Documentation Projet SISR

Infrastructure réseau multi-sites — Paris & Honfleur

Léo DAUTRICHE

BTS SIO — Option SISR

EPSI

Année scolaire 2025 – 2026

Table des matières

1. Présentation générale du projet

Ce document présente la documentation technique complète du projet SISR réalisé dans le cadre du BTS SIO option SISR à l'EPSI. Le projet consiste en la mise en place d'une infrastructure réseau multi-sites reliant deux sites distants : Paris et Honfleur, connectés par un VPN site-à-site via WireGuard. L'infrastructure repose sur la virtualisation Proxmox et intègre de nombreux services réseau : Active Directory, serveur web WordPress, GLPI, Nextcloud, supervision Prometheus/Grafana, ainsi que le routage, le pare-feu et la commutation.

1.1 Schéma réseau

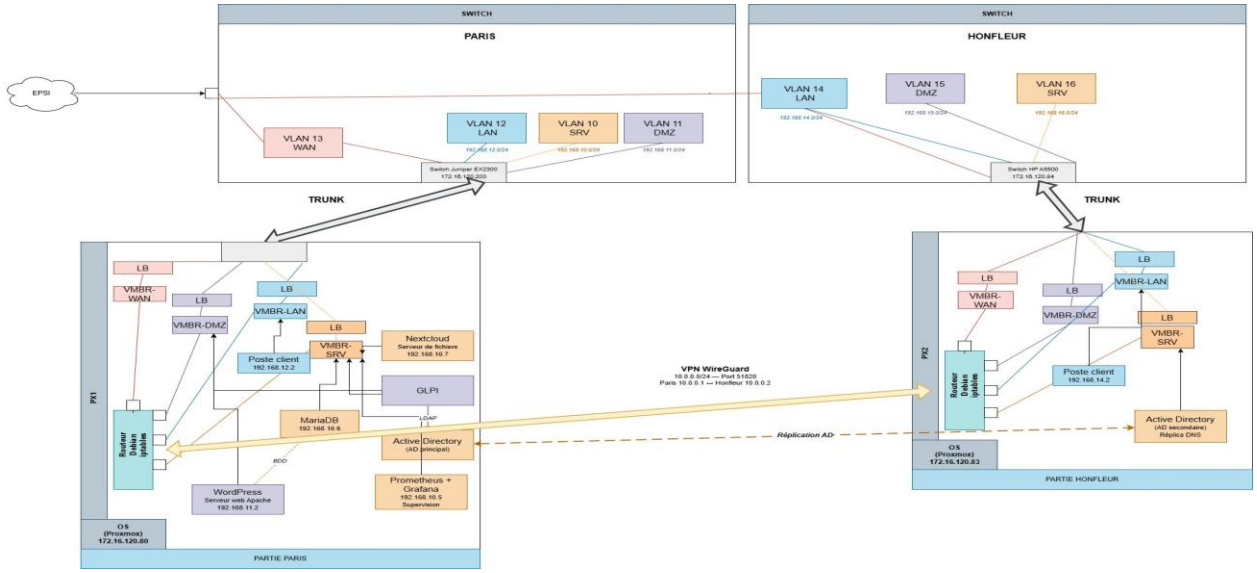


Figure 1 : Schéma de l'infrastructure réseau du projet SISR (Paris — Honfleur)

1.2 Informations générales

Infrastructure physique

Élément	Détails
Proxmox 1 (Paris)	Interface : enp2s0 — FQDN : px1.labo.loc — IP : 172.16.120.80/16
Proxmox 2 (Honfleur)	Interface : enp4s0 — FQDN : px2.labo.loc — IP : 172.16.120.83/16
Switch Juniper (Paris)	IP management : 172.16.120.200/16 — WAN : 172.16.120.88
Switch HP (Honfleur)	IP management : 172.16.120.84/16 — WAN : 172.16.120.87
VPN WireGuard	Réseau : 10.0.0.0/24 — Paris : 10.0.0.1 — Honfleur : 10.0.0.2

VLANs

Site	VLAN	Réseau

Paris	SRV (VLAN 10)	192.168.10.0/24
Paris	DMZ (VLAN 11)	192.168.11.0/24
Paris	LAN (VLAN 12)	192.168.12.0/24
Paris	WAN (VLAN 13)	—
Honfleur	LAN (VLAN 14)	192.168.14.0/24
Honfleur	DMZ (VLAN 15)	192.168.15.0/24
Honfleur	SRV (VLAN 16)	192.168.16.0/24

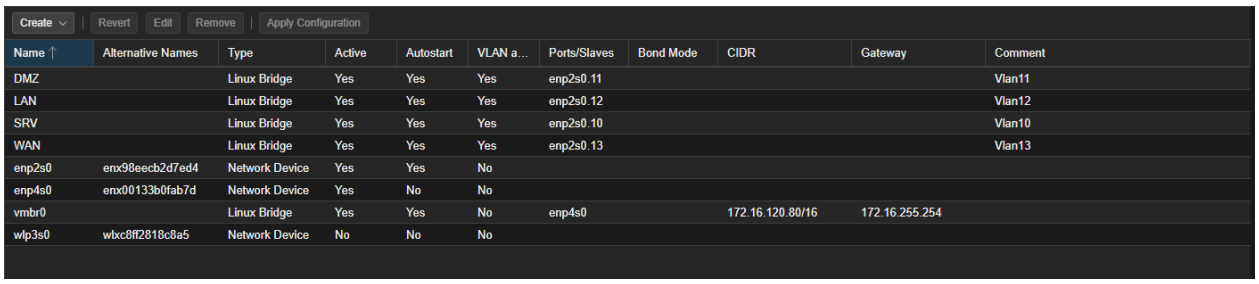
Comptes Active Directory créés

Utilisateur	Utilisateur	Utilisateur
martin.claire	dubois.julien	lambert.sophie
laurent.maxime	moreau.emilie	girard.thomas
rousseau.jeanne	blanchard.arnaud	perrin.charlotte
francois.lucas	renard.camille	barbier.antoine
collet.marine	gauthier.vincent	rocher.anais
lefebvre.hugo	lemaire.sarah	boucher.mathieu
bonnet.manon	chevalier.paul	millet.alice
perrot.jeremy	dupont.laura	moulin.bastien
benoit.adeline	nicolas.damien	baron.elise
masson.etienne	berger.romane	caron.fabien
braun.justine	pichon.adrien	forestier.louise
jacquet.raphael	roland.eva	

2. Site Paris

2.1 Linux Bridge (Proxmox 1)

Le serveur Proxmox 1 (px1.labo.loc) héberge l'ensemble des machines virtuelles du site Paris. La configuration réseau repose sur des Linux Bridges permettant de segmenter le trafic entre les différents VLANs (SRV, DMZ, LAN, WAN).



