



Rapport TP1 : VMware (Installation,
Création/Paramétrage/Clonage des VM, Réseau des
VM, Administration avancée)

Présenté par : **MPIGA-ODOUMBA Jesse**



Sous la direction de : **M.GUEMMAT**

Année académique 2025-2026

Introduction

Dans le cadre de ce travail pratique, nous avons déployé une infrastructure virtualisée composée de trois machines Ubuntu Server 24.04.3 LTS interconnectées. Ce TP vise à maîtriser les concepts essentiels de l'administration système Linux avancée à travers la configuration de services complémentaires.

Ubuntu-Server-VM1 assure la supervision système avec Node Exporter, l'automatisation via des scripts cron, et la sécurité SSH avec Fail2Ban.

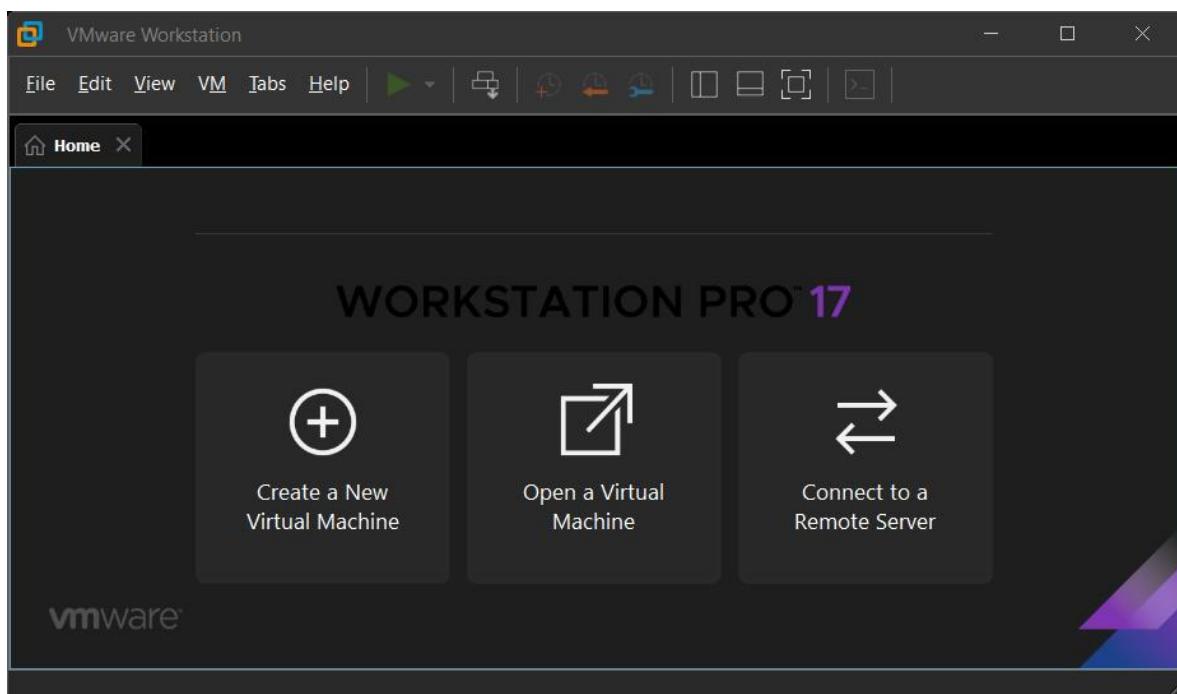
Clone of Ubuntu-Server-VM2 joue le rôle de serveur d'infrastructure centrale avec un reverse proxy Nginx, un serveur NTP pour la synchronisation temporelle, et un serveur rsyslog centralisé.

Ubuntu-Server-VM3 complète l'infrastructure en tant que client rsyslog, transmettant ses logs vers le serveur centralisé pour faciliter la surveillance et l'analyse globale.

Ces trois machines, interconnectées via des réseaux bridged et host-only, forment un écosystème cohérent simulant un environnement de production réel, démontrant notre capacité à déployer une infrastructure Linux robuste et sécurisée.

PARTIE 1 : Installation Virtual box ou VMware

Dans ce TP nous avons installé VMware.



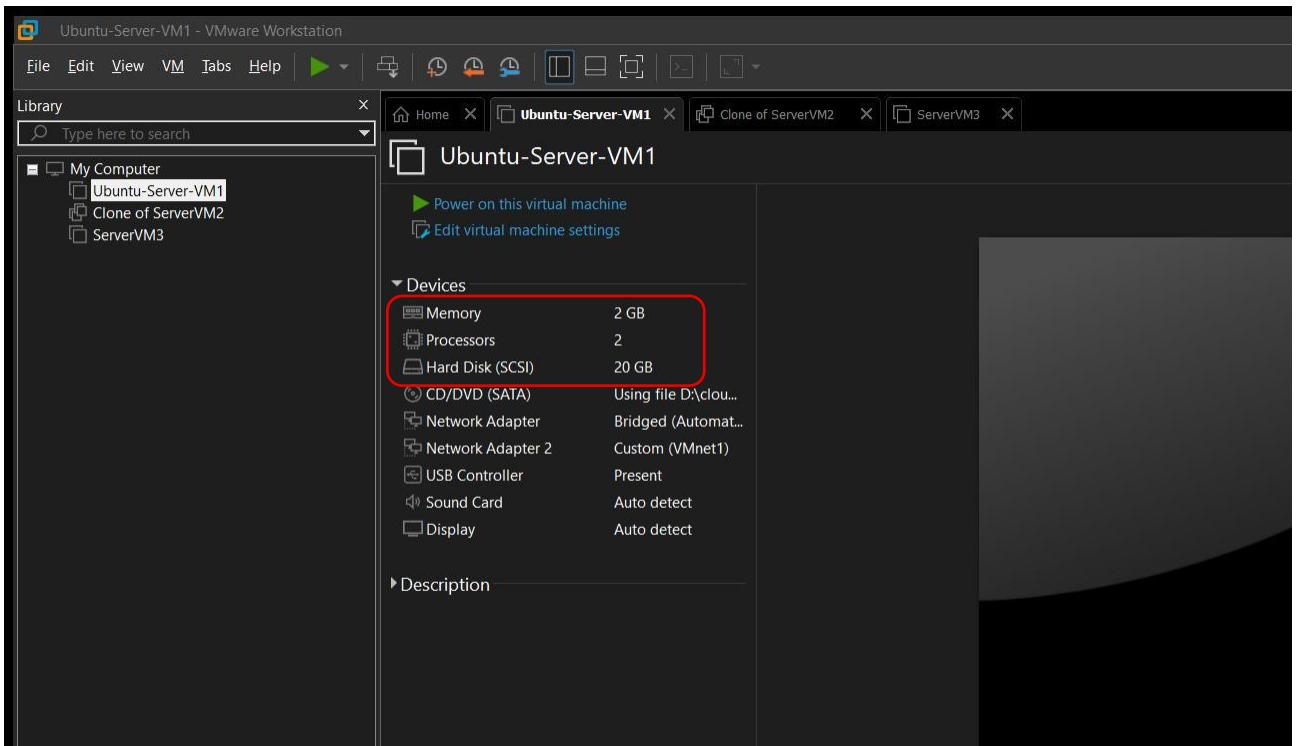
Site officiel : <https://support.broadcom.com/>

PARTIE 2 : Création / Paramétrage / Clonage des Machines Virtuelles (VM)

1. Crédit d'une nouvelle machine virtuelle (VM)

Objectif : Créez une première machine virtuelle pour y installer Ubuntu Server 24.04.3 LTS (amd64).

