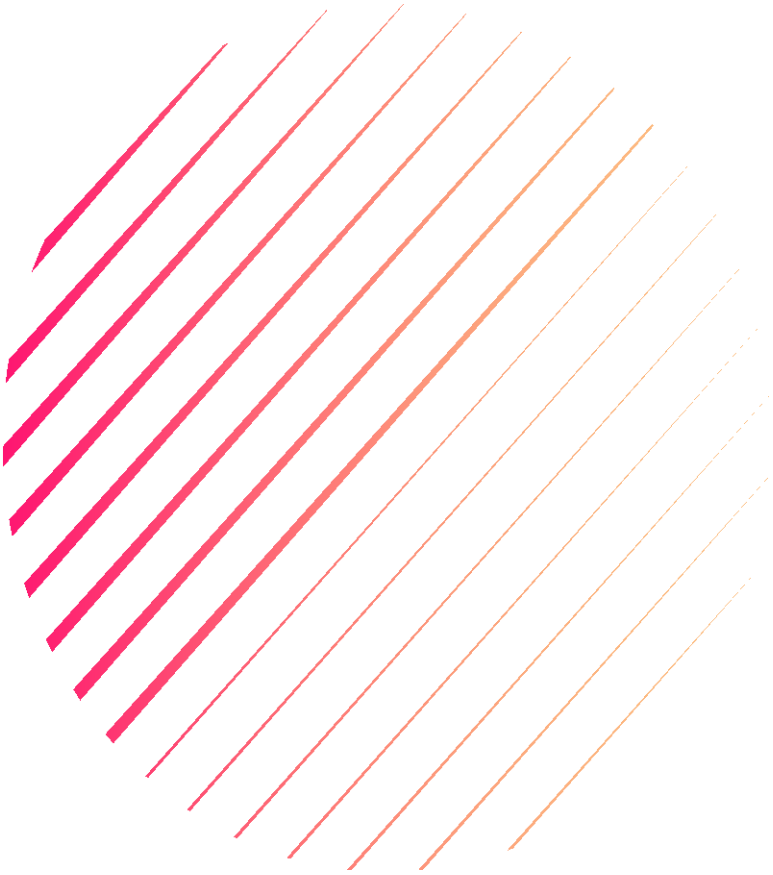
Atelier Pro 4



17/04/2026
Projet : Déploiement d’une solution d’EDR

Ethan BAMBARA--DASYLVA
BTS SIO 2024 – 2026

Table des matières

Contexte du Projet

Description du laboratoire GSB

Le secteur d'activité :
L'industrie pharmaceutique est un secteur très lucratif dans lequel le mouvement de fusion acquisition est très fort. Les regroupements de laboratoires ces dernières années ont donné naissance à des entités gigantesques au sein desquelles le travail est longtemps resté organisé selon les anciennes structures.
Des déboires divers récents autour de médicaments ou molécules ayant entraîné des complications médicales ont fait s'élever des voix contre une partie de l'activité des laboratoires : la visite médicale, réputée être le lieu d' arrangements entre l'industrie et les praticiens, et tout du moins un terrain d'influence opaque.

L'entreprise :
Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy (spécialisé dans le secteur des maladies virales dont le SIDA et les hépatites) et le conglomérat européen Swiss Bourdin (travaillant sur des médicaments plus conventionnels), lui même déjà union de trois petits laboratoires .
En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris.
Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis.
La France a été choisie comme témoin pour l'amélioration du suivi de l'activité de visite.

Description du système informatique

Le système informatique :
Sur le site parisien, toutes les fonctions administratives (gestion des ressources humaines, comptabilité, direction, commerciale, etc.) sont présentes. On trouve en outre un service labo-recherche, le service juridique et le service communication.

La salle serveur occupe le 6ème étage du bâtiment et les accès y sont restreints (étage accessible par ascenseur à l'aide d'une clé sécurisée, portes d'accès par escalier munies d'un lecteur de badge, sas d'entrée avec gardien présent 24h/24).

Les serveurs assurent les fonctions de base du réseau (DHCP, DNS, Annuaire et gestion centralisée des environnements) et les fonctions de communication (Intranet, Messagerie, Agenda partagé, etc.).
On trouve aussi de nombreuses applications métier (base d'information pharmaceutique, serveurs dédiés à la recherche, base de données des produits du laboratoire, base de données des licences d'exploitation pharmaceutique, etc.) et les fonctions plus génériques de toute entreprise (Progiciel de Gestion Intégré avec ses modules RH, GRC, etc.).
Un nombre croissant de serveurs est virtualisé.

Constitué autour de VLAN, le réseau segmente les services de manière à fluidifier le trafic.

Les données de l'entreprises sont considérées comme stratégiques et ne peuvent tolérer ni fuite, ni destruction. L'ensemble des informations est répliqué quotidiennement aux Etats-Unis par un lien dédié. Toutes les fonctions de redondances (RAID, alimentation, lien réseau redondant, Spanning-tree, clustering, etc.) sont mises en œuvre pour assurer une tolérance aux pannes maximale.

L'équipement :
L'informatique est fortement répandue sur le site. Chaque employé est équipé d'un poste fixe relié au système central. On dénombre ainsi plus de 350 équipements terminaux et un nombre de serveurs physiques conséquent (45 en 2012) sur lesquels tournent plus de 100 serveurs virtuels.

On trouve aussi des stations de travail plus puissantes dans la partie labo-recherche, et une multitude d'ordinateurs portables (personnels de direction, service informatique, services commerciaux, etc).

Les visiteurs médicaux reçoivent une indemnité bisannuelle pour s'équiper en informatique (politique Swiss-Bourdin) ou une dotation en équipement (politique Galaxy). Il n'y a pas à l'heure actuelle d'uniformisation des machines ni du mode de fonctionnement

Chaque employé de l'entreprise a une adresse de messagerie de la forme **nomUtilisateur@steph.com**. Les anciennes adresses de chaque laboratoire ont été définitivement fermées au 1er janvier 2011.

Organisation du réseau

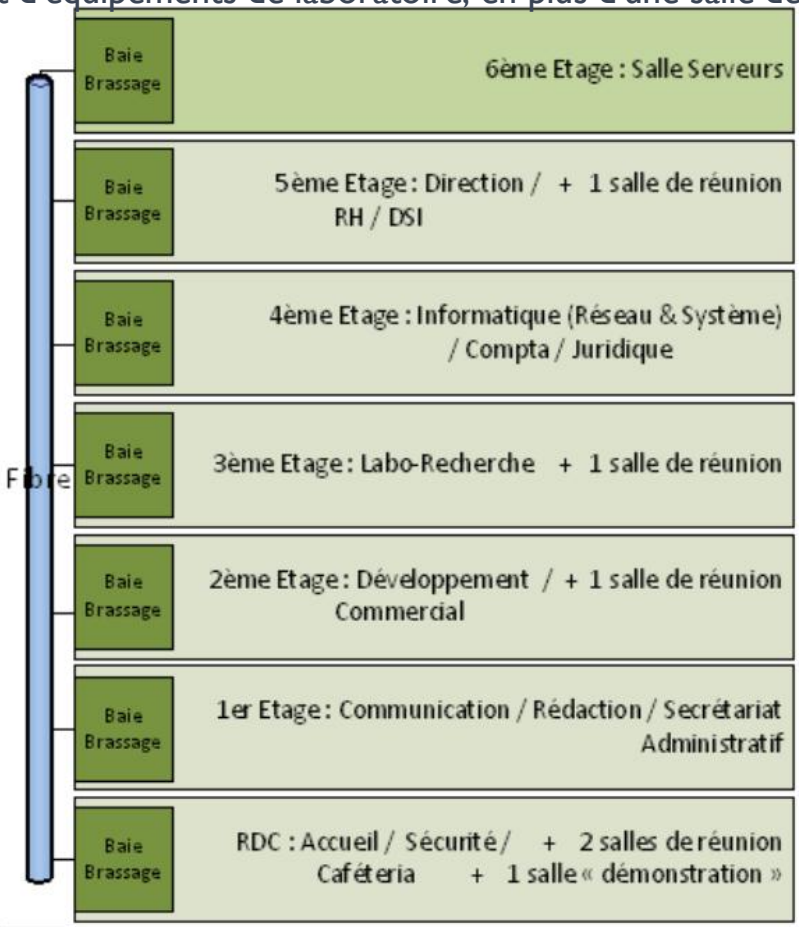
Répartition des services :
Chaque étage dispose d'une baie de brassage qui le relie par une fibre à la baie centrale de la salle serveurs.

Toutes les salles de réunion sont équipées d'un point d'accès Wifi positionné par défaut dans le VLAN "Visiteurs" qui autorise uniquement un accès Internet.
Les portables connectés en wifi à ce point d'accès reçoivent ainsi une adresse IP et n'ont, par conséquent accès qu'aux services DHCP et DNS.

Le point d'accès peut être configuré à la demande pour être raccordé à un VLAN présent au niveau de l'étage.

Chaque salle de réunion dispose d'un vidéoprojecteur, d'enceintes et d'un tableau numérique interactif.

La salle "Démonstration" est destinée à l'accueil des organismes de santé (AFSSAPS notamment) et des partenaires scientifiques. Elle dispose de paillasses et d'équipements de laboratoire, en plus d'une salle de réunion.