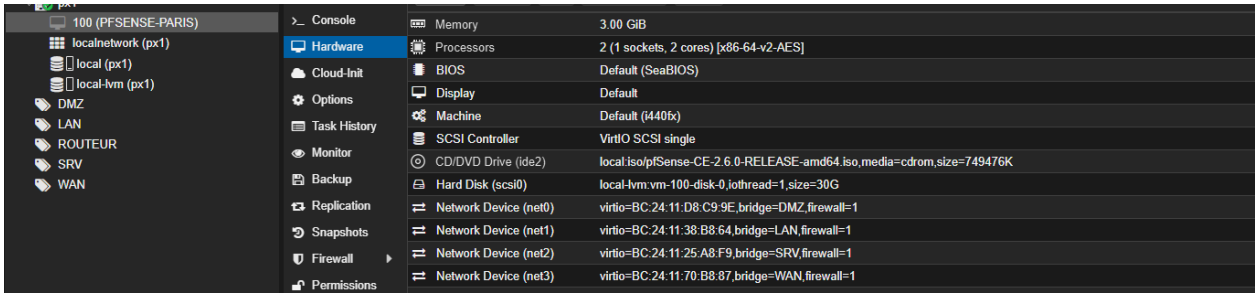
Name	Alternative Names	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR	Gateway	Comment
DMZ		Linux Bridge	Yes	Yes	Yes	enp2s0.11				Vlan11
LAN		Linux Bridge	Yes	Yes	Yes	enp2s0.12				Vlan12
SRV		Linux Bridge	Yes	Yes	Yes	enp2s0.10				Vlan10
WAN		Linux Bridge	Yes	Yes	Yes	enp2s0.13				Vlan13
enp2s0	enx98eecb2d7ed4	Network Device	Yes	Yes	No					
enp4s0	enx00133b0fab7d	Network Device	Yes	No	No					
vmbr0		Linux Bridge	Yes	Yes	No	enp4s0		172.16.120.80/16	172.16.255.254	
wlp3s0	wlxc8ff2818c8a5	Network Device	No	No	No					

Configuration des bridges réseau sur Proxmox 1

2.2 Routeur / Firewall / VPN Paris

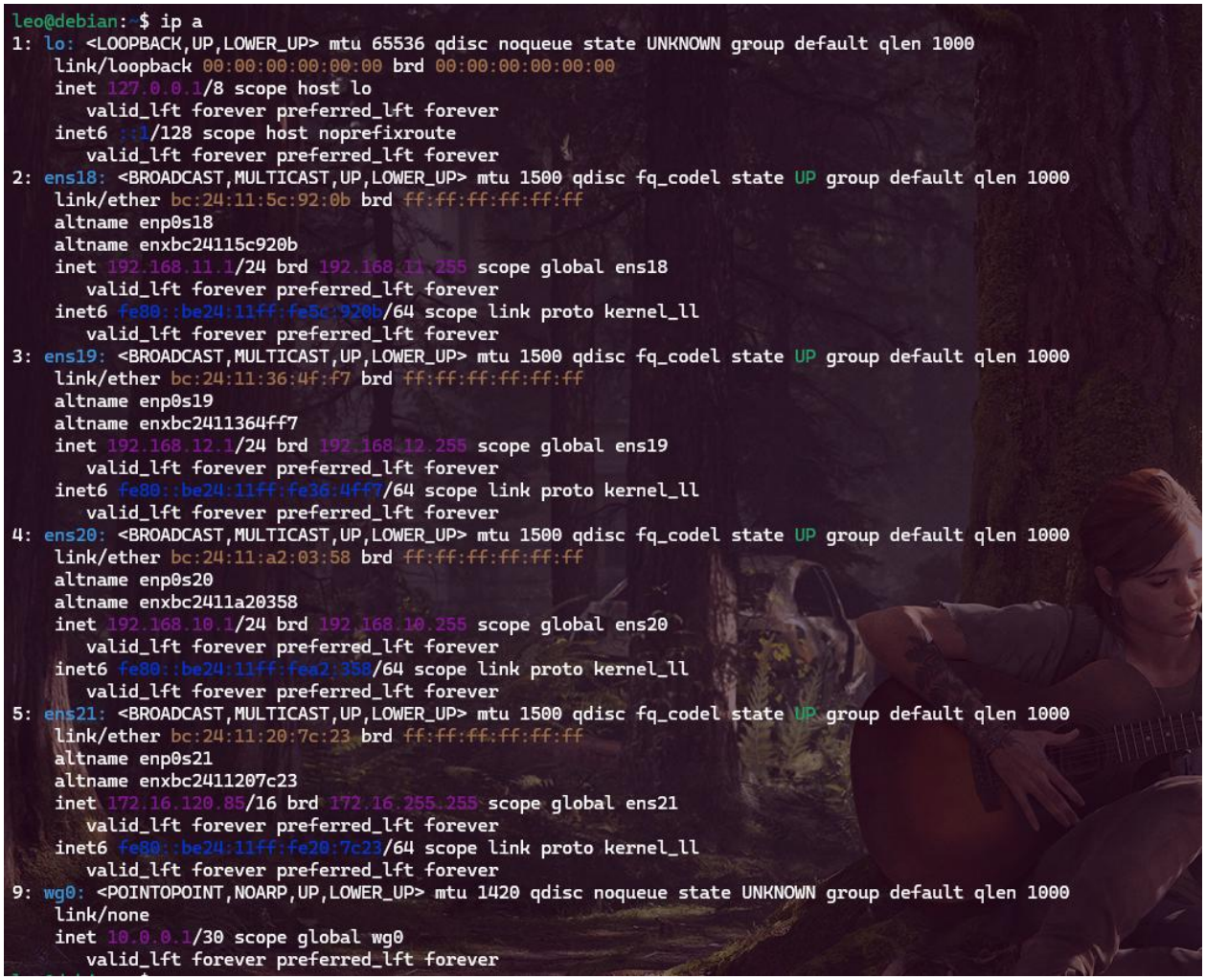
Partie routage / interfaces réseaux

On met en place tous les VLANs sur 4 cartes réseaux sur la VM :



Configuration des 4 interfaces réseau sur la VM routeur Paris

On lance la VM, puis on configure les interfaces réseaux via /etc/network/interfaces :



Configuration /etc/network/interfaces — Paris

On active le forwarding IPv4 (routage) dans /usr/lib/sysctl.d/50-default.conf :

```
#on ajoute cette ligne
net.ipv4.ip_forward=1
```

