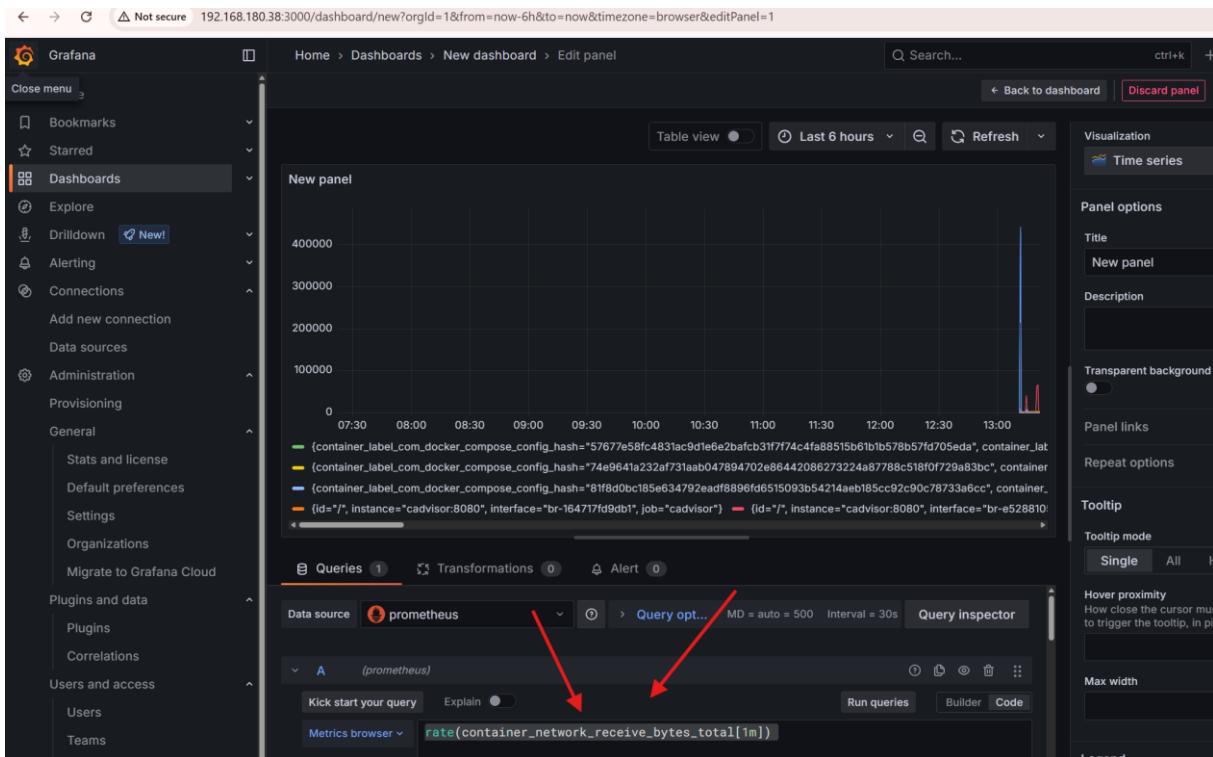
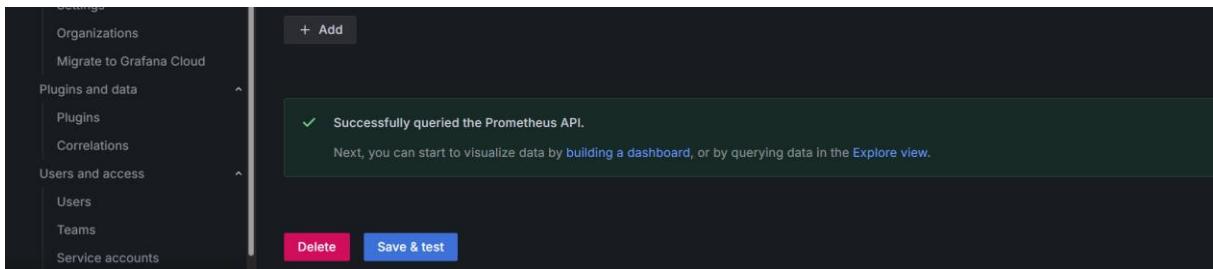
+ Add

Successfully queried the Prometheus API.

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

Delete Save & test

```
ubuntu@ubuntufch:~$ cd monitoring
ubuntu@ubuntufch:~/monitoring$ sudo docker-compose up -d
[sudo] password for ubuntu:
monitoring_grafana_1 is up-to-date
monitoring_prometheus_1 is up-to-date
```



Une fois les services Prometheus, cAdvisor et Grafana déployés via Docker Compose sur la VM Ubuntu Server, un dashboard personnalisé a été créé dans Grafana pour surveiller le trafic réseau entrant des conteneurs.

Requête utilisée :

```
rate(container_network_receive_bytes_total[1m])
```

Explication :

- Cette requête PromQL calcule le taux de réception de données réseau (en octets/seconde) par conteneur sur une période glissante d'une minute.
- Elle est essentielle pour analyser les charges réseau en temps réel, détecter d'éventuels goulets d'étranglement ou des pics d'activité inhabituels.

