

Supervision Zabbix avec Active Directory et Authentification LDAP/LDAPS

 Windows Server 2025

ZABBIX

Réalisé par : **SOULAIMAN Rayane**

Administrateur système, réseau et sécurité

Table des matières

Table des matières	2
Introduction	4
Partie 1 — Installation de Zabbix sur Debian 12.....	5
1.1 Prérequis : configuration de l'adresse IP statique	5
1.2 Mise à jour du système	5
1.3 Ajout du dépôt officiel Zabbix 7.4	6
1.4 Installation des composants Zabbix	7
1.5 Installation et configuration de MariaDB.....	7
1.6 Import du schéma de base de données	8
1.7 Configuration du fichier de service Zabbix	9
1.8 Démarrage des services et vérification	10
1.9 Finalisation via l'interface web.....	11
Partie 2 — Supervision du serveur Windows Server	13
2.1 Installation de l'agent Zabbix sur Windows Server.....	13
2.2 Autorisation de l'agent dans le pare-feu Windows	13
2.3 Déclaration de l'hôte Windows dans Zabbix	14
Paramètres de l'hôte.....	14
Validation de la connexion.....	15
Partie 3 — Déploiement de l'Active Directory	18
3.1 Installation du rôle AD DS	18
3.2 Création du domaine rayane.local.....	18
3.3 Création du compte de liaison Zabbix.....	19
Partie 4 — Configuration de l'authentification LDAP	22
4.1 Accès aux paramètres d'authentification.....	22
4.2 Configuration du serveur Active Directory	22
4.3 Test et validation de la connexion LDAP	23
Partie 5 — Sécurisation avec LDAPS et Autorité de Certification	25
5.1 Préparation du serveur SRV-PKI-01	25
Configuration IP	25
Jonction au domaine.....	25
5.2 Installation du rôle ADCS	26
5.3 Configuration de l'Autorité de Certification	27
5.4 Export du certificat depuis SRV-PKI-01	29
5.5 Transfert et installation du certificat sur Zabbix.....	31
5.6 Test du canal LDAPS.....	33
5.7 Activation de LDAPS dans l'interface Zabbix	34

Partie 6 — Analyse de sécurité avec Wireshark.....	37
6.1 Comparaison LDAP vs LDAPS	37
6.2 Conclusion de l'analyse.....	38
Conclusion	39

Introduction

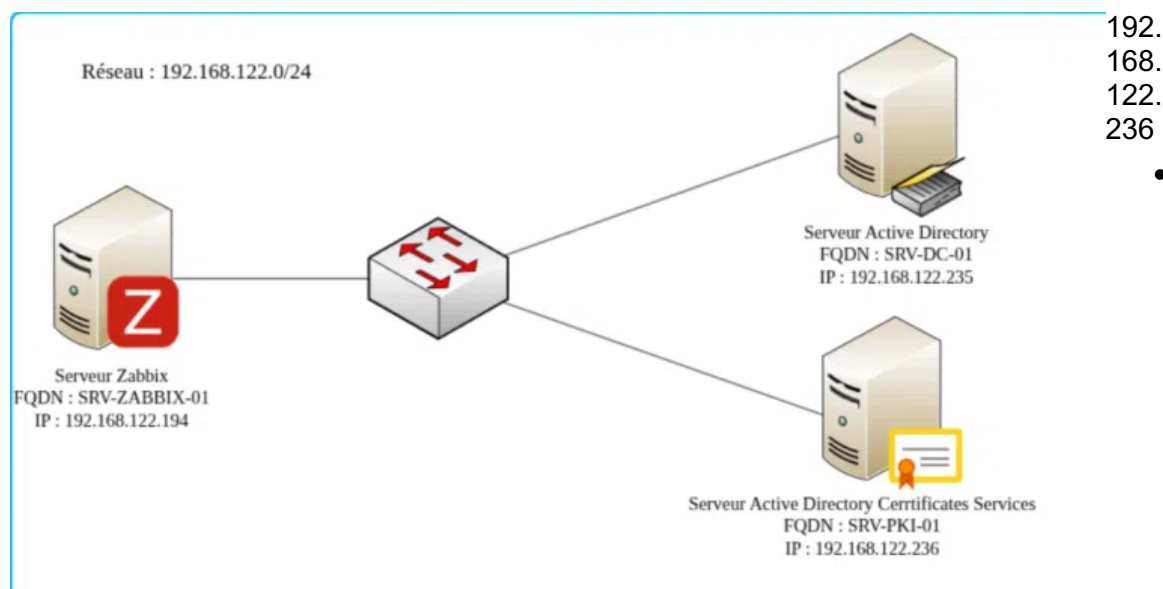
Ce document constitue le compte-rendu technique d'un travail pratique portant sur la mise en place d'une solution de supervision réseau Zabbix 7.4 intégrée à un annuaire Active Directory. Il s'inscrit dans le cadre de la formation BTS Services Informatiques aux Organisations (SIO) et illustre des compétences en administration système, en sécurité réseau et en gestion d'identités.

L'objectif de ce TP est triple :

- Déployer et configurer Zabbix 7.4 sur un serveur Debian 12 avec une base de données MariaDB et un serveur web Apache.
- Intégrer un agent Zabbix sur un serveur Windows Server membre d'un domaine Active Directory pour en assurer la supervision.
- Configurer l'authentification LDAP puis LDAPS (version chiffrée) afin que les comptes Active Directory puissent se connecter à l'interface web de Zabbix de manière sécurisée.

L'infrastructure mise en œuvre repose sur un environnement virtualisé composé des éléments suivants :

- Serveur Zabbix : VM Debian 12 — IP 192.168.122.194
- Contrôleur de domaine : VM Windows Server (SRV-DC-01) — IP 192.168.122.235
- Serveur PKI (Autorité de Certification) : VM Windows Server (SRV-PKI-01) — IP 192.168.122.236



Partie 1 — Installation de Zabbix sur Debian 12

Cette première partie décrit l'installation complète du serveur de supervision Zabbix 7.4, incluant la configuration de l'adresse IP statique, l'installation des paquets nécessaires, la création de la base de données et la finalisation via l'interface web.

1.1 Prérequis : configuration de l'adresse IP statique

Avant toute installation de logiciel, il convient d'attribuer une adresse IP fixe au serveur afin de garantir la cohérence des communications réseau avec les autres machines de l'infrastructure.



```
zabbix@debian:~$ nano /etc/network/interfaces
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

# The loopback network interface
auto ens33
iface ens33 inet static
    address 192.168.122.194
    netmask 255.255.0
    gateway 192.168.122.2
```

Figure 1 — Configuration de l'adresse IP statique sur Debian 12

1.2 Mise à jour du système

La première opération consiste à s'assurer que le système dispose des dernières mises à jour de sécurité et de stabilité avant d'installer de nouveaux paquets.

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

```

zabbix@debian:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for zabbix:
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease
Get:3 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Fetched 55.4 kB in 1s (38.8 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
zabbix@debian:~$

```

1. 3 Ajout du dépôt officiel Zabbix 7.4

Zabbix ne figure pas dans les dépôts standard de Debian. Il est nécessaire d'ajouter le dépôt officiel du projet pour accéder à la version 7.4.

```

wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian12_all.deb
sudo dpkg -i zabbix-release_latest_7.4+debian12_all.deb
sudo apt update

```

```

zabbix@debian:~$ wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian12_all.deb
zabbix@debian:~$ sudo dpkg -i zabbix-release_latest_7.4+debian12_all.deb
zabbix@debian:~$ sudo apt update
--2026-05-04 06:08:15-- https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian12_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7132 (7.0K) [application/octet-stream]
Saving to: 'zabbix-release_latest_7.4+debian12_all.deb'

zabbix-release_late 100%[=====] 6.96K --.-KB/s in 0s

2026-05-04 06:08:16 (53.2 MB/s) - 'zabbix-release_latest_7.4+debian12_all.deb' saved [7132/7132]

Selecting previously unselected package zabbix-release.
(Reading database ... 35056 files and directories currently installed.)
Preparing to unpack zabbix-release_latest_7.4+debian12_all.deb ...
Unpacking zabbix-release (1:7.4-1+debian12) ...
Setting up zabbix-release (1:7.4-1+debian12) ...
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Get:4 https://repo.zabbix.com/zabbix/7.4/release/debian bookworm InRelease [2,459 B]
Get:5 https://repo.zabbix.com/zabbix-tools/debian-ubuntu bookworm InRelease [2,476 B]
Get:6 https://repo.zabbix.com/zabbix/7.4/stable/debian bookworm InRelease [4,663 B]
Get:7 https://repo.zabbix.com/zabbix/7.4/release/debian bookworm/main Sources [932 B]
Get:8 https://repo.zabbix.com/zabbix/7.4/release/debian bookworm/main all Packages [628 B]
Get:9 https://repo.zabbix.com/zabbix-tools/debian-ubuntu bookworm/main Sources [1,571 B]
Get:10 https://repo.zabbix.com/zabbix-tools/debian-ubuntu bookworm/main all Packages [969 B]
Get:11 https://repo.zabbix.com/zabbix/7.4/stable/debian bookworm/main Sources [19.8 kB]
Get:12 https://repo.zabbix.com/zabbix/7.4/stable/debian bookworm/main all Packages [7,299 B]
Get:13 https://repo.zabbix.com/zabbix/7.4/stable/debian bookworm/main amd64 Packages [39.1 kB]
Fetched 79.9 kB in 3s (27.7 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
zabbix@debian:~$

```

Figure 2 — Ajout et validation du dépôt officiel Zabbix

1.4 Installation des composants Zabbix

L'installation inclut le serveur Zabbix, le frontal web PHP, le module de configuration Apache, les scripts SQL d'initialisation et l'agent local.

```
sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf
zabbix-sql-scripts zabbix-agent -y
```

```
zabbix@debian: ~
Package apache2 is not configured yet. Will defer actions by package libapache2-mod-php8.2.
Creating config file /etc/php/8.2/apache2/php.ini with new version
No module matches
Setting up libapache2-mod-php (2:8.2+93) ...
Setting up php8.2-gd (8.2.30-1-deb12u1) ...

Creating config file /etc/php/8.2/mods-available/gd.ini with new version
Setting up php-gd (2:8.2+93) ...
Setting up zabbix-server-mysql (1:7.4.9-1+debian12) ...
Setting up apache2 (2.4.66-1-deb12u1) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
info: Switch to mpm prefork for package libapache2-mod-php8.2
Module mpm_event disabled.
Enabling module mpm_prefork.
info: Executing deferred 'a2enmod php8.2' for package libapache2-mod-php8.2
Enabling module php8.2.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Setting up zabbix-frontend-php (1:7.4.9-1+debian12) ...
update-alternatives: using /usr/share/fonts/truetype/dejavu/DejaVuSans.ttf to provide /usr/share/zabbix/ui/assets/fonts/graphfont.ttf (zabbix-frontend-font) in auto mode
Setting up zabbix-apache-conf (1:7.4.9-1+debian12) ...
Enabling conf zabbix.
To activate the new configuration, you need to run:
  systemctl reload apache2
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u13) ...
Processing triggers for php8.2-cli (8.2.30-1-deb12u1) ...
Processing triggers for libapache2-mod-php8.2 (8.2.30-1-deb12u1) ...
zabbix@debian:~$
```

Figure 3 — Installation des paquets Zabbix

1.5 Installation et configuration de MariaDB

Zabbix utilise une base de données relationnelle pour stocker l'ensemble de ses données de supervision. MariaDB est ici utilisé comme SGBD.

```
sudo apt install mariadb-server -y
```

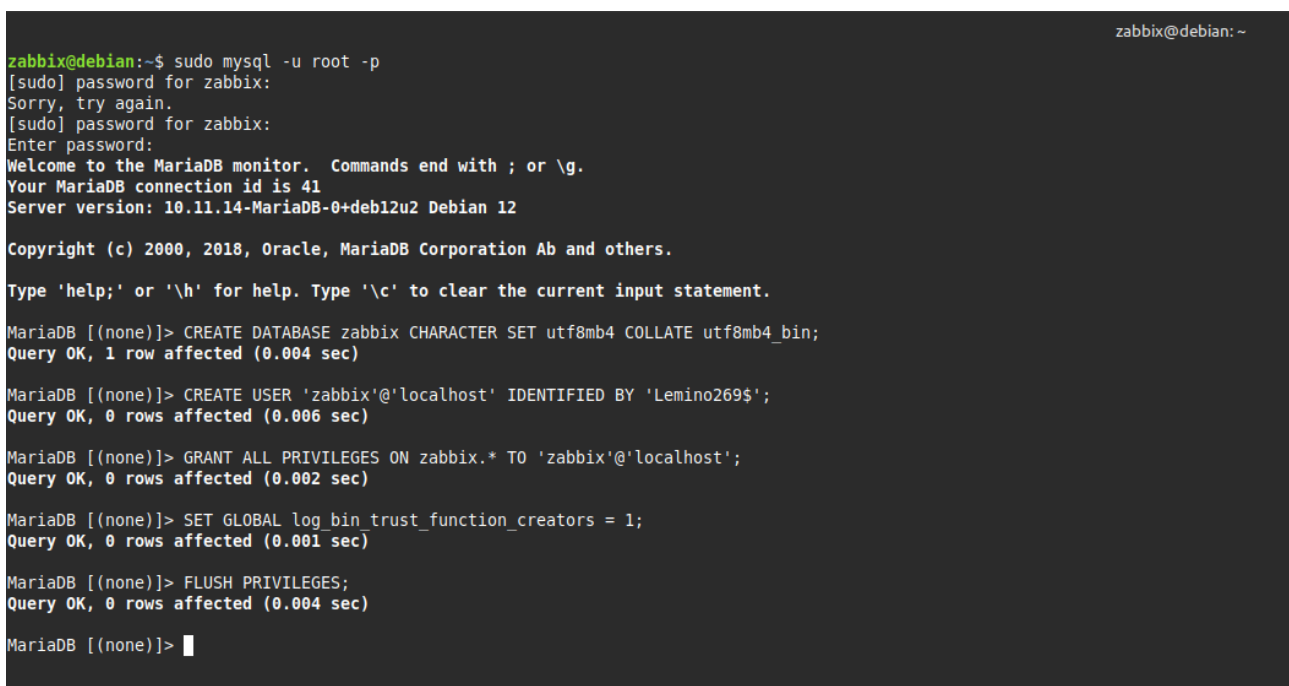
```
sudo systemctl start mariadb && sudo systemctl enable mariadb
```

Sécurisation de l'installation (répondre Y à toutes les questions et définir un mot de passe root robuste) :

```
sudo mysql_secure_installation
```

Création de la base de données dédiée à Zabbix et de son utilisateur :

```
sudo mysql -u root -p
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'MotDePasseForte';
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
SET GLOBAL log_bin_trust_function_creators = 1;
FLUSH PRIVILEGES; EXIT;
```



```
zabbix@debian:~$ sudo mysql -u root -p
[sudo] password for zabbix:
Sorry, try again.
[sudo] password for zabbix:
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.14-MariaDB-0+deb12u2 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;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]> CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Lemino269$';
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> SET GLOBAL log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> █
```

Figure 4 — Création de la base de données Zabbix dans MariaDB

1.6 Import du schéma de base de données

L'initialisation de la structure de la base de données s'effectue via le script SQL fourni par le paquet Zabbix.

```
sudo zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql --
default-character-set=utf8mb4 -u zabbix -p zabbix
```