Projet

Segmentation du réseau :

L'organisation des VLAN et de l'adressage IP est la suivante :

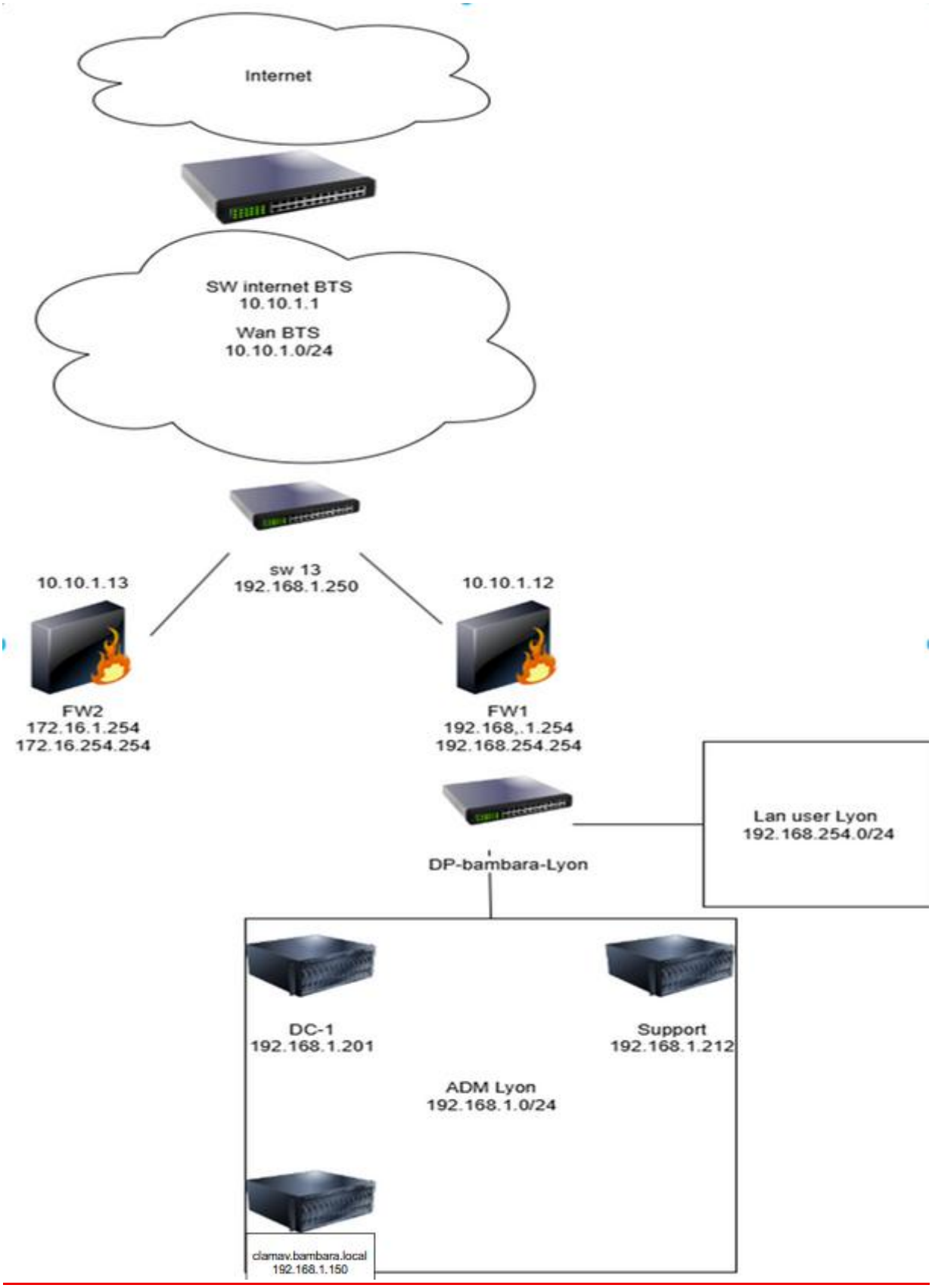
N° VLAN	Service(s)	Adressage IP
20	ADM-LYON	192.168.1.0/24
21	LAN-LYON	192.168.254.0/24
22	ADM-SLM	172.16.1.0/24
23	LAN-SLM	172.16.254.0/24
555	WAN	10.10.1.0/24

Salle serveur et connexion internet

L'organisation des serveurs et des équipements réseaux est la suivante :

- Le serveur principal est virtualisé sous le système Vmware Vcenter 7.0
- Un Commutateur Multicouche Cisco permet l’interconnection du serveur principal et la liaison vers le firewall de proximité (Internet).
- L’environnement Virtuel et réseau des Projets d’Atelier de Professionnalisation sont référencées ci-dessous :

Schéma réseau



Rapport Technique

La réalisation du projet se divise en deux parties. D'abord, nous allons faire la configuration des clients ClamAV et du serveur.

Ensuite, nous allons nous occuper de configurer un serveur Wazuh pour récupérer les logs.

Creation DU serveur Clamav

Depuis une machine Debian, ouvrez un terminal bash.

Depuis le terminal, nous allons installer l'agent ClamAV et un serveur web. Dans notre cas, nous installons Nginx.

Pour ce faire, exécutez les commandes suivantes :

```
sudo apt update
sudo apt install nginx -y
sudo apt install clamav clamav-daemon
```

Attention : nous avons eu des problèmes avec la dernière version de ClamAV, nous avons donc installé la version 1.4.4 de celui-ci.

Une fois ClamAV installé sur le serveur, nous pouvons configurer le serveur web. Ce dernier va nous permettre de rendre nos bases de données locales que nos antivirus vont utiliser. Pour pouvoir mettre à jour les bases de données des clients, il faudra mettre à jour celles du serveur.

Dans un premier temps, il faut créer le fichier de configuration du site.

Exécutez la commande suivante pour créer le fichier et adaptez les champs selon vos besoins :

```
vi /etc/nginx/sites-available/clamav

## vous pouvez aussi éditer le fichier de configuration « default »
```

Ensuite, mettez cette configuration :

nginx

```
server {
   listen 80;
   listen [::]:80;

   server_name clamav.bambara.local; ## mettez le nom de domaine qui vous convient

   root /var/www/clamav;

   location / {
      autoindex on; ## bien préciser cette variable
      try_files $uri $uri/ =404;
   }
}
```

Exécutez la commande suivante pour créer un lien symbolique :

```
ln -s /etc/nginx/sites-available/clamav /etc/nginx/sites-enabled/clamav
```

Ensuite, redémarrez Nginx :

```
systemctl restart nginx
```

Votre site est normalement accessible depuis l'IP du serveur. Si vous vous y connectez, vous arriverez sur une page vierge.

Pour ajouter du contenu à cette page, notamment les bases de données des agents, nous devrons ajouter les fichiers dans le dossier défini plus tôt (**/var/www/clamav**).

Si le dossier n'est pas créé, créez-le :

```
mkdir -p /var/www/clamav
```

Ensuite, il faut créer les fichiers de BDD dans ce dossier. Il faut installer python3-requests et pipx. Exécutez les commandes suivantes :

```
apt install python3-requests
apt install pipx
sudo pipx ensurepath
```

Maintenant, pour créer et mettre à jour la base de données, exécutez :

```
pip3 install cvdupdate
cvd config set --dbdir /var/www/clamav/
cvd update
```

Vous pouvez vérifier le contenu du dossier pour voir si tout s'est bien passé.

Ensuite, nous allons créer une tâche cron pour mettre à jour les BDD automatiquement. Exécutez les commandes suivantes :

crontab -e

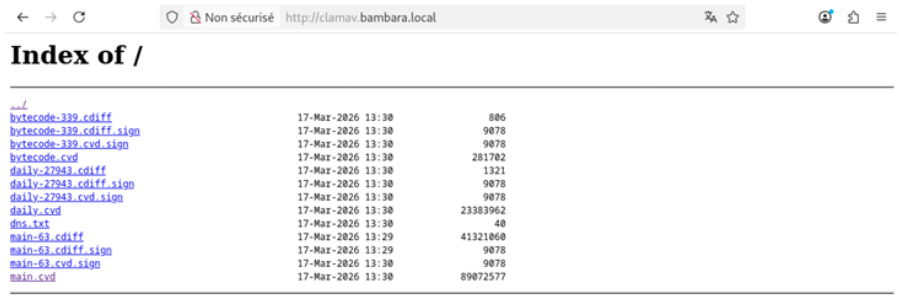
Ajoutez la ligne suivante :

0 * * * * /bin/sh -c "~/.local/bin/cvd update &> /dev/null"

Maintenant, redémarrez Nginx :

systemctl restart nginx

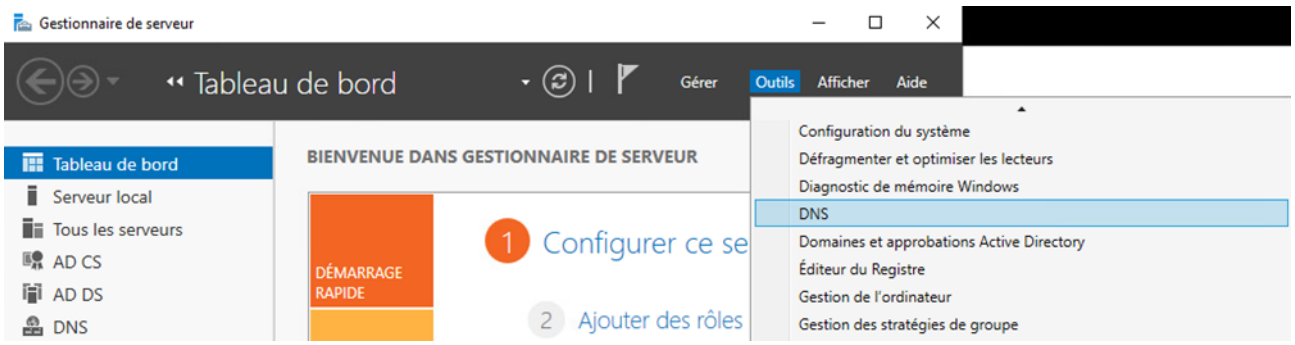
Et si vous vous connectez sur votre site, vous arriverez sur une page ressemblant à ceci.



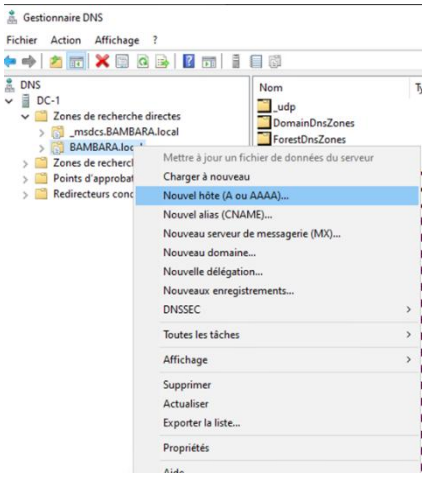
Sur l'image, vous pouvez voir que j'accède au site via un nom de domaine.

Pour configurer ceci, rendez-vous sur votre DNS (dans notre cas, sur notre serveur AD).

Depuis le Gestionnaire de serveur : **Outils > DNS**.

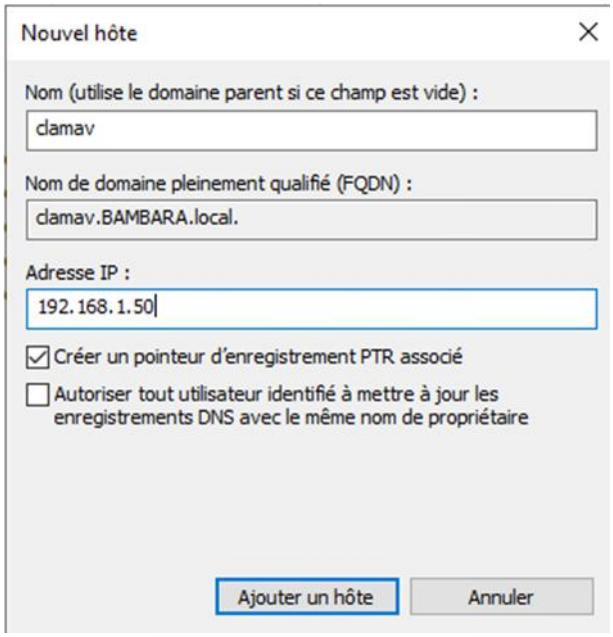


Ensuite, dans la console DNS : Zones de recherche directe > votre nom de domaine > Clic droit > Nouvel hôte (A ou AAAA).



