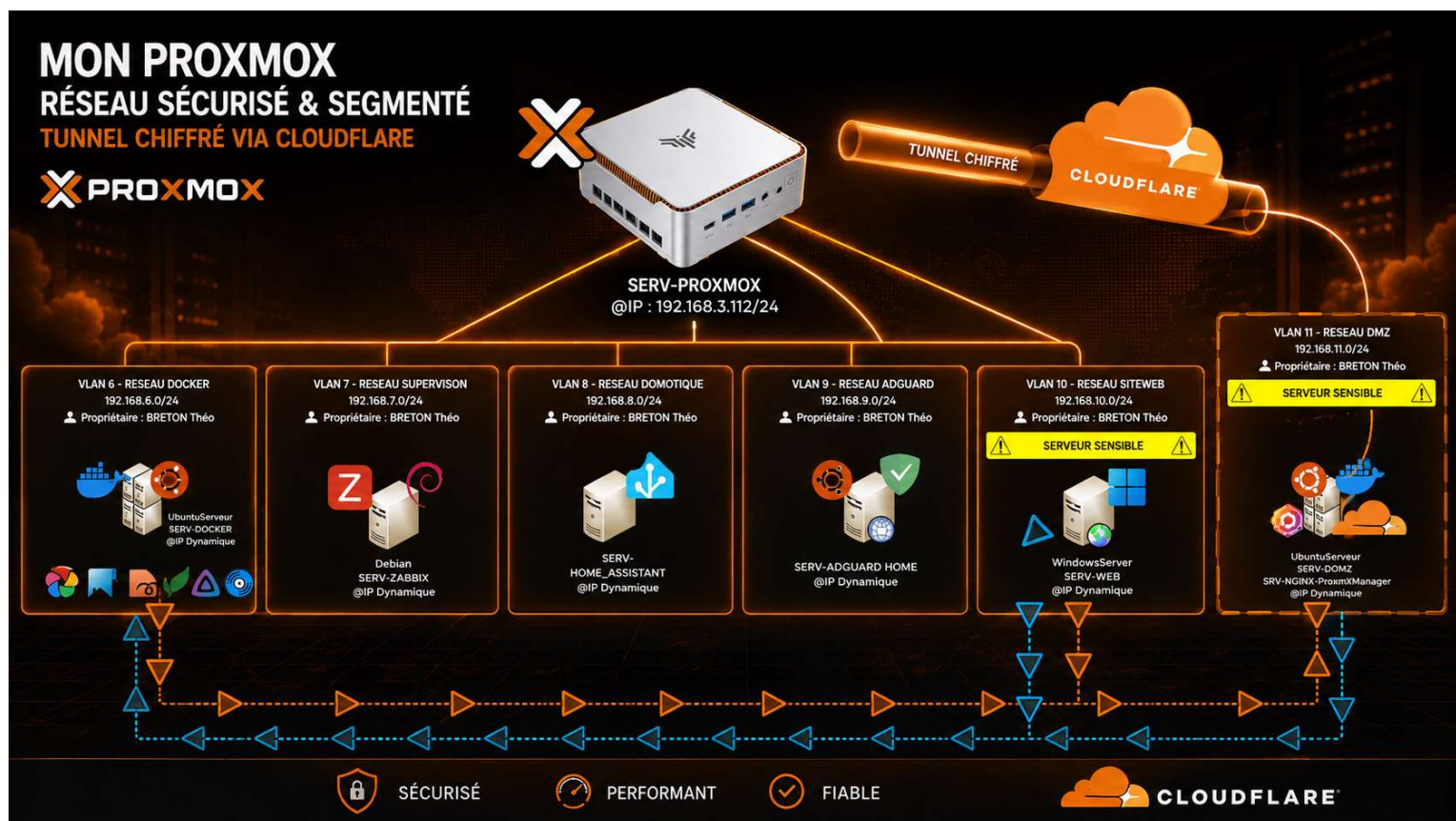


BRETON Théo



PROJET N°9 - HEBERGEMENT DE MA PLATEFORME, AVEC DIFFUSION DE MES SERVICES DOCKER



Contexte :

Dans le cadre de mon cursus en informatique, j'ai initialement développé un portfolio académique pour centraliser mes premiers travaux de formation. Cependant, au fil de mon parcours, la multiplication de mes projets personnels et l'affirmation de mon identité technique ont rendu les limites de cette plateforme scolaire contraignantes.

C'est dans cette dynamique d'évolution qu'est né le projet de concevoir une plateforme web entièrement indépendante. L'objectif de ce site est double : s'affranchir des cadres préétablis pour acquérir une totale autonomie technique, et concevoir un hub unique capable de centraliser, valoriser et pérenniser l'ensemble de mes réalisations.

Une infrastructure robuste et sécurisée Pour soutenir cette ambition, j'ai conçu une architecture réseau exigeante, représentative de mon affirmation en tant que futur professionnel. Au cœur de cette plateforme, j'ai déployé un écosystème de services auto-hébergés incluant Immich (pour la gestion intelligente de médias) et Jellyfin (pour la diffusion multimédia).

La gestion du trafic entrant est assurée par Nginx Proxy Manager, orchestré derrière un tunnel Cloudflare afin d'optimiser la sécurité et la performance des accès. L'ensemble est rigoureusement cloisonné au sein d'une DMZ (Zone Démilitarisée) sur mon pare-feu Fortinet, garantissant une séparation stricte entre les flux publics et mon réseau interne.

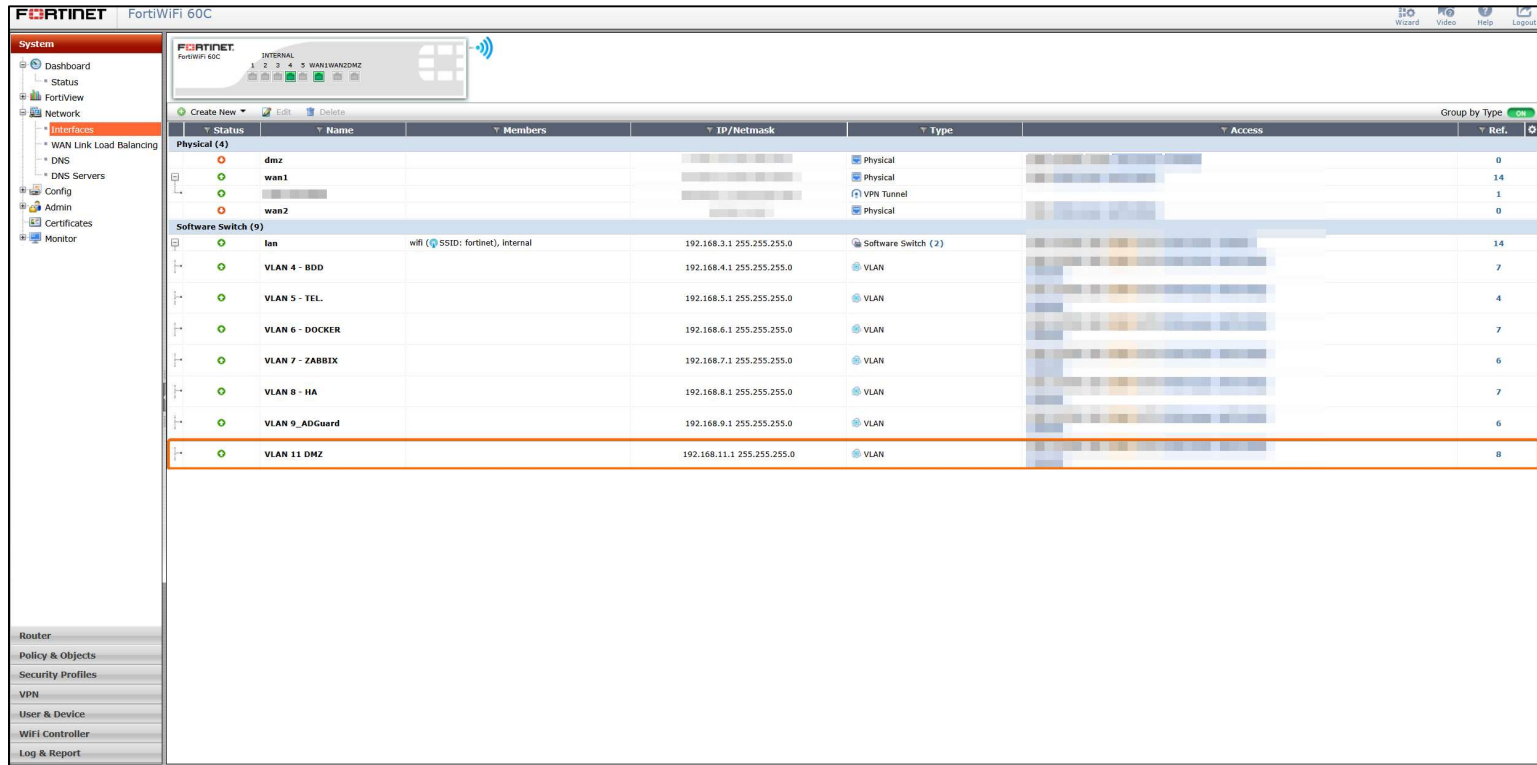
Ce projet marque ainsi la transition entre mon statut d'étudiant et mon entrée dans le monde professionnel, démontrant ma capacité à concevoir, sécuriser et maintenir des architectures systèmes complexes.

PARTIE 1 : MISE EN PLACE D'UN SERVEUR REVERSE PROXY ET D'UN TUNNEL CLOUDFLARE

Etape 1 : Création du VLAN pour la DMZ

```
SW-HOMELAB_THEO(config)#vlan 11
SW-HOMELAB_THEO(config-vlan)#name Reseau_DMZ
SW-HOMELAB_THEO(config-vlan)#exit
```

Etape 2 : Création de la sous interface sur le pare-feu pour le VLAN



Interface	Name	Members	IP/Netmask	Type	Access	Ref.
Physical (4)						
dmz				Physical		0
wan1				Physical		14
wan2				Physical		1
Software Switch (9)						
lan	wifi (SSID: fortinet), internal		192.168.3.1 255.255.255.0	Software Switch (2)		14
VLAN 4 - BDD			192.168.4.1 255.255.255.0	VLAN		7
VLAN 5 - TEL			192.168.5.1 255.255.255.0	VLAN		4
VLAN 6 - DOCKER			192.168.6.1 255.255.255.0	VLAN		7
VLAN 7 - ZABBIX			192.168.7.1 255.255.255.0	VLAN		6
VLAN 8 - HA			192.168.8.1 255.255.255.0	VLAN		7
VLAN 9_ADGuard			192.168.9.1 255.255.255.0	VLAN		6
VLAN 11 DMZ			192.168.11.1 255.255.255.0	VLAN		8

Une fois la partie VLAN réalisés je me suis occupée de l'intégration d'un nouveau serveur, qui aura pour rôle d'être un reverse proxy.

Notions apprises :