Remarque : après l'import, désactiver l'option temporaire `log_bin_trust_function_creators` :

```
sudo mysql -u root -p -e "SET GLOBAL log_bin_trust_function_creators = 0;"
```

```
zabbix@debian:~$ sudo zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -u zabbix -p zabbix
Enter password:
zabbix@debian:~$
```



Figure 5 — Import du schéma de base de données

1.7 Configuration du fichier de service Zabbix

Le fichier de configuration principal du serveur Zabbix doit être mis à jour avec le mot de passe de la base de données créée précédemment.

```
sudo nano /etc/zabbix/zabbix_server.conf
```

Modifier la ligne : `DBPassword=<mot_de_passe_défini_précédemment>`

```
zabbix@debian: ~  
GNU nano 7.2 /etc/zabbix/zabbix_server.conf *  
# DBHost=localhost  
### Option: DBName  
# Database name.  
#  
# Mandatory: yes  
# Default:  
# DBName=  
DBName=zabbix  
### Option: DBSchema  
# Schema name. Used for PostgreSQL.  
#  
# 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=Lemino269$  
### Option: DBSocket  
# Path to MySQL socket.  
#  
# Mandatory: no  
# Default:  
# DBSocket=  
### Option: DBPort  
# Database port when not using local socket.  
#  
# Mandatory: no  
# Range: 1024-65535  
# Default for MySQL: 3306  
# Default for PostgreSQL: 5432  
# DBPort=  
### Option: AllowUnsupportedDBVersions  
# Allow server to work with unsupported database versions.  
# 0 - do not allow  
# 1 - allow  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^/_ Go To Line  M-E Redo      M-6 Copy  
[Icons]
```

Figure 6 — Configuration du mot de passe de la base de données

1.8 Démarrage des services et vérification

Une fois la configuration terminée, les services sont démarrés et activés au démarrage du système.

```
sudo systemctl restart zabbix-server zabbix-agent apache2  
sudo systemctl enable zabbix-server zabbix-agent apache2  
sudo systemctl status zabbix-server
```

```

zabbix@debian:~$ sudo systemctl restart zabbix-server zabbix-agent apache2
sudo systemctl enable zabbix-server zabbix-agent apache2
[sudo] password for zabbix:
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/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.service → /lib/systemd/system/zabbix-server.service.
zabbix@debian:~$

```

```

zabbix@debian:~$ systemctl status zabbix-server
● zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; preset: enabled)
   Active: active (running) since Mon 2026-05-04 07:45:35 EDT; 1min 4s ago
     Main PID: 15008 (zabbix_server)
        Tasks: 77 (Limit: 2250)
      Memory: 71.8M
         CPU: 2.291s
    CGroup: /system.slice/zabbix-server.service
            └─15008 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
                └─15026 /usr/sbin/zabbix_server: ha manager*
                    └─15030 /usr/sbin/zabbix_server: service manager #1 [processed 0 events, updated 0 event tags, deleted 0 problems, synced 0 service updates, idle 5.006146 sec during 5.006340 sec]*
                        └─15031 /usr/sbin/zabbix_server: configuration syncer [synced configuration in 0.070644 sec, idle 10 sec]*
                            └─15084 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.007339 sec during 5.007726 sec]*
                                └─15085 /usr/sbin/zabbix_server: alerter #1 started*
                                    └─15087 /usr/sbin/zabbix_server: alerter #2 started*
                                        └─15088 /usr/sbin/zabbix_server: alerter #3 started*
                                            └─15089 /usr/sbin/zabbix_server: preprocessing manager #1 [queued 2, processed 3 values, idle 5.498590 sec during 5.508144 sec]*
                                                └─15090 /usr/sbin/zabbix_server: lld manager #1 [processed 0 LLD rules, idle 5.985384sec during 5.985688 sec]*
                                                    └─15091 /usr/sbin/zabbix_server: lld worker #1 [processed 1 LLD rules, idle 21.004492 sec during 21.029315 sec]*
                                                        └─15092 /usr/sbin/zabbix_server: lld worker #2 [processed 1 LLD rules, idle 20.843000 sec during 20.877940 sec]*
                                                            └─15093 /usr/sbin/zabbix_server: housekeeper [startup idle for 30 minutes]*
                                                                └─15094 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.001532 sec, idle 59 sec]*
                                                                    └─15098 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.000852 sec, idle 5 sec]*
                                                                        └─15099 /usr/sbin/zabbix_server: browser poller #1 [got 0 values in 0.000053 sec, idle 5 sec]*
                                                                            └─15106 /usr/sbin/zabbix_server: discovery manager #1 [processing 0 rules, 0 unsaved checks]*
                                                                                └─15108 /usr/sbin/zabbix_server: history syncer #1 [processed 1 values, 1+0 triggers in 0.004887 (0.004,0.000,0.000,0.000,0.000) sec, idle 1 sec]*
                                                                                    └─15116 /usr/sbin/zabbix_server: history syncer #2 [processed 0 values, 0+0 triggers in 0.000059 (0.000,0.000,0.000,0.000,0.000) sec, idle 1 sec]*
                                                                                        └─15117 /usr/sbin/zabbix_server: history syncer #3 [processed 0 values, 0+0 triggers in 0.000093 (0.000,0.000,0.000,0.000,0.000) sec, idle 1 sec]*
                                                                                            └─15123 /usr/sbin/zabbix_server: history syncer #4 [processed 0 values, 0+0 triggers in 0.000065 (0.000,0.000,0.000,0.000,0.000) sec, idle 1 sec]*
                                                                                                └─15124 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.002735 sec, idle 3 sec]*
                                                                                                    └─15125 /usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 proxies in 0.000046 sec, idle 5 sec]*
                                                                                                        └─15127 /usr/sbin/zabbix_server: self-monitoring [processed data in 0.000057 sec, idle 1 sec]*
                                                                                                            └─15128 /usr/sbin/zabbix_server: task manager [processed 0 task(s) in 0.001134 sec, idle 5 sec]*
                                                                                                                └─15130 /usr/sbin/zabbix_server: poller #1 [got 0 values in 0.000113 sec, idle 5 sec]*
                                                                                                                    └─15133 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000036 sec, idle 5 sec]*
                                                                                                                        └─15135 /usr/sbin/zabbix_server: poller #3 [got 0 values in 0.000055 sec, idle 5 sec]*
                                                                                                                            └─15136 /usr/sbin/zabbix_server: poller #4 [got 0 values in 0.000109 sec, idle 5 sec]*
                                                                                                                                └─15137 /usr/sbin/zabbix_server: poller #5 [got 0 values in 0.000045 sec, idle 5 sec]*
                                                                                                                                    └─15138 /usr/sbin/zabbix_server: unreachable poller #1 [got 0 values in 0.000060 sec, idle 5 sec]*
                                                                                                                                        └─15139 /usr/sbin/zabbix_server: trapper #1 [processed data in 0.000242 sec, waiting for connection]*
                                                                                                                                            └─15140 /usr/sbin/zabbix_server: trapper #2 [processed data in 0.000696 sec, waiting for connection]*
                                                                                                                                                └─15141 /usr/sbin/zabbix_server: trapper #3 [processed data in 0.000000 sec, waiting for connection]*
                                                                                                                                                    └─15145 /usr/sbin/zabbix_server: trapper #4 [processed data in 0.000000 sec, waiting for connection]*
                                                                                                                                                        └─15146 /usr/sbin/zabbix_server: trapper #5 [processed data in 0.000000 sec, waiting for connection]*
                                                                                                                                                            └─15149 /usr/sbin/zabbix_server: icmp pingler #1 [got 0 values in 0.000063 sec, idle 5 sec]*
                                                                                                                                                                └─15151 /usr/sbin/zabbix_server: alert syncer [queued 0 alerts(s), flushed 0 result(s) in 0.000001 sec, idle 1 sec]*
                                                                                                                                                                    └─15155 /usr/sbin/zabbix_server: history poller #1 [got 0 values in 0.000058 sec, idle 5 sec]*
                                                                                                                                                                        └─15161 /usr/sbin/zabbix_server: history poller #2 [got 0 values in 0.000056 sec, idle 5 sec]*
                                                                                                                                                                            └─15168 /usr/sbin/zabbix_server: history poller #3 [got 0 values in 0.000113 sec, idle 5 sec]*
                                                                                                                                                                                └─15170 /usr/sbin/zabbix_server: history poller #4 [got 0 values in 0.000018 sec, idle 5 sec]*
                                                                                                                                                                                    └─15172 /usr/sbin/zabbix_server: history poller #5 [got 0 values in 0.000044 sec, idle 5 sec]*
                                                                                                                                                                                        └─15179 /usr/sbin/zabbix_server: availability manager #1 [queued 0, processed 0 values, idle 5.398220 sec during 5.398588 sec]*
                                                                                                                                                                                            └─15183 /usr/sbin/zabbix_server: trigger housekeeper [deleted 0 problems records in 0.003234 sec, idle for 60 second(s)]*
                                                                                                                                                                                                └─15184 /usr/sbin/zabbix_server: odbc poller #1 [got 0 values in 0.000053 sec, idle 5 sec]*
                                                                                                                                                                                                    └─15190 /usr/sbin/zabbix_server: http agent poller #1 [got 0 values, queued 0 in 5 sec, awaiting 0]*
                                                                                                                                                                                                        └─15192 /usr/sbin/zabbix_server: agent poller #1 [got 1 values, queued 1 in 5 sec, awaiting 0]*
                                                                                                                                                                                                            └─15193 /usr/sbin/zabbix_server: snmp poller #1 [got 0 values, queued 0 in 5 sec, awaiting 0]*
                                                                                                                                                                                                                └─15196 /usr/sbin/zabbix_server: configuration syncer worker [synced 0, updated 0 item names in 0.003888 sec, idle]*

```

Figure 7 — Vérification du statut des services

1.9 Finalisation via l'interface web

L'assistant de configuration web est accessible à l'adresse <http://192.168.122.194/zabbix>. Il guide l'administrateur à travers les étapes suivantes :

- Vérification des prérequis PHP (toutes les cases doivent être validées en vert).
- Paramétrage de la connexion à la base de données (hôte : localhost, base : zabbix, utilisateur : zabbix).
- Attribution d'un nom au serveur et sélection du fuseau horaire (Europe/Paris).
- Connexion initiale avec les identifiants par défaut : Login Admin / Mot de passe zabbix.

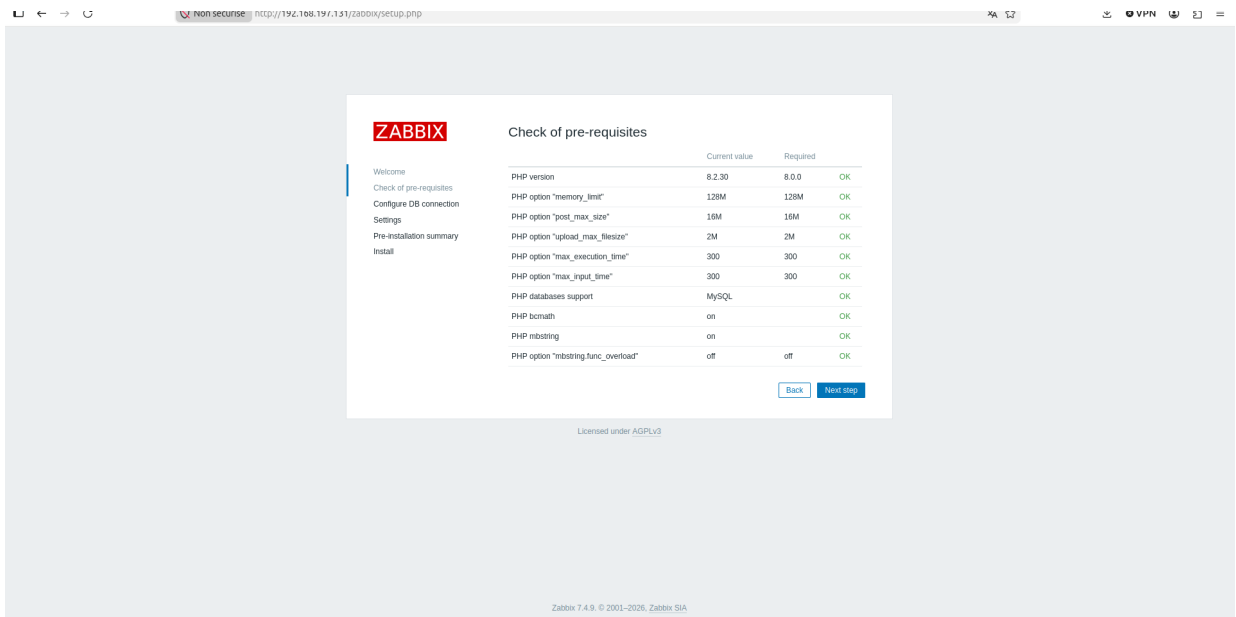


Figure 8 — Assistant de configuration web Zabbix

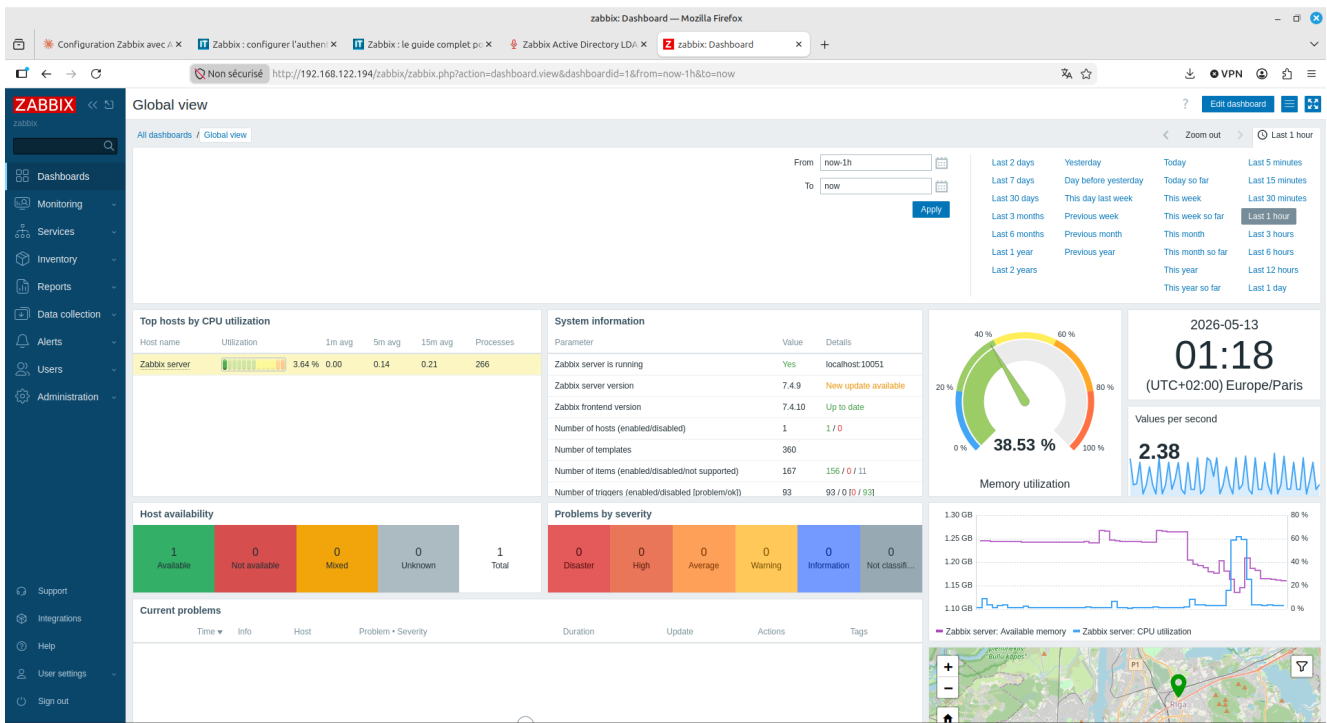


Figure 9 — Interface web Zabbix après connexion réussie

Partie 2 — Supervision du serveur Windows Server

Cette partie décrit l'installation de l'agent Zabbix sur le serveur Windows Server, sa déclaration dans l'interface de supervision et la configuration du pare-feu Windows pour permettre la communication.