Remplissez les champs comme suit :

Nom : nom du serveur
Adresse IP : IP de votre serveur



Normalement, votre serveur devrait être accessible depuis son nom de domaine.

Configuration des agents clamav

Nous allons configurer l'agent ClamAV sur notre serveur AD avant de le déployer.

Pour ce faire, téléchargez et installez ClamAV depuis le site officiel. Dans notre cas, nous allons télécharger la version **1.4.4**.

Par défaut, ClamAV s'installe dans le dossier suivant :

C:\Program Files\ClamAV

Ici, il faudra récupérer les fichiers suivants :
clamd.conf
freshclam.conf

Ensuite, nous allons commencer par configurer le fichier clamd.conf.

Dans ce fichier, nous allons ajouter ou modifier les variables suivantes :

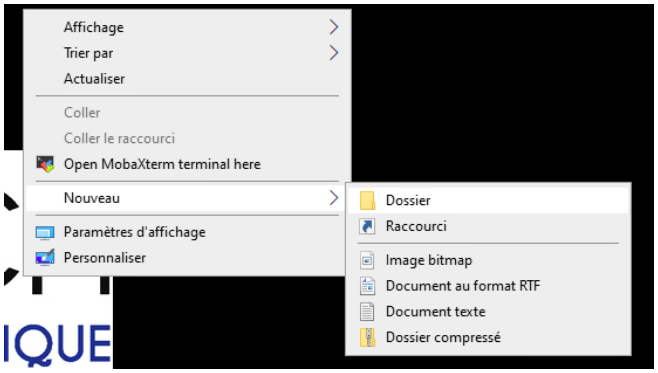
ExcludePath "C:\\Windows" ## Attention : on définit cette variable uniquement pour gagner du temps pour nos tests
TCPAddr 0.0.0.0
TCPSocket 3310
DatabaseDirectory "C:\Program Files\ClamAV\database"
TemporaryDirectory "C:\Program Files\ClamAV\quarantine"
PidFile "C:\Program Files\ClamAV\clamd.pid"
ExtendedDetectionInfo yes
LogSyslog yes
LogRotate yes
LogTime yes
LogFileMaxSize 15M
LogFile "C:\Program Files\ClamAV\logs\clamscan.log"

Ensuite, nous allons configurer le fichier freshclam.conf en modifiant ou ajoutant ces variables :

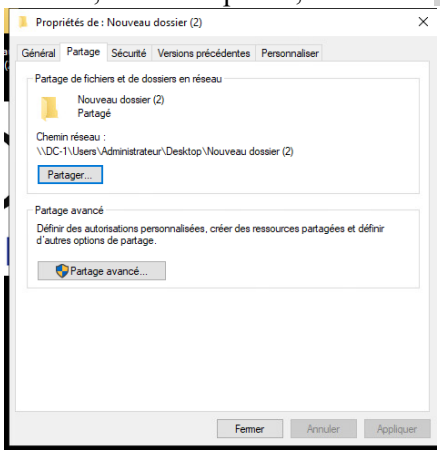
LogSyslog yes
DatabaseMirror http://clamav.bambara.local/ ## renseigner le nom de domaine ou l'IP du serveur
ScriptedUpdates no

Déploiement des Agents Clamav

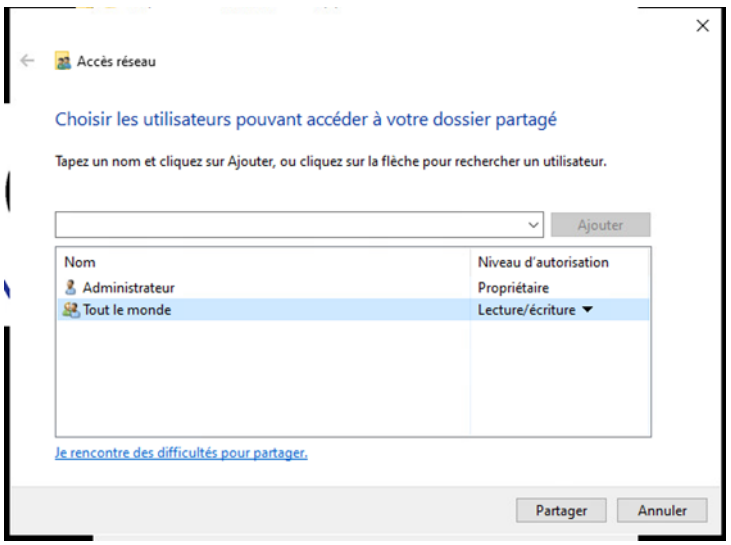
Pour commencer, nous allons créer un dossier partagé.
Sur votre bureau**, créez un dossier.**



Ensuite, dans les options, allez dans **Partage**, puis cliquez sur **Partager**.



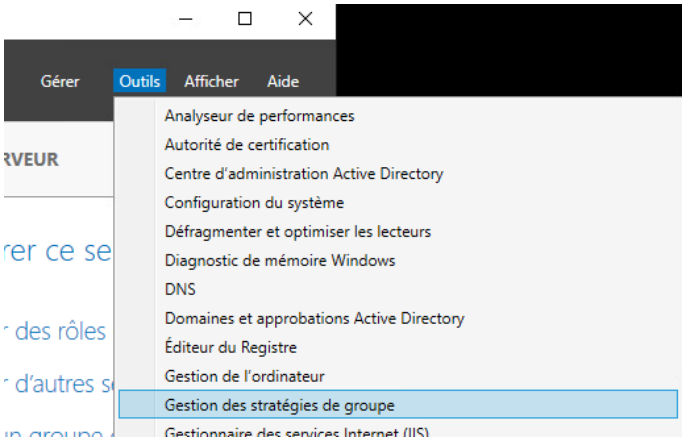
Sélectionnez **Tout le monde** et donnez-leur les droits en **Lecture / Écriture**.



Ensuite, dans ce dossier, copiez l'installateur de ClamAV ainsi que les fichiers **freshclam.conf et clamd.conf.**

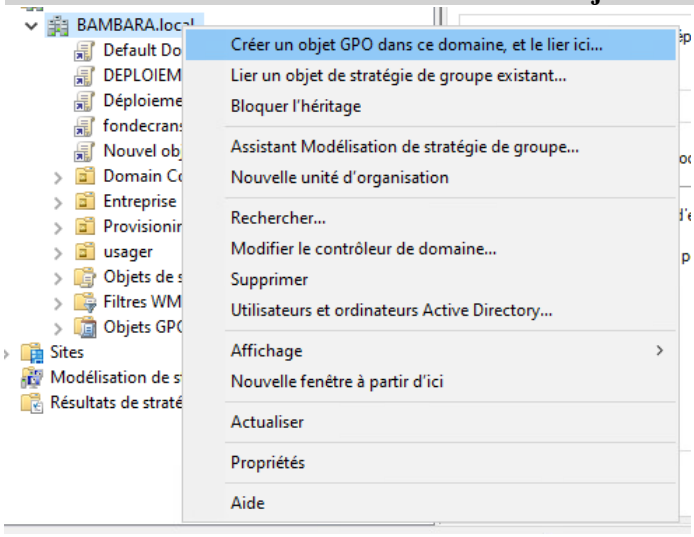
Ensuite, depuis le Gestionnaire de serveur, ouvrez la console de configuration des GPO :

Outils > Gestion des stratégies de groupe.



Maintenant, créez une nouvelle GPO sur l'ensemble de votre domaine :

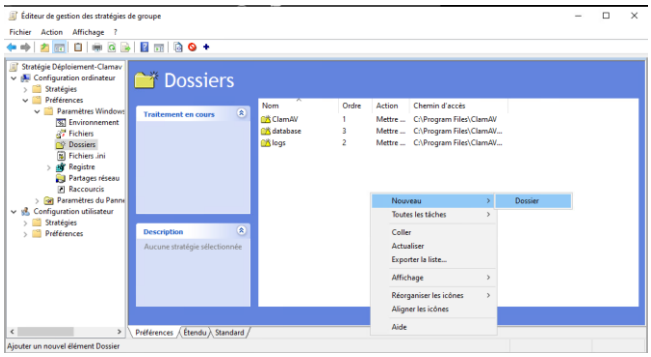
Clic droit sur votre domaine > Créer un objet GPO dans ce domaine.



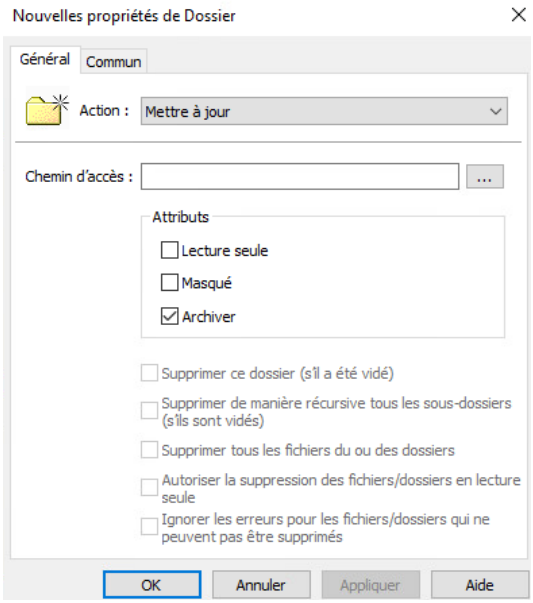
Nommez la GPO comme vous le souhaitez.

Ensuite, modifiez la GPO et allez dans Configuration de l'ordinateur > Préférences > Paramètres Windows > Dossiers.

Ensuite, nous devrons créer 3 dossiers avec des chemins différents. Pour créer un dossier, faites **Clic droit > Nouveau.**



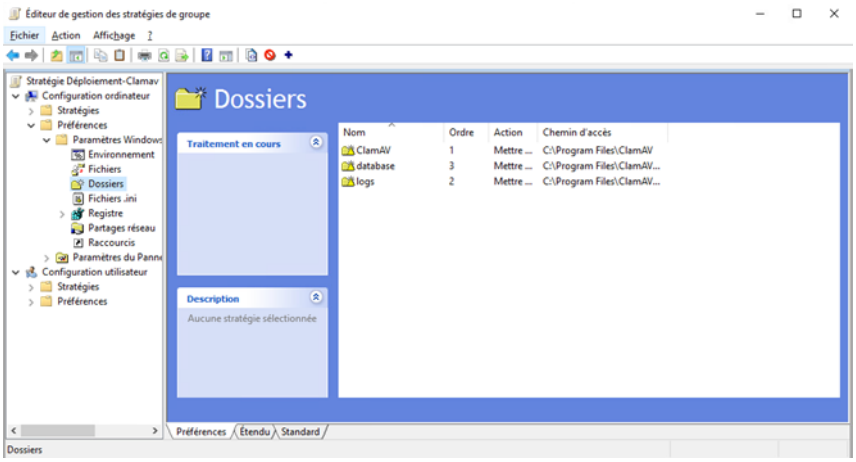
Ensuite, comme action, mettez **Mettre à jour**.