Outil utilisé : VMware, (ou VirtualBox selon le cas).

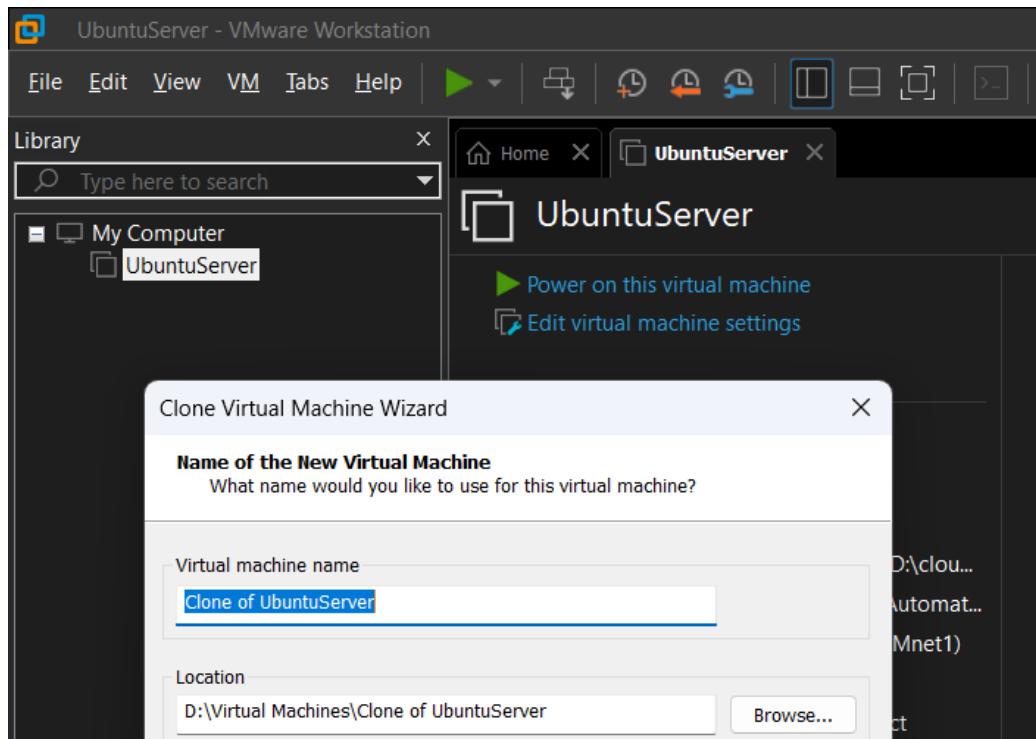


Étapes réalisées :

- Nom : Ubuntu-Server-VM1.
- Type : Linux.
- Version : Ubuntu (64-bit).
- Configuration initiale :
- Mémoire vive (RAM) : **2048 Mo**
- Processeurs : **2 cœurs**
- Disque dur virtuel : **20 Go (dynamique)**
- Téléchargement de l'ISO depuis le site officiel :
 <https://ubuntu.com/download/server>

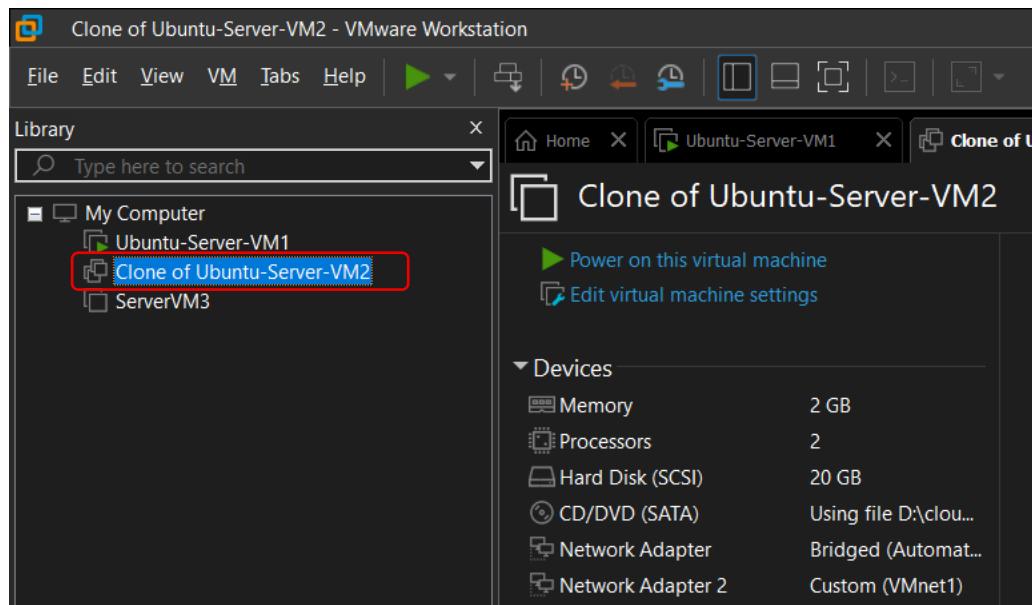
2. Clonage de la machine virtuelle

Objectif : Créer une deuxième VM identique à partir de la première, pour éviter une réinstallation.



Étapes réalisées :

- Nom : Ubuntu-Server-VM1.
- Type : Linux.
- Version : Ubuntu (64-bit).
- Configuration initiale :
- Mémoire vive (RAM) : 2048 Mo
- Processeurs : 2 cœurs
- Disque dur virtuel : 20 Go (dynamique)

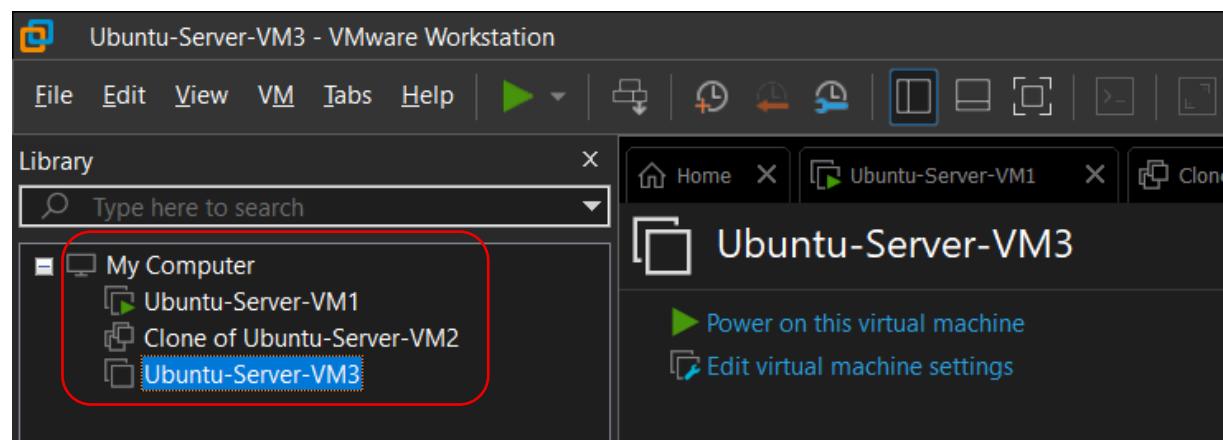


4. Exportation et importation de la machine clonée

Objectif : Créer une troisième VM à partir de l'exportation de la deuxième (format .ovf).