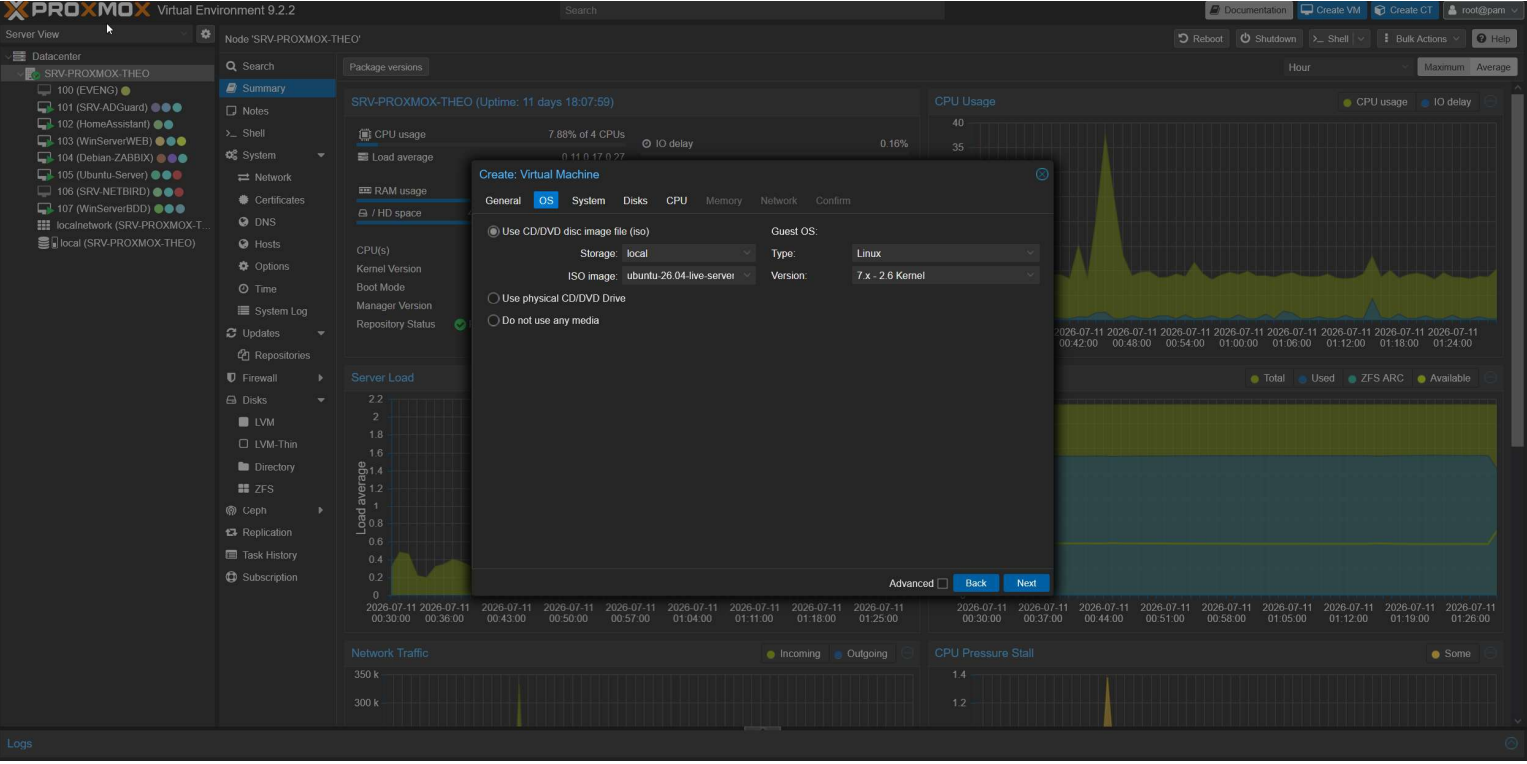
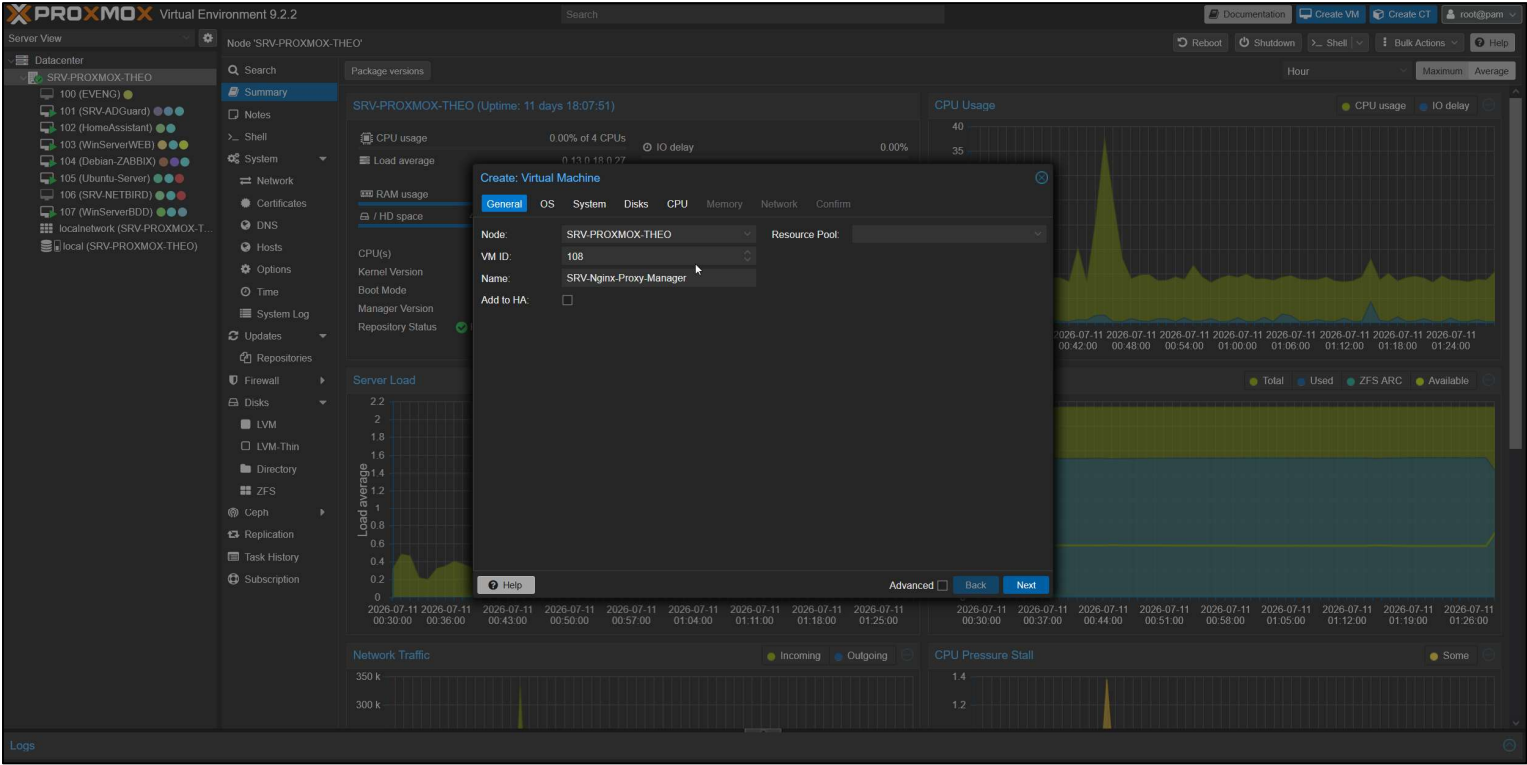
Un proxy qu'est que c'est exactement ?

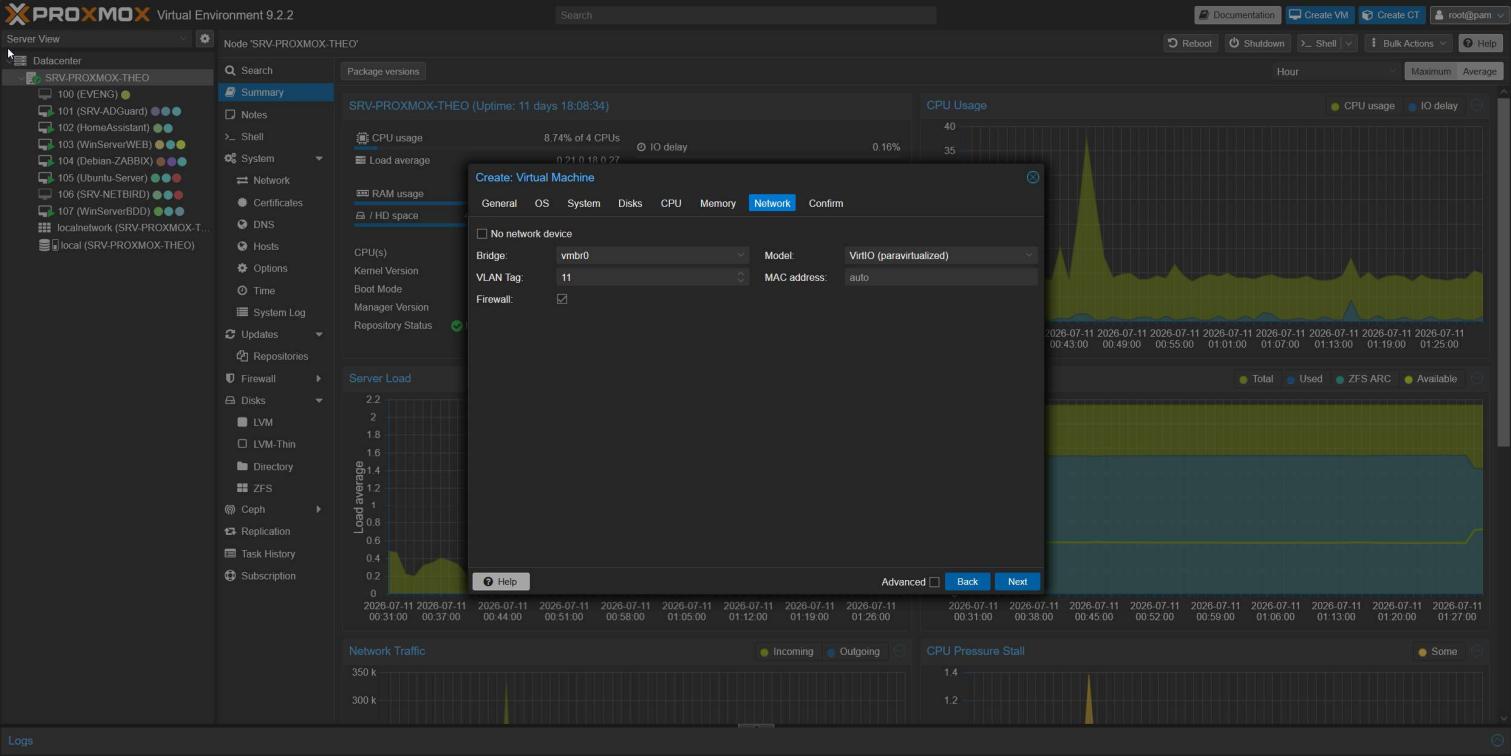
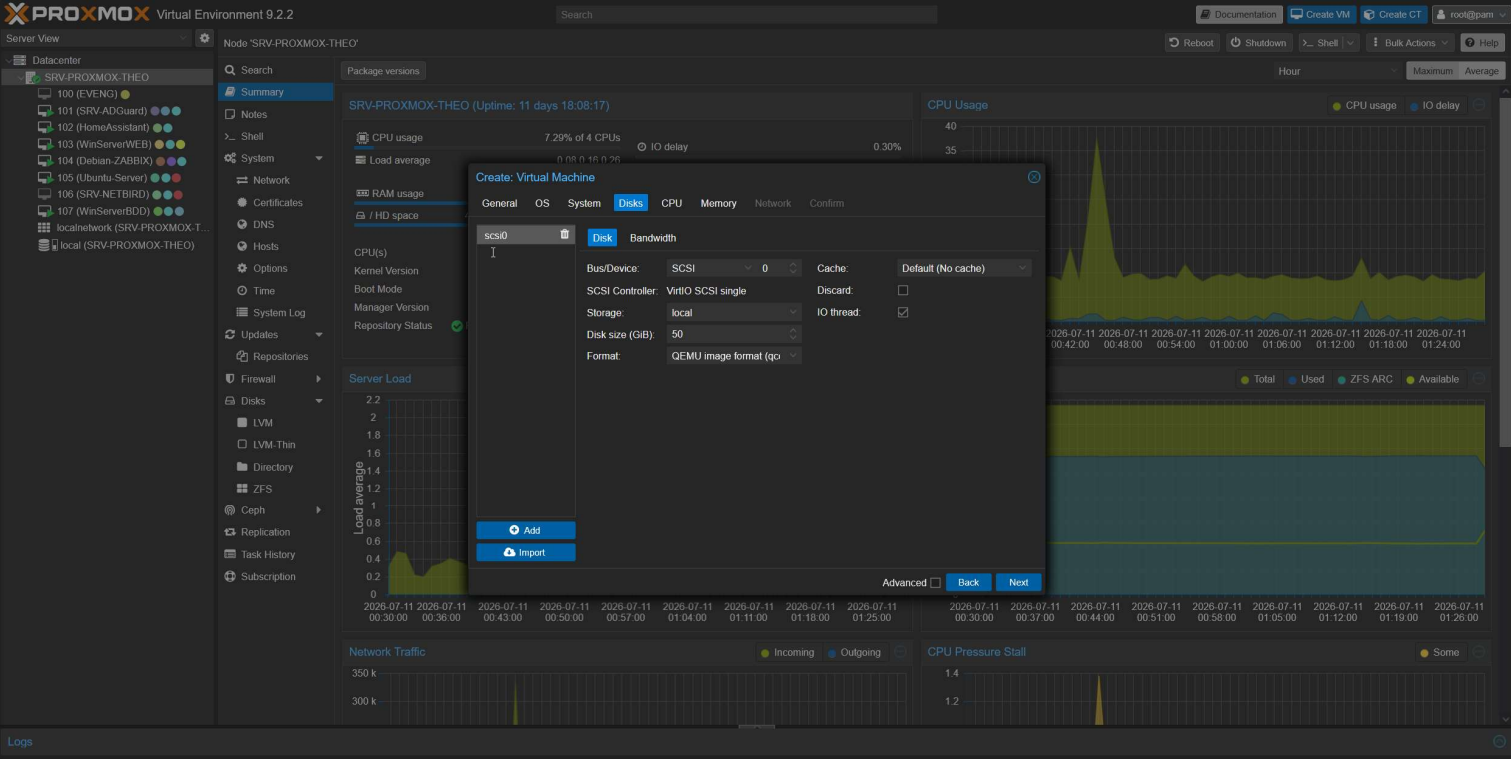
i Un proxy classique va agir comme intermédiaire entre l'utilisateur et un serveur client. Un client qui va essayer d'accéder à ressources externe.

Un reverse proxy qu'est que c'est exactement ?

i Un reverse proxy va agir toujours entre l'utilisateur est un serveur client. Mais cette fois le client va essayer d'accéder à ressource situé en interne.

Etape 3 : Installation de la machine virtuelle pour serveur Nginx Proxy Manager





Etape 3 : Exposition de la DMZ sur Internet :

J'ai créé deux règle filtrage pour l'exposition de la DMZ sur Internet

1^{ère} règle : Elle autorise le VLAN 11 – DMZ à communiquer avec le réseau WAN.