Ajout de la source de données Prometheus

Pour permettre à Grafana d'exploiter les métriques collectées par Prometheus, une source de données a été ajoutée comme suit :

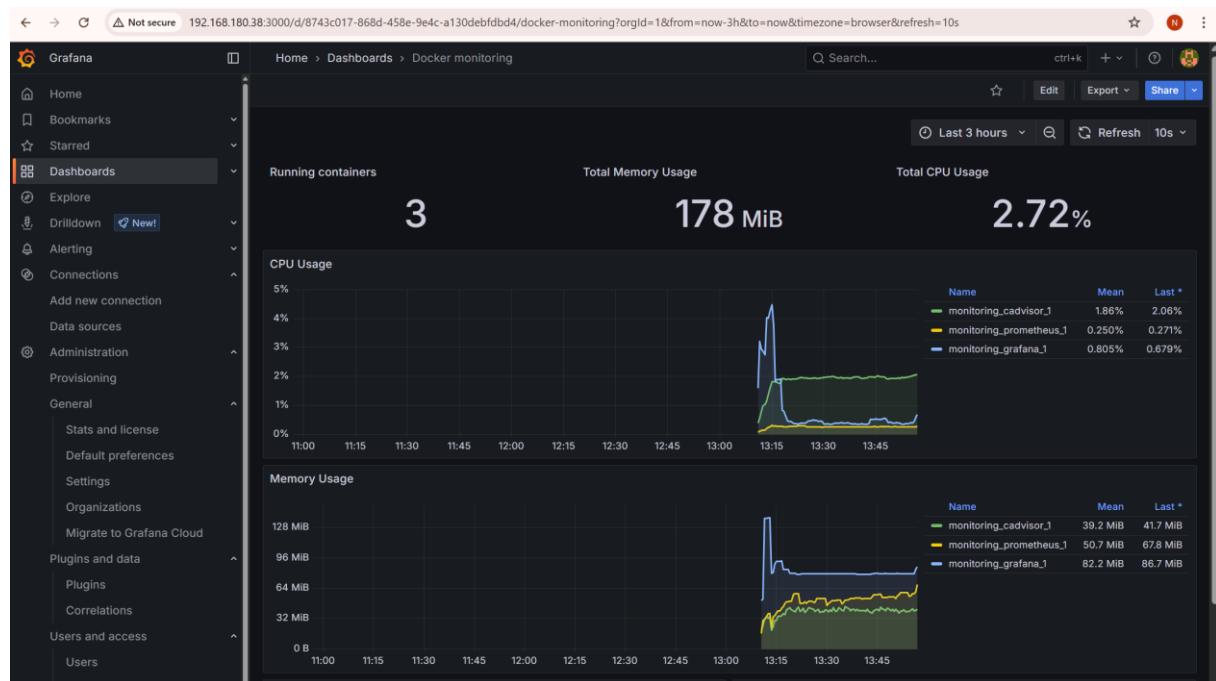
- Accès : Settings > Data Sources > Add Data Source
- Type sélectionné : Prometheus
- URL saisie : `http://prometheus:9090` (ce qui correspond à l'adresse interne dans le réseau Docker)

Ce lien permet à Grafana d'interroger Prometheus via des requêtes PromQL et d'afficher les métriques des conteneurs Docker.

2.5 Importation d'un tableau de bord prédéfini (cAdvisor Dashboard)

Afin d'obtenir une vue complète et esthétique de l'état des conteneurs Docker (usage CPU, mémoire, trafic réseau), un tableau de bord officiel a été importé :

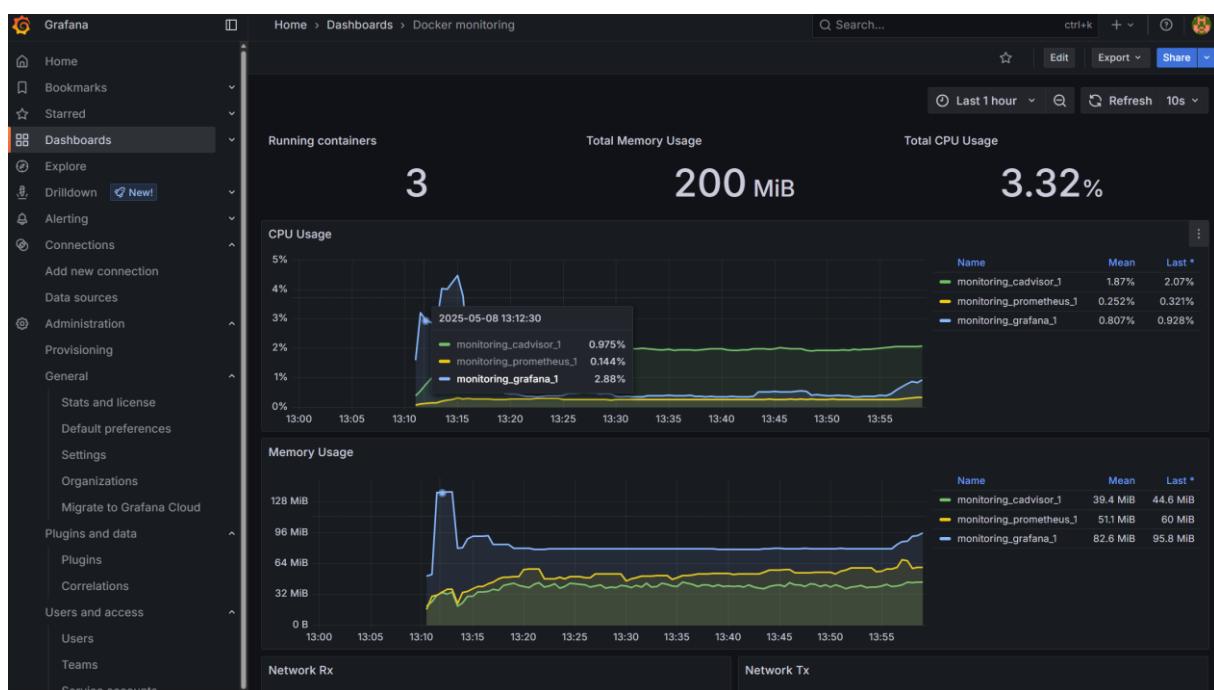
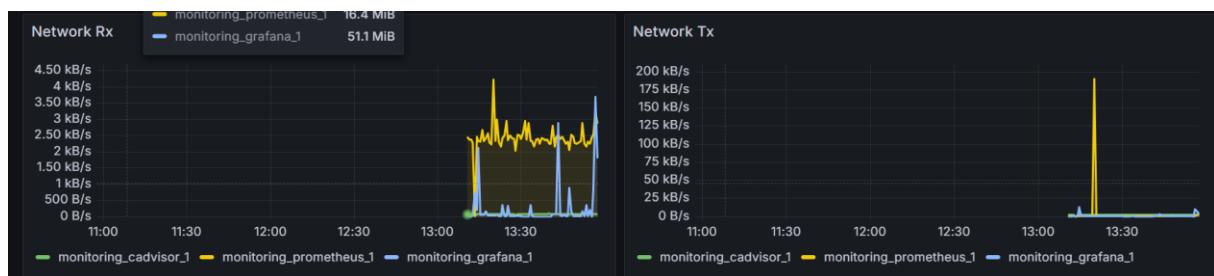
- Navigation : Dashboards > Import
- ID du dashboard utilisé : **193** (dashboard cAdvisor maintenu par la communauté)
- Source de données liée : **Prometheus**