Ensuite, comme chemin d'action, veuillez définir les chemins suivants pour les 3 dossiers :

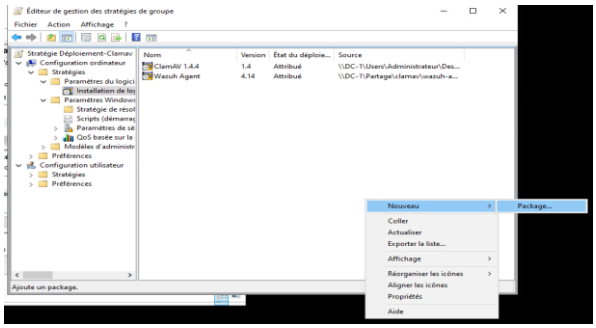
C:\Program Files\ClamAV
C:\Program Files\ClamAV\logs
C:\Program Files\ClamAV\database

Normalement, vous obtiendrez ce résultat.

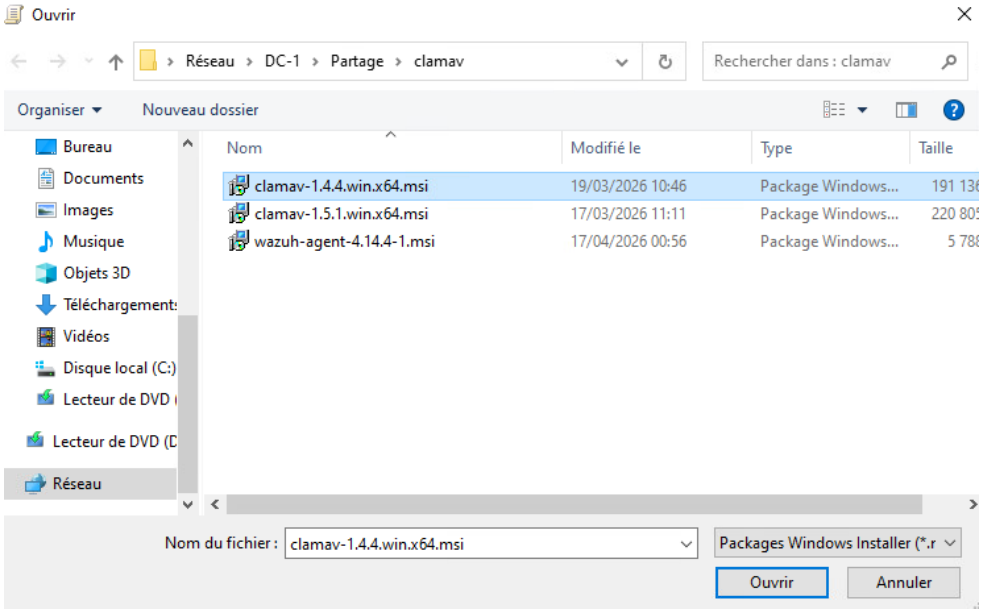


Cette partie de la GPO va nous permettre de déployer les agents. Donc, toujours dans la même GPO :

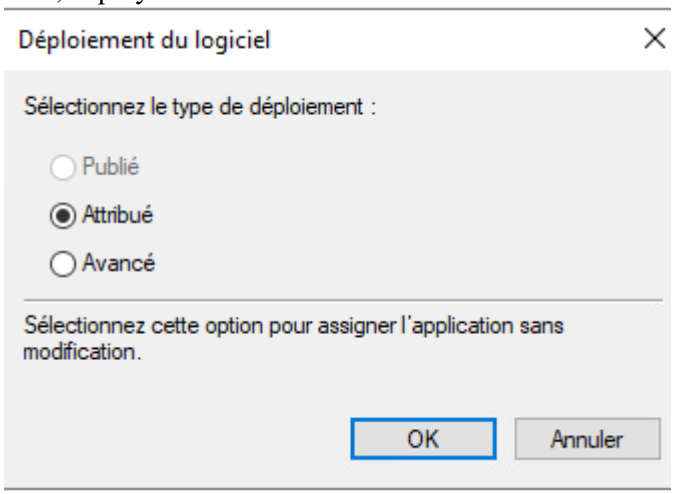
Allez dans Configuration de l'ordinateur > Stratégies > Paramètres du logiciel > Installation du logiciel > Clic droit > Nouveau > Package.



Ensuite, sélectionnez l'installateur de ClamAV dans le dossier partagé.



Puis, déployez-le en mode **Attribué**.



Actuellement, notre GPO permet de déployer les agents ClamAV et de créer les fichiers dont ils ont besoin. Il manque plus qu'à déployer les fichiers de configuration.

Dans la même GPO, allez dans **Configuration de l'ordinateur > Préférences > Paramètres Windows > Fichiers > Nouveau > Fichier**.

Ensuite, comme action, définissez **Remplacer**.
Pour les fichiers source et destination, définissez-les selon les fichiers que nous allons déployer :

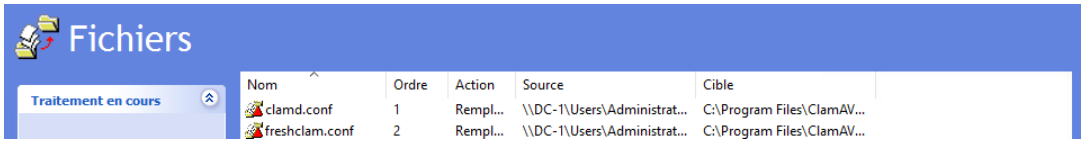
Clamd.conf

- Source : dossier partagé\clamd.conf
- Destination : C:\Program Files\ClamAV\

Freshclam.conf

- Source : dossier partagé\freshclam.conf
- Destination : C:\Program Files\ClamAV\

Vous devriez obtenir un résultat similaire.



Actuellement notre GPO est opérationnel pour déployer les agents.

Création et déploiement du script

Maintenant, nous allons ajouter un script qui, à chaque démarrage du PC, effectuera un scan via ClamAV pour détecter d'éventuels virus.

Nous allons nous baser sur un script déjà existant disponible depuis ce lien GitHub.

Voici le script avec les variables que nous avons ajoutées :

```
#English Language Edition
#Created by Daniel Amorim

# Remove PowerShell Script Execution Restriction for the current user
Set-ExecutionPolicy Unrestricted -Scope CurrentUser

# Define variables

# Directory for log files
$logDir = "C:\Program Files\ClamAV\logs"

# Get current date and time in Day-Month-Year_Hour_minute format
#Change it at will
$timestamp = Get-Date -Format dd-MM-yyyy_HH-mm

# Create the log file name with the timestamp
$logFile = Join-Path $logDir "clamscan.log"

# Create the user based on the person logged on
$domainUser = whoami

# Remove the domain name in the username returned
$currentUser = ($domainUser -split '\\')[1]

# Path to the folder to be scanned
$scanPath = "C:\Users\Administrateur\Downloads"
# ici on définit ce chemin dans le cadre de nos test

# Path to the clamdscan executable
$clamdscanPath = "C:\Program Files\ClamAV\clamdscan.exe"

# Path to the clamscan executable
$clamscanPath = "C:\Program Files\ClamAV\clamscan.exe"

# Path to the Database
$DatabaseDir = "C:\Program Files\ClamAV\database"

# Path for Temporary Quarantined files
$TempDir = "C:\Program Files\ClamAV\quarantine"

# Database Update
```

```
$freshclamPath = "C:\Program Files\ClamAV\freshclam.exe"

# Define the name of the Scheduled Task to be created
$TaskName = "ClamScanTask"

# Check and create directories
$dirs = @($logDir, $TempDir, $DatabaseDir)
foreach ($dir in $dirs) {
   if (!(Test-Path $dir)) {
       New-Item -ItemType Directory -Path $dir
   }
}

# Check if Scheduled Task "ClamScanTask" exists, if not, create it
$task = Get-ScheduledTask

if ($task.TaskName -eq $TaskName) {

   try {
       # Get current date and time in dd-MM-yyyy_HH-mm format
       $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
       $timeStart = Get-Date -Format HH:mm

           # Create the log file name with the timestamp
       $logFile = Join-Path $logDir "clamscan.log"

       schtasks /CHANGE /TR "'$clamscanPath' -r -i -o '$scanPath' --database='$DatabaseDir'
--move='$TempDir' --log='$logFile' " /RU SYSTEM /ST $timestart /TN "$TaskName"

       # Get current date and time in dd-MM-yyyy_HH-mm-ss format
       $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm-ss
       $logTaskFile = Join-Path $logDir "log_task_creation_$timestamp.txt"
       Write-Host "The task $TaskName exists. -- $timestamp"
       Write-Host "Task $TaskName EDITED successfully. -- $timestamp"
       echo "The task $TaskName exists. -- $timestamp" | Out-File -FilePath $logTaskFile -
Append
       echo "Task $TaskName EDITED successfully. -- $timestamp" | Out-File -FilePath
$logTaskFile -Append
           }

   catch {

       # Get current date and time in dd-MM-yyyy_HH-mm format
       $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
       $logTaskFile = Join-Path $logDir "log_task_creation_$timestamp.txt"
       Write-Host "ERROR editing task $TaskName -- $timestamp "
       echo "ERROR editing task $TaskName -- $timestamp -" | Out-File -FilePath
$logTaskFile -Append
   }
}
```

```
if (!($task.TaskName -eq $TaskName)) {

   try {
      schtasks /create /sc hourly /mo 21 /tn $TaskName /RU SYSTEM /tr "'$clamscanPath' -r -
i -o '$scanPath' --database='$DatabaseDir' --move='$TempDir' --log='$logFile' "
      # Get current date and time in dd-MM-yyyy_HH-mm format
      $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
      $logTaskFile = Join-Path $logDir "log_task_creation_$timestamp.txt"
      Write-Host "Task $TaskName CREATED successfully. -- $timestamp"
      echo "Task $TaskName CREATED successfully. -- $timestamp" | Out-File -FilePath
$logTaskFile -Append

   }
   catch {
      $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
      $logTaskFile = Join-Path $logDir "log_task_creation_$timestamp.txt"
      Write-Host "ERROR creating task $TaskName -- $timestamp"
      echo "ERROR creating task $TaskName -- $timestamp -" | Out-File -FilePath
$logTaskFile -Append # Include error message
         }
}

# New-Item -Path "C:\Program Files\ClamAV\logs\freshclam_$timestamp.log" -ItemType
File -Force
$timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
$logDatabaseFile = Join-Path $logDir "freshclam.txt"

# Execute Database Update
& $freshclamPath --datadir=$DatabaseDir --log=$logDatabaseFile

# Check for errors
if ($LastExitCode -ne 0) {
                              $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
                              echo "ERROR during DATABASE UPDATE. -- $timestamp $_ " |
Out-File -FilePath $logDatabaseFile -Append
                           }

# Clamav's Main Commands
#-r Recursive
#-i Only print Infected files
#-o --suppress-ok-results Skip printing OK files

# Execute clamscan
& $clamscanPath -r -i -o $scanPath --database=$DatabaseDir --move=$TempDir --
log=$logFile

# Create a folder with logs using some machine data - hostname and IP, useful for uploading
files to another server, THIS PART IS STILL UNDER DEVELOPMENT.
$hostname = hostname
```