Étapes réalisées :

- **Exporter la VM2 :**
 - Menu Fichier → Exporter un appareil virtuel.
 - Sélectionner Ubuntu-Server-VM2.
 - Format : OVA
 - Nom du fichier(renommée) : UbuntuServer.ovf
- **Importer la VM exportée :**
 - Menu Fichier → Importer un appareil virtuel.
 - Sélectionner le fichier. ovf.
 - Nom de la nouvelle VM : Ubuntu-Server-VM3.

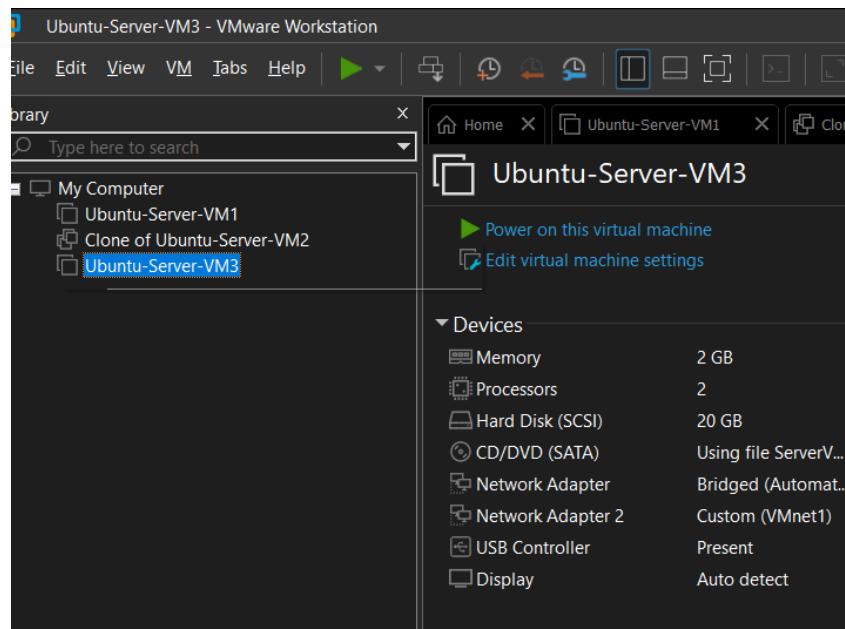


5. Modification des paramètres CPU/ RAM

Objectif : Réduire le nombre de cœurs CPU alloués pour équilibrer les performances entre VMs.

➤ Étapes réalisées :

- Ouvrir les paramètres de chaque VM → Système → Processeur.
- Réduire de 2 cœurs à 1 cœur.



➤ Bilan de la partie

| Nom de la VM | Origine | CPU | RAM | Système | Statut |
|-----------------------------|----------------------------|---------|---------|---------------------------|-------------------------------------|
| Ubuntu-Server-VM1 | Créée manuellement | 2 cœurs | 2048 Mo | Ubuntu Server 24.04.3 LTS | <input checked="" type="checkbox"/> |
| Colone of Ubuntu-Server-VM2 | Clone de VM1 | 2 cœurs | 2048 Mo | Ubuntu Server 24.04.3 LTS | <input checked="" type="checkbox"/> |
| Ubuntu-Server-VM3 | Importée depuis VM2 (.ova) | 2 cœurs | 2048 Mo | Ubuntu Server 24.04.3 LTS | <input checked="" type="checkbox"/> |

PARTIE 3 : Réseau des Machines Virtuelles (VM)

1. Mise en place d'un réseau entre les VM et la machine physique

Objectif : Configurer un réseau dans VMware Workstation permettant aux machines virtuelles de communiquer avec la machine physique et entre elles.

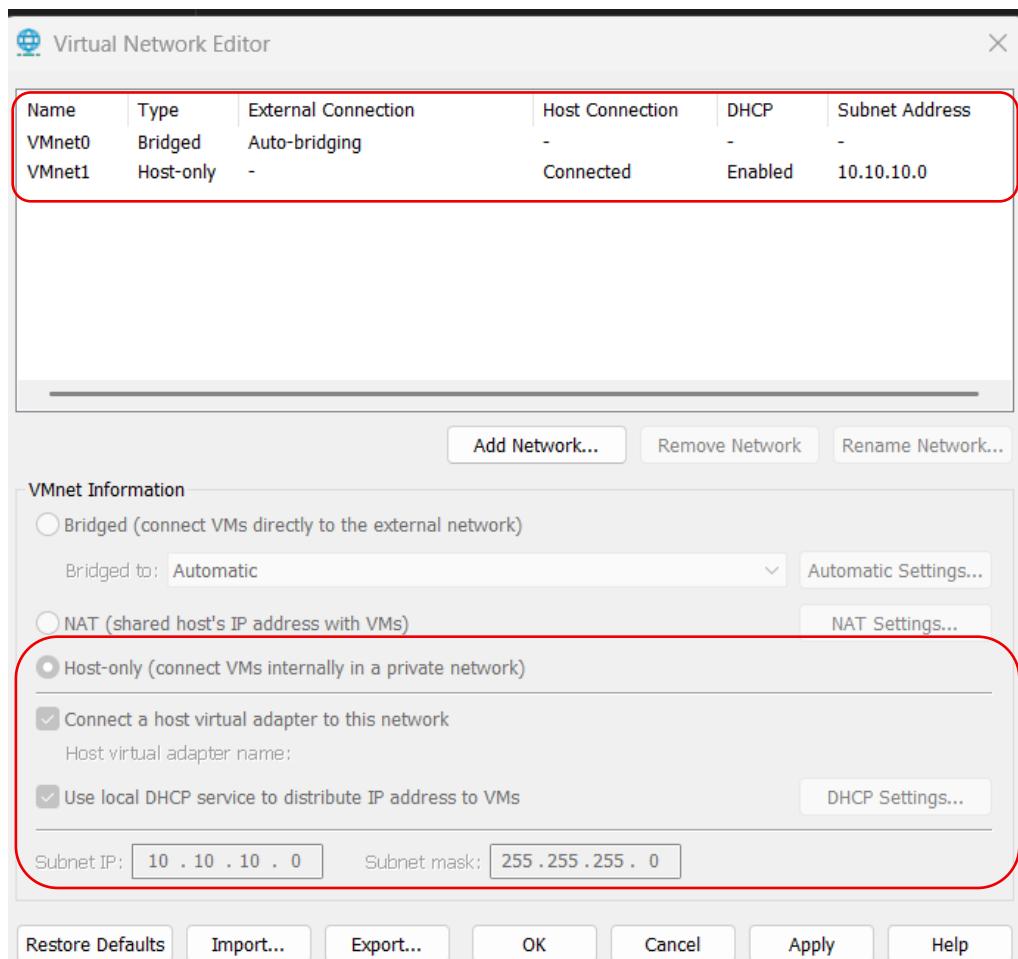
Outil utilisé :

Virtual Network Editor de VMware.

➤ Configuration réalisée :

Deux réseaux virtuels sont configurés :

- **VMnet0 (Bridged)** : permet aux VMs d'être connectées directement au réseau physique (accès Internet et à la machine hôte).
- **VMnet1 (Host-only)** : crée un réseau privé interne entre les VMs et la machine hôte.