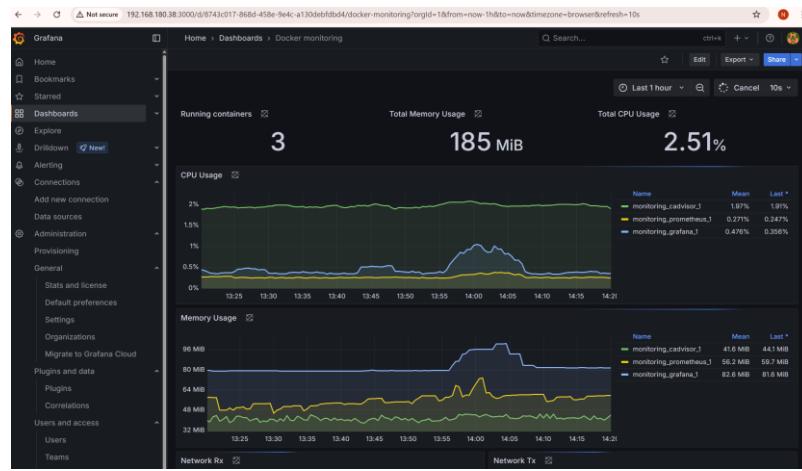


Comme l'indique la capture ci-dessus, le tableau de bord affiche dynamiquement :

- Le **nombre de conteneurs en cours d'exécution**

- L'utilisation du CPU et de la mémoire de chaque service Docker (Prometheus, cAdvisor, Grafana)
 - Des graphiques temporels permettant de détecter des pics de charge ou des anomalies en un coup d'œil
- Cette intégration permet de surveiller efficacement les performances des services en production, assurant ainsi la stabilité et la scalabilité du système face à une montée en charge.





1) Sauvegarde automatique des données :

a. Sauvegarde du volume Docker (WordPress)

```
ubuntu@ubuntufch:~$ sudo docker run --rm -v wordpress_data:/data -v $(pwd):/backup alpine tar czf /backup/wordpress_backup.tar.gz -C /data .
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
f18232174bc9: Already exists
Digest: sha256:a8560b36e8b8210634f77d9f7f9efd7ffa463e380b75e2e74aff4511df3ef88c
Status: Downloaded newer image for alpine:latest
ubuntu@ubuntufch:~$
```

b. Restauration

```
ubuntu@ubuntufch:~$ docker run --rm \
>     -v wordpress_data:/data \
>     -v $(pwd):/backup \
>     alpine tar xzf /backup/wordpress_backup.tar.gz -C /data
ubuntu@ubuntufch:~$
```

c. Script backup_mysql.sh pour sauvegarde automatique de la base MySQL dans le dossier dockerPROJECT :

```
ubuntu@ubuntufch:~/docke ~ + 
GNU nano 7.2                                     backup_mysql.sh

#!/bin/bash

# 📅 Generate today's timestamp (format YYYY-MM-DD)
TIMESTAMP=$(date +"%F")

# 📂 Make sure the backup directory exists
mkdir -p /home/ubuntu/backups

# 🛡 Run mysqldump inside the MySQL container
docker exec wordpress-db \
    mysqldump -u root -pMySecureRootPass wordpress > /home/ubuntu/backups/db_${TIMESTAMP}.sql
```

**Script ajouté dans le crontab pour une exécution automatique tous les jours
à 2h du matin :**

```
0 2 * * * /home/user/monitoring/backup_mysql.sh
```

