Projet 2

Par Rémi Albertucci
Le 06/04/2026
Durée: 1 mois

Table des matières

I Contexte et mise en situation.…3

II Adressage réseau.....4

II.1 Adressage.....4
II.2 Schéma.....6

III OPNSense.....7

III.1 Pourquoi opnsense.....7
III.2 Création de la machine.....7
III.3 Création des réseaux.....8
III.4 Création des règles.....9
Création d’alias.....10
.....10
Création des règles.....11
Test.....12
III.5 Trafic shaping.....12

IV Active Directory.....15

IV.1 Création de la machine.....15
IV.2 Configuration.....15
IV.3 Import des utilisateurs et OU….17

V Serveur de fichier.....21

V.1 Configuration de base.....21
V.2 Configuration du stockage partagé
.....22
V.3 Test.....26

VI VPN Client-to-Site.....28

VI.1 Choix protocole.....28
VI.2 Créer une instance.....28
VI.3 Créer client.....30
VI.4 Configurer interface et pare feu 31
VI.5 Configurer client (windows).....32

VII ERP.....33

VII.1 Choix de la solution.....33
VII.2 Creation VM.....34
VII.3 Configuration VM.....37
Mise a jour.....37
Fixer Ip.....37
Création utilisateur et ajout à sudo.....38
Configuration de sécurité.....38
VII.4 Installation de docker.....38
VII.5 Création conteneur.....40
VII.6 Mise en route de dolibarr.....41
Modules.....42
créer groupes.....43
créer users.....45

VIII Supervision.....47

VIII.1 Prometheus.....47
Creation ct lxc et mise en route docker et conteneur.....47
Configuration.....49
Installer sur linux.....54
Installer l’exporter sur Windows
.....55
Accéder au dashboard.....57
Installer sur Opnsense.....58
Accéder aux dashboards.....59

IX Sauvegardes.....62

rsync.....62
Test.....63
Planification.....63

X Conclusion.....64

I Contexte et mise en situation

Vita Big Pharma est une entreprise du secteur pharmaceutique, spécialisée dans la fabrication de compléments alimentaires. Dans le cadre d’une stratégie d'expansion de ses activités à l'échelle nationale, l’entreprise a choisi de s'implanter en France à travers le déploiement de deux sites distants et complémentaires :

Le site de Toulouse, qui constitue le siège administratif de l'entreprise. Il regroupe les services de la Direction, des Ressources Humaines ainsi que de la Finance.

Le site de Marseille, à vocation purement technique, qui héberge le service technique et le pôle de support informatique.

L'objectif principal de ce projet est de concevoir, maquetter et déployer une infrastructure réseau et système centralisée, hautement sécurisée, évolutive et alignée sur les bonnes pratiques de l'ingénierie système. Cette infrastructure doit répondre à plusieurs enjeux métiers critiques : assurer la continuité d'activité des services, offrir une solution de télétravail sécurisée aux collaborateurs mobiles, faciliter l'assistance technique inter-sites et garantir la qualité de service (QoS) globale du réseau. Afin d'assurer un contrôle permanent de l'environnement, le projet intègre également des mécanismes centralisés de supervision ainsi qu'une politique rigoureuse de sauvegarde.

II Adressage réseau

II.1 Adressage

Afin de garantir une sécurité accrue, une isolation optimale des flux et une gestion simplifiée de l'infrastructure du site de Toulouse, un plan d'adressage IP privé rigoureux a été défini.

Nous avons opté pour un découpage en sous-réseaux utilisant un masque de sous-réseau standard en `/24` (255.255.255.0). Ce choix technique offre jusqu'à 254 adresses IP utiles par sous-réseau, ce qui répond largement aux besoins actuels de l'entreprise tout en offrant une grande flexibilité pour une extension future.

L'infrastructure est segmentée en trois zones logiques distinctes (Collaborateurs, Serveurs et VPN), étanches entre elles et contrôlées par le pare-feu OPNSense.

Le tableau ci-dessous synthétise la configuration des sous-réseaux ainsi que l'attribution statique des adresses IP pour l'ensemble des serveurs et services de la filiale.

Réseau	IP/CIDR	GW	IP Range
Collaborateurs	192.168.200.0/24	192.168.200.1	10-200
Serveurs/Administration	192.168.100.0/24	192.168.100.1	10-200
VPN client	10.11.12.0/24	10.11.12.1	/2-255
WAN	10.34.40.XXX/32		

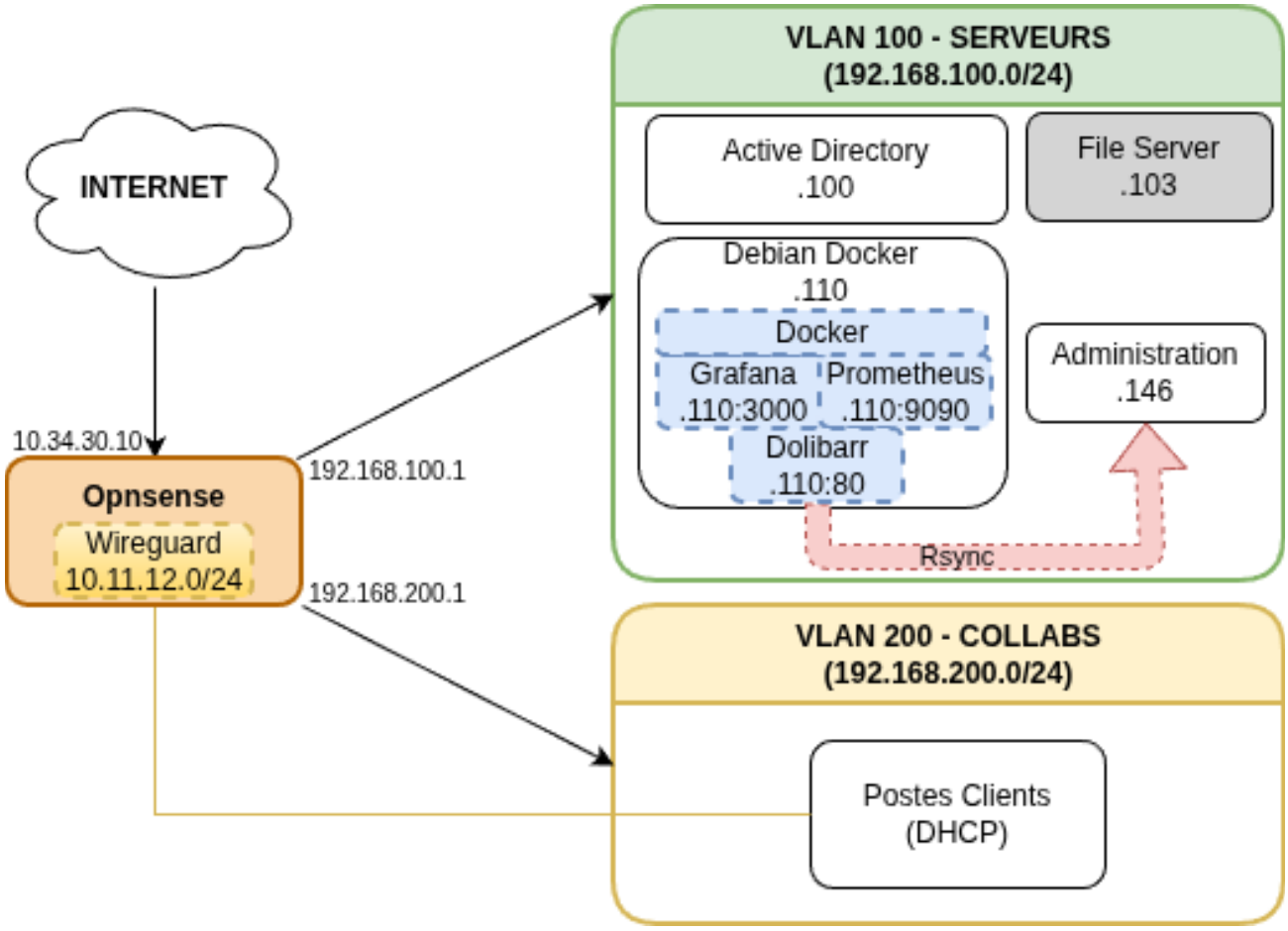
Services	ID	IP
OPNSense		192.168.100.1
Active Directory		192.168.100.100
Active Directory (2)		192.168.100.101
Serveur de fichiers		192.168.100.103
Dolibarr		192.168.100.110:80
Monitoring		192.168.100.110:3000
Machine d'administration		192.168.100.146

Domaine	tls.vitabigpharma.local
DNS externe	1.1.1.1

II.2 Schéma

Le schéma de topologie réseau présenté ci-après permet de visualiser de manière synthétique l'architecture logique implémentée sur le site de Toulouse.

Cette modélisation met en évidence la centralisation de l'infrastructure autour du pare-feu OPNSense, qui fait office de cœur de réseau en assurant le routage inter-réseau.



III OPNSense

III.1 Pourquoi opnsense

OPNsense est une distribution pare-feu open source, activement maintenue et disposant d'une interface web complète.

Elle a été retenue pour sa gestion native du routage inter-réseau, du DHCP par interface, et pour son support intégré de WireGuard.

C'est une solution utilisée en entreprise et bien documentée, ce qui facilite l'administration et la résolution d'incidents.

III.2 Création de la machine

Le déploiement du pare-feu a été réalisé sur l'hyperviseur Proxmox VE.

La procédure suivante détaille la création de la machine virtuelle, le dimensionnement de ses ressources ainsi que l'association des cartes réseau physiques virtuelles aux différents ponts de l'hôte pour isoler nos futurs flux.

On lui attribue le minimum de ressources nécessaires pour sa bonne exécution (https://docs.opnsense.org/manual/hardware.html).

Comme nous faisons la configuration par l'intermédiaire du GUI de Proxmox, nous ne pouvons ajouter qu'une carte réseau lors de cette étape. Nous ajoutons donc deux cartes par la suite et nous notons leur adresse mac :

- vmbr41 : BC:24:11:56:64:6D
- vmbr42 : BC:24:11:96:16:AE
- vmbr43 : BC:24:11:B8:76:DD

Elles correspondra à nos réseaux :

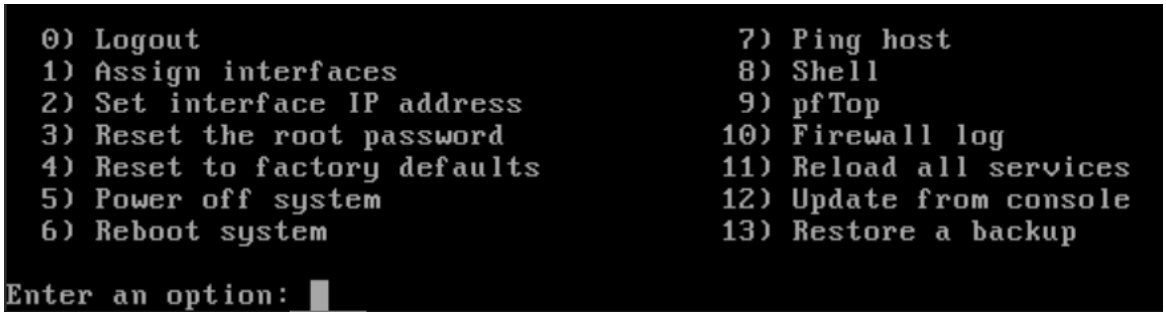
- Lan Serveurs: 192.168.100.1/24
- Lan Collaborateurs : 192.168.200.1/24
- WAN : 10.34.40.10/24

Mémoire	2.00 Gio
Processeurs	1 (1 sockets, 1 cores) [x86-64-v2-AES]
BIOS	Par défaut (SeaBIOS)
Affichage	Par défaut
Machine	Par défaut (i440fx)
Contrôleur SCSI	VirtIO SCSI single
Disque dur (scsi0)	lv_remi.albertucci:vm-40100-disk-0,iothread=1,size=32G
Carte réseau (net0)	virtio=BC:24:11:56:64:6D,bridge=vmbr41
Carte réseau (net1)	virtio=BC:24:11:96:16:AE,bridge=vmbr42
Carte réseau (net2)	virtio=BC:24:11:B8:76:DD,bridge=vmbr43

III.3 Création des réseaux

Une fois le système initialisé, l'étape suivante a consisté à configurer les interfaces logiques (WAN, LAN Serveurs, LAN Collaborateurs) et à activer les services DHCP pour automatiser la distribution des adresses IP. Voici les étapes de configuration et de validation des accès à l'interface d'administration :

On rentre dans la console du conteneur, on s’identifie et et on commence la configuration :



Dans l’ordre, nous allons

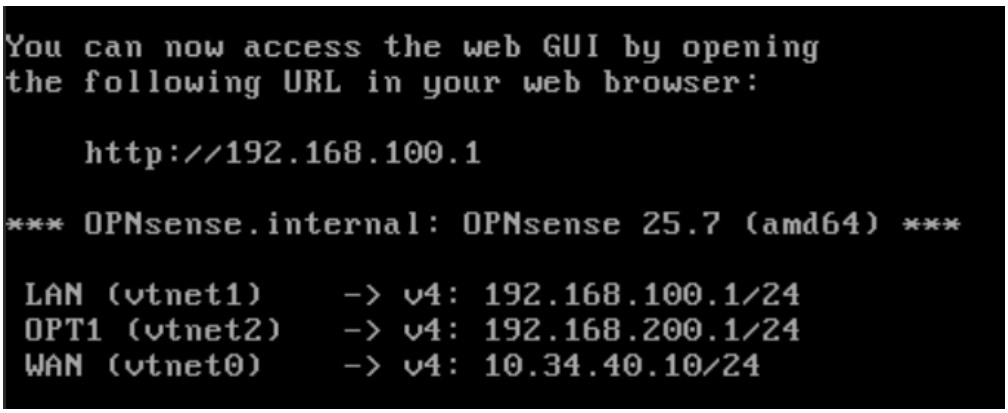
- modifier le mot de passe root
- assigner les interfaces
- adresser des ips aux interfaces

Chaque interface LAN dispose d'un service DHCP pour distribuer automatiquement les adresses aux machines du réseau correspondant, avec des machines qui pourront prendre des adresses entre 192.168.X.10 et 192.168.X.200.

C’est utile lorsque l’on veut réserver certains créneaux.

Les passerelles sont 192.168.100.1 pour le réseau serveurs et 192.168.200.1 pour le réseau collaborateurs.

On peut maintenant joindre le GUI pour la suite de la configuration à l’adresse http://192.168.100.1



On va donc créer une machine Debian pour continuer l'administration, avec un bureau lxde économe en ressources.
Le but sera d'administrer opnsense et d'autres services par GUI à partir de cette machine sur le réseau interne.

III.4 Création des règles

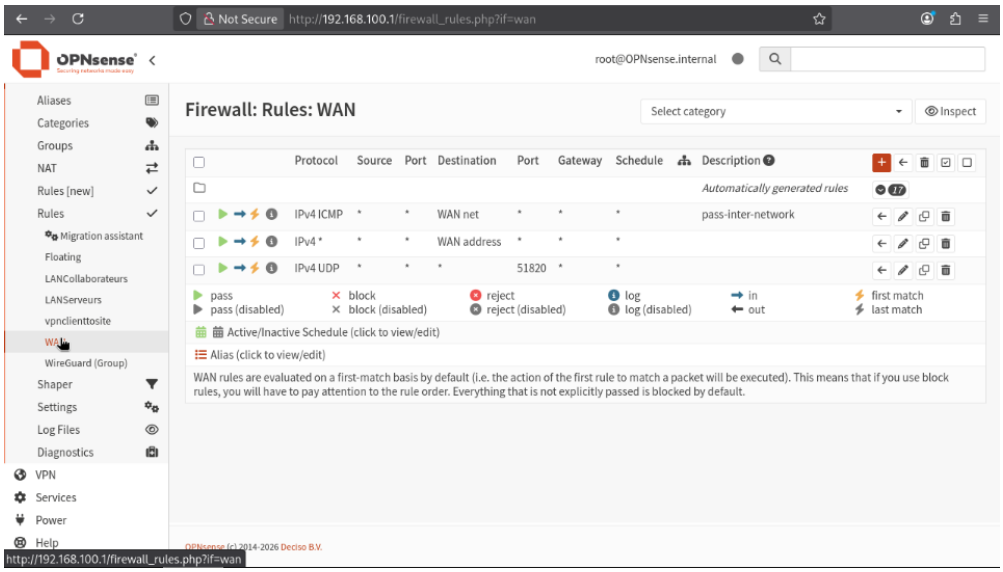
Sur l'interface LANServeurs, les règles par défaut autorisent tout le trafic sortant depuis le réseau serveurs.

Sur le WAN, trois règles sont configurées : autorisation de l'ICMP, accès à l'adresse WAN, et ouverture du port UDP 51820 qui nous servira plus tard pour WireGuard.

Tout trafic non explicitement autorisé est bloqué par défaut, nous n'ajouterons donc pas de règle « deny all » en dernière règle.

Actuellement sous la version 25.7, Opnsense s'apprête à mettre à jour leur système de règles : actuellement nous définissons les règles pour chaque interface, mais à l'avenir il y aura un système « centralisé ».

Dans l'onglet de l'interface Wan, nous avons donc :

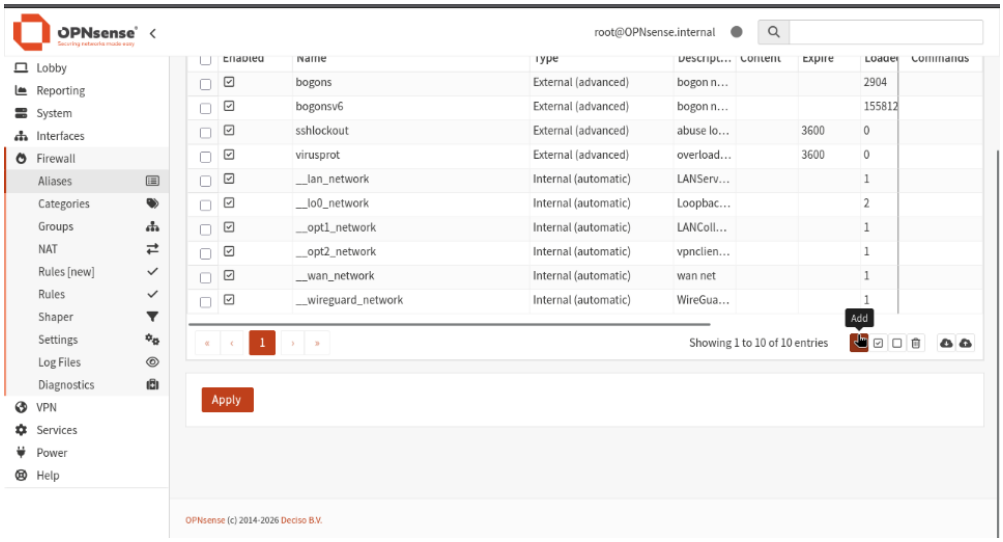


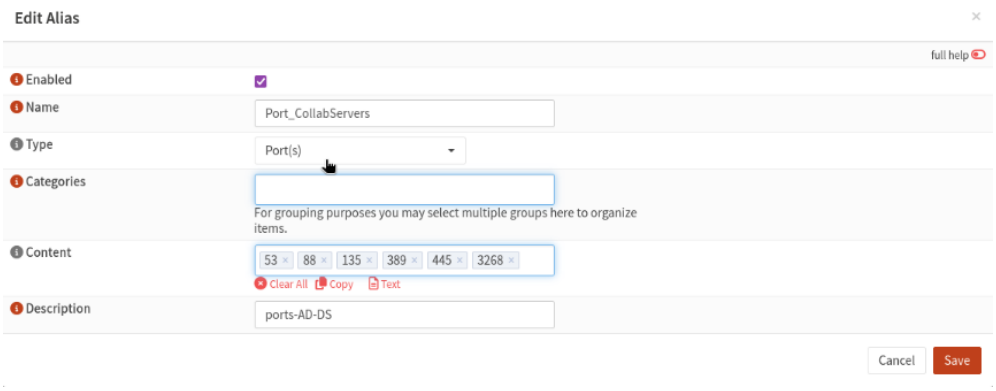
Dans l’interface Lan Collaborateurs, par souci de simplicité et dans un contexte hors production, une règle autorisant tout le trafic de LANCollaborateurs vers LANServeurs a été mise en place.

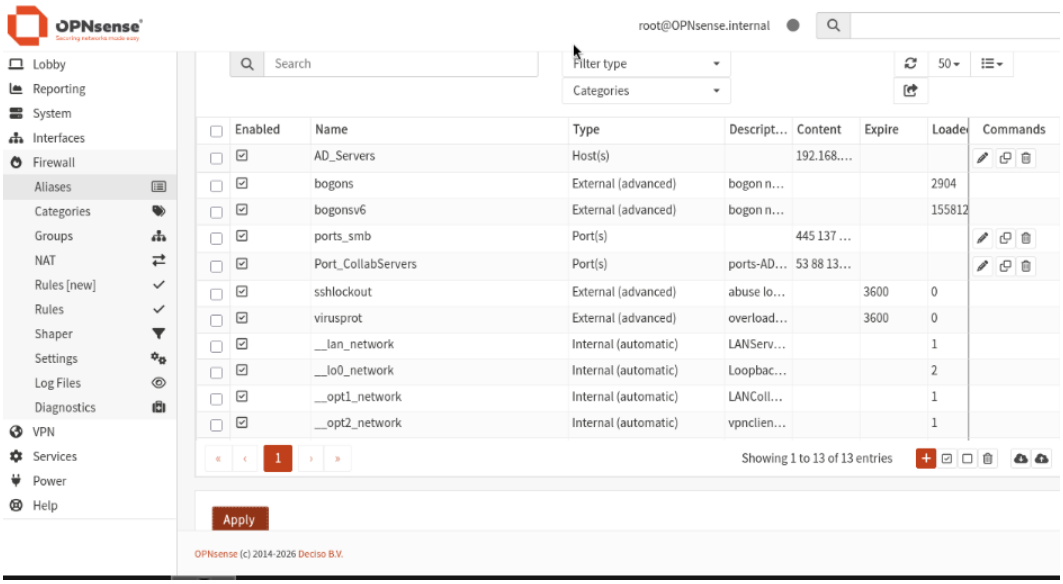
Cette règle serait affinée par service :

- AD/DNS : TCP/UDP 53, 88, 135, 389, 445, 636, 3268, 49152:65535
- Dolibarr : TCP 80
- Serveur de fichiers : TCP 445
- Monitoring : pas besoin car on ne va pas monitorer les pc, on surveille les serveurs
- Monitoring => on monitore les serveurs, donc sur le même réseau

Création d’alias





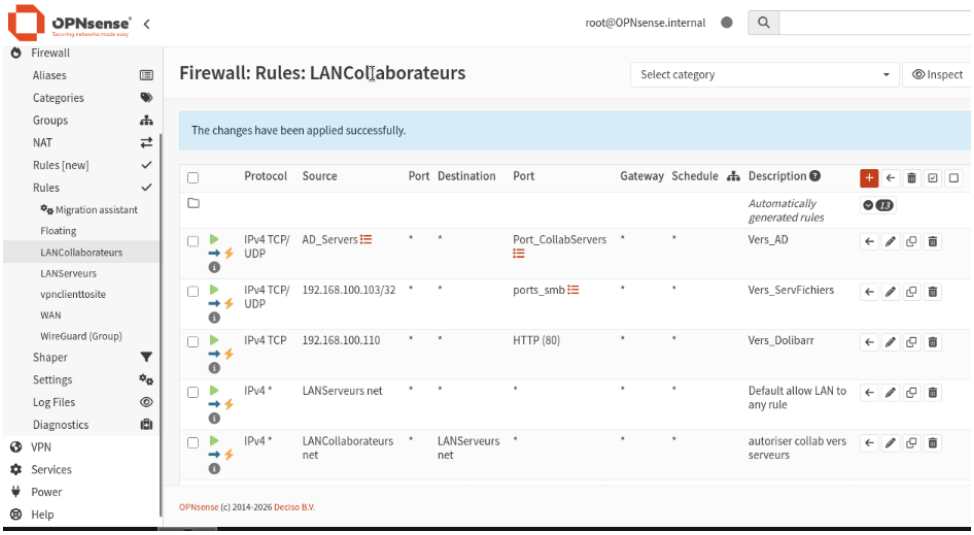


Création des règles

Afin de sécuriser l'infrastructure selon le principe du moindre privilège, une politique de filtrage stricte a été implémentée. Les manipulations ci-dessous décrivent d'abord la création d'alias pour simplifier la gestion des règles, puis le déploiement des autorisations d'accès entre les zones Collaborateurs et Serveurs.

On crée 3 règles en se servant des alias que l'on a créé précédemment :

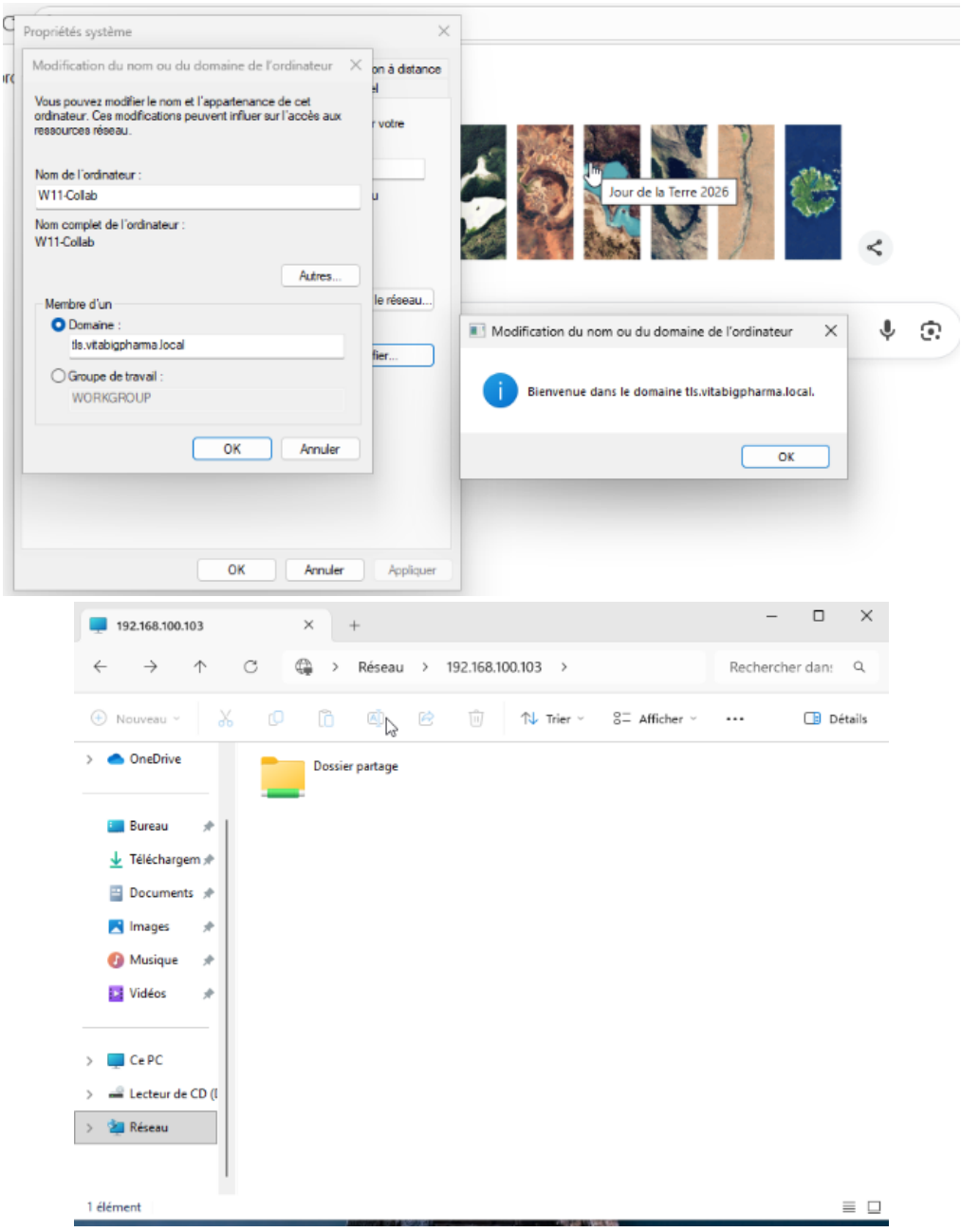
	Action	Protocole	Source	Destination	Dest. port
AD	Pass	TCP/UDP	LANCollaborateurs net	servers_AD	ports_AD
Fichiers	Pass	TCP/UDP	LANCollaborateurs net	192.168.100.103	ports_SMB
Dolibarr	Pass	UDP	LANCollaborateurs net	192.168.100.110	80



Avec une description claire pour pouvoir s'y retrouver plus tard

Test

Ici, on se connecte à l’ad sur notre machine Windows :

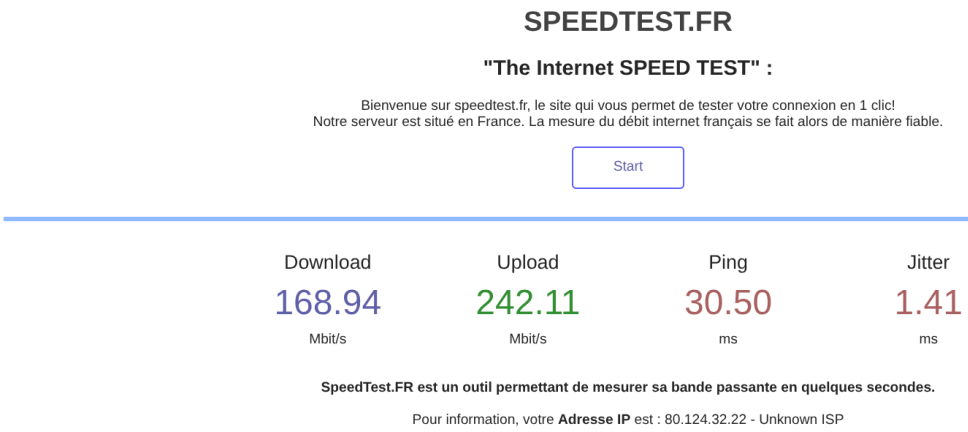


III.5 Trafic shaping

Pour garantir une qualité de service (QoS) optimale et éviter qu'un utilisateur isolé ne sature la bande passante de la filiale, un mécanisme de limitation de débit à 5mbp/s a été mis en œuvre. La procédure suivante détaille la création des règles de limitation de trafic et les tests de validation associés.

