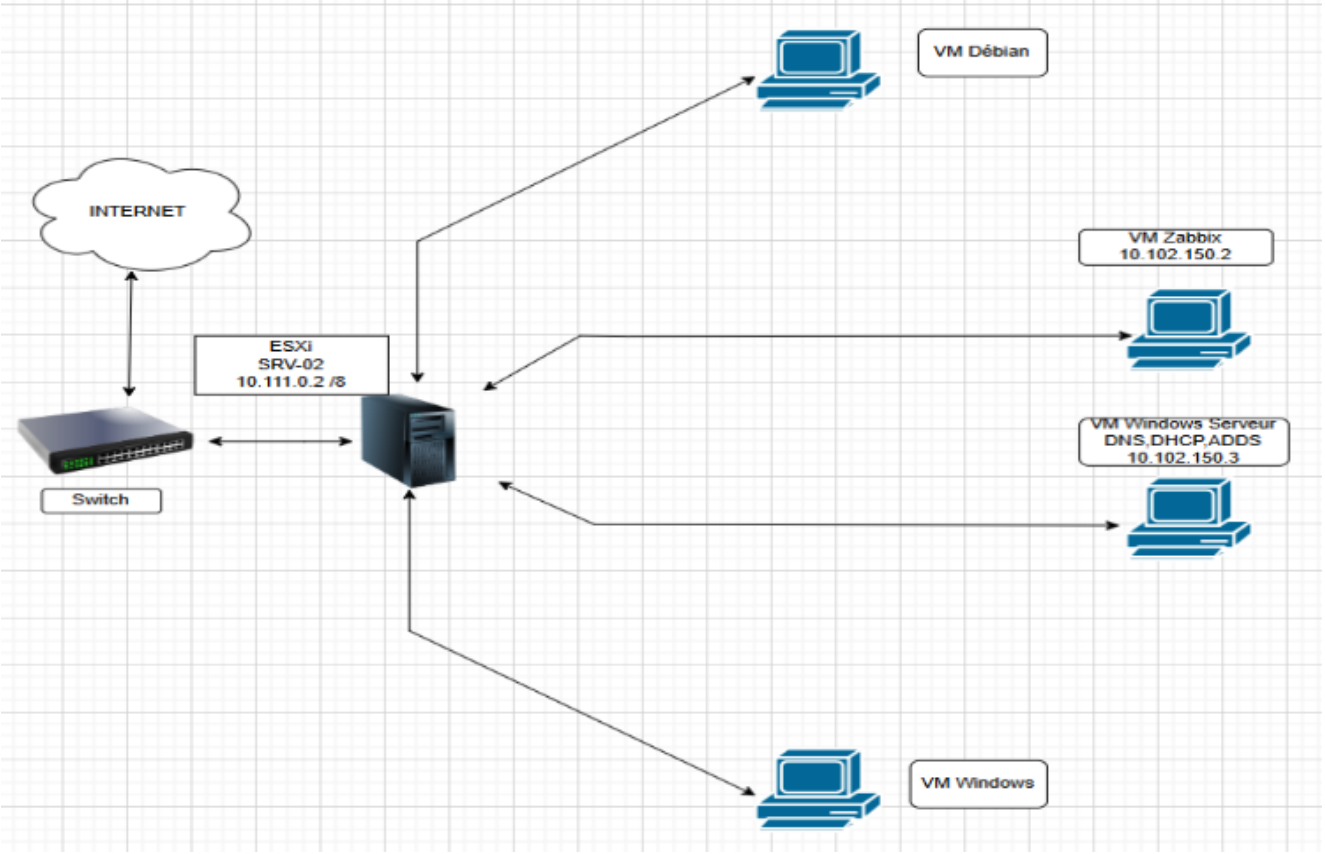
DÉPLOIEMENT DE ZABBIX POUR LA SUPERVISION DES INFRASTRUCTURES D'AUREL INFORMATIQUE

I-Contexte

L'entreprise *Aurel Informatique* rencontre des difficultés à surveiller efficacement ses infrastructures informatiques, ce qui entraîne des retards dans la détection des problèmes et des risques de sécurité. La surveillance étant manuelle, la gestion des incidents et la prévention des défaillances sont complexes. La solution proposée est de déployer **Zabbix** sur un serveur Debian 12 pour centraliser la supervision des systèmes, réseaux et applications, et ainsi améliorer la réactivité, la sécurité et la gestion proactive des infrastructures.

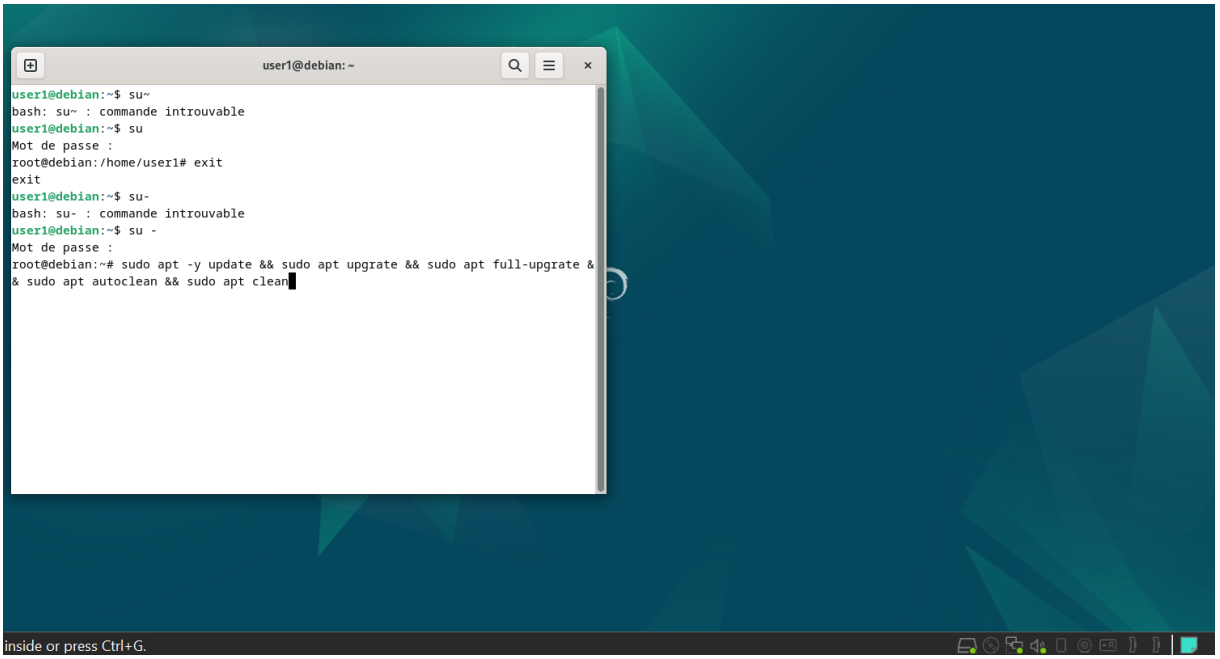


II- Installation de Zabbix

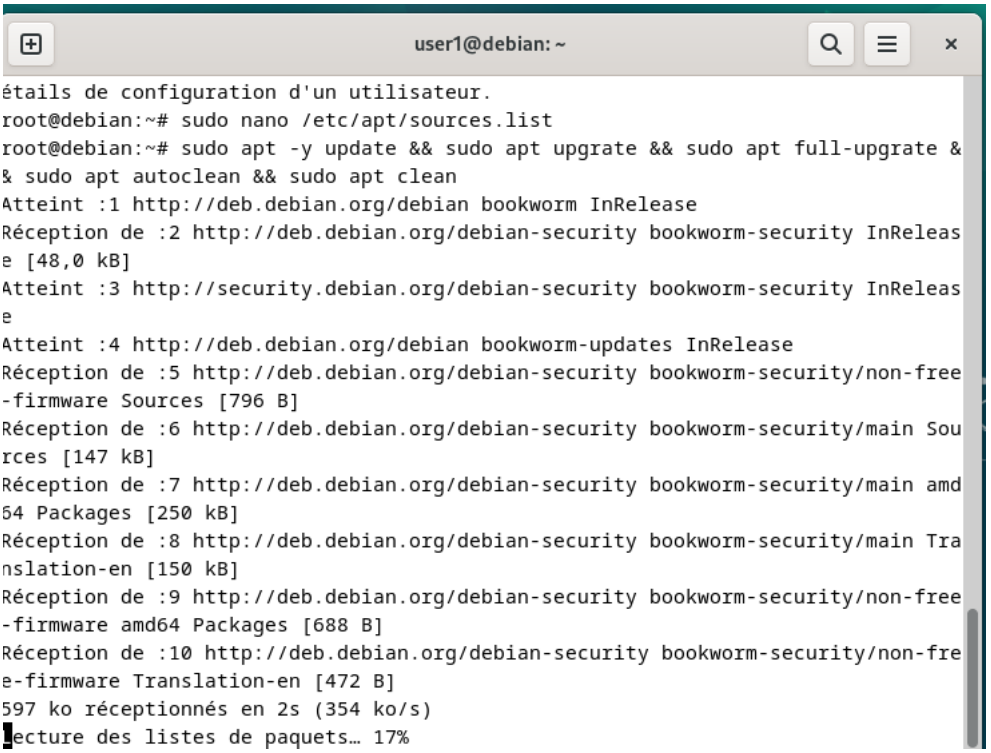
1) Mise à jour du système

Avant d'installer et de configurer Zabbix sur Debian 12, avec MariaDB comme base de données et Apache comme serveur web, il est essentiel de s'assurer que le système est entièrement à jour.

Exécutez la commande suivante pour mettre à jour tous les paquets et nettoyer les fichiers obsolètes : **sudo apt update && sudo apt upgrade -y && sudo apt full-upgrade -y && sudo apt autoclean && sudo apt clean**



Nous patientons jusqu'à la fin de l'installation une fois la commande lancée, nous attendons que l'installation de MariaDB se termine complètement avant de poursuivre.



2) Installation du serveur de base de données MariaDB

La base de données est un élément fondamental de Zabbix, car elle stocke toutes les données de supervision (hôtes, métriques, historiques, alertes, etc.). Nous allons utiliser MariaDB**,** une solution open source compatible avec MySQL. Pour installer le serveur et le client MariaDB, exécutez la commande suivante : **sudo apt install mariadb-server mariadb-client -y**

```
user1@debian: ~
Paramétrage de libcgi-fast-perl (1:2.15-1) ...
Paramétrage de mariadb-client-core (1:10.11.11-0+deb12u1) ...
Paramétrage de libdbd-mariadb-perl (1.22-1+b1) ...
Paramétrage de mariadb-server-core (1:10.11.11-0+deb12u1) ...
Invalid file '/usr/sbin/mysqld' for capability operation
Setcap failed on /usr/sbin/mysqld, required with --memlock if insufficent RLIMIT
_MEMLOCK
Paramétrage de mariadb-client (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-server (1:10.11.11-0+deb12u1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /l
ib/systemd/system/mariadb.service.
Paramétrage de mariadb-plugin-provider-bzip2 (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-lzma (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-lzo (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-lz4 (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-snappy (1:10.11.11-0+deb12u1) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) .
..
Traitement des actions différées (« triggers ») pour mariadb-server (1:10.11.11-
0+deb12u1) ...
root@debian:~# sudo apt install mariadb-server mariadb-client -y
```

Nous patientons jusqu'à la fin de l'installation **u**ne fois la commande lancée, nous attendons que l'installation de MariaDB se termine complètement avant de poursuivre.

```
user1@debian: ~
'aramétrage de libfcgi-bin (2.4.2-2) ...
'aramétrage de libhtml-template-perl (2.97-2) ...
'aramétrage de socat (1.7.4.4-2) ...
'aramétrage de libmariadb3:amd64 (1:10.11.11-0+deb12u1) ...
'aramétrage de pv (1.6.20-1) ...
'aramétrage de libfcgi-perl (0.82+ds-2) ...
'aramétrage de libterm-readkey-perl (2.38-2+b1) ...
'aramétrage de liburing2:amd64 (2.3-3) ...
'aramétrage de libdbi-perl:amd64 (1.643-4) ...
'aramétrage de rsync (3.2.7-1+deb12u2) ...
:sync.service is a disabled or a static unit, not starting it.
'aramétrage de libcgi-fast-perl (1:2.15-1) ...
'aramétrage de mariadb-client-core (1:10.11.11-0+deb12u1) ...
'aramétrage de libdbd-mariadb-perl (1.22-1+b1) ...
'aramétrage de mariadb-server-core (1:10.11.11-0+deb12u1) ...
:nvalid file '/usr/sbin/mysqld' for capability operation
;etcap failed on /usr/sbin/mysqld, required with --memlock if insufficent RLIMIT
_MEMLOCK
'aramétrage de mariadb-client (1:10.11.11-0+deb12u1) ...
'aramétrage de mariadb-server (1:10.11.11-0+deb12u1) ...
:reated symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /l
.b/systemd/system/mariadb.service.

rogression : [ 90%] [##############################################.......]
```

3) Démarrage et sécurisation de MariaDB

Nous allons maintenant démarrer le service MariaDB, l'activer pour qu'il se lance automatiquement au démarrage du système, puis procéder à sa sécurisation de base à l'aide du script intégré **mysql_secure_installation**.Exécutez les commandes suivantes :

sudo systemctl start mariadb et **sudo systemctl enable mariadb**

```
user1@debian: ~
Paramétrage de libdbd-mariadb-perl (1.22-1+b1) ...
Paramétrage de mariadb-server-core (1:10.11.11-0+deb12u1) ...
Invalid file '/usr/sbin/mysqld' for capability operation
Setcap failed on /usr/sbin/mysqld, required with --memlock if insufficent RLIMIT
_MEMLOCK
Paramétrage de mariadb-client (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-server (1:10.11.11-0+deb12u1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /l
ib/systemd/system/mariadb.service.
Paramétrage de mariadb-plugin-provider-bzip2 (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-lzma (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-lzo (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-lz4 (1:10.11.11-0+deb12u1) ...
Paramétrage de mariadb-plugin-provider-snappy (1:10.11.11-0+deb12u1) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) .
..
Traitement des actions différées (« triggers ») pour mariadb-server (1:10.11.11-
0+deb12u1) ...
root@debian:~# sudo systemctl start mariabdb
Failed to start mariabdb.service: Unit mariabdb.service not found.
root@debian:~# ^C
root@debian:~# sudo systemctl start mariadb
root@debian:~# sudo systemctl enable mariadb
```

Ensuite, nous lançons le script de sécurisation de MariaDB : **sudo mysql_secure_installation**

```
user1@debian: ~
root@debian:~# ^C
root@debian:~# sudo systemctl start mariadb
root@debian:~# sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /lib/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
root@debian:~# sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n]
```

```
user1@debian: ~

      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
```

Remarque importante :
Lors de l'exécution du script mysql_secure_installation, nous appuyons sur **y** à toutes les étapes, sauf lorsqu'il s'agit de définir un mot de passe (si l'utilisateur root en a déjà un).Pour éviter tout problème lors de la configuration de MariaDB, il est recommandé de **ne pas utiliser** les caractères spéciaux dans les mots de passe : **Aurel123**

```
user1@debian: ~

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
```

Ajout du dépôt officiel Zabbix pour pouvoir installer la dernière version stable de Zabbix, nous devons ajouter son dépôt officiel à notre système Debian 12.Nous commençons par télécharger le paquet du dépôt avec la commande suivante **: wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zabbix-release_latest+debian12_all.deb**

```
user1@debian: ~
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@debian:~# wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zab
bix-release_latest+debian12_all.deb
```

```
user1@debian: ~

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@debian:~# wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zab
bix-release_latest+debian12_all.deb
--2025-03-26 15:34:27--  https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-releas
e/zabbix-release_latest+debian12_all.deb
Résolution de repo.zabbix.com (repo.zabbix.com)… 178.128.6.101, 2604:a880:2:d0::2062:d001
Connexion à repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 8096 (7,9K) [application/octet-stream]
Sauvegarde en : « zabbix-release_latest+debian12_all.deb »

zabbix-release_latest+ 100%[===========================>]   7,91K  --.-KB/s    ds 0s

2025-03-26 15:34:27 (219 MB/s) — « zabbix-release_latest+debian12_all.deb » sauvegardé [8096
/8096]

root@debian:~#
```

Une fois le fichier téléchargé, nous l'installons à l'aide de la commande suivante **: sudo dpkg -i zabbix-release_latest+debian12_all.deb**

```
user1@debian: ~
e/zabbix-release_latest+debian12_all.deb
Résolution de repo.zabbix.com (repo.zabbix.com)… 178.128.6.101, 2604:a880:2:d0::2062:d001
Connexion à repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 8096 (7,9K) [application/octet-stream]
Sauvegarde en : « zabbix-release_latest+debian12_all.deb »

zabbix-release_latest+ 100%[===========================>]   7,91K  --.-KB/s    ds 0s

2025-03-26 15:34:27 (219 MB/s) – « zabbix-release_latest+debian12_all.deb » sauvegardé [8096
/8096]

root@debian:~# sudo dpkg -i zabbix-release_latest+debian_all.deb
dpkg: erreur: ne peut pas accéder à l'archive « zabbix-release_latest+debian_all.deb »: Aucu
n fichier ou dossier de ce type
root@debian:~# ^C
root@debian:~# sudo dpkg -i zabbix-release-latest+debian12_all.deb
dpkg: erreur: ne peut pas accéder à l'archive « zabbix-release-latest+debian12_all.deb »: Au
cun fichier ou dossier de ce type
root@debian:~# sudo dpkg -i zabbix-release_latest+debian12_all.deb
Sélection du paquet zabbix-release précédemment désélectionné.
(Lecture de la base de données... 155815 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de zabbix-release_latest+debian12_all.deb ...
Dépaquetage de zabbix-release (1:7.0-2+debian12) ...
Paramétrage de zabbix-release (1:7.0-2+debian12) ...
root@debian:~#
```

Après l'ajout du dépôt, nous mettons à jour la liste des paquets disponibles pour que le système prenne en compte les nouveaux packages Zabbix : **sudo apt update**

```
user1@debian: ~
e/zabbix-release_latest+debian12_all.deb
Résolution de repo.zabbix.com (repo.zabbix.com)… 178.128.6.101, 2604:a880:2:d0::2062:d001
Connexion à repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 8096 (7,9K) [application/octet-stream]
Sauvegarde en : « zabbix-release_latest+debian12_all.deb »

zabbix-release_latest+ 100%[===========================>]   7,91K  --.-KB/s    ds 0s

2025-03-26 15:34:27 (219 MB/s) – « zabbix-release_latest+debian12_all.deb » sauvegardé [8096
/8096]

root@debian:~# sudo dpkg -i zabbix-release_latest+debian_all.deb
dpkg: erreur: ne peut pas accéder à l'archive « zabbix-release_latest+debian_all.deb »: Aucu
n fichier ou dossier de ce type
root@debian:~# ^C
root@debian:~# sudo dpkg -i zabbix-release-latest+debian12_all.deb
dpkg: erreur: ne peut pas accéder à l'archive « zabbix-release-latest+debian12_all.deb »: Au
cun fichier ou dossier de ce type
root@debian:~# sudo dpkg -i zabbix-release_latest+debian12_all.deb
Sélection du paquet zabbix-release précédemment désélectionné.
(Lecture de la base de données... 155815 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de zabbix-release_latest+debian12_all.deb ...
Dépaquetage de zabbix-release (1:7.0-2+debian12) ...
Paramétrage de zabbix-release (1:7.0-2+debian12) ...
root@debian:~# sudo apt update
```

4) Installation de Zabbix et de ses dépendances

Nous allons maintenant installer les composants essentiels de Zabbix, à savoir : Le serveur Zabbix (zabbix-server-mysql), L'interface web (zabbix-frontend-php), L'agent Zabbix (installé plus loin).Commencez par exécuter la commande suivante: **sudo apt install zabbix-server-mysql zabbix-frontend-php**

```
user1@debian: ~
apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11 (main/dep11/Components-all.yml) est spécifiée plusieurs fois dans /etc/ap
t/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11-icons-small (main/dep11/icons-48x48.tar) est spécifiée plusieurs fois dan
s /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11-icons (main/dep11/icons-64x64.tar) est spécifiée plusieurs fois dans /etc
/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible Packages (non-free-firmware/binary-amd64/Packages) est spécifiée plusieurs fois
dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible Packages (non-free-firmware/binary-all/Packages) est spécifiée plusieurs fois da
ns /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible Translations (non-free-firmware/i18n/Translation-fr_FR) est spécifiée plusieurs
fois dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible Translations (non-free-firmware/i18n/Translation-fr) est spécifiée plusieurs foi
s dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible Translations (non-free-firmware/i18n/Translation-en) est spécifiée plusieurs foi
s dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11 (non-free-firmware/dep11/Components-amd64.yml) est spécifiée plusieurs fo
is dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11 (non-free-firmware/dep11/Components-all.yml) est spécifiée plusieurs fois
 dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11-icons-small (non-free-firmware/dep11/icons-48x48.tar) est spécifiée plusi
eurs fois dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
N: La cible DEP-11-icons (non-free-firmware/dep11/icons-64x64.tar) est spécifiée plusieurs f
ois dans /etc/apt/sources.list:8 et /etc/apt/sources.list:18
root@debian:~# sudo apt install zabbix-server-mysql zabbix-frontend-php
```

```
user1@debian: ~
url amd64 8.2.28-1~deb12u1 [35,9 kB]
Réception de :15 http://deb.debian.org/debian bookworm/main amd64 php-curl all 2:8.2+93 [3 6
56 B]
Réception de :16 http://deb.debian.org/debian-security bookworm-security/main amd64 php8.2-g
d amd64 8.2.28-1~deb12u1 [28,9 kB]
Réception de :17 http://deb.debian.org/debian bookworm/main amd64 php-gd all 2:8.2+93 [3 648
 B]
Réception de :18 http://deb.debian.org/debian-security bookworm-security/main amd64 php8.2-l
dap amd64 8.2.28-1~deb12u1 [31,4 kB]
Réception de :19 http://deb.debian.org/debian bookworm/main amd64 php-ldap all 2:8.2+93 [3 6
52 B]
Réception de :20 http://deb.debian.org/debian-security bookworm-security/main amd64 php8.2-m
bstring amd64 8.2.28-1~deb12u1 [443 kB]
Réception de :21 http://deb.debian.org/debian bookworm/main amd64 php-mbstring all 2:8.2+93
[3 664 B]
Réception de :22 http://deb.debian.org/debian-security bookworm-security/main amd64 php8.2-m
ysql amd64 8.2.28-1~deb12u1 [117 kB]
Réception de :23 http://deb.debian.org/debian bookworm/main amd64 php-mysql all 2:8.2+93 [3
660 B]
Réception de :24 http://deb.debian.org/debian-security bookworm-security/main amd64 php8.2-x
ml amd64 8.2.28-1~deb12u1 [113 kB]
Réception de :25 http://deb.debian.org/debian bookworm/main amd64 php-xml all 2:8.2+93 [3 67
2 B]
Réception de :26 https://repo.zabbix.com/zabbix/7.0/debian bookworm/main all zabbix-frontend
-php all 1:7.0.10-1+debian12 [8 268 kB]
65% [26 zabbix-frontend-php 2 556 kB/8 268 kB 31%]                    839 kB/s 6s8s
```

Nous complétons ensuite l’installation avec les autres composants nécessaires : **sudo apt install zabbix-apache-conf zabbix-sql-scripts zabbix-agent -y**

```
user1@debian: ~
Paramétrage de php-xml (2:8.2+93) ...
Paramétrage de libevent-extra-2.1-7:amd64 (2.1.12-stable-8) ...
Paramétrage de php-curl (2:8.2+93) ...
Paramétrage de php-mysql (2:8.2+93) ...
Paramétrage de zabbix-server-mysql (1:7.0.10-1+debian12) ...
Paramétrage de php-bcmath (2:8.2+93) ...
Paramétrage de zabbix-frontend-php (1:7.0.10-1+debian12) ...
update-alternatives: utilisation de « /usr/share/fonts/truetype/dejavu/DejaVuSans.ttf » pour
 fournir « /usr/share/zabbix/assets/fonts/graphfont.ttf » (zabbix-frontend-font) en mode aut
omatique
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@debian:~# zabbix-apache-conf zabbix-sql-scripts zabbix-agent -y
-bash: zabbix-apache-conf : commande introuvable
root@debian:~# ^C
root@debian:~# sudo apt install zabbix-apache-conf zabbix-sql-scripts zabbix-agent -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  apache2 apache2-data apache2-utils libapache2-mod-php libapache2-mod-php8.2 libmodbus5
  php8.2-cli php8.2-opcache php8.2-readline
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-data apache2-utils libapache2-mod-php libapache2-mod-php8.2 libmodbus5
```

5) Configuration de la base de données pour Zabbix

Nous allons maintenant configurer la base de données qui sera utilisée par Zabbix pour stocker toutes ses informations (éléments supervisés, historiques, alertes...).Commençons par nous connecter à MariaDB en tant que superutilisateur : **sudo mysql -u root -p**


```
user1@debian: ~
Paramétrage de php-xml (2:8.2+93) ...
Paramétrage de libevent-extra-2.1-7:amd64 (2.1.12-stable-8) ...
Paramétrage de php-curl (2:8.2+93) ...
Paramétrage de php-mysql (2:8.2+93) ...
Paramétrage de zabbix-server-mysql (1:7.0.10-1+debian12) ...
Paramétrage de php-bcmath (2:8.2+93) ...
Paramétrage de zabbix-frontend-php (1:7.0.10-1+debian12) ...
update-alternatives: utilisation de « /usr/share/fonts/truetype/dejavu/DejaVuSans.ttf » pour
 fournir « /usr/share/zabbix/assets/fonts/graphfont.ttf » (zabbix-frontend-font) en mode aut
omatique
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@debian:~# zabbix-apache-conf zabbix-sql-scripts zabbix-agent -y
-bash: zabbix-apache-conf : commande introuvable
root@debian:~# ^C
root@debian:~# sudo apt install zabbix-apache-conf zabbix-sql-scripts zabbix-agent -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  apache2 apache2-data apache2-utils libapache2-mod-php libapache2-mod-php8.2 libmodbus5
  php8.2-cli php8.2-opcache php8.2-readline
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-data apache2-utils libapache2-mod-php libapache2-mod-php8.2 libmodbus5
```


Une fois connectés à MariaDB, nous créons une base de données dédiée à Zabbix, puis nous définissons un utilisateur spécifique avec les bons droits.Saisissons les commandes

suivantes dans le terminal MariaDB :**CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;**

```
user1@debian: ~
MariaDB [(none)]> exit
Bye
root@debian:~# sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
    -> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123'
    -> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
    -> SET GLOBAL log_bin_trust_function_creators = 1
    -> exit
    -> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'CREATE USER 'zabbix'@'local
host' IDENTIFIED BY 'Aurel123'
GRANT ALL PRIVILEGE...' at line 2
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]>
```

Ensuite, nous créons un utilisateur nommé zabbix avec un mot de passe sécurisé (ici Aurel123), qui sera utilisé par le serveur Zabbix pour accéder à la base de données : **CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';**

```
user1@debian: ~
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
    -> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123'
    -> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
    -> SET GLOBAL log_bin_trust_function_creators = 1
    -> exit
    -> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'CREATE USER 'zabbix'@'local
host' IDENTIFIED BY 'Aurel123'
GRANT ALL PRIVILEGE...' at line 2
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]>
```

Nous attribuons ensuite tous les privilèges sur la base de données zabbix à l’utilisateur que nous venons de créer : **GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';**

```
user1@debian: ~
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
    -> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123'
    -> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
    -> SET GLOBAL log_bin_trust_function_creators = 1
    -> exit
    -> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'CREATE USER 'zabbix'@'local
host' IDENTIFIED BY 'Aurel123'
GRANT ALL PRIVILEGE...' at line 2
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]>
```