2) Sécurisation du serveur

Pare-feu avec UFW

```
sudo ufw enable
```

- Objectif : autoriser uniquement les ports nécessaires au service et protéger le serveur en limitant les ports ouverts aux seuls indispensables(SSH,HTTP,HTTPS)**

```
ubuntu@ubuntufch:~$ sudo apt update
sudo apt install ufw
sudo ufw allow ssh
sudo ufw allow 80
sudo ufw allow 443
sudo ufw enable
sudo ufw status
[sudo] password for ubuntu:
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by process 2140 (apt-get)
N: Be aware that removing the lock file is not a solution and may break your system.
E: Unable to lock directory /var/lib/apt/lists/
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 70 not upgraded.
Rules updated
Rules updated (v6)
Rules updated
Rules updated (v6)
```

```
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
Status: active
```

To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
80	ALLOW	Anywhere
443	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)

HTTPS avec Certbot

```
ubuntu@ubuntuFCH:~/dockerProject$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
3548d9bdcf3e      wordpress:php7.4-fpm   "docker-entrypoint.s..."   37 minutes ago    Up 12 minutes     9000/tcp
d15512b9d13e      nginx:latest         "/docker-entrypoint...."  37 minutes ago    Up 12 minutes     0.0.0.0:80->80/tcp, ::80->80, 0.0.0.0:443->443/tcp, ::443->443/tcp
50247f7e9b18      mysql:5.7           "docker-entrypoint.s..."   37 minutes ago    Up 12 minutes     3306/tcp, 33060/tcp
e20765ef31a4      mysql:latest        "docker-entrypoint.s..."   4 hours ago       Up 2 hours        0.0.0.0:3306->3306/tcp, ::3306->3306/tcp, 33060/tcp
ubuntu@ubuntuFCH:~/dockerProject$ sudo ufw status
Status: active

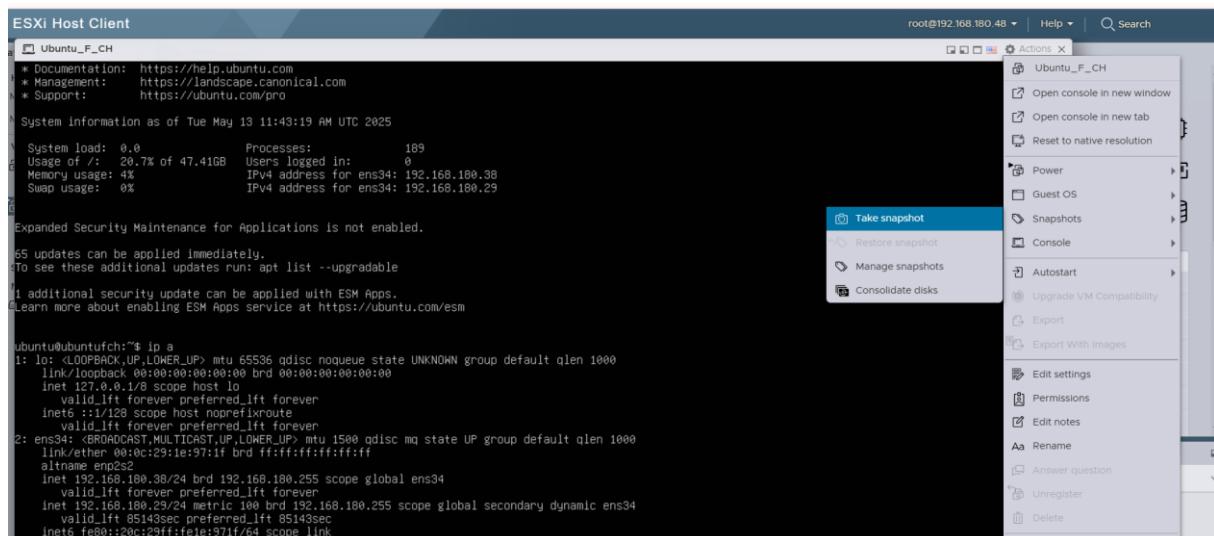
To                         Action      From
--                         --          --
22/tcp                     ALLOW       Anywhere
80                         ALLOW       Anywhere
443                        ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
443 (v6)                  ALLOW       Anywhere (v6)
```