2.1 Installation de l'agent Zabbix sur Windows Server

L'agent Zabbix Windows est disponible en téléchargement sur le site officiel du projet. Son installation nécessite de renseigner les informations suivantes lors de la procédure :

- Host name : nom exact du serveur Windows (SRV-DC-01 — respecter la casse).
- Zabbix server IP : adresse IP du serveur Zabbix (192.168.122.194).
- Listen port : laisser la valeur par défaut (10050).

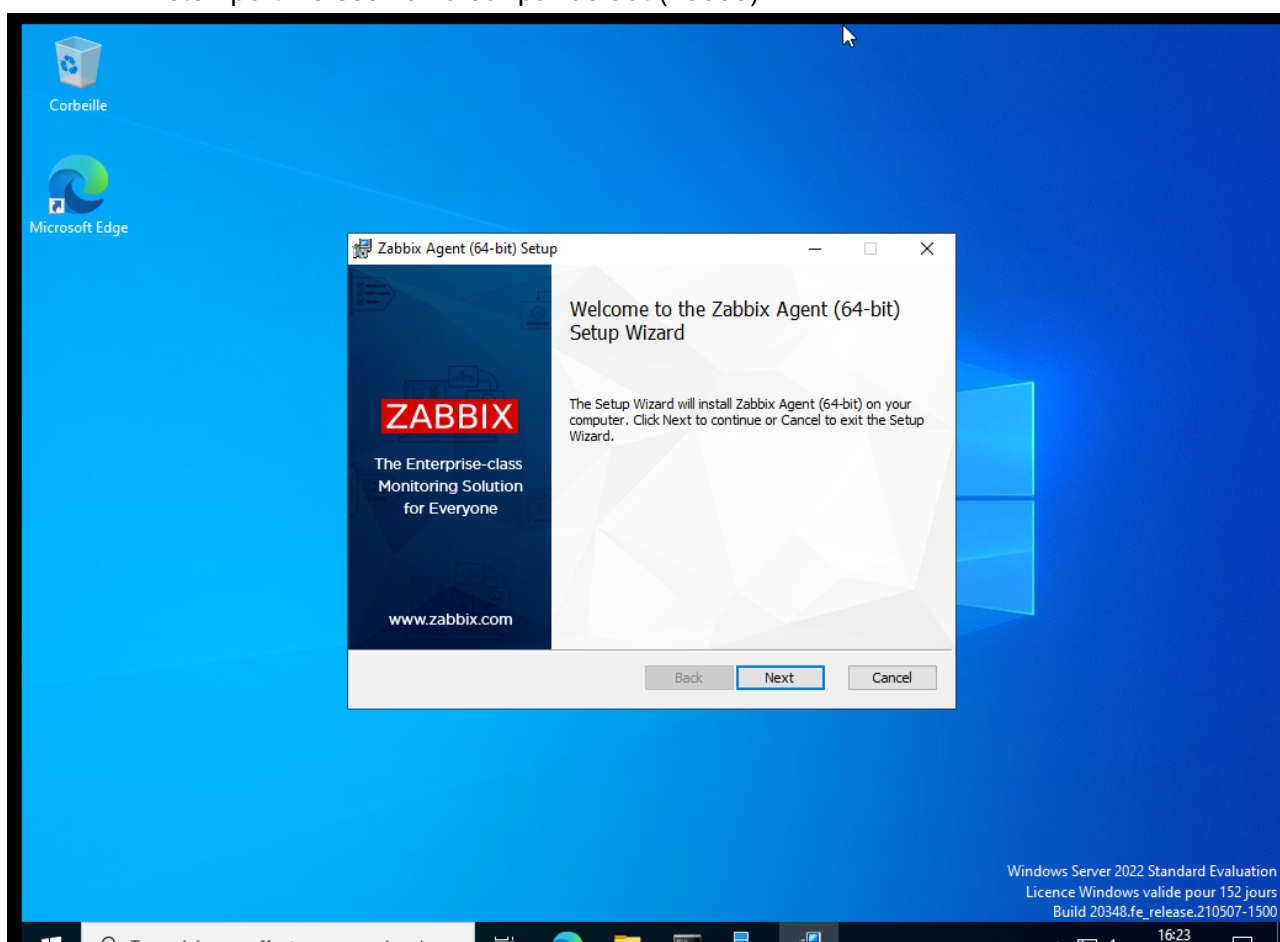


Figure 10 — Installation de l'agent Zabbix sur Windows Server

2.2 Autorisation de l'agent dans le pare-feu Windows

Windows bloque par défaut les connexions entrantes. Il est nécessaire d'ajouter une règle de pare-feu autorisant le trafic sur le port TCP 10050 utilisé par l'agent Zabbix.

Exécuter la commande suivante dans PowerShell (en tant qu'Administrateur) :

```
netsh advfirewall firewall add rule name="Zabbix" dir=in action=allow
protocol=TCP localport=10050
```

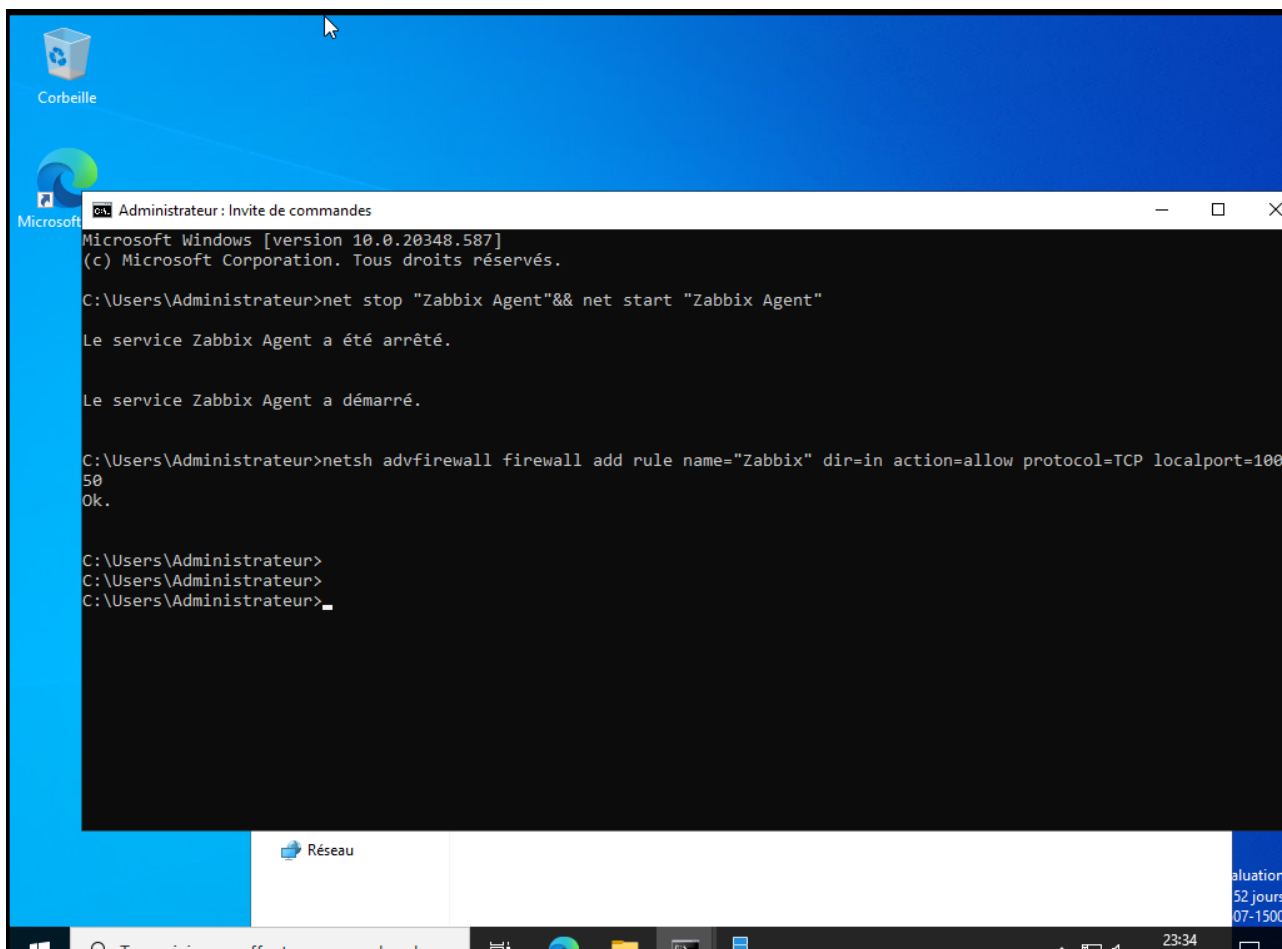


Figure 11 — Règle de pare-feu Windows pour l'agent Zabbix

2.3 Déclaration de l'hôte Windows dans Zabbix

Depuis l'interface web Zabbix, la déclaration de l'hôte s'effectue via le menu Data Collection > Hosts > Create host.

Paramètres de l'hôte

- Host name : SRV-DC-01 (respecter exactement la casse).
- Templates : Windows by Zabbix agent (ce modèle définit automatiquement les métriques à surveiller).
- Host groups : sélectionner un groupe existant ou créer un nouveau groupe (ex. : Serveurs Windows).
- Interfaces : ajouter une interface de type Agent avec l'IP 192.168.122.235 et le port 10050.

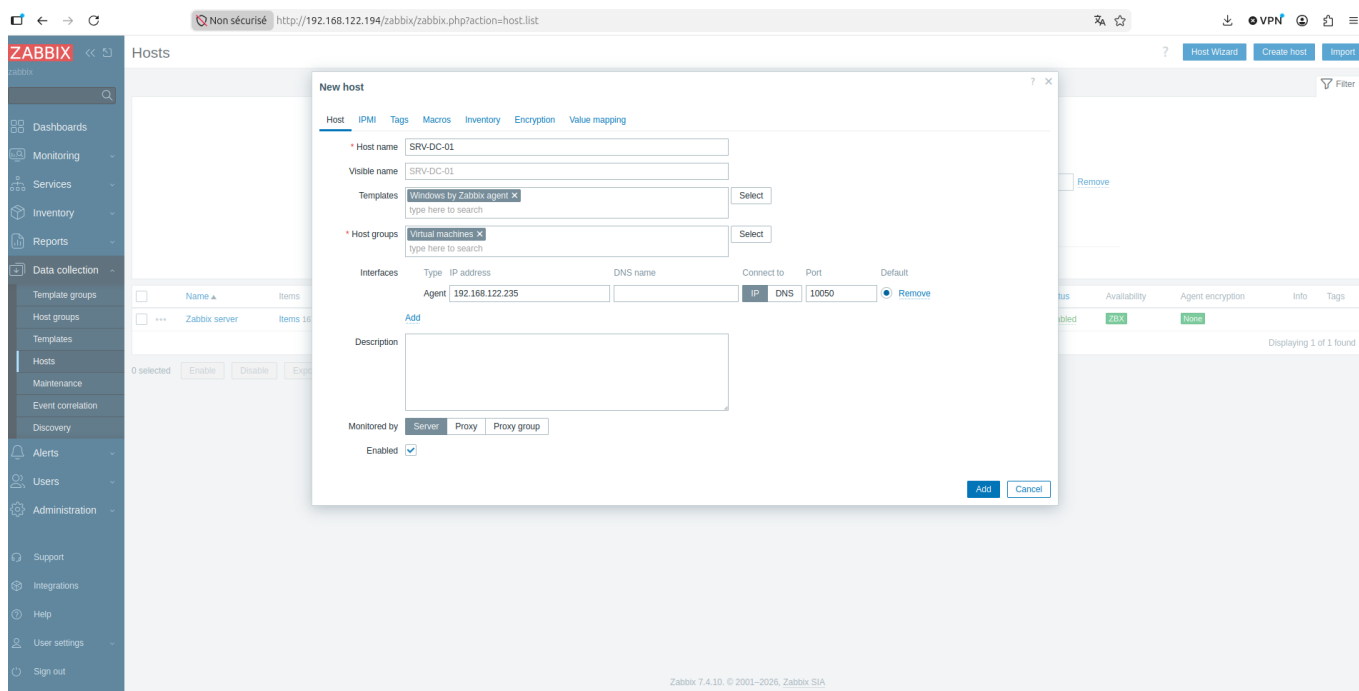


Figure 12 — Déclaration de l'hôte SRV-DC-01 dans Zabbix

Validation de la connexion

Après l'ajout, patienter une à deux minutes puis actualiser la page. L'icône ZBX verte indique que la communication entre le serveur Zabbix et l'agent Windows est opérationnelle et que la supervision a démarré.