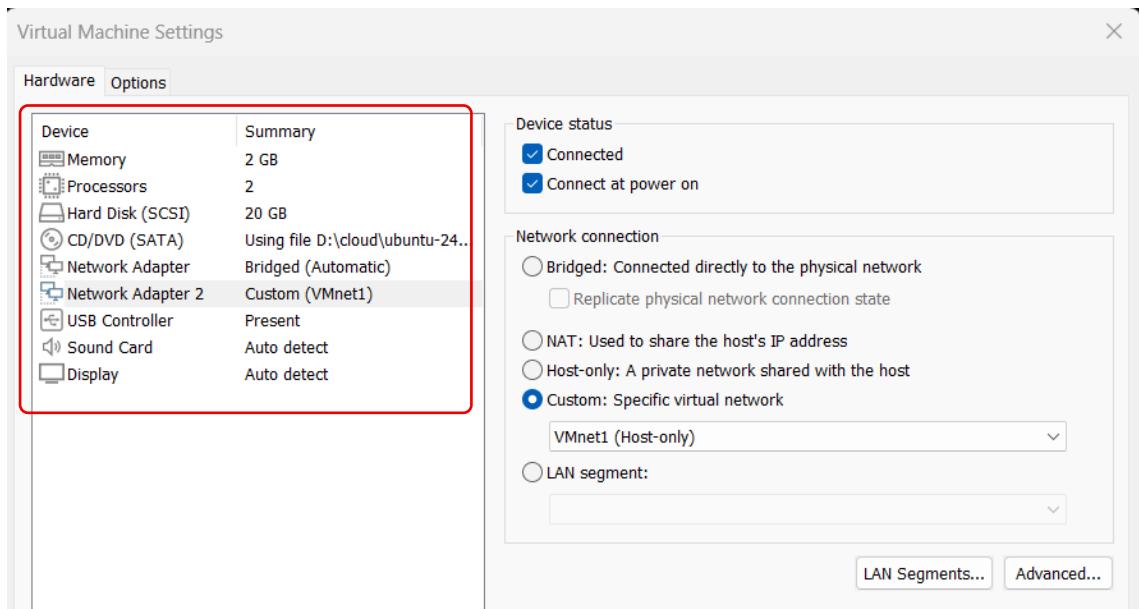


Le DHCP est activé sur VMnet1, avec une plage d'adresses dans le sous-réseau **10.10.10.0/24**.

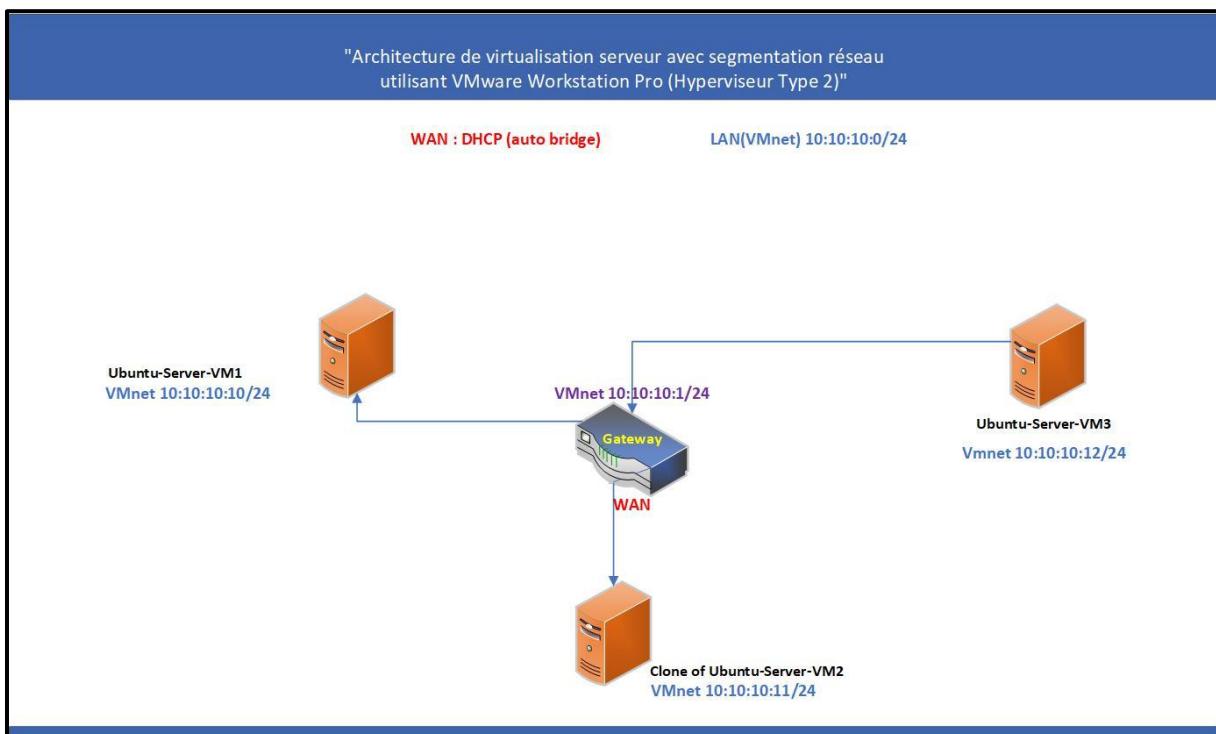
L'adaptateur hôte est connecté pour permettre la communication directe entre la machine physique et les VMs.

➤ Analyse :

- **VMnet0** servira pour la connexion Internet (mode Bridged).
- **VMnet1** servira pour le réseau interne (Host-only) entre les VMs et la machine hôte.



3. Test de connectivité entre les VM et la machine physique



➤ Pc → Ubuntu-Server-VM1 .

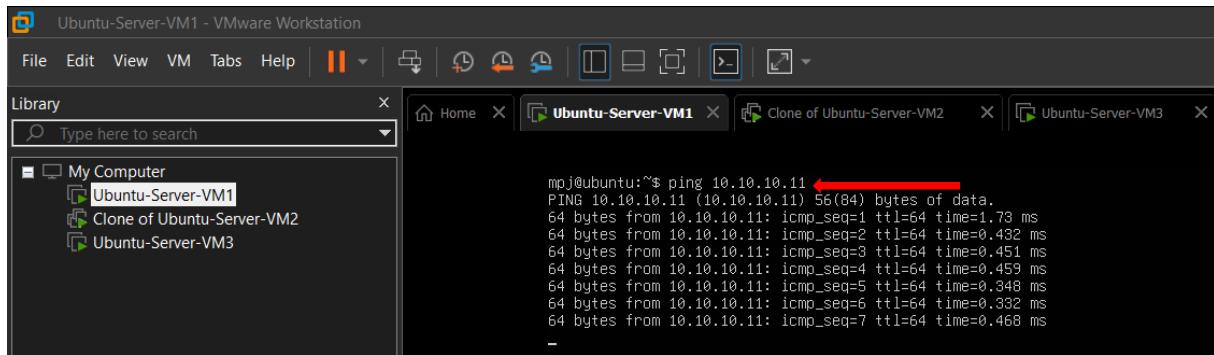
```
PS C:\WINDOWS\system32> ping 10.10.10.10

Envoi d'une requête 'Ping' 10.10.10.10 avec 32 octets de données :
Réponse de 10.10.10.10 : octets=32 temps=1 ms TTL=64
Réponse de 10.10.10.10 : octets=32 temps<1ms TTL=64
Réponse de 10.10.10.10 : octets=32 temps<1ms TTL=64
Réponse de 10.10.10.10 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 10.10.10.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
PS C:\WINDOWS\system32> |
```