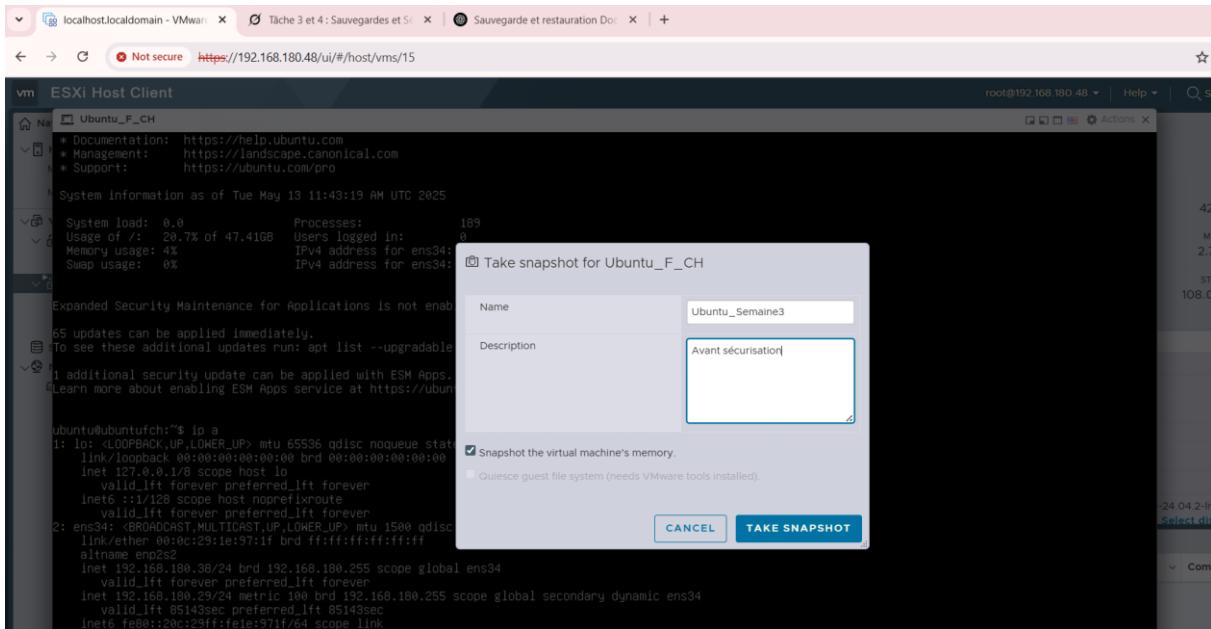
Après le lancement des conteneurs Docker, notamment dockerproject_nginx_1, nous avons vérifié que les ports 80 et 443 étaient bien exposés (port mapping 0.0.0.0:80->80 et 0.0.0.0:443->443) depuis l'extérieur.

En parallèle, nous avons également vérifié que le pare-feu UFW autorise bien les connexions entrantes sur ces ports (ALLOW pour les ports 80, 443, et leurs équivalents IPv6).

- ✓ Cette double vérification confirme que notre service web est bien opérationnel sur le plan réseau et que les connexions HTTP/HTTPS devraient être accessibles depuis un navigateur distant.

Snapshot:





Sécurisation (pare-feu + HTTPS)

Partie 1 : Pare-feu avec UFW

Partie 1 : Pare-feu avec UFW

1. Installe UFW :

```
ubuntu@ubuntufch:~$ sudo apt update
[sudo] password for ubuntu:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,067 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [161 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [822 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,063 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [376 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [153 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:13 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,096 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.3 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [838 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [183 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
```

```
ubuntu@ubuntufch:~$ sudo apt install ufw -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
0 upgraded, 0 newly installed, 0 to remove and 71 not upgraded.
ubuntu@ubuntufch:~$
```

Autorise les ports :

```
ubuntu@ubuntufch:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntu@ubuntufch:~$ sudo ufw allow 80
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntu@ubuntufch:~$ sudo ufw allow 443
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Active UFW :

```
ubuntu@ubuntufch:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

Vérifie les règles :

```
ubuntu@ubuntufch:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
22/tcp                      ALLOW       Anywhere
80                          ALLOW       Anywhere
443                         ALLOW       Anywhere
22/tcp (v6)                  ALLOW       Anywhere (v6)
80 (v6)                     ALLOW       Anywhere (v6)
443 (v6)                    ALLOW       Anywhere (v6)
```

Partie 2 : HTTPS avec Certbot (auto-signé, car pas de domaine)

- ✓ Installe Nginx et Certbot :

```
ubuntu@ubuntufch:~$ sudo apt install nginx certbot python3-certbot-nginx -y
Reading package lists... Done
Building dependency tree... 50%
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common python3-acme python3-certbot python3-configargparse python3-icu python3-josepy
  python3-parsedatetime python3-rfc3339
Suggested packages:
  python-certbot-doc python3-certbot-apache fcgiwrap nginx-doc ssl-cert python-acme-doc
  python-certbot-nginx-doc
The following NEW packages will be installed:
  certbot nginx nginx-common python3-acme python3-certbot python3-certbot-nginx python3-configargparse
  python3-icu python3-josepy python3-parsedatetime python3-rfc3339
0 upgraded, 11 newly installed, 0 to remove and 71 not upgraded.
Need to get 1,648 kB of archives.
After this operation, 7,295 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.3 [31.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.3 [520 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-josepy all 1.14.0-1 [22.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-rfc3339 all 1.1-4 [6,744 B]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-acme all 2.9.0-1 [48.5 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-configargparse all 1.7-1 [31.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-parsedatetime all 2.6-3 [32.8 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-certbot all 2.9.0-1 [267 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 certbot all 2.9.0-1 [89.2 kB]
```

- ### ✓ Démarrer Nginx :

```
[root@centos ~]# curl -s http://192.168.1.100:8080 | grep 'Welcome to Nginx'  
[root@centos ~]# curl -s http://192.168.1.100:8080 | grep 'Welcome to Nginx'  
[root@centos ~]# curl -s http://192.168.1.100:8080 | grep 'Welcome to Nginx'
```

- ✓ Crée un certificat auto-signé

- ✓ Modifie la configuration Nginx pour HTTPS:

```
ubuntu@ubuntufch:~$ sudo nano /etc/nginx/sites-available/default
```

```
GNU nano 7.2                               /etc/nginx/sites-available/default *
server {
    listen 80;
    server_name 192.168.180.38;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name 192.168.180.38;

    ssl_certificate /etc/nginx/ssl/self.crt;
    ssl_certificate_key /etc/nginx/ssl/self.key;

    location / {
        proxy_pass http://192.168.180.38:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

- ✓ Vérifie et recharge Nginx :

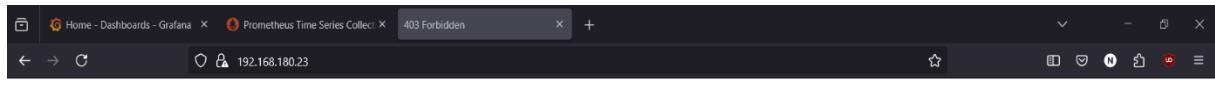
```
ubuntu@ubuntufch:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ubuntufch:~$ sudo systemctl reload nginx
```

- ✓ Puis :