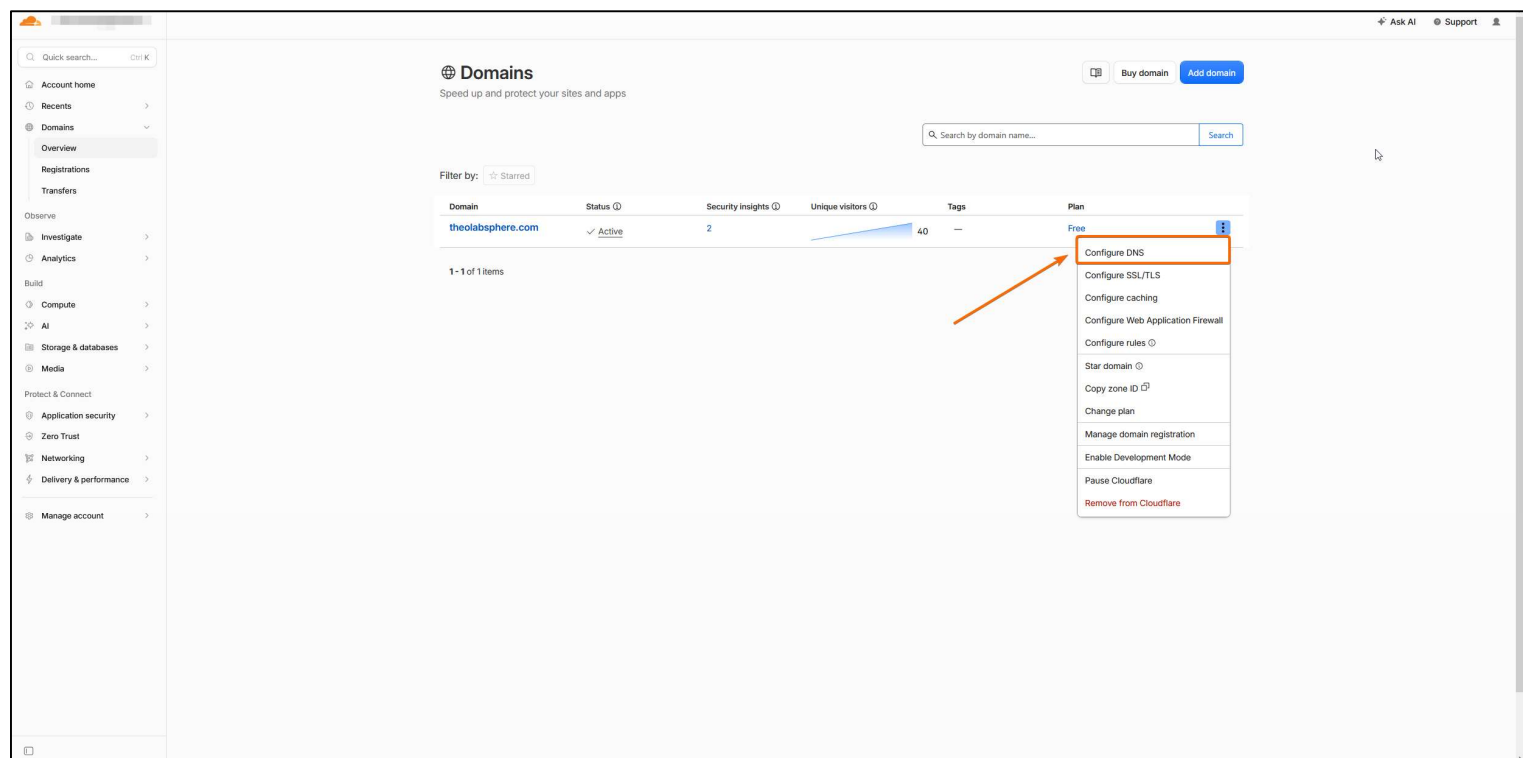
2^{ème} règle : Elle va assurer le retour, elle autorise le WAN à communiqué avec la DMZ.

Donc le trajet aller-retour et assuré par ces deux règles de filtrage.

Etape 4 : Configuration DNS sur Cloudflare :

J'ai acquis un domaine pour mon homelab qui est héberger sur Cloudflare.

Pour commencer la configuration j'ai créé sur le Dashbord de Cloudflare dans la configuration DNS du domaine met « ProxyHost »



Création des enregistrements DNS de type A :

J'ai laissé l'enregistrement des ProxyHost avec le mode « Proxied » pour mettre à Cloudflare de répondre avec sa propre @IP plutôt qu'avec la @IP du serveur.

DNS records for theolabsphere.com

DNS Setup: FullDNS documentation

Manage how the Internet finds your web content, verifies services, and routes traffic.

Recommendations

Search DNS RecordsFiltersDisplay optionsImportExportAdd record

2 records

Name	Type	Content	Proxy status	TTL	Tags	Comment	Details
.theolabsphere.com	Tunnel	HomeLab_Théo	Proxied	Auto			Close

Name

Type

Target

Proxy status

TTL

CNAME

Proxied

Auto

Record Attributes

SaveCancelDelete

Name

Type

Target

Proxy status

TTL

CNAME

Proxied

Auto

Record Attributes

SaveCancelDelete

Showing 1-2 of 2

Etape 7 : Installation de Nginx Proxy Manager :

❶ Installation de Docker et Docker Compose

Mise à jour et installation des dépendances

```
sudo apt update && sudo apt install -y curl docker.io docker-compose
```

Démarrage et activation du service Docker

```
sudo systemctl enable --now docker
```

❷ Configuration de Nginx Proxy Manager

Création du répertoire de projet

```
mkdir -p ~/docker/npm && cd ~/docker/npm
```

Création du fichier de configuration

```
nano docker-compose.yml
```

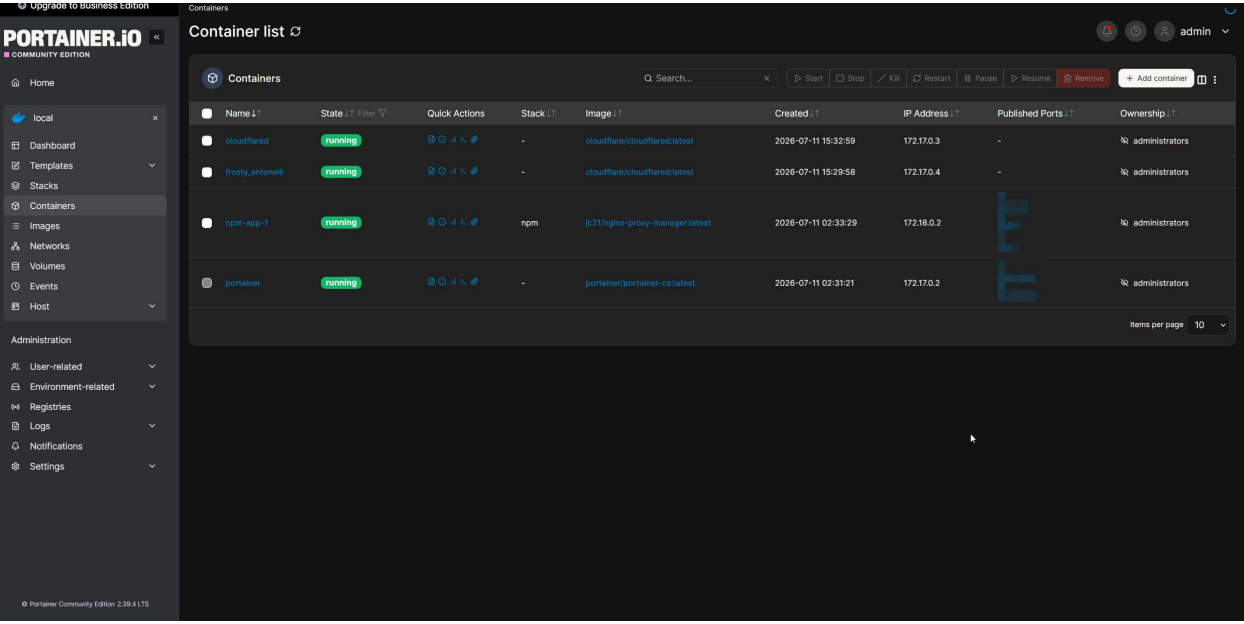
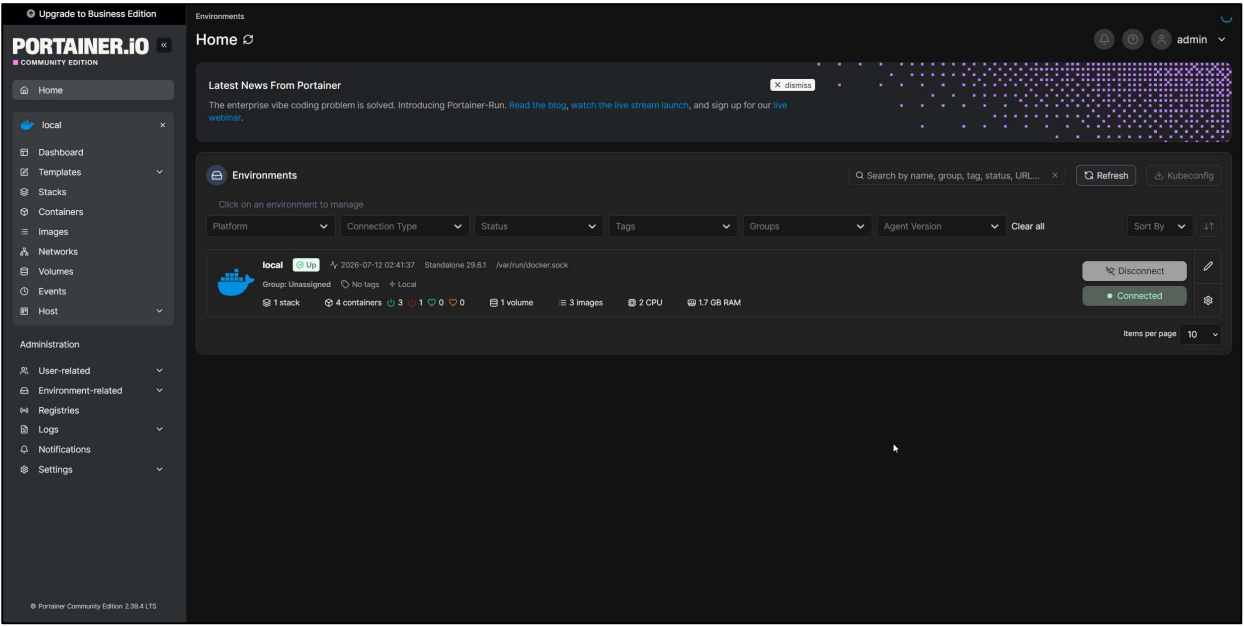
Contenu du fichier docker-compose.yml :

```
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      - '80:80'    # Port HTTP
      - '81:81'    # Interface d'administration
      - '443:443'  # Port HTTPS
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

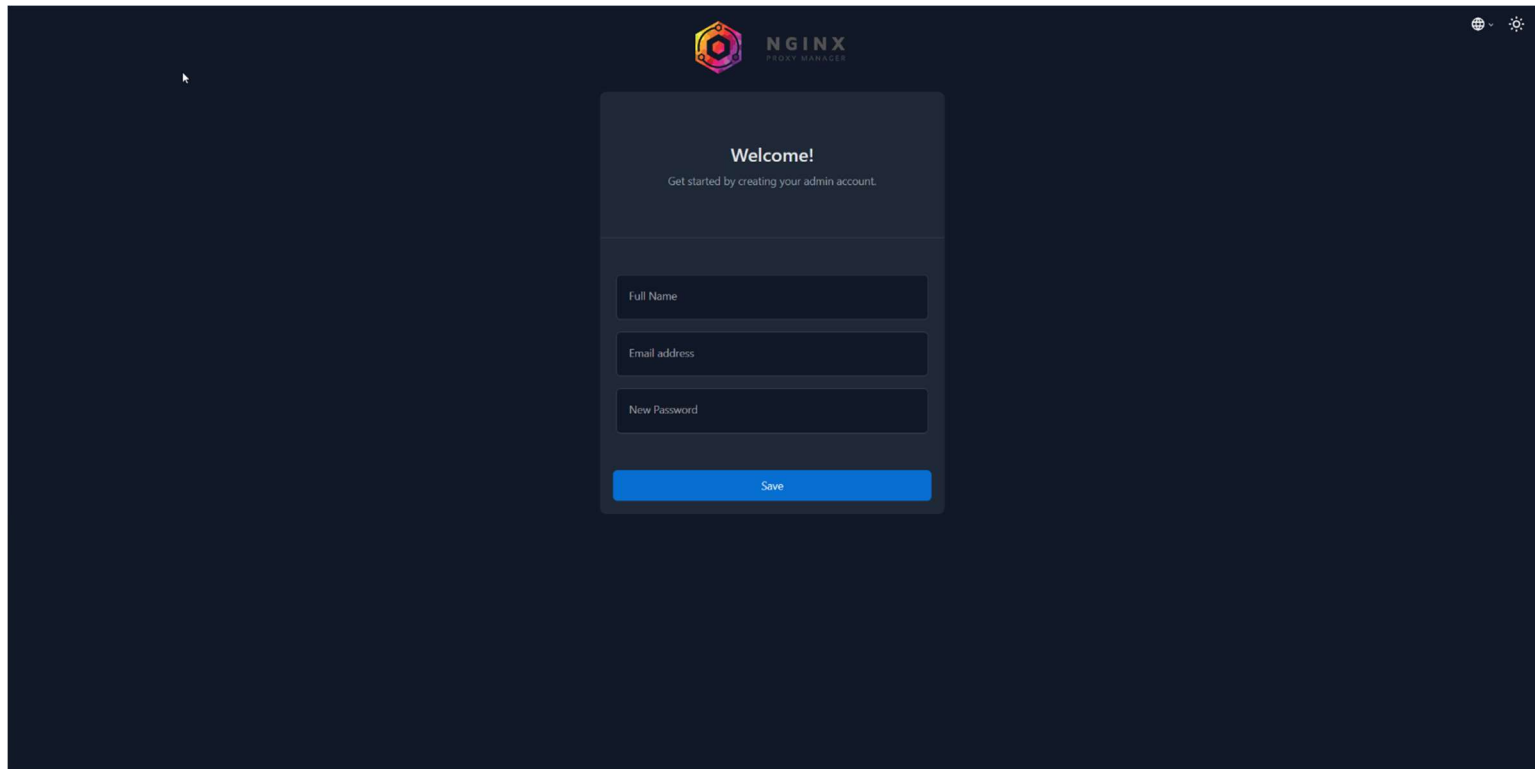
Lancement du service :