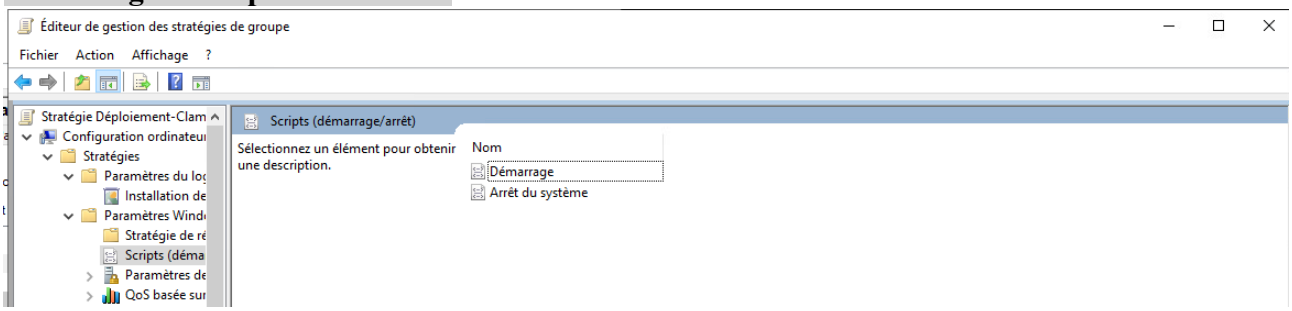
```
$ip = Get-NetIPAddress -AddressFamily IPv4 | Where-Object {$_.InterfaceAlias -NotLike
"Loopback Pseudo-Interface*"} |Select-Object -ExpandProperty IPAddress
$source = "C:\Program Files\ClamAV\logs"
$destination = "C:\Program Files\ClamAV\logs\-$hostname-$ip"

if (!(Test-Path -Path $destination)) {
    New-Item -ItemType Directory -Path $destination | Out-Null
}

# Get all files in the source folder
$files = Get-ChildItem -Path $source -File

# Loop to copy each file
foreach ($file in $files) {
    $destinationFile = Join-Path -Path $destination -ChildPath $file
    Copy-Item -Path $file.FullName -Destination $destinationFile
}

# Check for errors
#if ($LastExitCode -ne 0) {
#    $timestamp = Get-Date -Format dd-MM-yyyy_HH-mm
#    echo "ERROR during scanning. -- $timestamp $_ " | Out-File -FilePath $logFile -Append
#}

$watcher = New-Object System.IO.FileSystemWatcher
$watcher.Path = $scanPath
$watcher.IncludeSubdirectories = $true
$watcher.Filter = "*.*"
$watcher.EnableRaisingEvents = $true

Register-ObjectEvent $watcher Created -Action {
    Start-Sleep -Milliseconds 500
    $filePath = $Event.SourceEventArgs.FullPath
    & "$clamscanPath" -r -i -o "$filePath" `
        --database="$DatabaseDir" `
        --move="$TempDir" `
        --log="$logFile"
}

Register-ObjectEvent $watcher Changed -Action {
    Start-Sleep -Milliseconds 500
    $filePath = $Event.SourceEventArgs.FullPath
    & "$clamscanPath" -r -i -o "$filePath" `
        --database="$DatabaseDir" `
        --move="$TempDir" `
        --log="$logFile"
}

# Keep the script alive
while ($true) {
```

```
    Start-Sleep -Seconds 5}
```

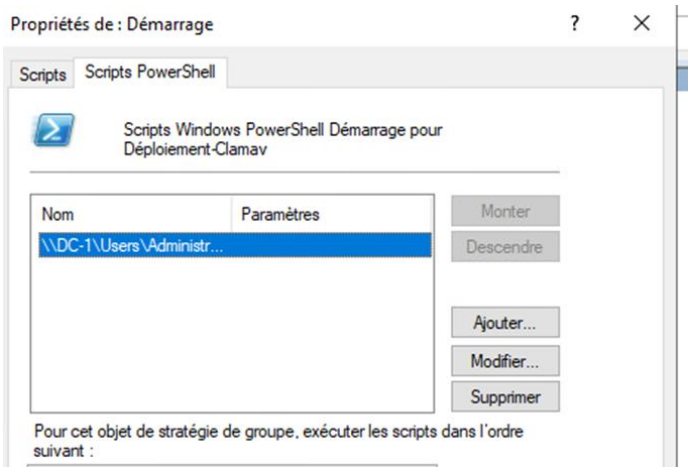
Une fois le script créé, nous allons le déployer par GPO.

Vous pouvez utiliser la GPO précédente pour cela.

Allez dans **Configuration de l'ordinateur > Stratégies > Paramètres Windows > Scripts > Démarrage > Scripts PowerShell.**



Maintenant, ajoutez le script précédemment créé depuis le dossier partagé.



Une fois cela fait, notre GPO sera complète et prête à déployer les agents ClamAV ainsi qu'à scanner le poste à son démarrage.

Création du server Wazuh

Nous allons déployer notre serveur Wazuh via Docker.
Pour ce faire, installez Docker et Git avec les commandes suivantes :

```
sudo apt install docker docker-compose
sudo apt install git
```

Ensuite, nous allons suivre la documentation officielle de Wazuh. Il faudra cloner le dépôt Git en local :

git clone https://github.com/wazuh/wazuh-docker.git -b v4.14.4

Ensuite, nous allons dans les fichiers du dépôt :

cd wazuh-docker/single-node/

Puis, exécutez cette commande pour générer un certificat automatiquement :

docker compose -f generate-indexer-certs.yml run --rm generator

Ensuite, nous démarrons le serveur avec cette commande :

docker compose up -d

Attention : si vous déployez votre serveur Wazuh sur la même machine que votre serveur ClamAV, veuillez faire attention à ce que les deux services n'utilisent pas les mêmes ports.

Ensuite, vous pouvez vous connecter à l'adresse suivante :

https://<DOCKER_HOST_IP>

Les identifiants par défaut sont les suivants :

Username : admin
Password : SecretPassword

Ensuite, nous allons déployer les agents.
Nous devons définir un mot de passe pour que les agents puissent se connecter au serveur.
Commencez par exécuter la commande suivante pour accéder au conteneur du serveur :

docker exec -it <id_de_votre_serveur> /bin/bash

Une fois dans le serveur, exécutez :

cd /var/ossec/etc/
echo "votreMDP" > authd.pass

Ensuite, changez les permissions :

chmod 640 /var/ossec/etc/authd.pass
chown root:wazuh /var/ossec/etc/authd.pass

Quittez le conteneur et redémarrez les conteneurs :

exit
docker compose down
docker compose up -d

Déploiement des agents Wazuh

Ensuite, vous pouvez télécharger l'agent depuis le site officiel.

Nous allons déployer la version suivante : **4.14.4.1**.

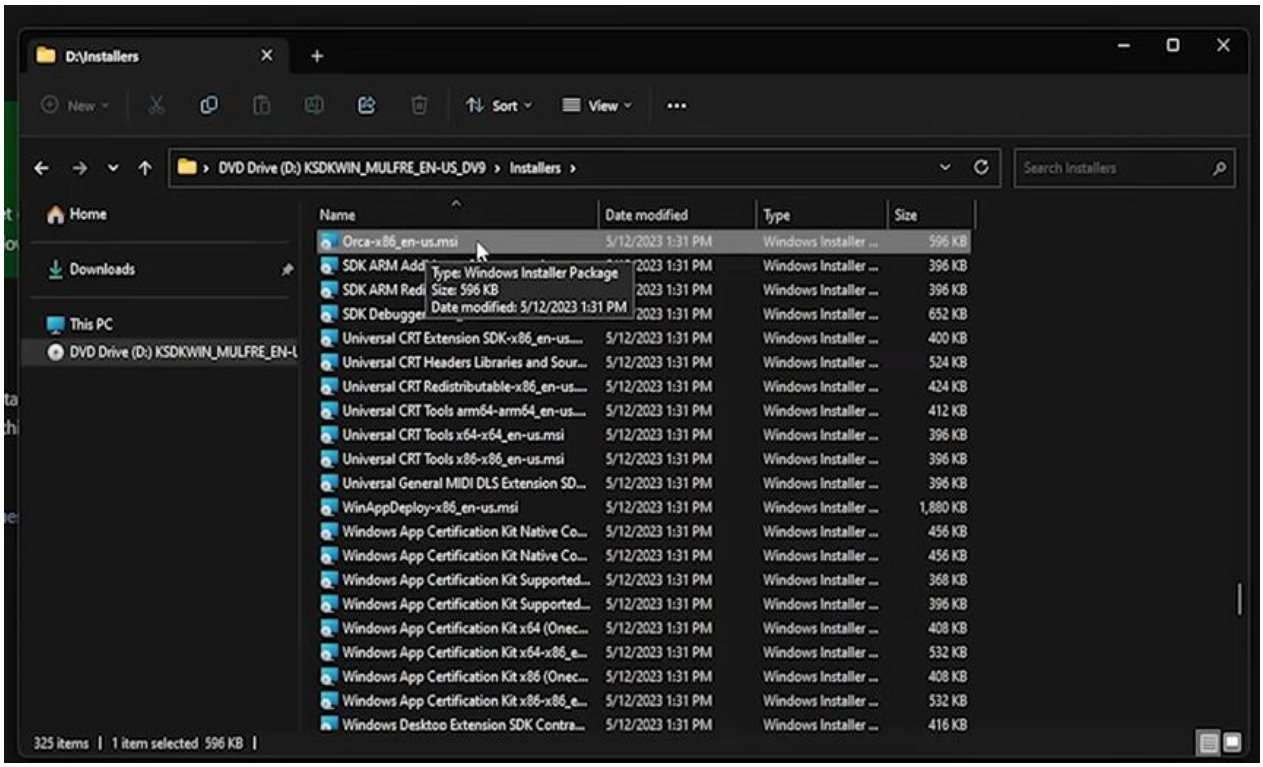
Pour que les agents se connectent automatiquement, nous devrons ajouter des paramètres. Pour cela, nous allons installer Orca et générer un fichier .mst afin d'ajouter ces configurations au déploiement.