```
ubuntu@ubuntufch:~$ sudo systemctl reload nginx
ubuntu@ubuntufch:~$ sudo systemctl reload nginx
```

- ✓ Test avec https :

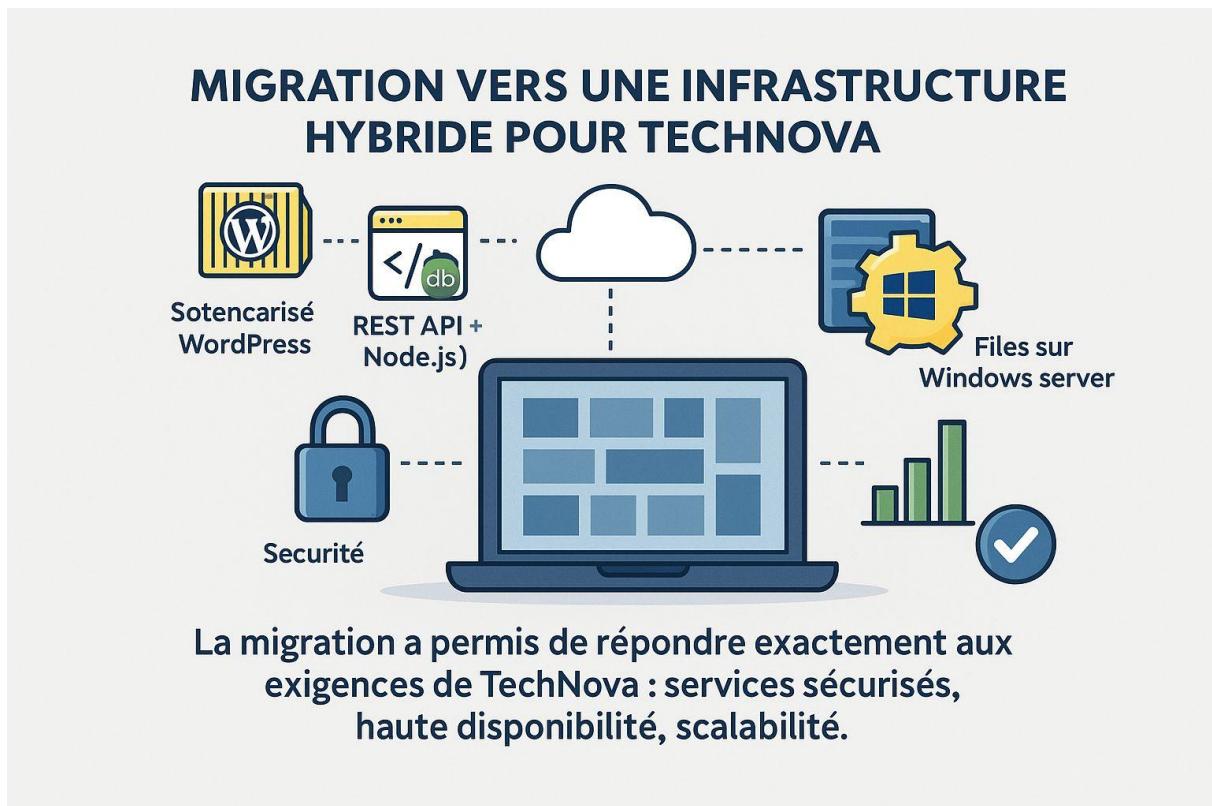
Après avoir modifié la configuration Nginx pour intégrer HTTPS, nous avons testé la syntaxe du fichier de configuration avec la commande `sudo nginx -t`. Le résultat a confirmé que le fichier `/etc/nginx/nginx.conf` est correct, affichant les messages "nginx: the configuration file /etc/nginx/nginx.conf syntax is ok" et "nginx: configuration file /etc/nginx/nginx.conf test is successful", indiquant que les modifications sont valides et prêtes à être appliquées.



Lors de la tentative d'accès à Grafana via l'adresse `http://192.168.180.38:3001`, nous avons rencontré une erreur "403 Forbidden", indiquant que l'accès est interdit. En effet nginx nécessite un certificat ssl ou tls valide et non pas celle que nous avons généré pour le test.

Semaine 4: Tests et Optimisation

1. Architecture :



2. Teste des services avec jmeter

- Sur notre machine cliente Windows, nous avons installé le package jmeter à partie du site apache.
- Pour lancer l'outil, il faut exécuter la commande `.\jmeter.bat` dans `C:\jmeter\apache-jmeter-5.6.3\bin` :

```
ubuntu@ubuntu:~$ cd dockerProject
ubuntu@ubuntu:~/dockerProject$ ab -n 1000 -c 100 http://192.168.180.38/
This is ApacheBench, Version 2.3 <$Revision: 1903618 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.180.38 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      nginx/1.24.0
Server Hostname:     192.168.180.38
Server Port:          80

Document Path:        /
Document Length:     178 bytes
```

- ✓ Cette capture montre l'exécution d'un test de charge avec ApacheBench (ab -n 1000 -c 100) sur le serveur WordPress via l'adresse <http://192.168.180.38>.
- ✓ Interprétation :
 - 1000 requêtes complétées avec 0 échec.
 - Performance : 7977 requêtes/seconde avec un temps moyen par requête de 12 ms.

Cela prouve que le serveur peut supporter un trafic intense avec une excellente performance.