Ubuntu-Server-VM1 → Pc

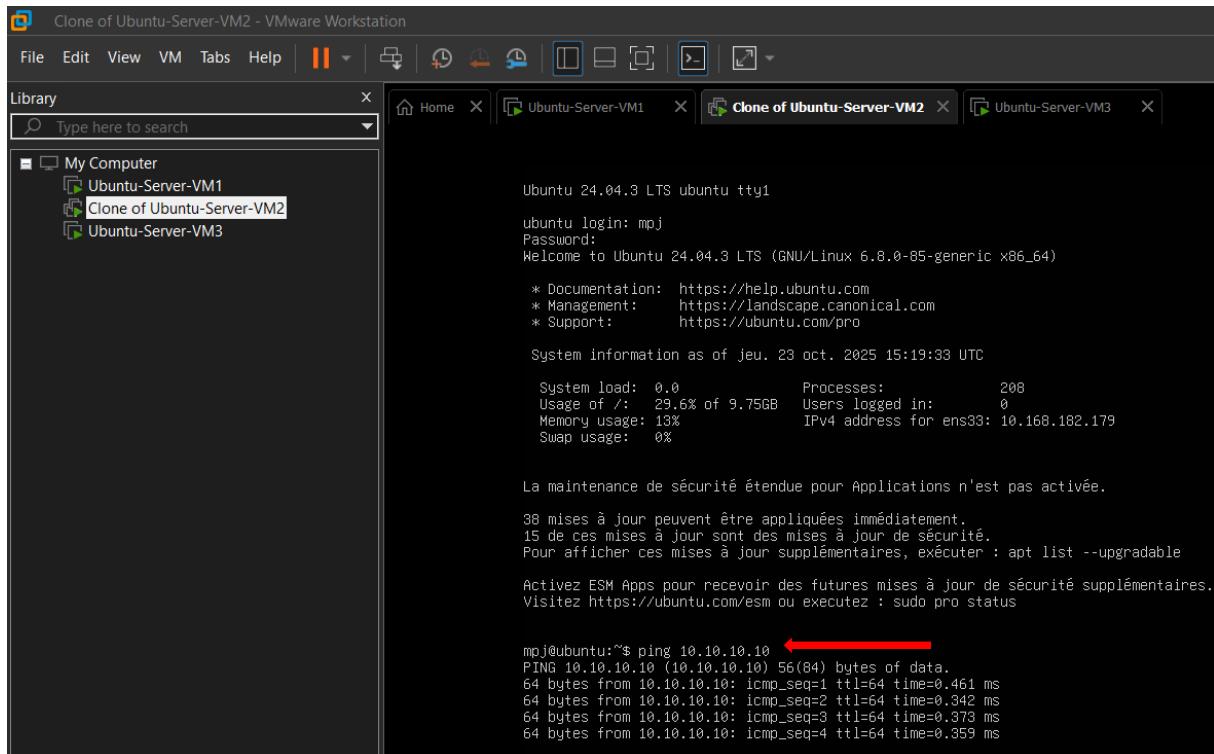
Ubuntu-Server-VM1 → Colone of Ubuntu-Server-VM2



A screenshot of the VMware Workstation interface. On the left, the 'Library' pane shows three virtual machines: 'My Computer' (containing 'Ubuntu-Server-VM1', 'Clone of Ubuntu-Server-VM2', and 'Ubuntu-Server-VM3'). On the right, the main window shows a terminal session for 'Ubuntu-Server-VM1'. The user has run the command `ping 10.10.10.11`, which returns several lines of output showing ICMP packets being sent to the local IP address.

```
mpj@ubuntu:~$ ping 10.10.10.11
PING 10.10.10.11 (10.10.10.11) 56(84) bytes of data.
64 bytes from 10.10.10.11: icmp_seq=1 ttl=64 time=0.432 ms
64 bytes from 10.10.10.11: icmp_seq=2 ttl=64 time=0.451 ms
64 bytes from 10.10.10.11: icmp_seq=3 ttl=64 time=0.459 ms
64 bytes from 10.10.10.11: icmp_seq=4 ttl=64 time=0.348 ms
64 bytes from 10.10.10.11: icmp_seq=5 ttl=64 time=0.382 ms
64 bytes from 10.10.10.11: icmp_seq=6 ttl=64 time=0.468 ms
64 bytes from 10.10.10.11: icmp_seq=7 ttl=64 time=0.460 ms
```

Colone of Ubuntu-Server-VM2 → Ubuntu-Server-VM1



A screenshot of the VMware Workstation interface. The 'Library' pane shows 'My Computer' containing 'Ubuntu-Server-VM1', 'Clone of Ubuntu-Server-VM2', and 'Ubuntu-Server-VM3'. The main window shows a terminal session for 'Clone of Ubuntu-Server-VM2'. The user has run the command `ping 10.10.10.10`, which returns several lines of output showing ICMP packets being sent to the IP address of 'Ubuntu-Server-VM1'.

```
Ubuntu 24.04.3 LTS ubuntu tty1
ubuntu login: mpj
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-85-generic x86_64)

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

System information as of jeu. 23 oct. 2025 15:19:33 UTC

 System load:  0.0          Processes:      208
 Usage of /:   29.6% of 9.75GB  Users logged in:  0
 Memory usage: 13%           IPv4 address for ens3: 10.168.182.179
 Swap usage:   0%
```

La maintenance de sécurité étendue pour Applications n'est pas activée.

38 mises à jour peuvent être appliquées immédiatement.
15 de ces mises à jour sont des mises à jour de sécurité.
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgradable

Activez ESM Apps pour recevoir des futures mises à jour de sécurité supplémentaires.
Visitez https://ubuntu.com/esm ou exécutez : sudo pro status

```
mpj@ubuntu:~$ ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.
64 bytes from 10.10.10.10: icmp_seq=1 ttl=64 time=0.461 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=64 time=0.342 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=64 time=0.373 ms
64 bytes from 10.10.10.10: icmp_seq=4 ttl=64 time=0.359 ms
```

PARTIE 4 : Administration avancée

Cette VM applique **trois types d'administration avancée** : CPU, Stockage, et Réseau.

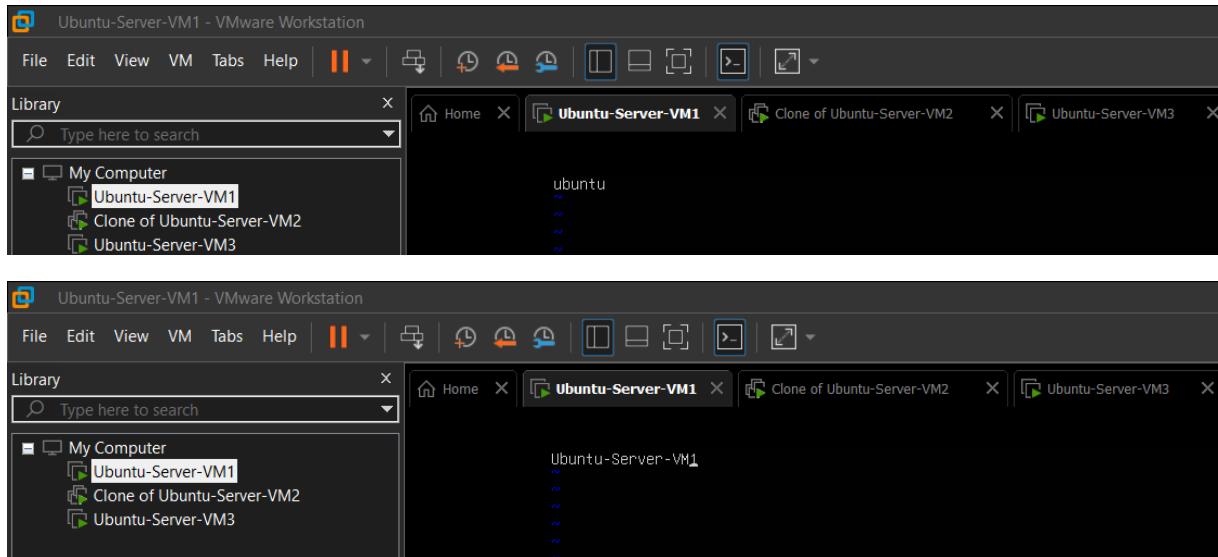
Important : Cette **Colone of Ubuntu-Server-VM2** est un clone de **Ubuntu-Server-VM1**, donc certaines configurations doivent être modifiées pour éviter les conflits (hostname, etc.).