Pour ce faire, téléchargez Orca via les composants du SDK.

Suivez ce lien :

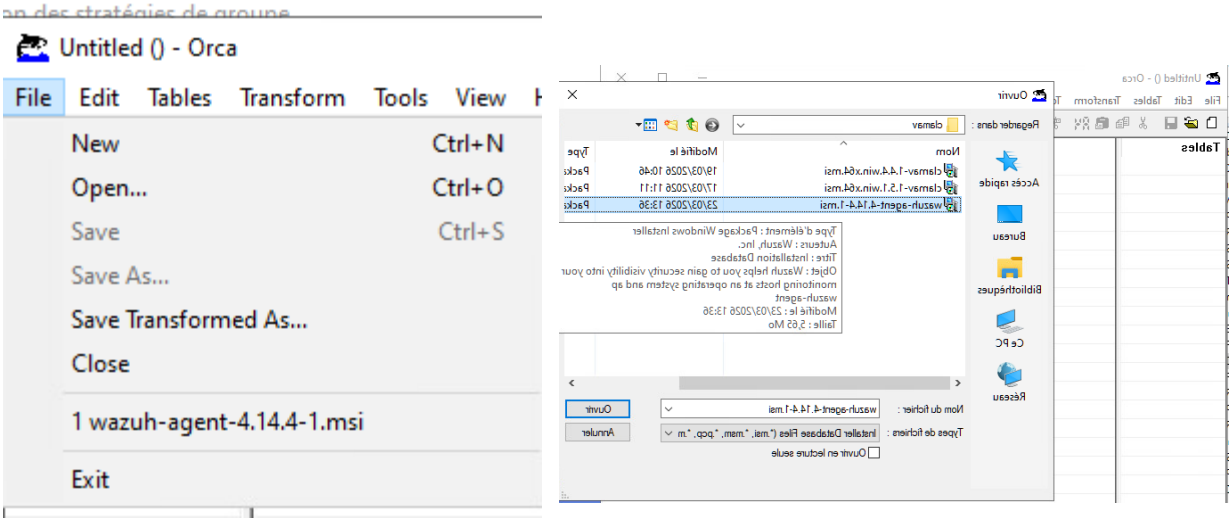
https://learn.microsoft.com/fr-fr/windows/apps/windows-sdk/downloads

Une fois votre version sélectionnée, montez l'ISO. Depuis le fichier de l'ISO, allez dans le dossier « **Installers** » (ou « **Installer** »), puis sélectionnez **Orca**.

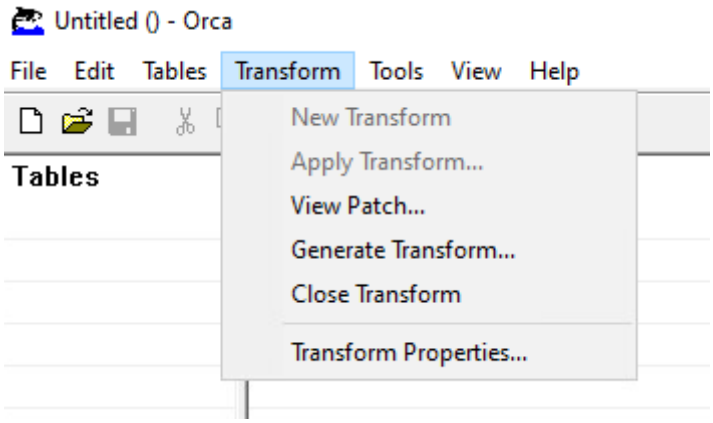


Une fois Orca installé, lancez-le.

Une fois dans Orca, dans l'onglet **File**, cliquez sur **Open** et sélectionnez votre installeur de l'agent Wazuh.



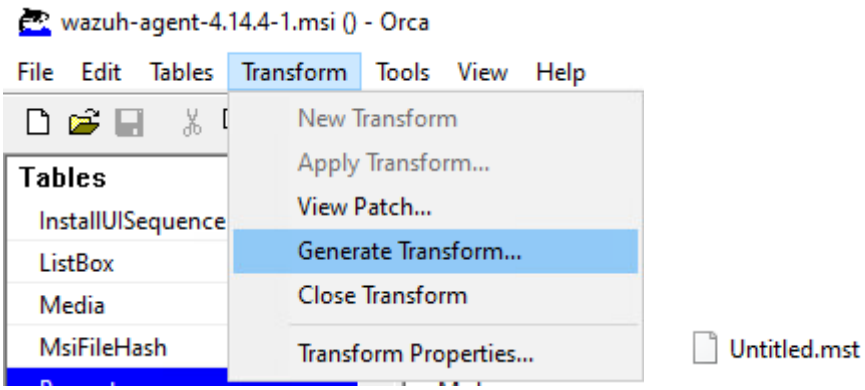
Ensuite, cliquez sur **Transform > New Transform**.



Ensuite, dans **Tables > Property**, faites un clic droit et sélectionnez **Add row**. Puis ajoutez les valeurs suivantes en créant une nouvelle ligne pour chaque valeur :

ADDRESS : « IP de votre Wazuh »
AUTHD_SERVER : « IP de votre Wazuh »
PROTOCOL : TCP
PASSWORD : « MDP défini plus tôt »

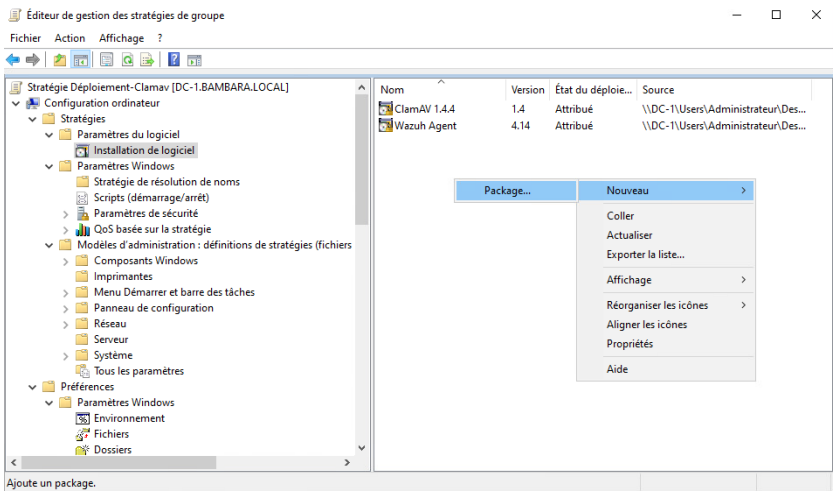
Une fois cela fait**, dans Transform > Generate Transform**, sauvegardez votre fichier en .mst (dans notre cas untitled.mst, le nom importe peu). Enregistrez bien votre .mst dans votre dossier partagé.



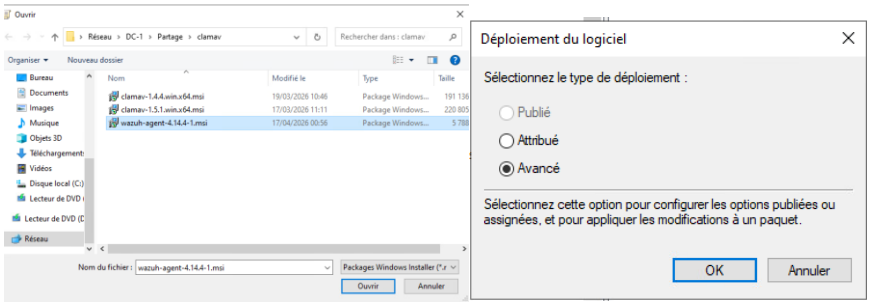
Ensuite, nous allons déployer l'agent avec notre .mst.

Depuis la console de gestion des stratégies de groupe, allez dans votre GPO de déploiement de ClamAV.

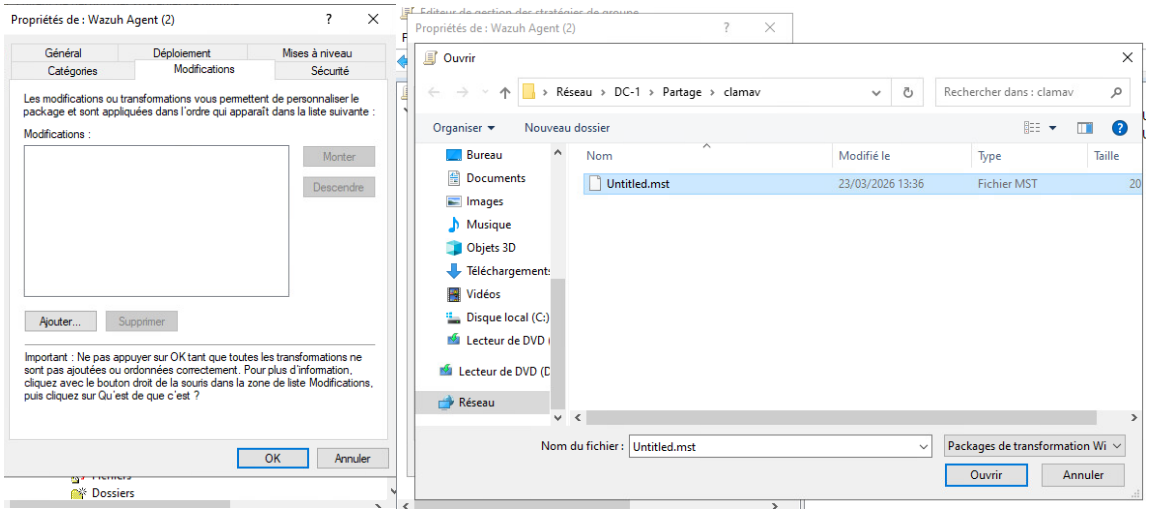
Puis allez dans **Configuration de l'ordinateur > Stratégies > Paramètres du logiciel > Installation du logiciel > Clic droit > Nouveau > Package**.



Ensuite, sélectionnez votre agent Wazuh et cochez bien la case **Avancé.**

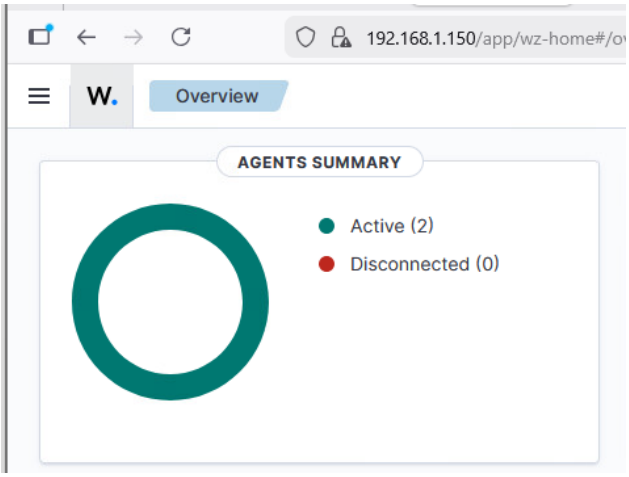


Allez dans Modification, cliquez sur Ajouter, sélectionnez votre fichier .mst, puis cliquez sur OK pour sauvegarder.