Partie VPN WireGuard — Paris

```
sudo su -
apt install wireguard wireguard-tools
cd /etc/wireguard
umask 077
wg genkey | tee privatekey_paris | wg pubkey > publickey_paris
nano /etc/wireguard/wg0.conf
[Interface]
# Configuration Paris
PrivateKey = wABC60L3agNQBRPmiEy+DyaopzbHcJj1AajDE5nCpmY=
Address = 10.0.0.1/30
ListenPort = 51820

[Peer]
# Honfleur
PublicKey = bGpDBGlznQISOfhsCboaWlff+qitmT/3Kn+zDPLAnmE=
Endpoint = 172.16.120.86:51820
AllowedIPs = 10.0.0.2/32, 192.168.14.0/24, 192.168.15.0/24, 192.168.16.0/24
PersistentKeepalive = 25
sudo systemctl start wg-quick@wg0
sudo systemctl status wg-quick@wg0
```

Partie Firewall — Paris

Installation d'iptables et netfilter-persistent :

```
sudo apt install iptables -y && sudo apt install netfilter-persistent
```

Règles FORWARD appliquées :

```
Chain FORWARD (policy ACCEPT)
  ACCEPT all -- ens18 ens21
  ACCEPT all -- ens19 ens21
  ACCEPT all -- ens20 ens21
  ACCEPT all -- ens21 ens18  state RELATED,ESTABLISHED
  ACCEPT all -- ens21 ens19  state RELATED,ESTABLISHED
  ACCEPT all -- ens21 ens20  state RELATED,ESTABLISHED
  ACCEPT all -- wg0 *
  ACCEPT all -- * wg0
```

Règles NAT (MASQUERADE) :

```
MASQUERADE all -- 192.168.12.0/24 anywhere
MASQUERADE all -- 10.0.0.0/30 192.168.12.0/24
MASQUERADE all -- 192.168.11.0/24 anywhere
MASQUERADE all -- 10.0.0.0/30 192.168.11.0/24
MASQUERADE all -- 192.168.10.0/24 anywhere
MASQUERADE all -- 10.0.0.0/30 192.168.10.0/24
```

Vérification du handshake WireGuard :

```
sudo wg show wg0
interface: wg0
  public key: gaWxqs0nDBGepADHZrgNzPorxC2/TninC9SK2wQ3BDg=
  listening port: 51820
peer: bGpDBGlznQISOfhsCboaWlff+qitmT/3Kn+zDPLAnmE=
  endpoint: 172.16.120.86:51820
  allowed ips: 10.0.0.2/32, 192.168.14.0/24, 192.168.15.0/24, 192.168.16.0/24
  latest handshake: 9 seconds ago
  transfer: 7.91 KiB received, 22.65 KiB sent
  persistent keepalive: every 25 seconds
```

Tests de connectivité depuis le client LAN Paris (192.168.12.2) :

```
ping 192.168.12.1   # passerelle locale
ping 10.0.0.1       # IP VPN Paris
ping 10.0.0.2       # IP VPN Honfleur
ping 192.168.14.1   # LAN Honfleur
ping 192.168.14.2   # Client LAN Honfleur
```

Tous les paquets sont bien routés — le site-to-site fonctionne.

2.3 Switch Juniper

Le switch Juniper EX2300-24P assure la commutation au niveau du site Paris. Il est configuré avec les VLANs nécessaires et un port trunk vers le routeur. Le protocole SNMP est activé pour la supervision via Prometheus.

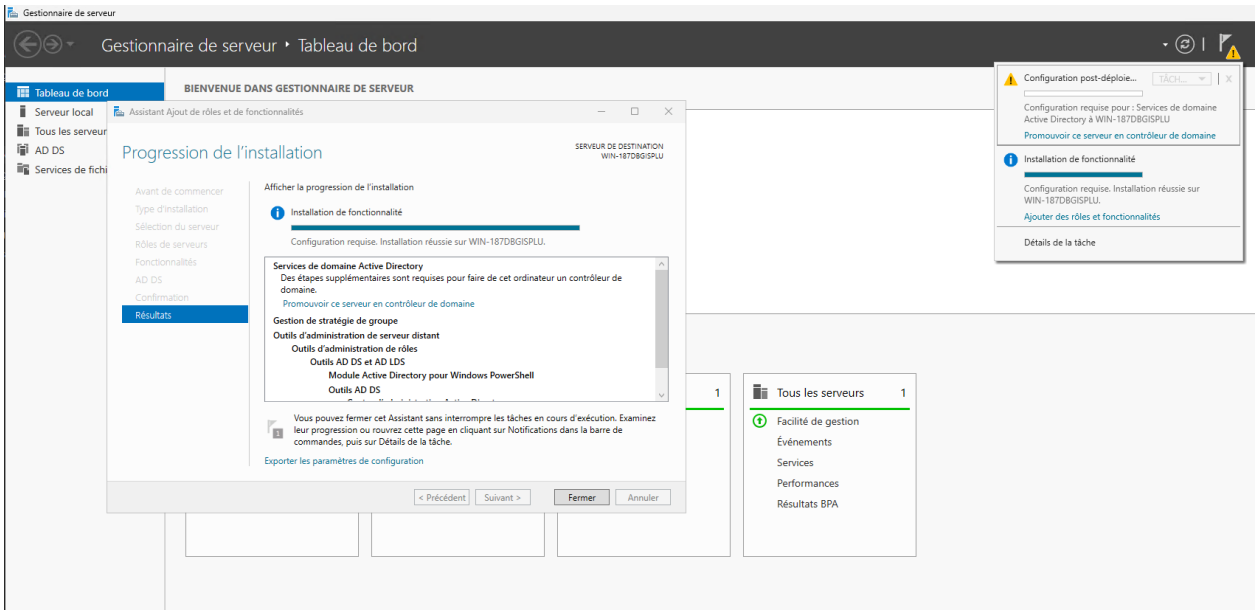
```
#login : root   mdp : vide
cli
configure

# Activer SSH
set system services ssh
set system root-authentication plain-text-password
set system services ssh root-login deny
set system services ssh connection-limit 10
set system services ssh rate-limit 5
```

```
# Créer un utilisateur admin
set system login user admin class super-user authentication plain-text-password

# Configurer les VLANs
set vlans SRV  vlan-id 10
set vlans DMZ  vlan-id 11
set vlans LAN  vlan-id 12
set vlans WAN  vlan-id 13
set vlans GUEST vlan-id 14

# Interface trunk (ge-0/0/1)
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members [10 11 12 13
14]

# Interface access WAN (ge-0/0/2)
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members WAN

# IRB pour SNMP
set vlans SRV l3-interface irb.10
set interfaces irb unit 10 family inet address 192.168.10.254/24

# SNMP
set snmp community moni authorization read-only
set snmp community moni clients 192.168.10.5/32

commit
```