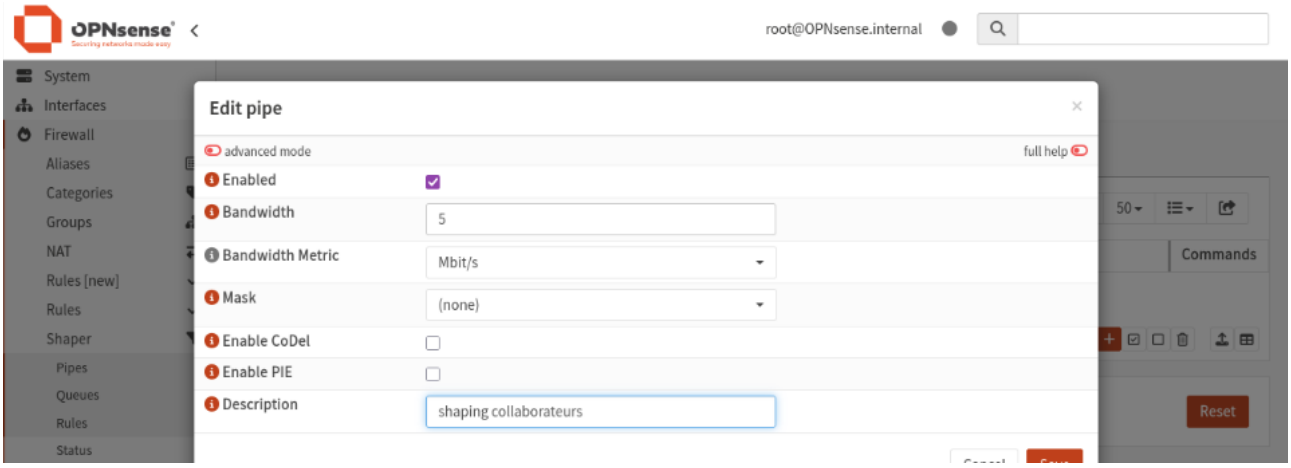
Nous allons intervenir sur le réseau collaborateur, 192.168.200.0/24.

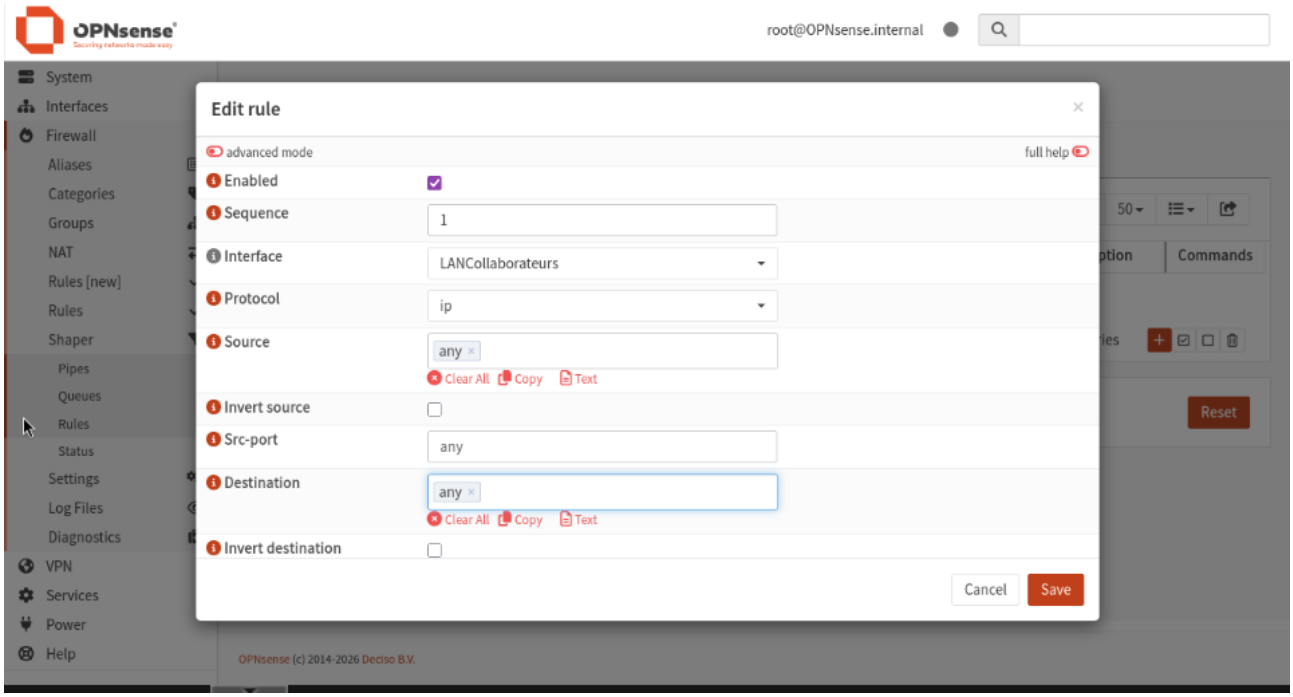
Avant toute chose, nous allons faire un test du réseau pour vérifier que la limite soit utilise : sur notre machine « collaborateur », on lance un speedtest :



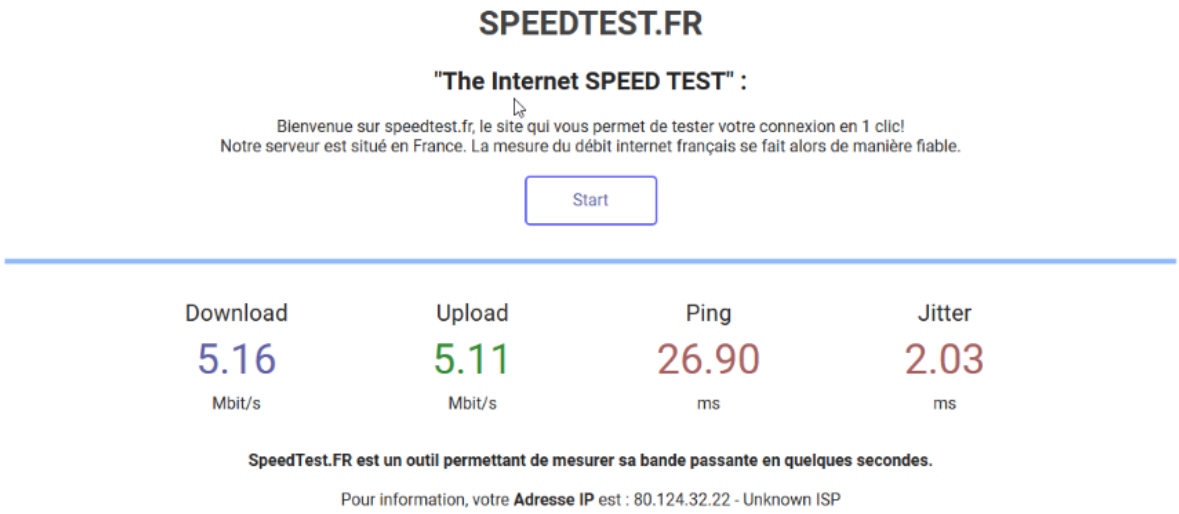
Il est donc utile de limiter le trafic pour éviter la congestion.

Dans OPNsense, on se rend dans l'onglet Shapers => Pipe et on crée un pipe :





On enregistre et on applique les paramètres avant de retester :



IV Active Directory

L’Active Directory (AD) est un service d'annuaire développé par Microsoft, indispensable au sein d'une infrastructure d'entreprise.

Il sert à centraliser la gestion des identités (utilisateurs, ordinateurs, serveurs) et des ressources du réseau. Grâce à l'AD, les collaborateurs bénéficient d'un mécanisme d'authentification unique permettant d'accéder à l'ensemble de leurs outils de travail (sessions Windows, partages de fichiers, applications métiers) avec un seul couple d'identifiants. Pour l'administrateur, il permet de sécuriser l'environnement de manière globale, de structurer l'entreprise via des Unités d'Organisation (OU) et d'appliquer des politiques de sécurité ou de configuration homogènes à l'aide de stratégies de groupe (GPO).

IV.1 Création de la machine

Le service d'annuaire est hébergé sur une machine virtuelle exécutant Windows Server 2022. La procédure suivante détaille les étapes de provisionnement de la machine sur l'hyperviseur Proxmox VE, son dimensionnement matériel, ainsi que son raccordement exclusif au réseau LAN Serveurs.

Nous regardons en premier lieu donc le minimum de ressources requises pour une utilisation hors production pour économiser les ressources sur un serveur partagé par l’ensemble de la classe.

Nous trouvons les informations ici : https://learn.microsoft.com/en-us/windows-server/get-started/hardware-requirements?tabs=cpu&pivots=windows-server-2022

et créons notre machine en conséquence :

Mémoire	4.00 Gio
Processeurs	4 (1 sockets, 4 cores) [x86-64-v2-AES]
BIOS	OVMF (UEFI)
Affichage	Par défaut
Machine	pc-q35-10.0+pve1
Contrôleur SCSI	VirtIO SCSI single
Disque dur (scsi0)	lv_remi.albertucci:vm-40400-disk-1,iothread=1,size=50G
Carte réseau (net0)	virtio=BC:24:11:FE:B9:9E,bridge=vmbr42
Disque EFI	lv_remi.albertucci:vm-40400-disk-0,efitype=4m,pre-enrolled-keys=1,size=4M
État TPM	lv_remi.albertucci:vm-40400-disk-2,size=4M,version=v2.0

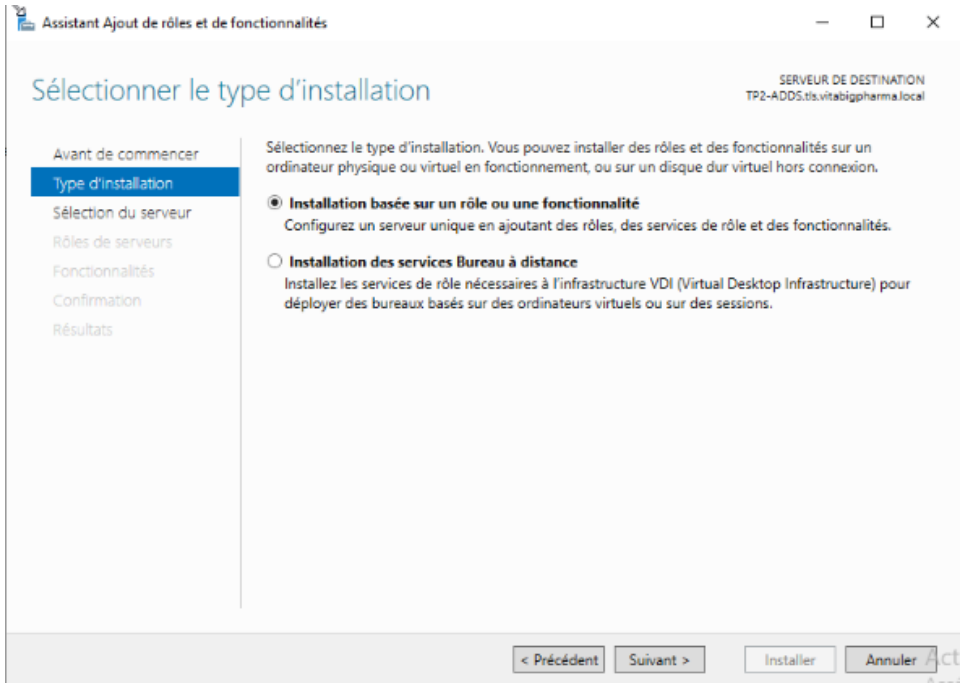
IV.2 Configuration

Une fois le système d'exploitation initialisé et l'adresse IP fixée de manière statique, l'étape suivante consiste à promouvoir le serveur en tant que Contrôleur de Domaine.

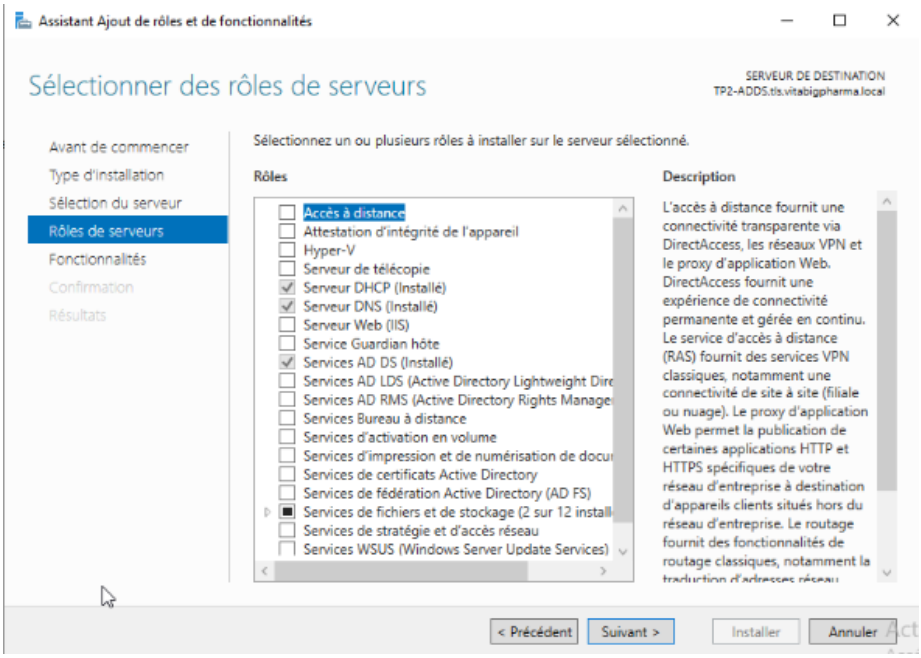
Voici le cheminement technique suivi pour installer le rôle AD DS (Active Directory Domain Services) et configurer notre nouvelle forêt de domaine. Les étapes seront :

- fixer l'IP statique (192.168.100.100)
- définir le hostname
- rebooter
- ajouter le rôle AD DS via le Gestionnaire de serveur
- promouvoir en contrôleur de domaine (DC) avec création du domaine tls.vitabigpharma.local
- rebooter
- vérifier DNS et DHCP

L’ajout du rôle et de la promotion en DC est le point crucial ici. Pour se faire, il faut aller dans la console « Gestionnaire de Serveur » et suivre « ajouter des rôles et des fonctionnalités » :



On a le choix de la sélection du serveur, n’en n’ayant qu’un le choix est évident et on continue donc sur les étapes suivantes. Nous allons ajouter les fonctionnalités de serveur DNS ainsi que le Service AD DS.



IV.3 Import des utilisateurs et OU

Afin d'éviter une création manuelle, fastidieuse, et sujette aux erreurs, le déploiement de l'arborescence de l'entreprise (Unités d'Organisation, groupes métiers et comptes utilisateurs) a été entièrement automatisé.

Nous avons conçu un script PowerShell qui lit un fichier structuré (CSV) contenant la liste du personnel pour générer l'annuaire de manière industrielle. Voici le code source implémenté et la validation de son exécution.

```
# ================= VARIABLES =================

$domainDN = "DC=tls,DC=vitabigpharma,DC=local"
$baseOU   = "OU=Collab,$domainDN"
$csvPath  = "C:\ton_chemin\users.csv"
$logFile  = "$env:USERPROFILE\Desktop\log_ad.txt"
$report   = "$env:USERPROFILE\Desktop\report_users.csv"

$DryRun = $false   # TRUE = simulation

# ================= FONCTIONS =================

function Write-MyLog ($msg) {
$line = "$(Get-Date -Format 'yyyy-MM-dd HH:mm:ss') : $msg"
Write-Host $line
$line | Out-File $logFile -Append -Encoding UTF8
}
```

```
# Enlever accents
function Remove-Accents($text) {
$normalized = $text.Normalize([Text.NormalizationForm]::FormD)
return ($normalized -replace '\p{Mn}', '')
}

# Générer mot de passe
function New-RandomPassword {
return [System.Web.Security.Membership]::GeneratePassword(12,2)
}

# Générer samAccountName unique
function Get-UniqueSam($prenom,$nom) {
$base = (Remove-Accents("$prenom.$nom")).ToLower() -replace " ", ""
$sam = $base
$i = 1

while (Get-ADUser -Filter "SamAccountName -eq '$sam'" -ErrorAction
SilentlyContinue) {
$sam = "$base$i"
$i++
}
return $sam
}

# ================= STRUCTURE OU =================

if (-not (Get-ADOrganizationalUnit -LDAPFilter "(distinguishedName=$baseOU)"
-ErrorAction SilentlyContinue)) {
if (!$DryRun) { New-ADOrganizationalUnit -Name "Collab" -Path $domainDN }
Write-MyLog "OU Collab créée"
}

foreach ($ou in @("Users","Ordinateurs","Groupes")) {
$target = "OU=$ou,$baseOU"
if (-not (Get-ADOrganizationalUnit -LDAPFilter "(distinguishedName=$target)"
-ErrorAction SilentlyContinue)) {
if (!$DryRun) { New-ADOrganizationalUnit -Name $ou -Path $baseOU }
Write-MyLog "OU $ou créée"
}
}

# ================= IMPORT CSV =================

$users = Import-Csv $csvPath -Delimiter ","
$reportData = @()

foreach ($u in $users) {

$sam = Get-UniqueSam $u.Prenom $u.Nom
```

```
$pwdPlain = "VitaBigPharma2026!"
$pwd = ConvertTo-SecureString $pwdPlain -AsPlainText -Force

# OU par service
$serviceOU = "OU=$($u.Service),OU=Users,$baseOU"

if (-not (Get-ADOrganizationalUnit -LDAPFilter "(ou=$($u.Service))"
-SearchBase "OU=Users,$baseOU" -ErrorAction SilentlyContinue)) {
    if (!$DryRun) { New-ADOrganizationalUnit -Name $u.Service -Path
"OU=Users,$baseOU" }
    Write-MyLog "OU service créée : $($u.Service)"
}

try {
    if (!$DryRun) {
        New-ADUser `
    -Name "$($u.Prenom) $($u.Nom)" `
    -GivenName $u.Prenom `
    -Surname $u.Nom `
    -SamAccountName $sam `
    -UserPrincipalName $u.Email `
    -Path $serviceOU `
    -AccountPassword $pwd `
    -Enabled $true `
    -ChangePasswordAtLogon $true
    }

    Write-MyLog "User créé : $sam"

    # Groupe
    if (-not (Get-ADGroup -Filter "Name -eq '$($u.Groupe)'" -ErrorAction
SilentlyContinue)) {
        if (!$DryRun) {
            New-ADGroup -Name $u.Groupe -GroupScope Global -GroupCategory
Security -Path "OU=Groupes,$baseOU"
        }
        Write-MyLog "Groupe créé : $($u.Groupe)"
    }

    if (!$DryRun) {
        Add-ADGroupMember -Identity $u.Groupe -Members $sam
    }
    Write-MyLog "$sam ajouté à $($u.Groupe)"

    # Rapport
    $reportData += [PSCustomObject]@{
        Prenom = $u.Prenom
        Nom = $u.Nom
        SamAccountName = $sam
        Email = $u.Email
```

```
                Service = $u.Service
                Groupe = $u.Groupe
                Password = $pwdPlain
                }
            }
        catch {
    Write-MyLog "ERREUR $sam : $($_.Exception.Message)"
        }
    }

# Export rapport
$reportData | Export-Csv $report -NoTypeInformation -Encoding UTF8
Write-MyLog "Rapport exporté : $report"
```

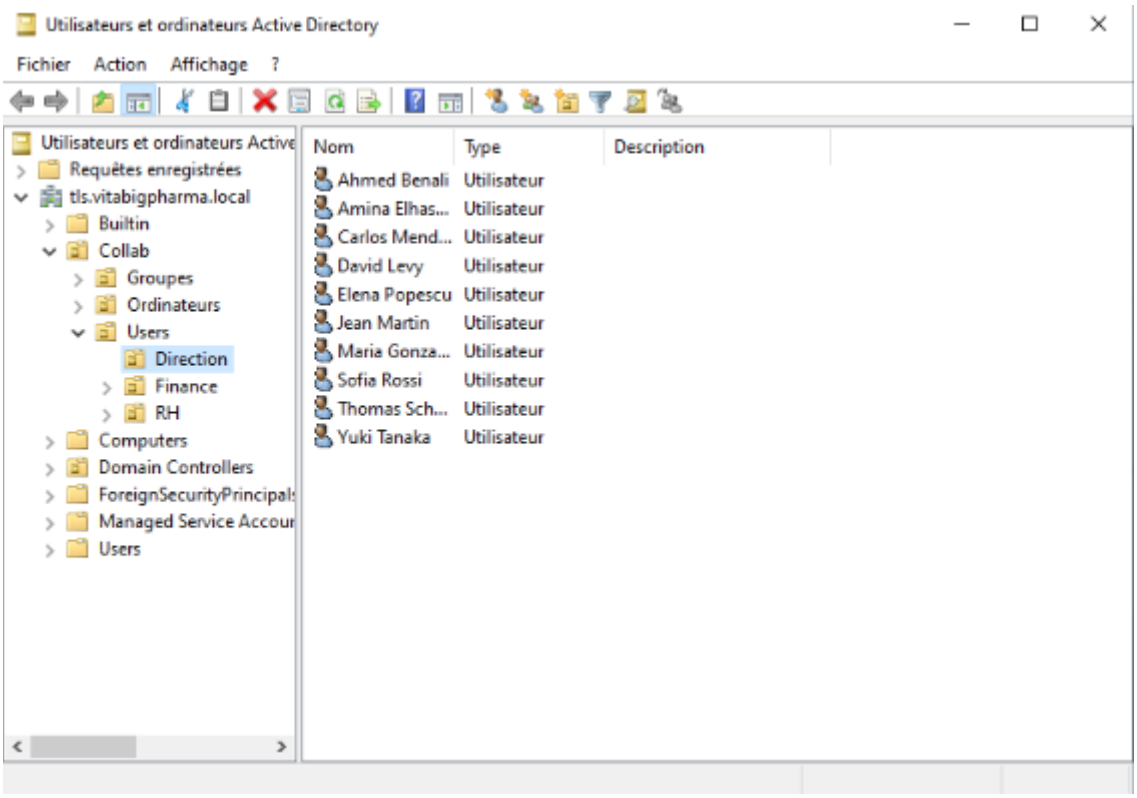
On ouvre ensuite powershell pour autoriser l'exécution de scripts :

```
Set-ExecutionPolicy RemoteSigned
```

et on se déplace jusque dans le dossier où l'on a créé le script pour le lancer :

```
.\import_users.ps1
```

Une fois fait, nous devrions avoir :



V Serveur de fichier

Le stockage et le partage d'informations au sein de Vita Big Pharma représentent un enjeu critique pour la collaboration inter-services.

La mise en place d'un serveur de fichiers dédié sous windows permet de centraliser les données de l'entreprise, de garantir leur intégrité et de contrôler précisément les accès.

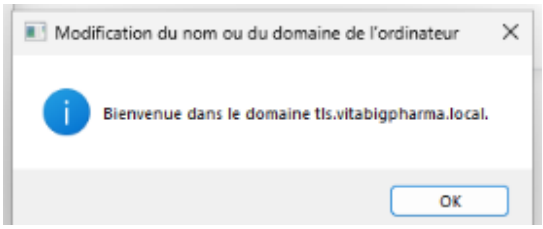
La machine sera un serveur Windows, nous partirons donc sur les mêmes ressources. L'installation sera similaire et une fois effectuée, nous allons suivre les étapes suivantes pour rendre le serveur de fichiers opérationnel.

V.1 Configuration de base

Une fois le système d'exploitation Windows Server initialisé sur notre nouvelle machine virtuelle, la première étape indispensable consiste à lui attribuer une configuration réseau fixe et stable. L'adresse IP statique a été configurée en parfaite cohérence avec la plage dédiée à notre zone LAN Serveurs.

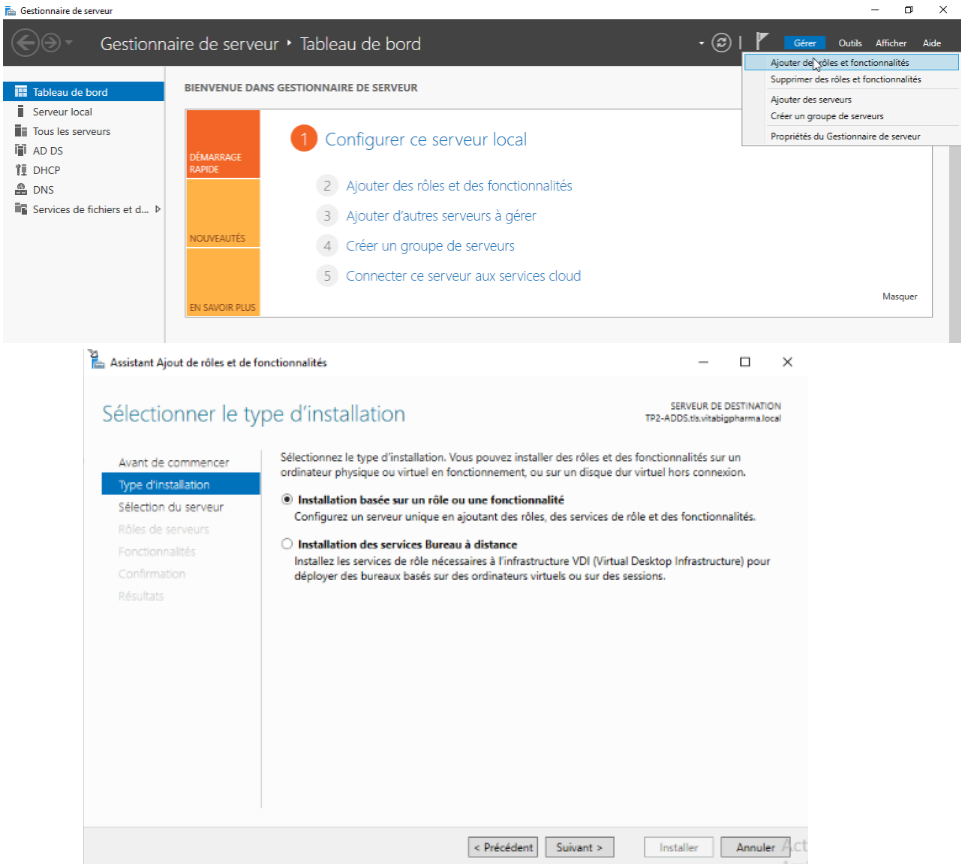
Afin de pouvoir gérer de manière centralisée les droits d'accès et s'appuyer sur l'annuaire de l'entreprise, le serveur de fichiers doit être intégré à l'infrastructure existante.