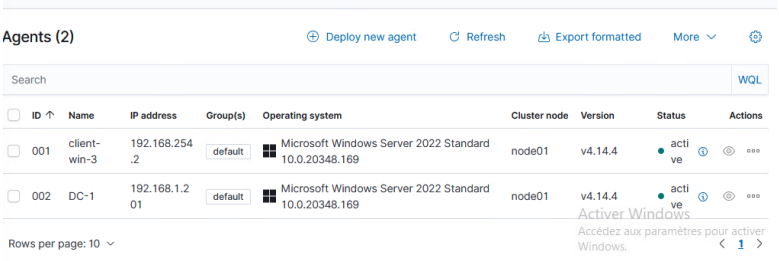


Une fois l'agent déployé, veuillez vérifier qu'il remonte bien sur la console de Wazuh.

Normalement, depuis **Overview,** vous pouvez voir le nombre d'agents actifs.



Et dans **Agent Management > Summary**, vous pourrez voir en détail les agents actifs.



Configuration des remonté des logs vers wazuh

Maintenant, pour autoriser la remontée des logs de ClamAV sur Wazuh, nous allons devoir modifier certains fichiers de configuration dans Wazuh.

Nous allons éditer les fichiers suivants :
filebeat.yml
local_rules.xml
ossec.conf

Récupérez le fichier filebeat.yml puis éditez-le. Vous pouvez le copier sur votre machine locale, puis le remettre dans le conteneur en faisant attention aux droits.

docker cp "id_du_container" /etc/filebeat.yml ./

vi ./filebeat.yml

Ensuite, changez le champ suivant comme suit :

archives:
enabled: true

```
# Wazuh - Filebeat configuration file
filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: true
```

Ensuite, recopiez-le dans le conteneur.

Le fichier local_rules.xml se trouve dans le chemin suivant :
/var/ossec/etc/rules/local_rules.xml

Nous allons suivre la même procédure pour éditer le fichier.

Ici, il faudra ajouter ces champs. Ces 3 champs correspondent au type de logs que Wazuh va remonter depuis ClamAV

```
<rule id="100002" level="10">
 <match>FOUND</match>
 <description>ClamAV detected a Malware</description>
</rule>

 <rule id="100003" level="10">
 <match>WARNING</match>
 <description>ClamAV detected a Malware</description>
</rule>

<rule id="100004" level="10">
 <match>ERROR</match>
 <description>ClamAV detected a Malware</description>
</rule>
```

Ensuite, vous pouvez répéter la procédure précédente pour enregistrer le fichier.

Et finalement, nous allons modifier le fichier **ossec.conf**.

Chemin :
/var/ossec/etc/ossec.conf

Ici, il faut simplement mettre l'option **logall** à yes

```
root@srv:/home/bts/wazuh-docker/single-node# cat ossec.conf
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
```

Vous pouvez répéter la procédure précédente pour enregistrer le fichier.
Ensuite, vous pouvez redémarrer vos conteneurs :

sudo docker compose down
sudo docker compose up -d

Maintenant, nous allons configurer le fichier **ossec.conf** de l'agent.
Nous allons ajouter ces valeurs afin que ClamAV puisse envoyer les logs au serveur Wazuh.

```
<localfile>
  <location>active-response\active-responses.log</location>
  <log_format>syslog</log_format>
</localfile>

<localfile>
  <location>C:\Program Files\ClamAV\logs\clamscan.log</location>
  <log_format>syslog</log_format>
</localfile>

<localfile>
  <location>C:\Program Files\ClamAV\logs\freshclam.txt</location>
  <log_format>syslog</log_format>
</localfile>
```

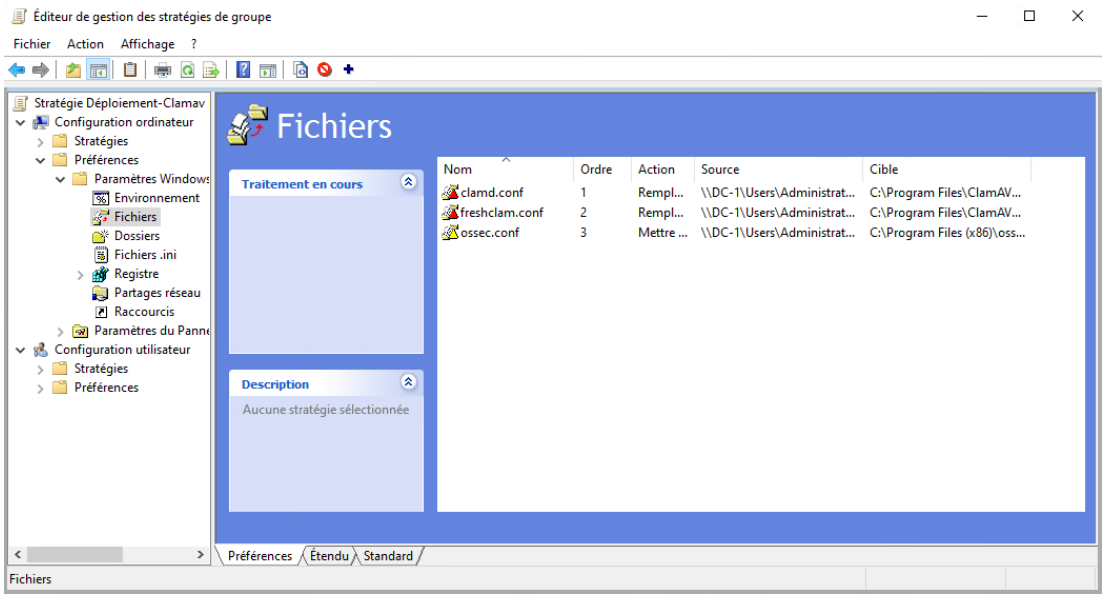
Nous allons devoir déployer ce fichier dans une GPO.

Placez le fichier **ossec.conf** dans votre dossier partagé.

Ensuite, vous pouvez reprendre votre ancienne GPO.

Puis allez dans :

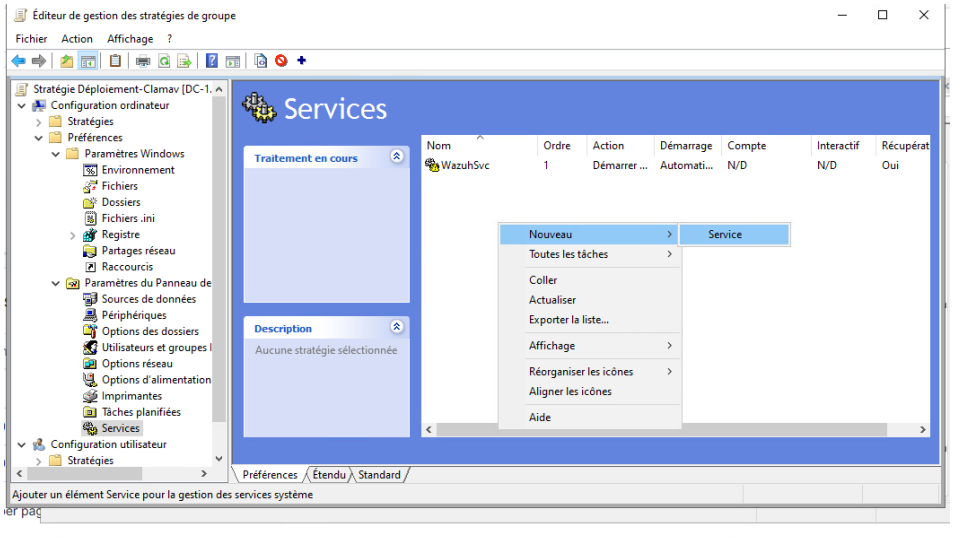
Configuration de l'ordinateur > Préférences > Paramètres Windows > Fichiers > Clic droit > Nouveau > Fichier.



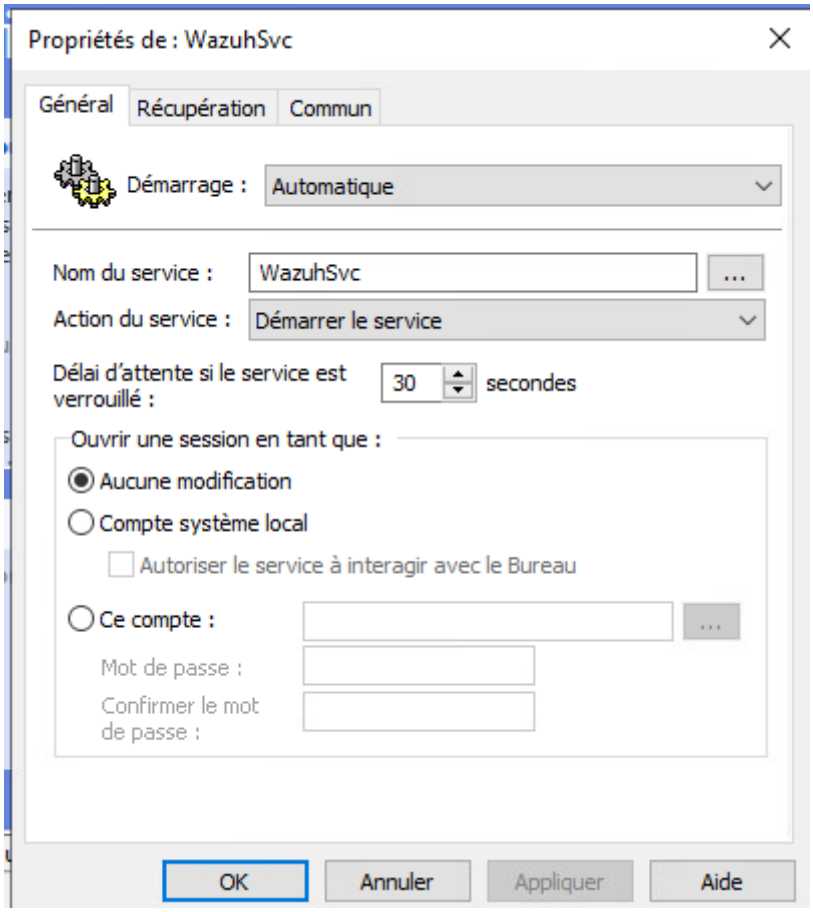
Comme action, sélectionnez Mettre à jour.
Comme source, sélectionnez le fichier dans le dossier partagé.
Comme destination, mettez le chemin suivant :
C:\Program Files (x86)\ossec-agent\ossec.conf

Ensuite, nous allons configurer un service pour que le client Wazuh démarre automatiquement.