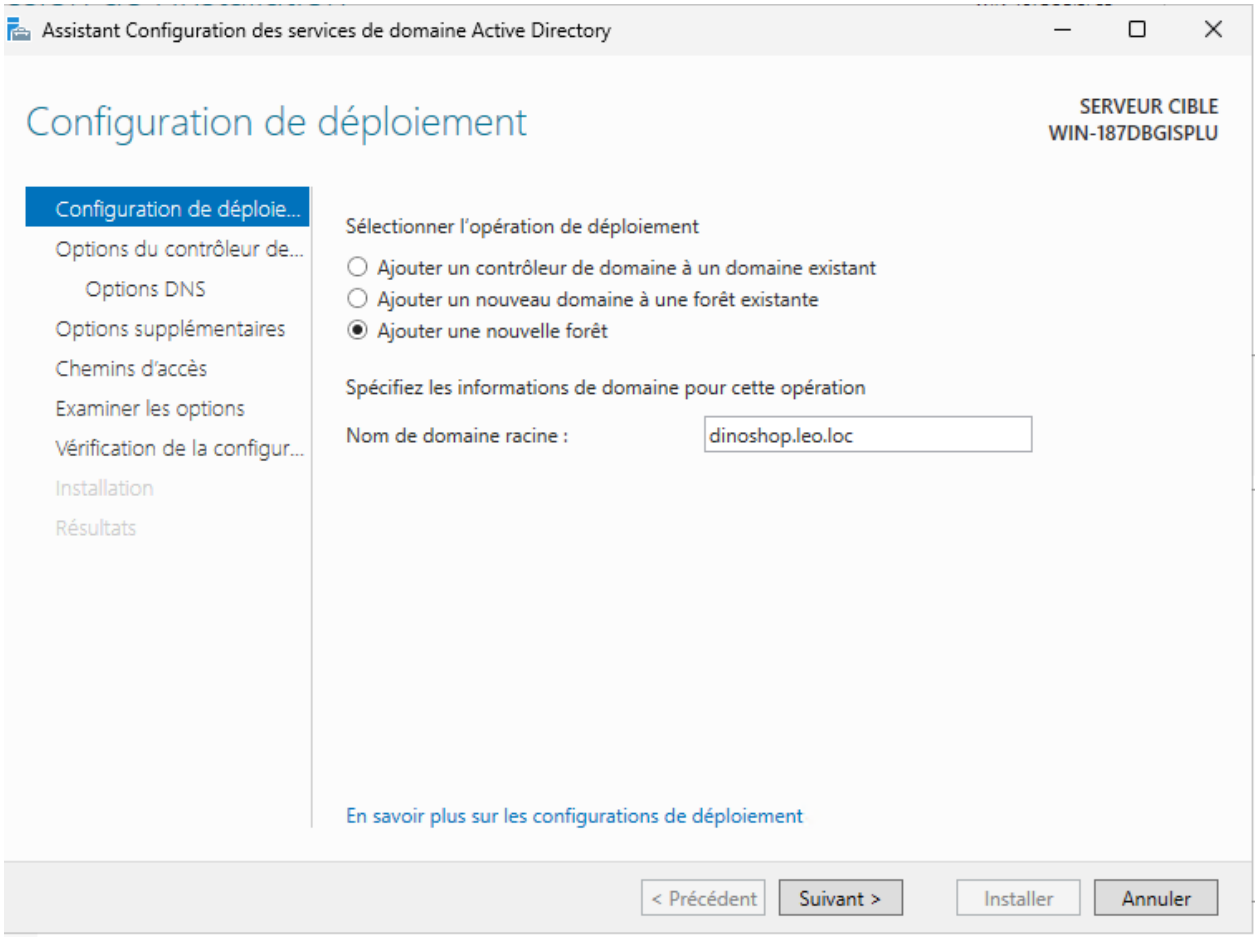
2.4 Active Directory Paris (AD principal)

Prétest : activer ICMP dans le pare-feu Windows pour vérifier que les deux AD se pinguent bien.

On crée une Windows Server sur le VLAN SRV, on lui affecte une IP fixe (rôle DNS), puis on choisit un nom de domaine unique, non routable et non public.



Installation du rôle Active Directory sur Windows Server



Promotion en contrôleur de domaine principal

2.5 Base de données et serveur web (WordPress)

Le serveur web WordPress est déployé dans la DMZ du site Paris (192.168.11.2). La base de données MariaDB est installée sur une machine séparée dans le VLAN SRV (192.168.10.6) pour des raisons de sécurité. Le serveur web Apache héberge l'application WordPress.

Installation MariaDB

```
sudo apt update
sudo apt install mariadb-server mariadb-client
sudo mysql_secure_installation

sudo mysql -u root -p
CREATE DATABASE wordpress CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'wpuser'@'%' IDENTIFIED BY 'mdp';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'%';
FLUSH PRIVILEGES;
EXIT;

# Écouter sur toutes les interfaces
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
bind-address = 0.0.0.0
```

Installation WordPress

```
sudo apt install apache2 php php-pgsql libapache2-mod-php
sudo wget https://fr.wordpress.org/latest-fr_FR.zip
sudo unzip latest-fr_Fr.zip && sudo rm latest-fr_Fr.zip
cd wordpress && sudo cp wp-config-sample.php wp-config.php

# wp-config.php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wpuser');
define('DB_PASSWORD', 'mdp');
define('DB_HOST', '192.168.10.6');
define('DB_CHARSET', 'utf8');
# VirtualHost Apache
sudo nano /etc/apache2/sites-available/wordpress.conf
<VirtualHost *:80>
    ServerName 192.168.11.2
    DocumentRoot /var/www/html/wordpress
    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

sudo a2ensite wordpress.conf
sudo a2dissite 000-default.conf
sudo systemctl restart apache2.service
```

2.6 GLPI

GLPI (Gestionnaire Libre de Parc Informatique) est déployé pour la gestion du parc informatique et le suivi des tickets d'incidents. Il est connecté à l'annuaire Active Directory via LDAP pour l'authentification des utilisateurs.