Certaines fonctions utilisées par Zabbix nécessitent des permissions spécifiques. Nous activons donc temporairement une option au niveau global pour permettre leur création : **SET GLOBAL log_bin_trust_function_creators = 1;**

```
user1@debian: ~
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
    -> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123'
    -> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
    -> SET GLOBAL log_bin_trust_function_creators = 1
    -> exit
    -> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'CREATE USER 'zabbix'@'local
host' IDENTIFIED BY 'Aurel123'
GRANT ALL PRIVILEGE...' at line 2
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> SET GLOBAL log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]>
```

Enfin, nous quittons la session MariaDB avec la commande suivante : **EXIT;**

```
user1@debian: ~

MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
    -> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123'
    -> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
    -> SET GLOBAL log_bin_trust_function_creators = 1
    -> exit
    -> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'CREATE USER 'zabbix'@'local
host' IDENTIFIED BY 'Aurel123'
GRANT ALL PRIVILEGE...' at line 2
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> SET GLOBAL log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> exit;
Bye
root@debian:~#
```

6) Import du schéma initial de la base Zabbix

Nous allons maintenant importer le schéma de base de données nécessaire au bon fonctionnement de Zabbix.Cette opération initialise toutes les tables et données système indispensables. Exécutez la commande suivante (un mot de passe vous sera demandé pour l'utilisateur zabbix) : **zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -u zabbix -p zabbix**

```
user1@debian: ~

host' IDENTIFIED BY 'Aurel123'
GRANT ALL PRIVILEGE...' at line 2
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> SET GLOBAL log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> exit;
Bye
root@debian:~# ^[[200~zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbi
x -p zabbix~
-bash: $'\E[200~zcat' : commande introuvable
Enter password:
ERROR 1045 (28000): Access denied for user 'zabbix'@'localhost' (using password: NO)
root@debian:~#
root@debian:~# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p za
bbix
Enter password:
root@debian:~#
```

Une fois l'import terminé, nous désactivons l'option log_bin_trust_function_creators pour des raisons de sécurité : **sudo mysql -u root -p -e "SET GLOBAL log_bin_trust_function_creators = 0;"**

```
user1@debian: ~
MariaDB [(none)]> CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Aurel123';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> SET GLOBAL log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> exit;
Bye
root@debian:~# ^[[200~zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbi
x -p zabbix~
-bash: $'\E[200~zcat' : commande introuvable
Enter password:
ERROR 1045 (28000): Access denied for user 'zabbix'@'localhost' (using password: NO)
root@debian:~#
root@debian:~# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p za
bbix
Enter password:
root@debian:~# sudo mysql -u root -p -e "SET GLOBAL log_bin_trust_function_creators = 0;"
Enter password:
root@debian:~#
```

7) Configuration du serveur Zabbix

Nous allons maintenant modifier le fichier de configuration du serveur Zabbix afin d'y renseigner les informations d'accès à la base de données MariaDB.Ouvrons le fichier avec un éditeur de texte : sudo nano /etc/zabbix/zabbix_server.confDans ce fichier, localisez les lignes suivantes et modifiez-les comme indiqué : **DBPassword=Aurel123 EnableGlobalScripts=1**

```
user1@debian: ~
 GNU nano 7.2                    /etc/zabbix/zabbix_server.conf *
#
# Mandatory: no
# Default:
# DBSchema=

### Option: DBUser
#       Database user.
#
# Mandatory: no
# Default:
# DBUser=

DBUser=zabbix

### Option: DBPassword
#       Database password.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=Aurel123
EnableGlobalScripts=1

^G Aide        ^O Écrire      ^W Chercher    ^K Couper      ^T Exécuter    ^C Emplacement
^X Quitter     ^R Lire fich.  ^\ Remplacer   ^U Coller      ^J Justifier   ^/ Aller ligne
```

8) Configuration d'Apache et de PHP

Pour garantir le bon fonctionnement de l'interface web de Zabbix, nous devons configurer correctement PHP, qui est exécuté via Apache.Ouvrons le fichier de configuration PHP de Zabbix : **sudo nano /etc/zabbix/apache.conf**, activez le module PHP 8.2 : **sudo a2enmod php8.2**

```
user1@debian: ~

MariaDB [(none)]> exit;
Bye
root@debian:~# ^[[200~zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbi
x -p zabbix~
-bash: $'\E[200~zcat' : commande introuvable
Enter password:
ERROR 1045 (28000): Access denied for user 'zabbix'@'localhost' (using password: NO)
root@debian:~#
root@debian:~# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p za
bbix
Enter password:
root@debian:~# sudo mysql -u root -p -e "SET GLOBAL log_bin_trust_function_creators = 0;"
Enter password:
root@debian:~# sudo nano /etc/zabbix/zabbix_server.conf
root@debian:~# sudo nano /etc/zabbix/apache.conf
root@debian:~# sudo nano /etc/php/*/apache2/php.ini
root@debian:~# sudo nano /etc/zabbix/apache.conf
root@debian:~# sudo a2enmod php8.2
Considering dependency mpm_prefork for php8.2:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php8.2:
Module php8.2 already enabled
root@debian:~#
```

Redémarrons Apache pour appliquer les modifications, puis activons-le pour qu'il se lance automatiquement au démarrage : **sudo systemctl restart apache2** et **sudo systemctl enable apache2**

```
user1@debian: ~

Enter password:
ERROR 1045 (28000): Access denied for user 'zabbix'@'localhost' (using password: NO)
root@debian:~#
root@debian:~# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p za
bbix
Enter password:
root@debian:~# sudo mysql -u root -p -e "SET GLOBAL log_bin_trust_function_creators = 0;"
Enter password:
root@debian:~# sudo nano /etc/zabbix/zabbix_server.conf
root@debian:~# sudo nano /etc/zabbix/apache.conf
root@debian:~# sudo nano /etc/php/*/apache2/php.ini
root@debian:~# sudo nano /etc/zabbix/apache.conf
root@debian:~# sudo a2enmod php8.2
Considering dependency mpm_prefork for php8.2:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php8.2:
Module php8.2 already enabled
root@debian:~# sudo systemctl restart apache2
root@debian:~# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sy
sv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@debian:~#
```

9) Configurer le Pare-feu

Pour sécuriser notre serveur, configurez le pare-feu pour n'autoriser que les ports nécessaires à Zabbix.Ouvrez les ports pour les communications de Zabbix :
Configuration du pare-feu afin de sécuriser l'accès au serveur, nous allons configurer le pare-feu pour autoriser uniquement les ports nécessaires au fonctionnement de Zabbix. Nous commençons par installer l'outil **iptables** : **sudo apt install -y iptables**

```
user1@debian: ~
Considering dependency mpm_prefork for php8.2:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php8.2:
Module php8.2 already enabled
root@debian:~# sudo systemctl restart apache2
root@debian:~# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sy
sv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@debian:~# sudo apt install -y iptables
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libip6tc2
Paquets suggérés :
  firewalld
Les NOUVEAUX paquets suivants seront installés :
  iptables libip6tc2
0 mis à jour, 2 nouvellement installés, 0 à enlever et 158 non mis à jour.
Il est nécessaire de prendre 380 ko dans les archives.
Après cette opération, 2 533 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 libip6tc2 amd64 1.8.9-2 [19
```

Après l'installation d'iptables, nous autorisons les connexions entrantes nécessaires pour le fonctionnement de Zabbix.Ouvrons d'abord le port utilisé par le serveur Zabbix (port 10051): **sudo iptables -A INPUT -p tcp --dport 10051 -j ACCEPT**

```
user1@debian: ~
(Lecture de la base de données... 159054 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../libip6tc2_1.8.9-2_amd64.deb ...
Dépaquetage de libip6tc2:amd64 (1.8.9-2) ...
Sélection du paquet iptables précédemment désélectionné.
Préparation du dépaquetage de .../iptables_1.8.9-2_amd64.deb ...
Dépaquetage de iptables (1.8.9-2) ...
Paramétrage de libip6tc2:amd64 (1.8.9-2) ...
Paramétrage de iptables (1.8.9-2) ...
update-alternatives: utilisation de « /usr/sbin/iptables-legacy » pour fournir « /usr/sbin/i
ptables » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-legacy » pour fournir « /usr/sbin/
ip6tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/iptables-nft » pour fournir « /usr/sbin/ipta
bles » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-nft » pour fournir « /usr/sbin/ip6
tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/arptables-nft » pour fournir « /usr/sbin/arp
tables » (arptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ebtables-nft » pour fournir « /usr/sbin/ebta
bles » (ebtables) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10051 -j ACCEPT
root@debian:~#
```

Ensuite, nous ouvrons également le port utilisé par les agents Zabbix pour communiquer avec le serveur (port 10050) : **sudo iptables -A INPUT -p tcp --dport 10050 -j ACCEPT**

```
user1@debian: ~
(Lecture de la base de données... 159054 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../libip6tc2_1.8.9-2_amd64.deb ...
Dépaquetage de libip6tc2:amd64 (1.8.9-2) ...
Sélection du paquet iptables précédemment désélectionné.
Préparation du dépaquetage de .../iptables_1.8.9-2_amd64.deb ...
Dépaquetage de iptables (1.8.9-2) ...
Paramétrage de libip6tc2:amd64 (1.8.9-2) ...
Paramétrage de iptables (1.8.9-2) ...
update-alternatives: utilisation de « /usr/sbin/iptables-legacy » pour fournir « /usr/sbin/i
ptables » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-legacy » pour fournir « /usr/sbin/
ip6tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/iptables-nft » pour fournir « /usr/sbin/ipta
bles » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-nft » pour fournir « /usr/sbin/ip6
tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/arptables-nft » pour fournir « /usr/sbin/arp
tables » (arptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ebtables-nft » pour fournir « /usr/sbin/ebta
bles » (ebtables) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10051 -j ACCEPT
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10050 -j ACCEPT
root@debian:~#
```

Après avoir configuré les règles du pare-feu, nous créons un dossier pour sauvegarder la configuration d'iptables afin qu'elle soit restaurée automatiquement au prochain redémarrage: **sudo mkdir -p /etc/iptables** ensuite, nous sauvegardons les règles actuelles dans un fichier afin de pouvoir les restaurer automatiquement : **sudo sh -c "iptables-save > /etc/iptables/rules.v4"**

```
user1@debian: ~
Dépaquetage de libip6tc2:amd64 (1.8.9-2) ...
Sélection du paquet iptables précédemment désélectionné.
Préparation du dépaquetage de .../iptables_1.8.9-2_amd64.deb ...
Dépaquetage de iptables (1.8.9-2) ...
Paramétrage de libip6tc2:amd64 (1.8.9-2) ...
Paramétrage de iptables (1.8.9-2) ...
update-alternatives: utilisation de « /usr/sbin/iptables-legacy » pour fournir « /usr/sbin/i
ptables » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-legacy » pour fournir « /usr/sbin/
ip6tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/iptables-nft » pour fournir « /usr/sbin/ipta
bles » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-nft » pour fournir « /usr/sbin/ip6
tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/arptables-nft » pour fournir « /usr/sbin/arp
tables » (arptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ebtables-nft » pour fournir « /usr/sbin/ebta
bles » (ebtables) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10051 -j ACCEPT
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10050 -j ACCEPT
root@debian:~# mkdir -p /etc/iptables
root@debian:~# sudo sh -c "iptables-save > /etc/iptables/rules.v4"
root@debian:~#
```

Puis, pour restaurer manuellement les règles si besoin : **sudo iptables-restore < /etc/iptables/rules.v4**

```
user1@debian: ~
Sélection du paquet iptables précédemment désélectionné.
Préparation du dépaquetage de .../iptables_1.8.9-2_amd64.deb ...
Dépaquetage de iptables (1.8.9-2) ...
Paramétrage de libip6tc2:amd64 (1.8.9-2) ...
Paramétrage de iptables (1.8.9-2) ...
update-alternatives: utilisation de « /usr/sbin/iptables-legacy » pour fournir « /usr/sbin/i
ptables » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-legacy » pour fournir « /usr/sbin/
ip6tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/iptables-nft » pour fournir « /usr/sbin/ipta
bles » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-nft » pour fournir « /usr/sbin/ip6
tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/arptables-nft » pour fournir « /usr/sbin/arp
tables » (arptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ebtables-nft » pour fournir « /usr/sbin/ebta
bles » (ebtables) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10051 -j ACCEPT
root@debian:~# sudo iptables -A INPUT -p tcp --dport 10050 -j ACCEPT
root@debian:~# mkdir -p /etc/iptables
root@debian:~# sudo sh -c "iptables-save > /etc/iptables/rules.v4"
root@debian:~# sudo iptables-restore < /etc/iptables/rules.v4
root@debian:~#
```

10) Démarrage et activation des services Zabbix

Une fois tous les composants installés et configurés, nous procédons au démarrage des services afin de rendre Zabbix pleinement opérationnel. Nous allons également les activer pour qu'ils se lancent automatiquement au démarrage du serveur.Exécutez les commandes suivantes : **sudo systemctl restart zabbix-server zabbix-agent apache2** et **sudo systemctl enable zabbix-server zabbix-agent apache2**

```
user1@debian: ~
user1@debian:~$ su -
Mot de passe :
root@debian:~# sudo systemctl restart zabbix-server zabbix-agent apache2
root@debian:~# sudo systemctl enable zabbix-server zabbix-agent apache2
Synchronizing state of zabbix-server.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-server
Synchronizing state of zabbix-agent.service with SysV service script with /lib/s
ystemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
Synchronizing state of apache2.service with SysV service script with /lib/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.servic
e → /lib/systemd/system/zabbix-server.service.
root@debian:~#
```

11) Configuration initiale via l'interface web

Après avoir sélectionné la langue **"Français"**, nous arrivons sur les étapes de configuration de la base de données. Nous remplissons les champs suivants :

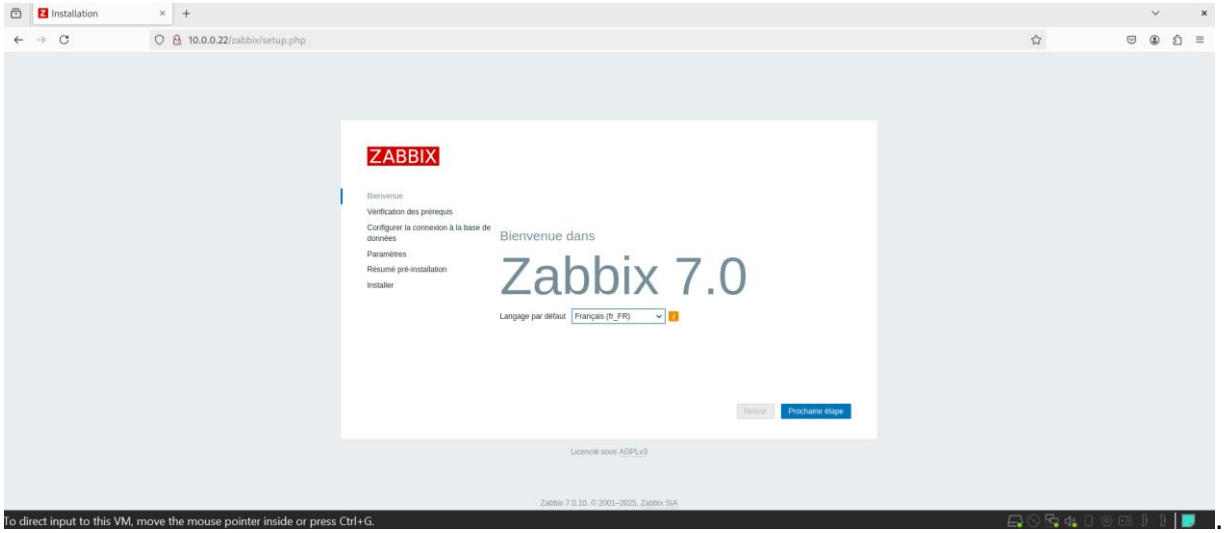
Nom de la base de données : zabbix

Utilisateur : zabbix

Mot de passe : Aurel123

Adresse du serveur de base de données : localhost (par défaut)*

Important : Lors de la migration vers un **serveur physique**, l'adresse IP du serveur Zabbix changera et deviendra :10.102.152.5



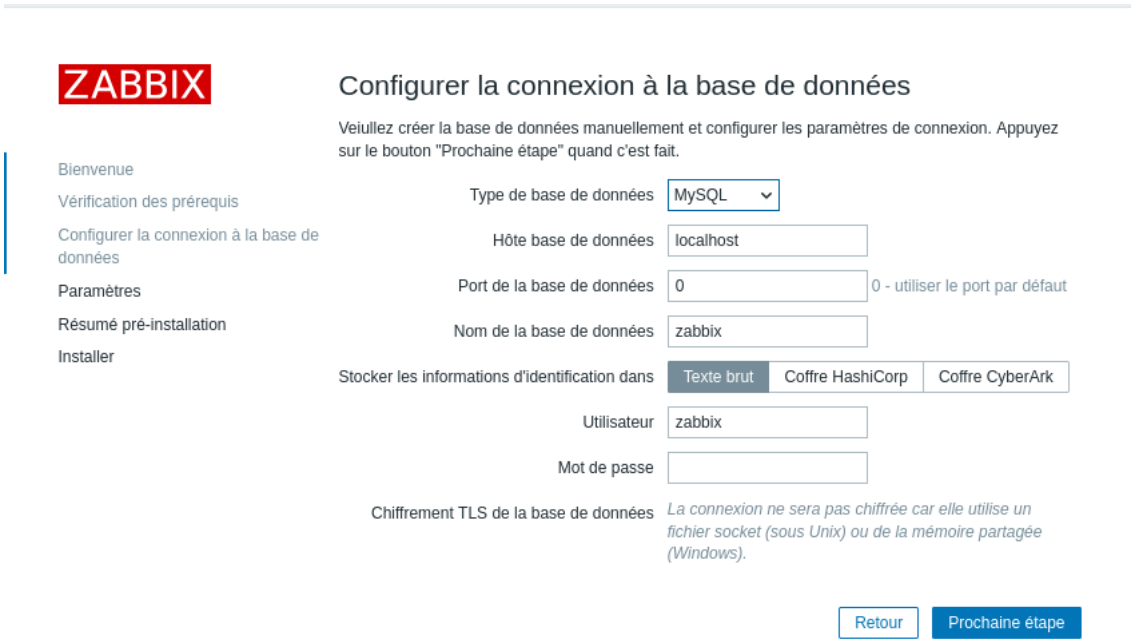
À cette étape, nous vérifions que tous les prérequis sont affichés en **"OK"**.

Nous poursuivons en renseignant les informations suivantes pour la connexion à la base de données :

Nom de la base de données : zabbix

Utilisateur : zabbix

Mot de passe : Aurel123

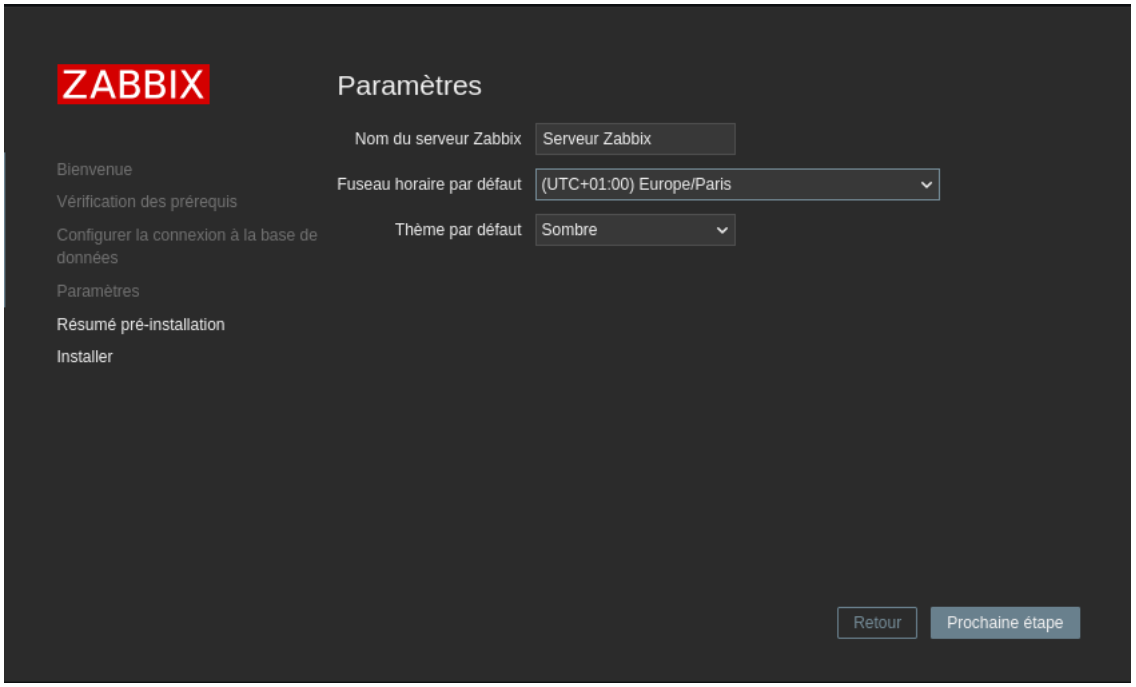


Ensuite, nous allons nommer notre serveur. Nous choisirons ici :

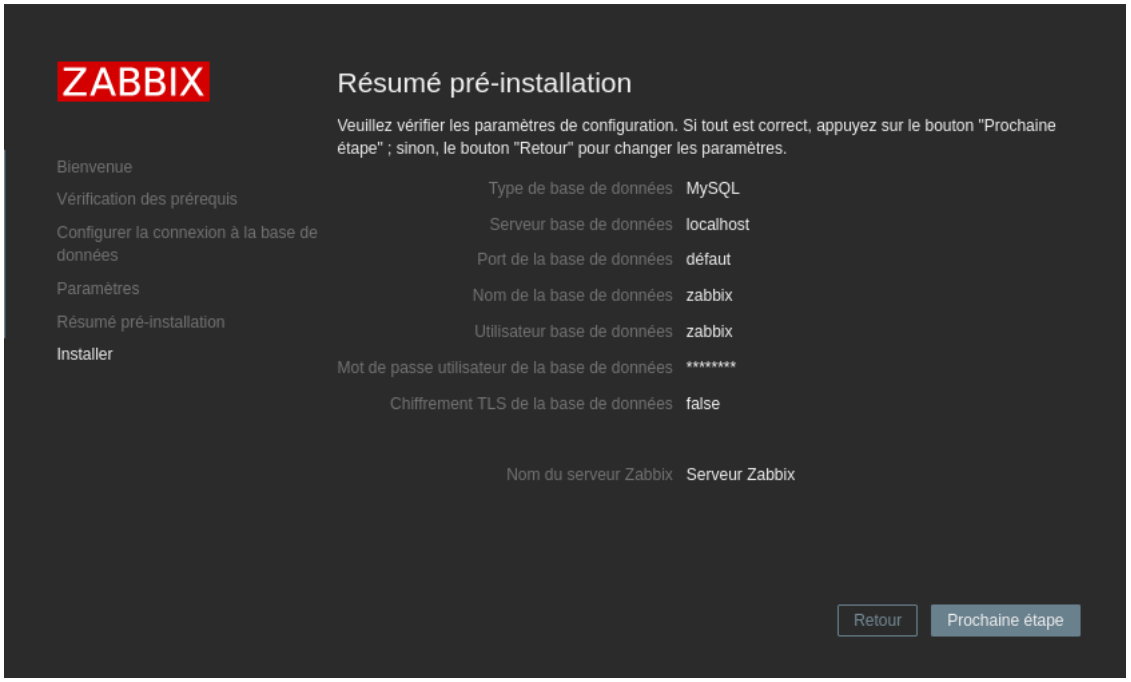
Nom du serveur : Serveur Zabbix

Fuseau horaire : Europe/Paris

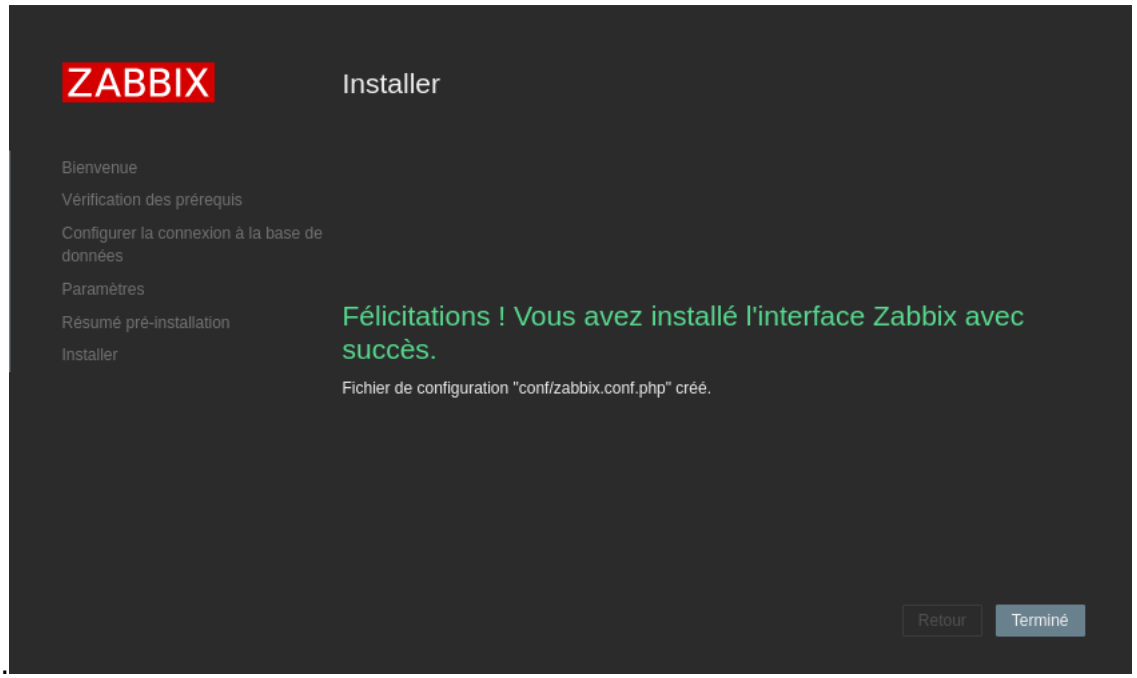
Ces paramètres permettent d'identifier le serveur dans l'interface Zabbix et d'assurer une bonne synchronisation des horaires des événements.



Après avoir renseigné tous les paramètres, un résumé de la configuration s'affiche.
Nous vérifions que toutes les informations sont correctes avant de finaliser l'installation.



Enfin, un **dernier message de confirmation** nous indique que **l'installation de Zabbix a été réalisée avec succès**



Après avoir finalisé la configuration, nous nous connectons à l'interface de Zabbix en utilisant les identifiants par défaut :

Nom d'utilisateur : Admin

Mot de passe : zabbix

Remarque :

Si nous avons créé nos propres identifiants lors de l'installation ou si nous les avons modifiés, la connexion fonctionnera de la même manière avec ces nouveaux identifiants.

Il est fortement recommandé de personnaliser les accès pour renforcer la sécurité du serveur.

III - Installation et configuration d'un serveur Windows 2022 (DHCP, AD DS, DNS)

Dans cette partie, nous allons installer et configurer un serveur Windows 2022 afin qu'il assure plusieurs rôles essentiels au sein de notre infrastructure :

Serveur DHCP : attribution automatique des adresses IP aux machines du réseau,

Contrôleur de domaine Active Directory (AD DS) : gestion centralisée des utilisateurs, groupes, stratégies et machines,

Serveur DNS : résolution des noms de domaine internes.

Pour mettre en place notre serveur Windows 2022, nous installons directement les **rôles nécessaires** via le **Gestionnaire de serveur**.

Les étapes principales sont :

Ouvrir le **Gestionnaire de serveur** après l'installation de Windows.