```
docker-compose up -d
```

Aperçu dans PORTAINER :

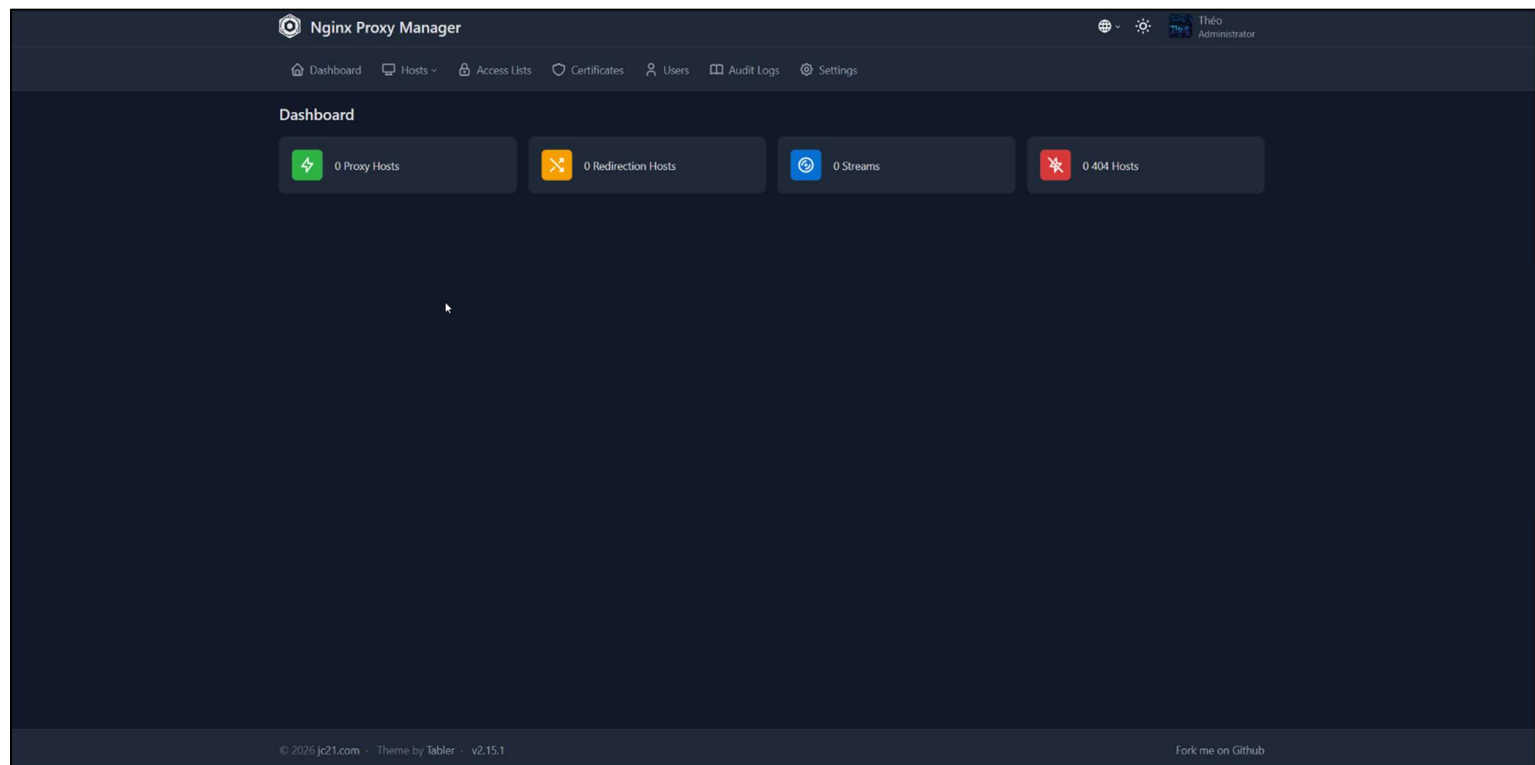


Une fois les conteneurs docker démarrer on peut aller sur l'interface de NGINX Proxy Manager :



Etape 6 : Configuration de Nginx Proxy Manager :

Création des enregistrements DNS de type A :

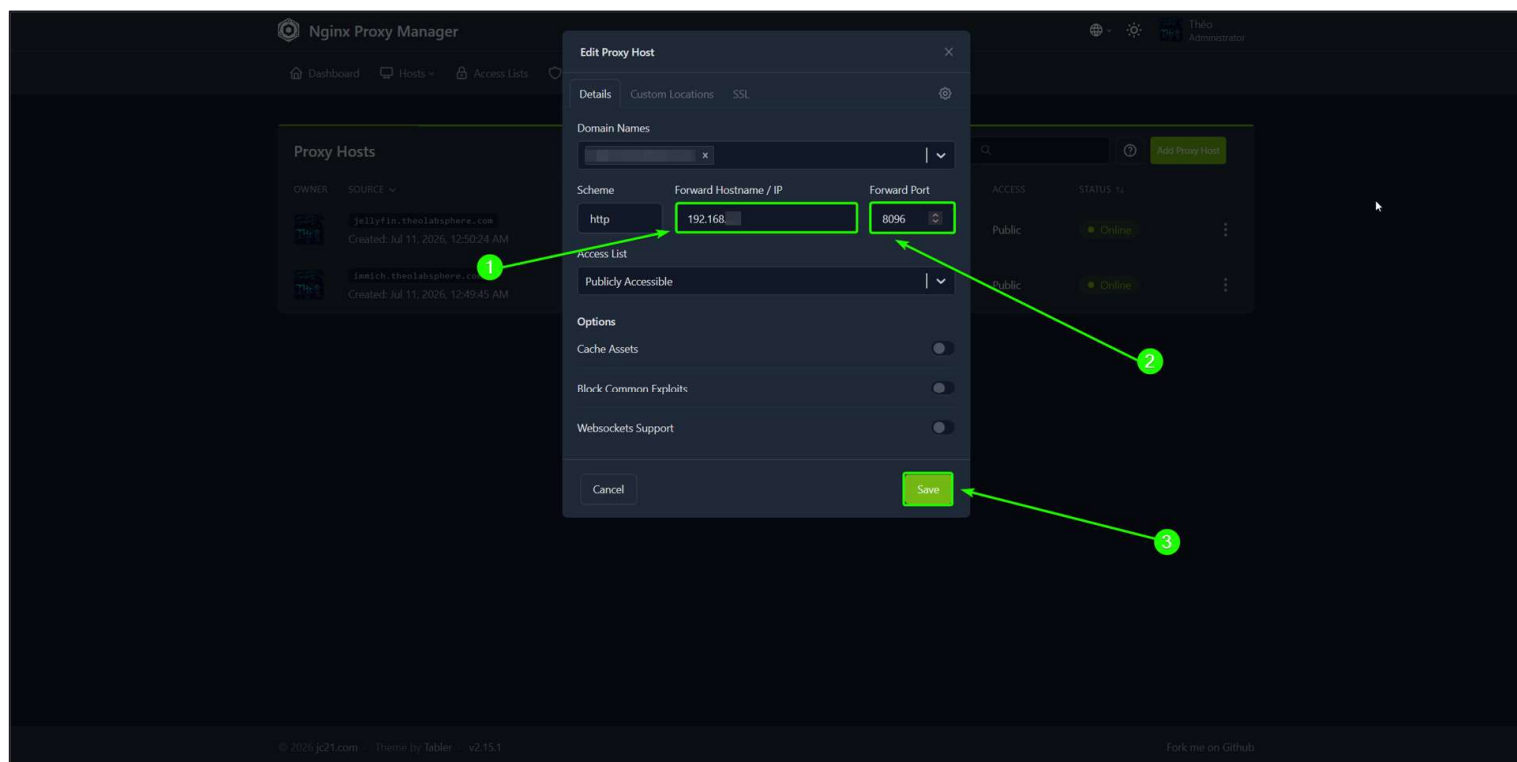




Jellyfin -



Immich :



Etape 7 : Sécurisation des deux règle filtrage :

C'est assurant la communication entre le VLAN et la DMZ doivent être restreinte au niveau des services :

Remplacement du service ALL par défaut, par ces deux services précis, réduisant la surface d'attaque en cas de compromission de NPM.

- IMMICH : TCP/2283
- JELLYFIN : TCP/8096

1 Création des services dans le pare feu.

System

Router

Policy & Objects

Policy

IPv4

Proxy Options

SSL Inspection

Objects

Addresses

Services

Schedules

Traffic Shapers

Virtual IPs

IP Pools

Monitor

FortiWiFi 60C

Create New Edit Delete Category Settings

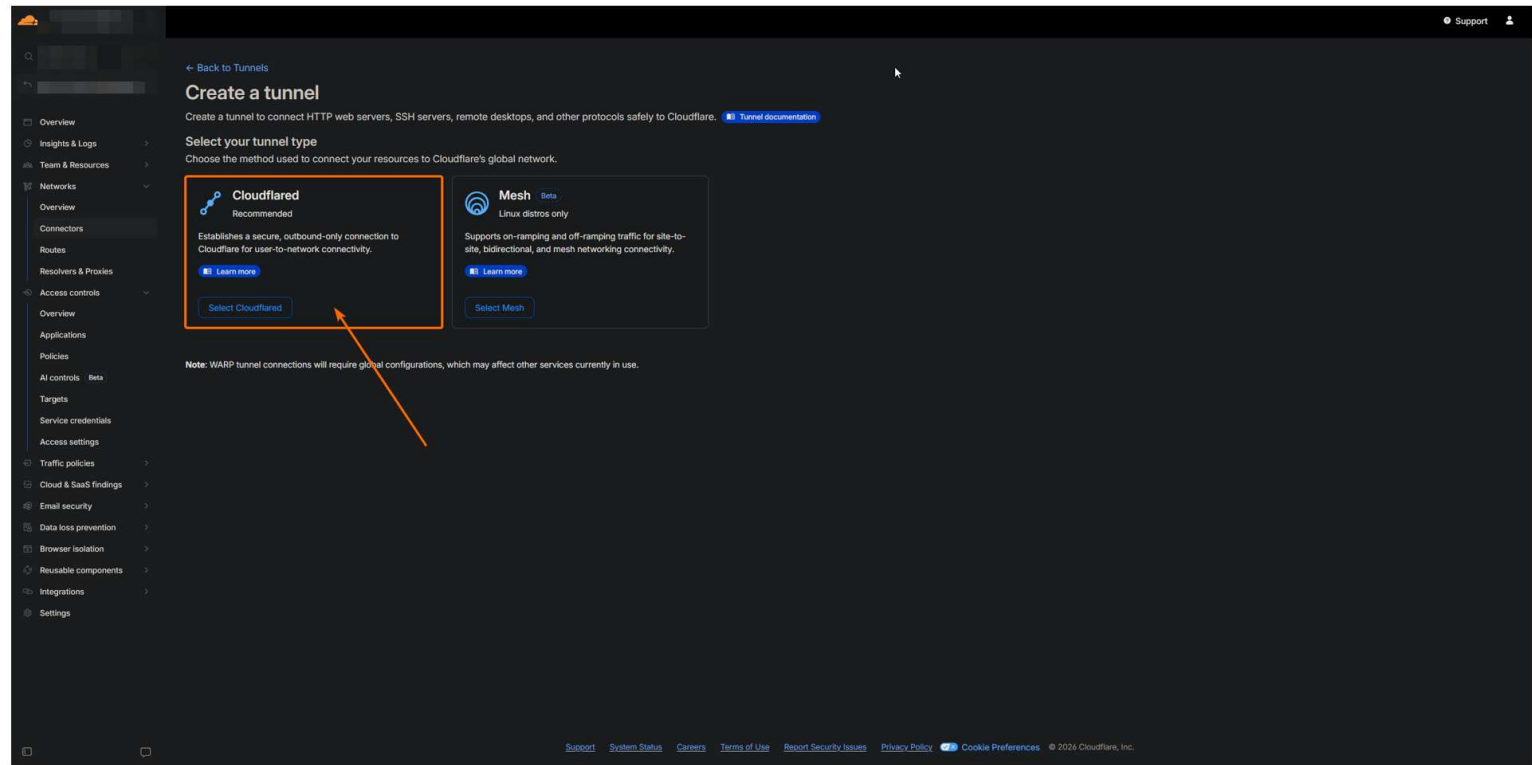
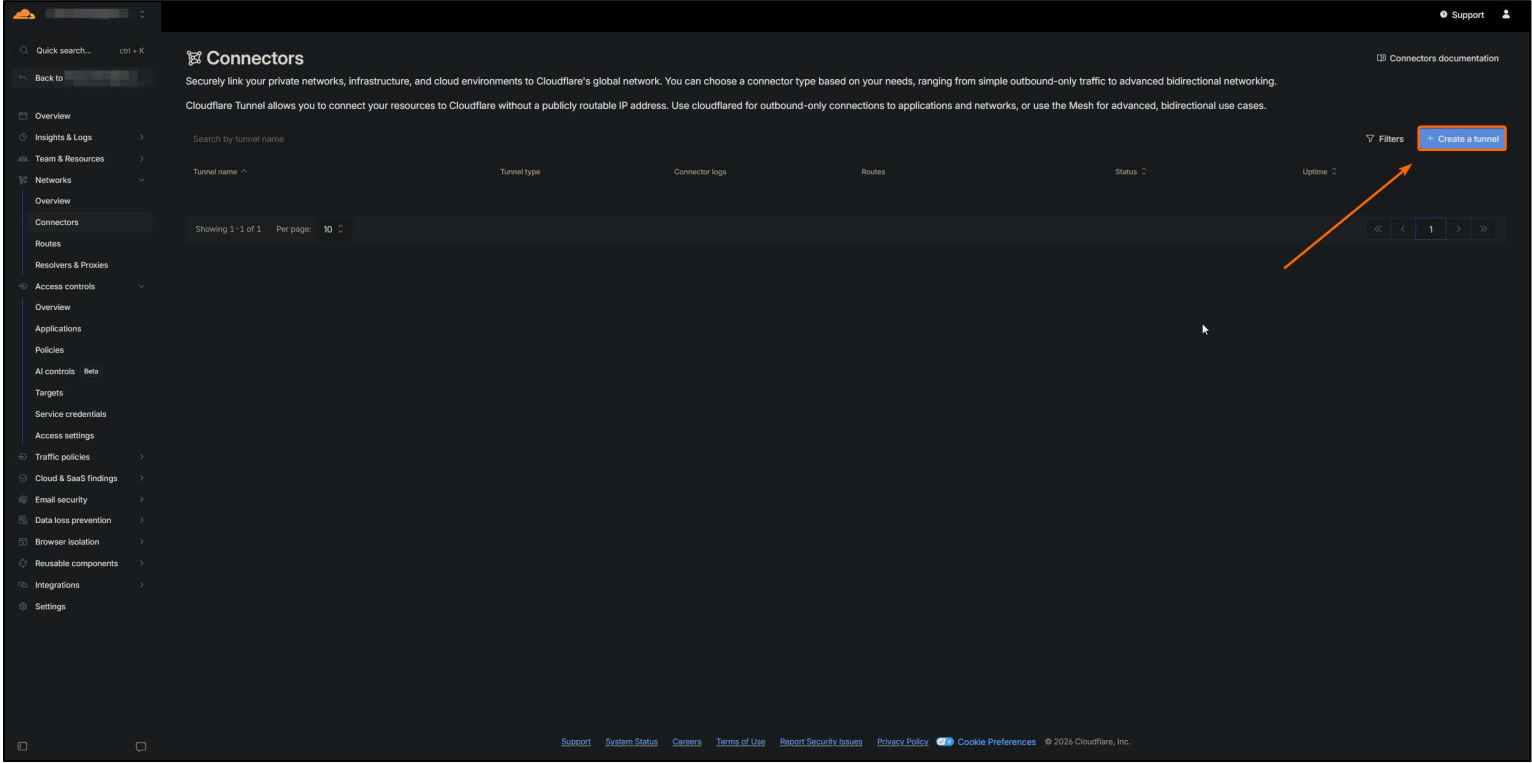
By Category Alphabetically Search

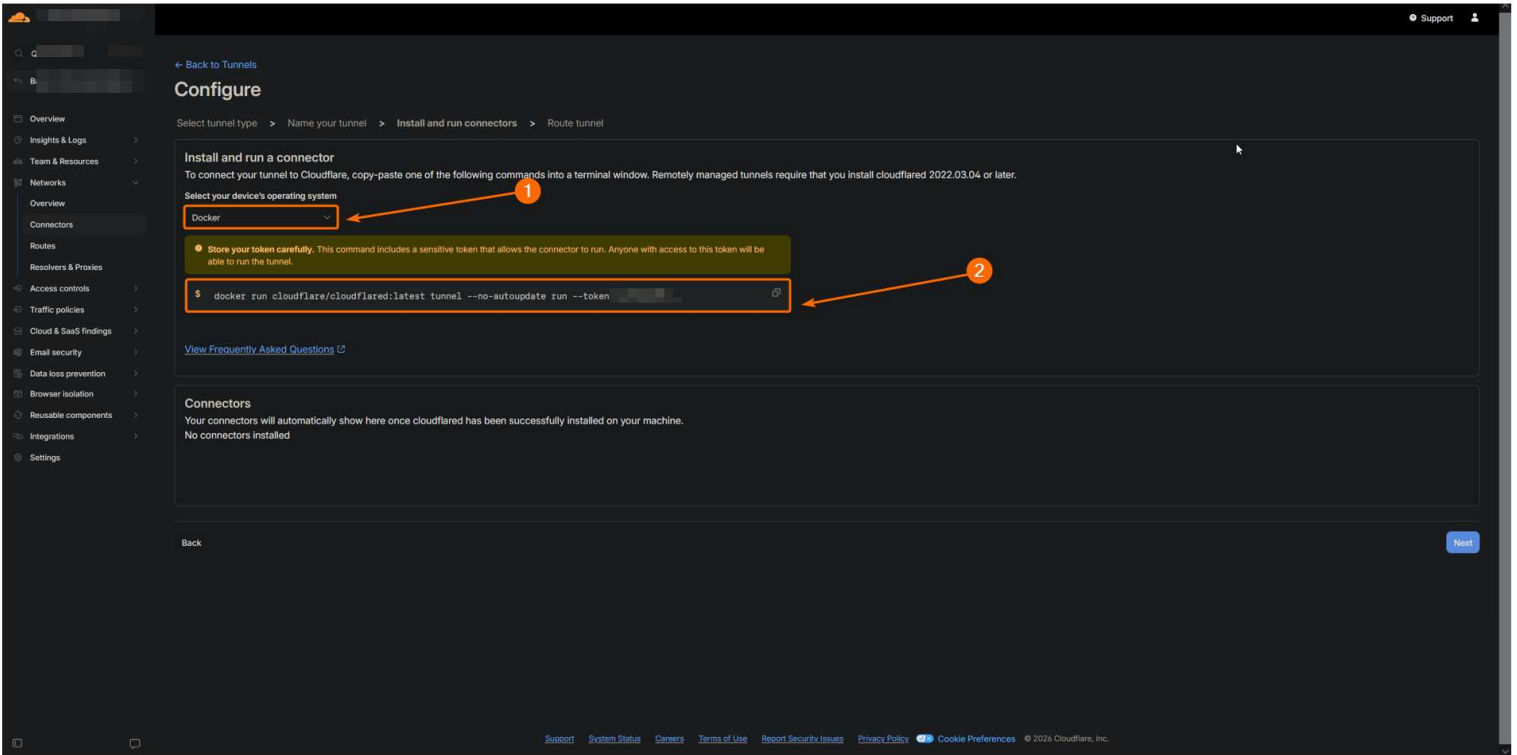
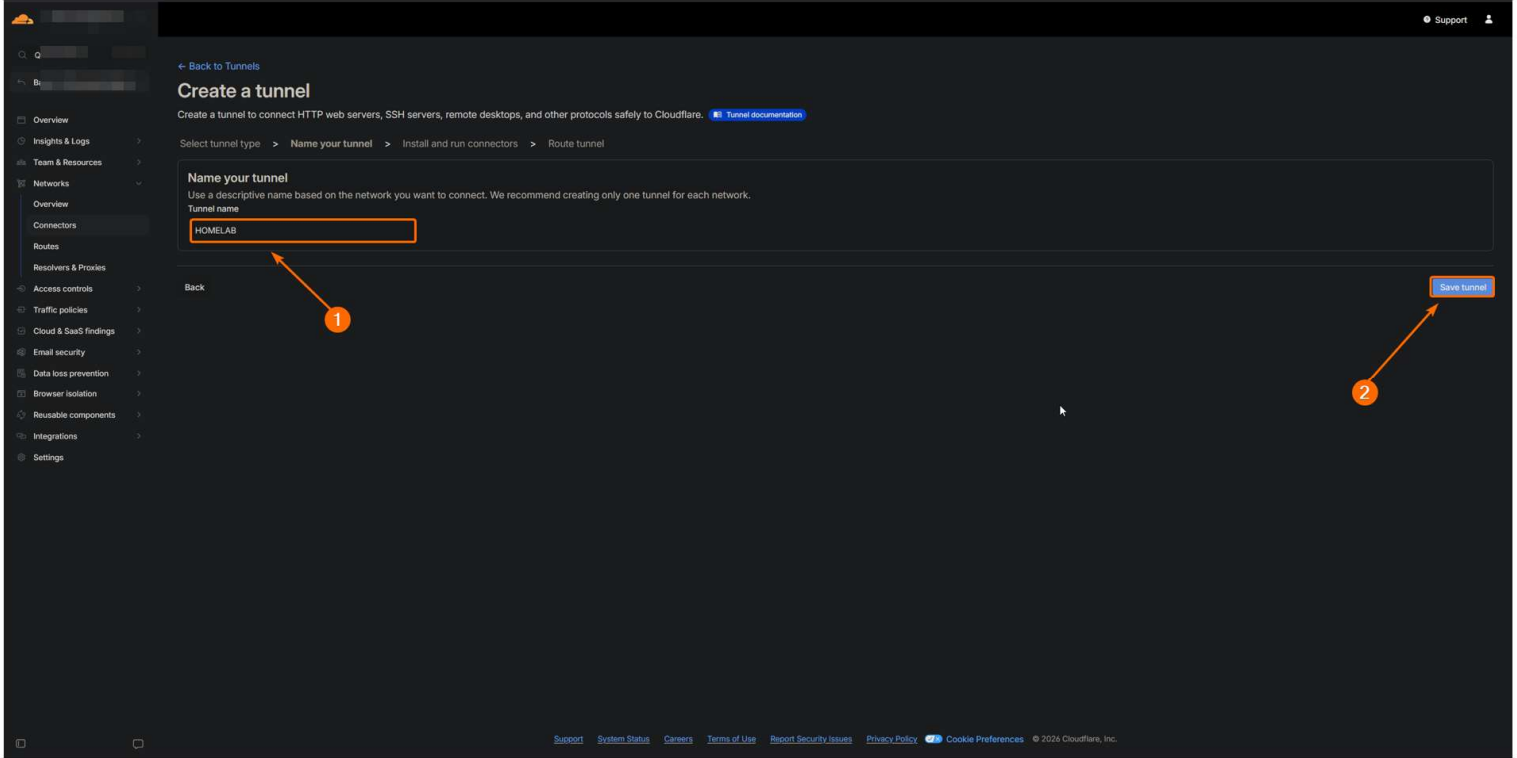
Ward Video Help Logout

Service Name	Category	Details	IP/T/QDN	Show in Service List	Ref.
VoIP, Messaging & Other Applications (8)					
H323	VoIP, Messaging & Other Applications	TCP/1720	TCP/1503	UDP/1719	0.0.0.0 0
IRC	VoIP, Messaging & Other Applications	TCP/6660-6669			0.0.0.0 0
MS-SQL	VoIP, Messaging & Other Applications	TCP/1433	TCP/1434		0.0.0.0 0
MYSQL	VoIP, Messaging & Other Applications	TCP/3306			0.0.0.0 0
RTSP	VoIP, Messaging & Other Applications	TCP/554	TCP/7070	TCP/8554	UDP/554 0.0.0.0 0
SCCP	VoIP, Messaging & Other Applications	TCP/2000			0.0.0.0 0
SIP	VoIP, Messaging & Other Applications	TCP/5060	UDP/5060		0.0.0.0 0
SIP-MSNMessenger	VoIP, Messaging & Other Applications	TCP/1863			0.0.0.0 0