Allez dans : **Configuration de l'ordinateur > Préférences > Paramètres du Panneau de configuration > Services > Nouveau > Service.**

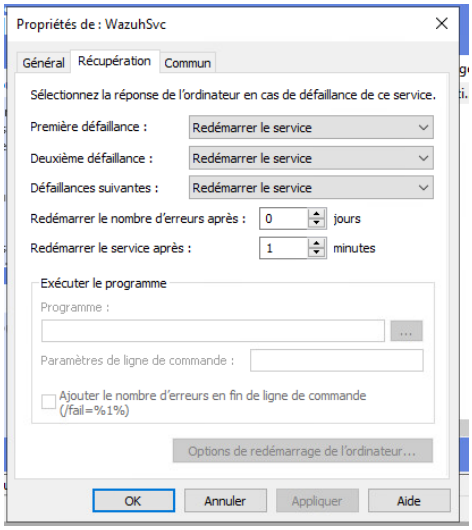


Comme nom, mettez : WazuhSvc
Action du service : Démarrer le service



Ensuite, dans l'onglet Récupération, mettez les options comme suit :

Première défaillance : redémarrer le service
Deuxième défaillance : redémarrer le service
Défaillances suivantes : redémarrer le service



Ensuite, sauvegardez la GPO.

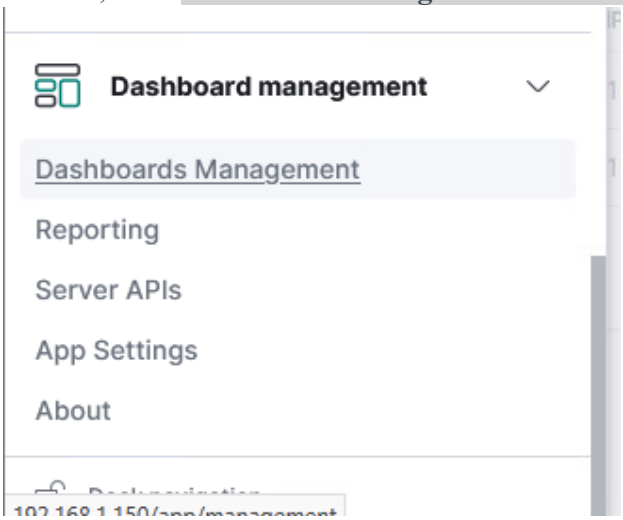
Tous nos fichiers sont configurés correctement. Vous pouvez exécuter la commande suivante sur l'AD et sur les clients pour qu'ils mettent à jour leur GPO :

GPUpdate /force

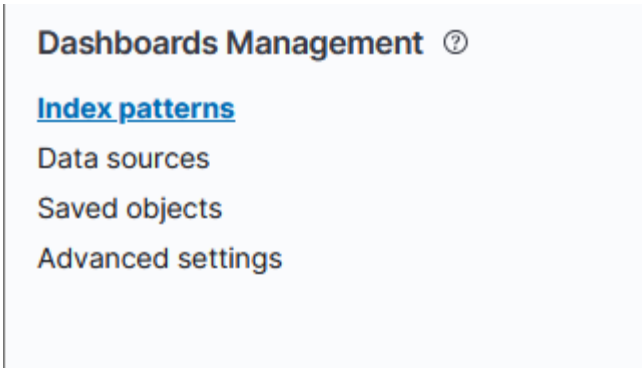
Maintenant, il faut que nous activions les archives sur l'interface web de Wazuh.

Connectez-vous à l'interface web de Wazuh.

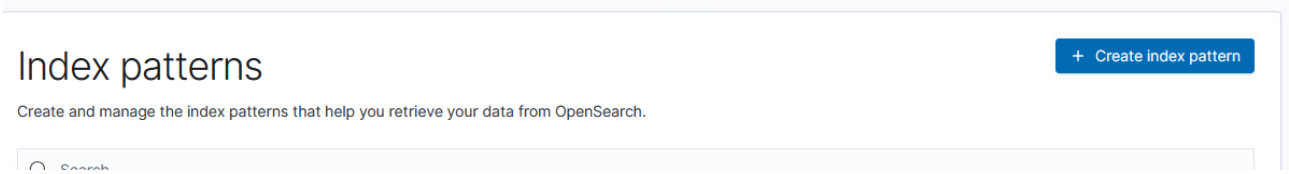
Ensuite, dans **Dashboard Management > Dashboard Management.**



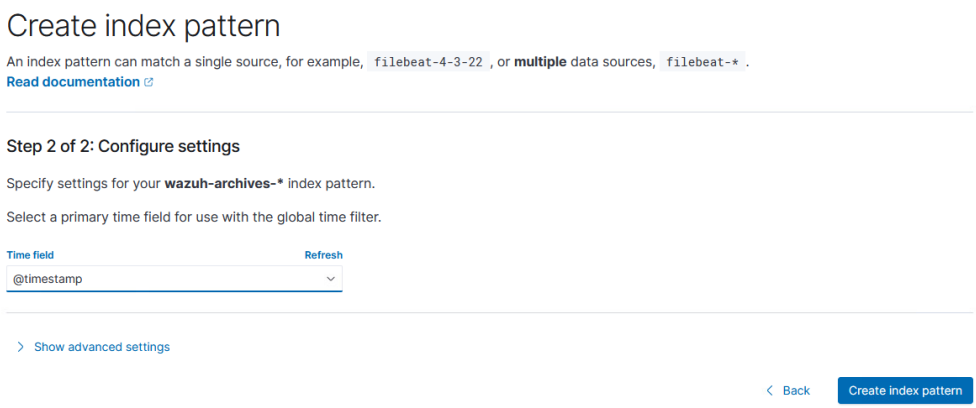
Puis cliquez sur **+Create index pattern.**



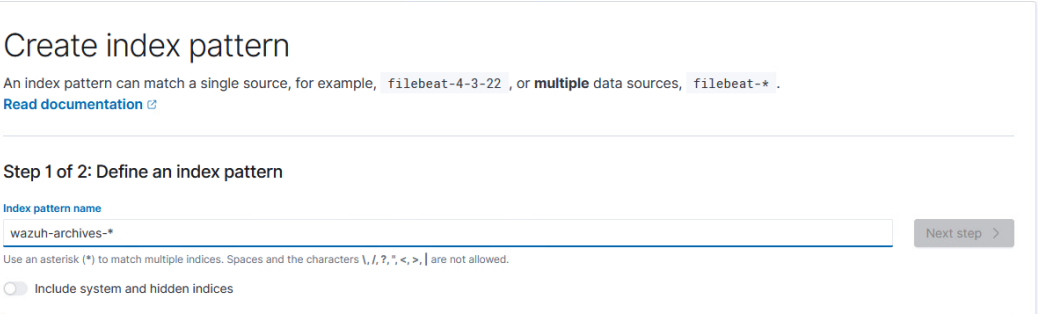
Puis cliquez sur **+ Create index pattern**



Dans "Index pattern", mettez : **wazuh-archives-***

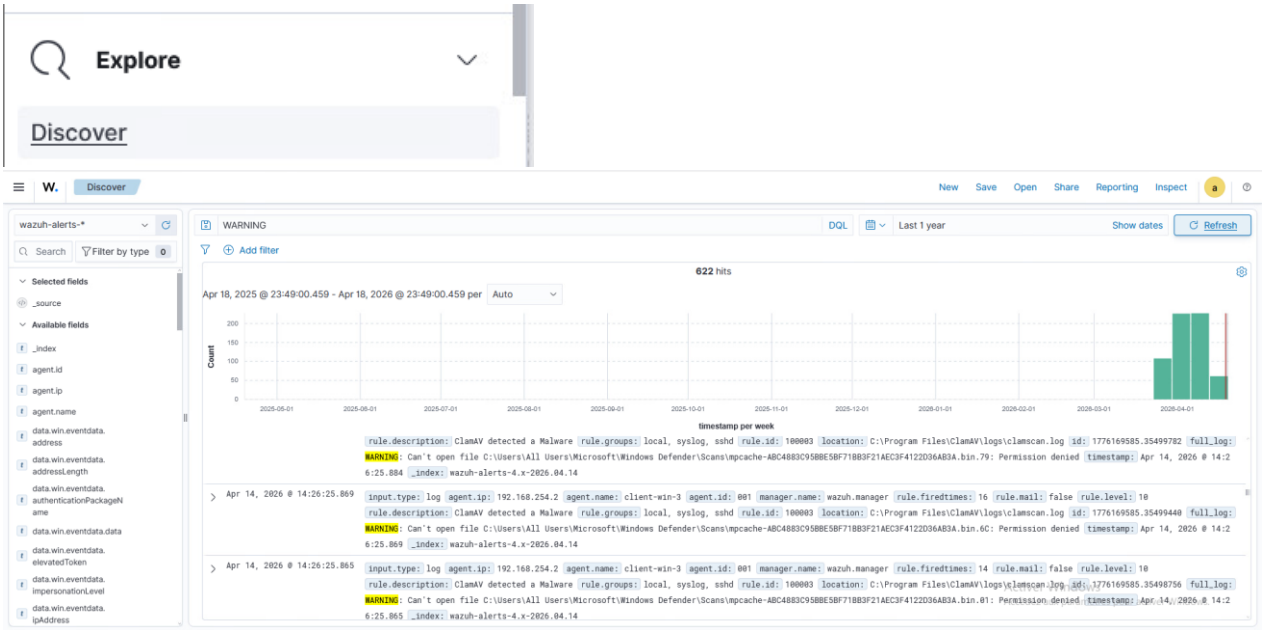


Ensuite sélectionnez **@timestamp** et créez votre pattern



Normalement, si tout fonctionne bien, vous pouvez voir les remontées de Wazuh dans l'onglet suivant :

Explore > Discover



Conclusion

Pour conclure, ClamAV peut être une solution intéressante si vous cherchez un simple antivirus pour une utilisation personnelle. En revanche, pour une utilisation dans un cadre professionnel, je vous le déconseille fortement, car dans le cas où vous voulez déployer une solution EDR sur un parc informatique, ClamAV n'est pas adapté à cela. La solution ClamAV ne présente pas de véritable interface permettant de centraliser tous les agents ni de les gérer. De plus, pour avoir une remontée des logs, nous sommes obligés de passer par une solution tierce.

Malheureusement, dans le cadre de notre AP, ClamAV ne coche pas toutes les cases du cahier des charges imposé car celui-ci ne dispose pas de vraie interface de configuration. Il est à noter qu'une version de ClamAV avec une GUI existe, mais uniquement pour les agents. Il est aussi possible de déployer le serveur ClamAV depuis Docker, mais encore une fois, il n'existe pas actuellement de serveur permettant de gérer de A à Z les agents