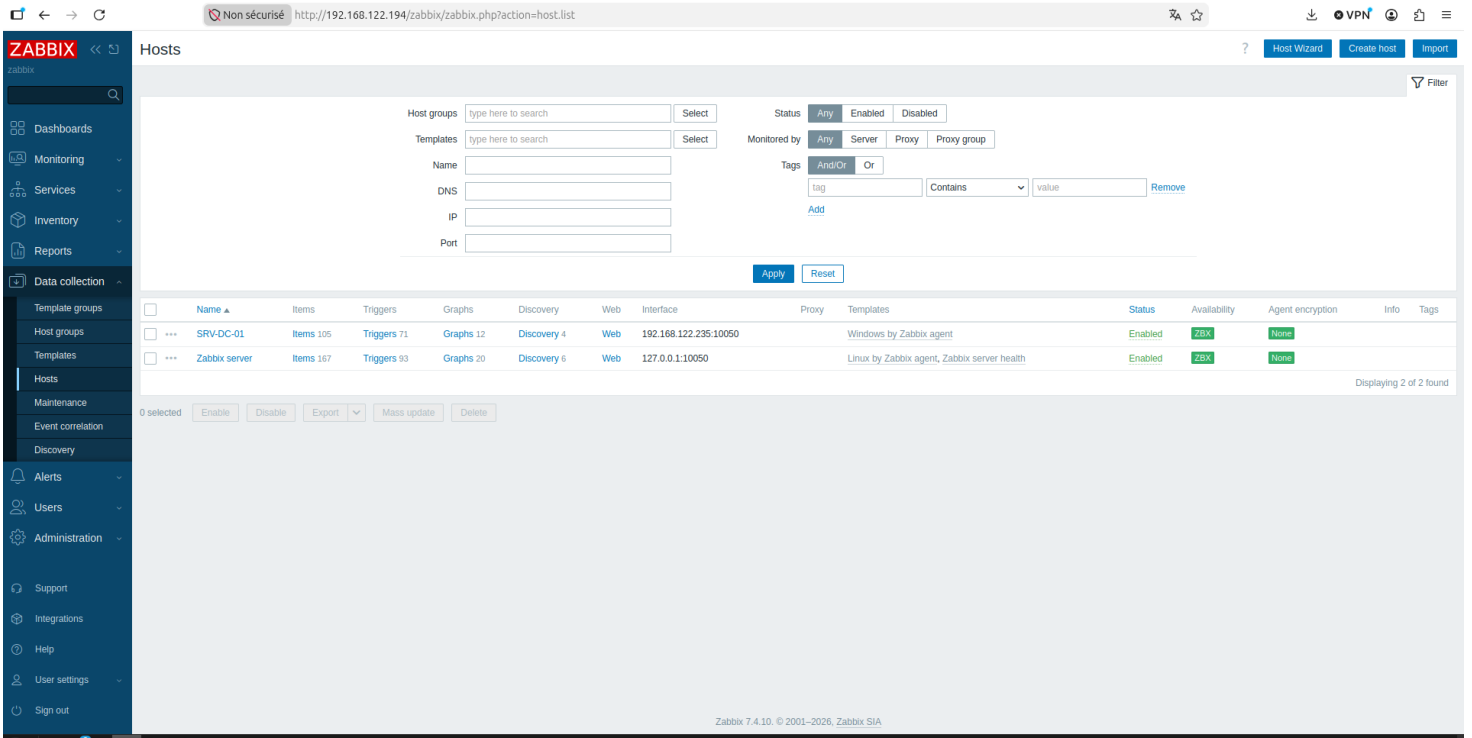


Figure 14 — Icône ZBX verte confirmant la supervision active

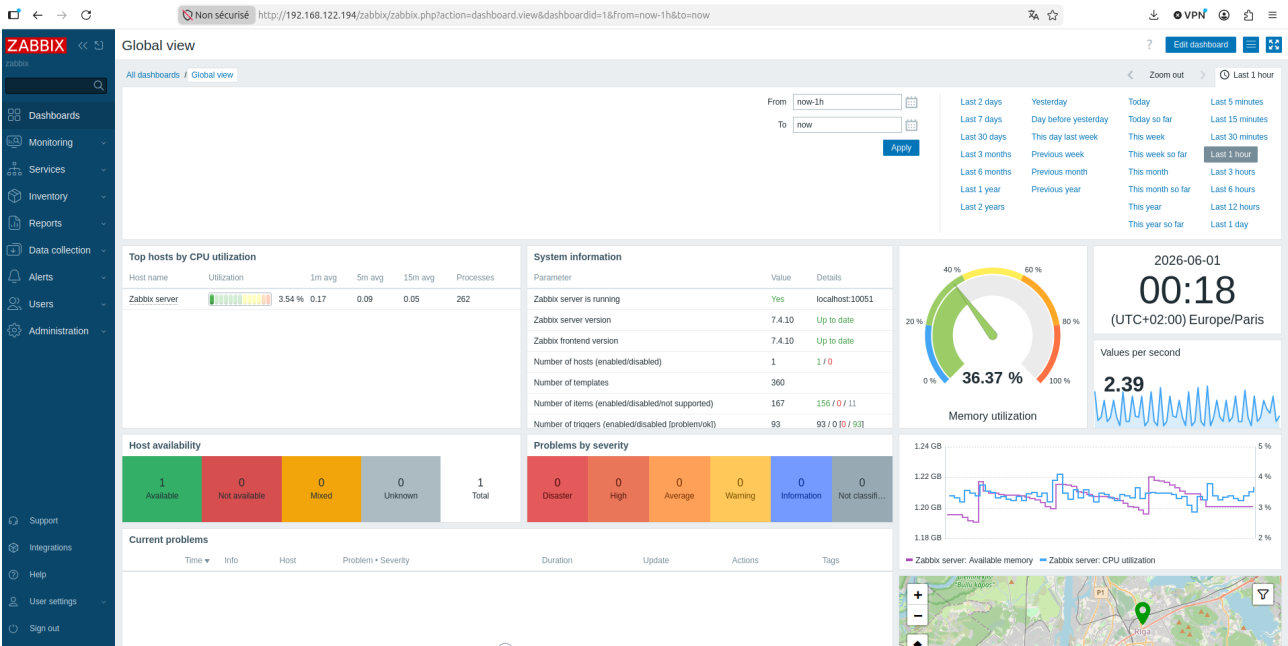
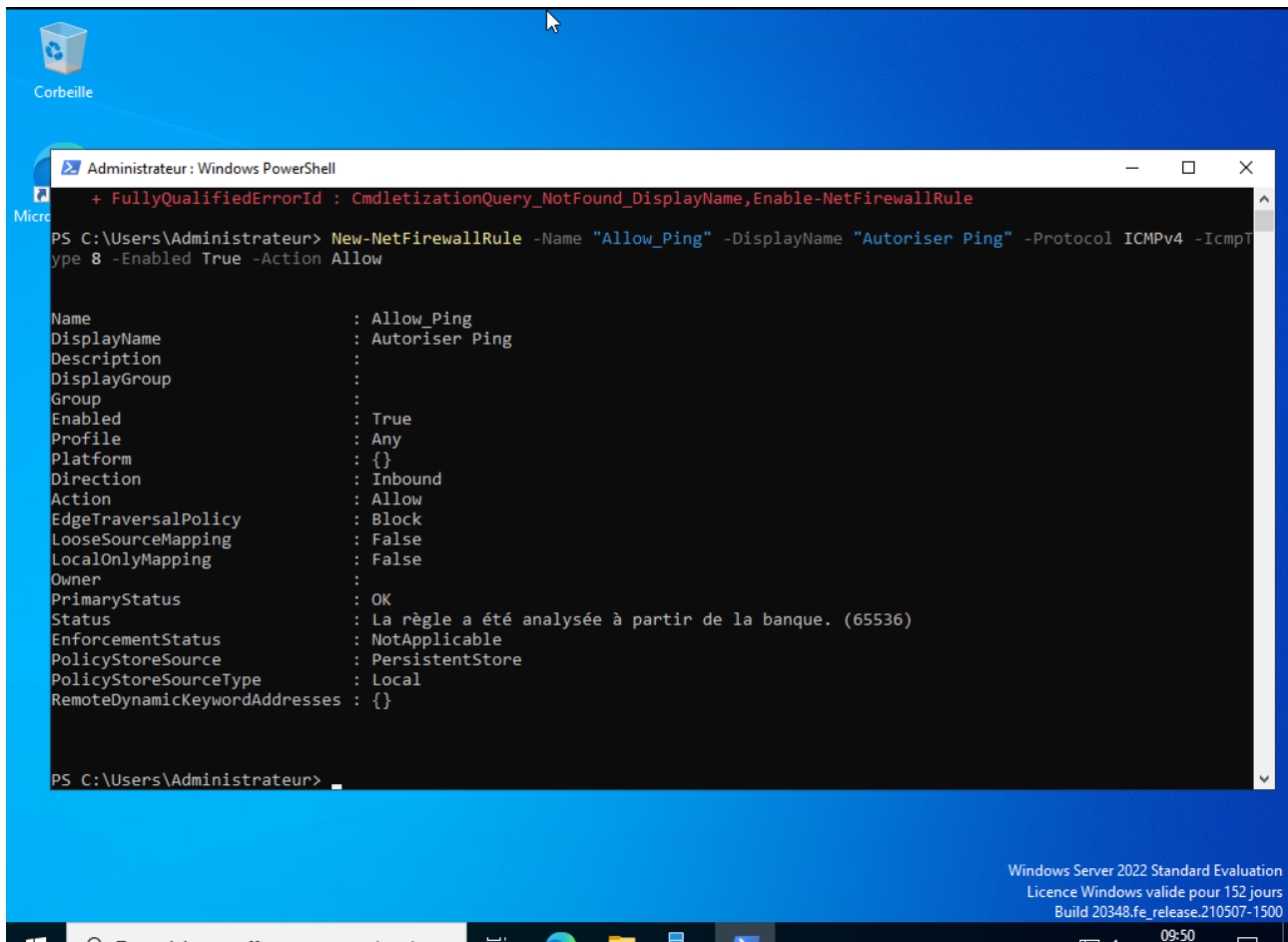


Figure 15 — Vue de la liste des hôtes supervisés



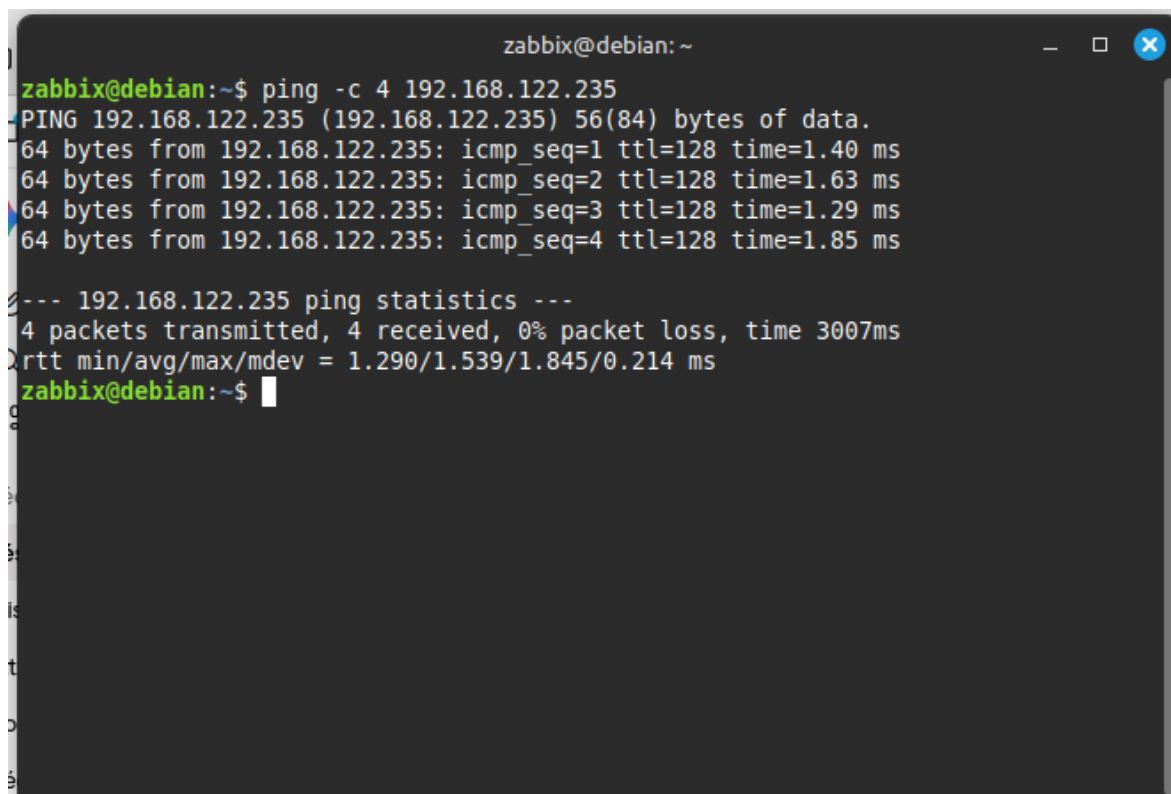
The screenshot shows a Windows PowerShell terminal window titled "Administrateur: Windows PowerShell". The command executed is `New-NetFirewallRule -Name "Allow_Ping" -DisplayName "Autoriser Ping" -Protocol ICMPv4 -IcmpType 8 -Enabled True -Action Allow`. The output displays the properties of the newly created rule, including its name, display name, protocol, and action.

```
PS C:\Users\Administrateur> New-NetFirewallRule -Name "Allow_Ping" -DisplayName "Autoriser Ping" -Protocol ICMPv4 -IcmpType 8 -Enabled True -Action Allow

Name                : Allow_Ping
DisplayName          : Autoriser Ping
Description          :
DisplayGroup        :
Group               :
Enabled             : True
Profile             : Any
Platform           : {}
Direction          : Inbound
Action              : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping  : False
LocalOnlyMapping    : False
Owner               :
PrimaryStatus       : OK
Status              : La règle a été analysée à partir de la banque. (65536)
EnforcementStatus   : NotApplicable
PolicyStoreSource   : PersistentStore
PolicyStoreSourceType : Local
RemoteDynamicKeywordAddresses : {}

PS C:\Users\Administrateur>
```

Windows Server 2022 Standard Evaluation
Licence Windows valide pour 152 jours
Build 20348.fe_release.210507-1500
09:50



The screenshot shows a terminal window on a Debian system with the prompt `zabbix@debian:~`. The user runs the command `ping -c 4 192.168.122.235`. The output shows four successful ping requests with response times ranging from 1.29 ms to 1.85 ms. The statistics summary indicates 4 packets transmitted, 4 received, and 0% packet loss.

```
zabbix@debian:~$ ping -c 4 192.168.122.235
PING 192.168.122.235 (192.168.122.235) 56(84) bytes of data:
64 bytes from 192.168.122.235: icmp_seq=1 ttl=128 time=1.40 ms
64 bytes from 192.168.122.235: icmp_seq=2 ttl=128 time=1.63 ms
64 bytes from 192.168.122.235: icmp_seq=3 ttl=128 time=1.29 ms
64 bytes from 192.168.122.235: icmp_seq=4 ttl=128 time=1.85 ms

--- 192.168.122.235 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.290/1.539/1.845/0.214 ms
zabbix@debian:~$
```

Partie 3 — Déploiement de l'Active Directory

Avant de configurer l'authentification LDAP dans Zabbix, il est nécessaire de disposer d'un domaine Active Directory opérationnel. Cette partie décrit la promotion du serveur Windows en contrôleur de domaine et la création des objets nécessaires.