Cliquer sur "Ajouter des rôles et fonctionnalités".

Sélectionner les rôles suivants :

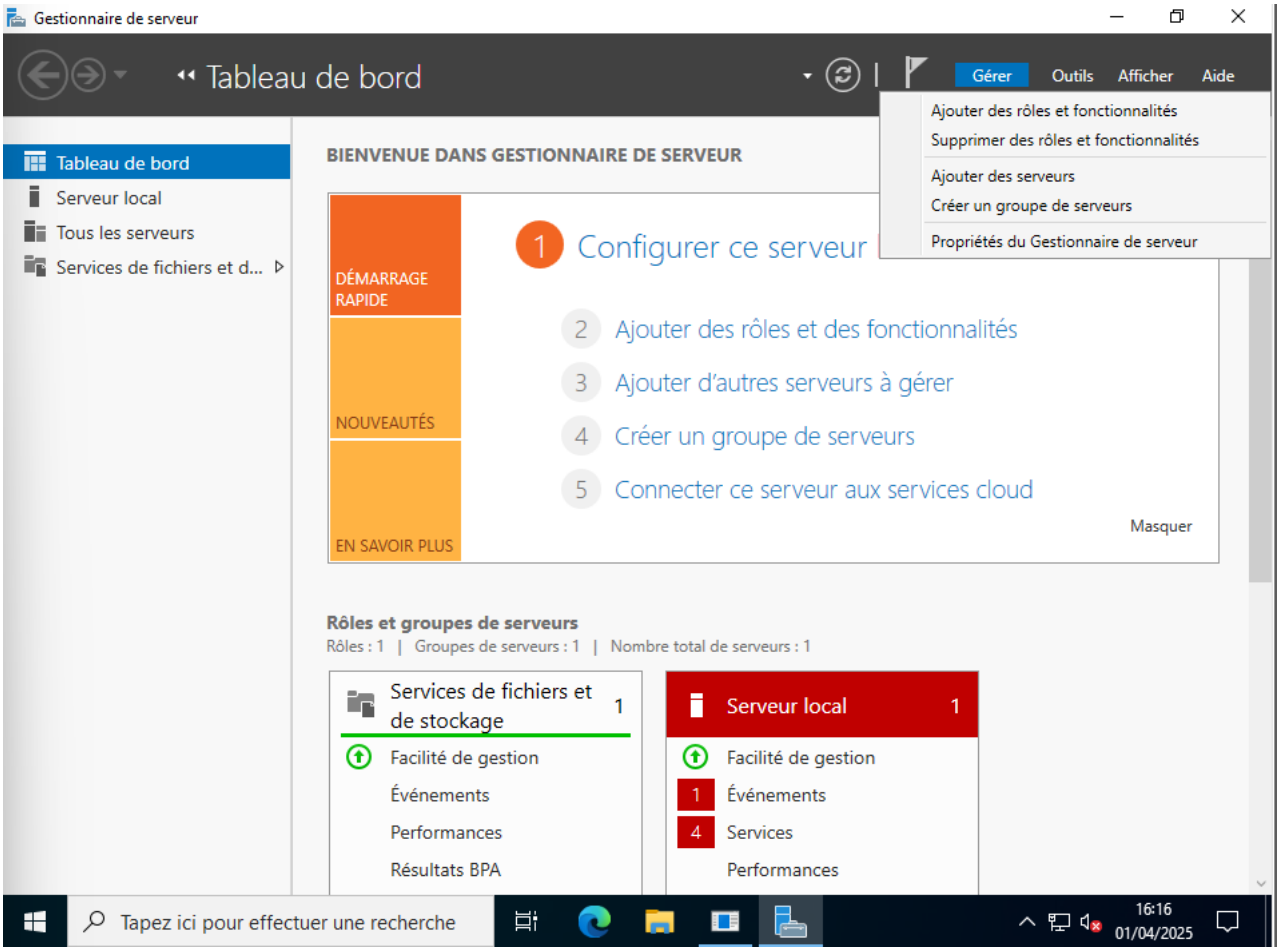
Serveur DHCP pour l'attribution dynamique des adresses IP,

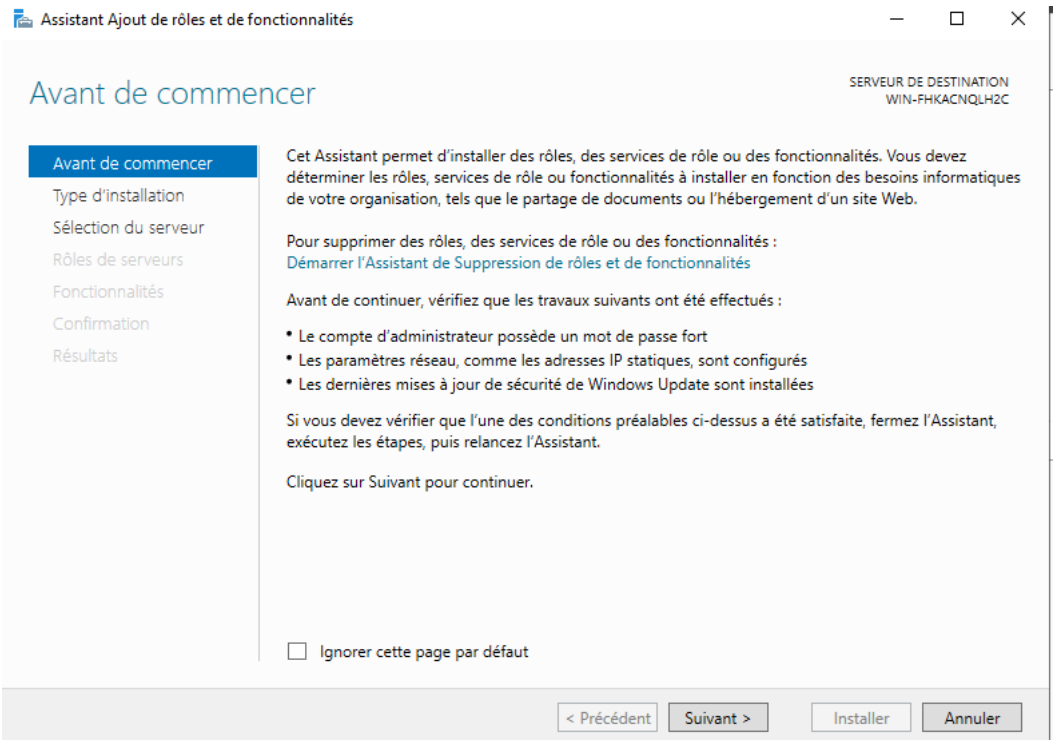
Active Directory Domain Services (AD DS) pour créer un contrôleur de domaine,

Serveur DNS pour gérer la résolution des noms internes.

Suivre l'assistant en laissant les paramètres par défaut si cela est proposé.

Valider et installer les rôles.


Gestionnaire de serveur
Tableau de bord
Gérer
Outils
Afficher
Aide
Ajouter des rôles et fonctionnalités
Supprimer des rôles et fonctionnalités
Ajouter des serveurs
Créer un groupe de serveurs
Propriétés du Gestionnaire de serveur
Tableau de bord
Serveur local
Tous les serveurs
Services de fichiers et d...
BIENVENUE DANS GESTIONNAIRE DE SERVEUR
DÉMARRAGE RAPIDE
NOUVEAUTÉS
EN SAVOIR PLUS
1 Configurer ce serveur
2 Ajouter des rôles et des fonctionnalités
3 Ajouter d'autres serveurs à gérer
4 Créer un groupe de serveurs
5 Connecter ce serveur aux services cloud
Masquer
Rôles et groupes de serveurs
Rôles : 1 | Groupes de serveurs : 1 | Nombre total de serveurs : 1
Services de fichiers et de stockage 1
Facilité de gestion
Événements
Performances
Résultats BPA
Serveur local 1
Facilité de gestion
1 Événements
4 Services
Performances
Tapez ici pour effectuer une recherche
16:16
01/04/2025


Assistant Ajout de rôles et de fonctionnalités
Avant de commencer
SERVEUR DE DESTINATION
WIN-FHKACNQLH2C
Avant de commencer
Type d'installation
Sélection du serveur
Rôles de serveurs
Fonctionnalités
Confirmation
Résultats
Cet Assistant permet d'installer des rôles, des services de rôle ou des fonctionnalités. Vous devez déterminer les rôles, services de rôle ou fonctionnalités à installer en fonction des besoins informatiques de votre organisation, tels que le partage de documents ou l'hébergement d'un site Web.
Pour supprimer des rôles, des services de rôle ou des fonctionnalités :
Démarrer l'Assistant de Suppression de rôles et de fonctionnalités
Avant de continuer, vérifiez que les travaux suivants ont été effectués :
• Le compte d'administrateur possède un mot de passe fort
• Les paramètres réseau, comme les adresses IP statiques, sont configurés
• Les dernières mises à jour de sécurité de Windows Update sont installées
Si vous devez vérifier que l'une des conditions préalables ci-dessus a été satisfaite, fermez l'Assistant, exécutez les étapes, puis relancez l'Assistant.
Cliquez sur Suivant pour continuer.
Ignorer cette page par défaut
< Précédent
Suivant >
Installer
Annuler

Assistant Ajout de rôles et de fonctionnalités

Sélectionner le type d'installation

SERVEUR DE DESTINATION
WIN-FHKACNQLH2C

- Avant de commencer
- Type d'installation
- Sélection du serveur
- Rôles de serveurs
- Fonctionnalités
- Confirmation
- Résultats

Sélectionnez le type d'installation. Vous pouvez installer des rôles et des fonctionnalités sur un ordinateur physique ou virtuel en fonctionnement, ou sur un disque dur virtuel hors connexion.

(●) **Installation basée sur un rôle ou une fonctionnalité**
Configurez un serveur unique en ajoutant des rôles, des services de rôle et des fonctionnalités.

( ) **Installation des services Bureau à distance**
Installez les services de rôle nécessaires à l'infrastructure VDI (Virtual Desktop Infrastructure) pour déployer des bureaux basés sur des ordinateurs virtuels ou sur des sessions.

< Précédent | Suivant > | Installer | Annuler

Assistant Ajout de rôles et de fonctionnalités

Sélectionner le serveur de destination

SERVEUR DE DESTINATION
WIN-FHKACNQLH2C

- Avant de commencer
- Type d'installation
- Sélection du serveur
- Rôles de serveurs
- Fonctionnalités
- Confirmation
- Résultats

Sélectionnez le serveur ou le disque dur virtuel sur lequel installer des rôles et des fonctionnalités.

(●) Sélectionner un serveur du pool de serveurs
( ) Sélectionner un disque dur virtuel

Pool de serveurs

Filtre :

Nom	Adresse IP	Système d'exploitation
WIN-FHKACNQLH2C	192.168.200.140	Microsoft Windows Server 2022 Datacenter

1 ordinateur(s) trouvé(s)

Cette page présente les serveurs qui exécutent Windows Server 2012 ou une version ultérieure et qui ont été ajoutés à l'aide de la commande Ajouter des serveurs dans le Gestionnaire de serveur. Les serveurs hors connexion et les serveurs nouvellement ajoutés dont la collecte de données est toujours incomplète ne sont pas répertoriés.

< Précédent | Suivant > | Installer | Annuler

Assistant Ajout de rôles et de fonctionnalités

Sélectionner des rôles de serveurs

SERVEUR DE DESTINATION
WIN-FHKACNQLH2C

- Avant de commencer
- Type d'installation
- Sélection du serveur
- Rôles de serveurs
- Fonctionnalités
- AD DS
- Serveur DNS
- Confirmation
- Résultats

Sélectionnez un ou plusieurs rôles à installer sur le serveur sélectionné.

Rôles

- [ ] Accès à distance
- [ ] Attestation d'intégrité de l'appareil
- [ ] Contrôleur de réseau
- [ ] Hyper-V
- [ ] Serveur de télécopie
- [x] Serveur DHCP (Installé)
- [x] Serveur DNS
- [ ] Serveur Web (IIS)
- [ ] Service Guardian hôte
- [x] Services AD DS
- [ ] Services AD LDS (Active Directory Lightweight Dire
- [ ] Services AD RMS (Active Directory Rights Manager
- [ ] Services Bureau à distance
- [ ] Services d'activation en volume
- [ ] Services d'impression et de numérisation de docu
- [ ] Services de certificats Active Directory
- [ ] Services de fédération Active Directory (AD FS)
- [■] Services de fichiers et de stockage (1 sur 12 install
- [ ] Services de stratégie et d'accès réseau

Description

Le serveur DNS (Domain Name System) permet la résolution de noms sur les réseaux TCP/IP. Le serveur DNS est plus facile à gérer lorsqu'il est installé sur le même serveur que les services de domaine Active Directory. Si vous sélectionnez le rôle Services de domaine Active Directory, vous pouvez installer et configurer le serveur DNS et les services de domaine Active Directory pour les faire fonctionner conjointement.

< Précédent | Suivant > | Installer | Annuler

Assistant Ajout de rôles et de fonctionnalités

Services de domaine Active Directory

SERVEUR DE DESTINATION
WIN-FHKACNQLH2C

- Avant de commencer
- Type d'installation
- Sélection du serveur
- Rôles de serveurs
- Fonctionnalités
- AD DS
- Serveur DNS
- Confirmation
- Résultats

Les services de domaine Active Directory (AD DS) stockent des informations sur les utilisateurs, les ordinateurs et les périphériques sur le réseau. Les services AD DS permettent aux administrateurs de gérer ces informations de façon sécurisée et facilitent le partage des ressources et la collaboration entre les utilisateurs.

À noter :

- Pour veiller à ce que les utilisateurs puissent quand même se connecter au réseau en cas de panne de serveur, installez un minimum de deux contrôleurs de domaine par domaine.
- Les services AD DS nécessitent qu'un serveur DNS soit installé sur le réseau. Si aucun serveur DNS n'est installé, vous serez invité à installer le rôle de serveur DNS sur cet ordinateur.

Azure Active Directory, un service en ligne distinct, peut fournir une gestion simplifiée des identités et des accès, des rapports de sécurité et une authentification unique aux applications web dans le cloud et sur site.

En savoir plus sur Azure Active Directory

Configurer Office 365 avec Azure Active Directory Connect

< Précédent | Suivant > | Installer | Annuler

Assistant Ajout de rôles et de fonctionnalités

Serveur DNS

SERVEUR DE DESTINATION
WIN-FHKACNQLH2C

- Avant de commencer
- Type d'installation
- Sélection du serveur
- Rôles de serveurs
- Fonctionnalités
- AD DS
- **Serveur DNS**
- Confirmation
- Résultats

Le système DNS (Domain Name System) fournit une méthode standard d'association de noms à des adresses Internet numériques. Cela permet aux utilisateurs de référencer les ordinateurs du réseau en utilisant des noms faciles à retenir au lieu de longues séries de chiffres. En outre, le système DNS intègre un espace de noms hiérarchique, ce qui permet que chaque nom d'hôte soit unique sur un réseau local ou étendu. Les services DNS Windows peuvent être intégrés aux services DHCP (Dynamic Host Configuration Protocol) sur Windows. Il n'est ainsi plus nécessaire d'ajouter des enregistrements DNS lorsque des ordinateurs sont ajoutés au réseau.

Éléments à noter :

- L'intégration du serveur DNS aux services de domaine Active Directory réplique les données DNS et d'autres données du service d'annuaire, ce qui facilite la gestion DNS.
- Les services de domaine Active Directory nécessitent l'installation d'un serveur DNS sur le réseau. Si vous installez un contrôleur de domaine, vous pouvez aussi installer le rôle serveur DNS avec l'Assistant Installation des services de domaine Active Directory, en sélectionnant le rôle Services de domaine Active Directory.

< Précédent | Suivant > | Installer | Annuler

Assistant Ajout de rôles et de fonctionnalités

Confirmer les sélections d'installation

SERVEUR DE DESTINATION
WIN-FHKACNQLH2C

- Avant de commencer
- Type d'installation
- Sélection du serveur
- Rôles de serveurs
- Fonctionnalités
- AD DS
- Serveur DNS
- **Confirmation**
- Résultats

Pour installer les rôles, services de rôle ou fonctionnalités suivants sur le serveur sélectionné, cliquez sur Installer.

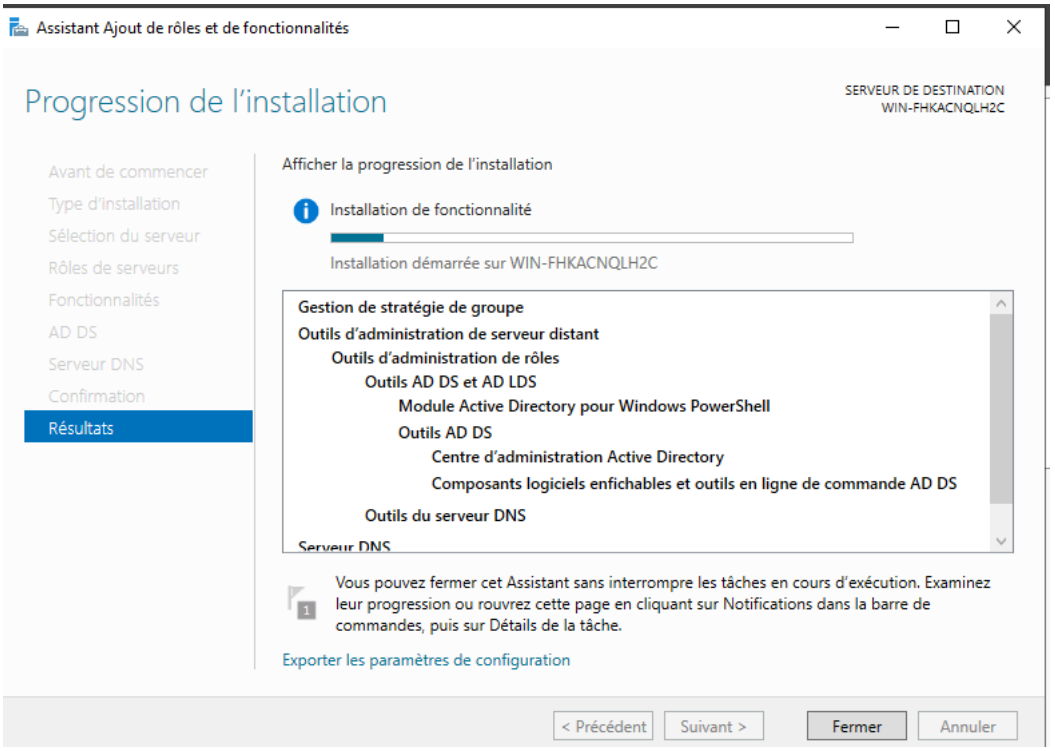
☐ Redémarrer automatiquement le serveur de destination, si nécessaire

Il se peut que des fonctionnalités facultatives (comme des outils d'administration) soient affichées sur cette page, car elles ont été sélectionnées automatiquement. Si vous ne voulez pas installer ces fonctionnalités facultatives, cliquez sur Précédent pour désactiver leurs cases à cocher.

Gestion de stratégie de groupe
Outils d'administration de serveur distant
  Outils d'administration de rôles
    Outils AD DS et AD LDS
      Module Active Directory pour Windows PowerShell
      Outils AD DS
        Centre d'administration Active Directory
        Composants logiciels enfichables et outils en ligne de commande AD DS
    Outils du serveur DNS

Exporter les paramètres de configuration
Spécifier un autre chemin d'accès source

< Précédent | Suivant > | Installer | Annuler

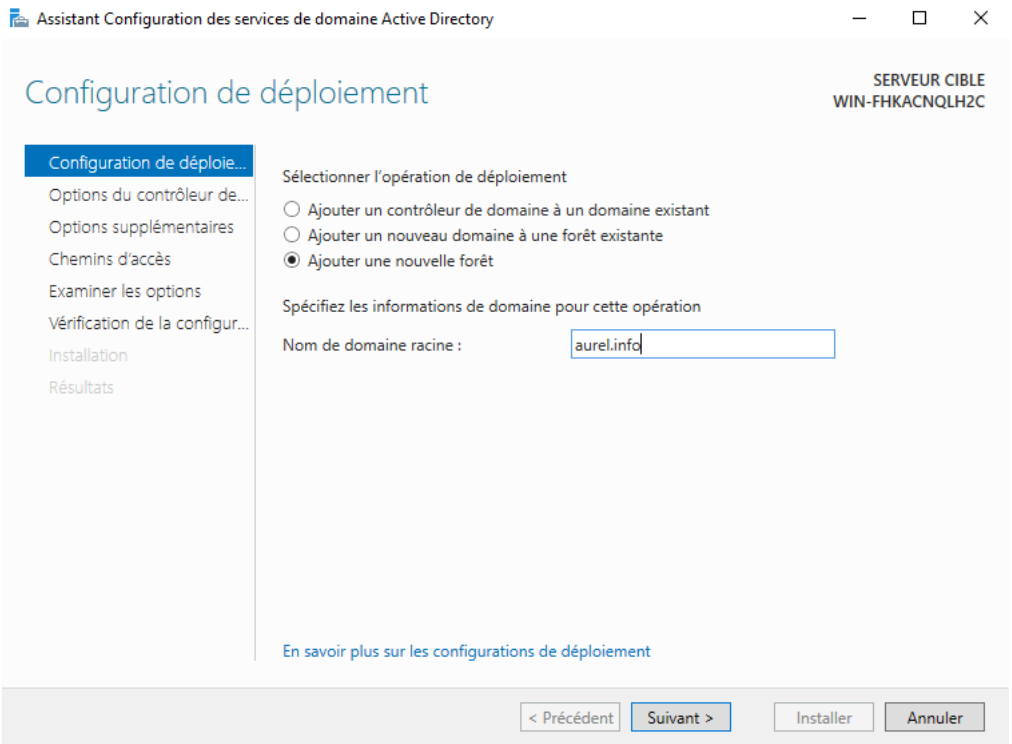


Après l'installation des rôles, nous procédons à la **promotion du serveur en tant que contrôleur de domaine**.

Dans l'assistant d'AD DS :

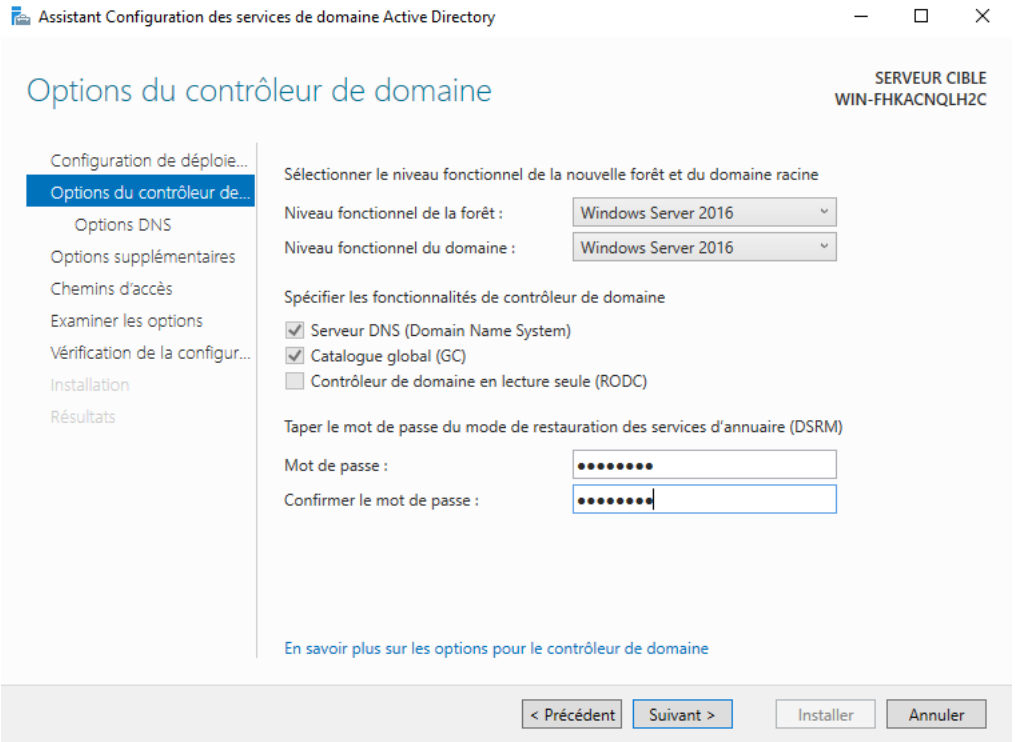
Nous cliquons sur **"Ajouter une nouvelle forêt"**,

Nous renseignons le **nom de domaine racine** suivant : **aurelinfo.local**

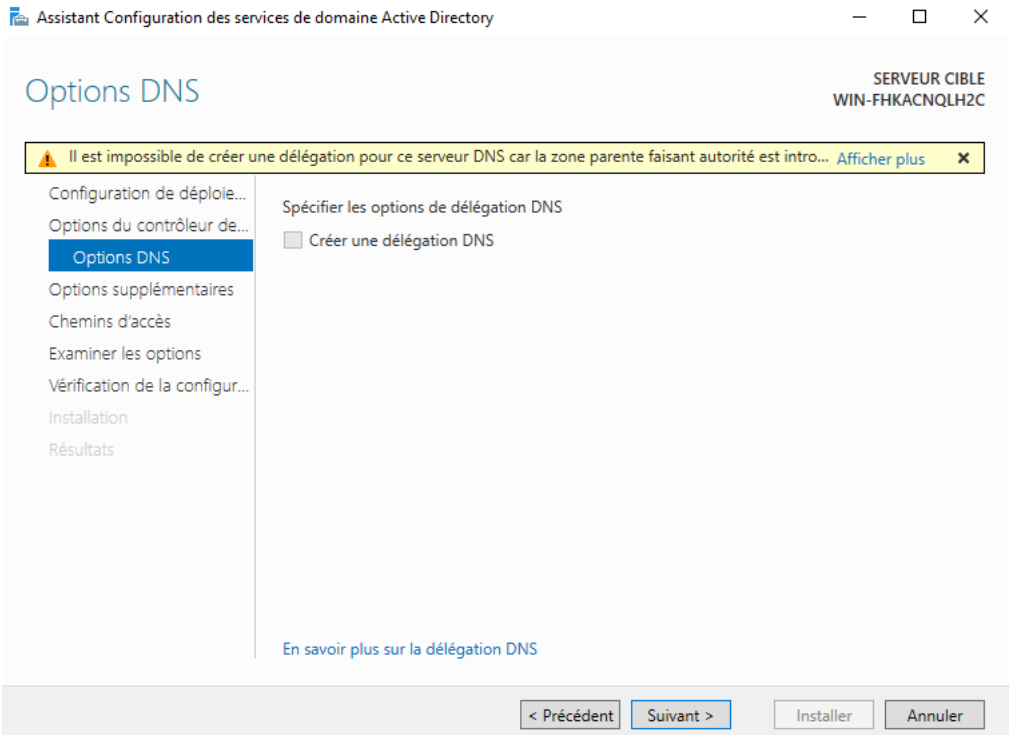


Nous définissons ensuite un **mot de passe du mode de restauration des services d'annuaire (DSRM)**, nécessaire pour les opérations de récupération d'Active Directory. Nous utilisons ici le mot de passe suivant : **Aurel1234**

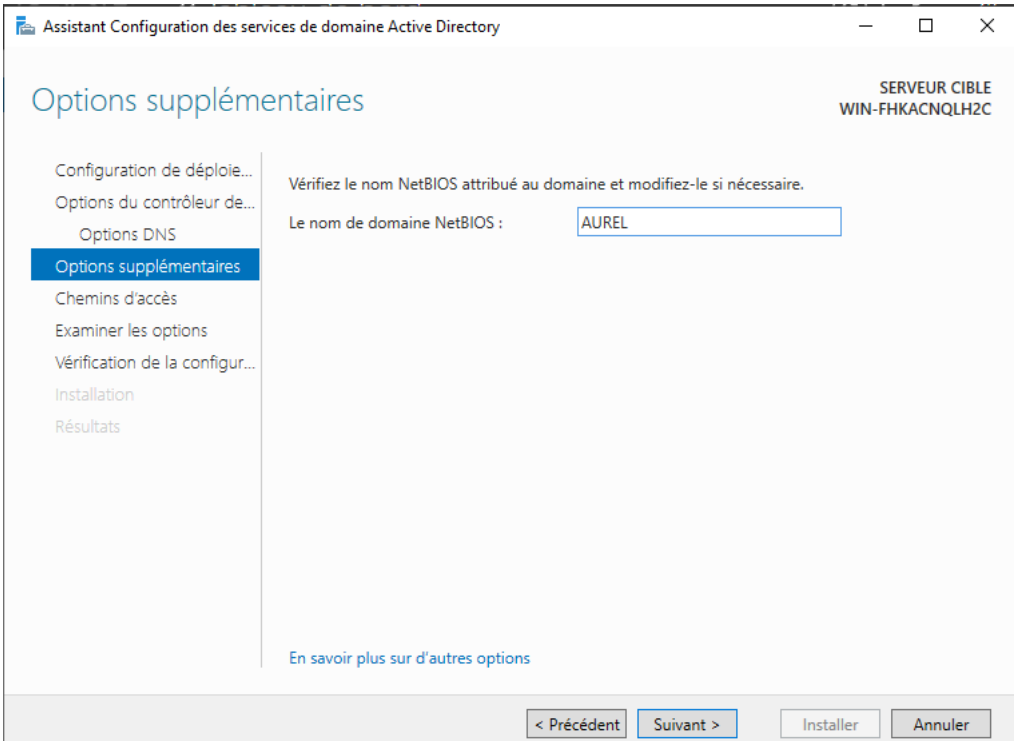
Puis, nous cliquons sur **Suivant** pour continuer la configuration.