Installation LAMP + GLPI

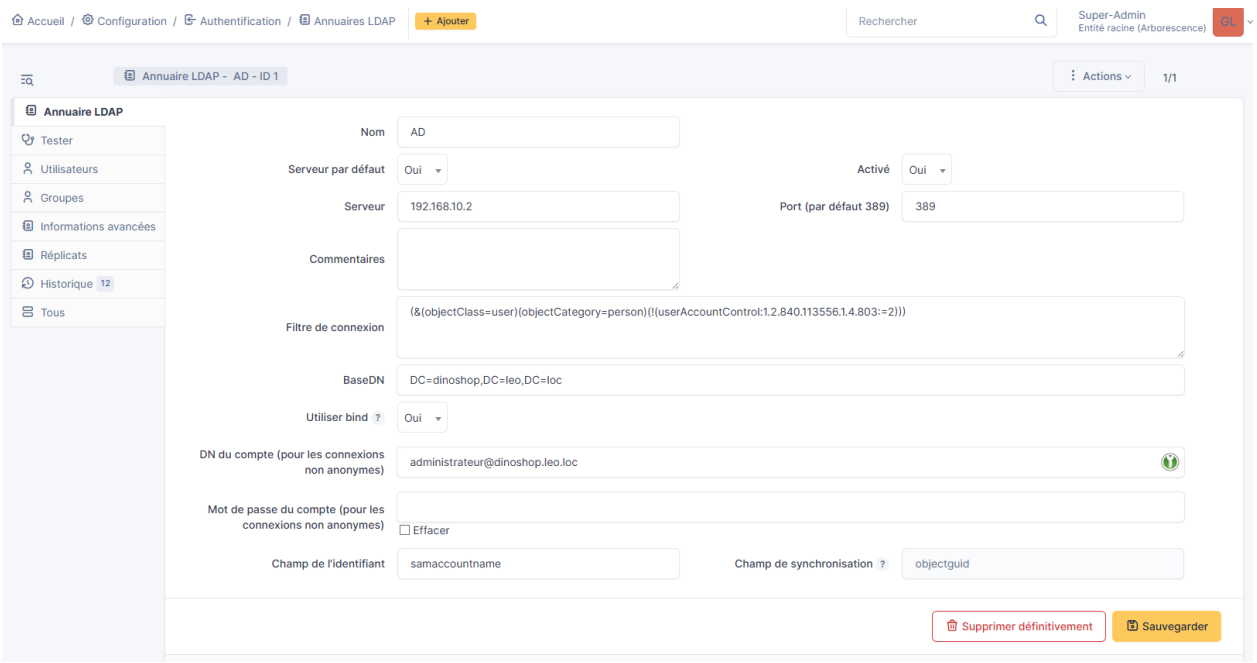
```
sudo apt-get install apache2 php mariadb-server
sudo apt-get install php-xml php-common php-json php-mysql php-mbstring php-curl
php-gd php-intl php-zip php-bz2

mysql_secure_installation
mysql -u root -p
CREATE DATABASE db_glpi;
GRANT ALL PRIVILEGES ON db_glpi.* TO glpi@localhost IDENTIFIED BY 'motdepasse';
FLUSH PRIVILEGES; EXIT;

cd /tmp && wget https://github.com/glpi-project/glpi/releases/download/11.0.2/glpi-
11.0.2.tgz
sudo tar -xzvf glpi-11.0.2.tgz -C /var/www/
chown www-data /var/www/glpi/ -R

mkdir /etc/glpi && chown www-data /etc/glpi/
mv /var/www/glpi/config /etc/glpi
mkdir /var/lib/glpi && chown www-data /var/lib/glpi/
mv /var/www/glpi/files /var/lib/glpi
mkdir /var/log/glpi && chown www-data /var/log/glpi
# /var/www/glpi/inc/downstream.php
<?php define('GLPI_CONFIG_DIR', '/etc/glpi/');
if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) { require_once
GLPI_CONFIG_DIR . '/local_define.php'; }

# /etc/glpi/local_define.php
<?php define('GLPI_VAR_DIR', '/var/lib/glpi/files'); define('GLPI_LOG_DIR',
'/var/log/glpi');
```

```
# VirtualHost Apache
<VirtualHost *:80>
    ServerName 172.16.130.140
    DocumentRoot /var/www/glpi/public
    <Directory /var/www/glpi/public>
        Require all granted
        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
</VirtualHost>

a2ensite glpi.conf && a2dissite 000-default.conf
a2enmod proxy_fcgi setenvif && a2enconf php8.2-fpm
systemctl reload apache2
rm /var/www/glpi/install/install.php
```

Connexion LDAP / Active Directory

Installation de l'extension LDAP pour PHP :

```
sudo apt-get install php-ldap
```



Connexion GLPI à l'annuaire LDAP (Active Directory)

Import des utilisateurs depuis l'AD dans GLPI

2.7 Serveur de fichiers (Nextcloud)

Nextcloud est déployé comme serveur de fichiers pour permettre le partage et la synchronisation de fichiers au sein de l'infrastructure. Il est installé sur le VLAN SRV du site Paris (192.168.10.7) avec une base de données MariaDB dédiée.

```
sudo apt install apache2 mariadb-server libapache2-mod-php php php-
{gd,xml,mbstring,curl,zip,intl,cli,mysql}
sudo a2enmod rewrite headers env dir mime ssl
sudo systemctl restart apache2

sudo mysql -u root
CREATE DATABASE nextcloud CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'nextclouduser'@'localhost' IDENTIFIED BY 'mdp';
GRANT ALL PRIVILEGES ON nextcloud.* TO 'nextclouduser'@'localhost';
FLUSH PRIVILEGES; EXIT;

cd /tmp && wget https://download.nextcloud.com/server/releases/nextcloud-32.0.2.zip
sudo mv nextcloud /var/www/html/
sudo chown -R www-data:www-data /var/www/html/nextcloud
sudo chmod -R 755 /var/www/html/nextcloud
<VirtualHost *:80>
    ServerName 192.168.10.7
    DocumentRoot /var/www/html/nextcloud
    <Directory /var/www/html/nextcloud/>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

sudo a2ensite nextcloud.conf && sudo systemctl reload apache2
```

2.8 Supervision : Prometheus et Grafana

La supervision de l'infrastructure est assurée par Prometheus pour la collecte des métriques et Grafana pour la visualisation. L'installation est automatisée via un playbook Ansible. Les machines Linux sont supervisées via Node Exporter, les serveurs Windows via Windows Exporter, et les switchs via SNMP Exporter.