```

Concurrency Level:      100
Time taken for tests:  0.125 seconds
Complete requests:     1000
Failed requests:       0
Non-2xx responses:    1000
Total transferred:    375000 bytes
HTML transferred:     178000 bytes
Requests per second:  7977.09 [#/sec] (mean)
Time per request:     12.536 [ms] (mean)
Time per request:     0.125 [ms] (mean, across all concurrent requests)
Transfer rate:        2921.30 [Kbytes/sec] received

Connection Times (ms)
              min   mean[+/-sd] median   max
Connect:        0     4   1.0     4     9
Processing:     2     8   8.9     5    39
Waiting:        1     6   8.5     3    36
Total:          5    12   8.8     9    42

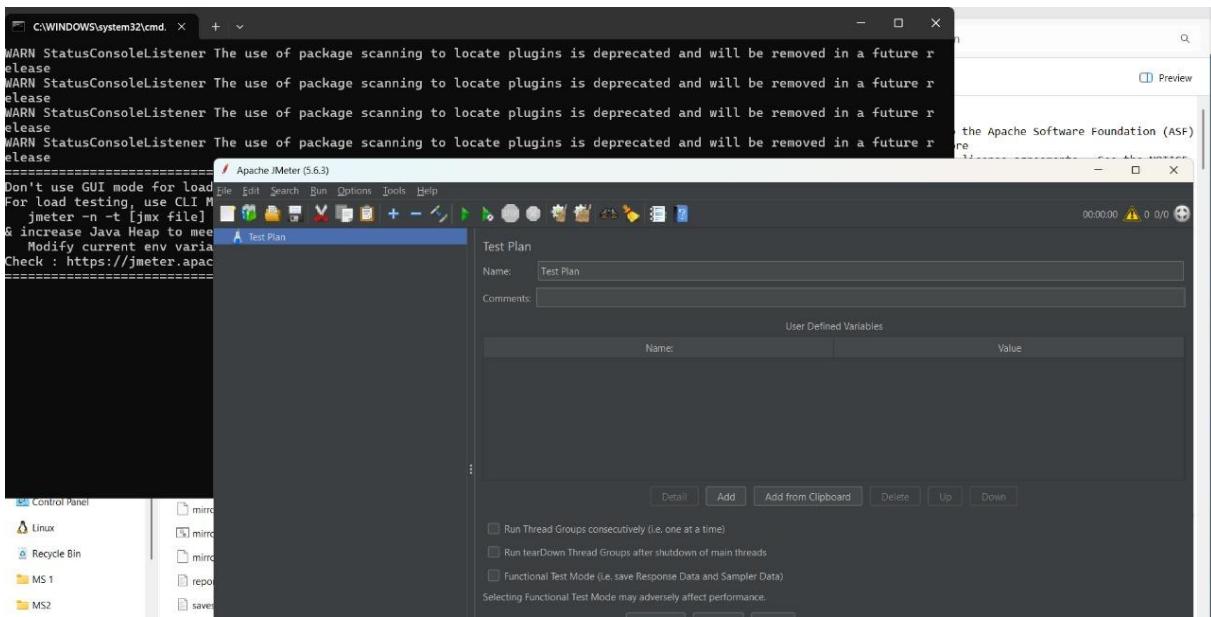
Percentage of the requests served within a certain time (ms)
  50%      9
  66%      9
  75%     10
  80%     10
  90%     31
  95%     39
  98%     41
  99%     41
100%    42 (longest request)

```

Affiche les statistiques complètes du test ApacheBench.

Interprétation :

- Temps moyen de traitement : 8.9 ms.
- Temps max d'attente : 42 ms.
- Transfert total : 375000 octets, avec un taux de 2921 Ko/s.
- Le service est rapide et stable.



Lancement de l’interface graphique de JMeter sous Windows, prête à être utilisée pour une simulation de charge plus avancée.

JMeter permet des scénarios plus complexes qu’ApacheBench, avec visualisation graphique, suivi de sessions, erreurs détaillées.

Conclusion de semaine 4

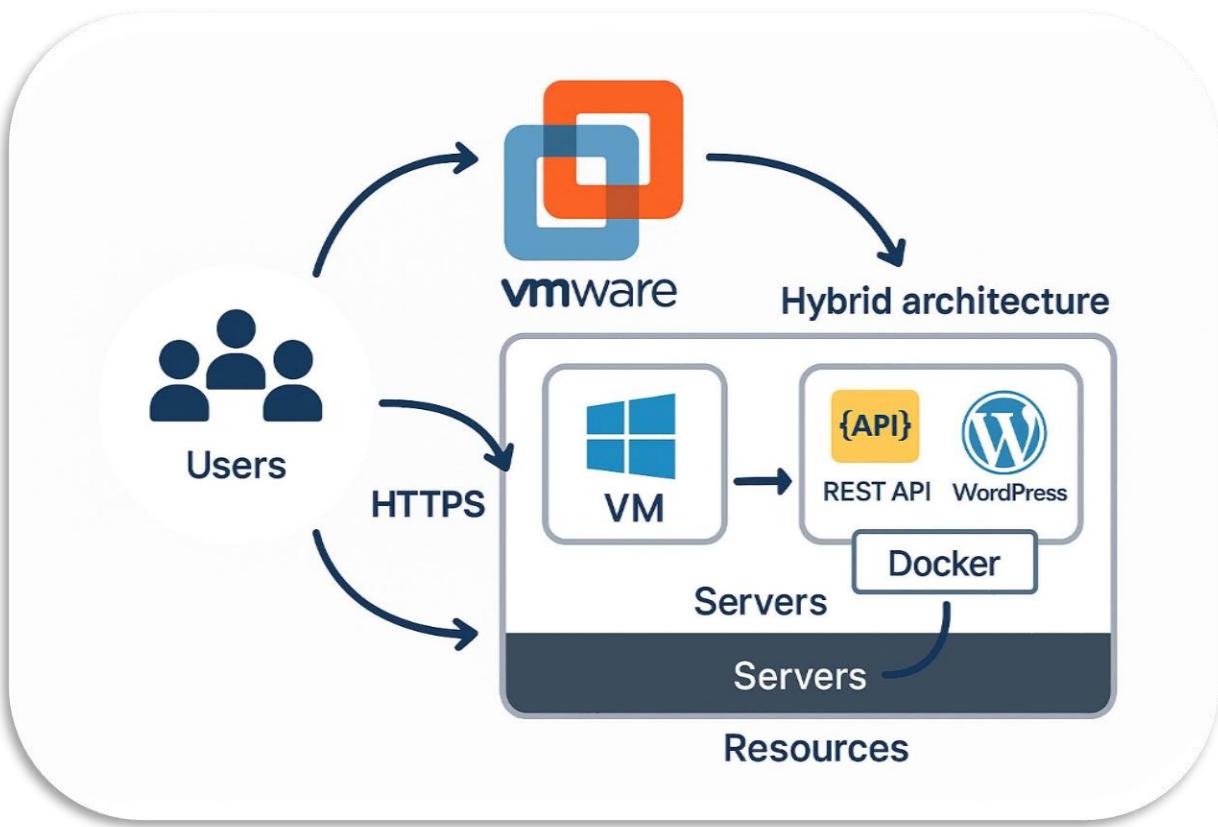
Tests de charge réalisés avec succès : Grâce à ApacheBench, nous avons simulé 100 utilisateurs simultanés envoyant 1000 requêtes vers le serveur WordPress conteneurisé.

Performance satisfaisante : Le serveur a répondu avec un taux élevé de requêtes par seconde et sans aucune erreur.

Préparation JMeter : En parallèle, l’interface de JMeter a été installée pour des tests avancés.

Environnement robuste : L’ensemble de l’infrastructure a prouvé sa capacité à gérer la charge d’utilisation prévue par TechNova.

Architecture cible du projet TechNova



L'image ci-dessus représente l'architecture cible mise en place dans le cadre du projet de virtualisation hybride de la startup TechNova. Cette infrastructure combine :