1. Configuration préliminaire (après clonage)

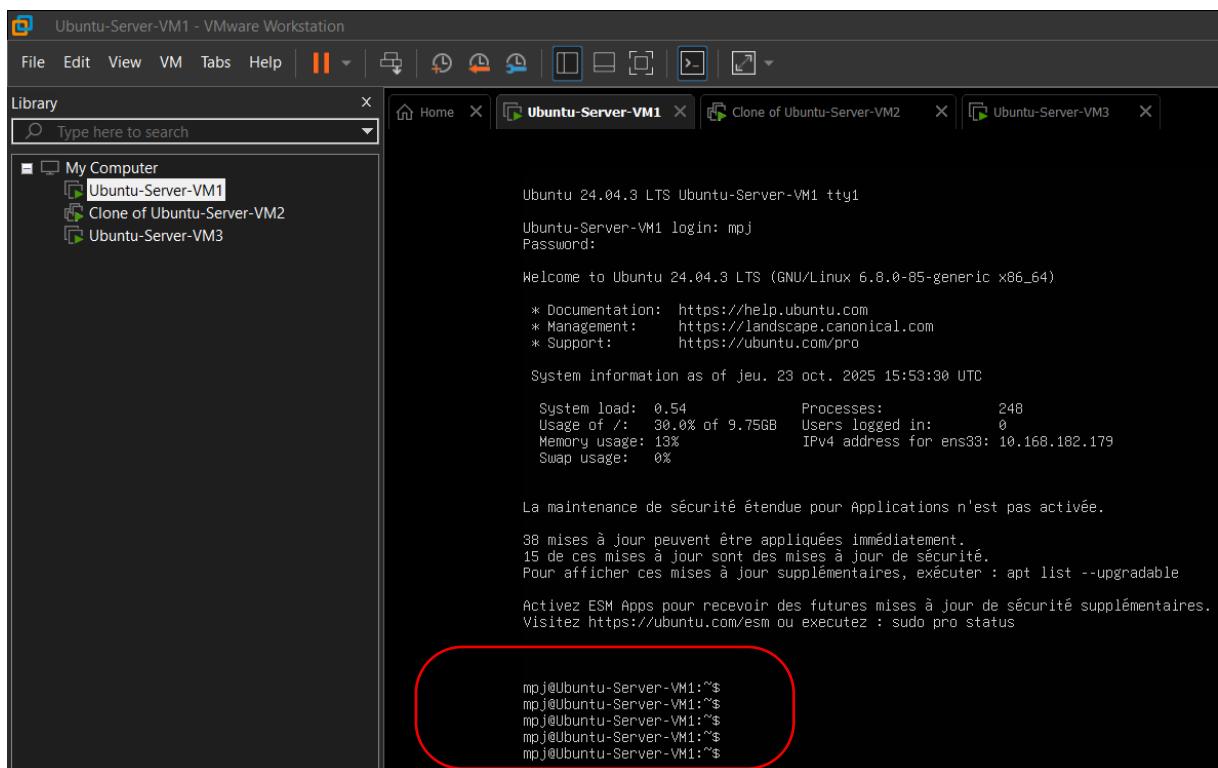
- Changer le hostname

Objectif : Différencier VM2 de VM1 car les 3 VMS on disposent du même « mpj » avec laquelle nous avons créé la VM de départ.

```
# Éditer /etc/hosts  
sudo nano /etc/hosts  
# Remplacer "ubuntu" par "Ubuntu-Server-vm2" dans la ligne 127.0.1.1
```



```
# Redémarrer pour appliquer  
sudo reboot
```



Ubuntu-Server-VM1: Configuration et Administration Avancée

Cette VM applique **trois types d'administration avancée** : CPU, Stockage, et Réseau.

1) Configuration avancée : Supervision système avec Prometheus & Node Exporter

➤ Objectif

Installation d'un système de monitoring moderne pour **collecter et exposer les métriques système** (CPU, RAM, disque, réseau).

1.1) Installation de Node Exporter

Node Exporter exposition des métriques du système au format Prometheus.

```
# Créer un utilisateur pour node_exporter
sudo useradd --no-create-home --shell /bin/false node_exporter

# Télécharger Node Exporter
cd /tmp
wget
https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz

# Extraire
tar xvfz node_exporter-1.7.0.linux-amd64.tar.gz

# Copie DU binaire
sudo cp node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/
sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter

# Nettoyer
rm -rf /tmp/node_exporter-1.7.0*
```
```

#### ### 1.2) Crée le service systemd

```
sudo vi /etc/systemd/system/node_exporter.service
```

```

[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target

```

**Activer et démarrer :**

```

sudo systemctl daemon-reload
sudo systemctl start node_exporter
sudo systemctl enable node_exporter

```

# Vérifier

```
sudo systemctl status node_exporter
```

```

root@Ubuntu-Server-VM1:/# sudo systemctl daemon-reload
root@Ubuntu-Server-VM1:/# sudo systemctl start node_exporter
root@Ubuntu-Server-VM1:/#
root@Ubuntu-Server-VM1:/# sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
root@Ubuntu-Server-VM1:/#
root@Ubuntu-Server-VM1:/#
root@Ubuntu-Server-VM1:/# sudo systemctl status node_exporter
root@Ubuntu-Server-VM1:/# node_exporter.service - Node Exporter
● node_exporter.service - Node Exporter
 Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset>
 Active: active (running) since Thu 2025-10-23 17:44:27 UTC; 15s ago
 Main PID: 7740 (node_exporter)
 Tasks: 4 (limit: 2210)
 Memory: 2.4M (peak: 2.6M)
 CPU: 10ms
 CGroup: /system.slice/node_exporter.service
 └─7740 /usr/local/bin/node_exporter

oct 23 17:44:27 Ubuntu-Server-VM1 node_exporter[7740]: ts=2025-10-23T17:44:27.>
oct 23 17:44:27 Ubuntu-Server-VM1 node_exporter[7740]: ts=2025-10-23T17:44:27.>
```

```

Les métriques sont exposées sur le port 9100
curl http://localhost:9100/metrics

```

```

root@Ubuntu-Server-VM1:/# curl http://localhost:9100/metrics
HELP go_gc_duration_seconds A summary of the pause duration of garbage collect
TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0.00025091
go_gc_duration_seconds{quantile="0.25"} 0.00025091
go_gc_duration_seconds{quantile="0.5"} 0.000279446
go_gc_duration_seconds{quantile="0.75"} 0.000279446
go_gc_duration_seconds{quantile="1"} 0.000279446
go_gc_duration_seconds_sum 0.000530356
go_gc_duration_seconds_count 2
HELP go_goroutines Number of goroutines that currently exist.
TYPE go_goroutines gauge
go_goroutines 8
HELP go_info Information about the Go environment.
TYPE go_info gauge
go_info{version="go1.21.4"} 1
HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.5168e+06
HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if fr
TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 4.725648e+06
HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bu

```

## Clone of Ubuntu-Server-VM2 : Configuration avancée : Reverse Proxy avec Nginx Objectif

Configurer Nginx comme reverse proxy pour router les requêtes HTTP vers différents services.