Comme nous l'avons fait précédemment, nous procédons à la jonction de la machine au domaine tls.vitabigpharma.local. La bonne communication avec le contrôleur de domaine a été validée, suivie d'un redémarrage du serveur pour appliquer les changements.



Le serveur étant désormais membre du domaine, il faut lui ajouter les fonctionnalités nécessaires à sa fonction principale.

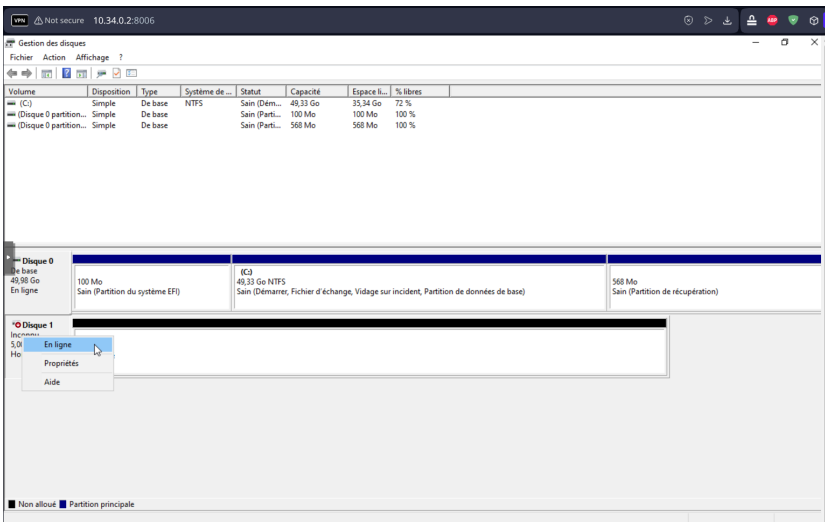
Nous procédons à l'installation du rôle requis en exécutant l'assistant d'ajout de rôles et de fonctionnalités via le Gestionnaire de serveur. Le rôle spécifique sélectionné est "Services de fichiers et de stockage" :



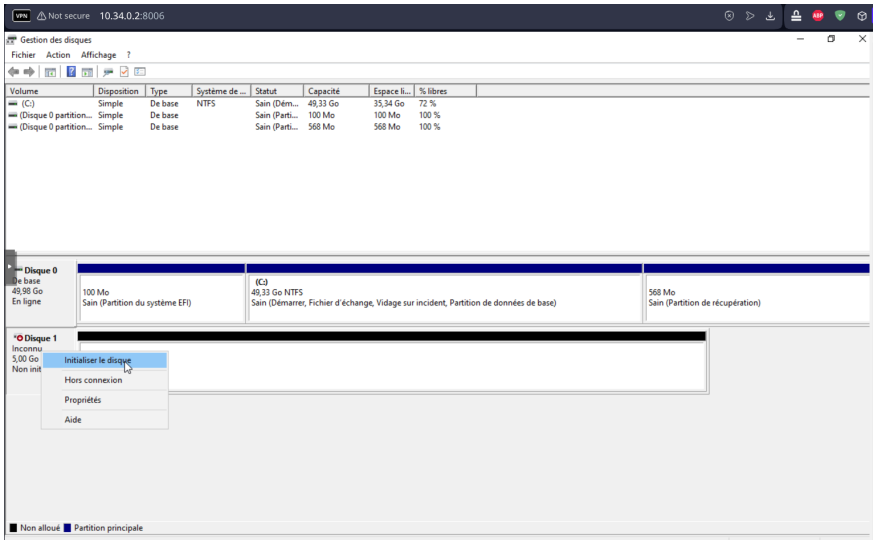
V.2 Configuration du stockage partagé

Afin de dissocier les fichiers de données des fichiers système (ce qui évite de bloquer le serveur en cas de saturation de l'espace disque), un second disque virtuel de 5 Go a été rattaché à la machine virtuelle.

Dans un premier temps, il est nécessaire de préparer le support physique au sein de l'utilitaire Gestion des disques. Le nouveau volume est détecté puis basculé sous le statut « en ligne » :



Une fois cette action effectuée, l'initialisation du disque est exécutée afin de créer la table de partition requise pour accueillir nos futurs répertoires :



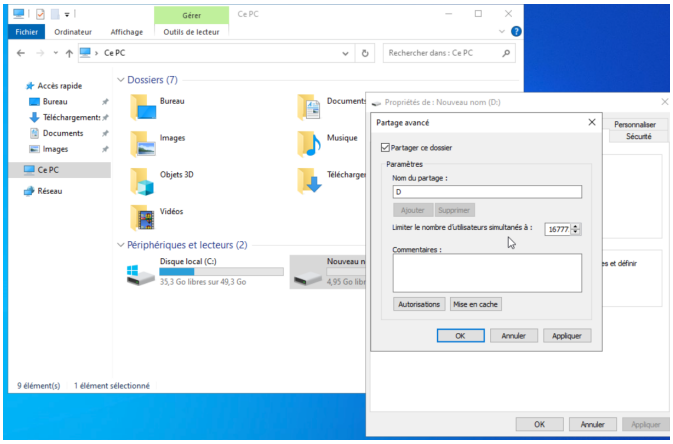
Une fois le disque initialisé, il faut lui attribuer une existence logique dans le système de fichiers de Windows Server avant de pouvoir le diffuser sur le réseau.

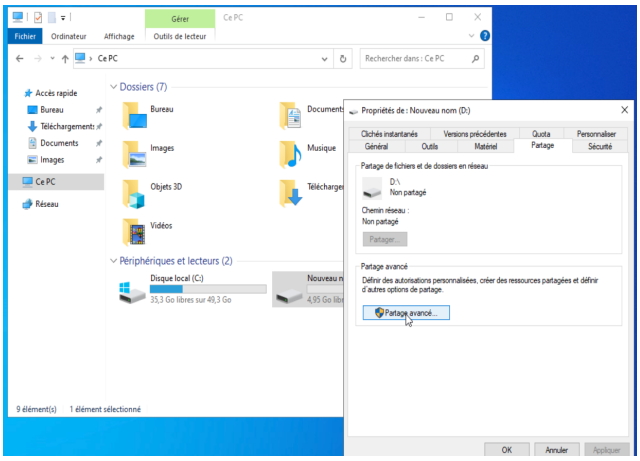
Nous procédons donc au formatage du volume en sélectionnant le système de fichiers NTFS (adapté pour la gestion fine des droits de sécurité). Lors de cette étape, nous lui attribuons la lettre de lecteur D: afin de le distinguer clairement du disque principal C: contenant le système d'exploitation.

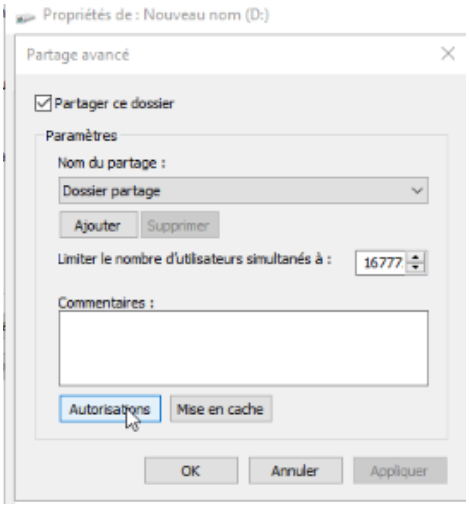
Le volume D: étant prêt à l'emploi, l'étape suivante consiste à le rendre accessible pour les utilisateurs de la filiale. Pour cela, nous configurons le partage du disque D: en effectuant un clic droit sur le lecteur, puis en nous rendant dans l'onglet Partage et en cliquant sur "Partage avancé".

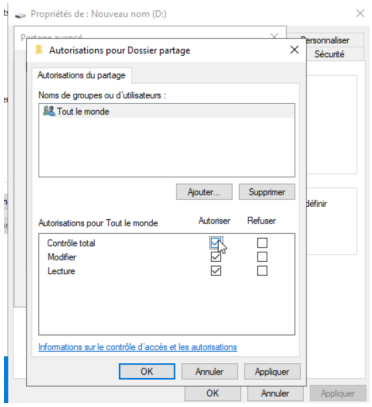
C'est dans ce menu que nous définissons :

- Le nom du partage : Nous le nommons explicitement "Dossier partage".
- Les autorisations du partage : Suivant les bonnes pratiques, nous configurons le groupe "Tout le monde" en Contrôle total. Ce choix volontairement permissif au niveau du protocole réseau (SMB) permet de centraliser la gestion réelle et restrictive des accès uniquement au niveau de l'onglet sécurité (permissions NTFS), évitant ainsi les conflits de droits cumulés.



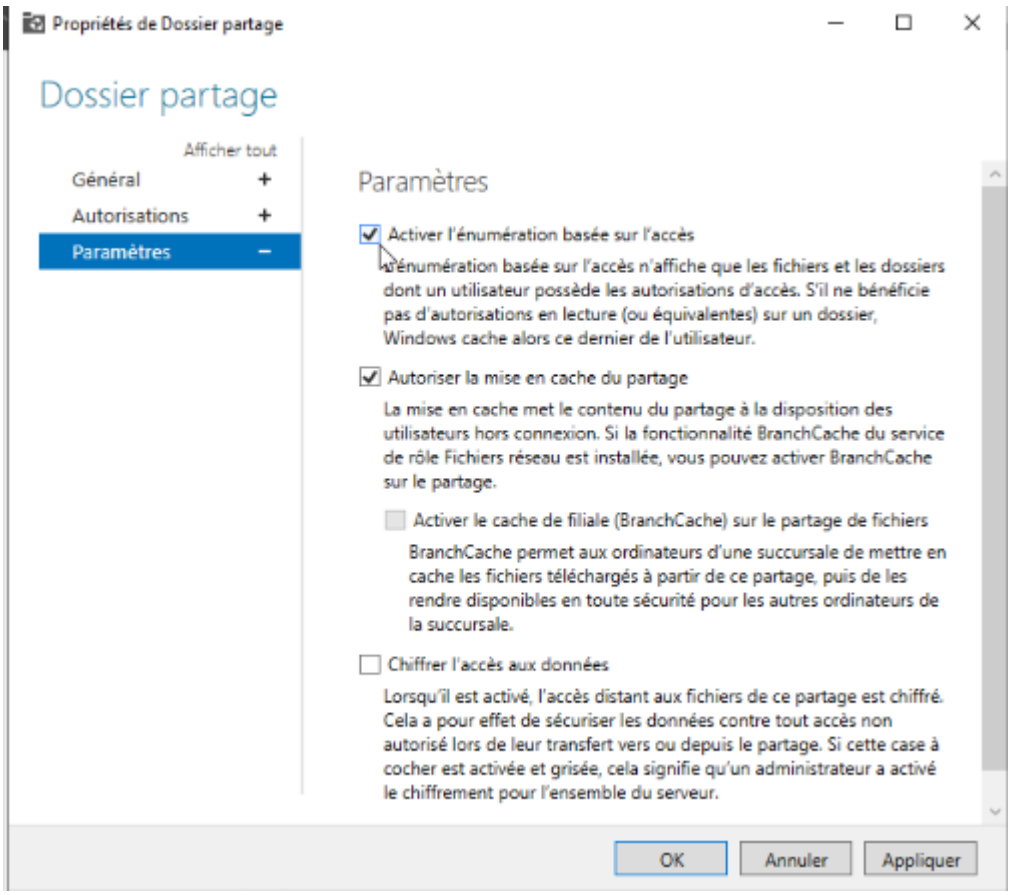






Lors de la configuration des propriétés avancées du partage, nous activons le paramètre d' « énumération basée sur l'accès » (*Access-Based Enumeration* ou *ABE*). Cette fonctionnalité de sécurité garantit que les utilisateurs ne peuvent voir que les fichiers et dossiers pour lesquels ils disposent explicitement de droits de lecture.

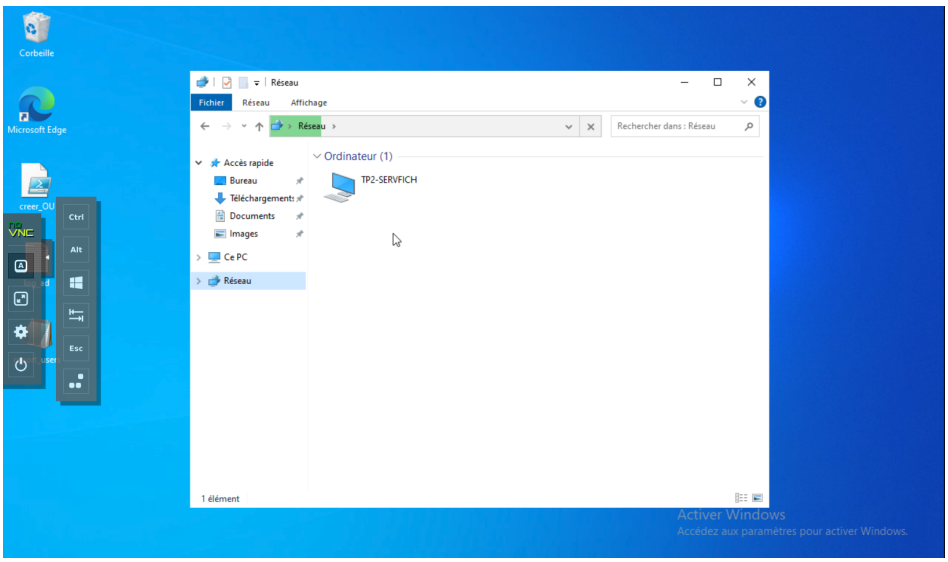
Si un utilisateur n'a pas les permissions requises sur un sous-dossier, celui-ci devient totalement invisible sur sa session.



V.3 Test

La configuration étant finalisée et validée sur le serveur, nous passons au test depuis un poste client, une machine windows 11 créée et configurée, sur le réseau des collaborateurs.

La vérification du chemin réseau « \\TP2-SERVFICH\Dossier partage » démontre l'accessibilité du service à travers le LAN. Ce test confirme d'une part que le mécanisme de résolution de noms (DNS) fonctionne correctement, et d'autre part que le protocole SMB autorise bien la connexion du poste de travail.


Corbeille
Microsoft Edge
creer_OU
Ctrl
Alt
Esc
Réseau
Fichier
Réseau
Affichage
Réseau
Rechercher dans : Réseau
Accès rapide
Bureau
Téléchargements
Documents
Images
Ce PC
Réseau
Ordinateur (1)
TP2-SERVFICH
1 élément
Activer Windows
Accédez aux paramètres pour activer Windows.

VI VPN Client-to-Site

Pour permettre aux collaborateurs itinérants d'accéder de manière sécurisée aux ressources internes de VitaBigPharma (notamment le serveur de fichiers et l'ERP Dolibarr), le déploiement d'une solution de réseau privé virtuel (VPN) est indispensable.

VI.1 Choix protocole

Trois choix s’offrent à nous dans OPNsense :

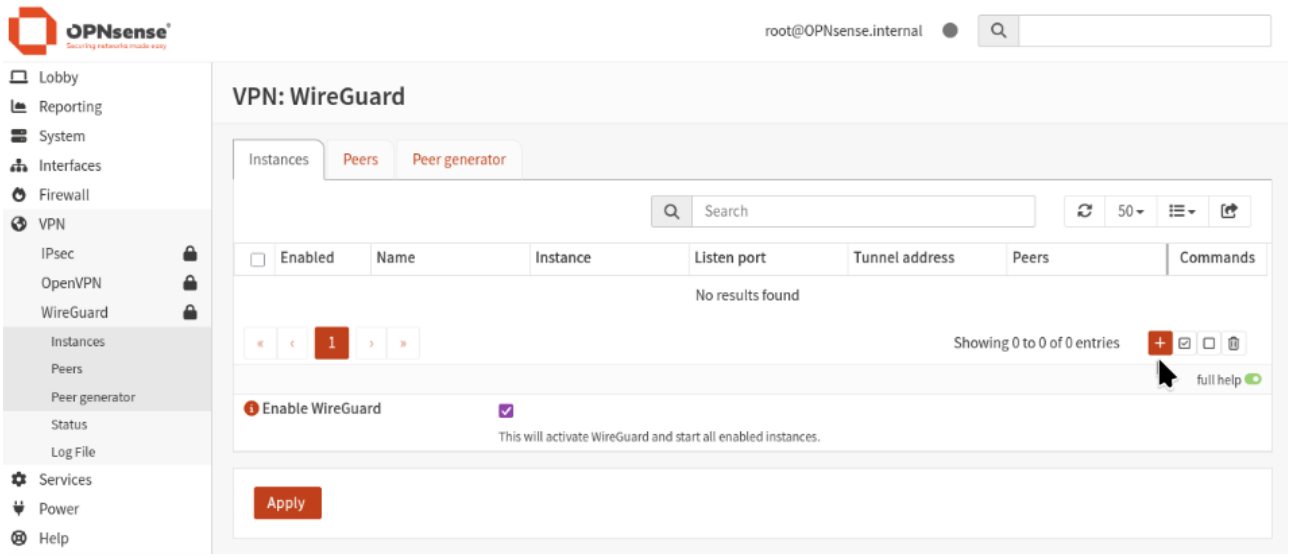
- OPNvpn : reference depuis des années, c’est un protocole éprouvé mais qui nécessite des manipulations et est un peu lourd en ressources CPU selon le chiffrement demandé
- Ipsec : robuste, il nécessite une configuration assez lourde. Plus adapté à du site to site qu’à un VPN roadwarrior
- Wireguard : protocole moderne, relativement peu de manipulations, et plus léger qu’OPNvpn coté CPU. Ce sera notre protocole de choix ici.

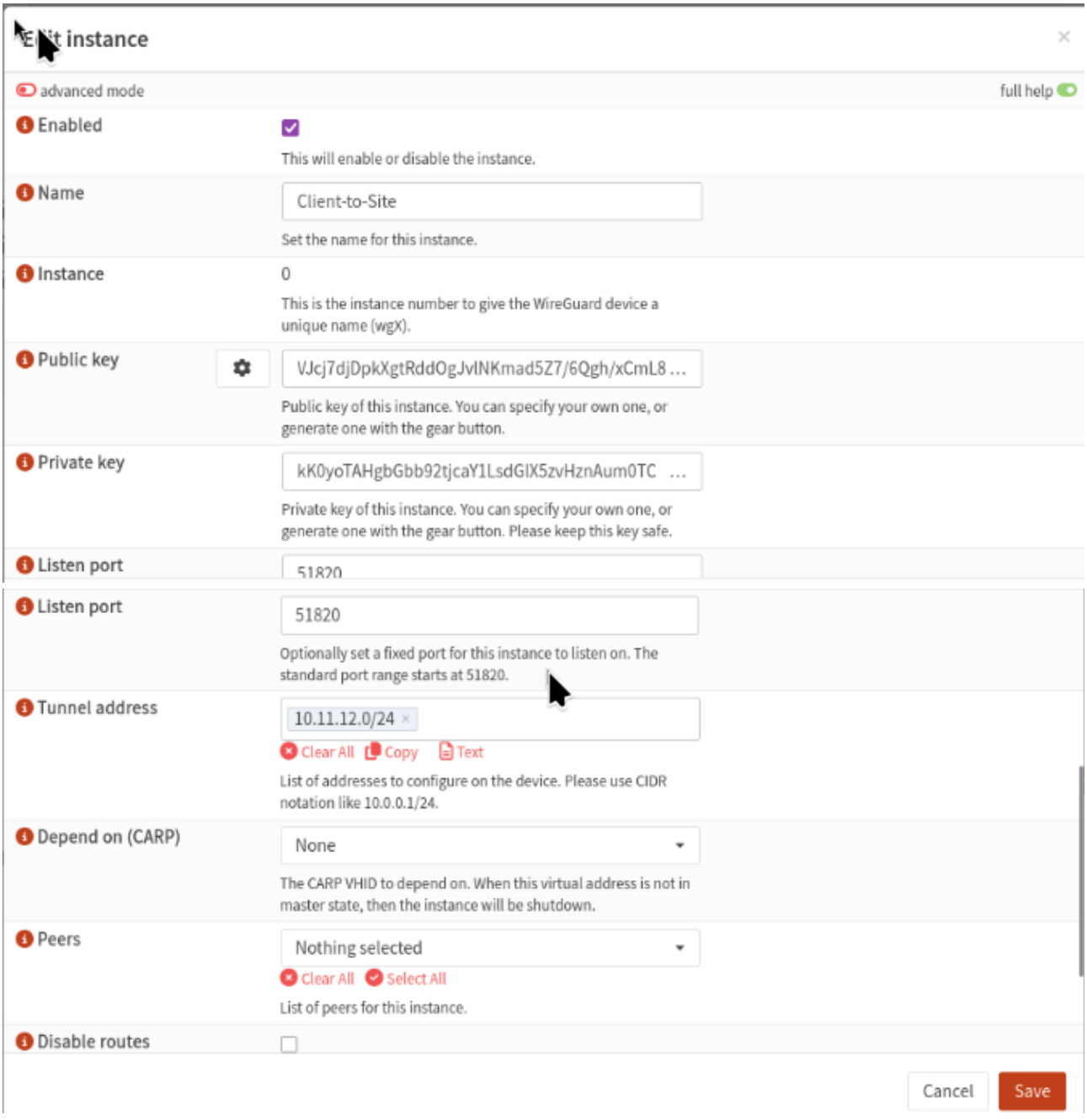
VI.2 Créer une instance

Nous nous appuierons sur la documentation officielle d’OPNsense pour mettre en place notre instance wireguard: https://docs.opnsense.org/manual/how-tos/wireguard-client.html.

La première étape de la mise en œuvre consiste à configurer l'instance "serveur" de WireGuard sur OPNSense.

C'est cette instance qui va écouter les requêtes de connexion et gérer le chiffrement global. La procédure suivante décrit la génération de la clé privée et de la clé publique du serveur, ainsi que la définition du tunnel logique réseau dédié aux clients VPN :



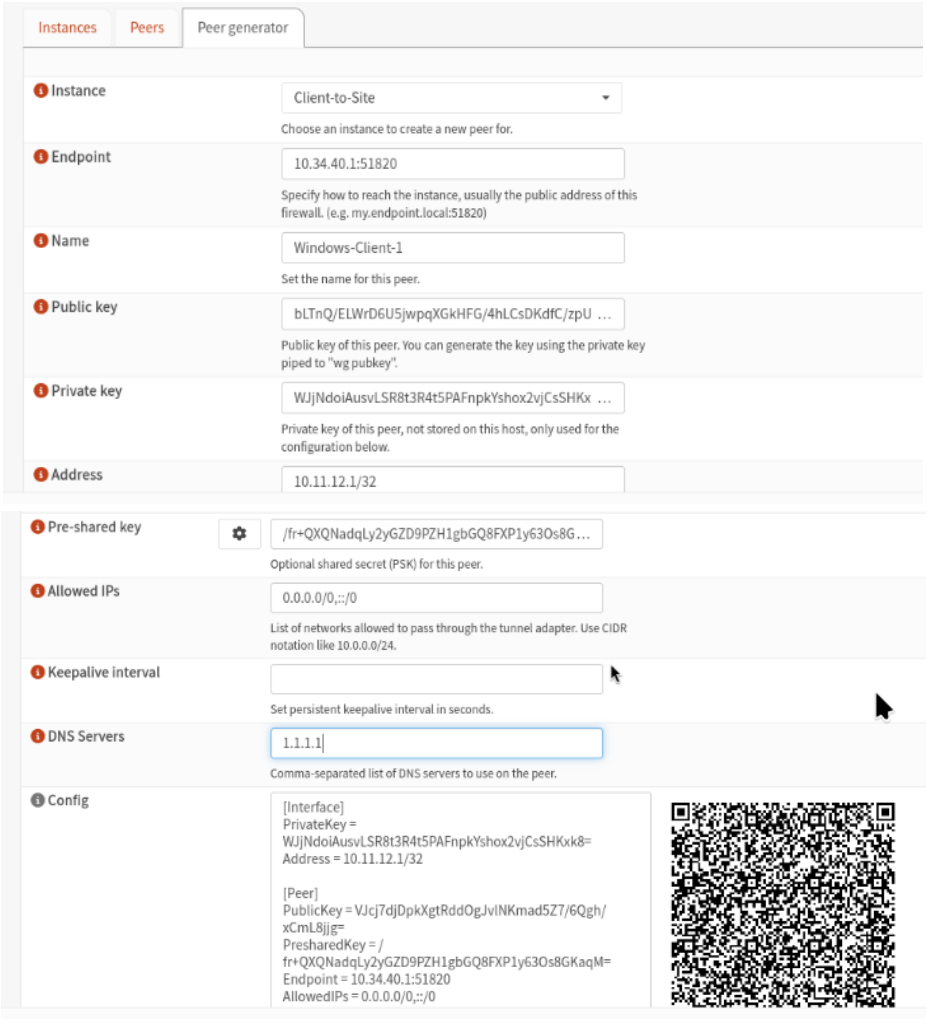


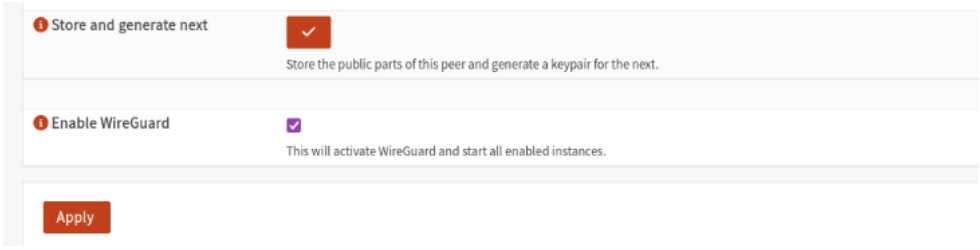
note : on n’expose pas sa clé privée normalement.

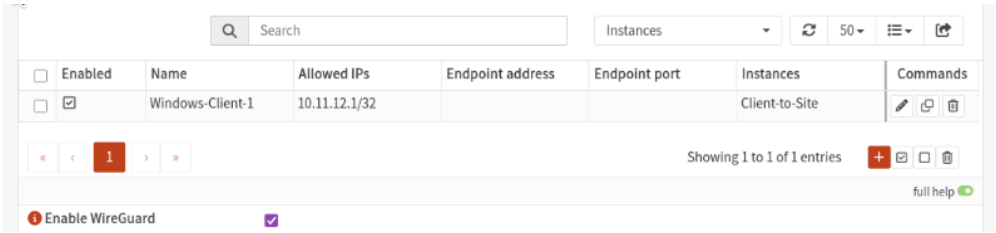
VI.3 Créer client

Le serveur VPN étant opérationnel, il faut désormais déclarer les utilisateurs distants autorisés à s'y connecter.