Uncategorized (28)					
ADL	Uncategorized	TCP/5190-5194			0.0.0.0 0
CYSPSERVER	Uncategorized	TCP/2401	UDP/2401		0.0.0.0 0
FINGER	Uncategorized	TCP/79			0.0.0.0 0
GOPHER	Uncategorized	TCP/70			0.0.0.0 0
HOMEASSISTANT	Uncategorized	TCP/8123			0.0.0.0 2
IMMICH	Uncategorized	TCP/2283			0.0.0.0 0
INFO_ADDRESS	Uncategorized	ICMP/17:ANY			0.0.0.0 0
INFO_REQUEST	Uncategorized	ICMP/15:ANY			0.0.0.0 0
Internet-Locator-Service	Uncategorized	TCP/389			0.0.0.0 0
JELLYFIN	Uncategorized	TCP/8096			0.0.0.0 2
MGCP	Uncategorized	UDP/2427	UDP/2727		0.0.0.0 0
HMS	Uncategorized	TCP/1755	UDP/1024-5000		0.0.0.0 0
NNTTP	Uncategorized	TCP/119			0.0.0.0 0
NONE	Uncategorized	TCP/0			0.0.0.0 0
NetMeeting	Uncategorized	TCP/1720			0.0.0.0 0
QUAKE	Uncategorized	UDP/26000			0.0.0.0 0
RADIUS-OLD	Uncategorized	UDP/1845	UDP/27000	UDP/27910	UDP/27960 0.0.0.0 0
RAUDIO	Uncategorized	UDP/7070	UDP/1646		0.0.0.0 0
REFVC	Uncategorized	TCP/512			0.0.0.0 0
RLOGIN	Uncategorized	TCP/513			0.0.0.0 0
RSH	Uncategorized	TCP/514			0.0.0.0 0
TALK	Uncategorized	UDP/517-518			0.0.0.0 0
TIMESTAMP	Uncategorized	ICMP/13:ANY			0.0.0.0 0
UDP/16320	Uncategorized	UDP/16320			16320 0.0.0.0 0
UUCP	Uncategorized	TCP/540			0.0.0.0 0
VDOLIVE	Uncategorized	TCP/7000-7010			0.0.0.0 0
WAIS	Uncategorized	TCP/210			0.0.0.0 0
WINFRAME	Uncategorized	TCP/1494	TCP/2598		0.0.0.0 0
Firewall Group (4)					
Email Access	Firewall Group	DNS POP3S	INAP SMTP	INAPS SMTPS	POP3 0
Exchange Server	Firewall Group	DCE-RPC	DNS	HTTPS	0
Web Access	Firewall Group	DNS	HTTP	HTTPS	0
Windows AD	Firewall Group	DCE-RPC LDAP_UDP	DNS SAMBA	KERBEROS SMB	LDAP 0

2 Ajouts des services aux deux règles de filtrages

Etape 8 : Mise en place du Cloudflare Tunnel

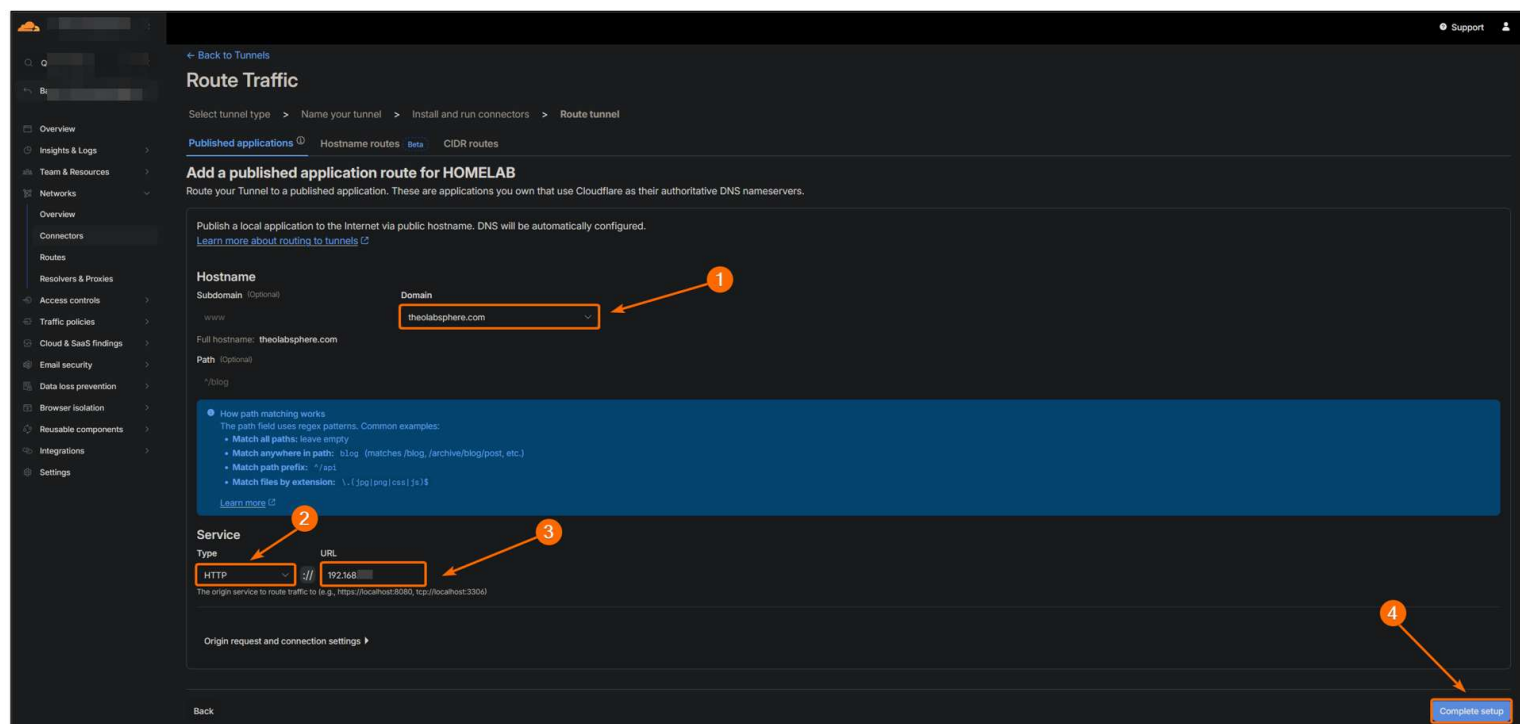




Déploiement du conteneur cloudflare sur le serveur Docker en DMZ, avec redémarrage automatique :