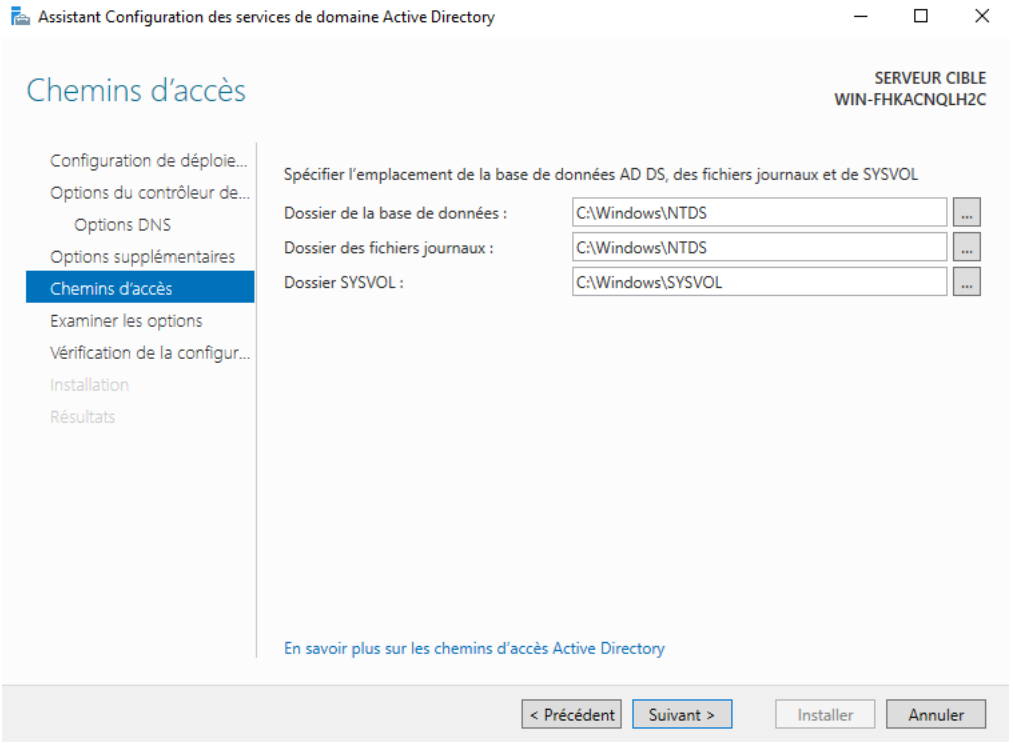
Lorsque nous arrivons à l'option **DNS** dans l'assistant de configuration, nous laissons les paramètres par défaut et cliquons simplement sur **Suivant**.



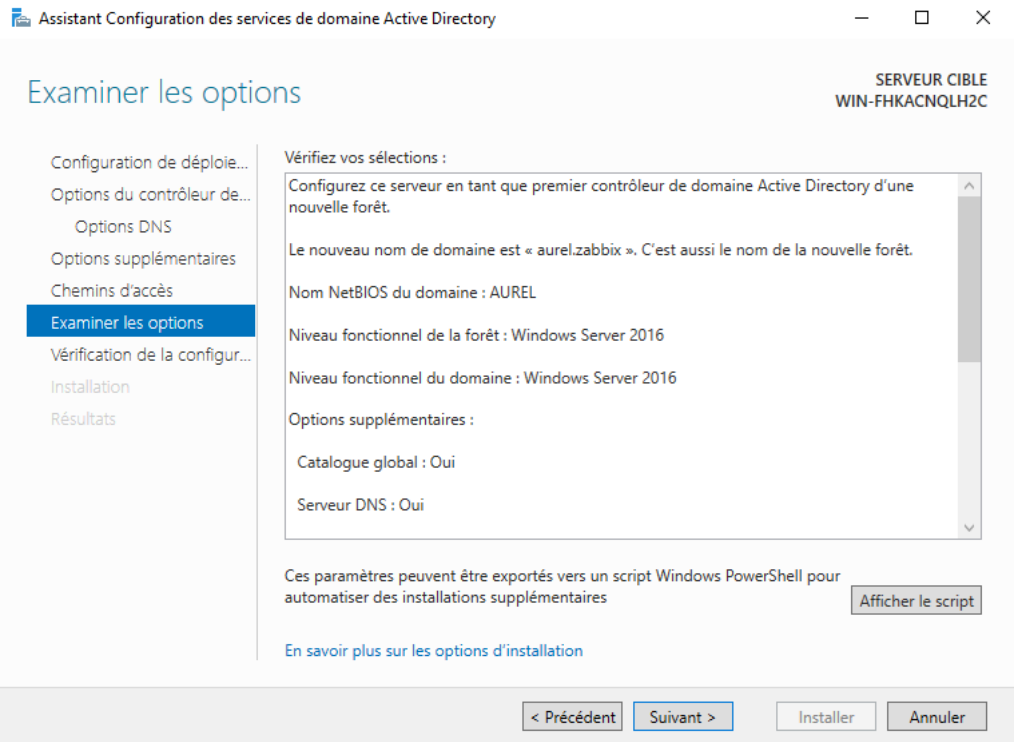
Concernant le **nom NetBIOS** du domaine, nous laissons la valeur proposée par défaut, qui est : **AUREL**



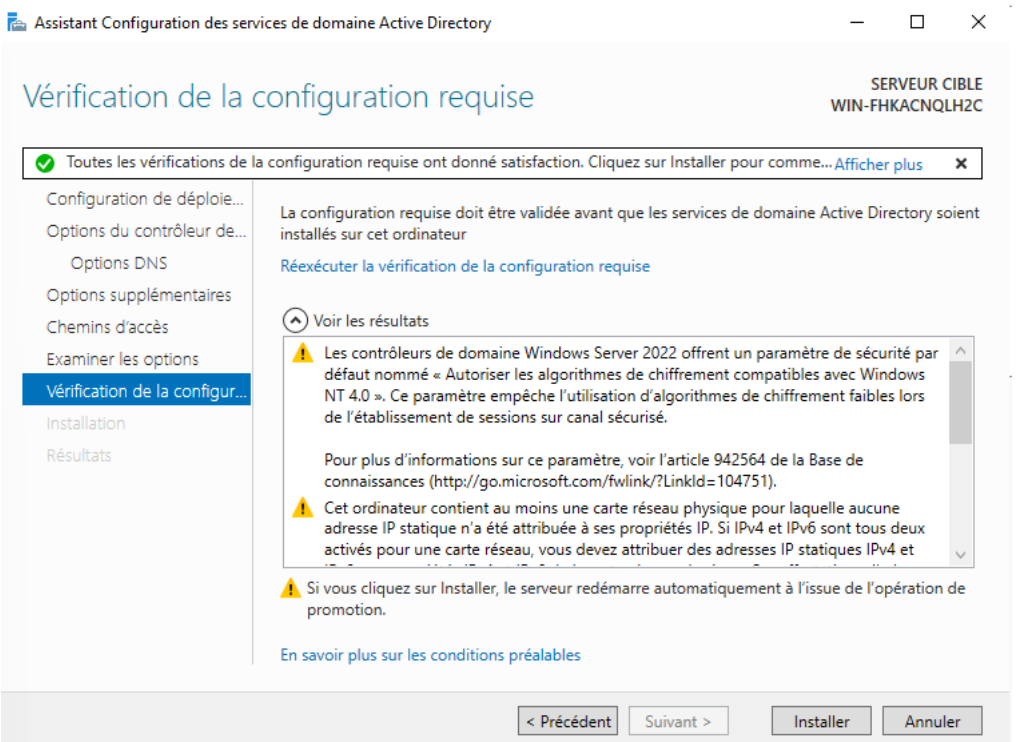
Au niveau du **chemin d'accès** pour la base de données, les fichiers journaux et le dossier SYSVOL, nous laissons les chemins par défaut proposés par l'assistant, puis nous cliquons sur **Suivant**.

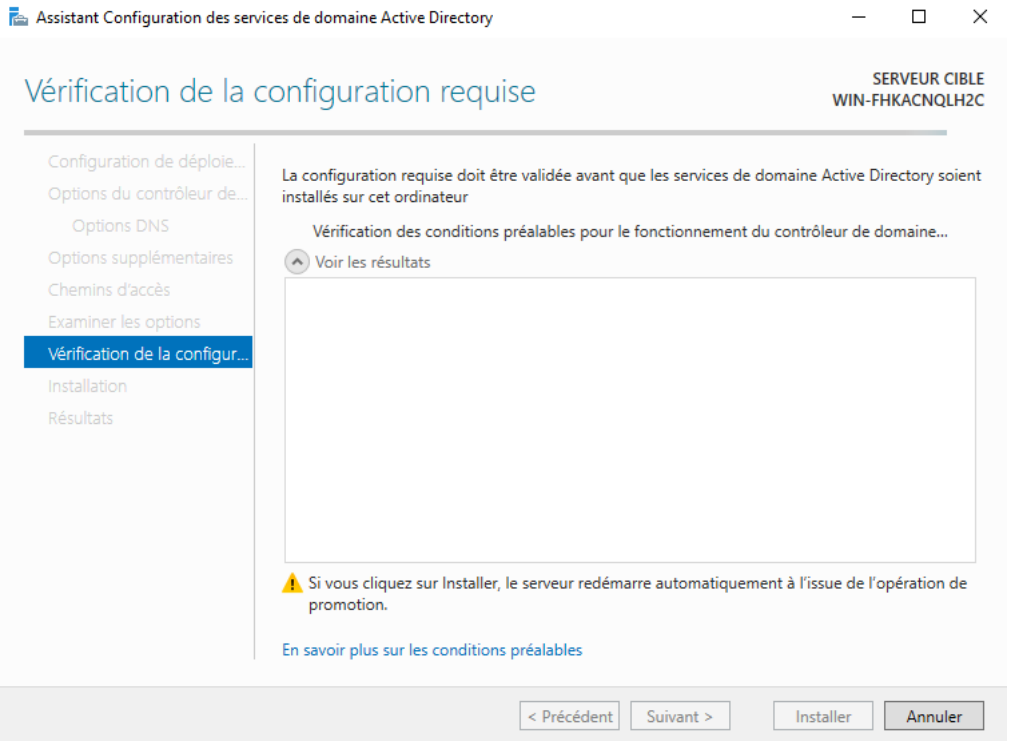


À l'étape **Examiner les options**, nous vérifions que toutes les informations sont correctes, puis nous cliquons sur **Suivant** pour continuer.

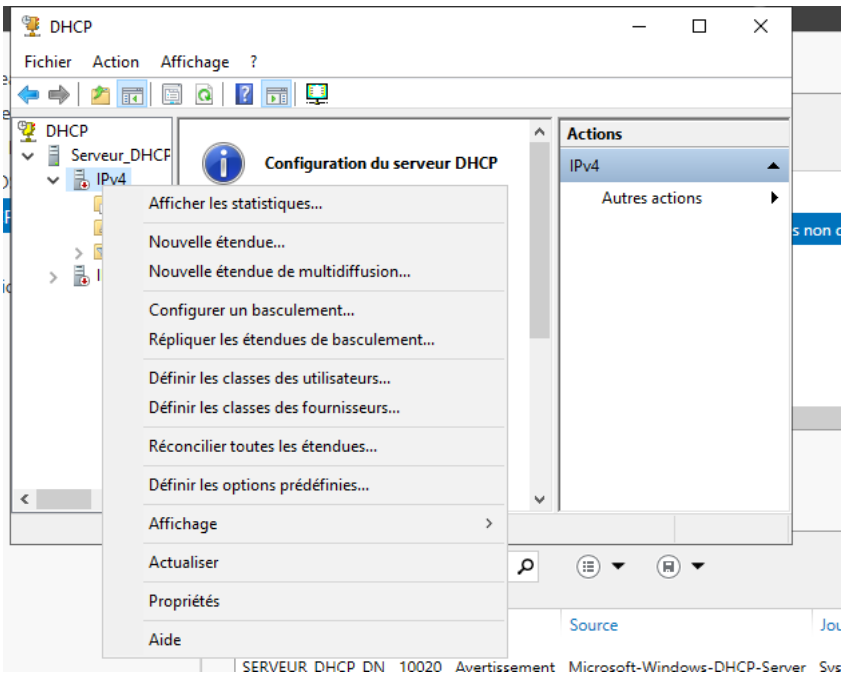


Après la vérification des prérequis, nous cliquons sur **Installer** pour lancer la configuration du serveur en tant que contrôleur de domaine.

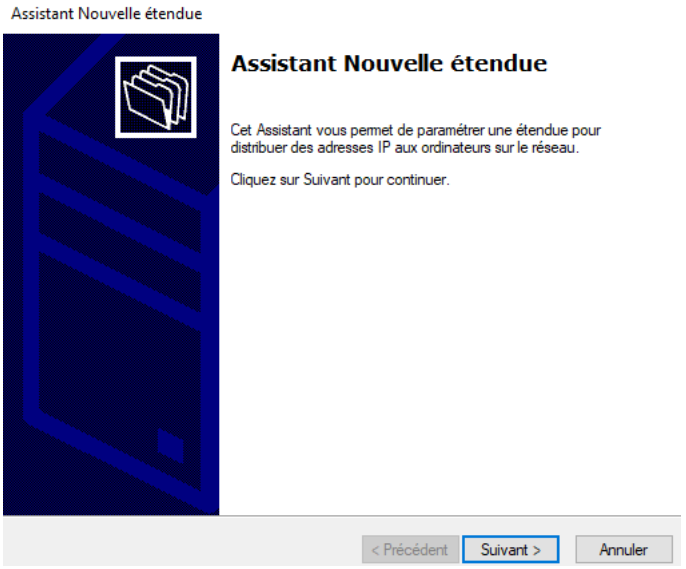




Après la vérification des prérequis, nous cliquons sur **Installer** pour lancer la promotion du serveur en tant que **contrôleur de domaine**.



Après le redémarrage et la connexion au serveur, nous lançons la configuration du rôle **DHCP**.
Dans l'assistant de configuration, nous cliquons sur **Suivant** pour **créer une nouvelle étendue**, qui permettra de définir la plage d'adresses IP que le serveur attribuera automatiquement aux clients du réseau.

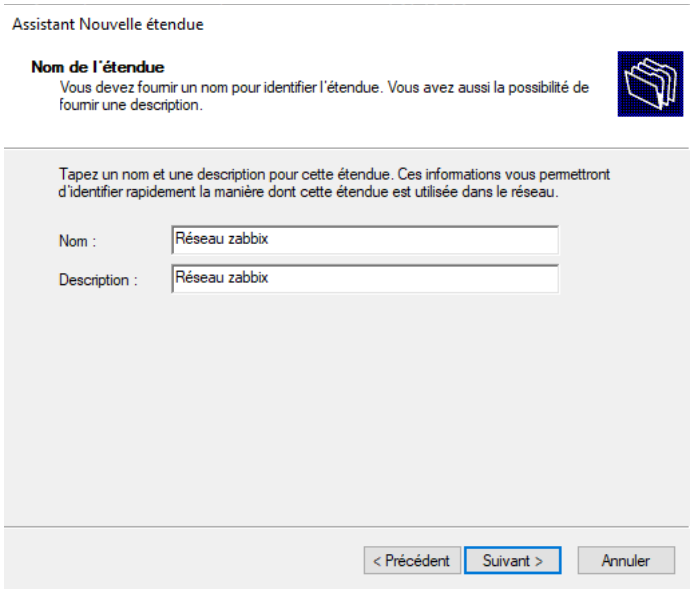


Nous nommons notre étendue :

Nom : Réseau Zabbix

Description : Réseau Zabbix

Puis, nous cliquons sur **Suivant** pour continuer la configuration.

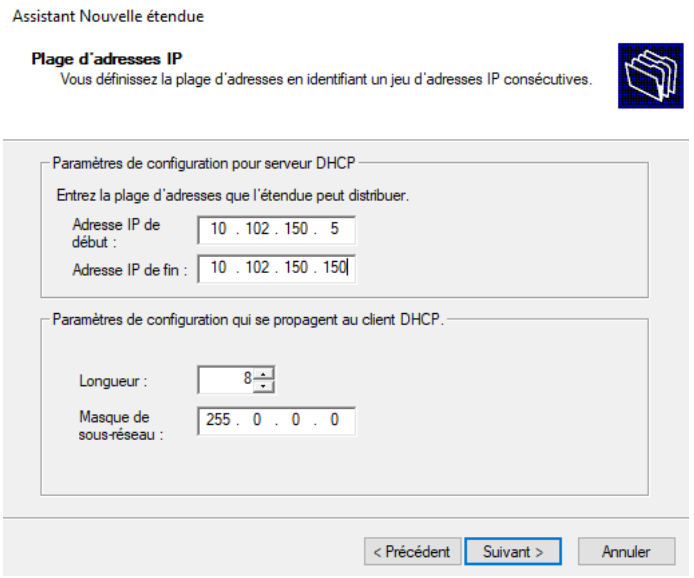


Ensuite, nous définissons la **plage d'adresses IP** qui sera attribuée automatiquement aux clients par le serveur DHCP :

Adresse IP de début : 10.102.152.5

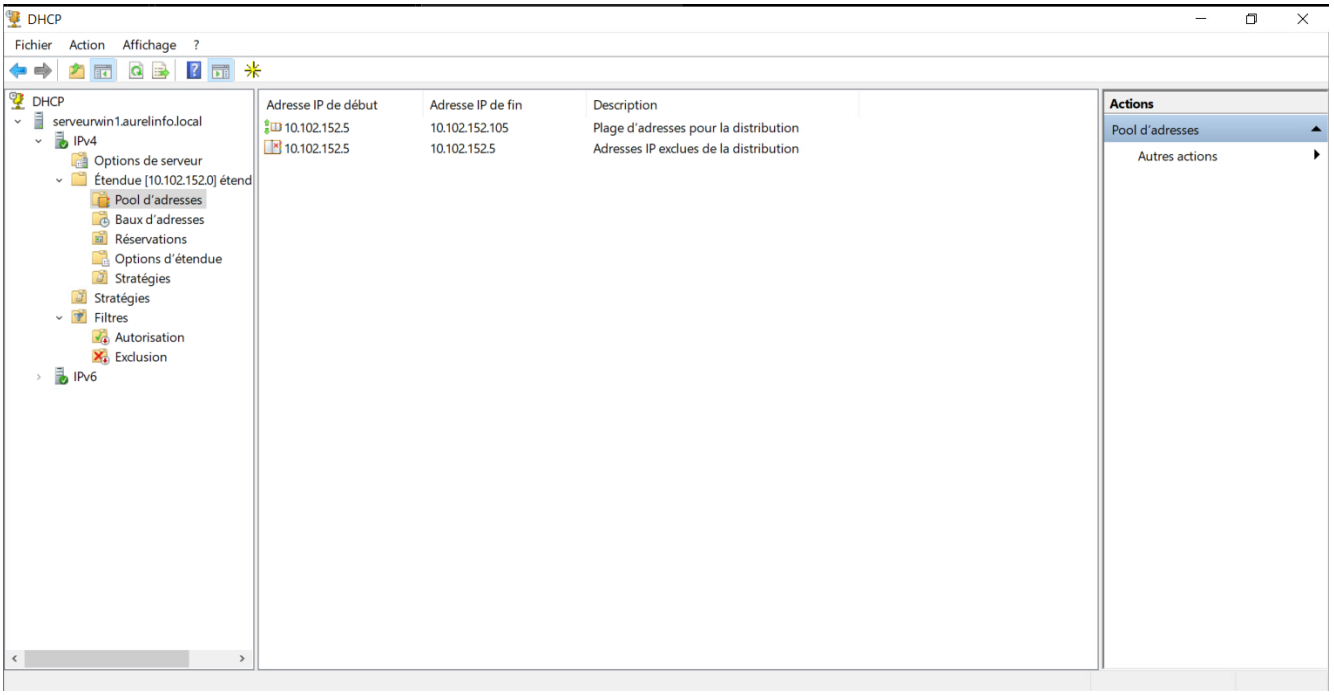
Adresse IP de fin : 10.102.152.105

Nous cliquons ensuite sur **Suivant** pour continuer.

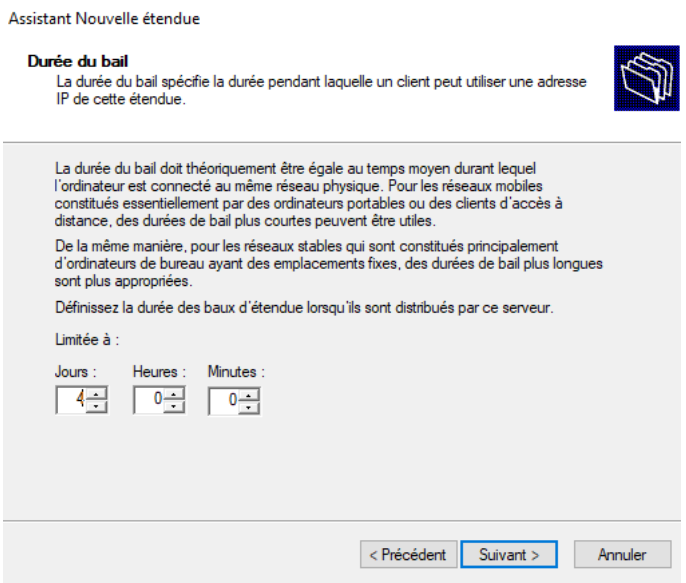


À l'étape **Exclusions et délai d'attribution**, nous allons **exclure l'adresse IP 10.102.152.5**. Cette adresse sera réservée à notre **machine Debian** hébergeant **Zabbix**, afin de garantir que son IP ne soit jamais attribuée dynamiquement par le serveur DHCP.

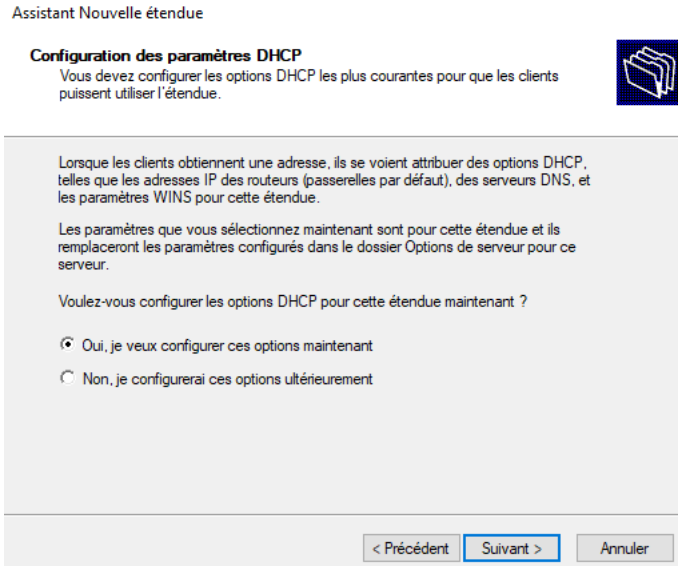
Après avoir ajouté cette exclusion, nous cliquons sur **Suivant** pour poursuivre.



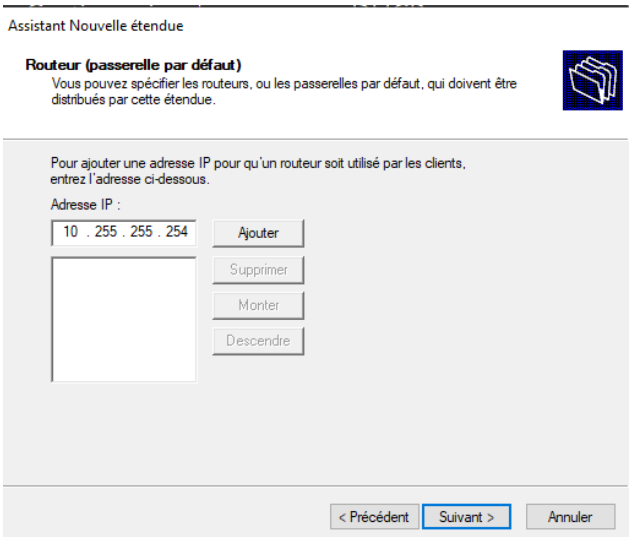
À l'étape **Durée du bail**, nous définissons la durée pendant laquelle une adresse IP est attribuée à un client.
Nous configurons ici une **durée de bail de 4 jours**, puis nous cliquons sur **Suivant**.



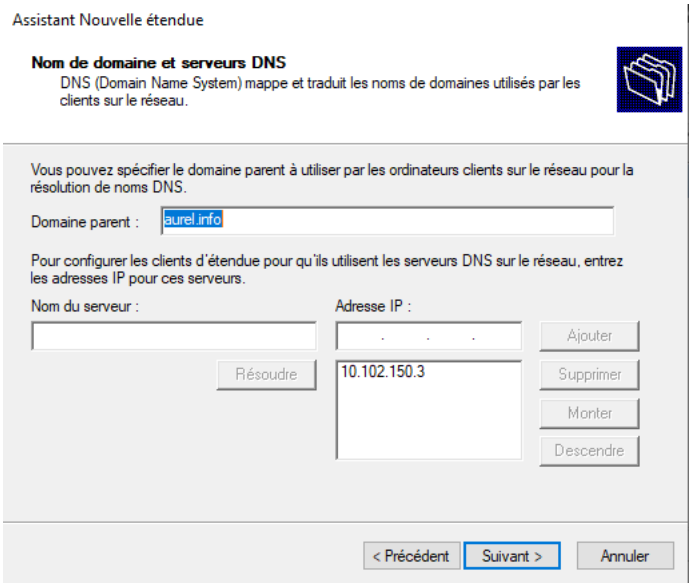
À l'étape suivante, nous répondons **Oui** lorsqu'il nous est demandé si nous souhaitons **configurer les options DHCP** maintenant (passerelle, DNS, etc.).