Pour chaque client, un échange de clés cryptographiques est effectué et une adresse IP fixe au sein du sous-réseau VPN lui est attribuée pour assurer la traçabilité des flux. Voici les étapes de création de notre premier utilisateur nomade :



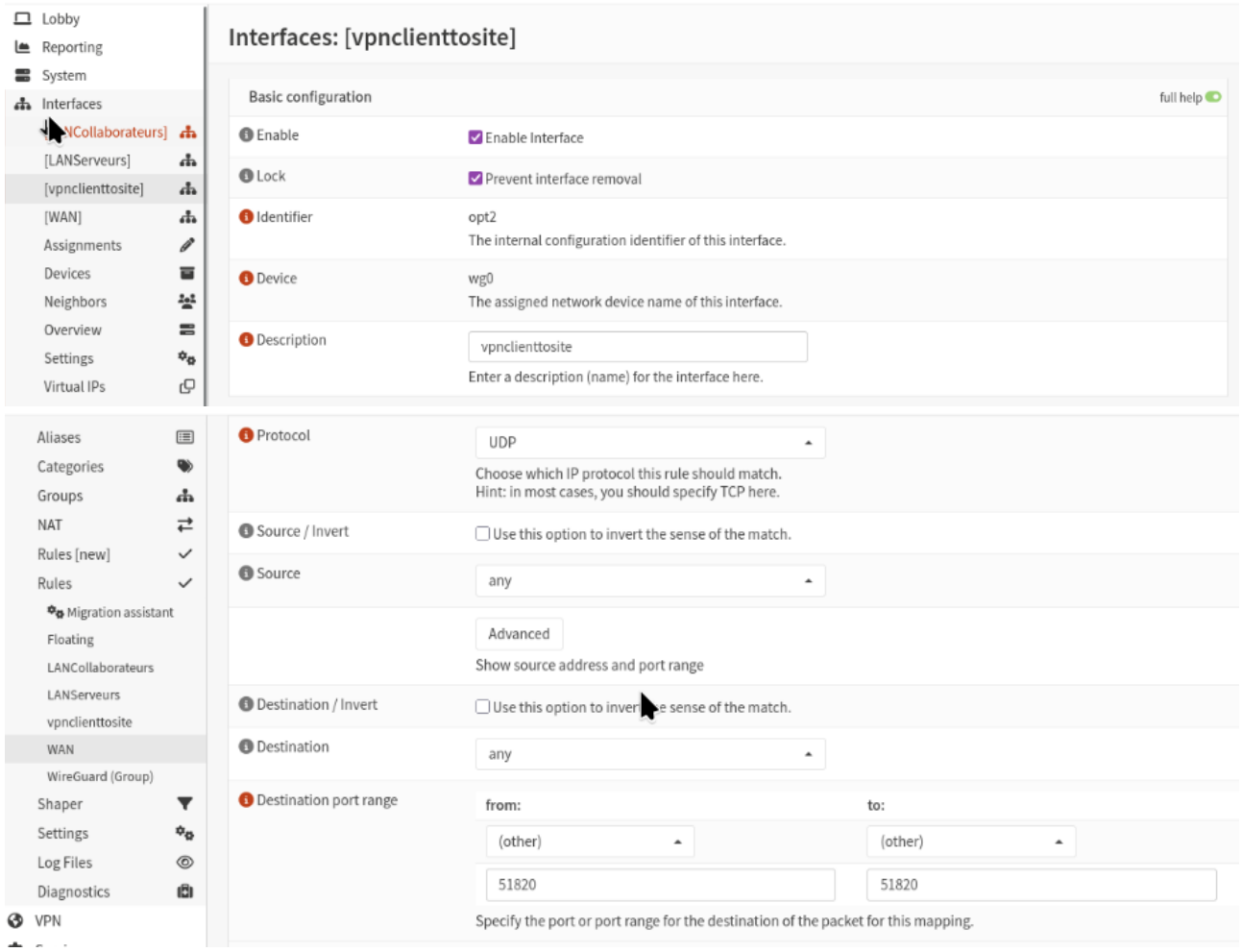


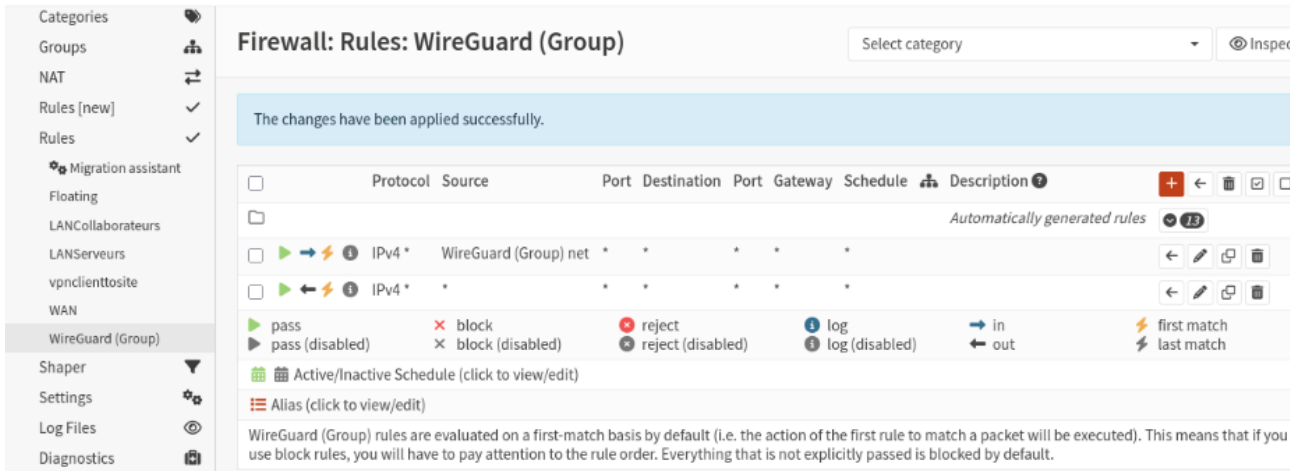


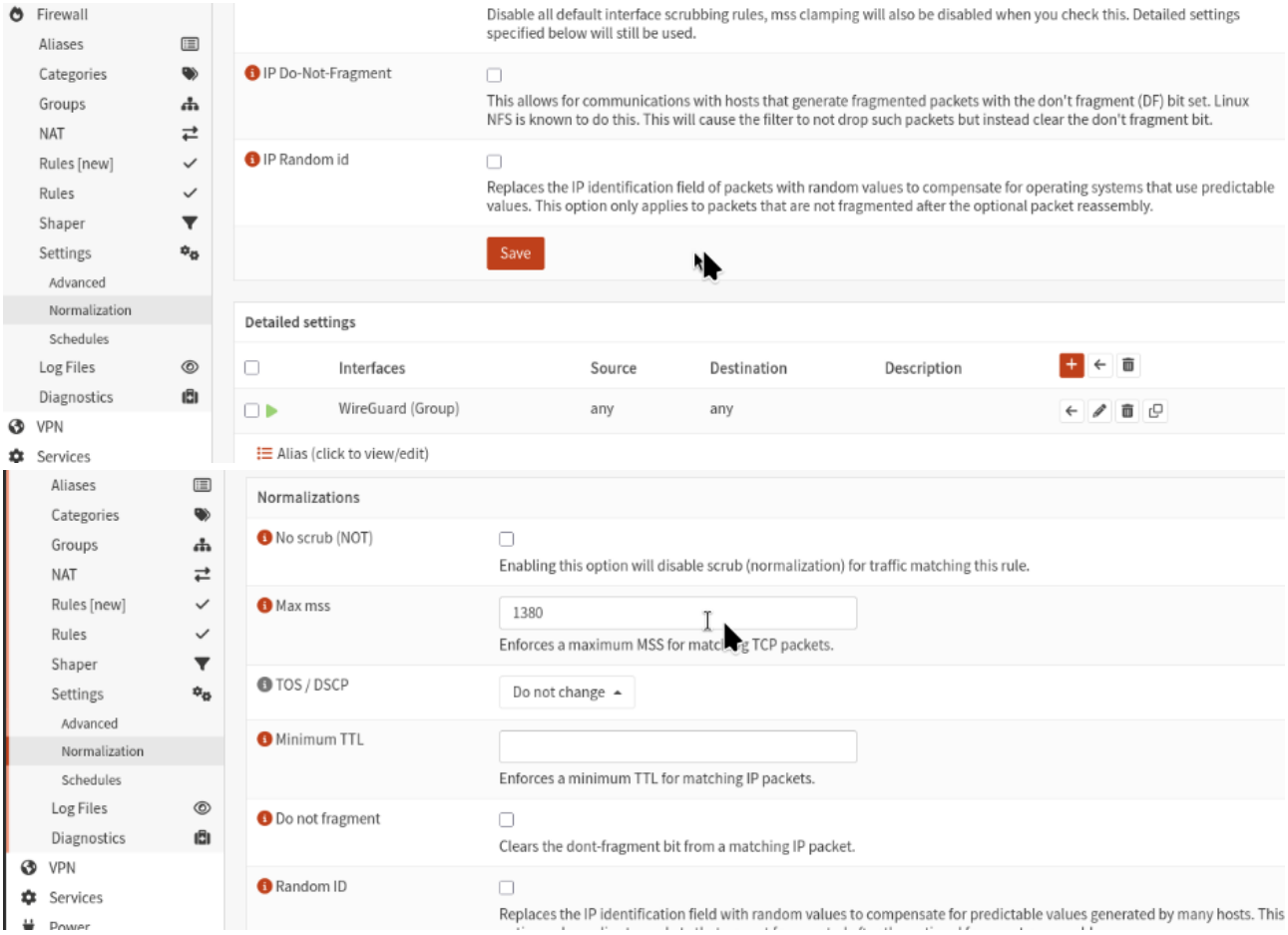
VI.4 Configurer interface et pare feu

Pour qu'OPNSense puisse acheminer et filtrer le trafic en provenance du tunnel VPN, ce dernier doit être rattaché à une interface réseau dédiée.

Par ailleurs, il est impératif d'ajuster la politique de sécurité du pare-feu. Les manipulations ci-dessous détaillent d'une part l'activation de l'interface virtuelle WireGuard, et d'autre part l'ouverture du port UDP 51820 sur l'interface WAN pour autoriser les flux chiffrés entrants :







VI.5 Configurer client (windows)

Installer le client

copier la configuration

connecter

La dernière phase consiste à déployer et configurer le logiciel client officiel WireGuard sur un poste Windows utilisateur.

Nous procédons à l'importation du fichier de configuration (.conf) généré par OPNSense, contenant les clés et l'adresse de la passerelle WAN. Les étapes suivantes illustrent la configuration du client et la validation de l'établissement du tunnel sécurisé, permettant au collaborateur d'accéder aux serveurs de Toulouse comme s'il était physiquement sur le site.

VII ERP

Pour centraliser la gestion des activités de Vita Big Pharma (gestion des stocks, suivi des fournisseurs, catalogue des compléments alimentaires, etc.), le déploiement d'un Progiciel de Gestion Intégré (PGI / ERP) est nécessaire.

Afin d'adopter une démarche moderne, agile et facilement maintenable, nous avons fait le choix d'adosser cette solution métier à une architecture conteneurisée.

VII.1 Choix de la solution

Pour le déploiement de notre solution applicative, deux approches d'architecture système majeures ont été étudiées : l'installation traditionnelle dite "bare-metal" s'appuyant sur une pile LAMP (Linux, Apache, MySQL, PHP) locale, et l'architecture moderne conteneurisée via Docker.

Afin de valider l'orientation technique du projet pour le siège de VitaBigPharma, une analyse comparative a été menée selon trois critères critiques :

- déploiement : une installation LAMP (Linux Apache Mysql PHP) classique exige la configuration manuelle, séquentielle et minutieuse de chaque composant (serveur web, serveur de base de données, extensions PHP spécifiques et gestion des permissions de fichiers)... qui s'avère chronophage et propice aux erreurs humaines. À l'inverse, Docker permet de packager l'intégralité de la pile applicative dans des environnements isolés (conteneurs). Le déploiement devient standardisé, reproductible à l'infini et s'exécute de manière automatisée en quelques secondes à l'aide d'un simple fichier de configuration.
- maintenabilité : sur un serveur LAMP traditionnel, l'application est dépendante des versions des paquets et des bibliothèques système installées sur l'OS hôte. Une mise à jour globale du système ou de PHP peut ainsi casser des dépendances et rendre l'ERP indisponible (on appelle ça « *dependency hell* »). Avec Docker, l'application et sa base de données sont totalement étanches vis-à-vis du système d'exploitation de la machine virtuelle. Les montées de version de l'application se résument au changement d'un simple tag d'image, sécurisant et simplifiant l'exploitation courante.
- sauvegarde et restauration : dans un environnement LAMP, les données sont dispersées (fichiers de configuration dans /etc, répertoires web dans /var/www, données MySQL dans /var/lib/mysql), rendant les scripts de sauvegarde complexes à maintenir. Sous Docker, l'utilisation de volumes isolés centralise les données persistantes. L'exécution de sauvegardes (comme un export de base de données directement à l'intérieur du conteneur) est standardisée et portable, ce qui optimise l'efficacité d'un futur Plan de Reprise d'Activité (PRA).

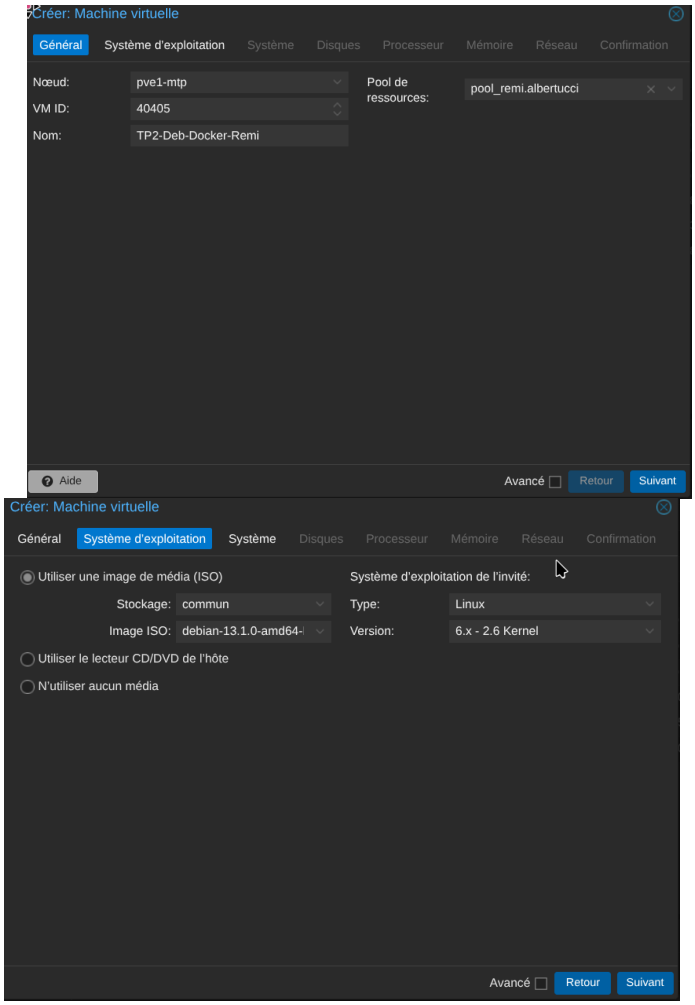
C'est pour cette agilité opérationnelle, cette sécurité et cette flexibilité de maintenance que la conteneurisation sous Docker a été retenue.

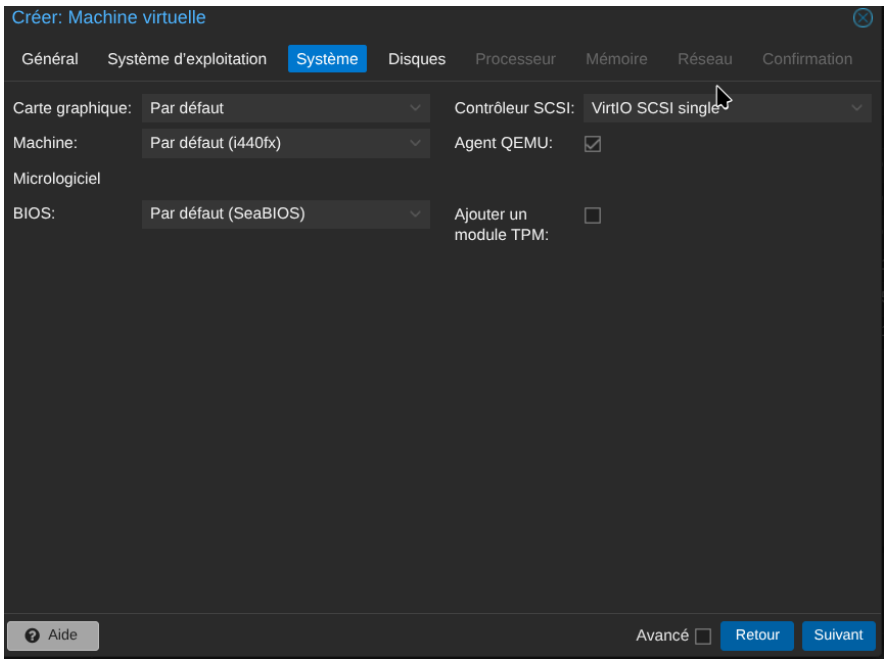
Une fois l'architecture technique validée, le choix du progiciel s'est porté sur Dolibarr pour piloter l'activité de l'entreprise. Cet ERP/CRM open-source se distingue par sa gratuité totale (absence de coûts de licence).

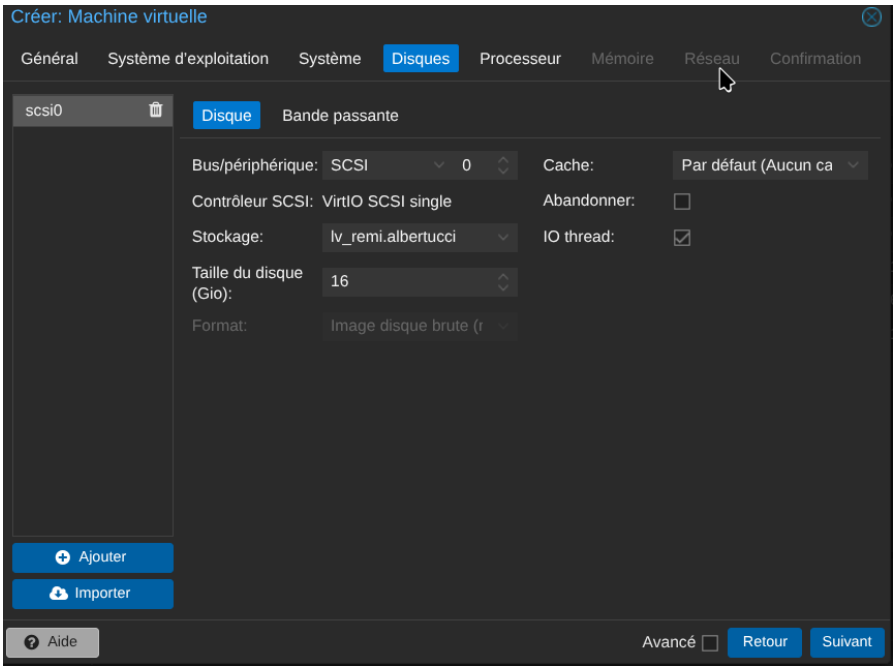
Sa force majeure réside dans sa structure modulaire : il permet d'activer à la carte uniquement les fonctionnalités nécessaires à VitaBigPharma (stocks, finances...), garantissant une interface épurée et performante pour les collaborateurs. Enfin, sa légèreté matérielle, sa prise en main intuitive via un navigateur web et sa communauté active en font la solution idéale pour assurer la pérennité du système d'information de l'entreprise.

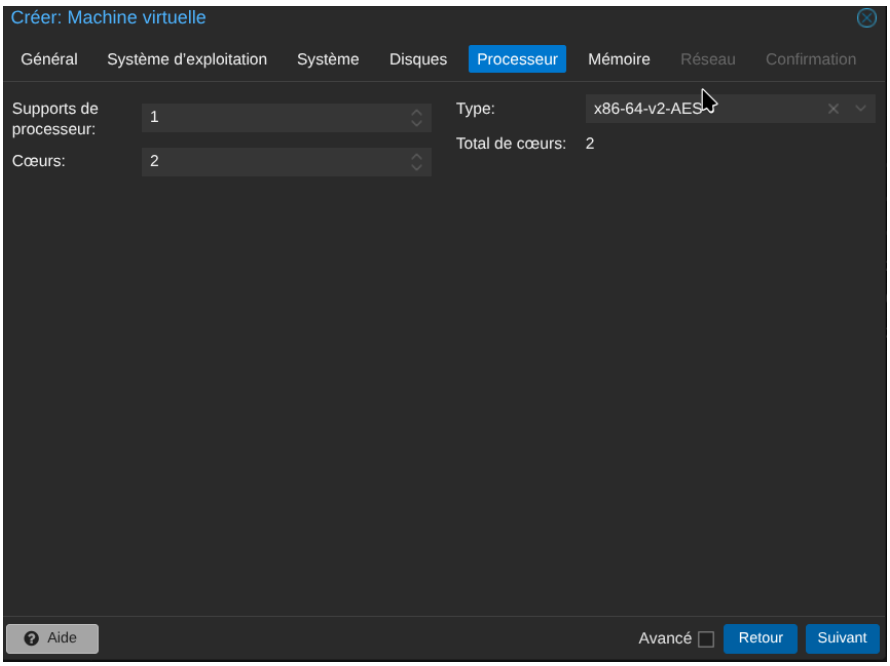
VII.2 Creation VM

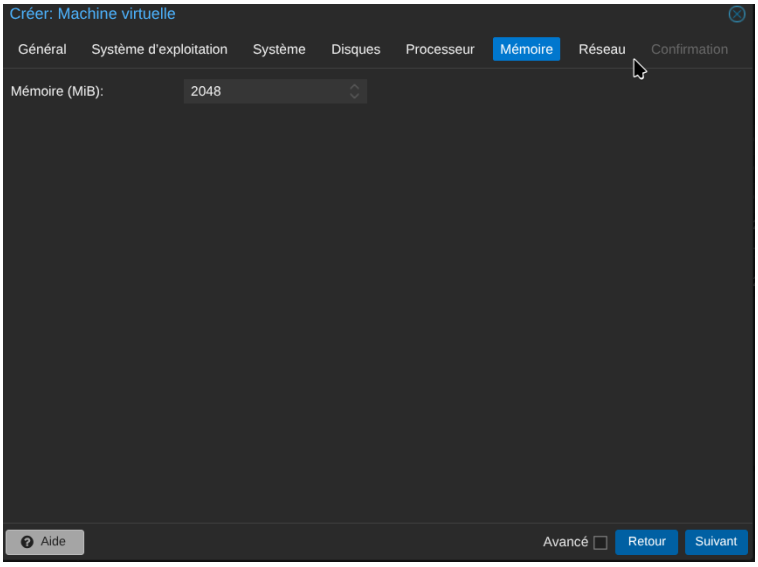
Nous devrons tout d'abord créer une machine Debian pour accueillir docker et notre conteneur :

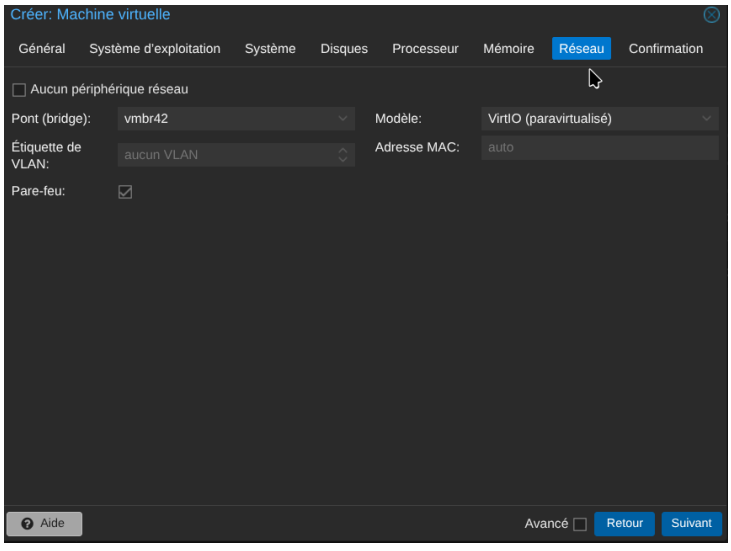


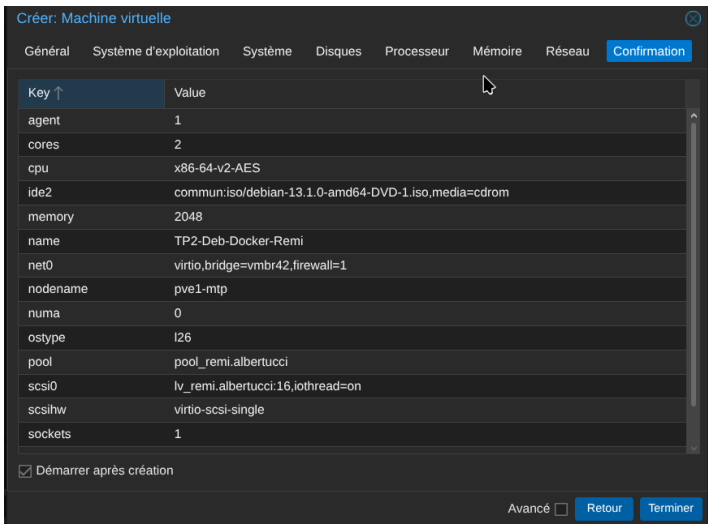

Créer: Machine virtuelle
Général
Système d'exploitation
Système
Disques
Processeur
Mémoire
Réseau
Confirmation
Carte graphique:
Par défaut
Contrôleur SCSI:
VirtIO SCSI single
Machine:
Par défaut (i440fx)
Agent QEMU:
Micrologiciel
BIOS:
Par défaut (SeaBIOS)
Ajouter un module TPM:
Aide
Avancé
Retour
Suivant


Créer: Machine virtuelle
Général
Système d'exploitation
Système
Disques
Processeur
Mémoire
Réseau
Confirmation
scsi0
Disque
Bande passante
Bus/périphérique:
SCSI
0
Cache:
Par défaut (Aucun ca
Contrôleur SCSI:
VirtIO SCSI single
Abandonner:
Stockage:
lv_remi.albertucci
IO thread:
Taille du disque (Gio):
16
Format:
Image disque brute (r
Ajouter
Importer
Aide
Avancé
Retour
Suivant


Créer: Machine virtuelle
Général
Système d'exploitation
Système
Disques
Processeur
Mémoire
Réseau
Confirmation
Supports de processeur:
1
Cœurs:
2
Type:
x86-64-v2-AES
Total de cœurs:
2
Aide
Avancé
Retour
Suivant

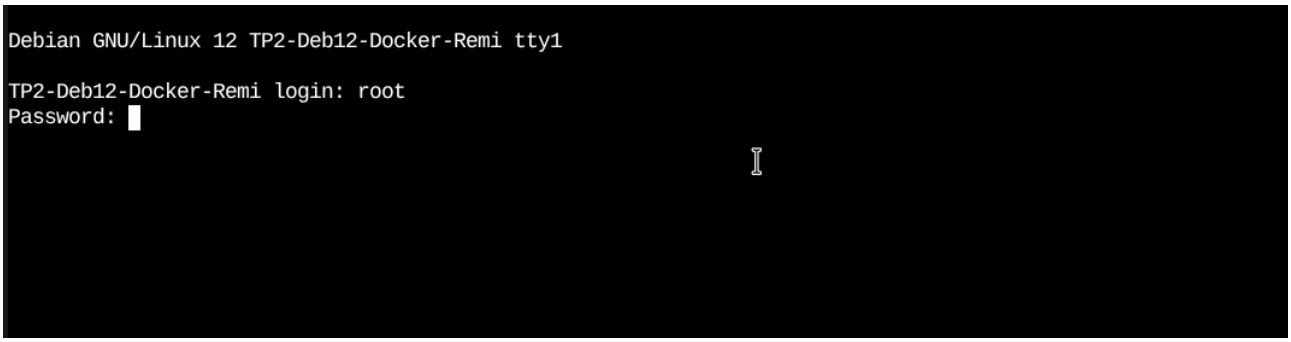

Créer: Machine virtuelle
Général
Système d'exploitation
Système
Disques
Processeur
Mémoire
Réseau
Confirmation
Mémoire (MiB):
2048
Aide
Avancé
Retour
Suivant


Créer: Machine virtuelle
Général
Système d'exploitation
Système
Disques
Processeur
Mémoire
Réseau
Confirmation
Aucun périphérique réseau
Pont (bridge):
vmbr42
Modèle:
VirtIO (paravirtualisé)
Étiquette de VLAN:
aucun VLAN
Adresse MAC:
auto
Pare-feu:
Aide
Avancé
Retour
Suivant



VII.3 Configuration VM

Une fois créée, on se connecte bien sur et nous allons faire les mises à jour, fixer l'ip, créer un user et le mettre en sudo.