Playbook Ansible — Installation Prometheus & Grafana

```
---
- name: Installer et configurer Prometheus et Grafana
  hosts: targets
  become: true
  vars:
    version_prometheus: "2.47.0"
    architecture_prometheus: "linux-amd64"
  tasks:
    - name: Creer l'utilisateur prometheus
      user:
        name: prometheus
        shell: /bin/false
        system: yes
        create_home: no

    - name: Creer les repertoires necessaires
      file:
        path: "{{ item.path }}"
        state: directory
        owner: prometheus
        group: prometheus
        mode: "0755"
      loop:
```

```
        - { path: "/etc/prometheus" }
        - { path: "/var/lib/prometheus" }
        - { path: "/opt/prometheus" }

    - name: Telecharger Prometheus
      get_url:
        url: "https://github.com/prometheus/prometheus/releases/download/..."
        dest: "/opt/prometheus.tar.gz"

    - name: Configurer le service systemd Prometheus
      copy:
        dest: /etc/systemd/system/prometheus.service
        content: |
          [Unit]
          Description=Prometheus Monitoring
          [Service]
          User=prometheus
          ExecStart=/usr/local/bin/prometheus \
            --config.file /etc/prometheus/prometheus.yml \
            --storage.tsdb.path /var/lib/prometheus/
          [Install]
          WantedBy=multi-user.target

    - name: Ajouter le depot Grafana
      apt_repository:
        repo: "deb https://packages.grafana.com/oss/deb stable main"
        state: present

    - name: Installer Grafana
      apt:
        name: grafana
        state: latest

    - name: Configurer datasource Prometheus dans Grafana
      copy:
        dest: /etc/grafana/provisioning/datasources/prometheus.yml
        content: |
          apiVersion: 1
          datasources:
            - name: Prometheus
              type: prometheus
              url: http://localhost:9090
              isDefault: true

ansible-playbook -i host.ini playbook.yml
```

Node Exporter (Linux)

```
sudo useradd --no-create-home --shell /bin/false node_exporter
sudo wget
https://github.com/prometheus/node_exporter/releases/download/v1.10.2/node_exporter
-1.10.2.linux-amd64.tar.gz
sudo tar -xvf node_exporter-1.10.2.linux-amd64.tar.gz
sudo mv node_exporter-1.10.2.linux-amd64/node_exporter /usr/local/bin/

# /etc/systemd/system/node_exporter.service
[Unit]
Description=Node Exporter
[Service]
User=node_exporter
ExecStart=/usr/local/bin/node_exporter
[Install]
WantedBy=multi-user.target

sudo systemctl daemon-reload
sudo systemctl enable --now node_exporter
```

```
# prometheus.yml — ajout de la cible
scrape_configs:
  - job_name: 'machines-linux'
    static_configs:
      - targets:
          - '192.168.12.2:9100'
sudo systemctl restart prometheus
```

Windows Exporter (Active Directory)

```
# Télécharger depuis :
# https://github.com/prometheus-community/windows_exporter/releases/latest

sc.exe create windows_exporter binPath= "C:\windowsexport\windows_exporter-0.31.3-
amd64.exe --collectors.enabled=..."
sc.exe start windows_exporter
New-NetFirewallRule -DisplayName "Windows Exporter 9182" -Direction Inbound -
Protocol TCP -LocalPort 9182 -Action Allow
```

SNMP Exporter (Switch Juniper)

```
# Vérification SNMP sur le switch
snmpwalk -v2c -c moni 192.168.10.254 1.3.6
# iso.3.6.1.2.1.1.1.0 = STRING: "Juniper Networks, Inc. ex2300-24p ..."

# Configuration snmp.yaml
auths:
  moni_auth:
    community: moni
    version: 2

# prometheus.yml — job SNMP
- job_name: 'junos_switches'
  scrape_interval: 60s
  static_configs:
    - targets:
        - 192.168.10.254
  metrics_path: /snmp
  params:
    module: [juniper]
    auth: [moni_auth]
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - target_label: __address__
      replacement: 127.0.0.1:9116
```

3. Site Honfleur

3.1 Linux Bridge (Proxmox 2)

Le serveur Proxmox 2 (px2.labo.loc) héberge l'ensemble des machines virtuelles du site Honfleur. La configuration réseau repose sur des Linux Bridges pour segmenter le trafic réseau.

DMZ		Linux Bridge	Yes	Yes	No	enp3s0.11			VLAN11
LAN		Linux Bridge	Yes	Yes	No	enp3s0.12			VLAN12
SRV		Linux Bridge	Yes	Yes	No	enp3s0.10			VLAN10
WAN		Linux Bridge	Yes	Yes	No	enp3s0.13			VLAN13
enp3s0	enx00151715a237	Network Device	Yes	No	No				
enp4s0	enx7446a0a036a5	Network Device	Yes	No	No				
vmbr0		Linux Bridge	Yes	Yes	No	enp4s0	172.16.120.83/16	172.16.255.254	