3.1 Installation du rôle AD DS

Le rôle Active Directory Domain Services (AD DS) est installé via PowerShell (en tant qu'Administrateur) :

```
Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools
```

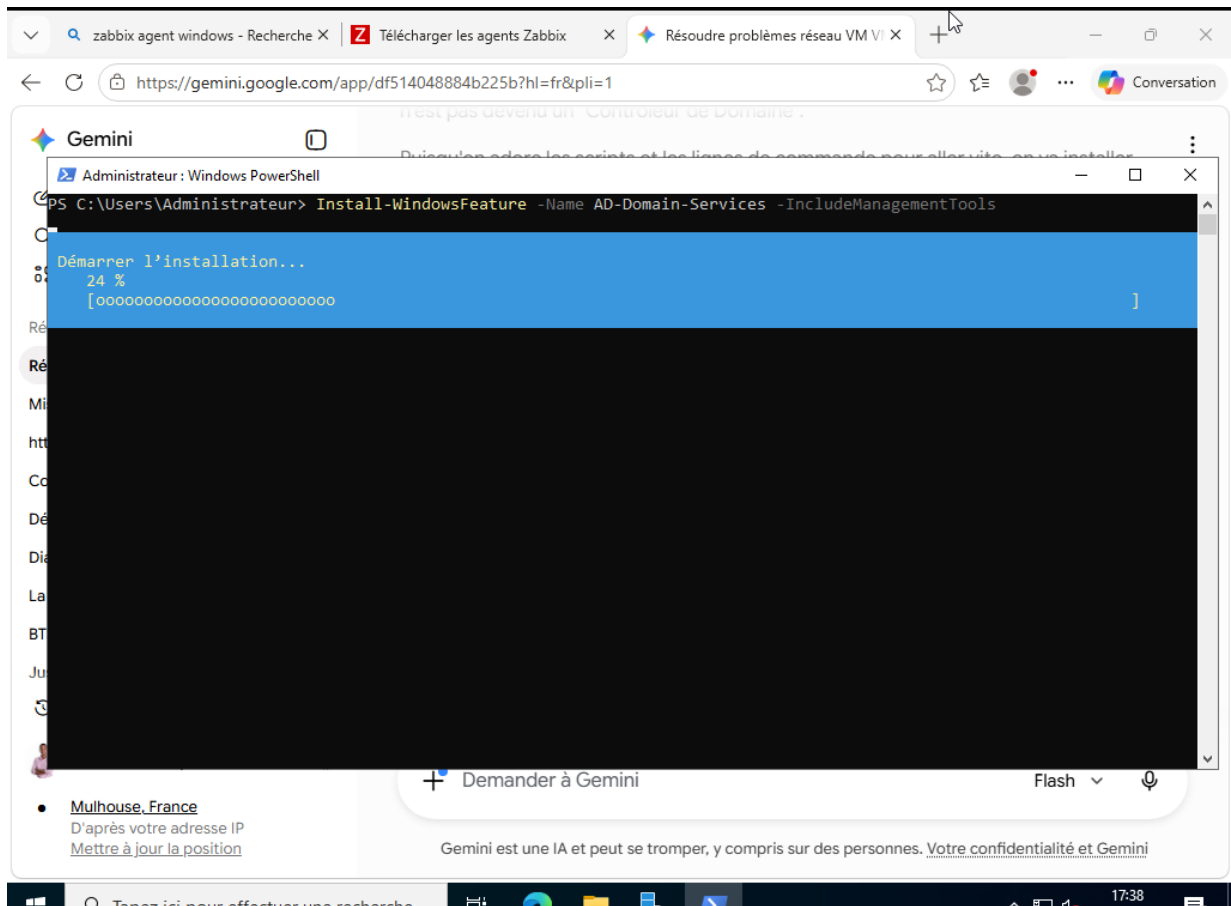
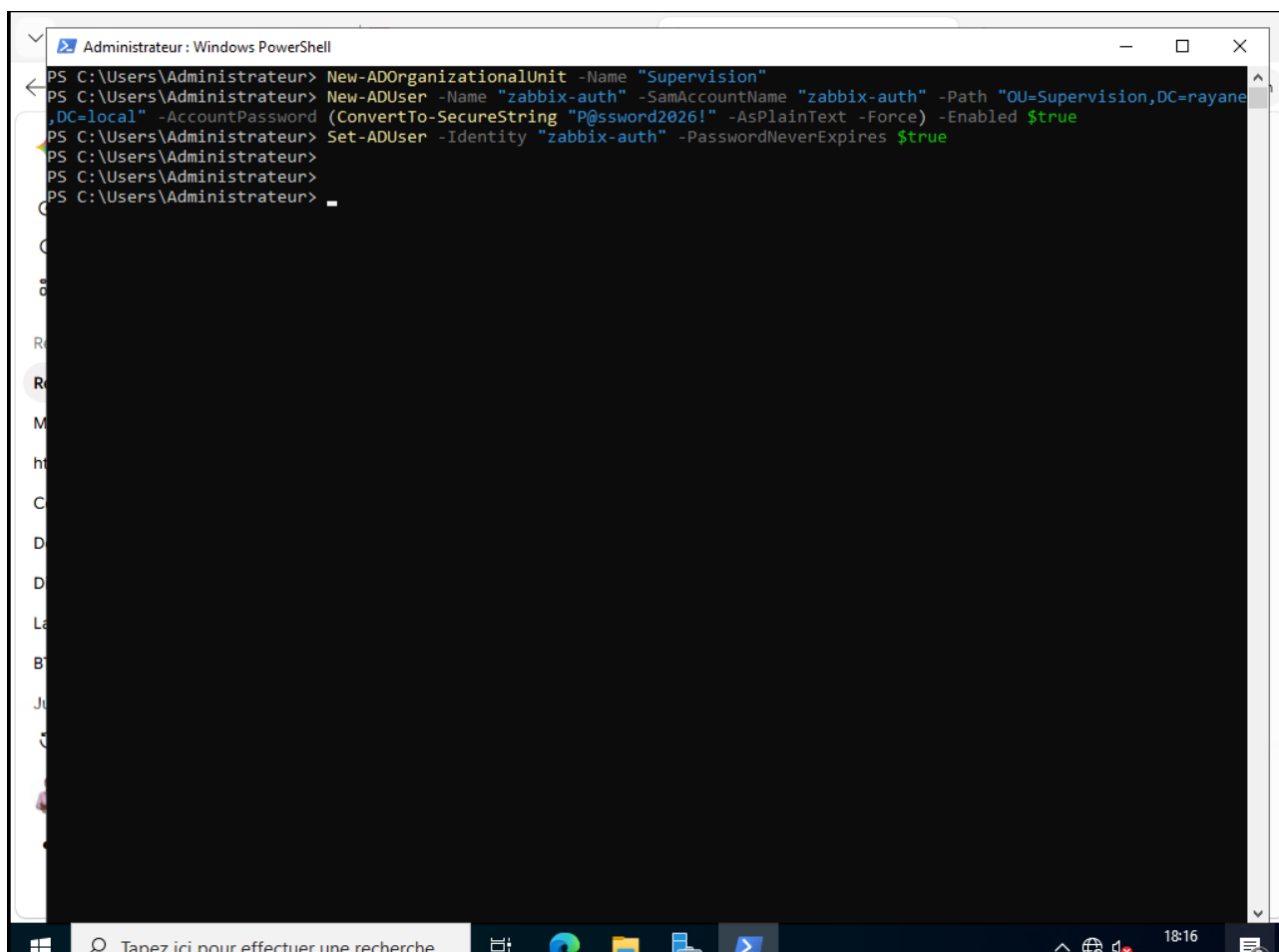


Figure 16 — Installation du rôle AD DS via PowerShell

3.2 Création du domaine rayane.local

Une fois le rôle installé, la commande suivante crée la forêt Active Directory et promeut le serveur en contrôleur de domaine. Le serveur redémarre automatiquement à l'issue de l'opération.

```
Install-ADDSForest -DomainName "rayane.local" -SafeModeAdministratorPassword  
(ConvertTo-SecureString "P@ssword2026!" -AsPlainText -Force) -Force
```

```
Administrateur : Windows PowerShell
PS C:\Users\Administrateur> New-ADOrganizationalUnit -Name "Supervision"
PS C:\Users\Administrateur> New-ADUser -Name "zabbix-auth" -SamAccountName "zabbix-auth" -Path "OU=Supervision,DC=rayane
,DC=local" -AccountPassword (ConvertTo-SecureString "P@ssword2026!" -AsPlainText -Force) -Enabled $true
PS C:\Users\Administrateur> Set-ADUser -Identity "zabbix-auth" -PasswordNeverExpires $true
PS C:\Users\Administrateur>
PS C:\Users\Administrateur>
PS C:\Users\Administrateur>
PS C:\Users\Administrateur>
```

Figure 18 — Création du compte de liaison zabbix-auth dans l'OU Supervision

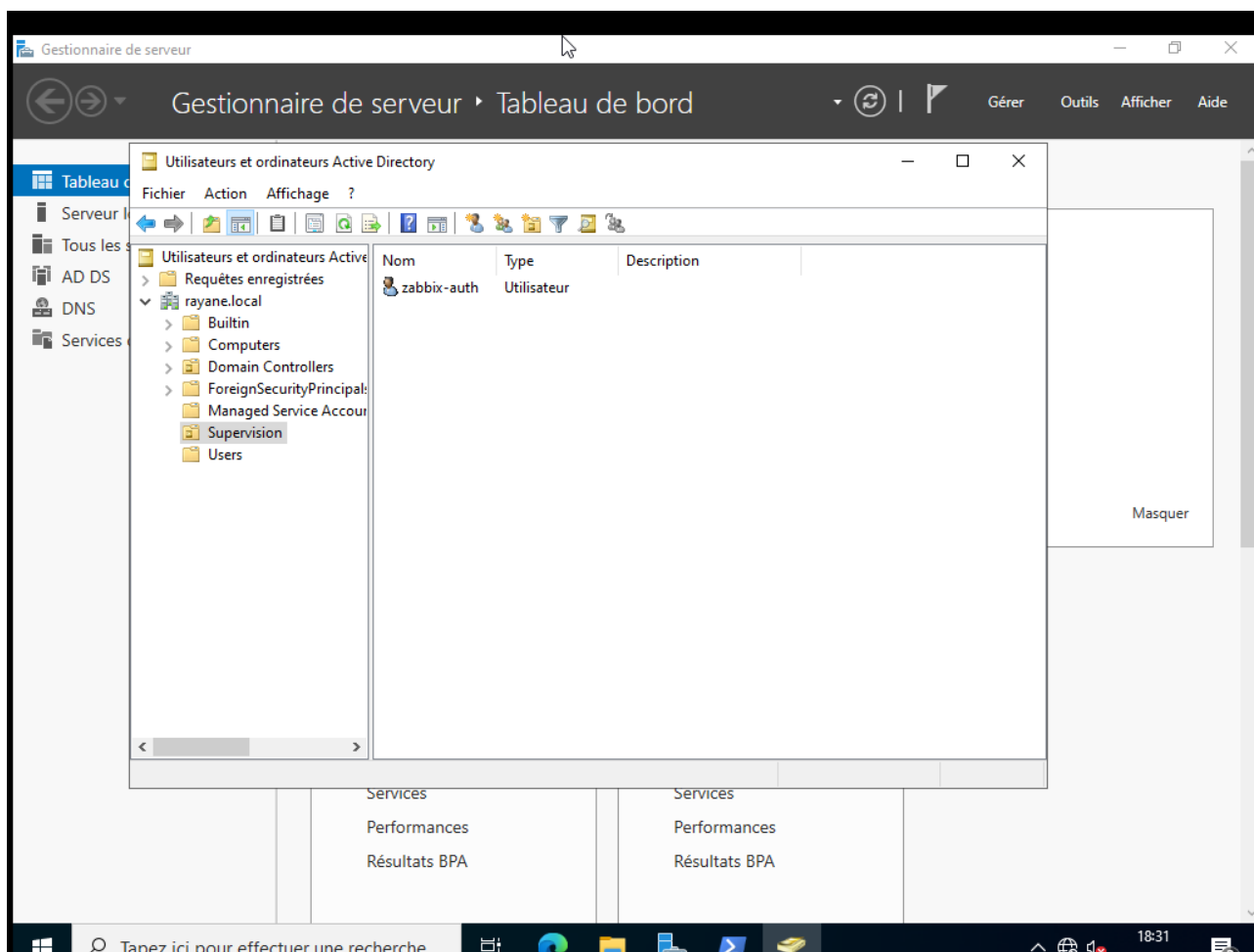


Figure 19 — Vérification de la création du compte dans l'annuaire

Partie 4 — Configuration de l'authentification LDAP

Cette partie décrit la configuration de l'authentification LDAP dans Zabbix, permettant aux utilisateurs du domaine Active Directory de se connecter à l'interface web de supervision.

4.1 Accès aux paramètres d'authentification

Dans l'interface web Zabbix, accéder au menu Administration (ou Users selon la version) > Authentication, puis sélectionner l'onglet LDAP settings et activer la case Enable LDAP authentication.

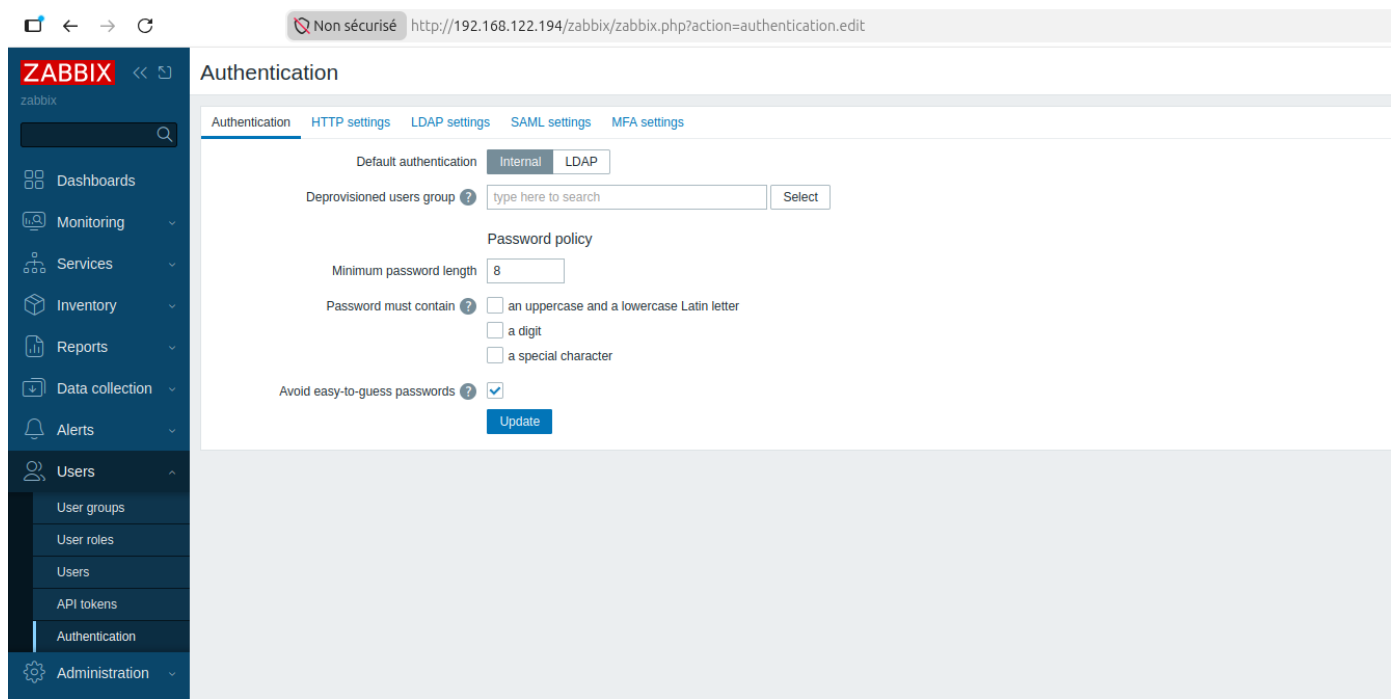


Figure 20 — Activation de l'authentification LDAP dans Zabbix

4.2 Configuration du serveur Active Directory

Cliquer sur le bouton Add et renseigner les paramètres de connexion à l'annuaire :

- Name : Active Directory Rayane
- Host : 192.168.122.235 (adresse IP du contrôleur de domaine)
- Port : 389 (port LDAP standard)
- Base DN : dc=rayane,dc=local
- Search attribute : sAMAccountName (identifiant Windows — respecter la casse)
- Bind DN : zabbix-auth@rayane.local (compte de liaison créé précédemment)
- Bind password : mot de passe du compte de liaison

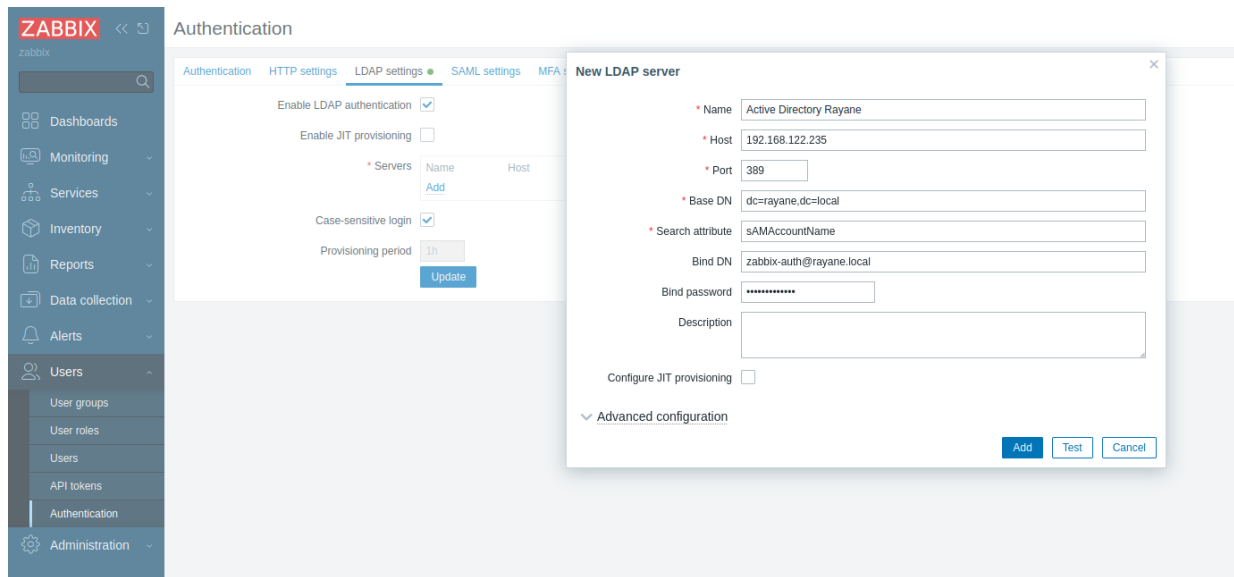


Figure 21 — Paramètres de connexion LDAP au serveur Active Directory

4.3 Test et validation de la connexion LDAP

La section Test authentication en bas du formulaire permet de valider la configuration en testant une authentification avec un compte du domaine. Un bandeau vert confirme le succès de la connexion. Cliquer sur Add pour enregistrer la configuration.