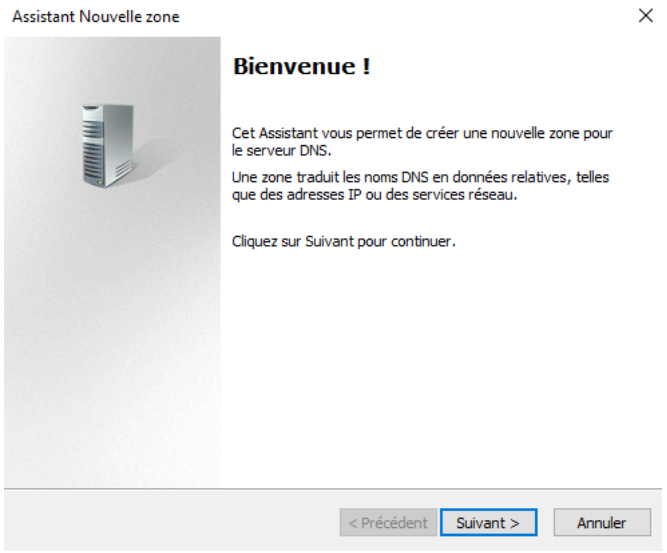
Nous poursuivons en configurant la **passerelle par défaut**. Nous renseignons ici l'adresse IP du routeur qui permet la sortie vers Internet pour notre réseau interne.



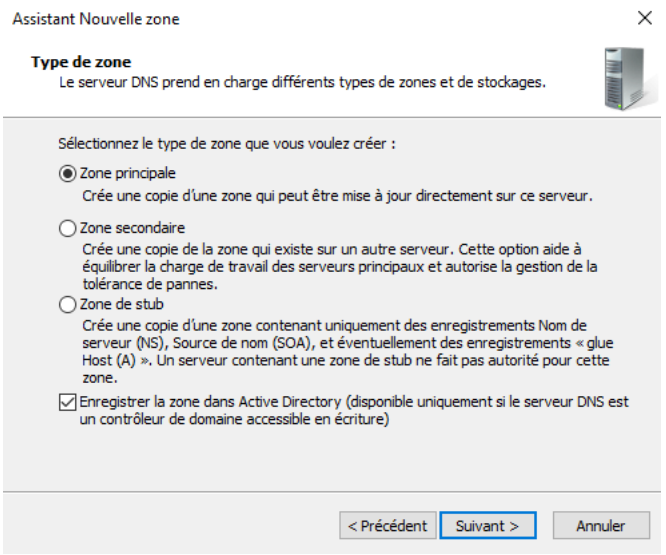
Après avoir renseigné la passerelle par défaut, nous cliquons sur **Suivant** pour continuer la configuration.



Nous configurons ensuite les **serveurs DNS** dans l'assistant DHCP.
Nous pouvons laisser les paramètres proposés ou renseigner l'adresse de notre **serveur DNS interne** (celui du contrôleur de domaine).Puis nous cliquons sur **Suivant** pour poursuivre.



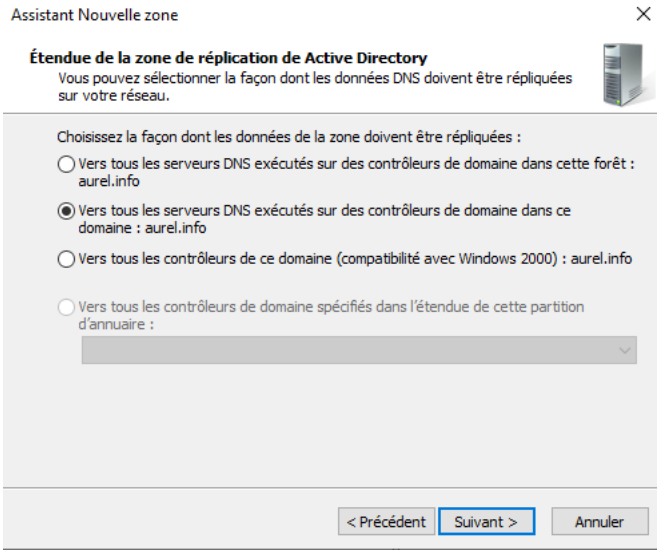
À l'étape suivante, nous sélectionnons la **zone principale** pour le service DHCP, puis nous cliquons sur **Suivant** pour continuer.



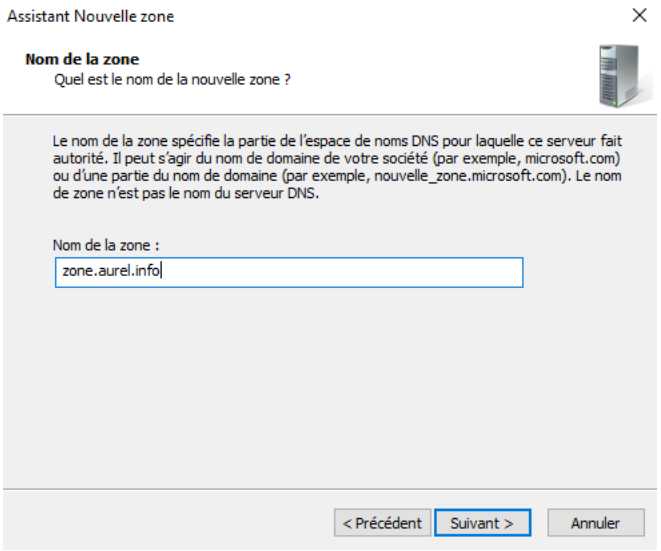
Ensuite, nous sélectionnons l'option :

Vers tous les serveurs DNS exécutés sur des contrôleurs de domaine dans mon domaine : aurelinfo.local

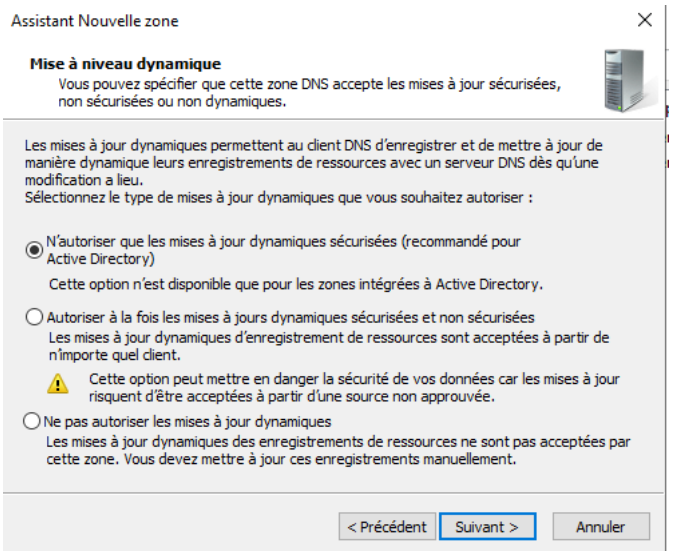
Puis, nous cliquons sur **Suivant** pour continuer la configuration.



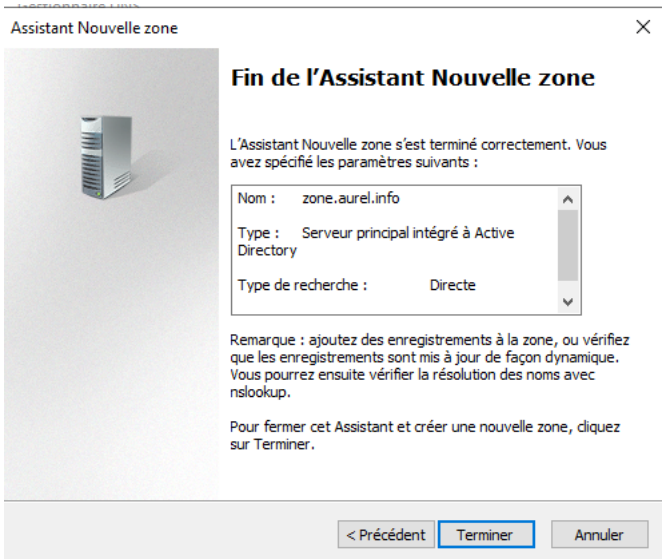
Nous allons nommé la nouvelle zone : **zone.aurelinfo.local**, Puis, nous cliquons sur **Suivant** pour continuer la configuration.



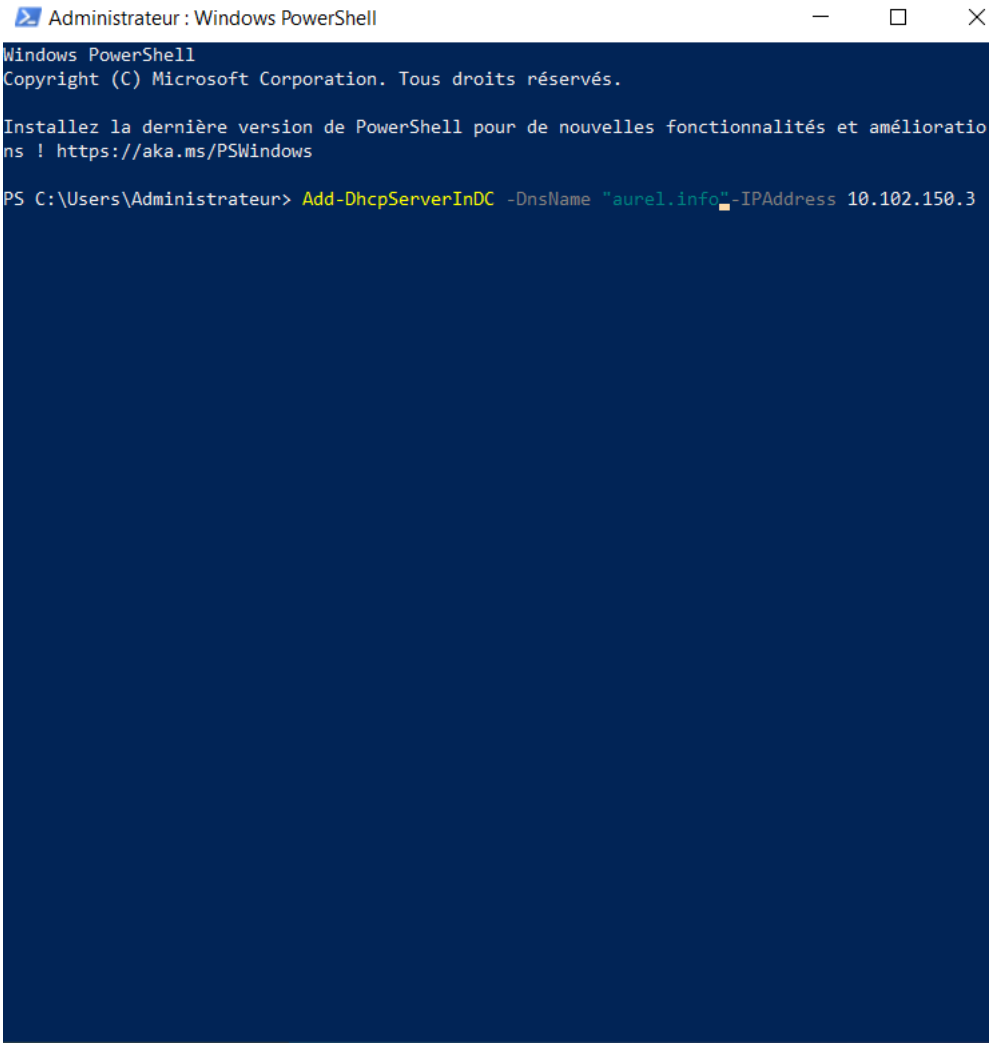
À l'étape des mises à jour dynamiques DNS, nous choisissons d'**autoriser uniquement les mises à jour sécurisées** pour renforcer la sécurité du réseau.
Puis, nous cliquons sur **Suivant** pour continuer.



Enfin, nous cliquons sur **Terminer** pour finaliser l'installation et la configuration du rôle **DNS**.

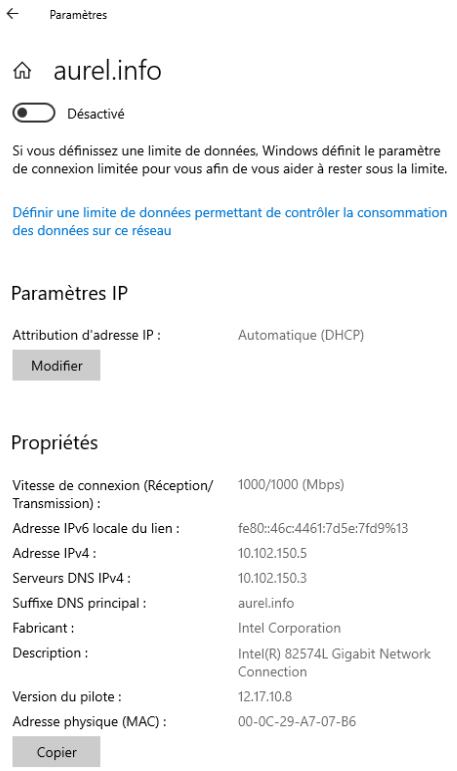


Ensuite, nous ouvrons **PowerShell en tant qu'administrateur** pour exécuter une commande qui permettra **d'autoriser notre serveur DHCP** à fonctionner dans l'environnement **Active Directory**.



Une fois le serveur DHCP autorisé, nous procédons à une **vérification** en connectant une **machine de test** au réseau.

Cela nous permet de vérifier que le **serveur DHCP distribue correctement une adresse IP** issue de l'étendue que nous avons configurée. Nous remarquons bien que la machine a reçu une adresse ip qui se trouve dans la plage du DHCP.



Intégration d'une machine Windows au domaine

Pour ajouter un PC au domaine **aurelinfo.local** :

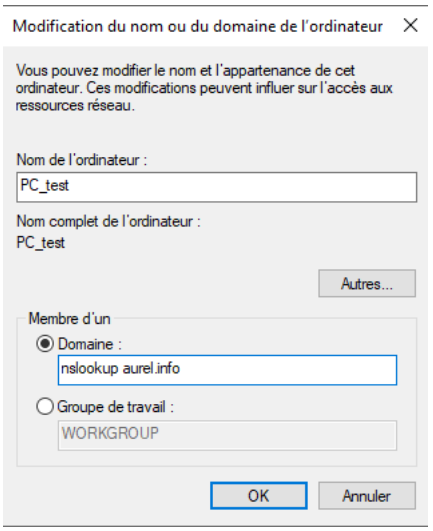
Nous allons dans **Paramètres** > **Système** > **Informations système**.

Nous renommons notre ordinateur en : **WIN10TEST1**

Nous cliquons sur **Rejoindre un domaine** et saisissons : **aurelinfo.local**

Nous entrons les **identifiants d'un administrateur du domaine**.

Enfin, nous redémarrons la machine pour finaliser son intégration au domaine.



Ensuite, nous allons réaliser la même opération pour un **poste Debian**. Pour vérifier si un **domaine Active Directory** est disponible sur notre réseau, nous exécutons la commande suivante dans un terminal : **realm discover aurelinfo.local**

```
user1@debian: ~
       valid_lft 689931sec preferred_lft 689931sec
    inet6 fe80::20c:29ff:fee0:2cd2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user1@debian:~$ realm discover aurelinfo.local
bash: realm : commande introuvable
user1@debian:~$ su -
Mot de passe :
root@debian:~# ^[[200~realm discover aurelinfo.local~
-bash: $'\E[200~realm' : commande introuvable
root@debian:~# realm discover aurelinfo.local
aurelinfo.local
  type: kerberos
  realm-name: AURELINFO.LOCAL
  domain-name: aurelinfo.local
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
root@debian:~#
```

Après avoir exécuté la commande realm discover, nous constatons que **Configured = no**, ce qui signifie que la machine Debian **n'est pas encore intégrée au domaine**.

Nous allons donc exécuter une commande pour **joindre la machine Debian au domaine Active Directory**.
Cette commande va créer automatiquement un **compte machine** dans Active Directory, permettant ainsi à Debian de devenir un **"membre officiel"** du réseau Windows.

La commande à exécuter est la suivante : **sudo realm join --user=Administrateur aurelinfo.local**

```
user1@debian: ~
    inet6 fe80::20c:29ff:fee0:2cd2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user1@debian:~$ realm discover aurelinfo.local
bash: realm : commande introuvable
user1@debian:~$ su -
Mot de passe :
root@debian:~# ^[[200~realm discover aurelinfo.local~
-bash: $'\E[200~realm' : commande introuvable
root@debian:~# realm discover aurelinfo.local
aurelinfo.local
  type: kerberos
  realm-name: AURELINFO.LOCAL
  domain-name: aurelinfo.local
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
root@debian:~# sudo realm join --user=Administrateur aurelinfo.local
Password for Administrateur:
```

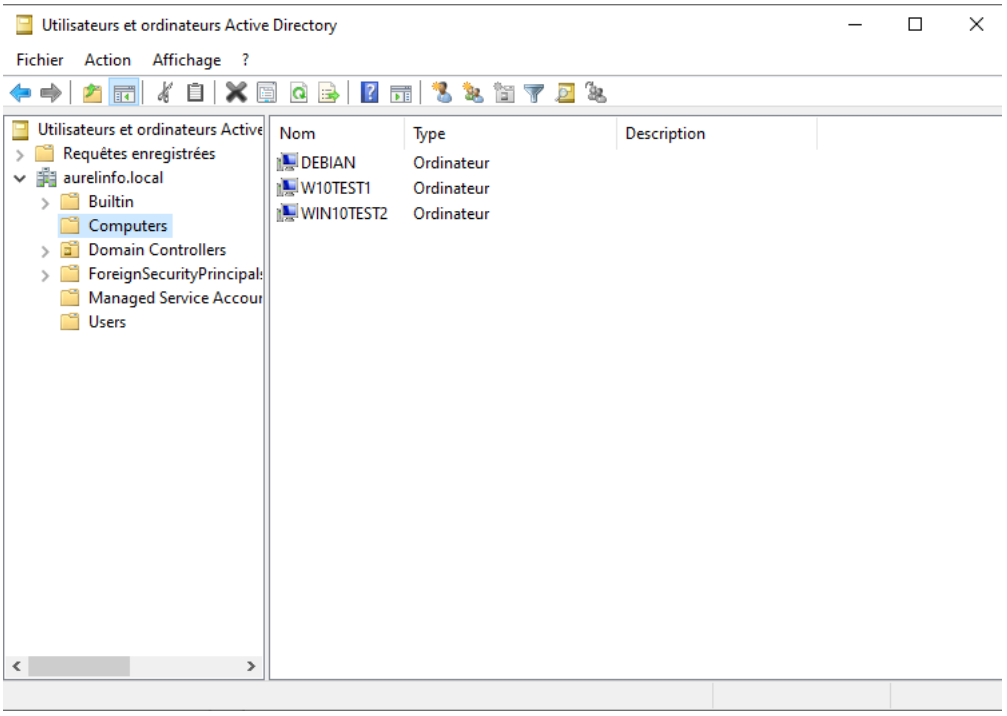
Lors de l'exécution de la commande, **il nous est demandé le mot de passe de l'administrateur du domaine**. Nous saisissons alors le **mot de passe du compte Administrateur** pour autoriser l'intégration de la machine Debian au domaine **aurelinfo.local**.

```
user1@debian: ~
:kages [272 kB]
Réception de :6 http://ftp.fr.debian.org/debian bookworm-backports/main Translat
ion-en [232 kB]
Réception de :7 http://ftp.fr.debian.org/debian bookworm-backports/contrib amd64
 Packages [5 664 B]
Réception de :8 http://ftp.fr.debian.org/debian bookworm-backports/contrib Trans
lation-en [5 448 B]
Réception de :9 http://ftp.fr.debian.org/debian bookworm-backports/non-free amd6
4 Packages [13,3 kB]
Réception de :10 http://ftp.fr.debian.org/debian bookworm-backports/non-free Tra
nslation-en [8 460 B]
Réception de :11 http://ftp.fr.debian.org/debian bookworm-backports/non-free-fir
mware amd64 Packages [3 868 B]
Réception de :12 http://ftp.fr.debian.org/debian bookworm-backports/non-free-fir
mware Translation-en [2 848 B]
503 ko réceptionnés en 1s (538 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
76 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les
voir.
root@debian:~# sudo realm join --user=Administrateur aurelinfo.local
Password for Administrateur:
root@debian:~#
```

Après avoir rejoint le domaine, nous vérifions que la configuration a bien été modifiée en exécutant la commande suivante : **realm list**

```
user1@debian: ~
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
76 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les
voir.
root@debian:~# sudo realm join --user=Administrateur aurelinfo.local
Password for Administrateur:
root@debian:~# realm list
aurelinfo.local
  type: kerberos
  realm-name: AURELINFO.LOCAL
  domain-name: aurelinfo.local
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
  login-formats: %U@aurelinfo.local
  login-policy: allow-realm-logins
root@debian:~#
```

Nous constatons que le champ **Configured** est bien passé à **yes**, confirmant que la machine Debian a rejoint le domaine avec succès. Nous allons maintenant **vérifier depuis notre serveur Active Directory** si la machine a bien été **ajoutée dans les ordinateurs du domaine**.



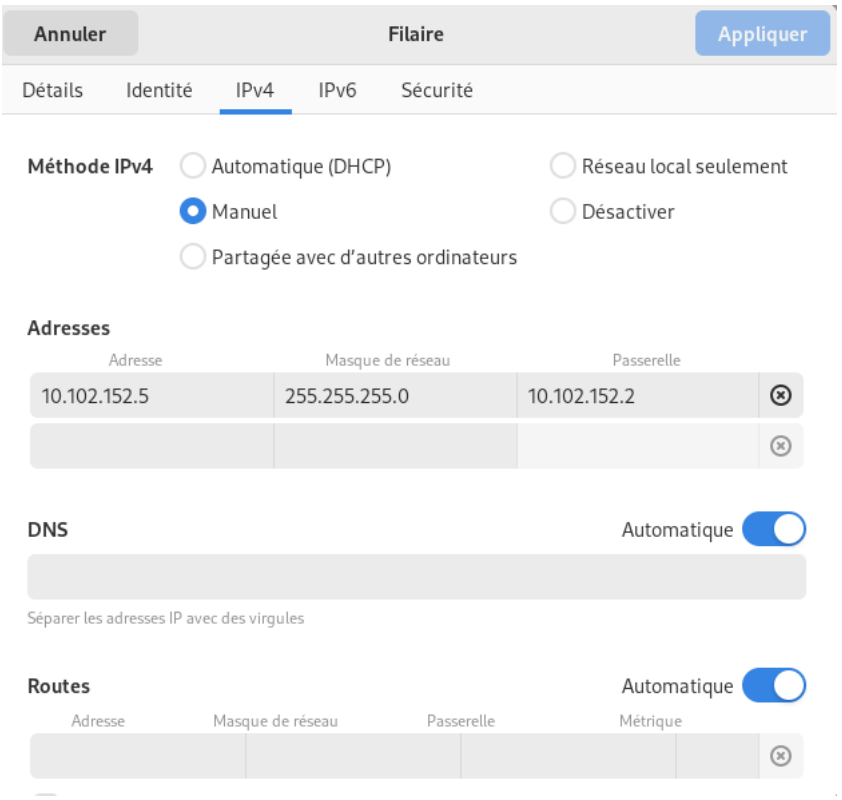
Nous remarquons bien qu'une machine Debian a été ajoutée au domaine. Nous allons maintenant effectuer la **même procédure**, mais cette fois-ci pour la **machine Debian qui héberge le serveur Zabbix**, afin de l'intégrer elle aussi au domaine Active Directory.

Avant cela, nous allons :

Fixer son adresse IP pour qu'elle soit permanente et ne soit **pas attribuée dynamiquement** par le serveur DHCP.
Nous utilisons l'adresse suivante : 10.102.152.5

Renommer la machine afin de l'identifier facilement dans l'annuaire Active Directory.
Nous la nommons : debian-avec-zabbix

Ensuite, nous suivons **exactement les mêmes étapes** (realm discover puis realm join) que précédemment pour intégrer cette machine au domaine **aurelinfo.local**.



Et voilà, toutes les configurations ont été réalisées avec succès.
Nos différentes machines — Windows et Debian, y compris le serveur hébergeant Zabbix — sont désormais correctement intégrées au domaine **aurelinfo.local** et prêtes à être supervisées.

```
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
root@DebianavecZabbix:~# sudo realm join --user=Administrateur aurelinfo.local
Password for Administrateur:
 * Installing necessary packages: sssd-tools adcli sssd libnss-sss libpam-sss
root@DebianavecZabbix:~# realm list
aurelinfo.local
  type: kerberos
  realm-name: AURELINFO.LOCAL
  domain-name: aurelinfo.local
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
  login-formats: %U@aurelinfo.local
  login-policy: allow-realm-logins
root@DebianavecZabbix:~#
```

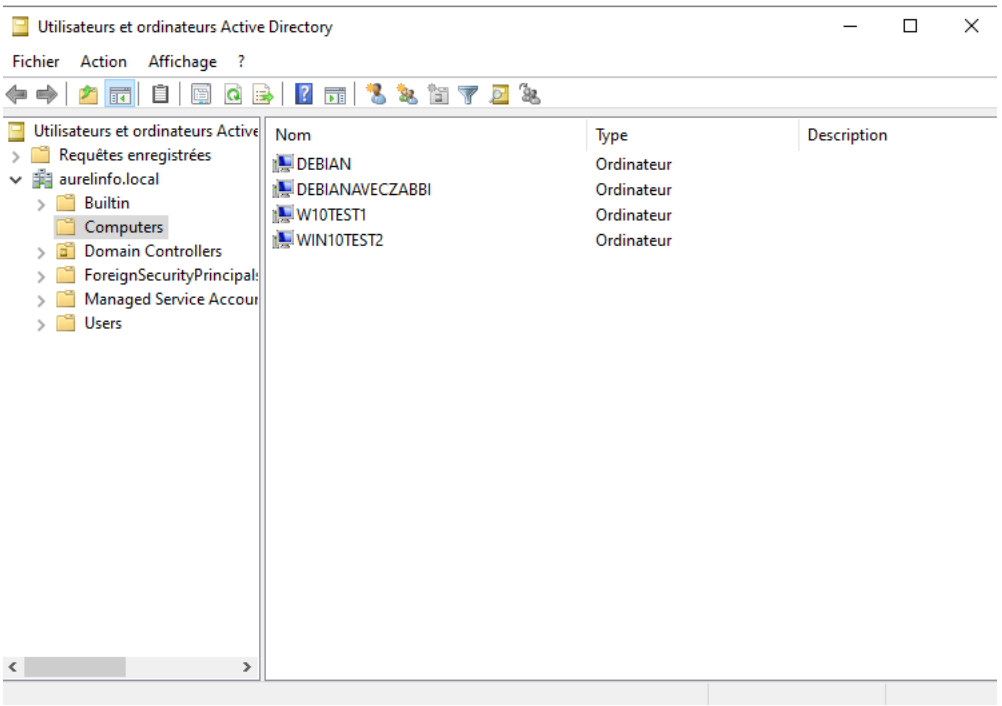
Nous allons maintenant **vérifier depuis notre serveur Active Directory** que toutes les machines ont bien été ajoutées au domaine. Pour cela, nous ouvrons la console **Utilisateurs et ordinateurs Active Directory** (dsa.msc) et nous nous assurons que :

WIN10TEST1

DEBIAN

DEBIAN-AVEC-ZABBIX

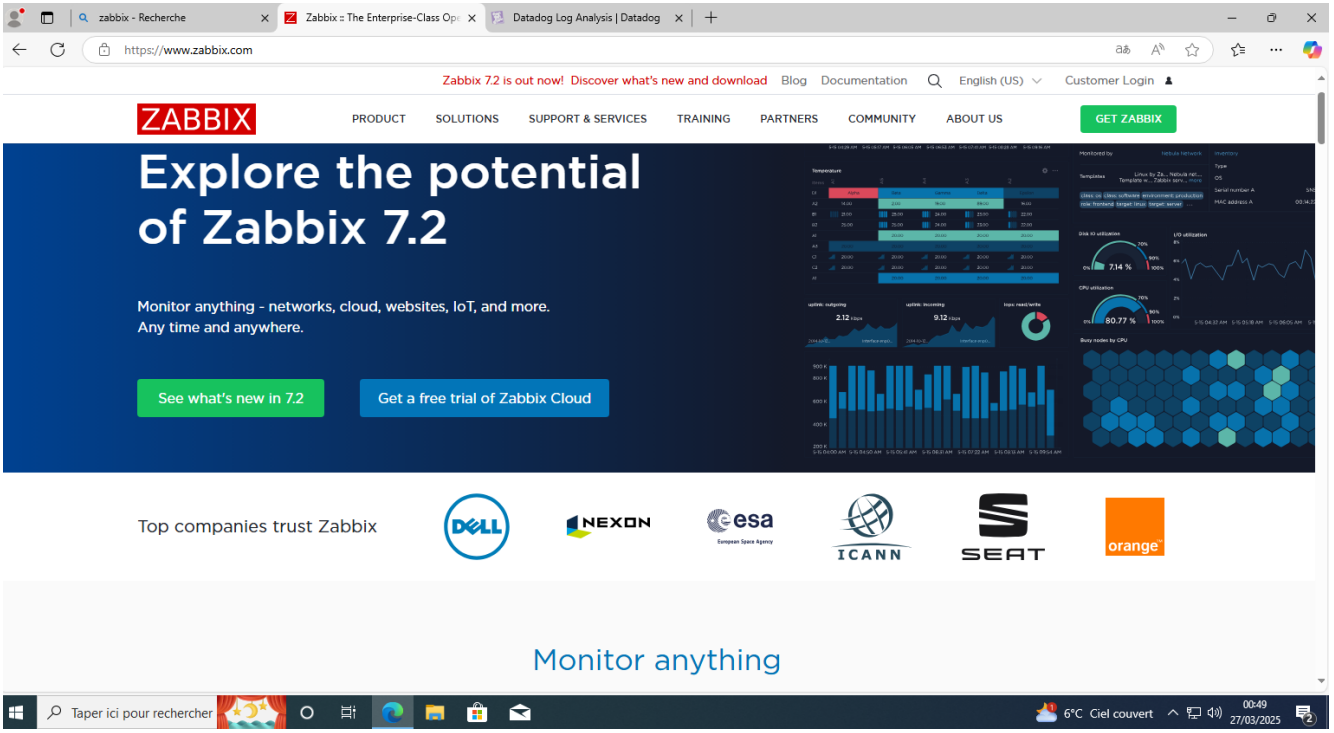
apparaissent bien dans le dossier **Computers**, confirmant leur appartenance au domaine **aurelinfo.local**.