```

1.1) Installation de Nginx
sudo apt update
sudo apt install nginx -y
Vérifier l'installation
sudo systemctl status nginx

```

```

root@CloneofUbuntu-Server-VM2:~# sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
 Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
 Active: active (running) since Thu 2025-10-23 21:29:53 UTC; 46s ago
 Docs: man:nginx(8)
 Process: 3267 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
 Process: 3268 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
 Main PID: 3304 (nginx)
 Tasks: 3 (limit: 2210)
 Memory: 2.4M (peak: 5.0M)
 CPU: 41ms
 CGroup: /system.slice/nginx.service
 ├─3304 "nginx: master process /usr/sbin/nginx -g daemon on; master_
 ├─3307 "nginx: worker process"
 └─3308 "nginx: worker process"

oct. 23 21:29:53 CloneofUbuntu-Server-VM2 systemd[1]: Starting nginx.service ->
oct. 23 21:29:53 CloneofUbuntu-Server-VM2 systemd[1]: Started nginx.service - A>
Lines 1-17/17 (END)

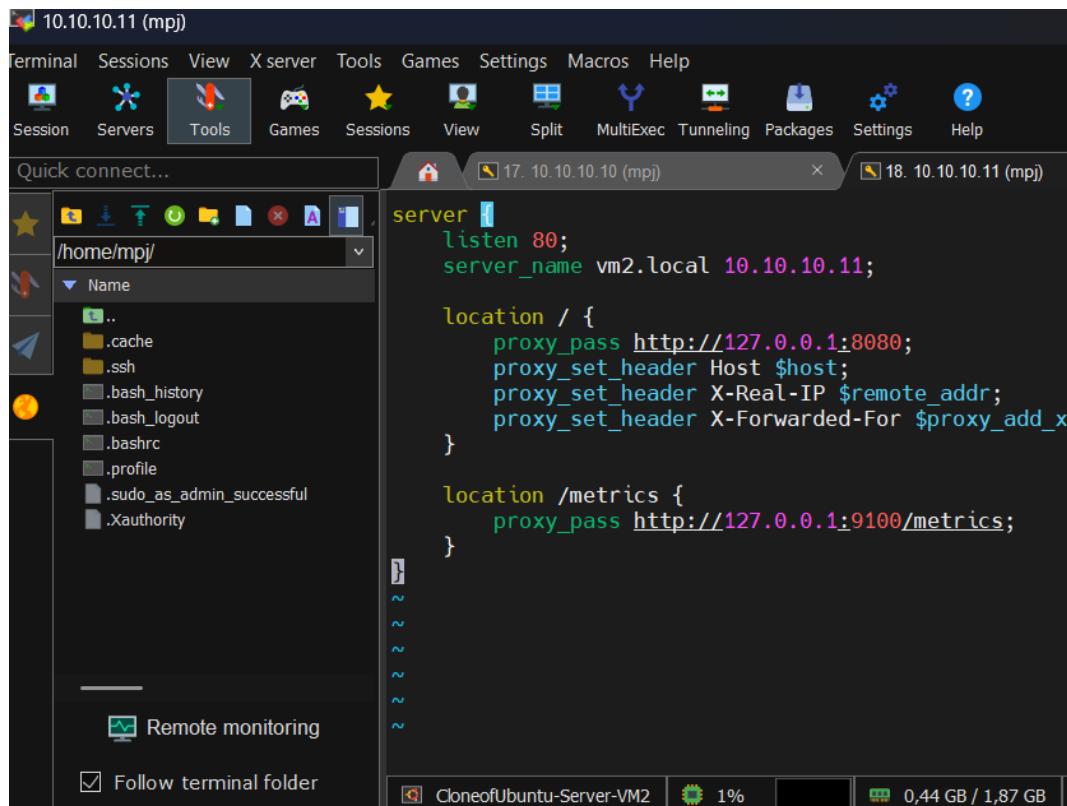
```

```
Créer un service web basique sur le port 8080
mkdir -p ~/webapp
echo "<h1>Application Web Clone of Ubuntu-Server-VM2</h1><p>Service backend
sur port 8080</p>" > ~/webapp/index.html
```



### ### 1.2) Configurer le reverse proxy

```
Créer un fichier de configuration
sudo vi /etc/nginx/sites-available/reverse-proxy
```



### 1.2) Teste

```
Depuis la VM
curl http://localhost
```

```

Process: 12339 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process
Process: 12341 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (>
Main PID: 12343 (nginx)
Tasks: 3 (limit: 2210)
Memory: 2.4M (peak: 2.7M)
CPU: 26ms
CGroup: /system.slice/nginx.service
 └─12343 "nginx: master process /usr/sbin/nginx -g daemon on; master_>
 ├─12344 "nginx: worker process"
 ├─12345 "nginx: worker process"
 └─12346 "nginx: worker process"

oct. 23 21:55:01 CloneofUbuntu-Server-VM2 systemd[1]: Starting nginx.service - >
oct. 23 21:55:01 CloneofUbuntu-Server-VM2 systemd[1]: Started nginx.service - A>
root@CloneofUbuntu-Server-VM2:~# curl http://localhost
127.0.0.1 - - [23/Oct/2025 21:55:31] "GET / HTTP/1.0" 200 -
<h1>Application Web Application Web Clone of Ubuntu-Server-VM2</h1><p>Service ba
ckend sur port 8080</p>
root@CloneofUbuntu-Server-VM2:~#
root@CloneofUbuntu-Server-VM2:~# 10.10.10.1 - - [23/Oct/2025 21:55:42] "GET / HT
TP/1.1" 304 -

```

## ## 2) Configuration avancée : Serveur NTP (Network Time Protocol)

### ##2.1) Objectif

Configurer VM2 comme serveur NTP pour synchroniser l'heure avec VM1 et VM3.

### ## 2.2) Installation de Chrony

`sudo apt update`

`sudo apt install chrony -y`

### ## 2.3) Configuration

# Éditer la configuration

`sudo vi /etc/chrony/chrony.conf`

```

This directive enables kernel synchronisation (every 11 minutes) of the
real-time clock. Note that it can't be used along with the 'rtcsync' directive.
rtcsync

Step the system clock instead of slewling it if the adjustment is larger than
one second, but only in the first three clock updates.
makestep 1 3

Get TAI-UTC offset and leap seconds from the system tz database.
This directive must be commented out when using time sources serving
leap-smeared time.
leapsectz right/UTC