```
docker run -d --name cloudflared --restart always \
  cloudflare/cloudflared:latest tunnel --no-autoupdate run --token MON_TOKEN
```

J'ai renseigné l'adresse IP privée de mon serveur Nginx Proxy Manager.



Etape 8 : Test de validation

① Test de la résolution du nom de domaine

Une fois tunnel mis en place la résolution du nom de domaine.

Le domaine ne répond pas avec mon @IP Publique mais bien avec une @IP appartenant à Cloudflare.

```
C:\Users\Theo>ping theolabsphere.com

Envoi d'une requête 'ping' sur theolabsphere.com [188.114.96.2] avec 32 octets de données :
Réponse de 188.114.96.2 : octets=32 temps=9 ms TTL=57
Réponse de 188.114.96.2 : octets=32 temps=7 ms TTL=57
Réponse de 188.114.96.2 : octets=32 temps=8 ms TTL=57
Réponse de 188.114.96.2 : octets=32 temps=7 ms TTL=57

Statistiques Ping pour 188.114.96.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 7ms, Maximum = 9ms, Moyenne = 7ms

C:\Users\Theo>
```

② Test de la résolution du des sous-domaine Jellyfin et Immich

Le Cloudflare Tunnel se charge de masquer mon IP publique pour personne ne puisse la voir dans le DNS.

L'adresse (104.21.42.199 et 172.67.165.145) ce sont des IP appartenant à Cloudflare. Peu importe le serveur DNS interrogé (Google, Quad9, OpenDNS, NextDNS, jusqu'en Russie ou en Afrique du Sud), tout le monde reçoit la même chose : l'IP de Cloudflare, jamais la tienne.

[Home](#)
[DNS Lookup](#)
[All Tools](#)
[Public DNS List](#)

88.178.1.190

CHECK DNS

A

Search

CD Flag

Refresh: 20 sec.

Mountain View CA, United States	104.21.42.199 172.67.165.145	✓
Google		
Berkeley, US	172.67.165.145 104.21.42.199	✓
Quad9		
San Francisco CA, United States	104.21.42.199 172.67.165.145	✓
OpenDNS		
Wilmington, United States of America	104.21.42.199 172.67.165.145	✓
NextDNS Inc.		
San Francisco, US	104.21.42.199 172.67.165.145	✓
Quad9		
Ashburn, United States	172.67.165.145 104.21.42.199	✓
NetStar		
United States	104.21.42.199 172.67.165.145	✓
CenturyLink		
Burnaby, Canada	104.21.42.199 172.67.165.145	✓
Fortinet Inc.		
Yekaterinburg, Russian Federation	172.67.165.145 104.21.42.199	✓
Skydns		
Cullinan, South Africa		
Liquid Telecommunications Ltd		