IV - Configuration d’un agent Windows Serveur sur Zabbix

Nous allons maintenant configurer un **agent Zabbix sur une machine Windows Serveur**, afin qu’elle puisse être supervisée par notre serveur Zabbix.

Téléchargement de l’agent depuis notre **serveur Windows**, nous ouvrons un navigateur et nous nous rendons sur le site officiel de Zabbix à l’adresse suivante : **https://www.zabbix.com/download_**



Depuis la page d’accueil du site https://www.zabbix.com/download_, nous cliquons sur **"Get Zabbix"**, puis sur **"Zabbix Agents"** afin d’accéder à la liste des agents compatibles avec différents systèmes d’exploitation. Nous sélectionnons le système **Windows**, puis nous téléchargeons la **dernière version disponible**, à savoir : **Zabbix Agent v7.2.4** en cliquant sur **Download**.

zabbix - Recherche | Download Zabbix agents | Datadog Log Analysis | Datadog

https://www.zabbix.com/download_agents

Zabbix 7.2 is out now! Discover what's new and download | Blog | Documentation | English (US) | Customer Login

ZABBIX

PRODUCT | SOLUTIONS | SUPPORT & SERVICES | TRAINING | PARTNERS | COMMUNITY | ABOUT US | GET ZABBIX

Home / Product /

Get Zabbix

Zabbix Packages
Download and install Zabbix for free

Zabbix Cloud (Free trial)
Available with a 5-day free trial

Third-Party cloud vendors
Run Zabbix on third-party cloud platforms

Zabbix Containers
Deploy Zabbix components from official Zabbix containers

Zabbix Appliance
Run Zabbix from a pre-configured Zabbix virtual appliance

Zabbix Sources
Download the Zabbix source code

Zabbix Agents
Download Zabbix agent for Unix-like and Windows environments

Download pre-compiled Zabbix agent binaries

For Agent DEBs and RPMs please visit Zabbix packages

☐ Show legacy downloads

Taper ici pour rechercher | Résultats à venir | 00:53 27/03/2025

zabbix - Recherche | Download Zabbix agents | Datadog Log Analysis | Datadog

https://www.zabbix.com/download_agents

Zabbix 7.2 is out now! Discover what's new and download | Blog | Documentation | English (US) | Customer Login

ZABBIX

PRODUCT | SOLUTIONS | SUPPORT & SERVICES | TRAINING | PARTNERS | COMMUNITY | ABOUT US | GET ZABBIX

4.0 LTS

3.0 LTS

Zabbix Release: 7.2.4

Zabbix agent v7.2.4

Read manual

Packaging: MSI
Encryption: OpenSSL
Linkage: Dynamic
Checksum:
sha256: c960666f18376faa719c2e05b278b0babcfa5b9ffc27d3f140ecbde775abb0a2
sha1: 1617a0f646874ce6644cef5521f7d927ffba2312
md5: 10d0f615f446ab230f542e2c320302d4

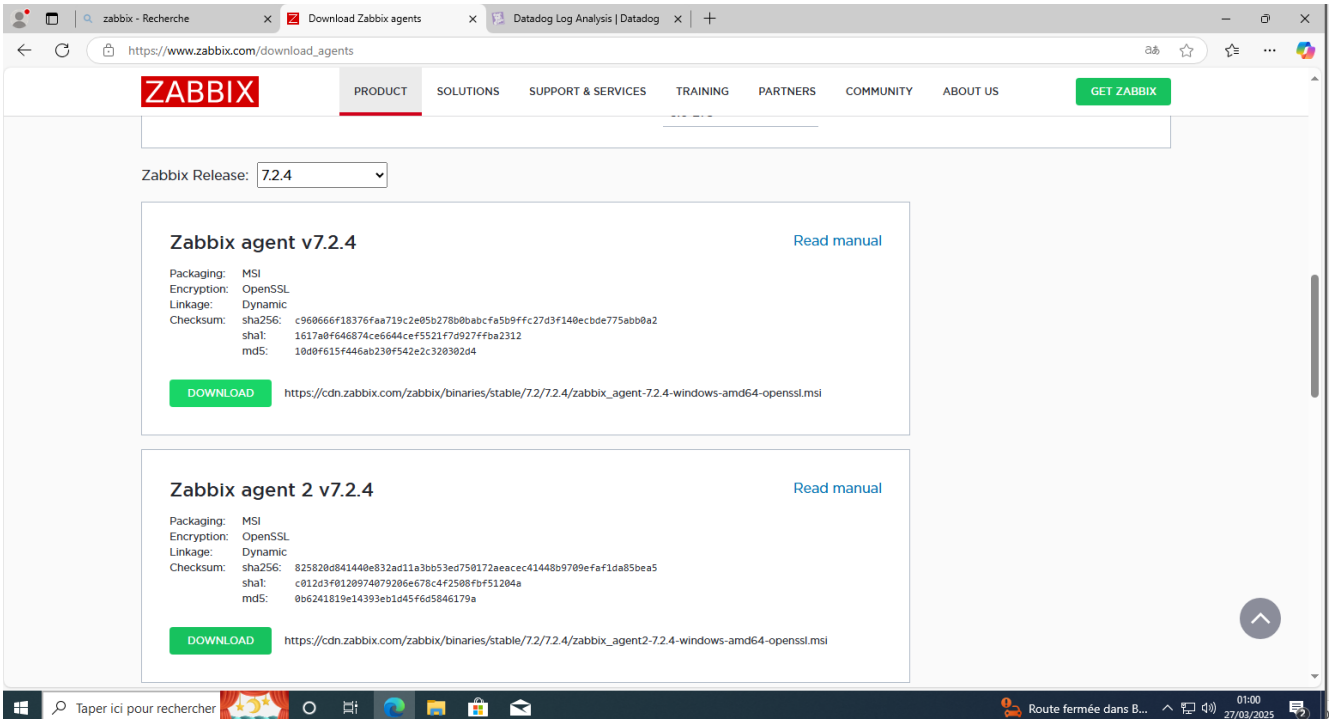
DOWNLOAD https://cdn.zabbix.com/zabbix/binaries/stable/7.2/7.2.4/zabbix_agent-7.2.4-windows-amd64-openssl.msi

Zabbix agent 2 v7.2.4

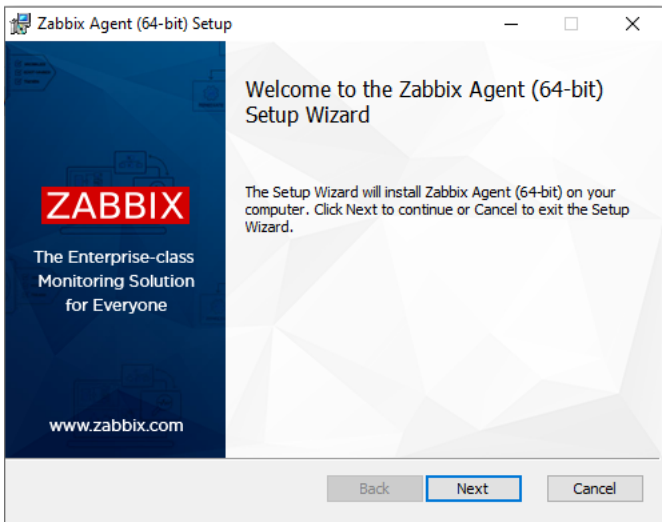
Read manual

Packaging: MSI
Encryption: OpenSSL
Linkage: Dynamic
Checksum:
sha256: 825820d841440e832ad11a3bb53ed750172aeacec41448b9709efaf1da85bea5
sha1: c012d3f0120974079206e678c4f2508fbf51204a
md5: 0b6241819e14393eb1d45f6d5846179a

Taper ici pour rechercher | Les salaires 2025 en L... | 00:57 27/03/2025

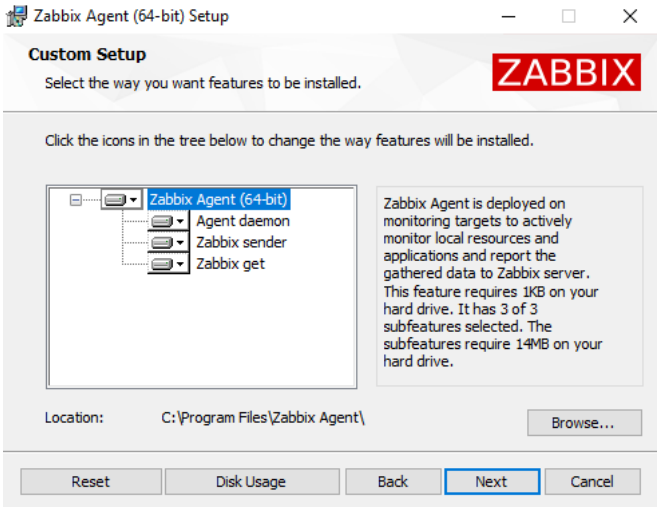


Une fois le fichier téléchargé, nous ouvrons l'**installateur de l'agent Zabbix** sur notre **serveur Windows**. Une **fenêtre de bienvenue** s'affiche : nous cliquons sur **"Next"** pour débuter l'installation.

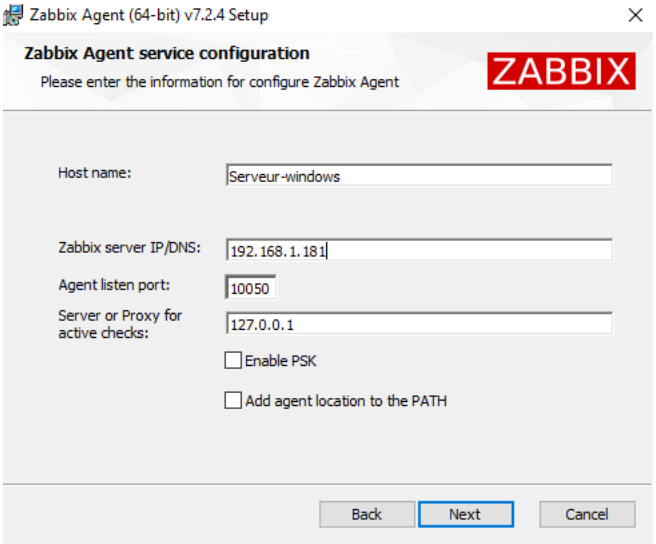


Nous arrivons ensuite sur la page des **conditions générales d'utilisation** (*End-User License Agreement*). Nous **acceptons la licence** en cochant la case correspondante, puis nous cliquons sur **"Next"** pour poursuivre l'installation sur notre **serveur Windows**.

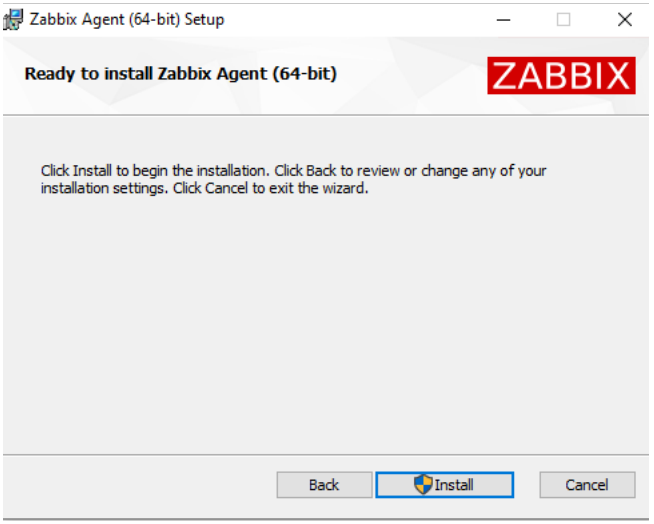
Ensuite, nous arrivons à l'étape **“Custom Setup”**. Nous laissons tous les paramètres par défaut sans effectuer de modifications, puis nous cliquons simplement sur **“Next”** pour continuer l’installation sur notre **serveur Windows**.



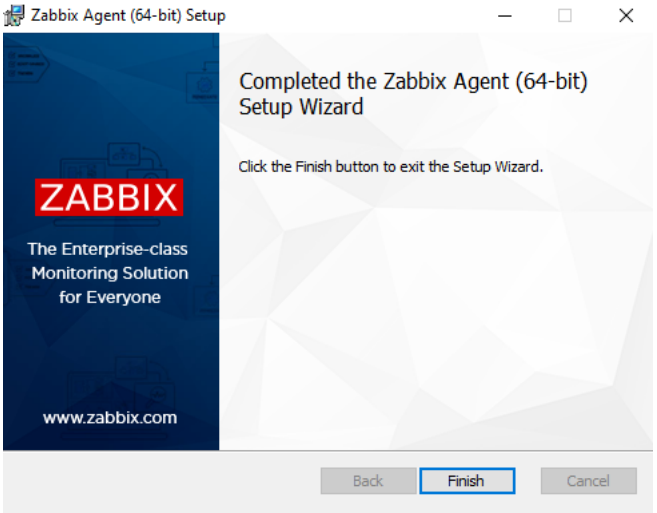
Après avoir récupéré l'adresse IP du serveur Zabbix (ici **10.102.152.5**), nous retournons sur notre **serveur Windows** afin de **renseigner cette adresse IP** dans le champ prévu lors de l'installation de l'agent.



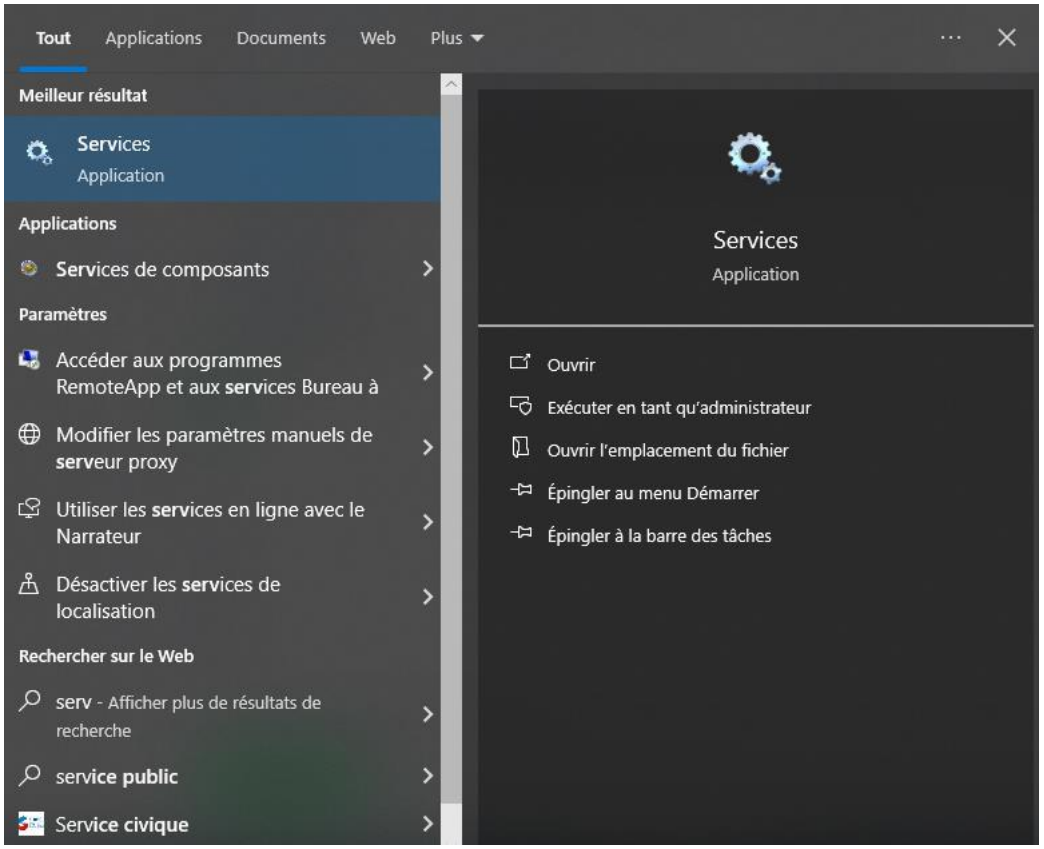
Une fois l'adresse IP renseignée, nous cliquons sur **“Next”**, puis sur **“Install”** pour lancer l'installation de l'agent Zabbix sur notre **serveur Windows**.



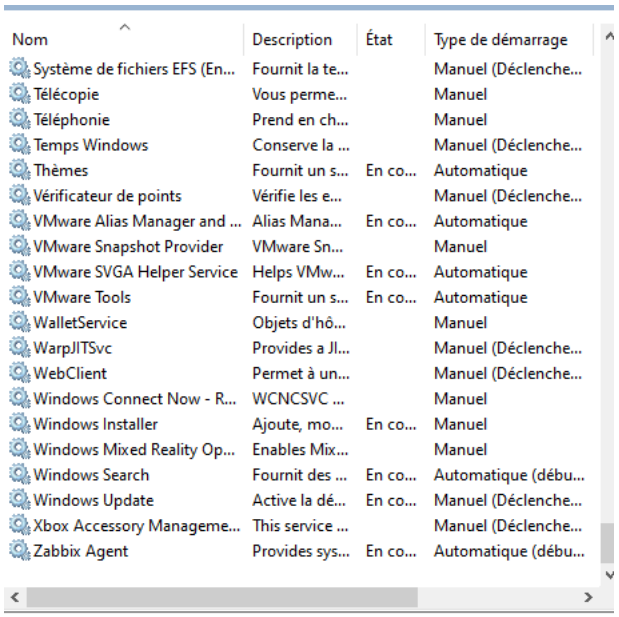
Lorsque l'installation est terminée, nous cliquons sur **“Finish”** pour clôturer l’assistant d’installation de l'agent Zabbix sur notre **serveur Windows**.



Après avoir terminé l'installation, nous allons dans la **barre de recherche** de Windows et nous tapons **“Services”** afin d'accéder à la console de gestion des services.

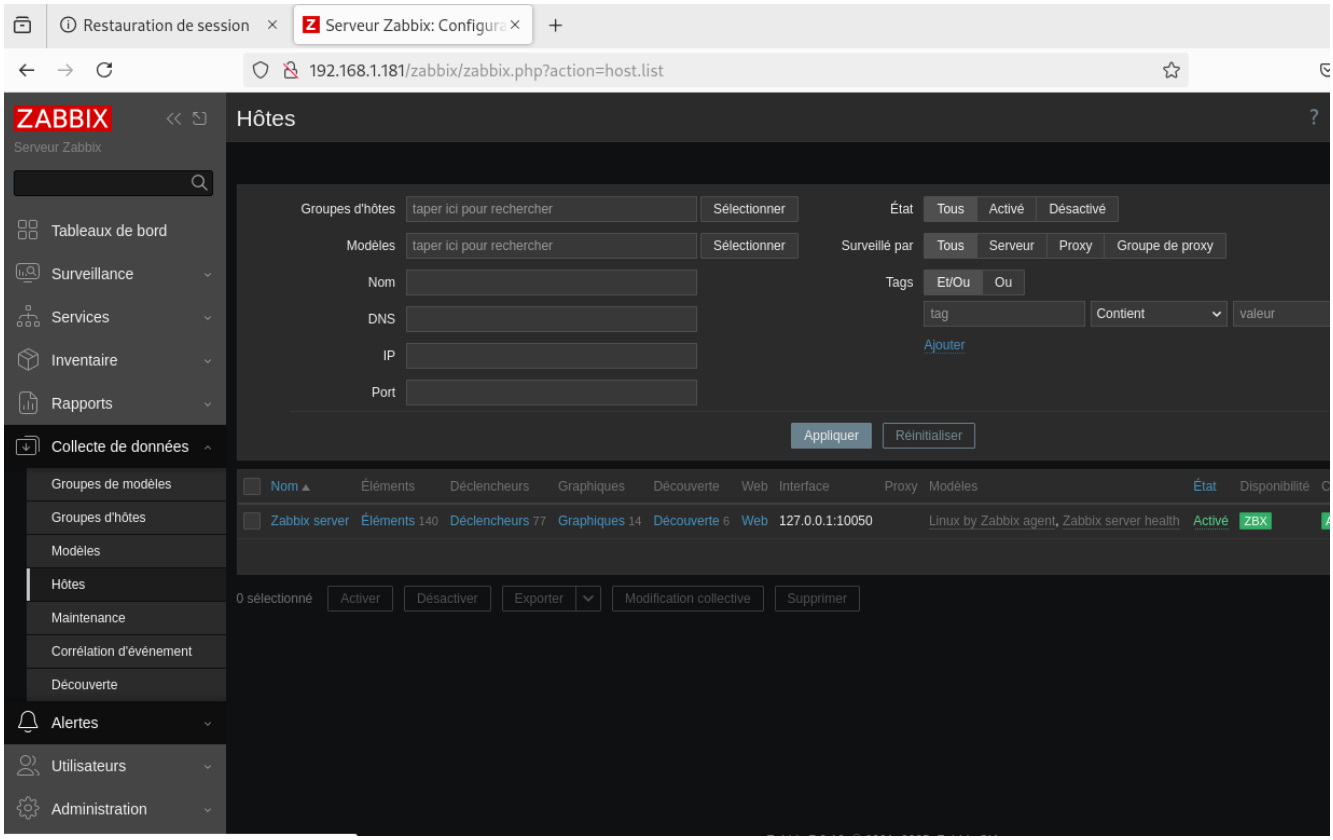


Nous cliquons sur **“Services”** pour ouvrir la console, puis nous descendons tout en bas de la liste afin de trouver le service **“Zabbix Agent”**.Nous vérifions que son **statut** est bien **"En cours d’exécution"**, confirmant que l’agent fonctionne correctement sur notre **serveur Windows**.

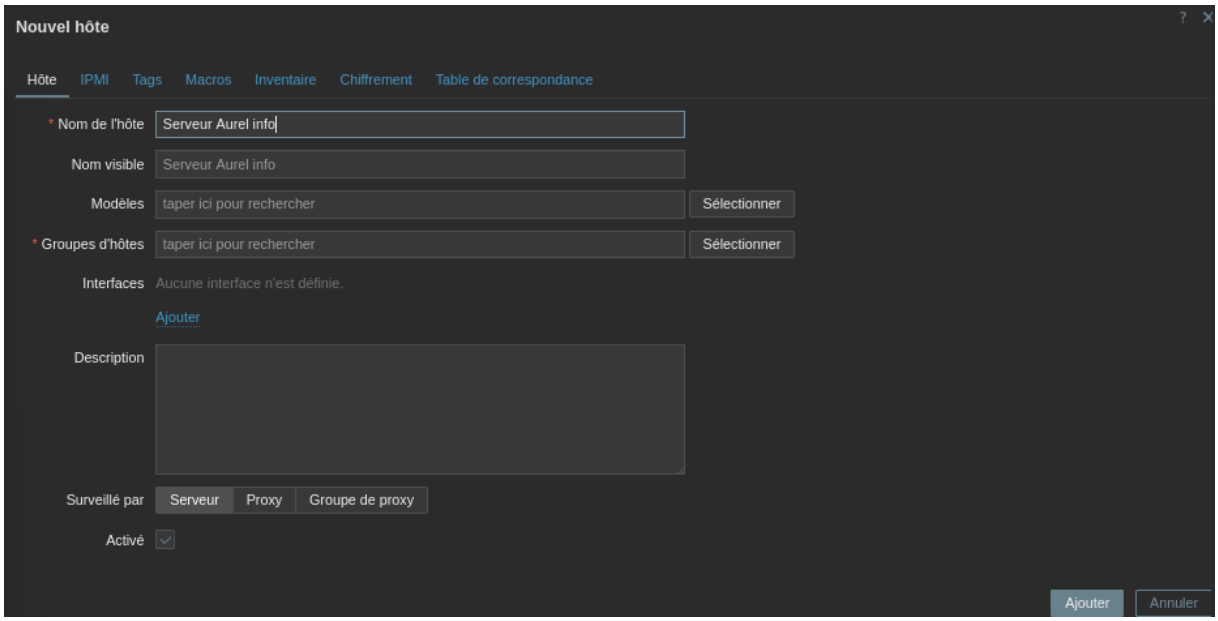


Nom	Description	État	Type de démarrage
Système de fichiers EFS (En...	Fournit la te...		Manuel (Déclenche...
Télécopie	Vous perme...		Manuel
Téléphonie	Prend en ch...		Manuel
Temps Windows	Conserve la ...		Manuel (Déclenche...
Thèmes	Fournit un s...	En co...	Automatique
Vérificateur de points	Vérifie les e...		Manuel (Déclenche...
VMware Alias Manager and ...	Alias Mana...	En co...	Automatique
VMware Snapshot Provider	VMware Sn...		Manuel
VMware SVGA Helper Service	Helps VMw...	En co...	Automatique
VMware Tools	Fournit un s...	En co...	Automatique
WalletService	Objets d'hô...		Manuel
WarpJITSvc	Provides a JI...		Manuel (Déclenche...
WebClient	Permet à un...		Manuel (Déclenche...
Windows Connect Now - R...	WCNCSVC ...		Manuel
Windows Installer	Ajoute, mo...	En co...	Manuel
Windows Mixed Reality Op...	Enables Mix...		Manuel
Windows Search	Fournit des ...	En co...	Automatique (débu...
Windows Update	Active la dé...	En co...	Manuel (Déclenche...
Xbox Accessory Manageme...	This service ...		Manuel (Déclenche...
Zabbix Agent	Provides sys...	En co...	Automatique (débu...