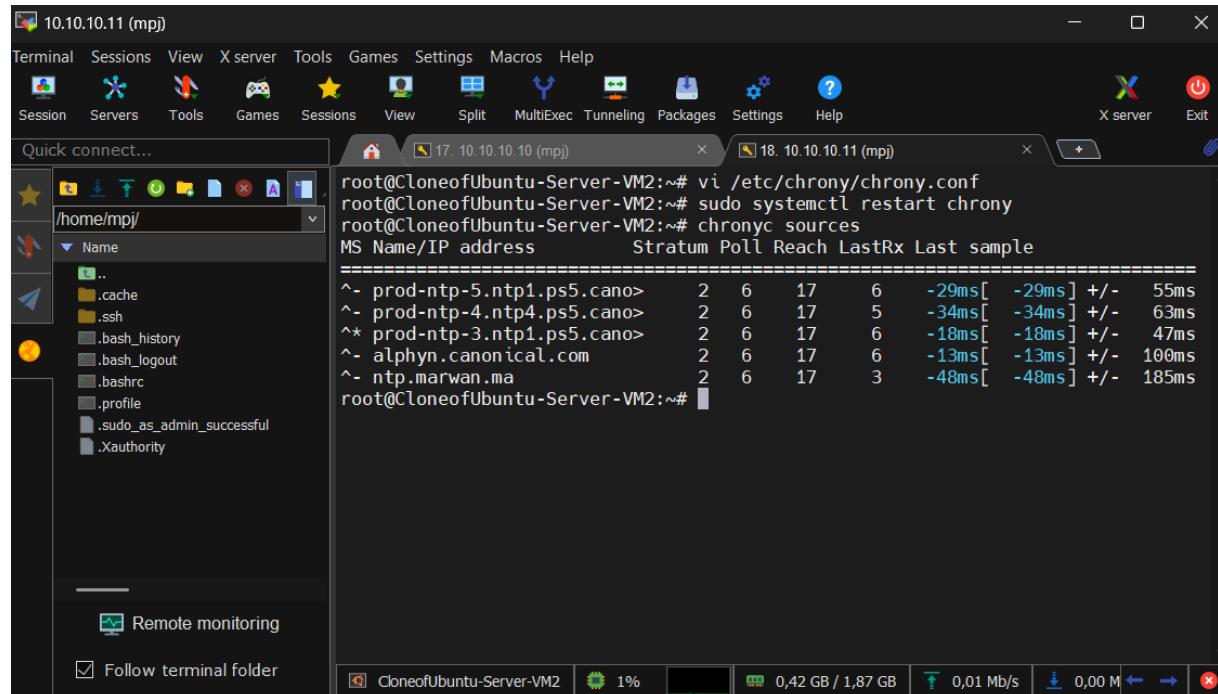
Permettre aux clients du réseau local de se synchroniser
allow 10.10.10.0/24

Servir l'heure même si non synchronisé avec Internet
local stratum 10

```

### ### 2.4) Vérifier et tester

```
Voir les sources NTP
chronyc sources
Voir les statistiques
chronyc tracking
```



```
Voir les clients connectés
```

```
chronyc clients
```

```
,
```

**\*\*Sur Ubuntu-Server-VM1 (client NTP)\*\* :**

```
Installer chrony si pas déjà fait
sudo apt install chrony -y
```

```
Éditer la config
sudo vi /etc/chrony/chrony.conf
```

```
Ajouter comme serveur NTP
server 10.10.10.11 iburst prefer
```

```
Redémarrer
sudo systemctl restart chrony
```

```
Vérifier la synchronisation
chronyc sources
```

```

root@Ubuntu-Server-VM1:~# vi /etc/chrony/chrony.conf
root@Ubuntu-Server-VM1:~# systemctl restart chrony
root@Ubuntu-Server-VM1:~# chronyc sources
MS Name/IP address Stratum Poll Reach LastRx Last sample
=====
~- alphyn.canonical.com 2 6 7 2 +1323us[-2496us] +/- 95ms
~- prod-ntp-5.ntp4.ps5.cano> 2 6 17 1 +5631us[+5631us] +/- 44ms
~- prod-ntp-3.ntp1.ps5.cano> 2 6 17 0 +2102us[+2102us] +/- 35ms
~- prod-ntp-4.ntp4.ps5.cano> 2 6 17 1 +5719us[+5808us] +/- 33ms
~- ntp.marwan.ma 2 6 17 1 -9777us[-9687us] +/- 164ms
~* 10.10.10.11 3 6 17 1 +12us[+101us] +/- 43ms
root@Ubuntu-Server-VM1:~#

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional Edition here: <https://mobaxterm.mobatek.net>

### 3) Configuration avancée : Journalisation centralisée avec rsyslog

#### 3.1) Objectif

Configuration du Clone of Ubuntu-Server-VM2 comme serveur de logs centralisé pour collecter les logs de Ubuntu-Server-VM1 et des autres VMs.

```

Créer un fichier par hôte
template(name="RemoteHost" type="string" string="/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log")

Sauvegarder les logs des clients distants
if $fromhost-ip != '127.0.0.1' then ?RemoteHost
& stop
~

"/etc/rsyslog.d/10-remote-logs.conf" 6L, 226B

```

```

root@CloneofUbuntu-Server-VM2:~# ls -la /var/log/remote/
total 16
drwxr-xr-x 4 syslog adm 4096 oct. 23 22:24 .
drwxrwxr-x 13 root syslog 4096 oct. 23 22:21 ..
drwxr-xr-x 2 syslog syslog 4096 oct. 23 22:25 Ubuntu-Server-VM1
drwxr-xr-x 2 syslog syslog 4096 oct. 23 22:25 Ubuntu-Server-VM3
root@CloneofUbuntu-Server-VM2:~#

```

| Différences clés entre Ubuntu-Server-VM1 et Clone of Ubuntu-Server-VM2 |                           |                            |
|------------------------------------------------------------------------|---------------------------|----------------------------|
| Aspect                                                                 | Ubuntu-Server-VM1         | Clone of Ubuntu-Server-VM2 |
| Hostname                                                               | Ubuntu-Server-VM1         | CloneofUbuntu-Server-VM2   |
| IP Host-only                                                           | 10.10.10.10               | 10.10.10.11                |
| IP Bridged                                                             | dhcp                      | dhcp                       |
| Config 1                                                               | Node Exporter (métriques) | Nginx Reverse Proxy        |
| Config 2                                                               | Automatisation cron       | Serveur NTP                |
| Config 3                                                               | Fail2Ban (sécurité SSH)   | Logs centralisés rsyslog   |

## CONCLUSION

Ce travail pratique a permis de déployer avec succès une infrastructure Linux complète de trois machines virtuelles, chacune remplissant un rôle spécifique et complémentaire.

**Ubuntu-Server-VM1** assure la supervision et la sécurité, **Clone of Ubuntu-Server-VM2** centralise les services réseau critiques (reverse proxy, NTP, logs centralisés), et **Ubuntu-Server-VM3** illustre l'intégration d'un client dans cette architecture distribuée.

Les compétences acquises couvrent la supervision avec Node Exporter, l'automatisation par scripts cron, la sécurisation SSH avec Fail2Ban, le routage HTTP via Nginx, la synchronisation temporelle NTP, et la centralisation des logs avec rsyslog. Les défis rencontrés (conflits de configuration, gestion des IP après clonage) ont renforcé nos compétences en dépannage système.

Cette infrastructure démontre l'application concrète des principes d'administration moderne : centralisation des services, automatisation des tâches, sécurisation des accès, et supervision continue. Le projet reflète une approche professionnelle et méthodique de la gestion d'infrastructures Linux, avec une documentation complète facilitant la maintenance et l'évolution future du système.

**NB :** Pour consulter les procédures de configuration détaillées de l'ensemble de la « PARTIE 4 : Administration Avancée », veuillez-vous référer aux annexes jointes au présent rapport.