Configuration des bridges réseau sur Proxmox 2

3.2 Routeur / Firewall / VPN Honfleur

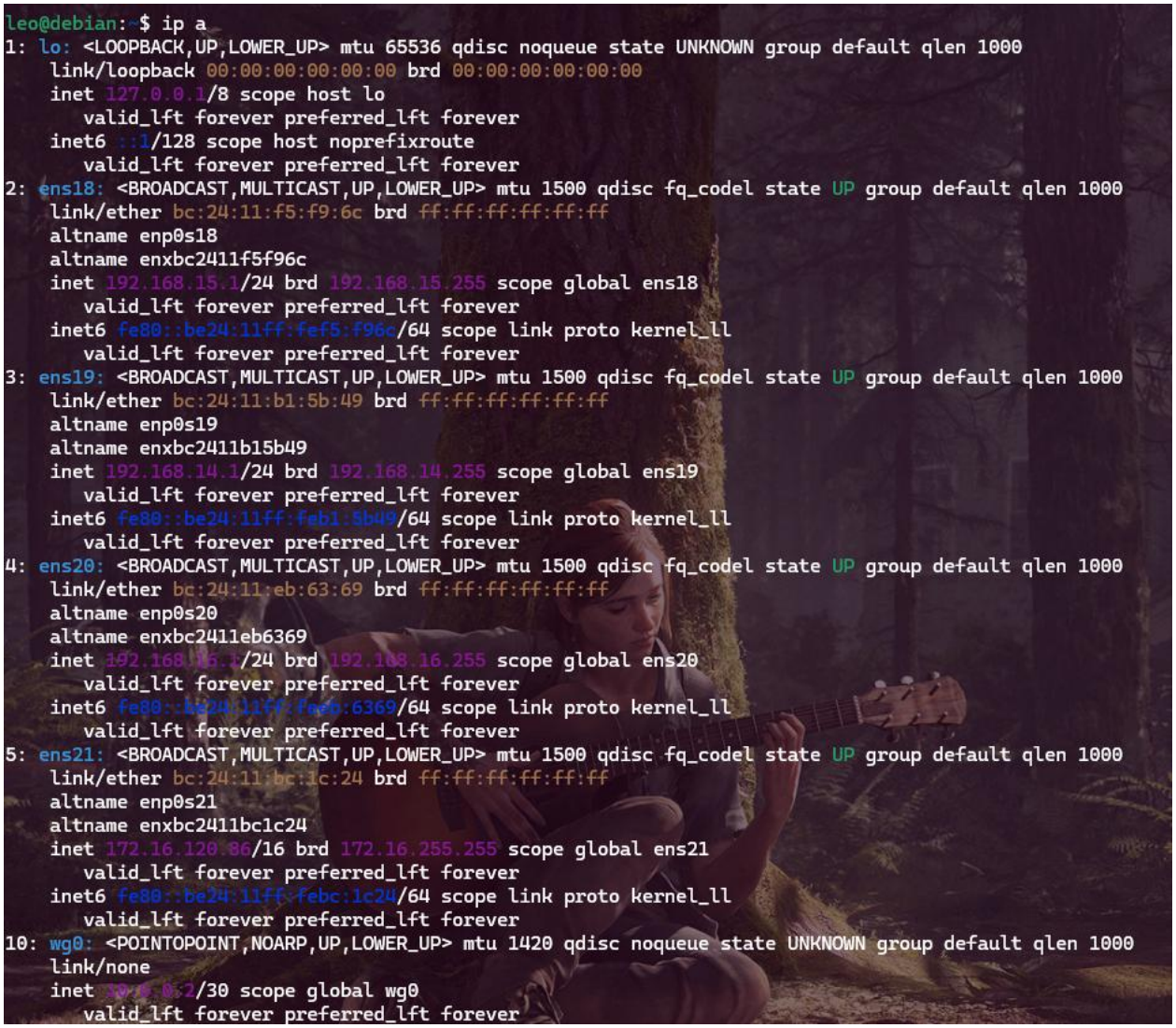
Partie routage / interfaces réseaux

On met en place tous les VLANs sur 4 cartes réseaux sur la VM :

Memory	3.00 GiB
Processors	2 (1 sockets, 2 cores) [kvm64]
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	local:iso/pfSense-CE-2.6.0-RELEASE-amd64.iso,media=cdrom,size=749476K
Hard Disk (scsi0)	local-lvm:vm-100-disk-0,iothread=1,size=20G
Network Device (net0)	virtio=BC:24:11:5D:47:24,bridge=DMZ,firewall=1
Network Device (net1)	virtio=BC:24:11:EC:C3:57,bridge=LAN,firewall=1
Network Device (net2)	virtio=BC:24:11:6B:77:F7,bridge=SRV,firewall=1
Network Device (net3)	virtio=BC:24:11:73:15:54,bridge=WAN,firewall=1

Configuration des 4 interfaces réseau — VM routeur Honfleur

Configuration des interfaces via /etc/network/interfaces :



Configuration /etc/network/interfaces — Honfleur

Activation du forwarding IPv4 :

```
net.ipv4.ip_forward=1
```

Partie VPN WireGuard — Honfleur

```
sudo su -
apt install wireguard wireguard-tools
cd /etc/wireguard && umask 077
wg genkey | tee privatekey_honfleur | wg pubkey > publickey_honfleur
[Interface]
# Configuration Honfleur
PrivateKey = 8KQ4Gp3gYpi5GYlUhOWMtcvLyAk+fqj9RumW/0wT3Ho=
Address = 10.0.0.2/30
ListenPort = 51820

[Peer]
# Paris
PublicKey = gaWxqs0nDBGepADHZrgNzPorxC2/TninC9SK2wQ3BDg=
Endpoint = 172.16.120.85:51820
AllowedIPs = 10.0.0.1/32, 192.168.10.0/24, 192.168.11.0/24, 192.168.12.0/24
PersistentKeepalive = 25
sudo systemctl start wg-quick@wg0
sudo systemctl status wg-quick@wg0
```

Partie Firewall — Honfleur

```
sudo apt install iptables -y && sudo apt install netfilter-persistent

# Règles FORWARD
Chain FORWARD:
  ACCEPT all -- ens18 ens21
  ACCEPT all -- ens19 ens21
  ACCEPT all -- ens21 ens18  state RELATED,ESTABLISHED
  ACCEPT all -- ens21 ens19  state RELATED,ESTABLISHED
  ACCEPT all -- ens19 wg0
  ACCEPT all -- wg0 ens19

# NAT
MASQUERADE all -- 192.168.14.0/24 anywhere
MASQUERADE all -- 192.168.15.0/24 anywhere
MASQUERADE all -- 192.168.16.0/24 anywhere
```

Vérification du handshake WireGuard côté Honfleur :

```
sudo wg show wg0
interface: wg0
  public key: bGpDBGlznQISOfhsCboaWlff+qitmT/3Kn+zDPLAnmE=
  listening port: 51820
peer: gaWxqs0nDBGepADHZrgNzPorxC2/TninC9SK2wQ3BDg=
  endpoint: 172.16.120.85:51820
  latest handshake: 1 minute, 41 seconds ago
  transfer: 22.41 KiB received, 8.11 KiB sent
  persistent keepalive: every 25 seconds
```

Tests de connectivité depuis le client LAN Honfleur (192.168.14.2) :

```
ping 192.168.14.1   # passerelle locale
ping 10.0.0.2       # IP VPN Honfleur
ping 10.0.0.1       # IP VPN Paris
ping 192.168.12.1   # LAN Paris
ping 192.168.12.2   # Client LAN Paris
```

Tous les paquets sont bien routés — le site-to-site fonctionne.