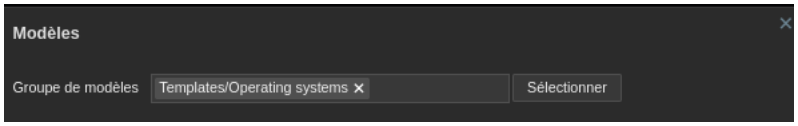
Une fois toutes ces étapes terminées, nous pouvons laisser de côté notre **serveur Windows**. Nous retournons sur notre **Debian** hébergeant le serveur **Zabbix**.Depuis l'interface web de Zabbix, nous allons dans **“Collecte de Données”**, puis dans **“Hôtes”**, et nous cliquons sur **“Créer un hôte”** afin d’ajouter notre serveur Windows à la supervision.



Une fois sur la page de création d'hôte, nous ajoutons notre **serveur Windows**.Nous commençons par **nommer l’hôte**, ici : **Serveur Aurel Info**

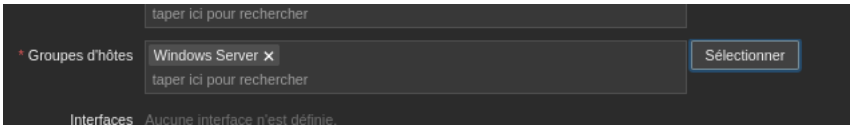


Ensuite, à côté de la rubrique **“Modèle”**, nous cliquons sur **“Sélectionner”**.Dans la liste qui s'affiche, nous choisissons la catégorie **“Templates/Operating Systems”** pour accéder aux modèles adaptés aux systèmes d’exploitation.

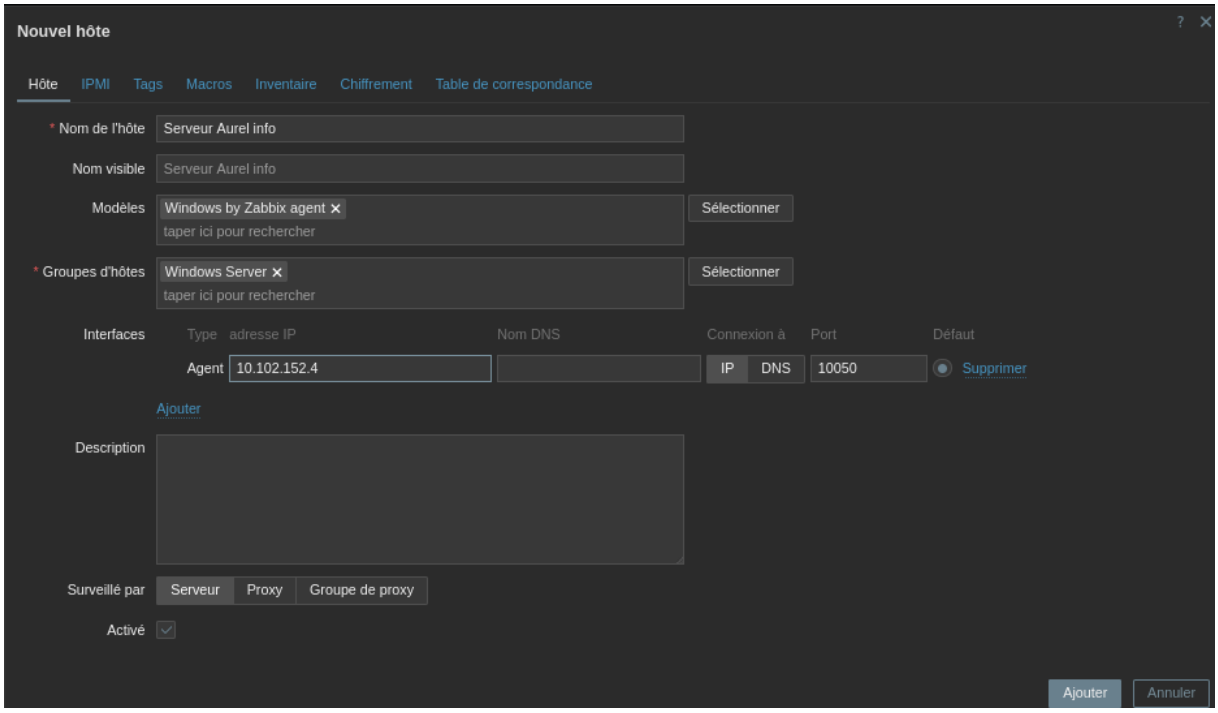


Dans la liste des modèles, nous sélectionnons **“Template OS Windows by Zabbix agent”** afin d'appliquer une supervision adaptée à notre serveur Windows.

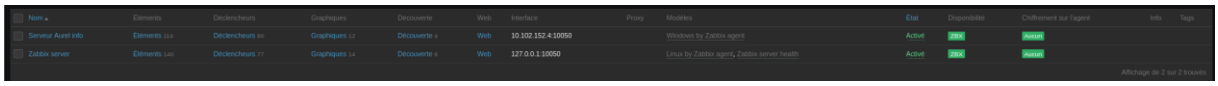
Ensuite, nous ajoutons notre hôte à un **groupe d’hôtes**.Dans notre cas, nous créons un nouveau groupe que nous nommons : **Windows Server**



Après avoir configuré le nom, le modèle et le groupe de l’hôte, nous cliquons sur **“Ajouter”** dans la section **Agent**, ce qui nous permet de renseigner l’adresse IP de la machine à superviser.Dans notre cas, nous indiquons l’adresse IP de notre **serveur Windows** : **10.102.152.4**



Une fois toutes les informations saisies, nous cliquons sur **“Ajouter”** pour enregistrer l’hôte.Nous devons ensuite **patienter quelques instants** jusqu’à ce que l'icône de disponibilité de l'agent devienne **verte**, indiquant que la communication entre le serveur Zabbix et notre serveur Windows est bien établie.



V - Ajout des templates Windows et création d'alertes

Le but de l'ajout d'un **template Windows** est de permettre à **Zabbix** de **surveiller automatiquement l'état général** de notre système Windows.

Lors de l'ajout de l'agent, nous avons déjà associé un template qui permet de superviser automatiquement :

L'utilisation du **processeur (CPU)**,

L'utilisation de la **mémoire (RAM)**,

L'espace disponible sur les **disques**,

Le **trafic réseau** (entrant/sortant),

L'état des **services système critiques**,

Le **temps de disponibilité** (*uptime*).

Nous avons également configuré la supervision des **services réseau essentiels** :

Le service **DHCP**,

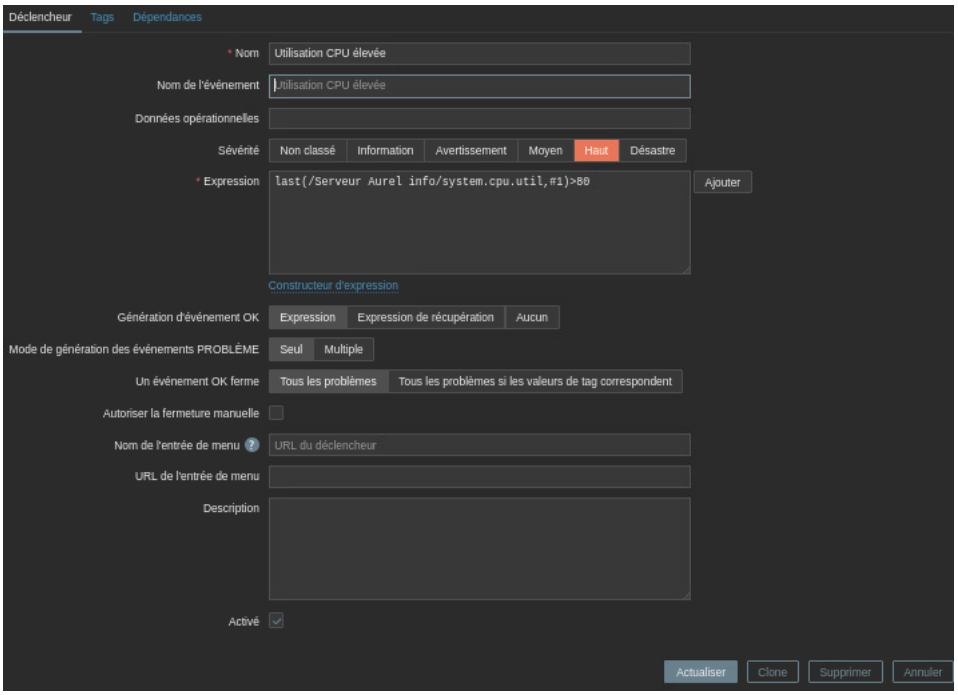
Le service **DNS**,

Le service **Active Directory (AD DS)**.

Maintenant, nous allons créer des **alertes personnalisées** afin que Zabbix nous **avertisse automatiquement** en cas de dysfonctionnement ou de surcharge de ces éléments critiques.

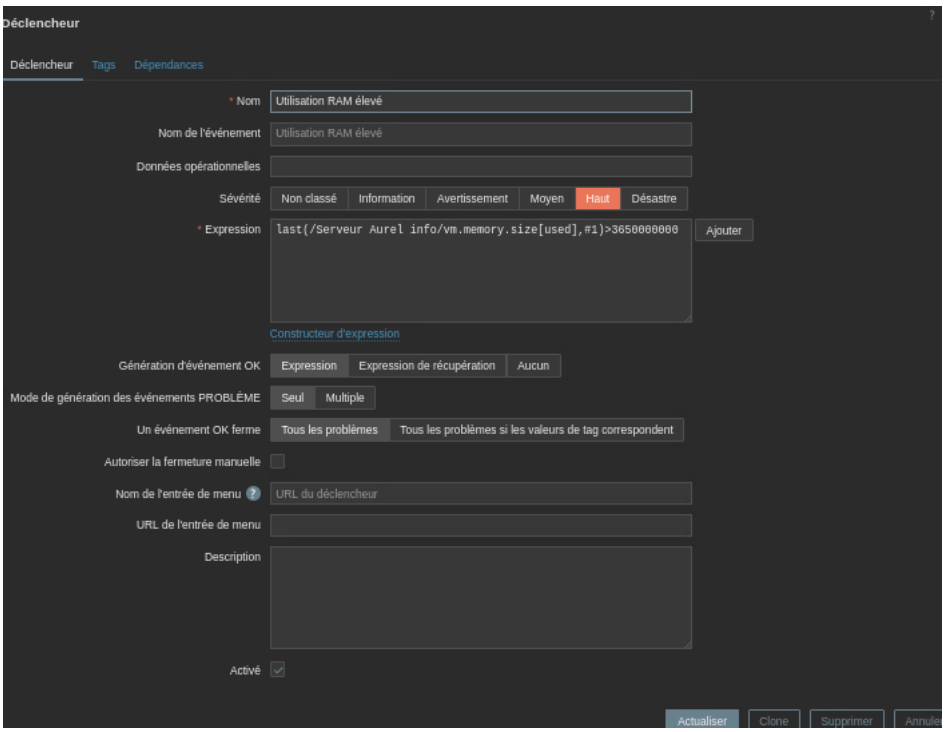
1) Création d'une alerte pour l'utilisation du CPU

C'est le même principe que pour les autres ressources.Nous allons créer un **déclencheur** qui nous alertera si l'**inactivité du CPU dépasse 80%**, ce qui indique une charge élevée.Nous allons dans **Collecte de Données**, puis **Hôtes**, et nous sélectionnons **Serveur Aurel Info**.Nous cliquons sur **Déclencheurs**, puis sur **Créer un déclencheur**.Nous nommons le déclencheur **"Utilisation CPU élevée"**, puis nous cliquons sur **Ajouter** à côté de **Expression** pour définir la condition.



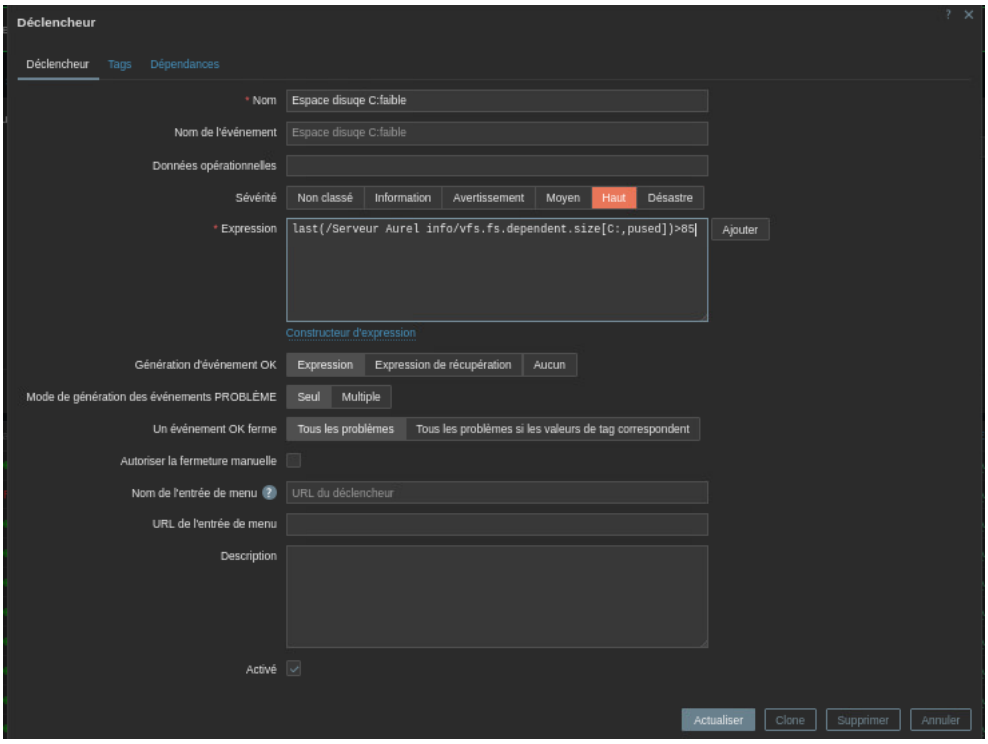
2) Création d'une alerte pour la RAM

C’est le même principe que pour le CPU.Nous allons simplement **changer le nom** du déclencheur et son **expression**.Ce déclencheur sera créé pour **alerter si l’utilisation de la RAM dépasse 85%** sur le **Serveur Aurel Info**.Nous cliquons sur **Ajouter** à côté de **Expression** pour définir la condition.



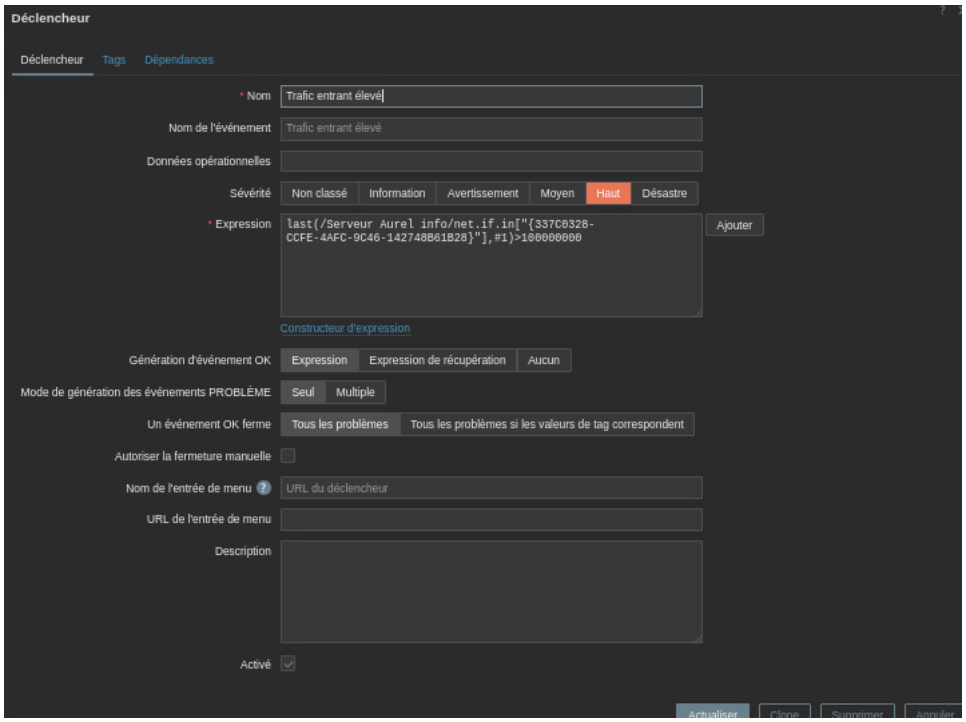
3) Création d'une alerte pour l’espace disque

C’est toujours le même principe.Nous allons créer un **déclencheur** pour recevoir une alerte si **l’utilisation du disque C:** dépasse **85%**.Nous allons dans **Collecte de Données**, puis **Hôtes**, et nous sélectionnons **Serveur Aurel Info**.Nous cliquons sur **Déclencheurs**, puis sur **Créer un déclencheur**.Nous nommons le déclencheur **"Espace disque C: faible"**, puis nous cliquons sur **Ajouter** à côté de **Expression**.



4) Création d'une alerte pour le trafic réseau entrant

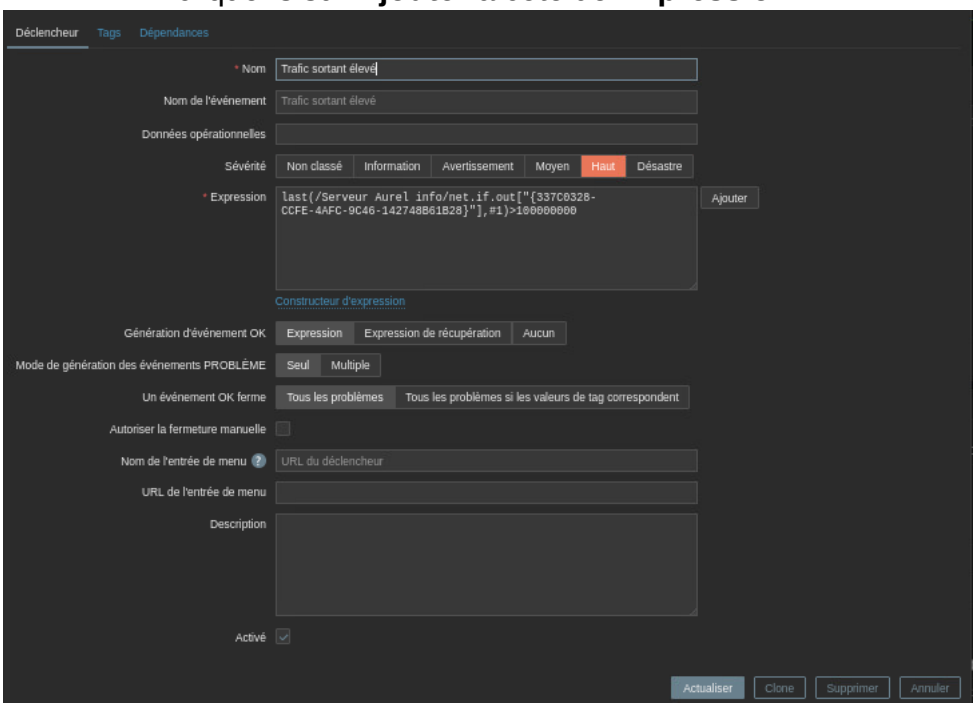
Un autre **déclencheur** sera mis en place pour nous **alerter en cas de trafic entrant élevé** sur l’interface réseau.Comme d’habitude, nous allons dans **Collecte de Données**, puis **Hôtes**, et nous sélectionnons **Serveur Aurel Info**.
Nous cliquons sur **Déclencheurs**, puis sur **Créer un déclencheur**.
Nous nommons le déclencheur **"Trafic entrant élevé"**, puis nous cliquons sur **Ajouter** à côté de **Expression**.



5) Création d'une alerte pour le trafic réseau sortant

Un autre **déclencheur** sera configuré pour **alerter si le trafic sortant sur l'interface réseau Ethernet0 dépasse 100 Mb/s**.Nous allons dans **Collecte de Données**, puis **Hôtes**, et nous sélectionnons **Serveur Aurel Info**.Nous cliquons sur **Déclencheurs**, puis sur **Créer un déclencheur**.

Nous nommons le déclencheur **"Trafic sortant élevé (Ethernet0)"**, puis nous cliquons sur **Ajouter** à côté de **Expression**.



6) Création d’alertes pour les services système (DHCP, DNS et AD DS)

Trois **déclencheurs** seront créés pour **alerter si l’un des services critiques est arrêté** : **DHCP**, **DNS**, ou **Active Directory (AD DS)**.

Zabbix déclenchera une alerte **si l’état du service est différent de 0**, ce qui signifie qu’il **n’est pas en cours d’exécution**.

Comme pour les autres alertes, nous allons dans **Collecte de Données**, puis **Hôtes**, et nous sélectionnons **Serveur Aurel Info**.
Nous cliquons sur **Déclencheurs**, puis sur **Créer un déclencheur** pour chaque service.
Nous les nommons respectivement :

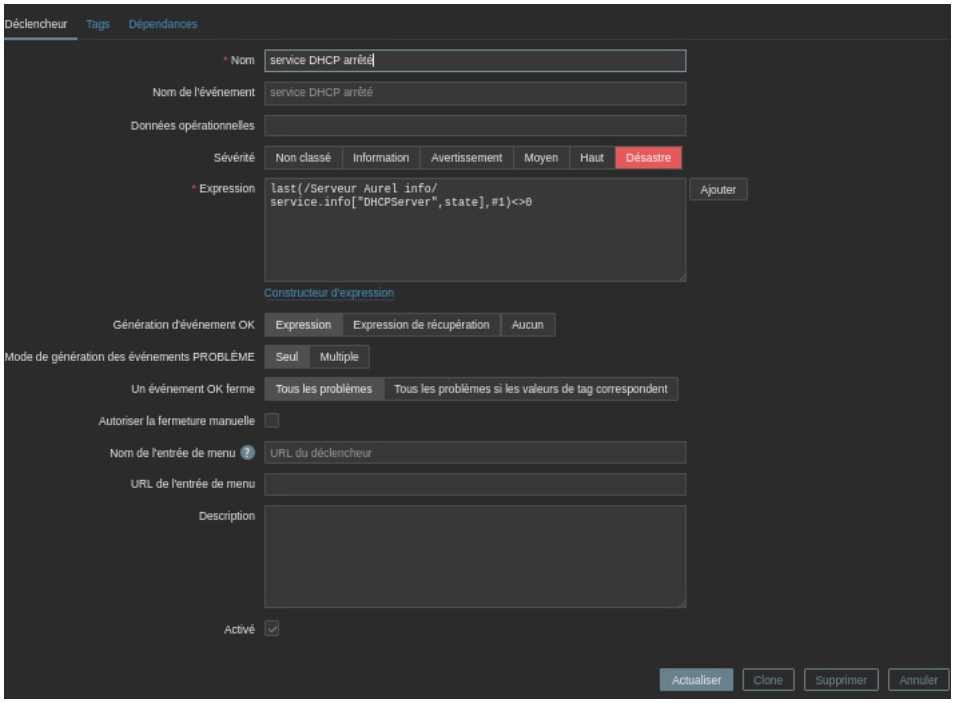
Service DHCP arrêté

Service DNS arrêté

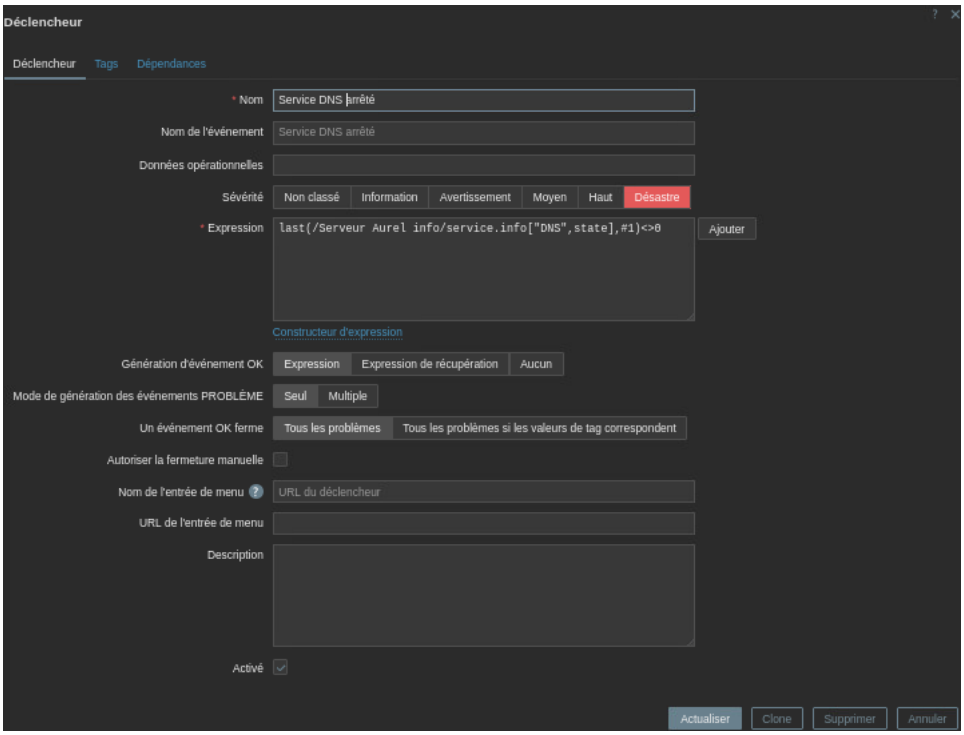
Service AD DS arrêté

Puis nous cliquons sur **Ajouter** à côté de **Expression** pour chacun d’eux.