Mise a jour

Simplement,

```
apt update && apt upgrade -y
```

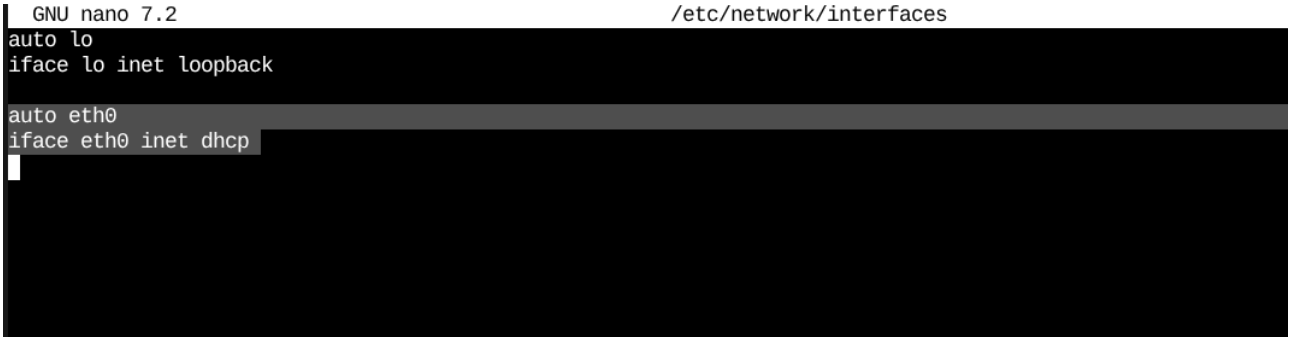
Fixer Ip

Le serveur ayant pour vocation d'héberger une application métier critique accessible par l'ensemble des collaborateurs du site, il ne peut dépendre d'une attribution DHCP.

Nous procédons donc à la fixation d'une adresse IP statique en modifiant le fichier de configuration réseau principal (/etc/network/interfaces). L'adresse attribuée est choisie de manière stricte au sein de la plage réservée à notre zone LAN Serveurs, en y renseignant également la passerelle (l'IP d'OPNSense) et le serveur DNS (notre contrôleur de domaine Active Directory). On édite donc le fichier :

```
nano /etc/network/interfaces
```

On modifie ensuite notre carte réseau, que l'on a pu vérifier avant avec la commande ip a au besoin, et de rentrer la configuration suivante :




```
auto eth0
iface ens18 inet static
    address 192.168.100.110
    netmask 255.255.255.0
    gateway 192.168.100.1
    dns-nameservers 1.1.1.1
```

Il suffit de redémarrer le service réseau avec la commande suivant puis de regarder si le changement a bien été pris en compte avec la commande ip a.

```
systemctl restart networking
```

Notons qu'il est également possible de fixer l'IP dans opnsense assez facilement.

Création utilisateur et ajout à sudo

Il convient ensuite de créer un utilisateur, de l'ajouter au groupe sudo pour pouvoir administrer la machine et de se connecter à ce compte :

```
apt install sudo -y
adduser <USER>
usermod -aG sudo <USER>
su <USER>
```

Configuration de sécurité

Pour accroître la sécurité de notre machine, il est possible d'installer unattended upgrades à ce moment pour recevoir les mise à jour de sécurité, fail2ban pour bloquer les tentatives de connexion par brute force ou encore de bloquer la connexion à root en ssh et de configurer une clé d'accès pour avoir une connexion sans mot de passe plus sécurisée.

VII.4 Installation de docker

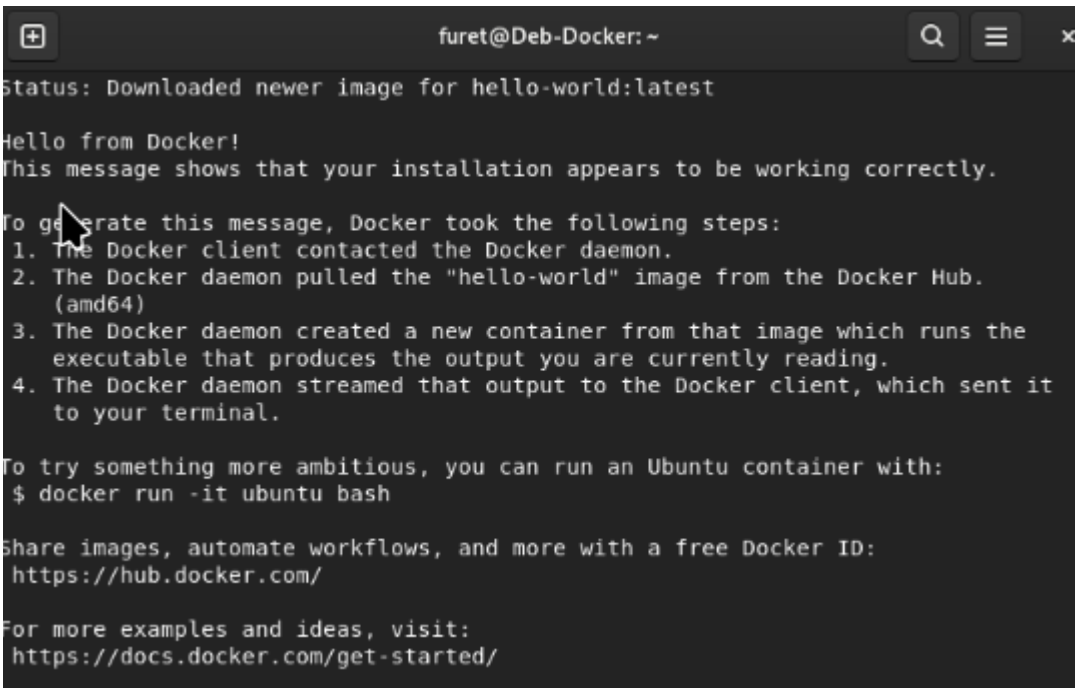
Nous nous appuierons sur la documentaiton officielle:
https://docs.docker.com/engine/install/debian/

Il suffit de suivre la documentation pas à pas, c’est à dire s’assurer que l’installation est vierge de toute installation précédente ou bloquante, avant d’installer les dépendances, d’ajouter les dépôts docker à la liste des dépôts sur notre machine et d’installer et lancer le paquet docker :

```
sudo apt remove $(dpkg --get-selections docker.io
docker-compose docker-doc podman-docker
containerd runc | cut -f1)
```

```
# Add Docker's official GPG key:
sudo apt update
sudo apt install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL
https://download.docker.com/linux/debian/gpg
-o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
sudo tee /etc/apt/sources.list.d/docker.sources
<<EOF
Types: deb
URIs:
https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo
"$VERSION_CODENAME")
Components: stable
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update

sudo docker run hello-world
```

Cette derniere commande devrrait nous renvoyer l’image suivante, preuve que tout est bien installé et configuré :



VII.5 Création conteneur

Nous allons maintenant ajouter notre utilisateur au groupe « docker » pour qu'il puisse l'utiliser :

```
sudo usermod -aG docker <USER>
cd ~
```

Par soucis d'organisation, nous allons créer un répertoire « docker » dans le dossier home de cet utilisateur car il administrera la machine. Dans ce dossier docker, nous y placerons un dossier « dolibarr » qui accueillera les fichiers de configuration de notre conteneur (config + compose) :

```
mkdir docker && mkdir docker/dolibarr
cd docker dolibarr && nano docker-
compose.yaml
```

Le docker compose est un « template » de notre installation. Grâce à lui, nous pouvons relancer le conteneur en une seule commande en cas de crash et de migration, sans changer les paramètres de configuration. On y renseigne notamment l'image utilisée (ici, latest puisque pas d'indication), les chemins de dossiers pour la persistance des données ainsi que les valeurs réseau si nécessaire ou toute autre variable d'environnement.

On l'édite dans notre cas de la façon suivante :

```
services:
  dolibarr_db:
    image: mariadb:10.11
    container_name : dolibarr
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
```

```
    MYSQL_DATABASE: dolibarr
    MYSQL_USER: dolibarr
    MYSQL_PASSWORD: dolibarr
  volumes:
    - db_data:/var/lib/mysql
  restart: unless-stopped

 dolibarr:
  image: dolibarr/dolibarr:latest
  environment:
    DOLI_DB_HOST: dolibarr_db
    DOLI_DB_NAME: dolibarr
    DOLI_DB_USER: dolibarr
    DOLI_DB_PASSWORD: dolibarr
    DOLI_ADMIN_LOGIN: admin
    DOLI_ADMIN_PASSWORD: admin123
    DOLI_URL_ROOT: http://192.168.100.110
  ports:
    - "80:80"
  depends_on:
    - dolibarr_db
  volumes:
    - dolibarr_data:/var/www/html/documents
  restart: unless-stopped

volumes:
 db_data:
 dolibarr_data:
```

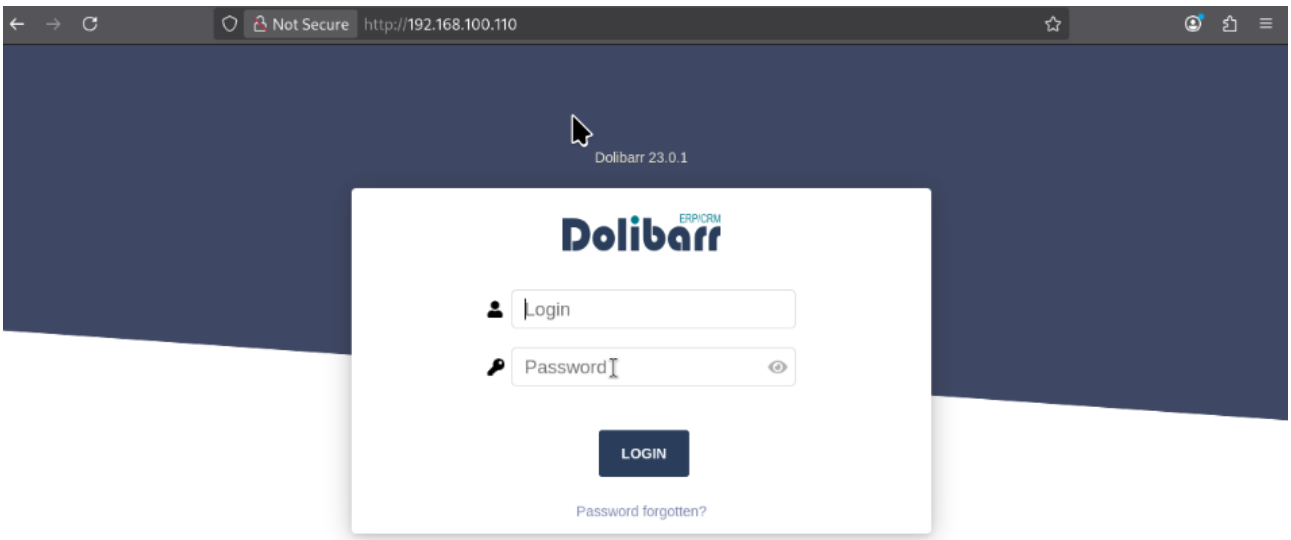
Pour lancer le conteneur, on se place dans le dossier où se trouve notre fichier yaml et on lance :

```
docker compose up -d
```

On peut vérifier que tout s'est bien déroulé avec la commande « docker ps ».

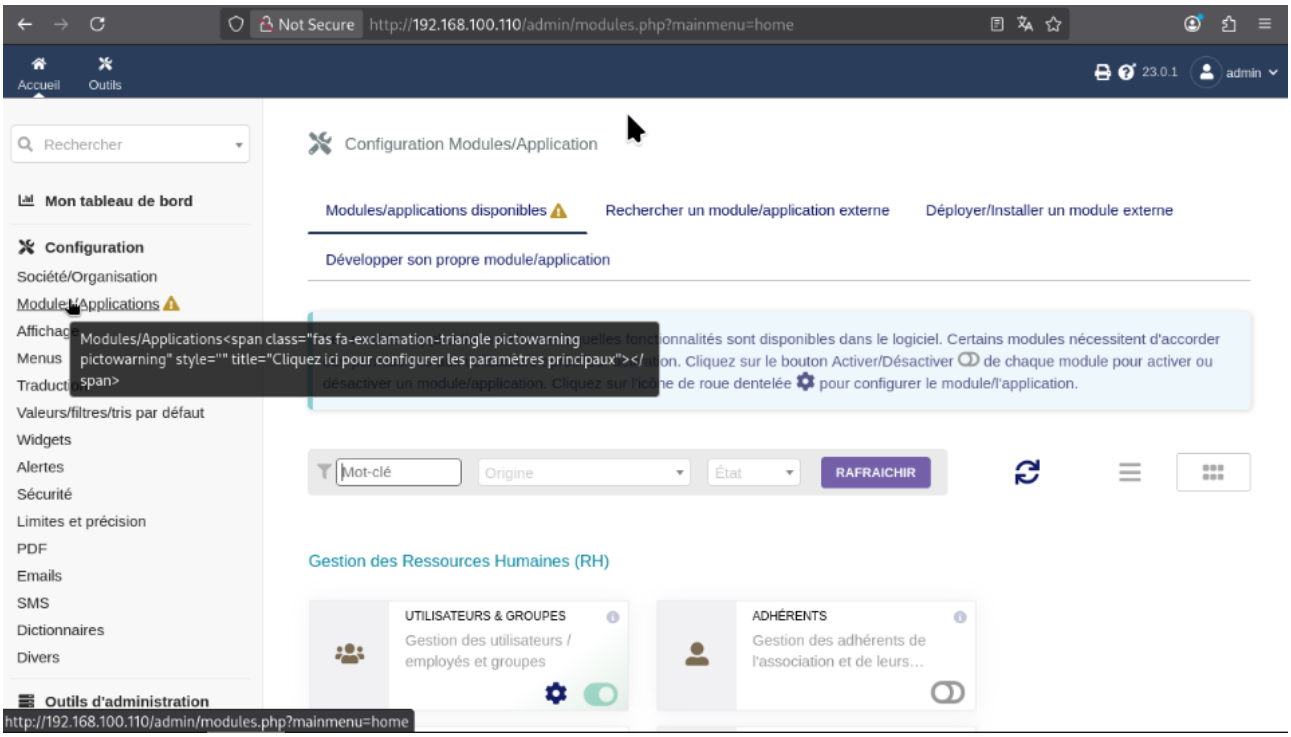
VII.6 Mise en route de dolibarr

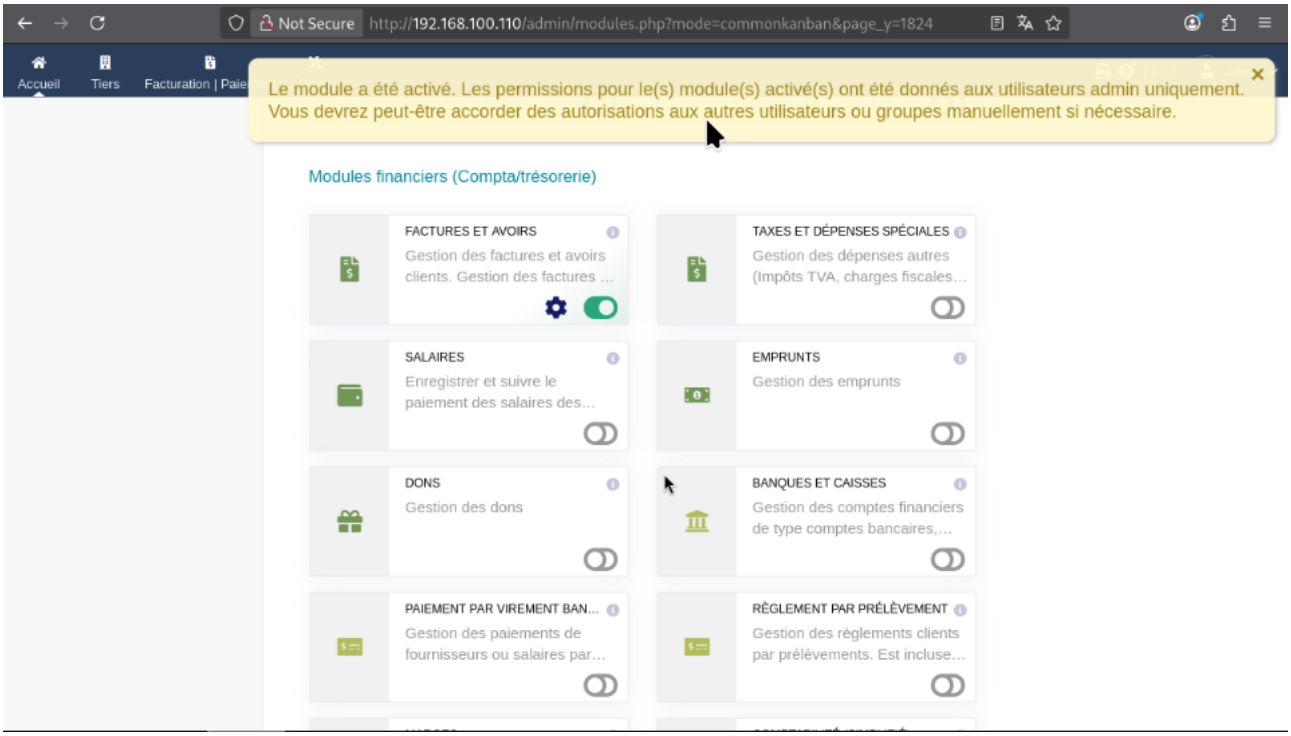
Notre installation étant prête, on peut maintenant se rendre sur l'addresse http://IP-DE-LA-VM pour accéder à l'interface de dolibarr :

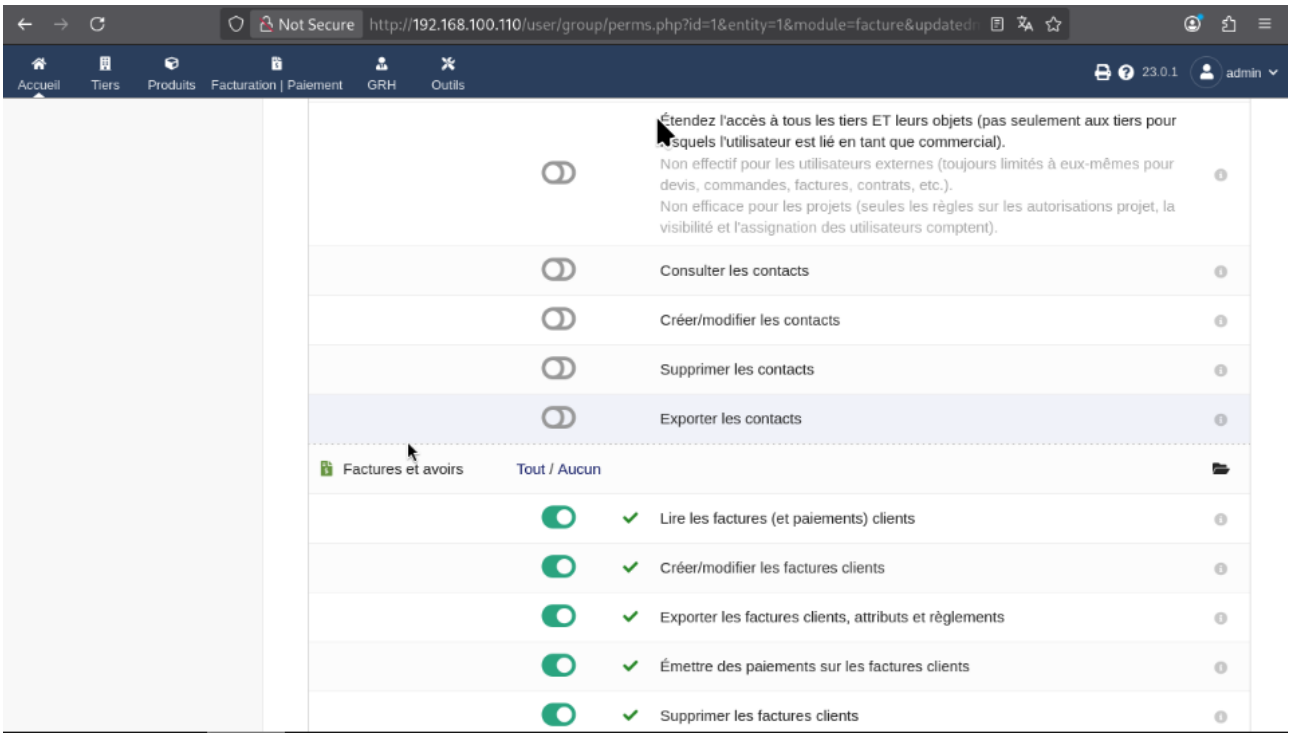


Modules

Tout d'abord, nous allons activer les modules qui nous serons utiles. Un module sert à ajouter des fonctionnalités, et même si on serait tenté de tous les activer, il faut se rappeler qu'un module activé consommera des ressources et offrira une nouvelle surface d'attaque sur notre machine.
Il est assez intuitif de naviguer dans dolibarr, vous pouvez suivre les étapes suivantes pour installer le module de facturation dans la version

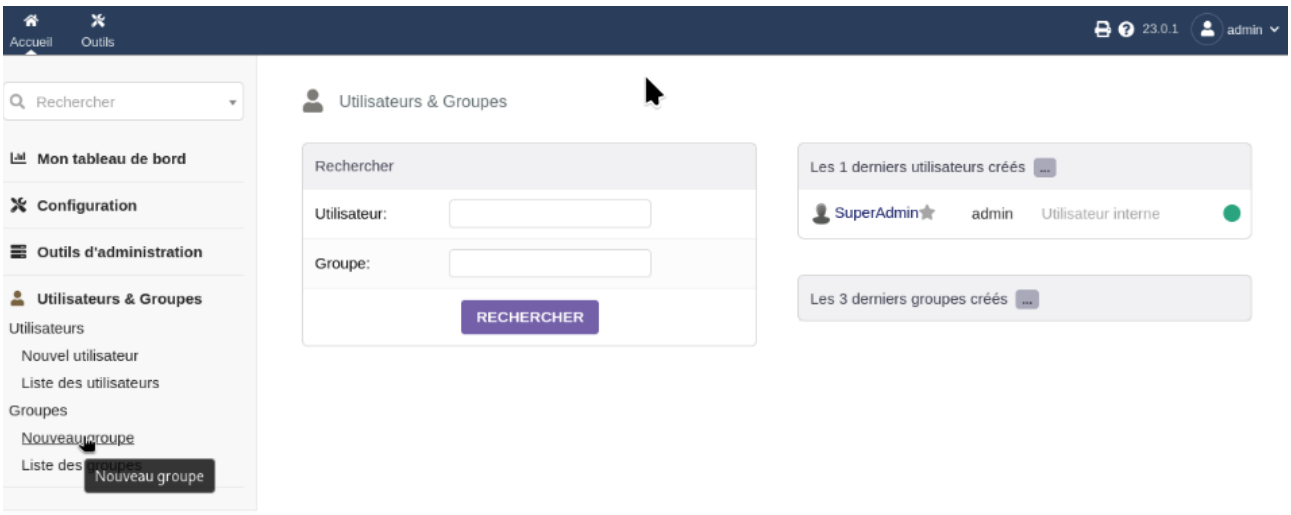


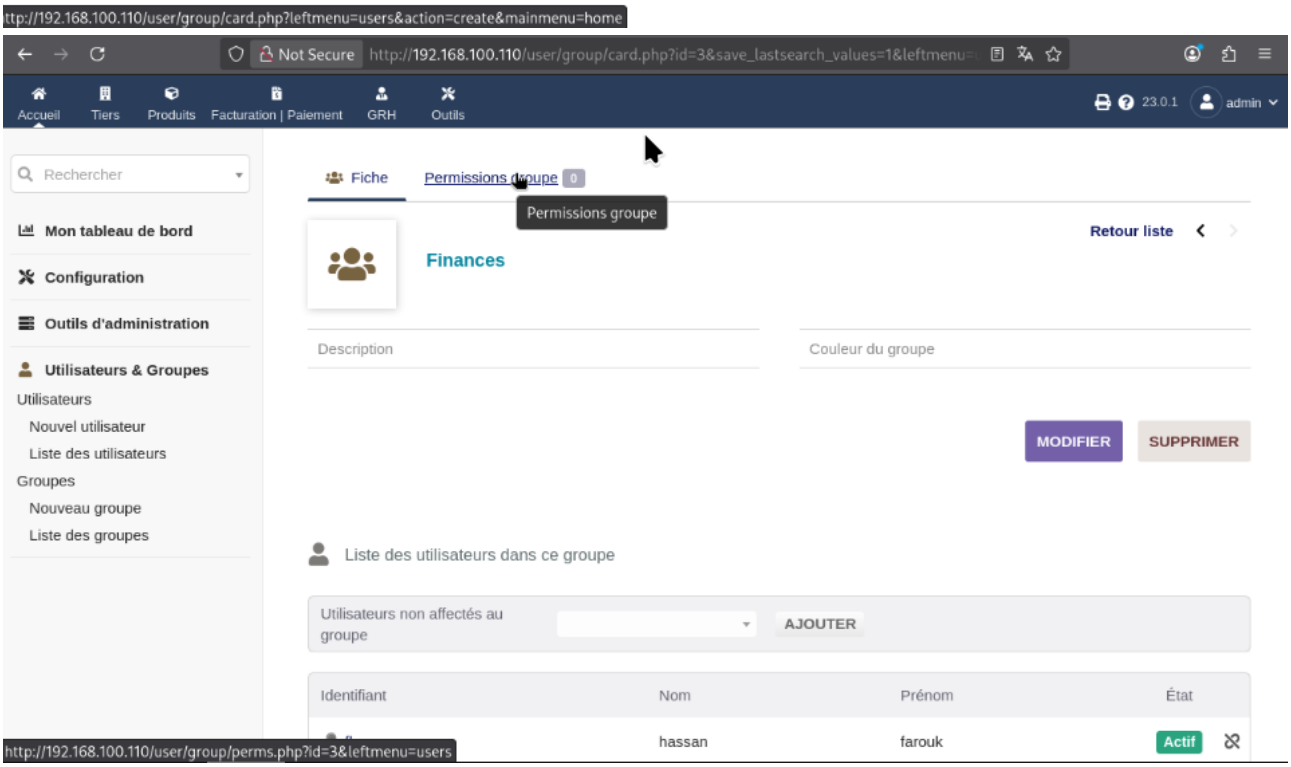


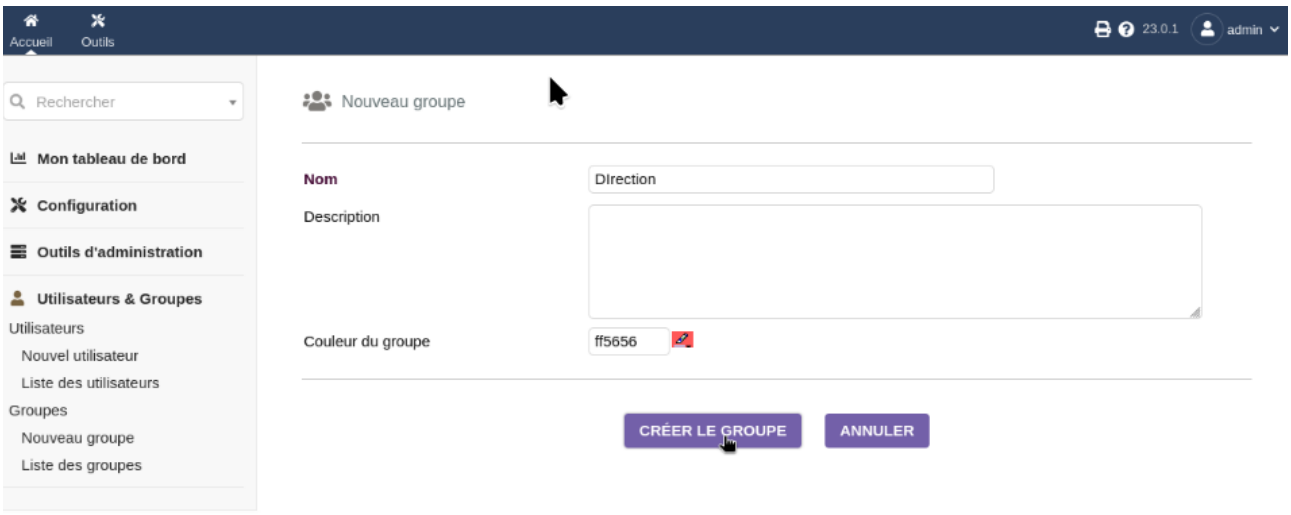


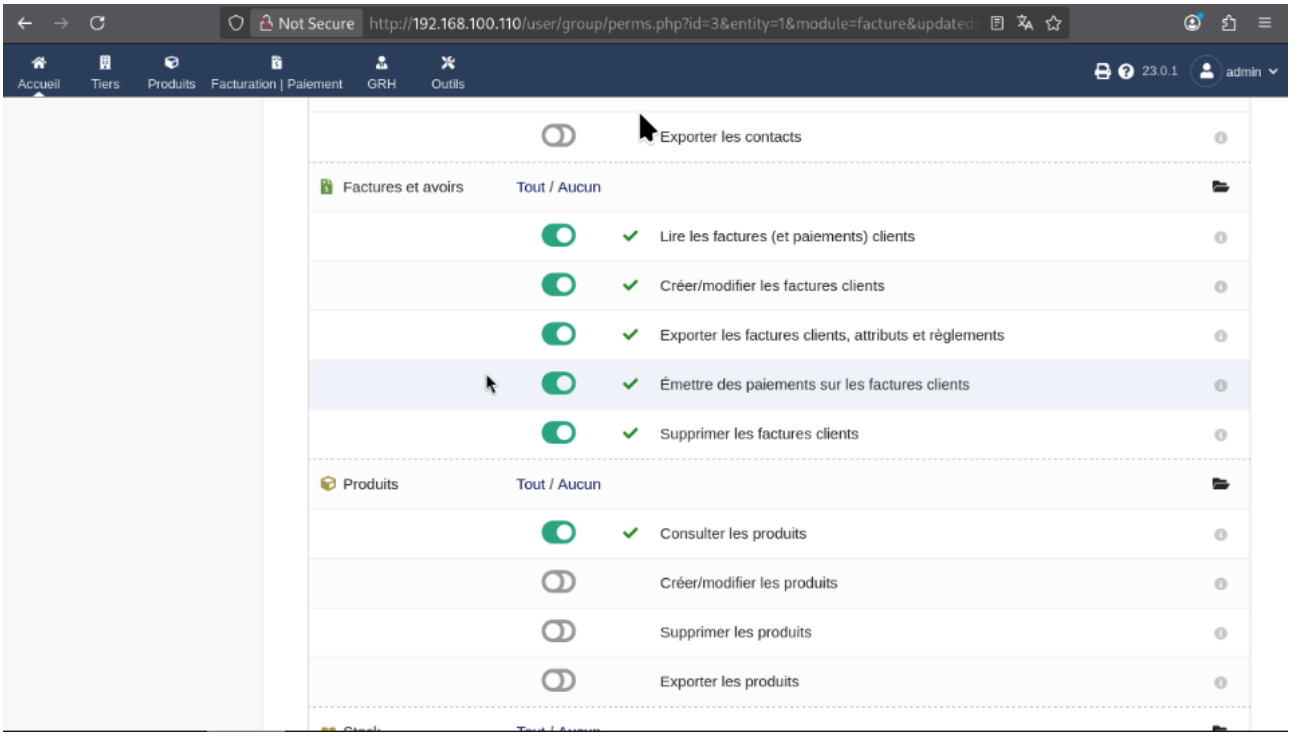
créer groupes

Il va maintenant falloir créer des groupes pour pouvoir segmenter nos futurs utilisateurs :