Amsterdam, Netherlands	188.114.97.3 188.114.96.3	✓
Freedom Registry BV		
Lille, France	188.114.96.2 188.114.97.2	✓
Comptel SAS		
Paterna de Rivera, Spain	188.114.96.5 188.114.97.5	✓
Surrenkey Networks S.L.		
Bruck, Austria	172.67.165.145 104.21.42.199	✓
verisurf.net		

CHECK DNS PROPAGATION

Whether you have recently changed your DNS records, switched web host, or started a new website, checking whether the DNS records are propagated globally is essential. DNS Checker provides a free DNS propagation check service to check Domain Name System records against a selected list of DNS servers in multiple regions worldwide. Perform a quick DNS propagation lookup for any hostname or domain, and check DNS data collected from all available DNS Servers to confirm that the DNS records are fully propagated.

DNS Propagation Map by DNSChecker.org

Server Location Resolved Not Resolved

DNS Lists

IPs

By clicking "Accept" or continuing to use our site, you agree to our Website's Privacy Policy

Accept

Decline

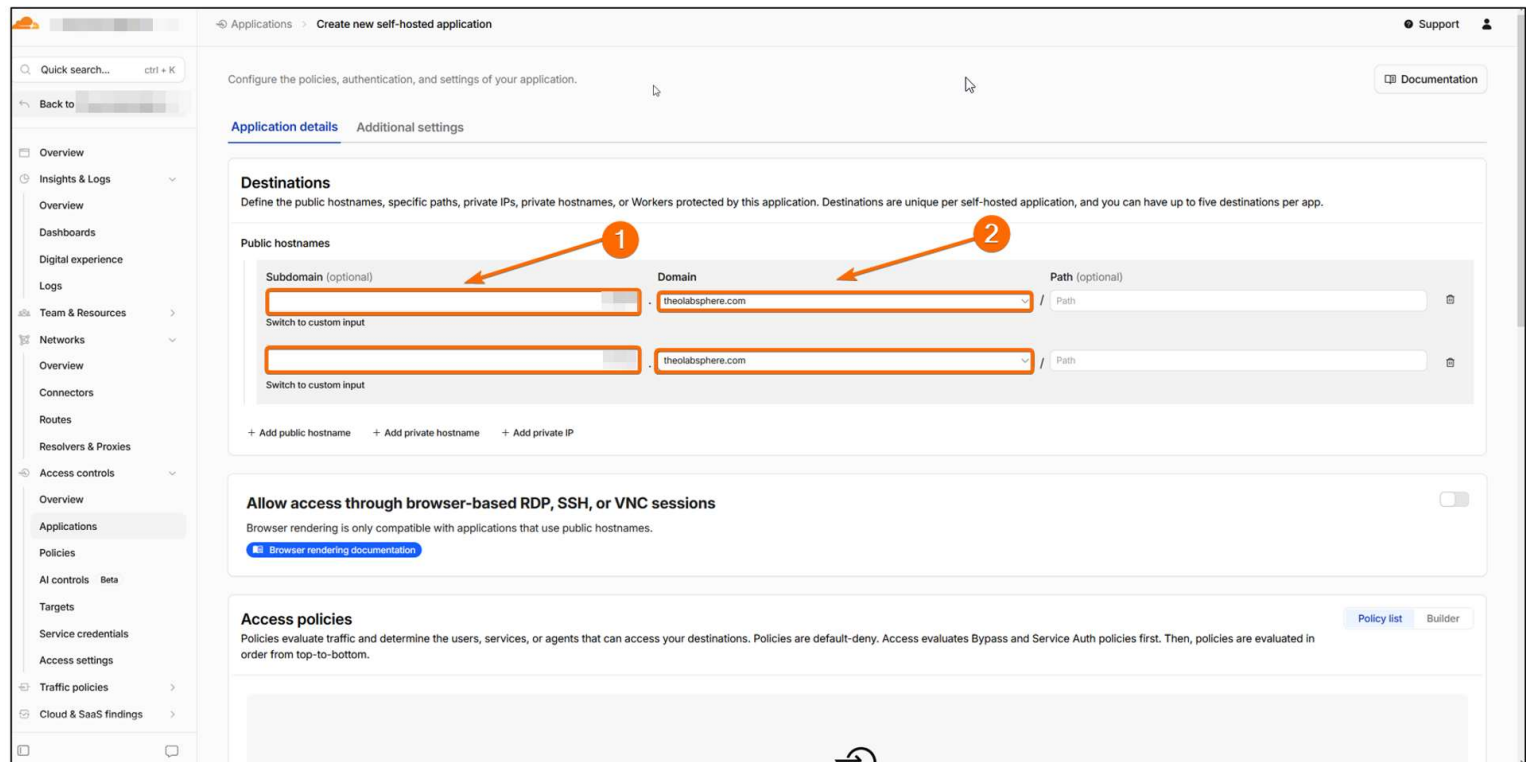
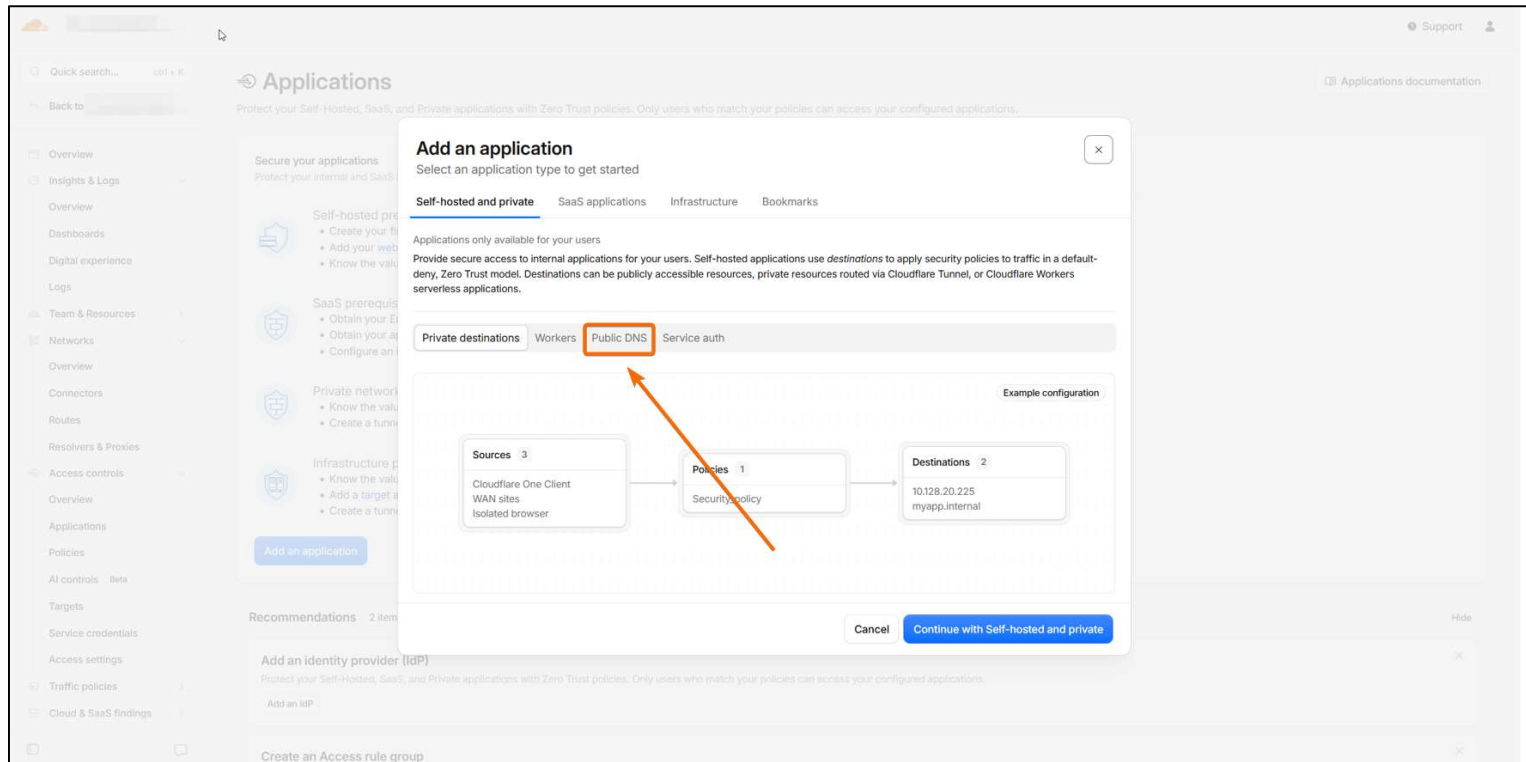
③ Test avancée avec la ligne de commande « curl -v »

Cette commande curl avec l'option -v affiche le contenu brut de la page HTML, elle affiche en plus tout le détail de l'échange réseau.

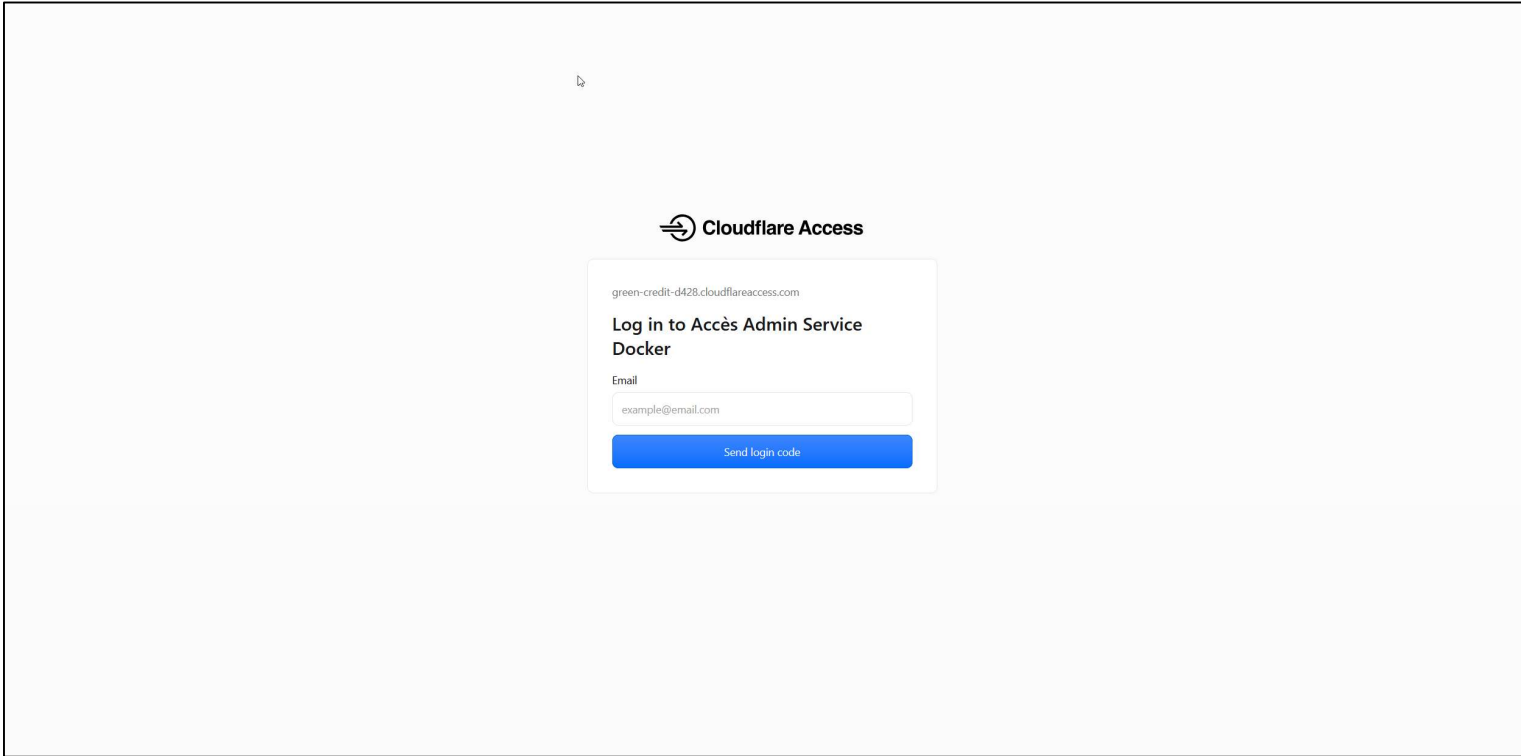
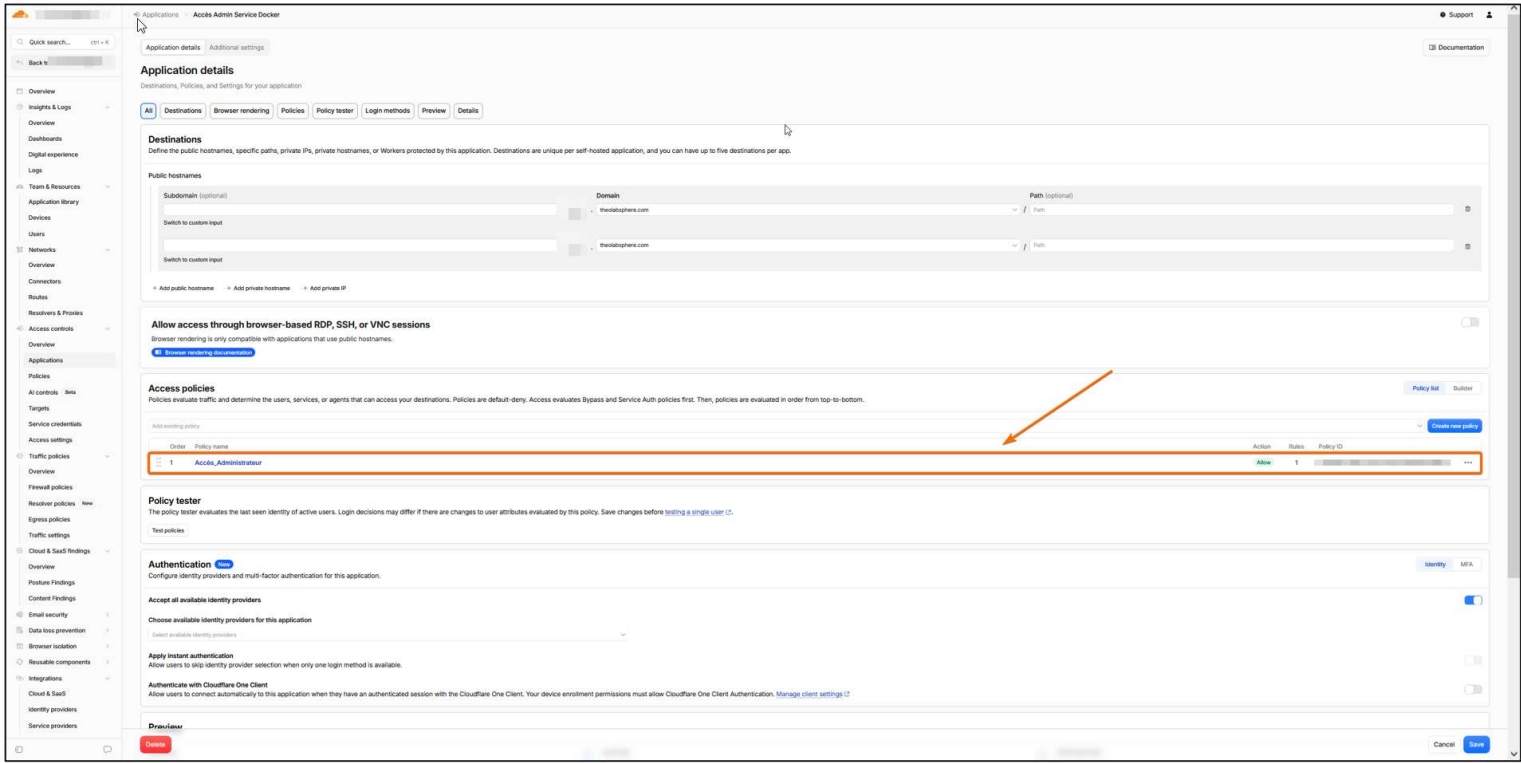
- La résolution DNS et l'IP à laquelle il se connecte
- La requête HTTP envoyée (méthode, headers)
- La réponse du serveur (code de statut, headers de réponse)
- Le contenu final de la page

Etape 9 : Cloudflare Acces (ZeroTrust) – Contrôle d'Accès

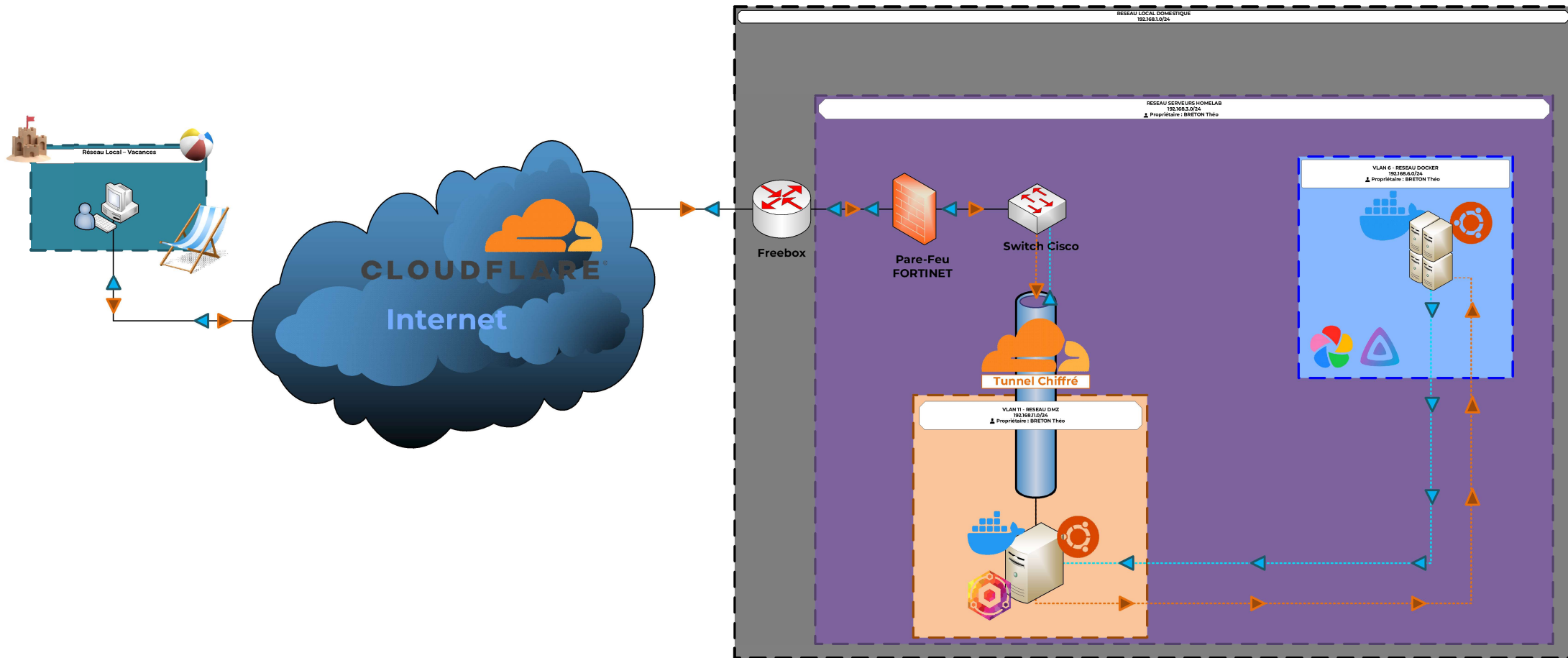
Création d'une application Self-hosted dans Cloudflare Zero Trust regroupant les deux hostnames.



Mise en place des règles de sécurité pour empêcher tout personne extérieure au domaine d'accéder aux services.



Etape 10 : Schéma Réseau de l'intégration de NGINX Proxy Manager et de Cloudflare dans le Homelab :

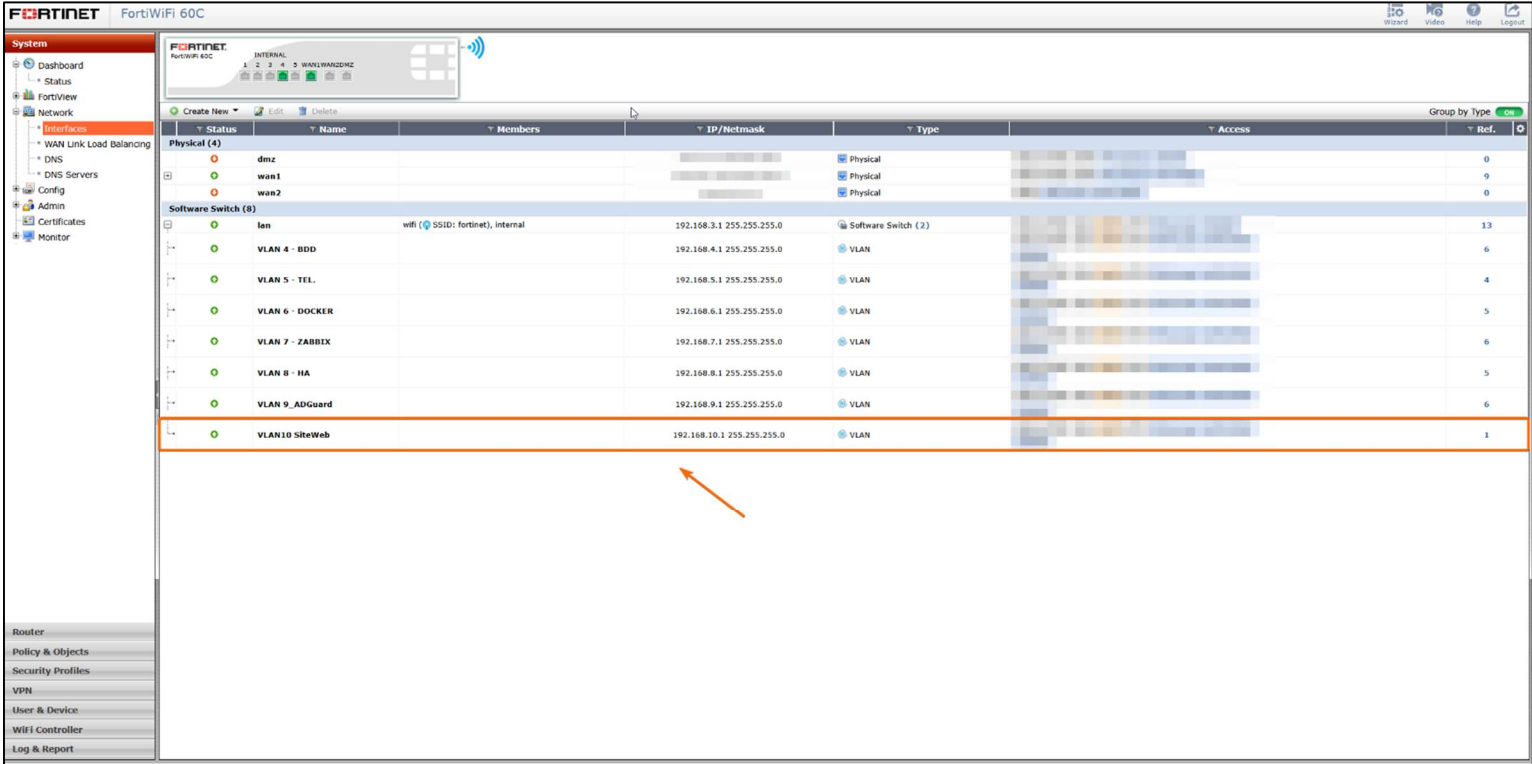


PARTIE 2 : MISE EN PLACE DU SITE WEB A HERBERGER SUR LE HOMELAB

Etape 1 : Création du VLAN pour le serveur WEB