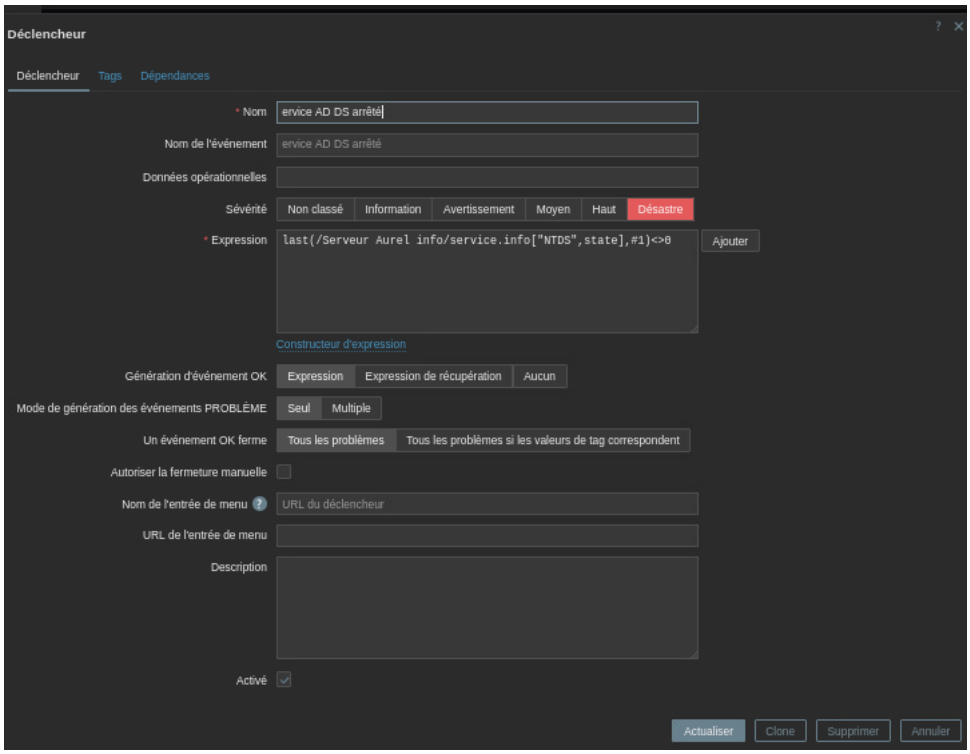
a) DHCP



b) DNS



c) AD DS



7) Supervision complémentaire : Zabbix Agent, disponibilité réseau

En complément de la supervision des services critiques, **d'autres déclencheurs** seront également mis en place pour surveiller :

L'**état du service Zabbix Agent**,

La **disponibilité réseau** du serveur (par **ping**),

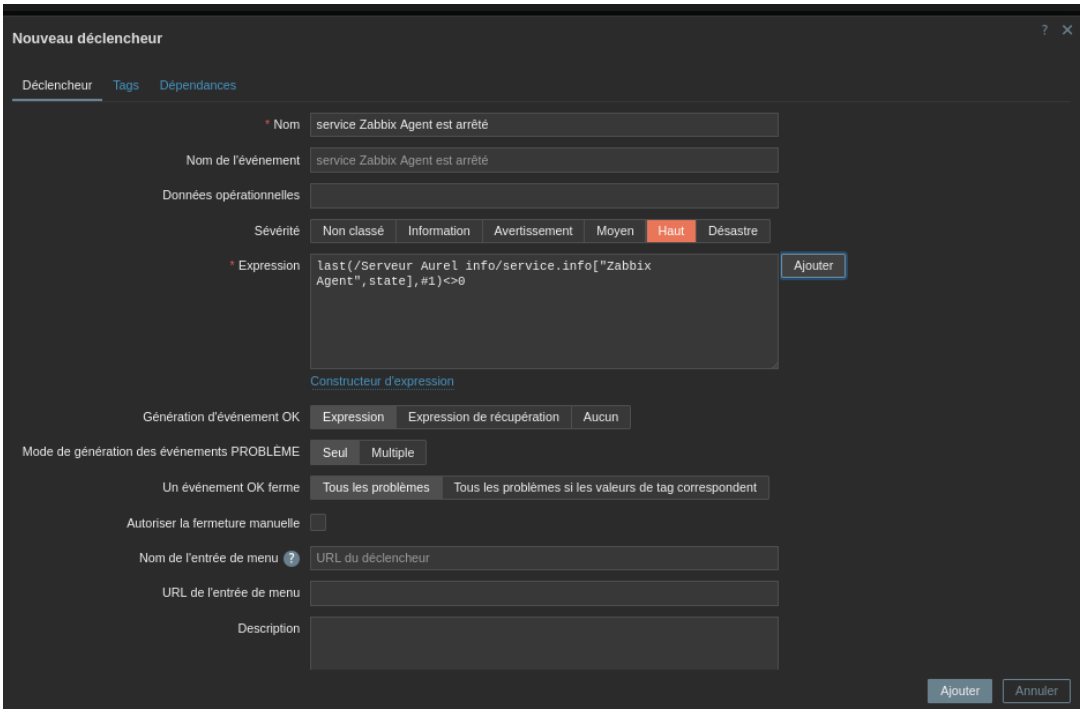
Certains **services secondaires** indispensables au bon fonctionnement du système.

Création d'une alerte pour l'état du Zabbix Agent

Un **déclencheur** sera ajouté pour **alerter si le service Zabbix Agent est arrêté**, car cela empêcherait la supervision du serveur par Zabbix.

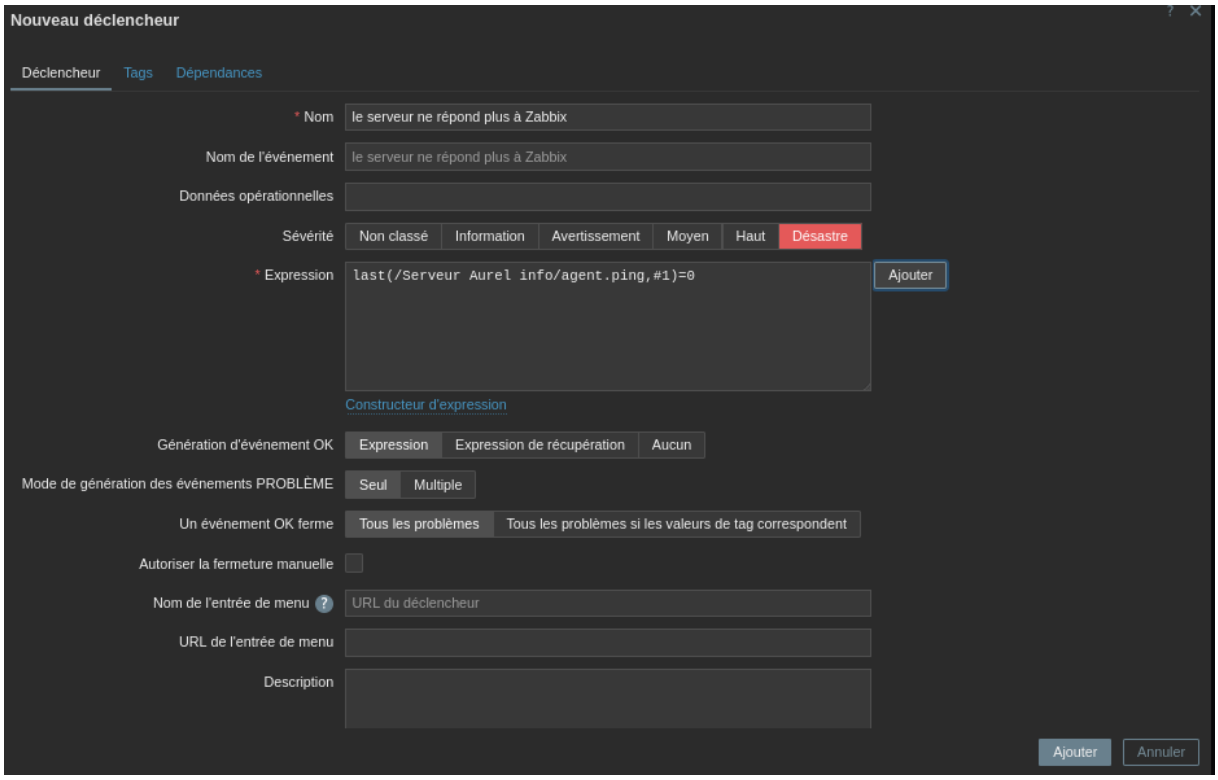
Comme d'habitude, nous allons dans **Collecte de Données**, puis **Hôtes**, et nous sélectionnons **Serveur Aurel Info**.Nous cliquons sur **Déclencheurs**, puis sur **Créer un déclencheur**.Nous le nommons: Service Zabbix Agent arrêté et nous cliquons sur **Ajouter** à côté de **Expression**.

a) Zabbix Agent



b) disponibilité réseau

Un déclencheur "ping" sera configuré pour alerter si le serveur ne répond plus à Zabbix, signalant une perte de communication.



8) Création d'alertes pour les services secondaires

Des **déclencheurs** sont également définis pour **surveiller les services secondaires** tels que :

Spooler (gestion de l'impression),

W32Time (synchronisation de l'heure système).

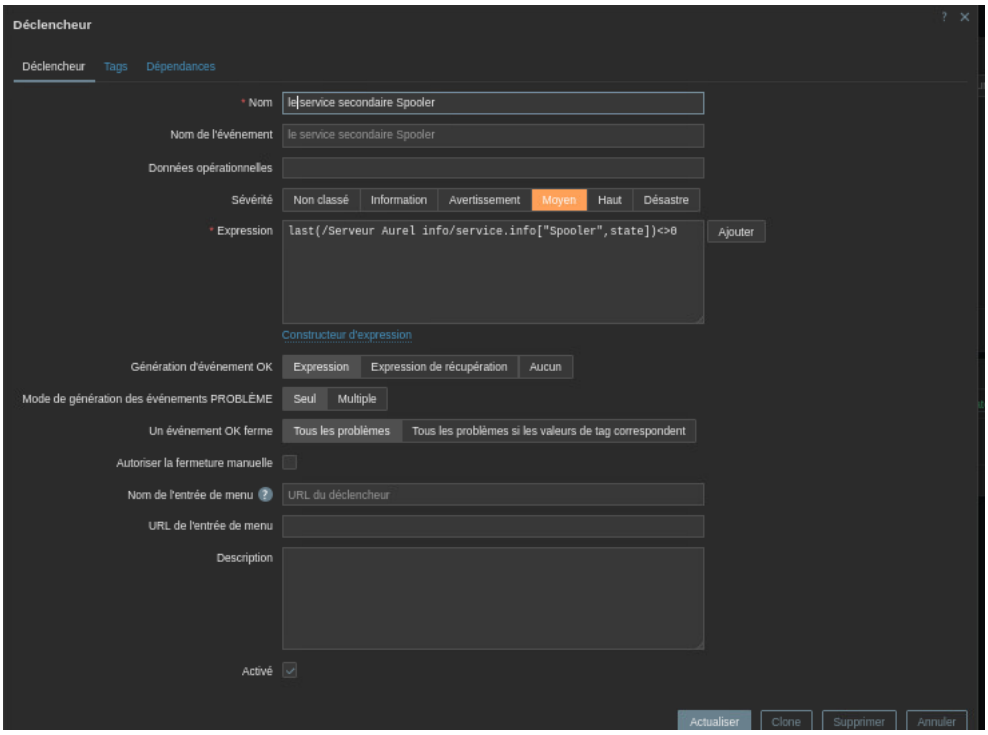
Bien qu'ils ne soient pas critiques pour la supervision de base, ces services sont essentiels pour garantir :

Le bon fonctionnement de l'**impression en réseau**,

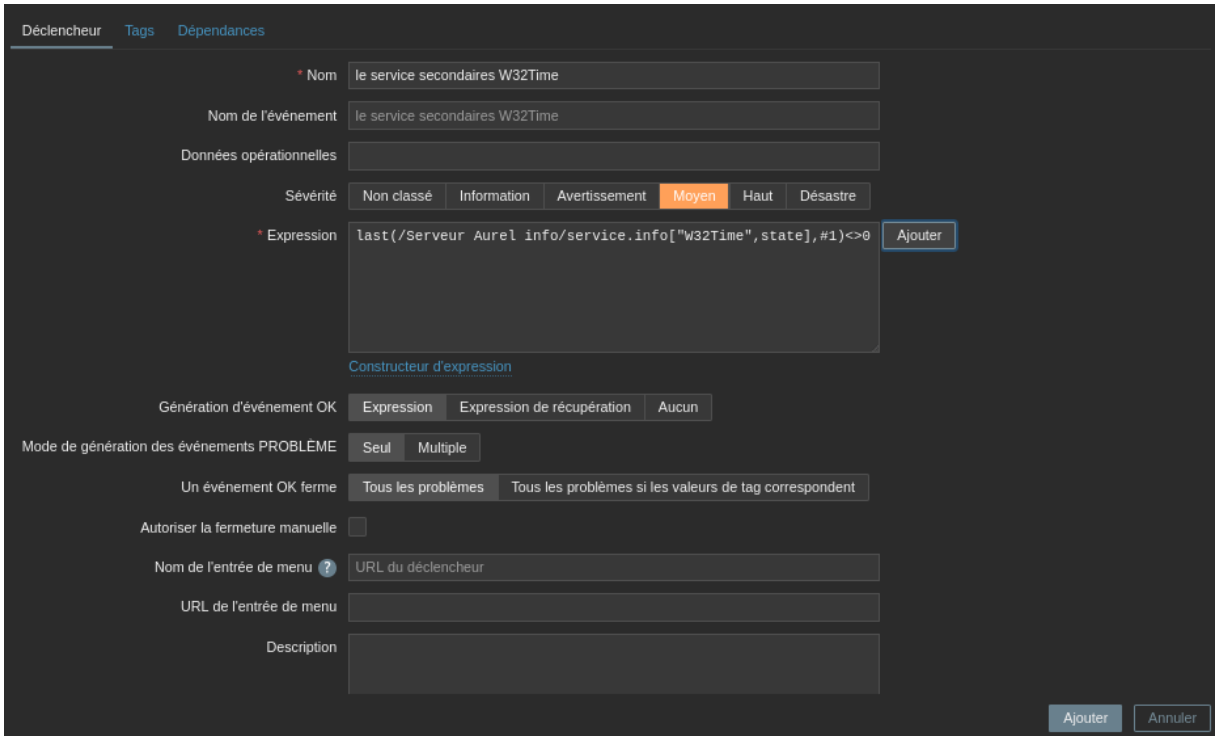
La **cohérence horaire** au sein du domaine Active Directory.

Nous procédons comme précédemment : nous allons dans **Collecte de Données**, puis **Hôtes**, nous sélectionnons **Serveur Aurel Info**, et nous créons un **déclencheur** pour chaque service à surveiller.

a) Spooler



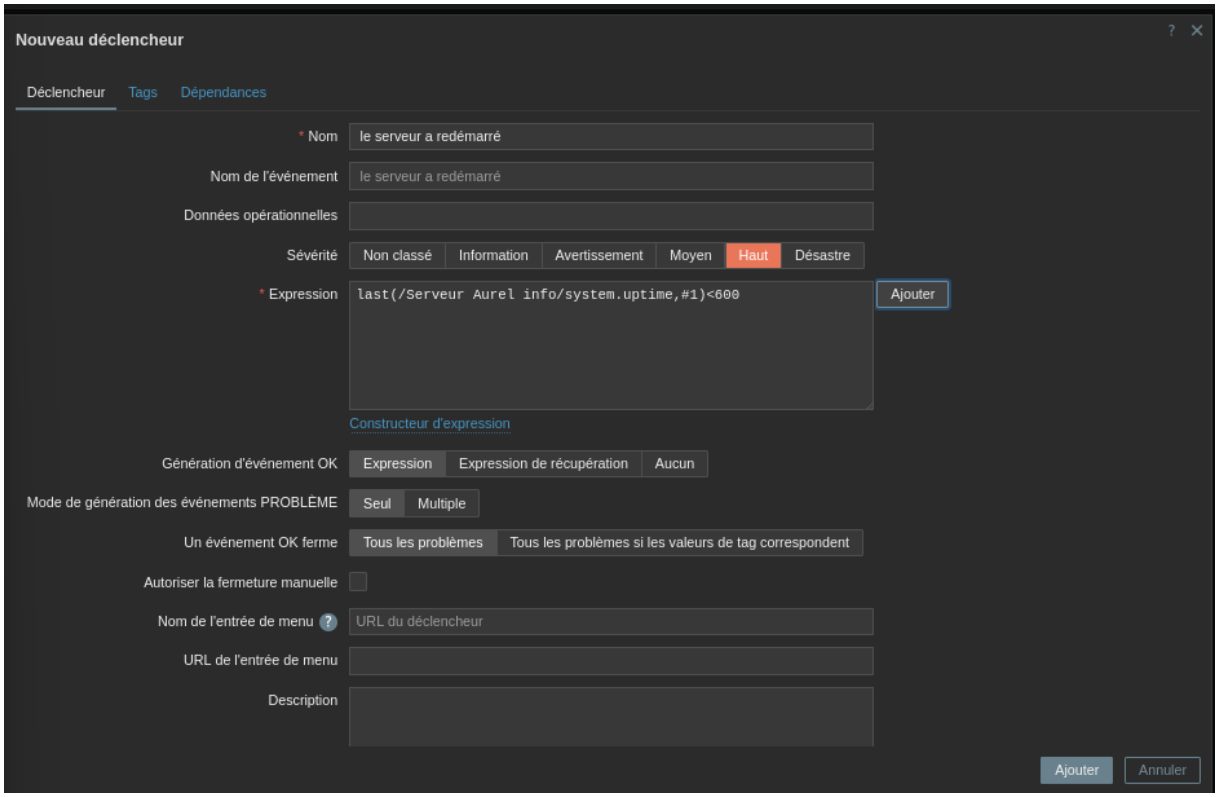
b) W32Time



Création d'une alerte pour le temps de disponibilité (Uptime)

Un **déclencheur** a été mis en place pour **alerter si le serveur a redémarré récemment**.
Grâce à l'élément **system.uptime**, nous pouvons détecter une **durée de fonctionnement inférieure à 10 minutes**, ce qui indiquerait un **redémarrage non planifié** ou un **incident système**.

Comme pour les autres déclencheurs, nous allons dans **Collecte de Données**, puis **Hôtes**, nous sélectionnons **Serveur Aurel Info**, et nous créons un **nouveau déclencheur** spécifique à l'uptime.



VI - Tableau de surveillance

Afin de **visualiser facilement l'état de notre infrastructure**, nous allons créer un **tableau de surveillance** (ou **dashboard**) dans **Zabbix**.
Ce tableau affichera en **temps réel** toutes les **alertes critiques** générées par les déclencheurs que nous avons configurés précédemment.

Nous allons procéder ainsi :

Nous allons dans le menu **"Tableaux de bord"** depuis l'interface web de Zabbix.

Nous créons un **nouveau tableau de bord**, que nous **nommons** selon notre choix (exemple : "Supervision Serveur Aurel Info").

Nous **personnalisons** ce tableau en ajoutant différents **widgets** :

Valeurs pour afficher des métriques en temps réel (CPU, RAM, Disque, etc.),

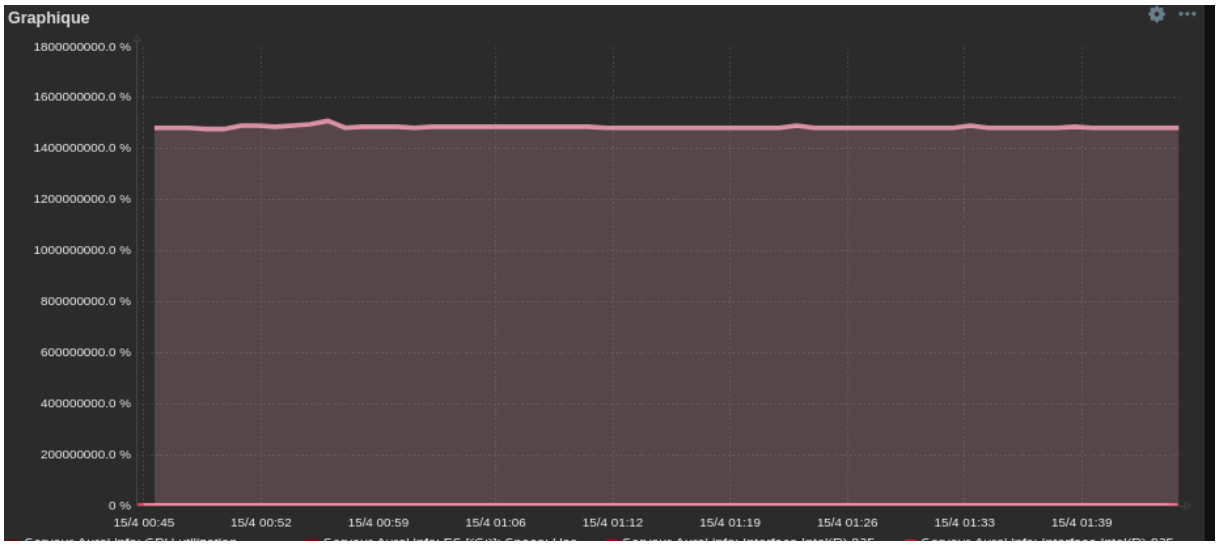
Graphiques pour visualiser l'évolution des ressources,

Alertes pour surveiller les incidents critiques.

Lors de la configuration des widgets, nous **sélectionnons l'hôte** (**Serveur Aurel Info**) et les **éléments à surveiller** (CPU, RAM, services, réseau...).

Nous **organisons** l'affichage par **glisser-déposer** pour rendre la présentation claire et lisible.

Enfin, nous **enregistrons** le tableau de bord pour permettre une **visualisation centralisée** et **rapide** de l'état du serveur.



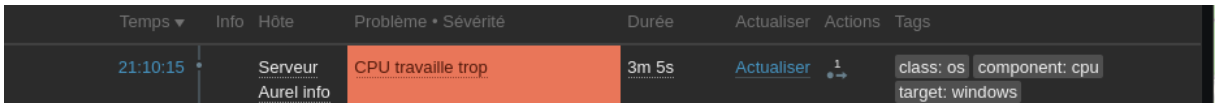
VII - tests

Nous allons maintenant **effectuer différents tests** afin de **vérifier que tous les déclencheurs configurés fonctionnent correctement** et que **les alertes sont bien remontées** dans l'interface Zabbix.

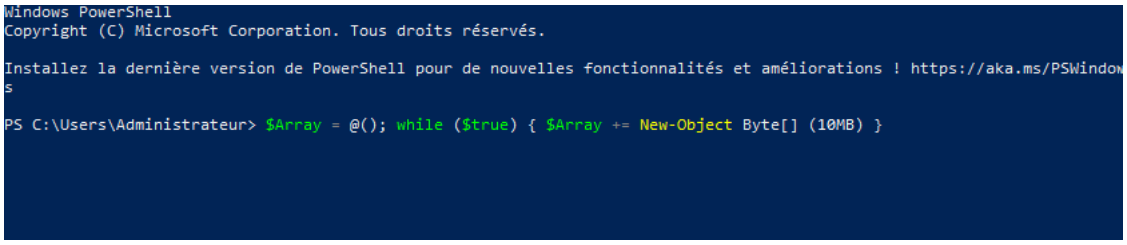
1) **Test du CPU (Utilisation CPU élevée) : for ($i=0; $i -lt 10; $i++) { Start-Job { while ($true) {} } }**

```
PS C:\Users\Administrateur> for ($i=0; $i -lt 4; $i++) { Start-Job { while ($true) {} } }
>>

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
9      Job9            BackgroundJob   Running       True            localhost             while ($true) {}
11     Job11           BackgroundJob   Running       True            localhost             while ($true) {}
13     Job13           BackgroundJob   Running       True            localhost             while ($true) {}
15     Job15           BackgroundJob   Running       True            localhost             while ($true) {}


PS C:\Users\Administrateur>
```

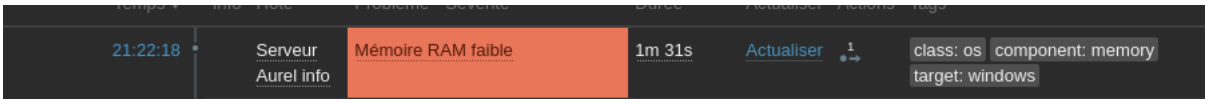
Nous recevons bien une alerte dans Zabbix, confirmant que le déclencheur fonctionne correctement et que la supervision est opérationnelle.



2) **Test de la RAM (Utilisation RAM élevée) : $Array = @(); while ($true) { $Array += New-Object Byte[] (10MB) }**



Nous recevons bien une alerte dans Zabbix, confirmant que le déclencheur fonctionne correctement et que la supervision est opérationnelle.



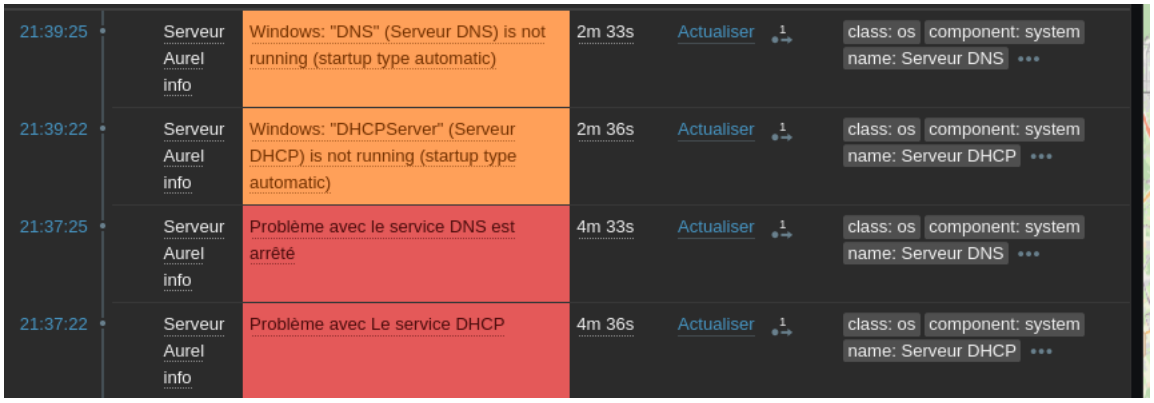
3) **Test des services critiques (DHCP, DNS)**

Arrêter le service DHCP : Stop-Service -Name dhcpserver

Arrêter le service DNS : Stop-Service -Name dns

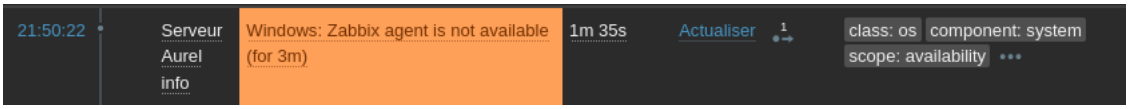
Le service **ntds (Active Directory Domain Services)** ne peut **pas être arrêté** directement avec Stop-Service, car il est **intégré au système** et **protégé par Windows** pour éviter l'interruption du contrôleur de domaine.

Nous recevons bien une alerte dans Zabbix, confirmant que le déclencheur fonctionne correctement et que la supervision est opérationnelle.



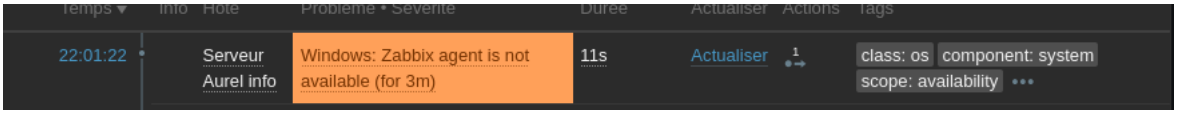
4) **Test du service Zabbix Agent : Stop-Service -Name "Zabbix Agent"**

Nous recevons bien une alerte dans Zabbix, confirmant que le déclencheur fonctionne correctement et que la supervision est opérationnelle.



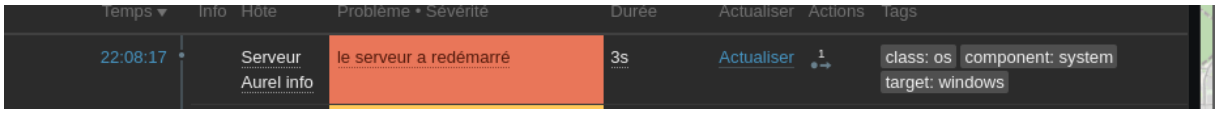
5) **Test de la perte de communication réseau (Ping) : Disable-NetAdapter -Name "Ethernet" -Confirm:$false**

Nous recevons bien une alerte dans Zabbix, confirmant que le déclencheur fonctionne correctement et que la supervision est opérationnelle.



Test de l'uptime (redémarrage) : Restart-Computer

Nous recevons bien une alerte dans Zabbix, confirmant que le déclencheur fonctionne correctement et que la supervision est opérationnelle.



VIII- Conclusion

Face aux difficultés rencontrées par **Aurel Informatique** pour superviser efficacement ses infrastructures informatiques, notamment les **retards dans la détection des incidents** et les **risques accrus de sécurité**, la solution mise en place avec **Zabbix sur un serveur Debian 12** s'est révélée pleinement adaptée.

La centralisation de la supervision permet désormais de :

- **Détecter rapidement** les anomalies sur les serveurs, réseaux et services critiques,
- **Réagir plus efficacement** face aux incidents, grâce aux alertes automatiques,
- **Renforcer la sécurité** en anticipant les pannes et les défaillances,
- **Gagner du temps** en automatisant la surveillance, ce qui allège la gestion quotidienne de l'infrastructure.

Grâce à cette solution, **Aurel Informatique** dispose aujourd'hui d'un **système de supervision fiable, proactif et évolutif**, contribuant ainsi à **sécuriser son activité** et à **améliorer la disponibilité de ses services**.