Accueil
Outils
23.0.1
admin
Rechercher
Mon tableau de bord
Configuration
Outils d'administration
Utilisateurs & Groupes
Utilisateurs
Nouvel utilisateur
Liste des utilisateurs
Groupes
Nouveau groupe
Liste des groupes
Nouveau groupe
Utilisateurs & Groupes
Rechercher
Utilisateur:
Groupe:
RECHERCHER
Les 1 derniers utilisateurs créés
SuperAdmin
admin
Utilisateur interne
Les 3 derniers groupes créés

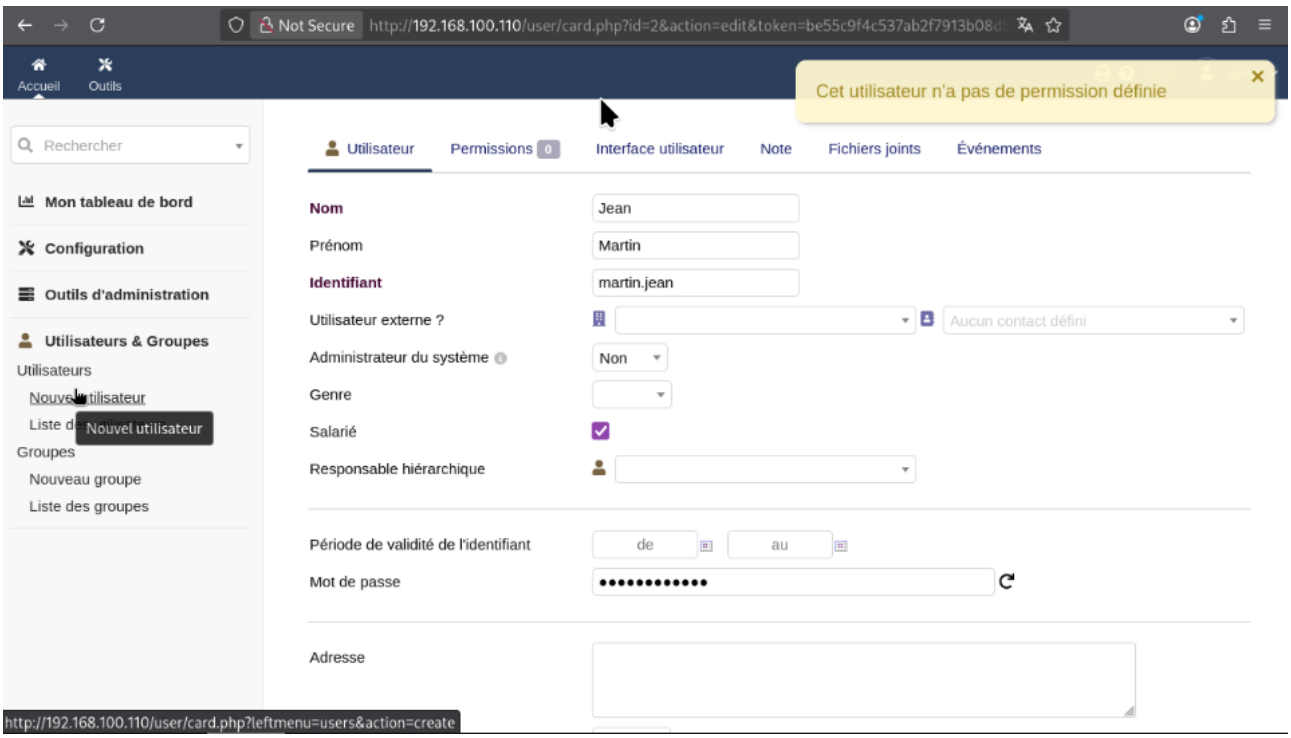

ttp://192.168.100.110/user/group/card.php?leftmenu=users&action=create&mainmenu=home
Not Secure
http://192.168.100.110/user/group/card.php?id=3&save_lastsearch_values=1&leftmenu=
Accueil
Tiers
Produits
Facturation | Paiement
GRH
Outils
23.0.1
admin
Rechercher
Mon tableau de bord
Configuration
Outils d'administration
Utilisateurs & Groupes
Utilisateurs
Nouvel utilisateur
Liste des utilisateurs
Groupes
Nouveau groupe
Liste des groupes
Fiche
Permissions groupe
Permissions groupe
Retour liste
Finances
Description
Couleur du groupe
MODIFIER
SUPPRIMER
Liste des utilisateurs dans ce groupe
Utilisateurs non affectés au groupe
AJOUTER
Identifiant
Nom
Prénom
État
hassan
farouk
Actif
http://192.168.100.110/user/group/perms.php?id=3&leftmenu=users

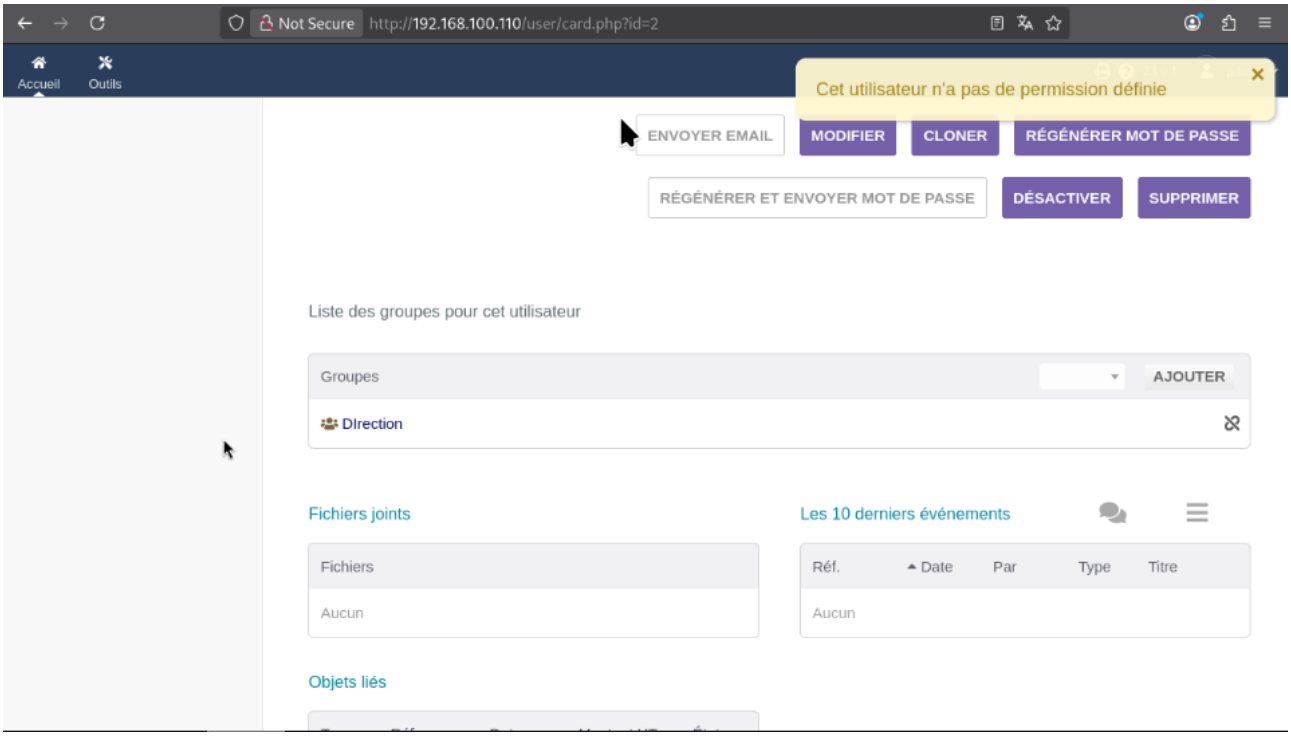




créer users

Les groupes créés, on peut maintenant créer des utilisateurs pour peupler notre ERP. L’étape est à faire pour chaque utilisateur :


Not Secure
http://192.168.100.110/user/card.php?id=2&action=edit&token=be55c9f4c537ab2f7913b08d
Accueil
Outils
Cet utilisateur n'a pas de permission définie
Rechercher
Mon tableau de bord
Configuration
Outils d'administration
Utilisateurs & Groupes
Utilisateurs
Nouvel utilisateur
Liste d
Nouvel utilisateur
Groupes
Nouveau groupe
Liste des groupes
Utilisateur
Permissions 0
Interface utilisateur
Note
Fichiers joints
Événements
Nom
Jean
Prénom
Martin
Identifiant
martin.jean
Utilisateur externe ?
Aucun contact défini
Administrateur du système
Non
Genre
Salarié
Responsable hiérarchique
Période de validité de l'identifiant
de
au
Mot de passe
Adresse
http://192.168.100.110/user/card.php?leftmenu=users&action=create


Not Secure
http://192.168.100.110/user/card.php?id=2
Accueil
Outils
Cet utilisateur n'a pas de permission définie
ENVOYER EMAIL
MODIFIER
CLONER
RÉGÉNÉRER MOT DE PASSE
RÉGÉNÉRER ET ENVOYER MOT DE PASSE
DÉSACTIVER
SUPPRIMER
Liste des groupes pour cet utilisateur
Groupes
AJOUTER
DIrection
Fichiers joints
Les 10 derniers événements
Fichiers
Aucun
Réf.
Date
Par
Type
Titre
Aucun
Objets liés

VIII Supervision

La supervision (ou monitoring) est un pilier fondamental dans la gestion d'une infrastructure informatique d'entreprise.

Elle consiste à collecter, analyser et afficher en temps réel les données de performance et d'état des différents équipements et services (taux d'utilisation du CPU, de la mémoire RAM, espace disque disponible, disponibilité des conteneurs).

L'objectif principal est de garantir le maintien en condition opérationnelle (MCO), d'anticiper les pannes par une approche proactive et de réduire le temps de diagnostic en cas d'incident.

Pour répondre à ce besoin sur notre bloc applicatif, notre choix s'est porté sur Prometheus. Il s'agit d'une solution de supervision open-source de référence, particulièrement optimisée pour les environnements conteneurisés.

Contrairement aux outils traditionnels (comme Nagios) qui fonctionnent par requêtes d'interrogations lourdes, Prometheus utilise un mécanisme de "pull" : il vient collecter (scrapper) périodiquement les métriques exposées au format HTTP par les conteneurs et la machine hôte.

VIII.1 Prometheus

Creation ct lxc et mise en route docker et conteneur

```
cd ~/docker && mkdir monitoring && cd monitoring
```

```
nano docker-compose.yaml
```

```
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    volumes:
      -
./prometheus.yml:/etc/prometheus/prometheus
.yml
    ports:
      - "9090:9090"
    restart: unless-stopped

  grafana:
    image: grafana/grafana:latest
```

```
    container_name: grafana
    ports:
      - "3000:3000"
    environment:
      GF_SECURITY_ADMIN_PASSWORD:
admin123
    volumes:
      - grafana_data:/var/lib/grafana
    restart: unless-stopped

  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    pid: host
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
    command:
      - '--path.procfs=/host/proc'
      - '--path.sysfs=/host/sys'
    ports:
      - "9100:9100"
    restart: unless-stopped

volumes:
  grafana_data:
```

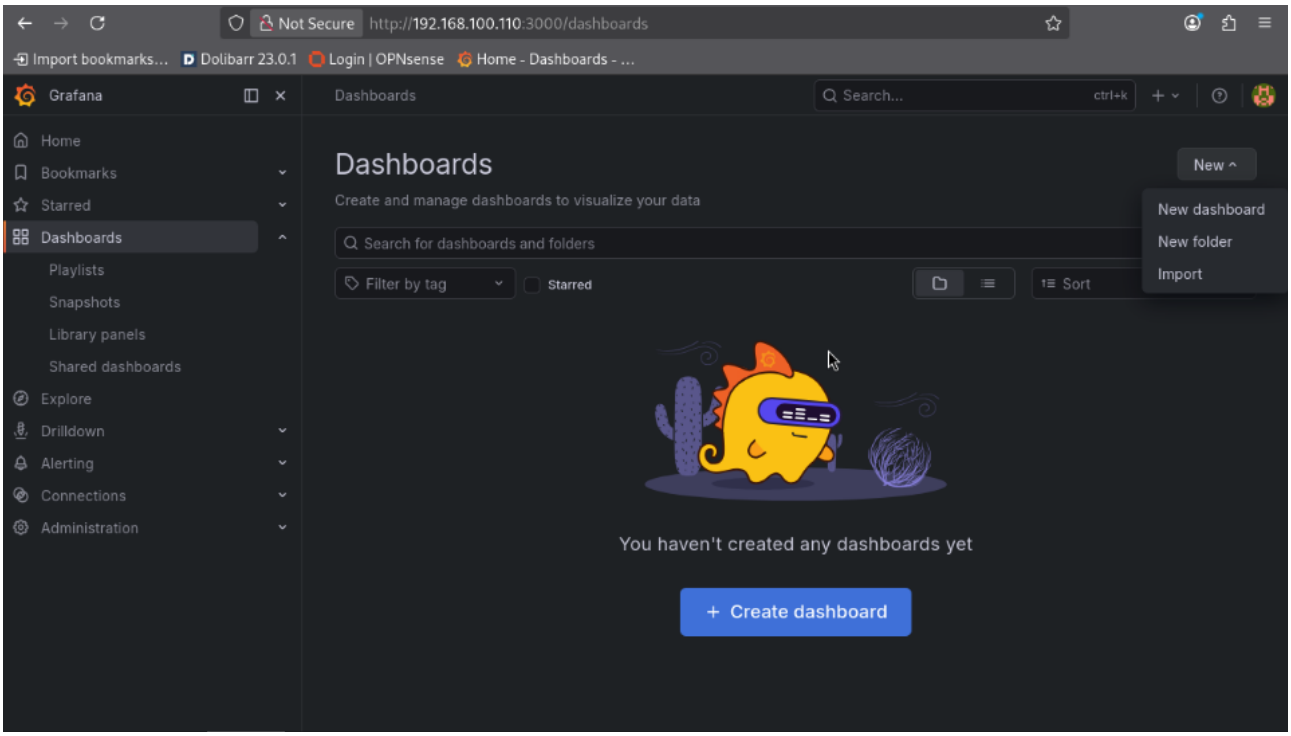
```
sudo nano prometheus.yaml
```

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'linux'
    static_configs:
      - targets: ['node-exporter:9100',
```

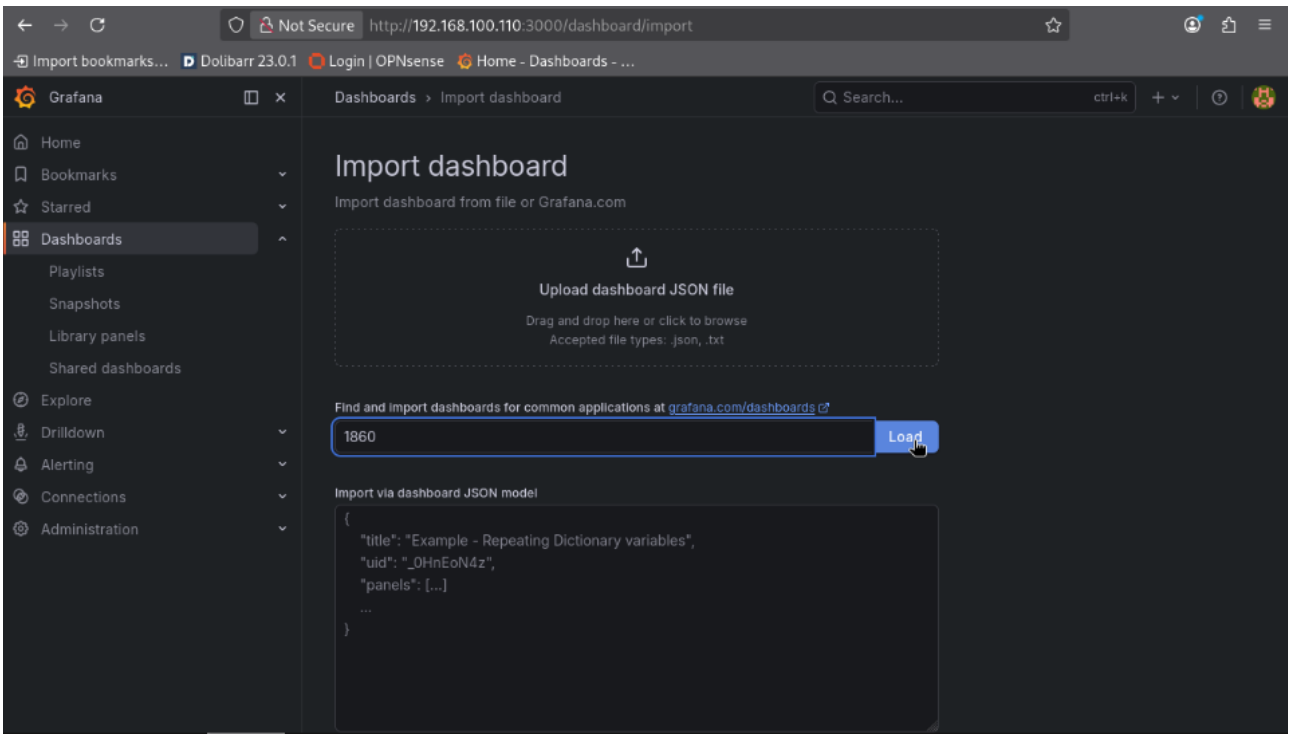
```
'192.168.100.146:9100', '192.168.100.1:9100']