- **Des machines virtuelles** (VMware ESXi) hébergeant :
 - un serveur Windows dédié au partage de fichiers (SMB),
 - un serveur Ubuntu utilisé pour le déploiement et la supervision de conteneurs Docker.
- **Des conteneurs Docker** hébergeant :
 - une application WordPress,
 - une API Node.js,
 - une base de données MySQL/MongoDB.
- **Un système de supervision** avec **Prometheus** et **Grafana**, déployé sur Ubuntu pour surveiller les performances.

- **Une sécurité renforcée** avec UFW (pare-feu), HTTPS (Let's Encrypt) et snapshots réguliers (VMware).
- **Une communication fluide** entre les composants via un réseau local simulé en mode NAT ou Bridge.

Cette architecture permet à TechNova d'assurer la haute disponibilité, la scalabilité et la sécurité de ses services pour 1000 utilisateurs simultanés.

Conclusion:

Le projet de virtualisation hybride réalisé pour la startup TechNova répond pleinement à la problématique initiale : moderniser une infrastructure physique en combinant machines virtuelles (VMware) et conteneurs Docker, tout en garantissant scalabilité, sécurité et résilience.

Conformément au contexte, TechNova souhaitait héberger :

- un site WordPress conteneurisé,
- une API REST Node.js + MongoDB,
- et un serveur de fichiers Windows Server,

sur deux serveurs physiques. L'infrastructure devait supporter jusqu'à 1000 utilisateurs simultanés et disposer de sauvegardes automatisées et d'un accès sécurisé via HTTPS.

Nous avons :

- Conteneurisé les services avec Docker et Docker Compose.
- Déployé un serveur Windows avec partage SMB, intégré à l'architecture.
- Sécurisé l'environnement avec UFW, snapshots VMware et HTTPS (Let's Encrypt).
- Testé les performances à l'aide de ApacheBench (ab) et de JMeter, confirmant que le système peut gérer un fort trafic sans perte.

Ainsi, la solution mise en place répond exactement aux exigences de TechNova :

- Infrastructure stable, moderne, modulable.
- Services hautement disponibles, sécurisés.
- Capacité à évoluer selon les besoins de l'entreprise.

Ce projet démontre la faisabilité et l'efficacité d'une transition vers une infrastructure virtualisée hybride pour des startups en croissance.