3.3 Switch HP

Modèle : HP A5500 Series Switch JG240A

Réinitialisation et configuration de base

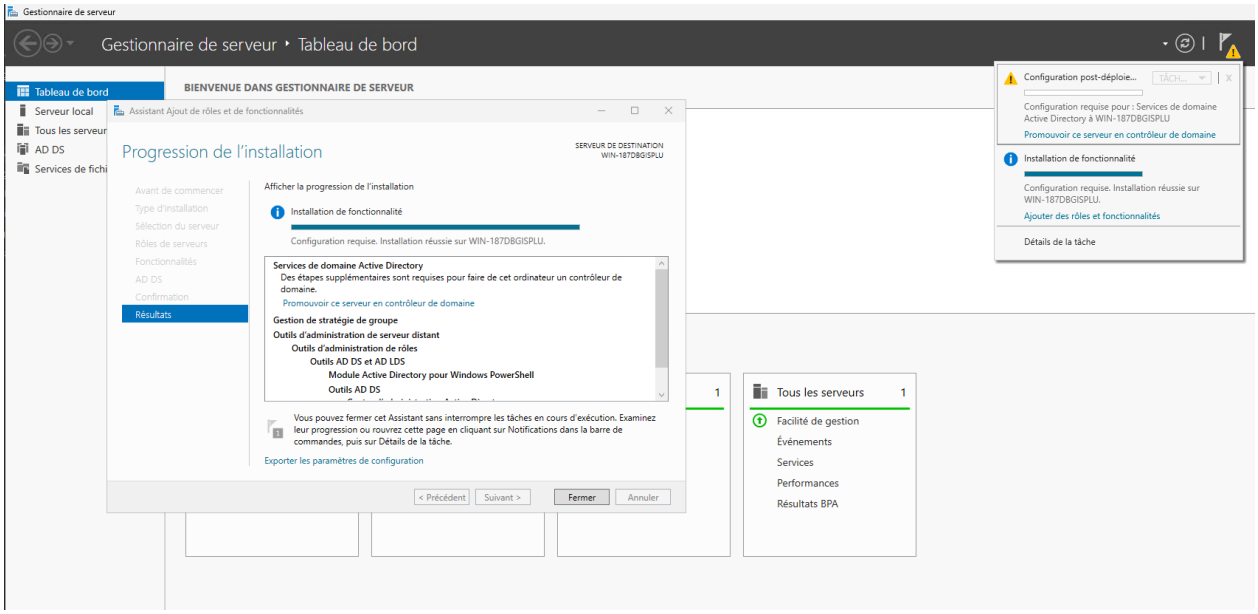
```
reset saved-configuration
Y → reboot → N → Y

# IP de management
system-view
interface vlan-interface 1
ip address 172.16.120.84 255.255.0.0
quit

# Accès gestion web
local-user admin
service-type telnet
authorization-attribute level 3
password simple PASSWORD
quit
ip http enable

# Accès : http://172.16.120.84
```

```
# Configuration VLANs
vlans SRV  vlan-id 10
vlans DMZ  vlan-id 11
vlans LAN  vlan-id 12
vlans WAN  vlan-id 13
```

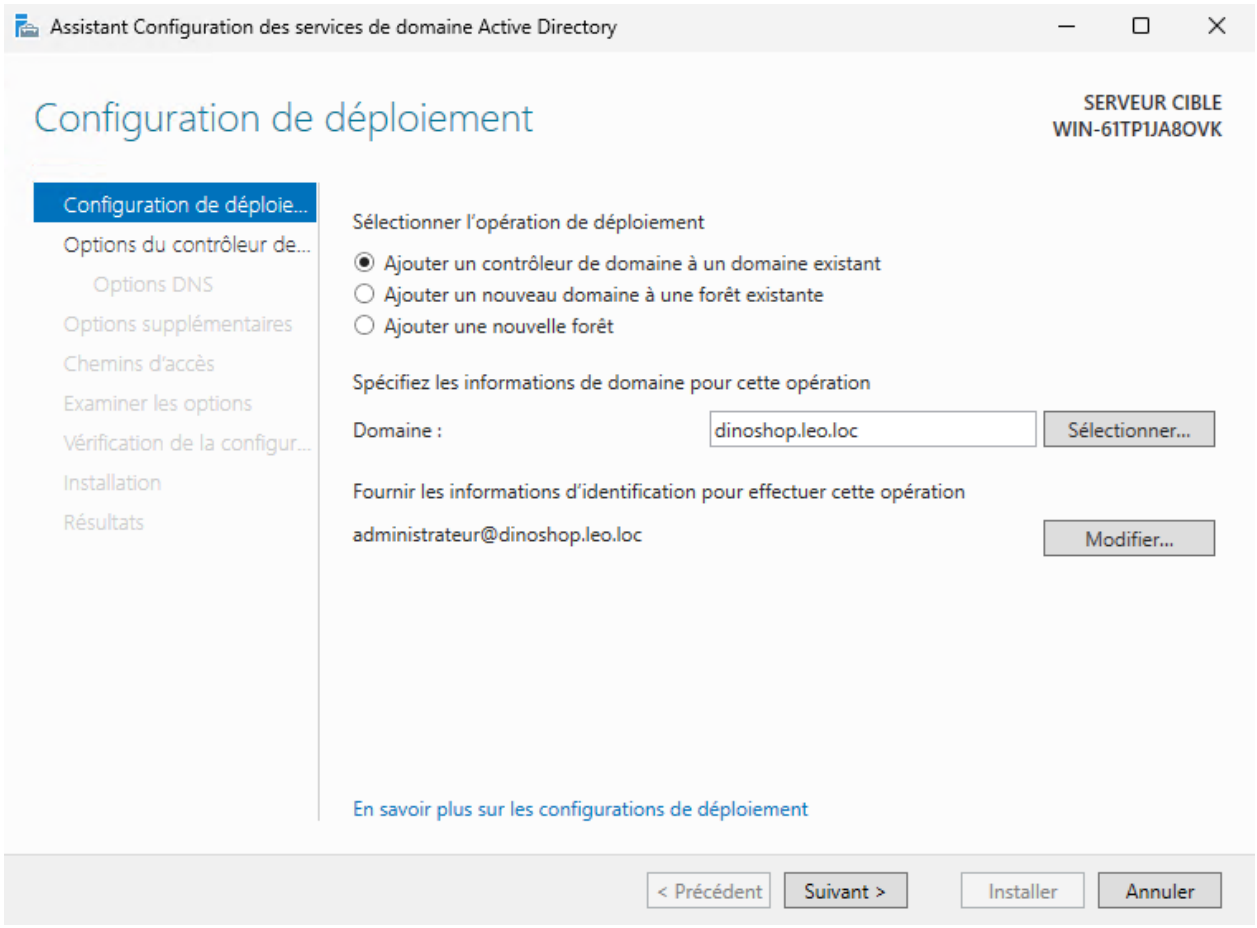
3.4 Active Directory Honfleur (AD secondaire)

Prétest : activer ICMP dans le pare-feu Windows pour vérifier que les deux AD se pinguent bien.

On crée une Windows Server sur le VLAN SRV avec une IP fixe. Lors de la promotion, il faut impérativement renseigner l'IP de l'AD primaire (Paris) comme serveur DNS sur la carte réseau, et rejoindre la forêt existante.



Installation du rôle Active Directory sur Windows Server



Promotion en contrôleur de domaine secondaire — rejoindre la forêt existante