  - job_name: 'windows'
    static_configs:
      - targets: ['192.168.100.100:9182',
'192.168.100.101:9182', '192.168.100.103:9182',
‘192.168.100.140:9182’]
```

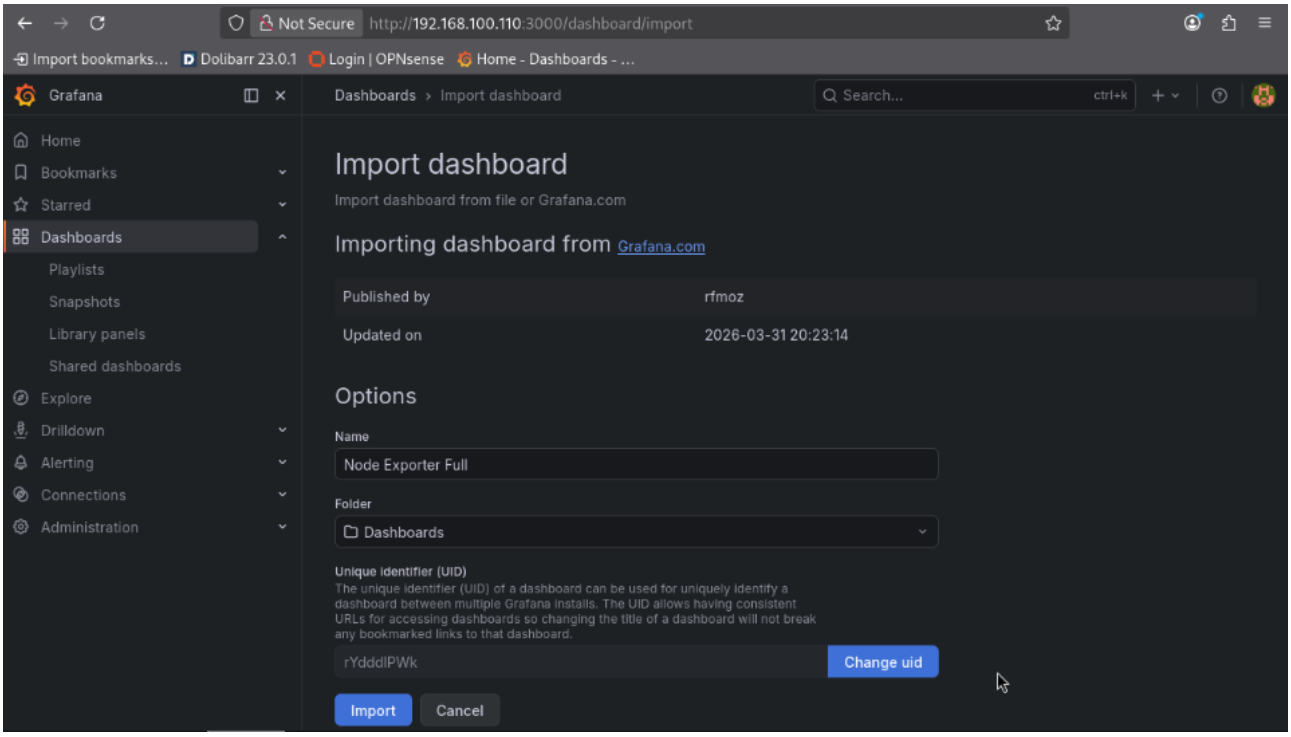
Configuration

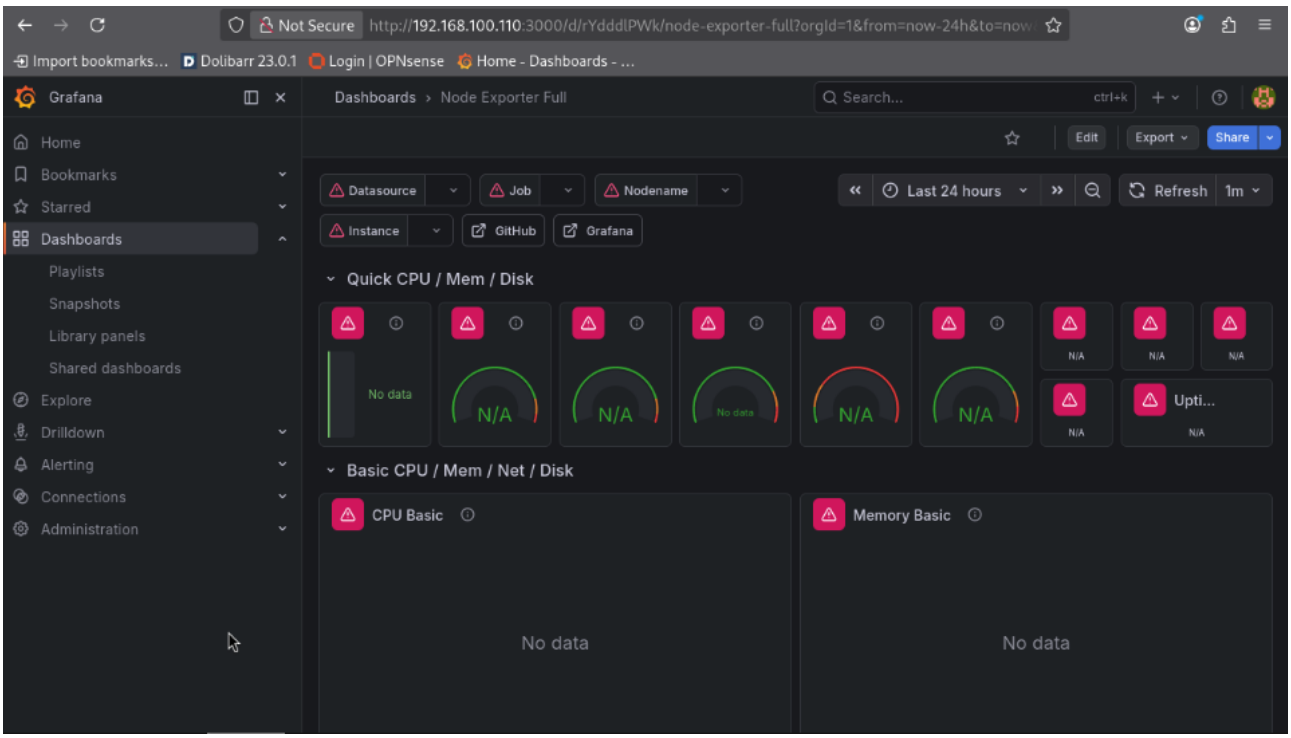
IPvm:3000

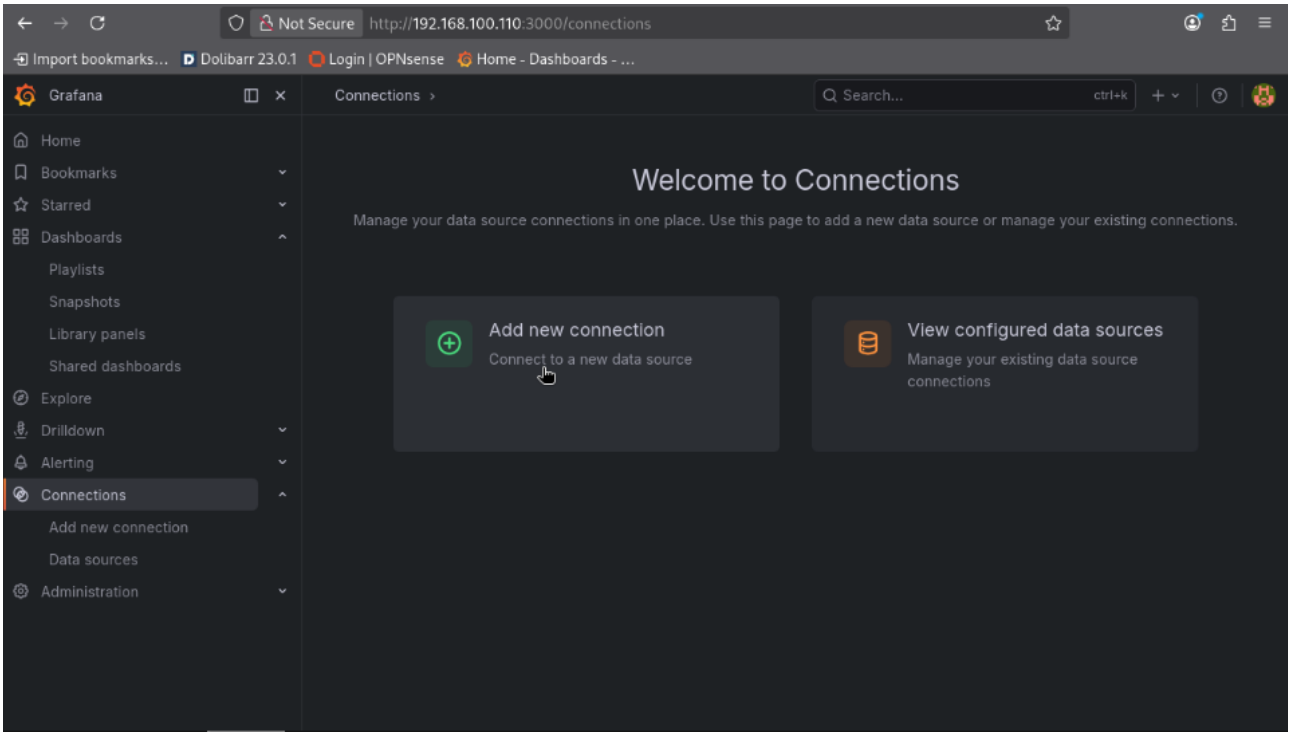


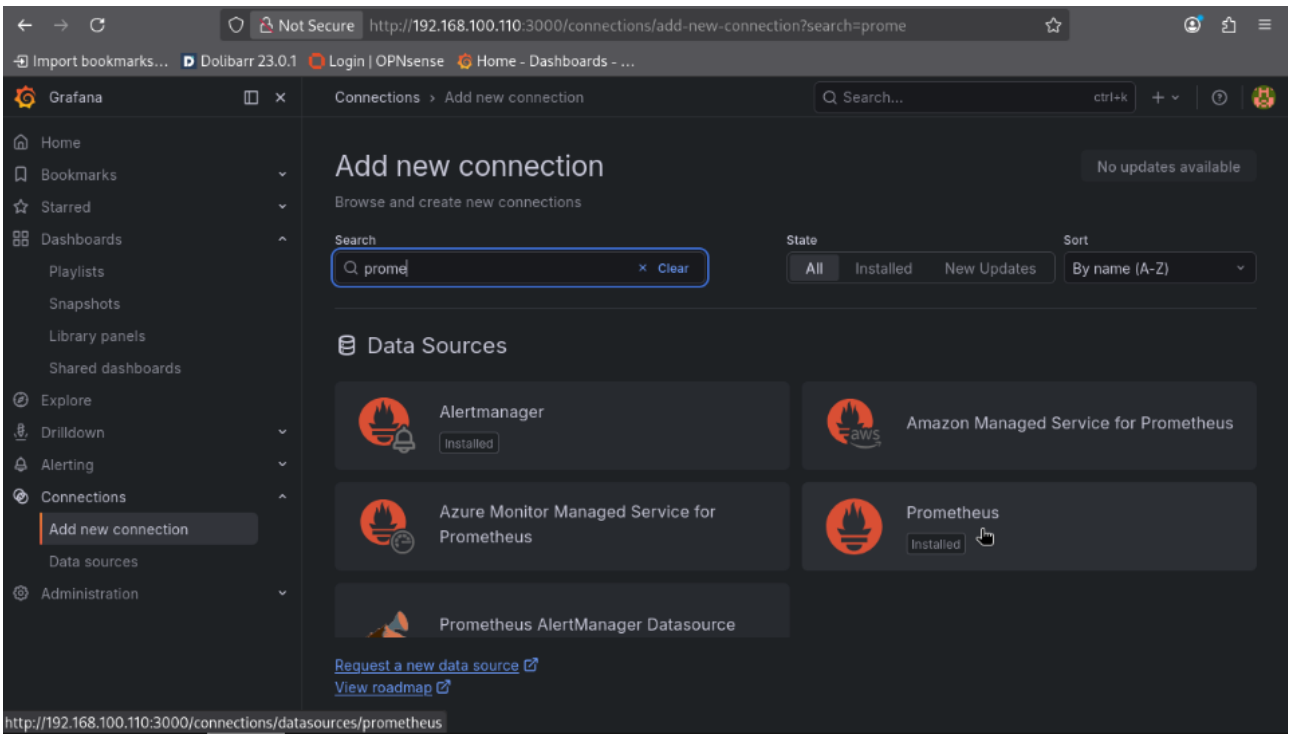
1860

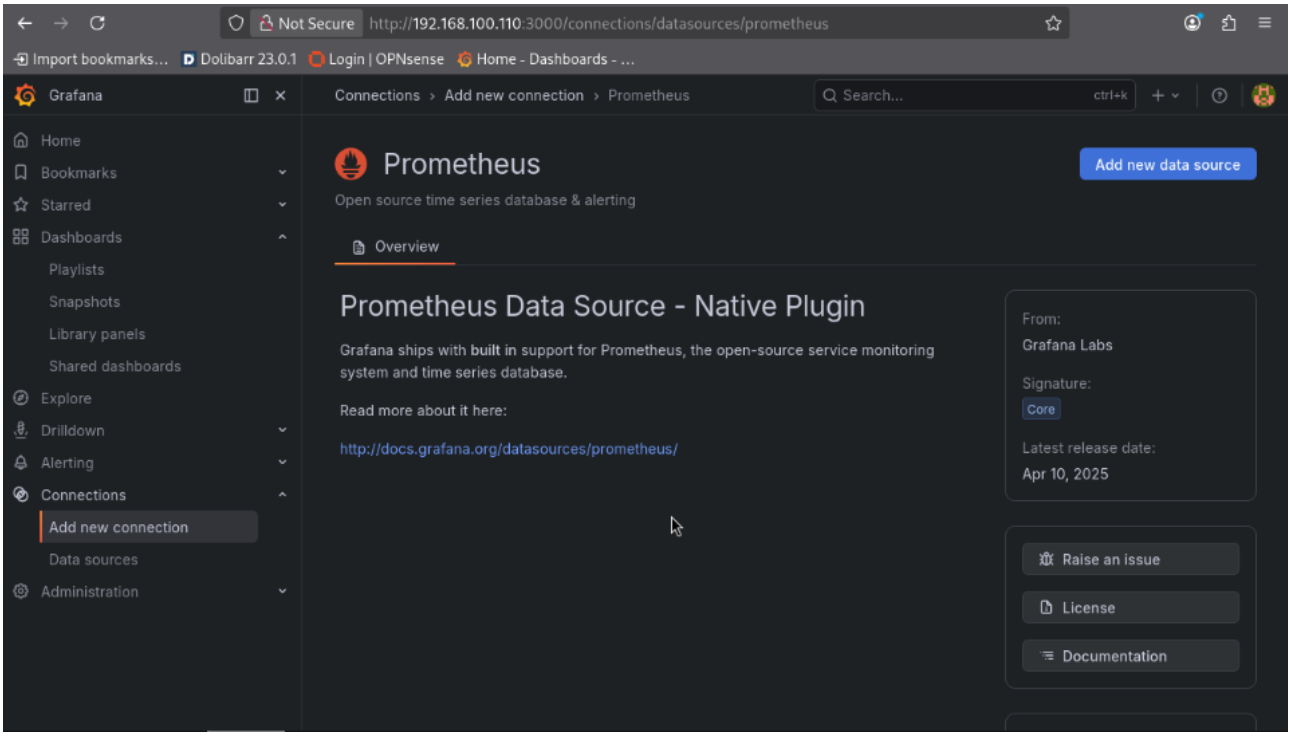

Not Secure
http://192.168.100.110:3000/dashboard/import
Import bookmarks...
Dolibarr 23.0.1
Login | OPNsense
Home - Dashboards - ...
Grafana
Dashboards › Import dashboard
Search...
ctrl+k
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Administration
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Find and import dashboards for common applications at grafana.com/dashboards
1860
Load
Import via dashboard JSON model
{
"title": "Example - Repeating Dictionary variables",
"uid": "_0HnEoN4z",
"panels": [...]
...
}

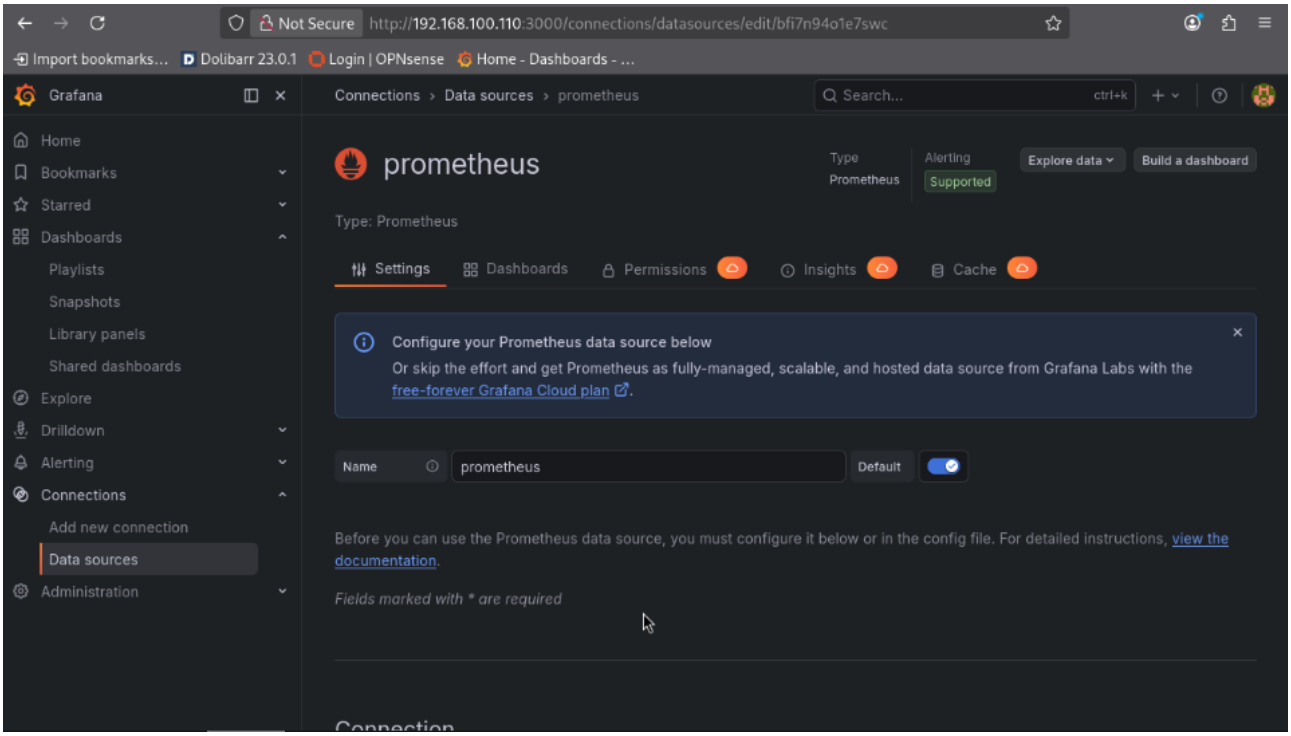

Not Secure
http://192.168.100.110:3000/dashboard/import
Import bookmarks...
Dolibarr 23.0.1
Login | OPNsense
Home - Dashboards - ...
Grafana
Dashboards › Import dashboard
Search...
ctrl+k
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Administration
Import dashboard
Import dashboard from file or Grafana.com
Importing dashboard from Grafana.com
Published by
rfmoz
Updated on
2026-03-31 20:23:14
Options
Name
Node Exporter Full
Folder
Dashboards
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
rYdddlPWk
Change uid
Import
Cancel


Not Secure
http://192.168.100.110:3000/d/rYdddlPWk/node-exporter-full?orgId=1&from=now-24h&to=now
Import bookmarks...
Dolibarr 23.0.1
Login | OPNsense
Home - Dashboards - ...
Grafana
Dashboards › Node Exporter Full
Search...
ctrl+k
Edit
Export
Share
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Administration
Datasource
Job
Nodename
Instance
GitHub
Grafana
Last 24 hours
Refresh
1m
Quick CPU / Mem / Disk
No data
N/A
Upti...
Basic CPU / Mem / Net / Disk
CPU Basic
Memory Basic
No data
No data

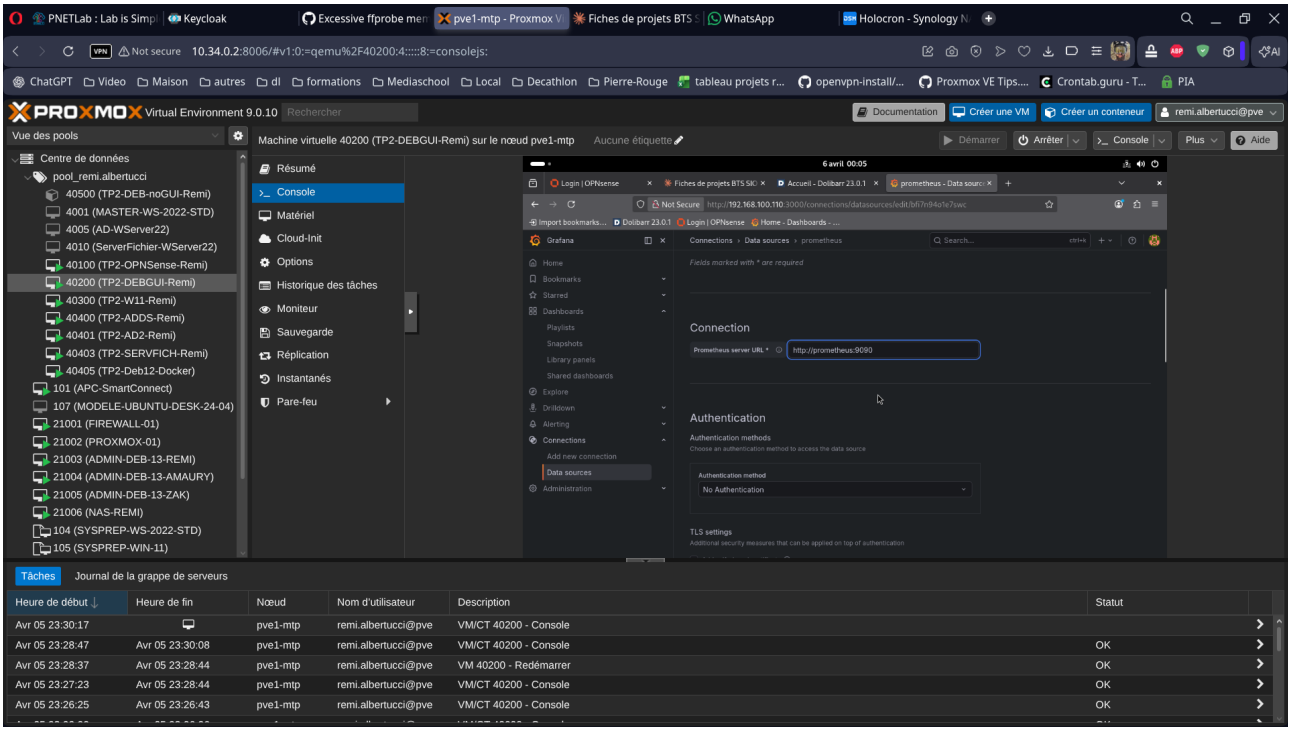

Not Secure
http://192.168.100.110:3000/connections
Import bookmarks...
Dolibarr 23.0.1
Login | OPNsense
Home - Dashboards - ...
Grafana
Connections ›
Search...
ctrl+k
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Add new connection
Data sources
Administration
Welcome to Connections
Manage your data source connections in one place. Use this page to add a new data source or manage your existing connections.
Add new connection
Connect to a new data source
View configured data sources
Manage your existing data source connections

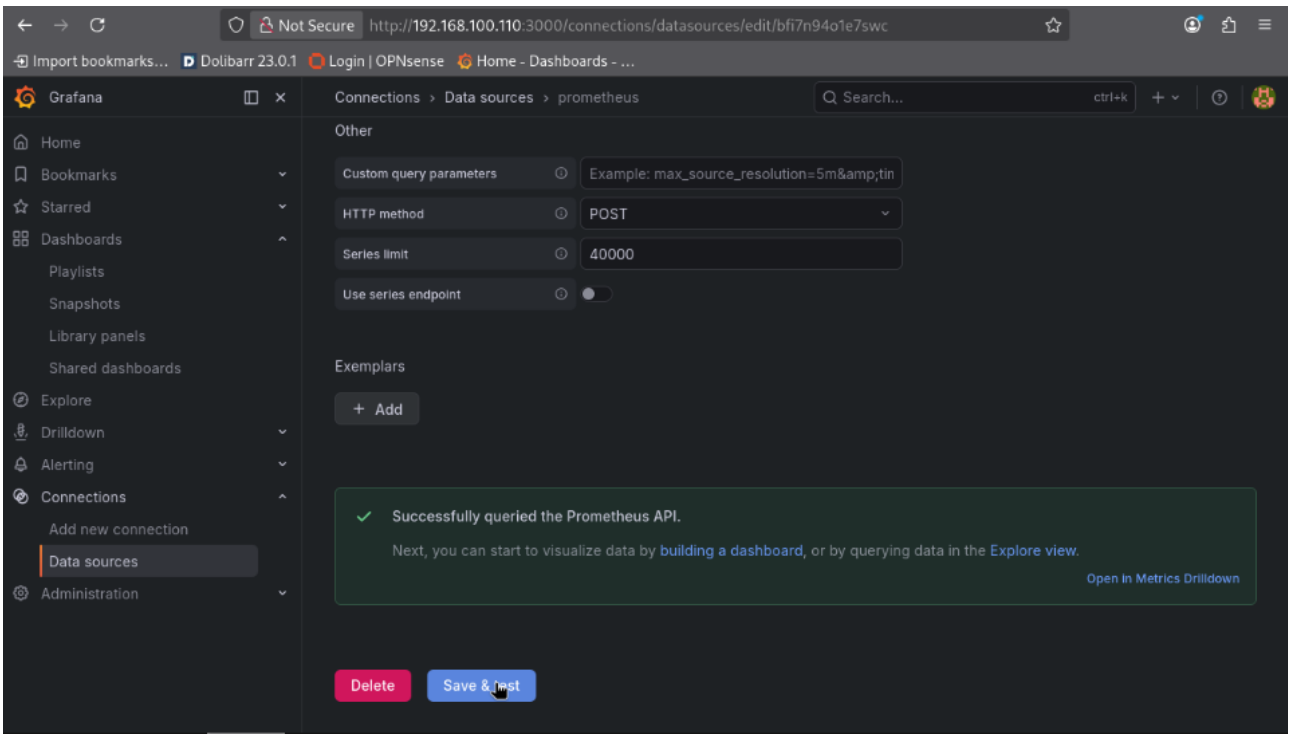

Not Secure http://192.168.100.110:3000/connections/add-new-connection?search=prome
Import bookmarks... Dolibarr 23.0.1 Login | OPNsense Home - Dashboards - ...
Grafana
Connections › Add new connection
Search... ctrl+k
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Add new connection
Data sources
Administration
Add new connection
Browse and create new connections
No updates available
Search
prome
Clear
State
All
Installed
New Updates
Sort
By name (A-Z)
Data Sources
Alertmanager
Installed
Amazon Managed Service for Prometheus
Azure Monitor Managed Service for Prometheus
Prometheus
Installed
Prometheus AlertManager Datasource
Request a new data source
View roadmap
http://192.168.100.110:3000/connections/datasources/prometheus


Not Secure http://192.168.100.110:3000/connections/datasources/prometheus
Import bookmarks... Dolibarr 23.0.1 Login | OPNsense Home - Dashboards - ...
Grafana
Connections › Add new connection › Prometheus
Search... ctrl+k
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Add new connection
Data sources
Administration
Prometheus
Open source time series database & alerting
Add new data source
Overview
Prometheus Data Source - Native Plugin
Grafana ships with built in support for Prometheus, the open-source service monitoring system and time series database.
Read more about it here:
http://docs.grafana.org/datasources/prometheus/
From:
Grafana Labs
Signature:
Core
Latest release date:
Apr 10, 2025
Raise an issue
License
Documentation

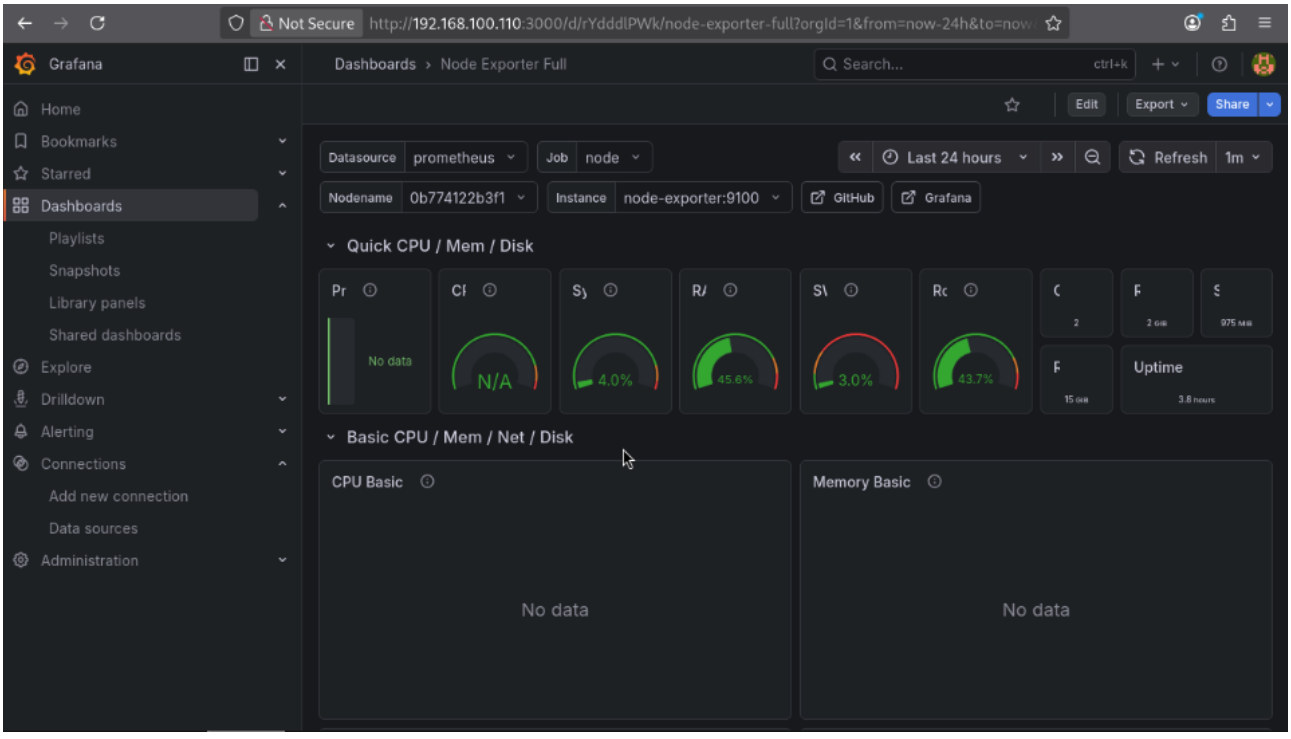


http://prometheus:9090





http://192.168.100.110:9090/targets

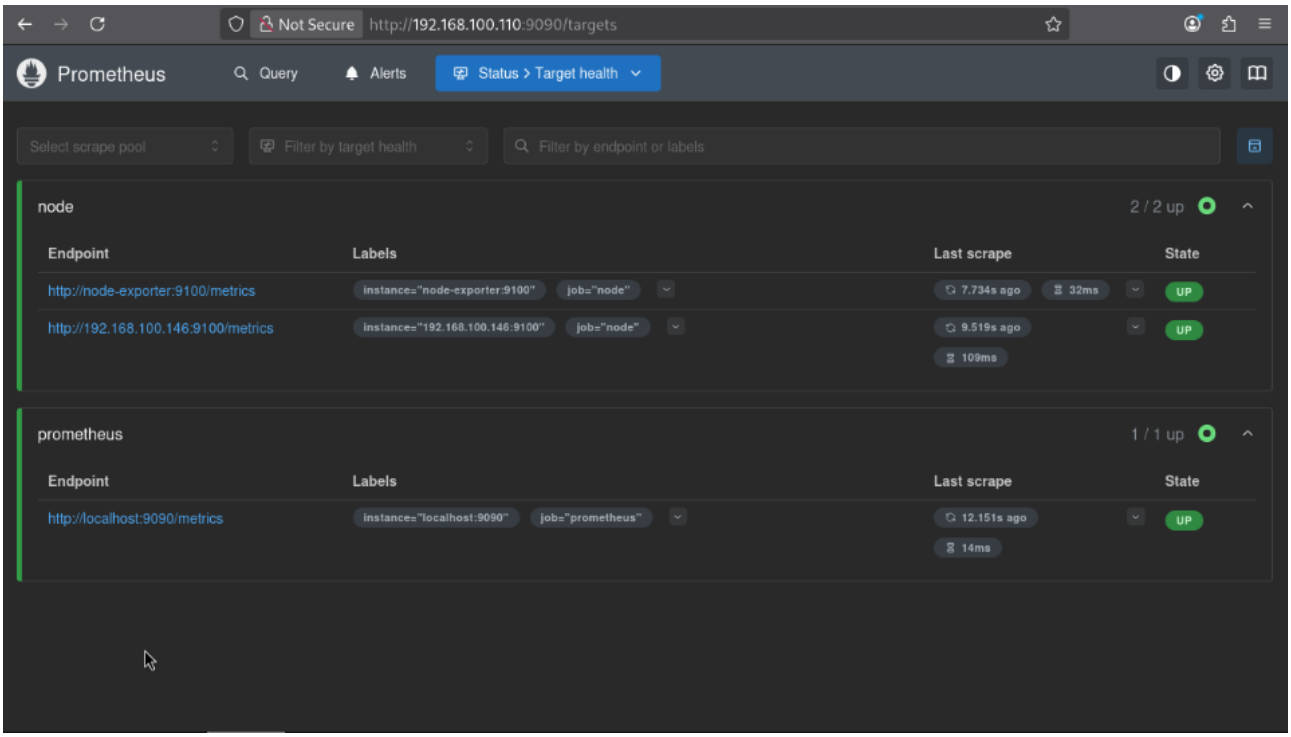


Installer sur linux

Sur deb gui
apt install prometheus-node-exporter -y