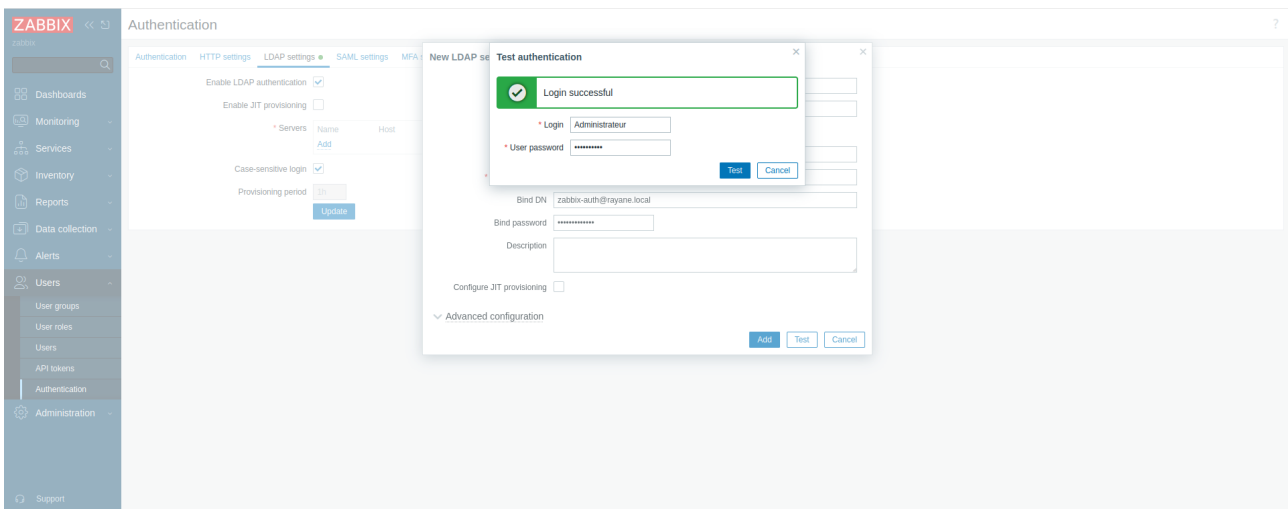


Figure 22 — Test de l'authentification LDAP réussi

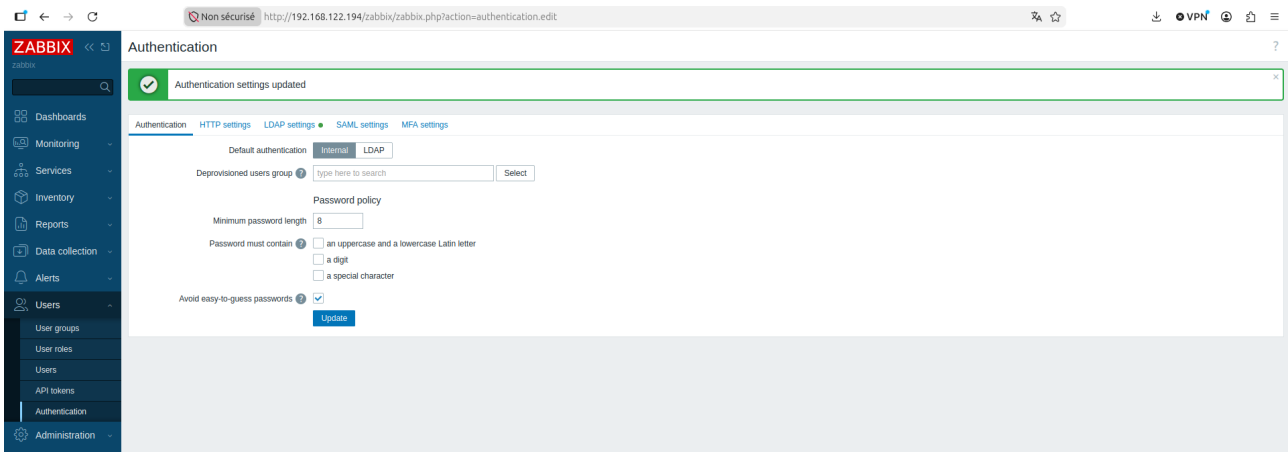


Figure 23 — Connexion LDAP validée

Partie 5 — Sécurisation avec LDAPS et Autorité de Certification

LDAP standard transmet les données, y compris les mots de passe, en clair sur le réseau. La mise en place de LDAPS (LDAP over SSL/TLS) chiffre ces échanges et garantit la confidentialité des authentifications. Cette partie décrit le déploiement d'une Autorité de Certification interne (PKI) et la configuration de LDAPS.

5.1 Préparation du serveur SRV-PKI-01

Configuration IP

Le serveur SRV-PKI-01 doit être configuré avec une adresse IP statique et pointer vers le contrôleur de domaine pour la résolution DNS, condition obligatoire pour rejoindre le domaine.

- Adresse IP : 192.168.122.236
- Masque de sous-réseau : 255.255.255.0
- Serveur DNS préféré : 192.168.122.235 (SRV-DC-01)

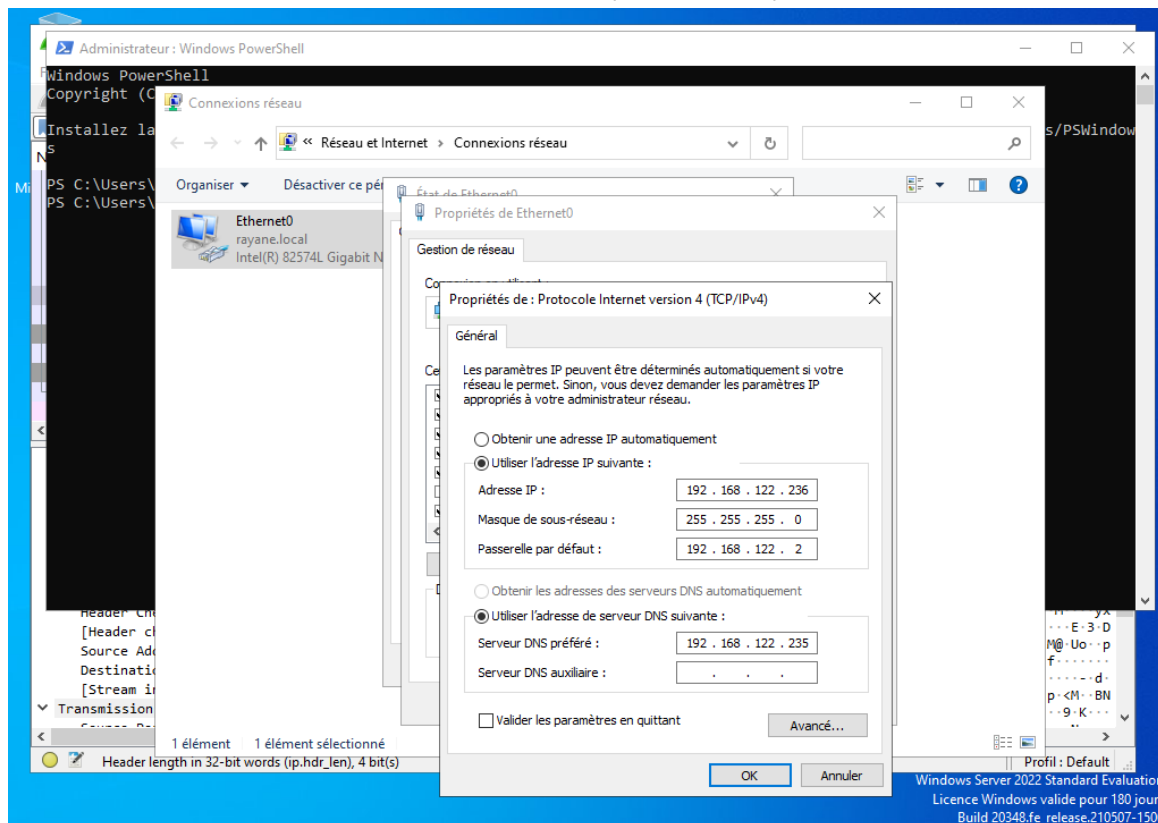


Figure 24 — Configuration IP du serveur SRV-PKI-01

Jonction au domaine

Depuis le Gestionnaire de serveur, dans Serveur local > Groupe de travail, cliquer sur Modifier, cocher Domaine et saisir rayane.local. Renseigner les identifiants de l'administrateur du domaine. Le serveur redémarre après jonction.

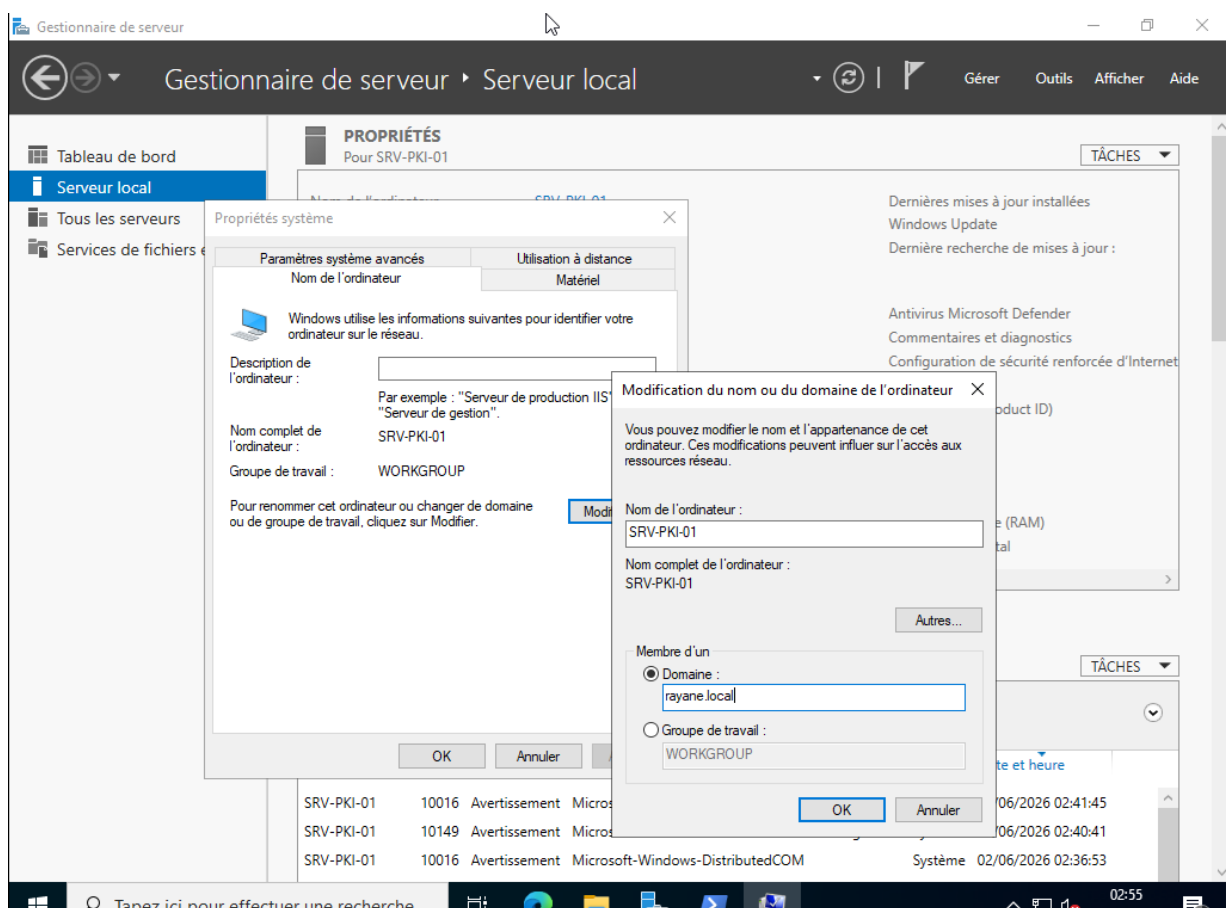


Figure 25 — Jonction du serveur SRV-PKI-01 au domaine rayane.local

5.2 Installation du rôle AD CS

L'installation des Services de certificats Active Directory (AD CS) s'effectue depuis le Gestionnaire de serveur via **Gérer > Ajouter des rôles et fonctionnalités**. Sélectionner Services de certificats Active Directory puis valider l'ajout des fonctionnalités.