```
global:
```

```
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'node'
    static_configs:
      - targets: ['node-exporter:9100',
'192.168.100.146:9100']
```



Installer l'exporter sur Windows

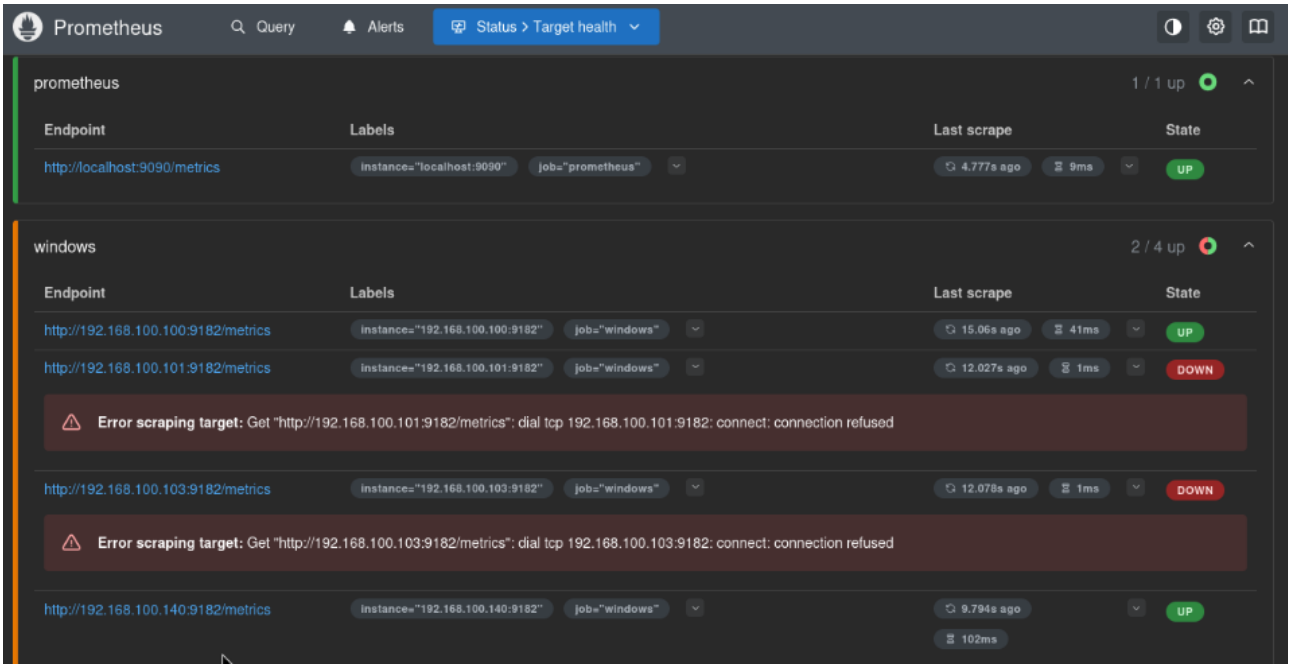
https://github.com/prometheus-community/windows_exporter/releases
ici on travaille avec v0.31.6
dl le msi et on l'installe

```
Invoke-WebRequest -Uri
"https://github.com/prometheus-community/wi
ndows_exporter/releases/download/v0.31.6/
windows_exporter-0.31.6-amd64.msi" -OutFile
"C:\Temp\windows_exporter.msi"

msiexec /i C:\Temp\windows_exporter.msi
```

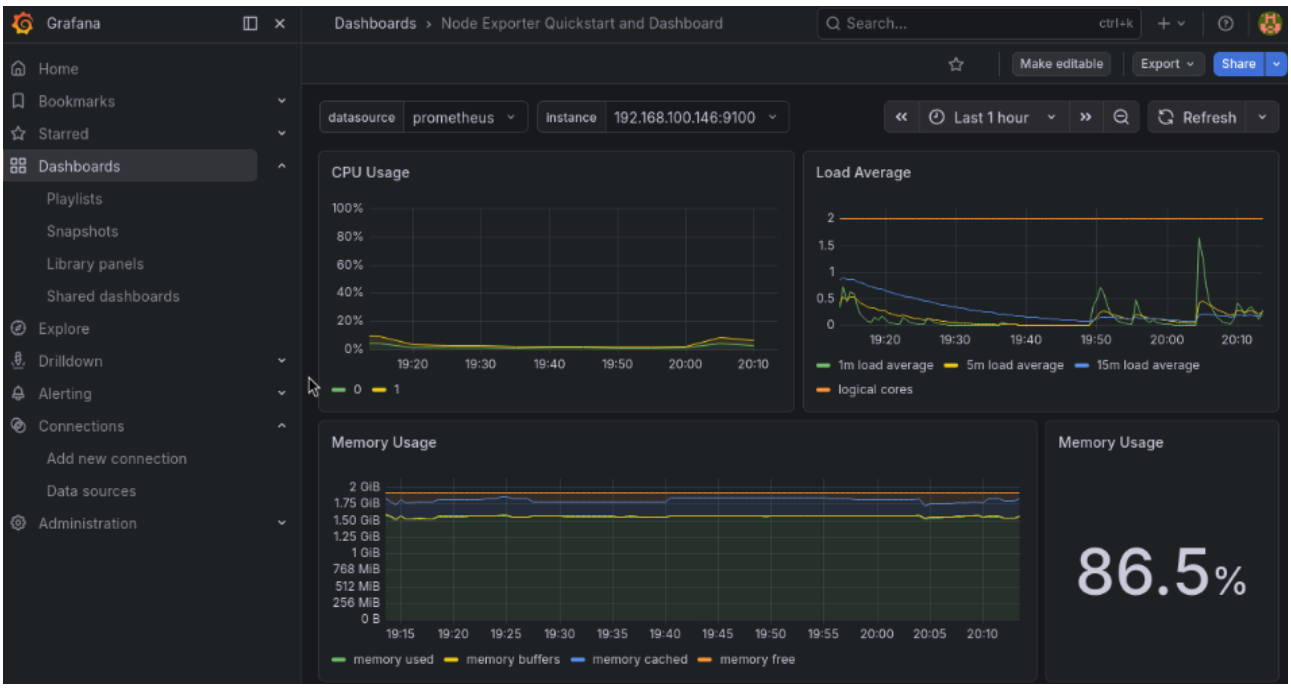
```
/quiet
ENABLED_COLLECTORS=cpu,cs,logical_disk,net
,os,system,service,process

New-NetFirewallRule -DisplayName
"windows_exporter" -Direction Inbound -Protocol
TCP -LocalPort 9182 -Action Allow
```

curl http://localhost:9182/metrics

on verifie ensuite dans notre interface :

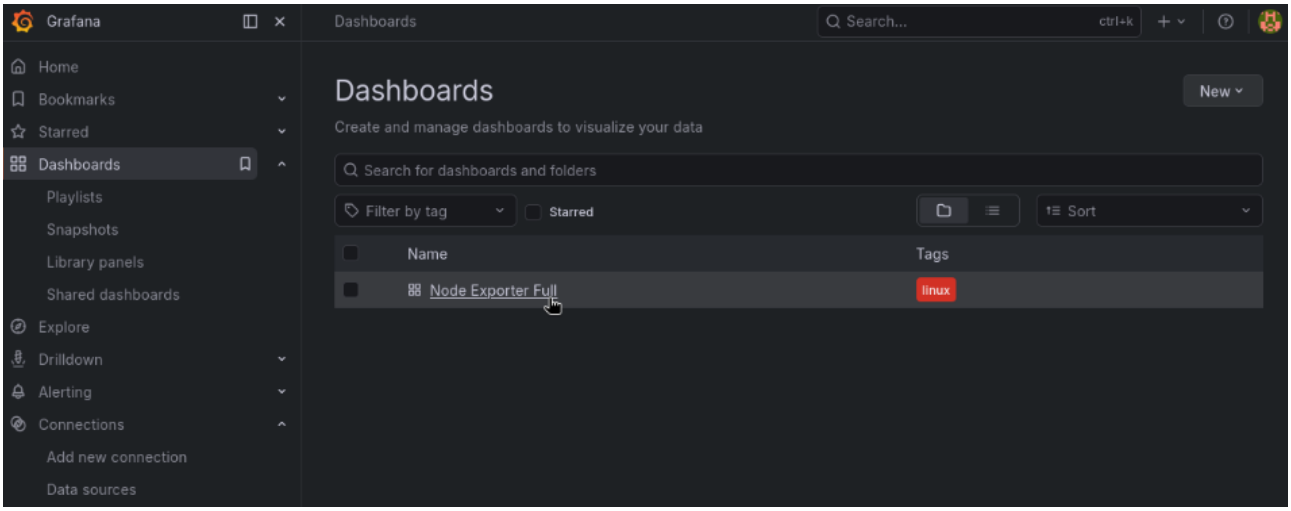


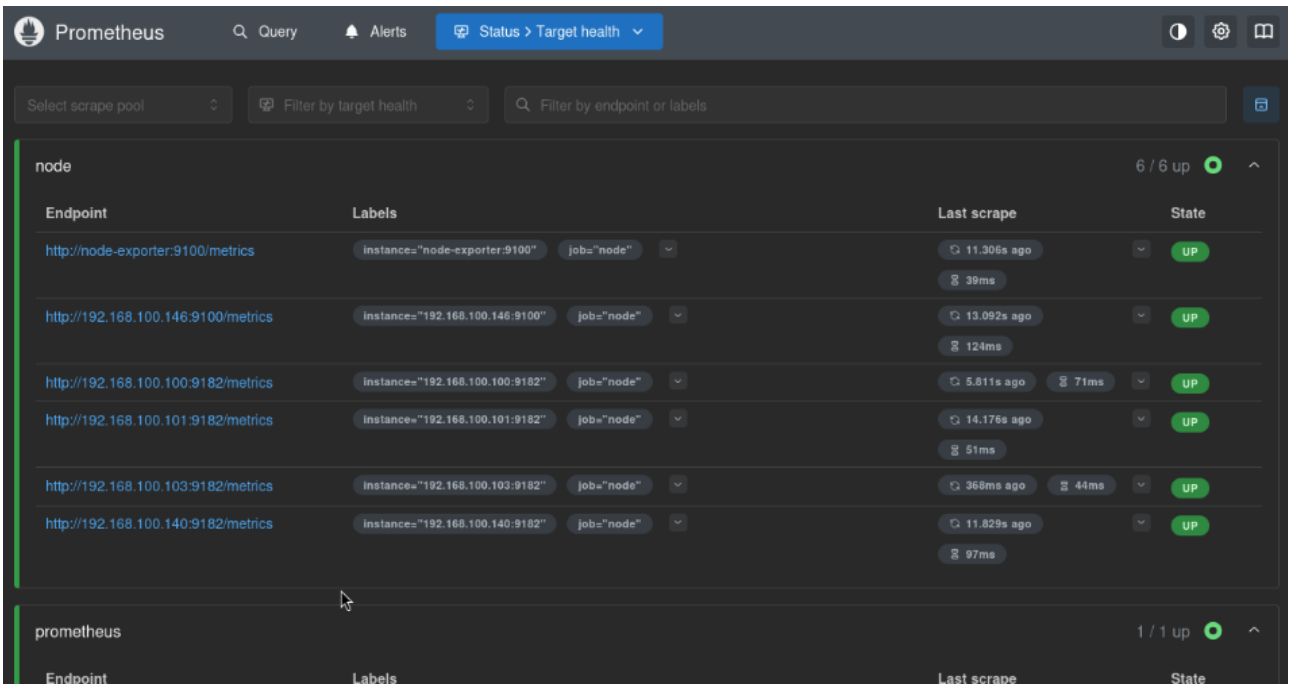
13978



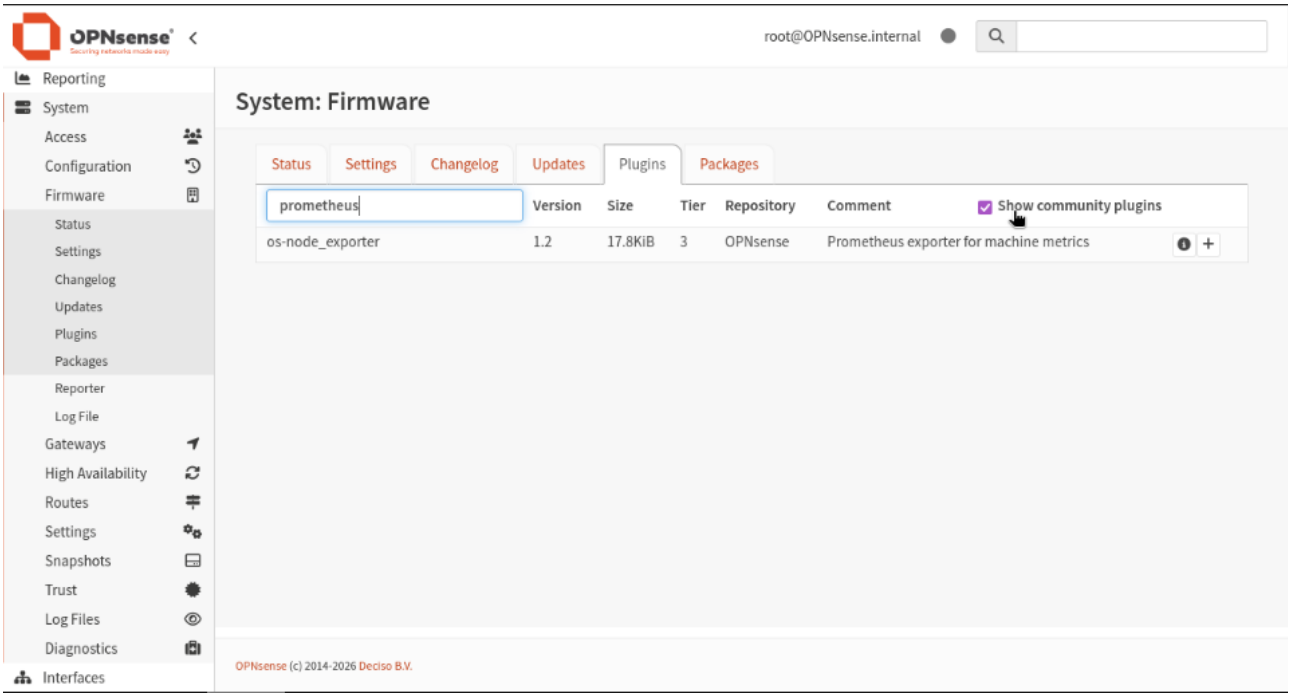
Accéder au dashboard

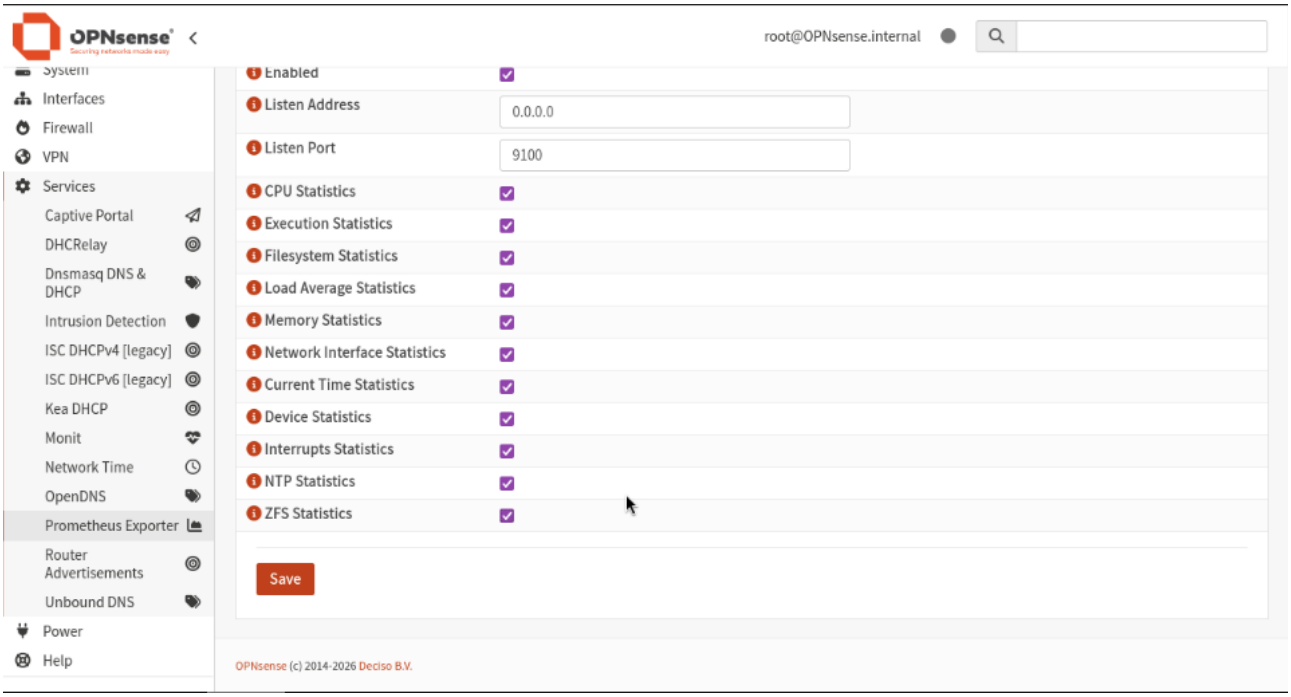
http://192.168.100.110:3000

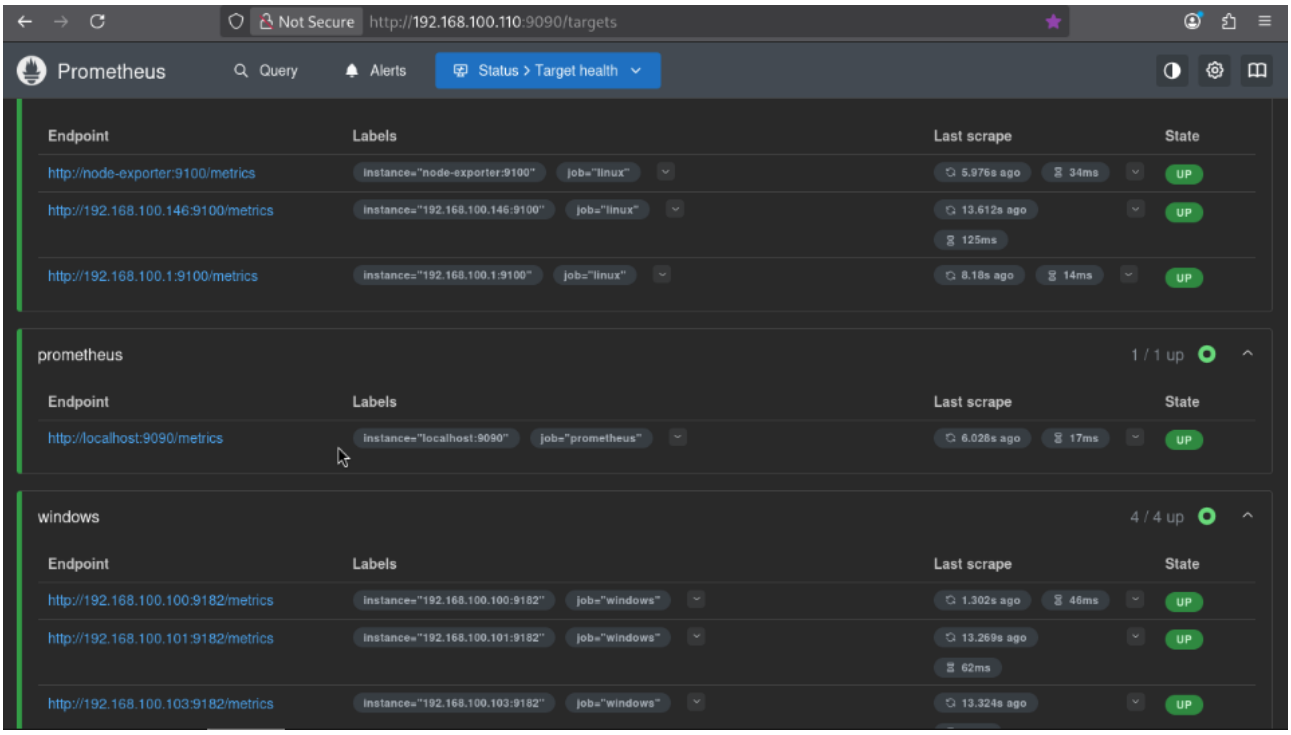




Installer sur Opnsense

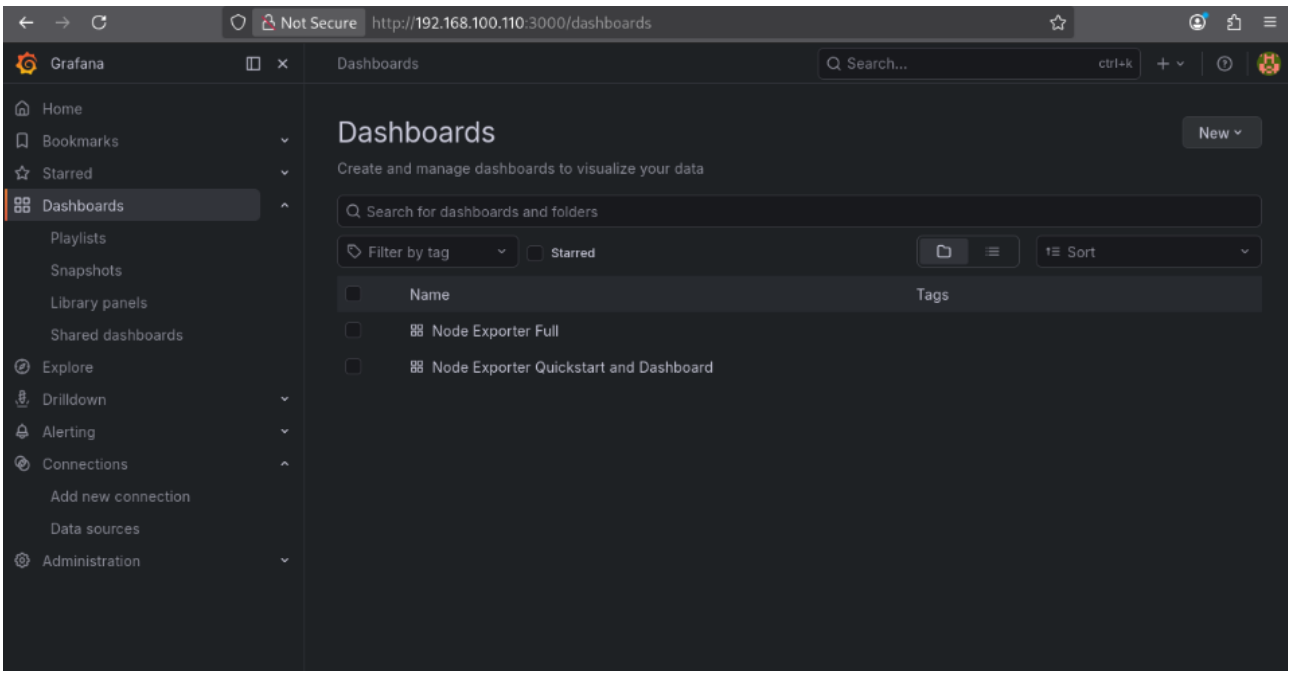




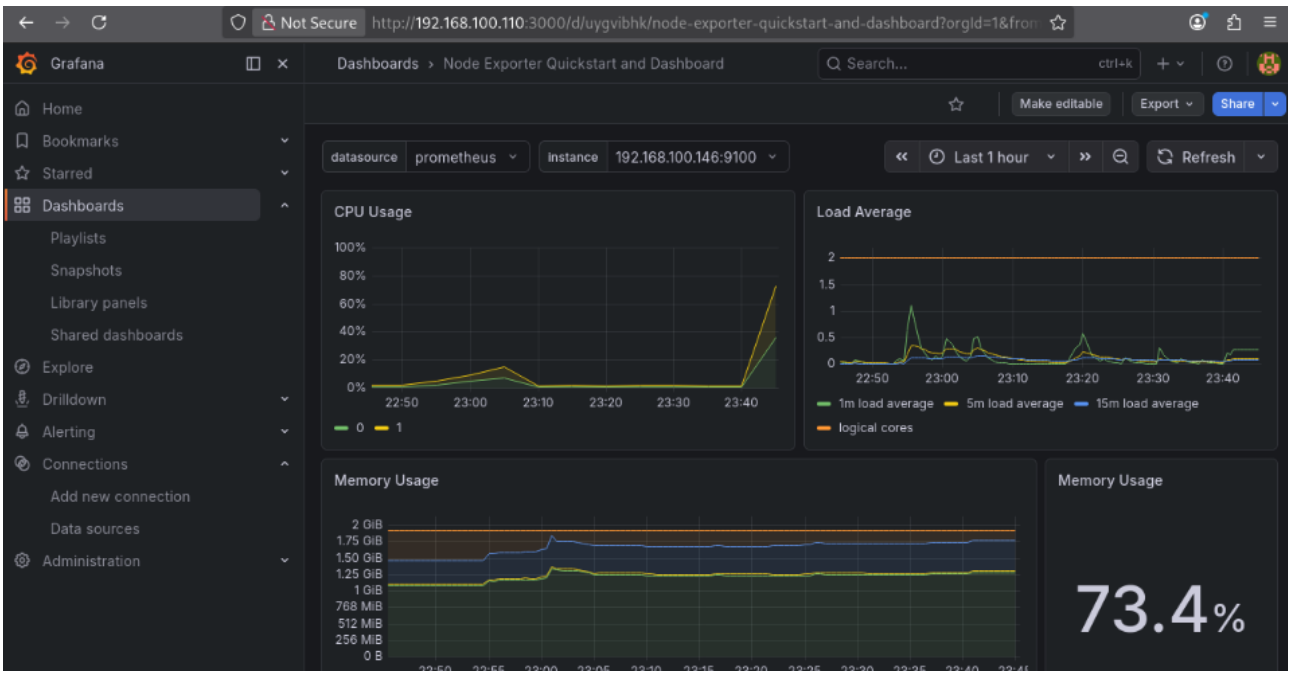


Accéder aux dashboards

192.168.100.110:3000



accéder aux dashboards linux/freebsd et windows séparemment



On sélectionne le job et la machine sur l’onglet du haut.

Il est possible de créer des dashboards personnalisés si besoin.

IX Sauvegardes

Pas d'accès aux sauvegardes de PVE donc pas de PBS car contexte d'école. Donc sauvegarde de la base de données Dolibarr.

Idem car contexte d'école, sauvegarde sur la machine d'administration pour économiser des ressources.

On sauvegardera la base de données de dolibarr

Bases de données => script bash avec rsync, planif

rsync

Sur 192.168.100.146 :

```
mkdir -p /home/<USER>/backup
mkdir -p /home/<USER>/backup/dolibarr
sudo apt update && sudo apt install rsync -y
```

Sur 192.168.100.110 :

```
ssh-keygen -t ed25519    # génère la clé, appuie juste sur Entrée à chaque question

ssh-copy-id <USER>@192.168.100.146    # copie la clé sur la machine admin

ssh <USER>@192.168.100.146    # test que ça connecte sans mot de passe

exit
```

```
touch /var/log/backup_dolibarr.log
chmod 644 /var/log/backup_dolibarr.log
```

```
mkdir ~/scripts && sudo nano dolibarr.sh
```

```
#!/bin/bash

BACKUP_DIR="/home/<USER>/backup/dolibarr"
DEST="<USER>@192.168.100.146:/home/<USER>/backup/dolibarr"
DATE=$(date +%Y%m%d_%H%M%S)
KEEP=7
```

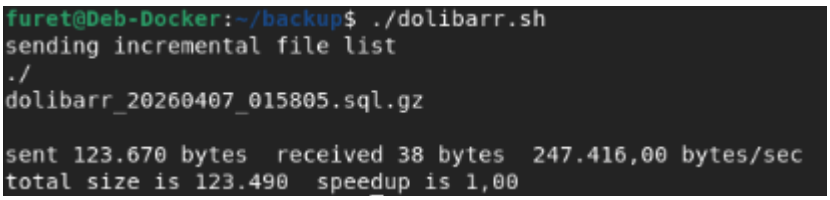
```
mkdir -p "$BACKUP_DIR"

sudo docker exec dolibarr mysqldump -h
dolibarr_db -u dolibarr -p dolibarr dolibarr \
| gzip > "$BACKUP_DIR/dolibarr_$DATE.sql.gz"

/usr/bin/rsync -avz "$BACKUP_DIR/" "$DEST"

ls -t "$BACKUP_DIR"/*.sql.gz | tail -n +$
((KEEP+1)) | xargs -r rm
```

```
chmod +x ~/scripts/dolibarr.sh
```

Test

Dans .110 :

./dolibar.sh



le script est fonctionnel, on peut donc lancer la planification

Planification

Toujours sur la VM hôte de Dolibarr :

```
crontab -e
```

```
0 2 * * * /opt/scripts/backup_dolibarr.sh >>
/var/log/backup_dolibarr.log 2>&1
```

crontabguru

X Conclusion