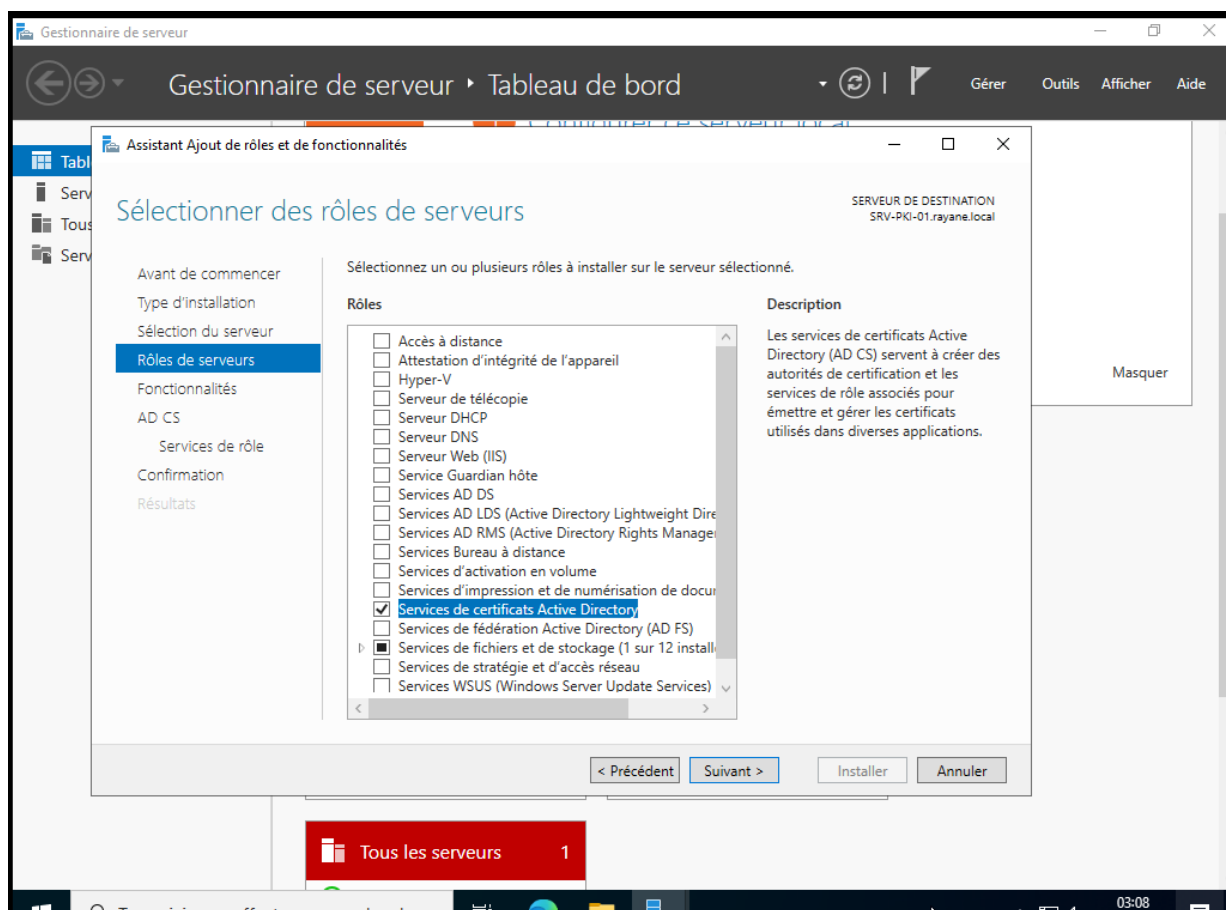


Figure 26 — Installation du rôle AD CS

5.3 Configuration de l'Autorité de Certification

À l'issue de l'installation, une notification apparaît dans le Gestionnaire de serveur pour finaliser la configuration. Renseigner les paramètres suivants :

- Type d'autorité : Autorité de certification d'entreprise (Enterprise CA)
- Type de CA : Autorité de certification racine (Root CA)
- Clé privée : Créer une nouvelle clé privée
- Chiffrement : SHA256 / 2048 bits (valeurs par défaut)
- Nom de la CA : rayane-SRV-PKI-01-CA (généré automatiquement)
- Période de validité : 5 ans

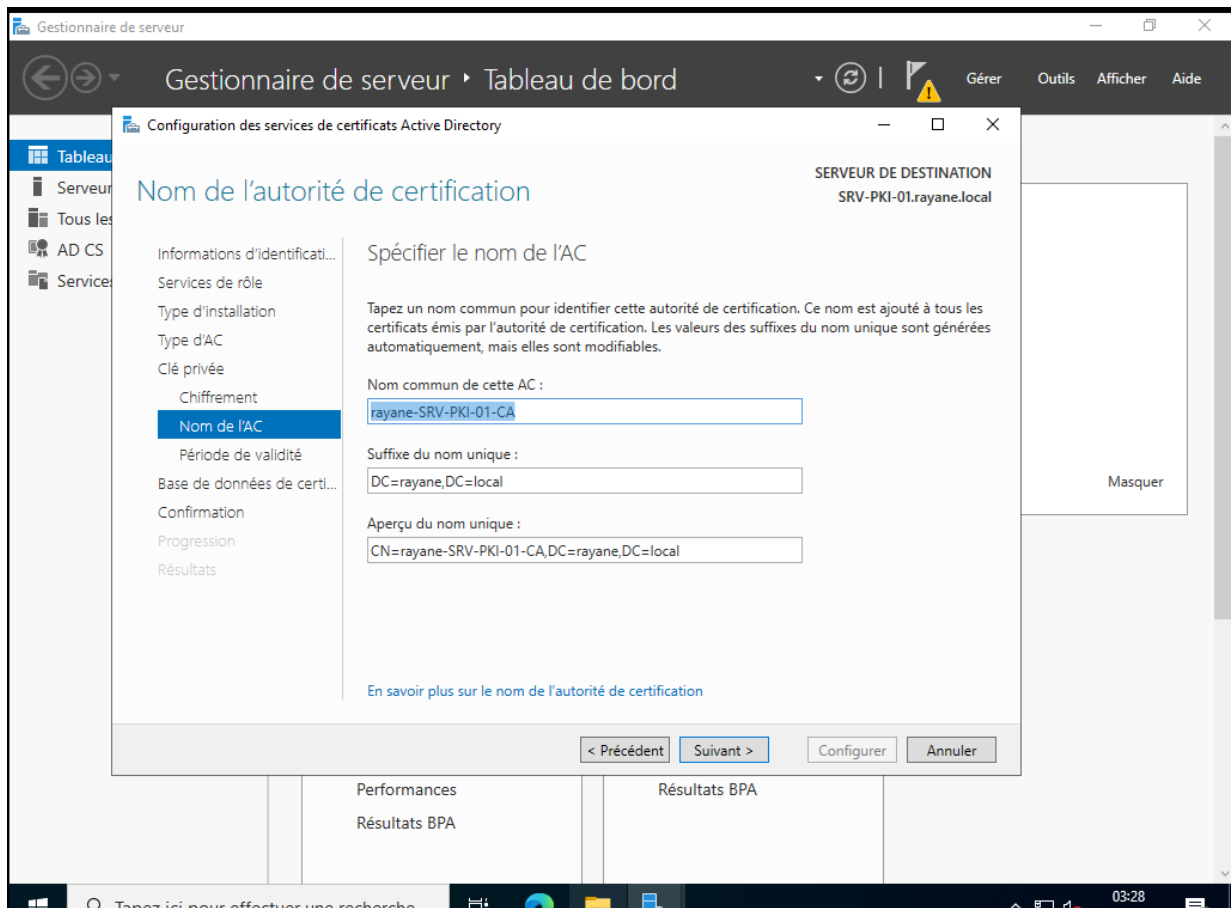


Figure 27 — Configuration de l'Autorité de Certification racine

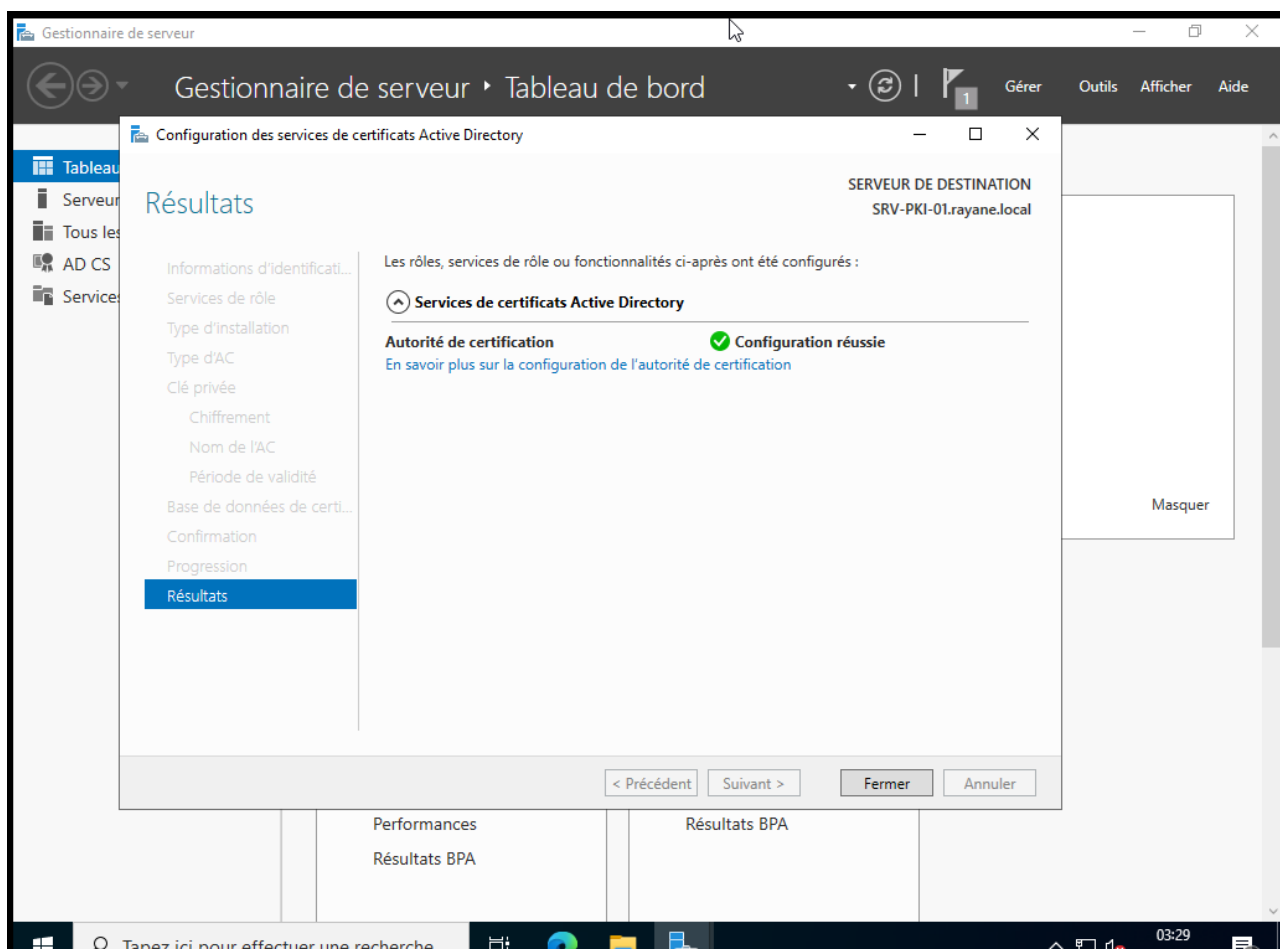


Figure 28 — Finalisation de la configuration ADCS

5.4 Export du certificat depuis SRV-PKI-01

Le certificat racine de l'autorité doit être exporté pour être installé sur le serveur Zabbix, lui permettant de valider les connexions SSL.

- Ouvrir MMC (Windows + R > mmc) et ajouter le composant Certificats (compte d'ordinateur, ordinateur local).
- Dans l'arborescence : Autorités de certification racines de confiance > Certificats.
- Effectuer un clic droit sur le certificat rayane-SRV-PKI-01-CA > Toutes les tâches > Exporter.
- Sélectionner le format DER encodé binaire (.CER) et enregistrer sous le nom rayane-CA.cer.

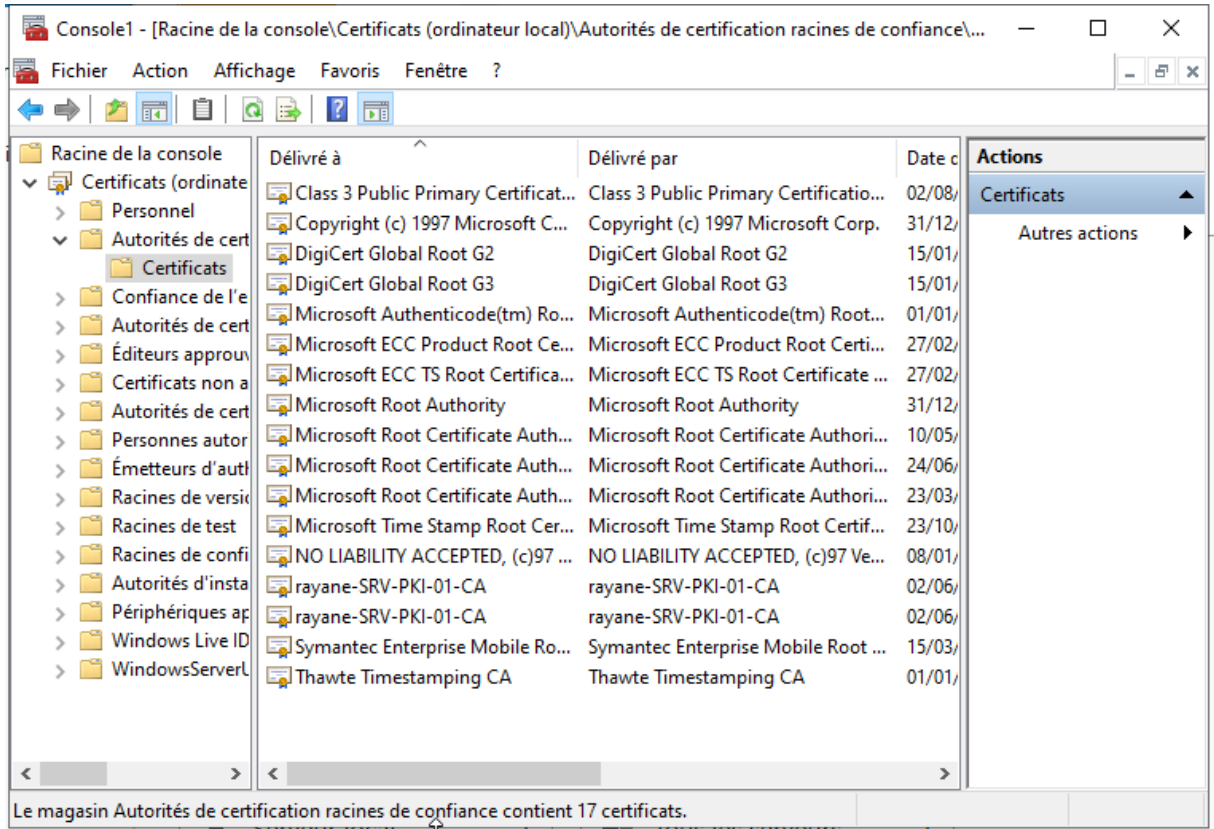


Figure 29 — Export du certificat de l'Autorité de Certification

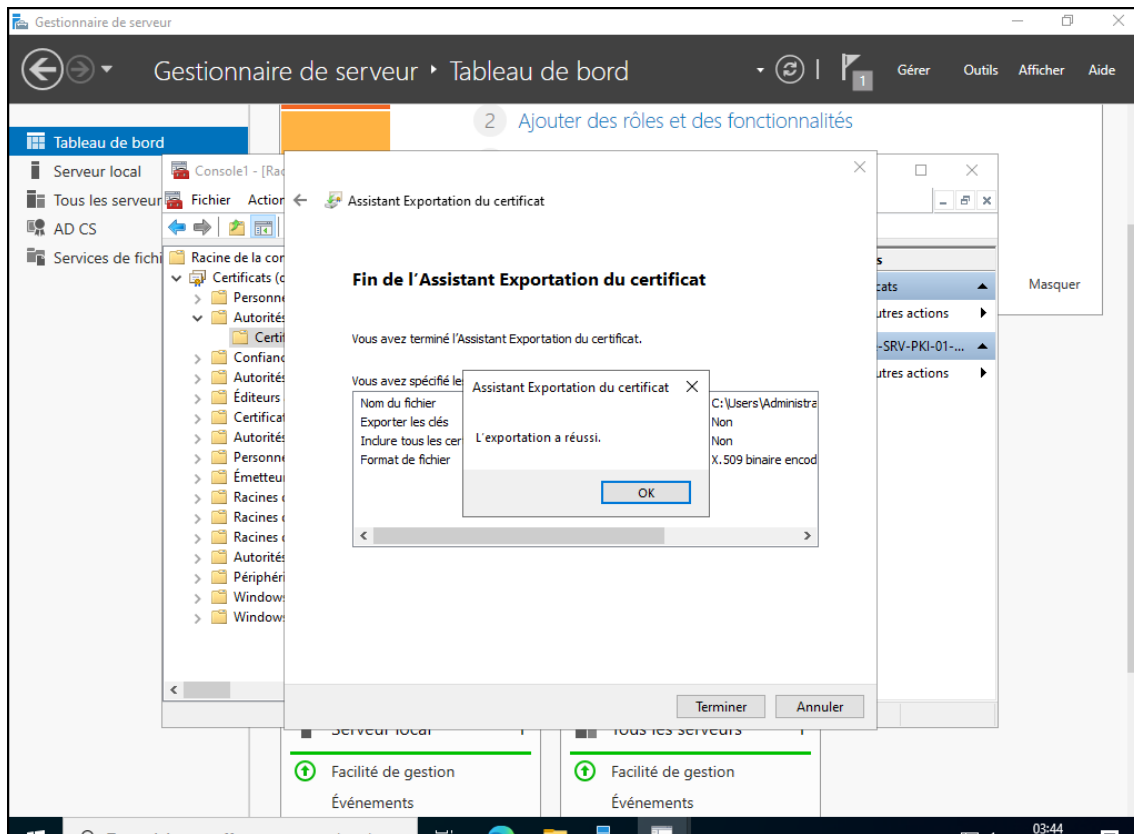
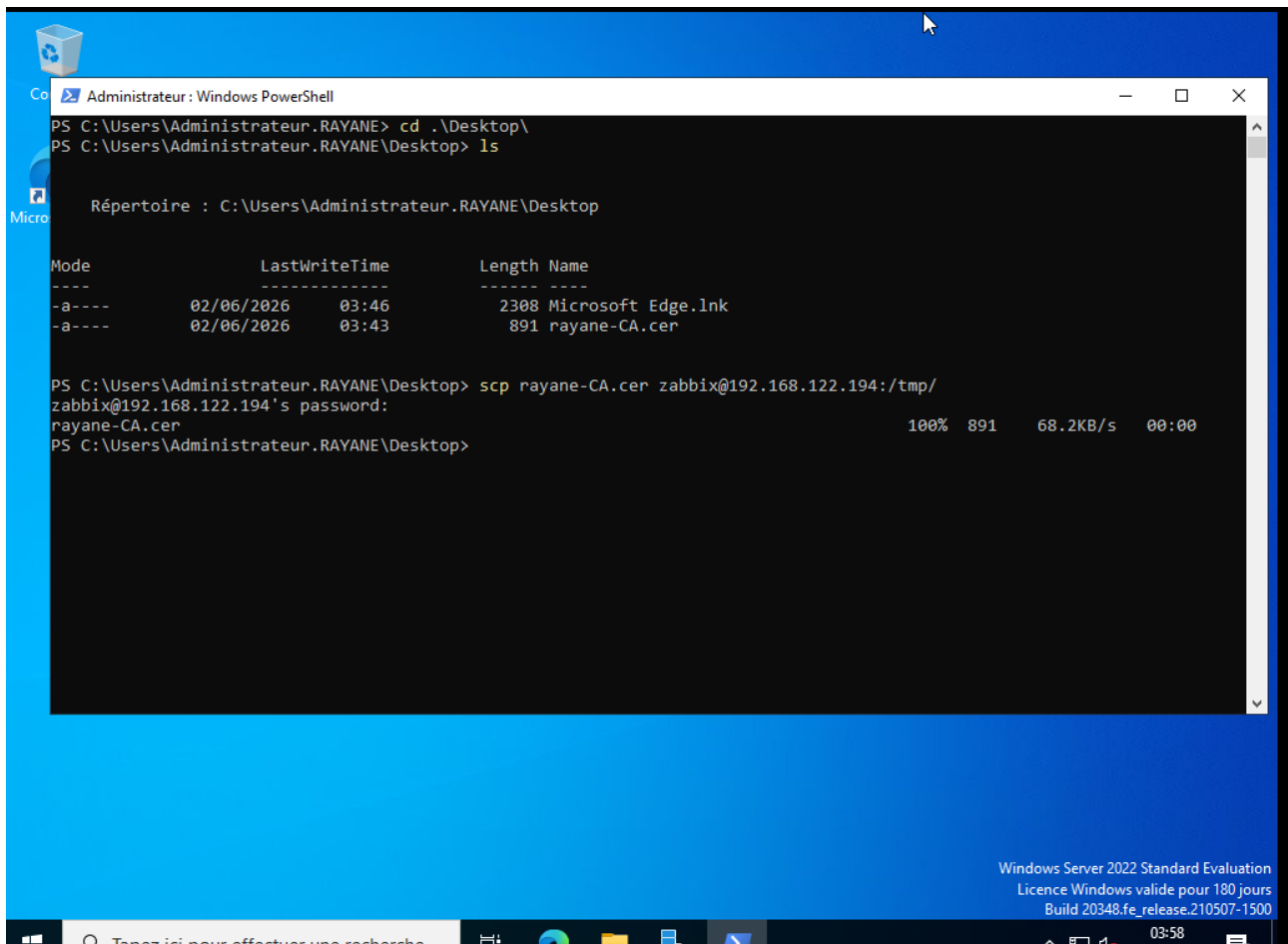


Figure 30 — Certificat exporté au format DER

5.5 Transfert et installation du certificat sur Zabbix

Le certificat est transféré vers le serveur Zabbix via SCP, puis installé dans le magasin de confiance du système Linux.

```
scp C:\Users\Administrateur\Desktop\rayane-CA.cer  
zabbix@192.168.122.194:/tmp/
```



Sur le serveur Zabbix (connexion SSH) :

```
cd /tmp
openssl x509 -inform der -in rayane-CA.cer -out rayane-CA.crt
sudo mv rayane-CA.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates
```

```
zabbix@debian:~/tmp
zabbix@debian:~$ # 1. Va dans le répertoire temporaire où se trouve le fichier t
ransféré
cd /tmp

# 2. Convertis le fichier .cer de Windows en certificat .crt lisible par Linux
openssl x509 -inform der -in rayane-CA.cer -out rayane-CA.crt

# 3. Déplace le certificat converti dans le magasin des autorités de confiance (
Debian/Ubuntu)
sudo mv rayane-CA.crt /usr/local/share/ca-certificates/

# 4. Mets à jour le magasin pour que le système prenne en compte la nouvelle clé

sudo update-ca-certificates
[sudo] password for zabbix:
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt, it does not contain exactly one ce
rtificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
zabbix@debian:~/tmp$
```

Figure 31 — Installation du certificat sur le serveur Zabbix

5.6 Test du canal LDAPS

La commande suivante permet de vérifier que la connexion chiffrée sur le port 636 fonctionne correctement entre le serveur Zabbix et le contrôleur de domaine.

```
openssl s_client -connect 192.168.122.235:636
```

Un retour « *Verification: OK* » confirme que le chiffrement TLS est opérationnel.

MFA s

LDAP Server

✕

* Name

* Host

* Port

* Base DN

* Search attribute

Bind DN

Bind password

Description

Configure JIT provisioning

Group configuration ? memberOf groupOfNames

Group name attribute

User group membership attribute

User name attribute

User last name attribute

* User group mapping

LDAP group pattern	User groups	User role	Action
*	Zabbix administrators	Super admin role	Remove
Add			

Media type mapping ?

Name	Media type	Attribute	Action
Add			

▼ [Advanced configuration](#)

Update
Test
Cancel

Figure 33 — Configuration LDAPS dans l'interface Zabbix

The screenshot shows the Zabbix 7.4 web interface. The left sidebar contains navigation menus for Dashboards, Monitoring, Services, Inventory, Reports, Data collection, Alerts, Users, Administration, Support, Integrations, and Help. The main content area is titled "Authentication" and has tabs for Authentication, HTTP settings, LDAP settings (selected), SAML settings, and MFA. Under "LDAP settings", there are checkboxes for "Enable LDAP authentication" and "Enable JIT provisioning", both of which are checked. Below these are fields for "Servers" (Name: Active Directory Rayane, Add button), "Case-sensitive login" (checked), and "Provisioning period" (1h, Update button). A "Test authentication" modal window is open in the foreground, displaying a green checkmark and the text "Login successful". The modal also shows the test details: Login: test-jury, User password: P@ssword12026!, User role: Super admin role, User groups: Zabbix administrators, and Media type: No value. Below the modal, the "LDAP Server" configuration section is visible, including "Configure JIT provisioning" (checked), "Group configuration" (memberOf, groupOfNames), "Group name attribute" (cn), "User group membership attribute" (memberOf), "User name attribute" (sAMAccountName), "User last name attribute" (empty), and "User group mapping" table. The "Media type mapping" table is also present. At the bottom right of the configuration section are "Update", "Test", and "Cancel" buttons.

Figure 34 — Connexion LDAPS validée avec succès

Partie 6 — Analyse de sécurité avec Wireshark

Cette dernière partie illustre, à travers une capture réseau Wireshark, l'intérêt concret du passage de LDAP à LDAPS du point de vue de la sécurité.

6.1 Comparaison LDAP vs LDAPS

Les comptes qui se connectent à l'interface de Zabbix sont généralement des membres de l'équipe informatique disposant de privilèges élevés au sein du système d'information. La sécurisation de leurs authentifications est donc critique.

Avec une connexion LDAP non chiffrée (port 389), l'intégralité du trafic transite en clair sur le réseau. Un attaquant capable de capturer le trafic peut obtenir sans difficulté les identifiants et mots de passe des comptes de l'annuaire, y compris le mot de passe du compte connecteur (zabbix-auth).

L'activation de LDAPS (port 636) chiffre l'ensemble de ces échanges via TLS, rendant toute interception inexploitable.

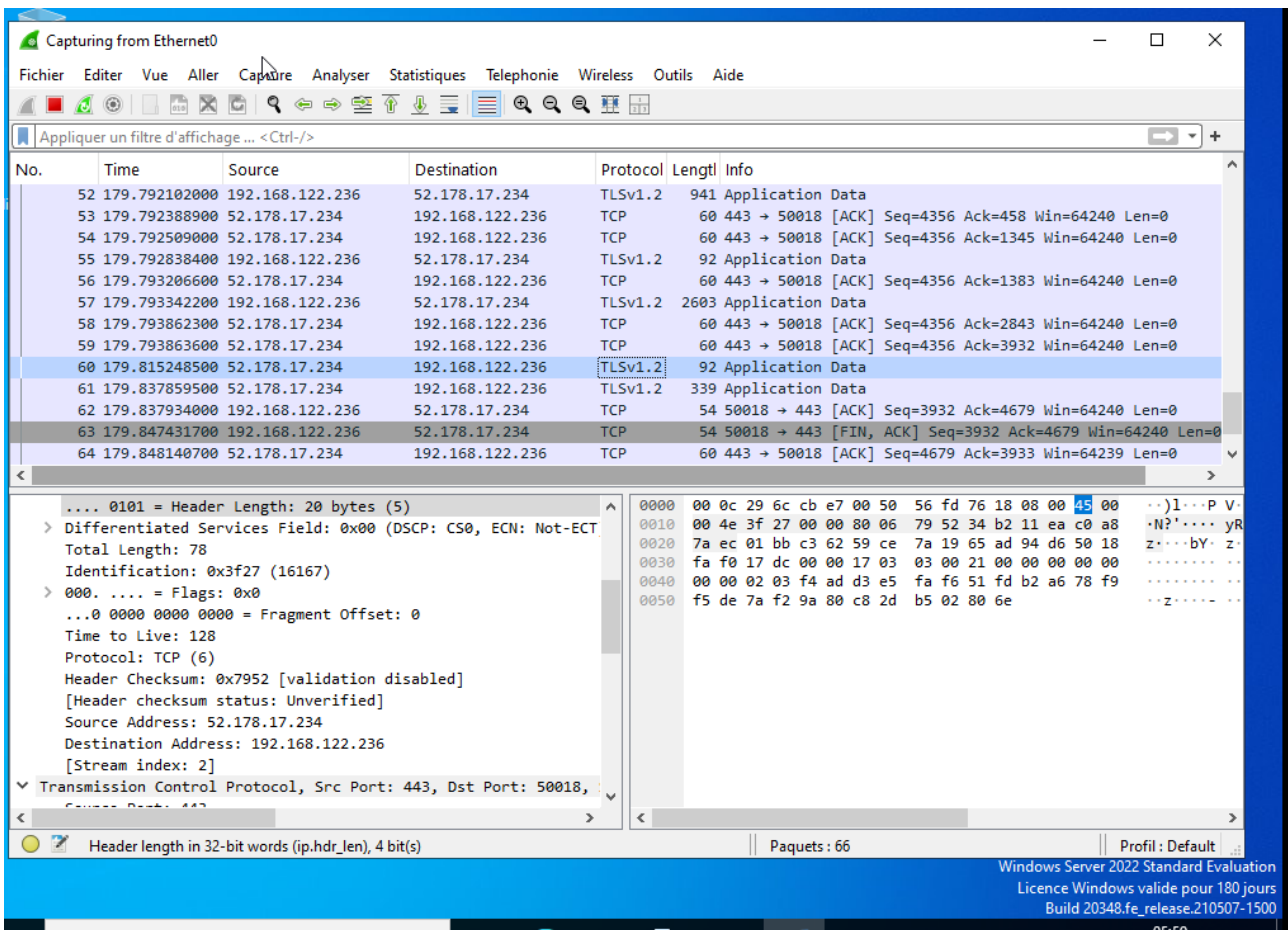


Figure 35 — Capture Wireshark

6.2 Conclusion de l'analyse

La mise en place de LDAPS représente une mesure de sécurité essentielle dans tout environnement de production. Elle protège les identifiants d'authentification contre les attaques de type Man-in-the-Middle (MitM) et garantit la confidentialité des échanges entre Zabbix et l'annuaire Active Directory.

Conclusion

Ce travail pratique a permis de mettre en œuvre une solution de supervision complète et sécurisée, illustrant plusieurs compétences fondamentales du BTS SIO :

- Administration système Linux : installation, configuration de services (Zabbix, MariaDB, Apache).
- Administration Windows Server : promotion en contrôleur de domaine, gestion des objets Active Directory, installation d'une PKI.
- Sécurité réseau : passage de l'authentification en clair (LDAP) à une authentification chiffrée (LDAPS) avec gestion des certificats.
- Supervision informatique : déploiement d'agents, déclaration d'hôtes, utilisation de modèles de supervision.
- Analyse réseau : utilisation de Wireshark pour comparer et valider les niveaux de sécurité.

L'ensemble de la chaîne de l'installation du serveur de supervision à la sécurisation des authentifications constitue une infrastructure cohérente et documentée, directement transposable en environnement professionnel.