```
SW-HOMELAB_THEO(config)#vlan 10
SW-HOMELAB_THEO(config-vlan)#name Reseau_SiteWeb
SW-HOMELAB_THEO(config-vlan)#exit
```

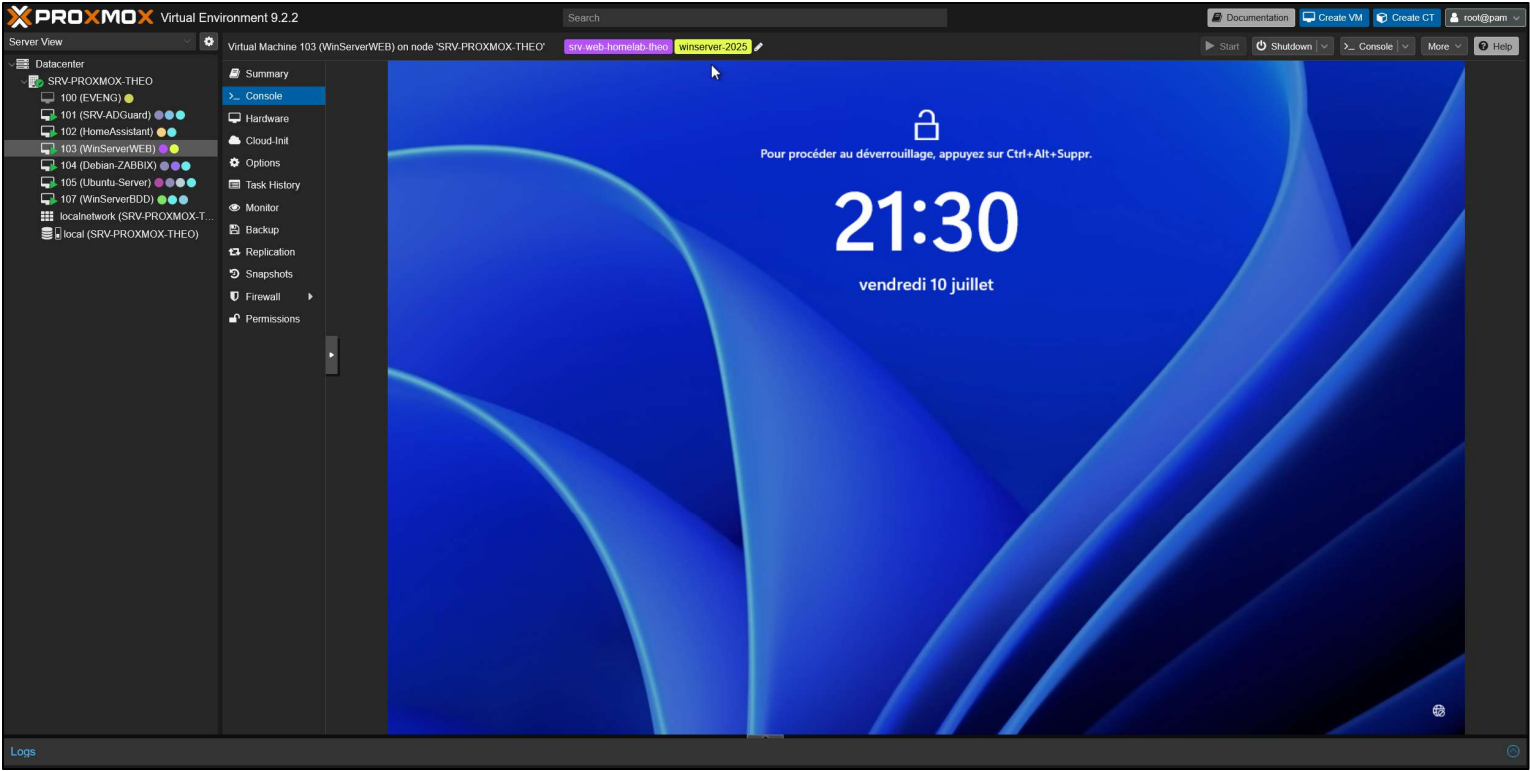
Etape 2 : Création de la sous interface sur le pare-feu pour le VLAN



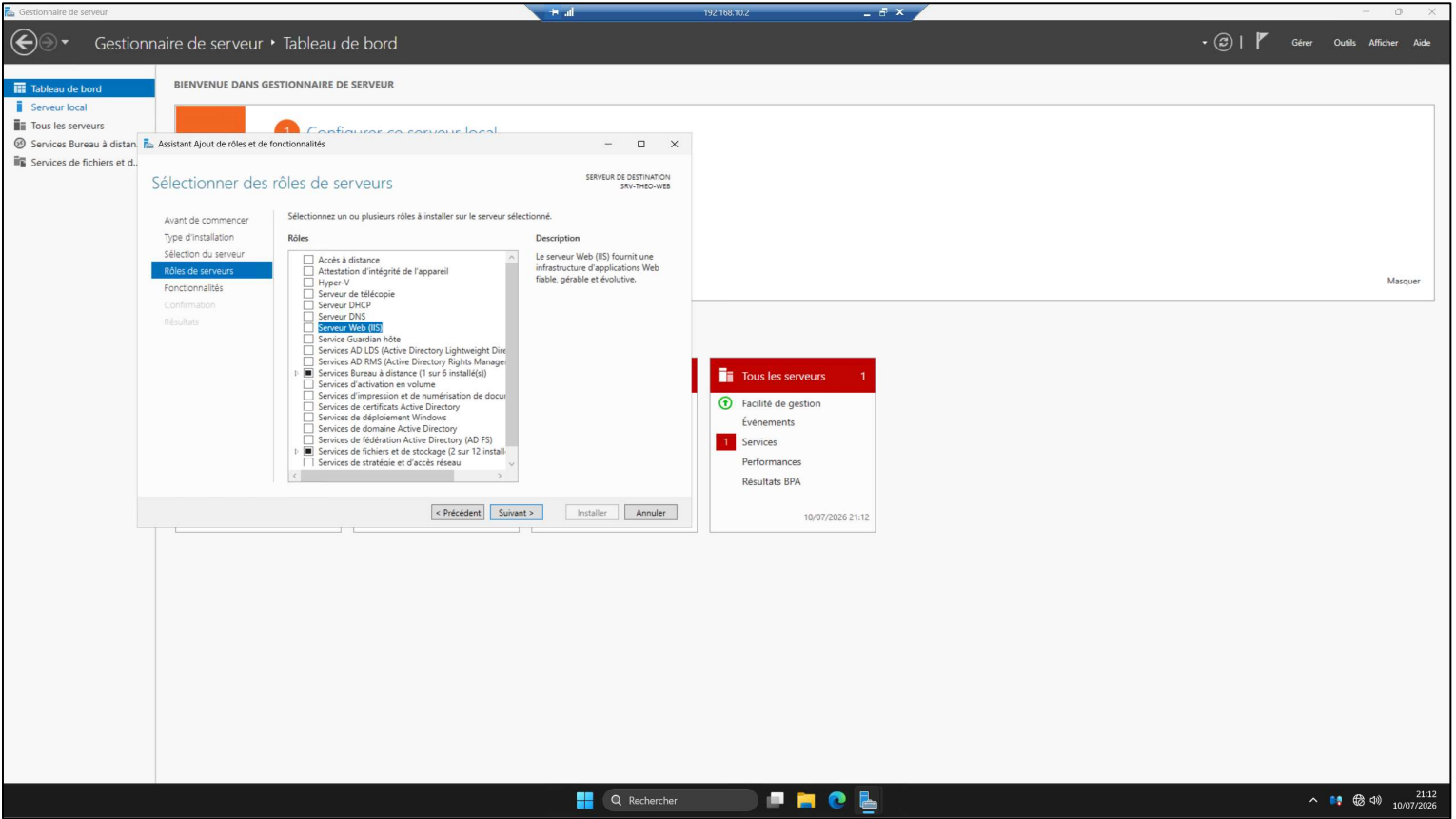
Etape 3 : Création et Modification règle de filtrage au niveau du pare-feu

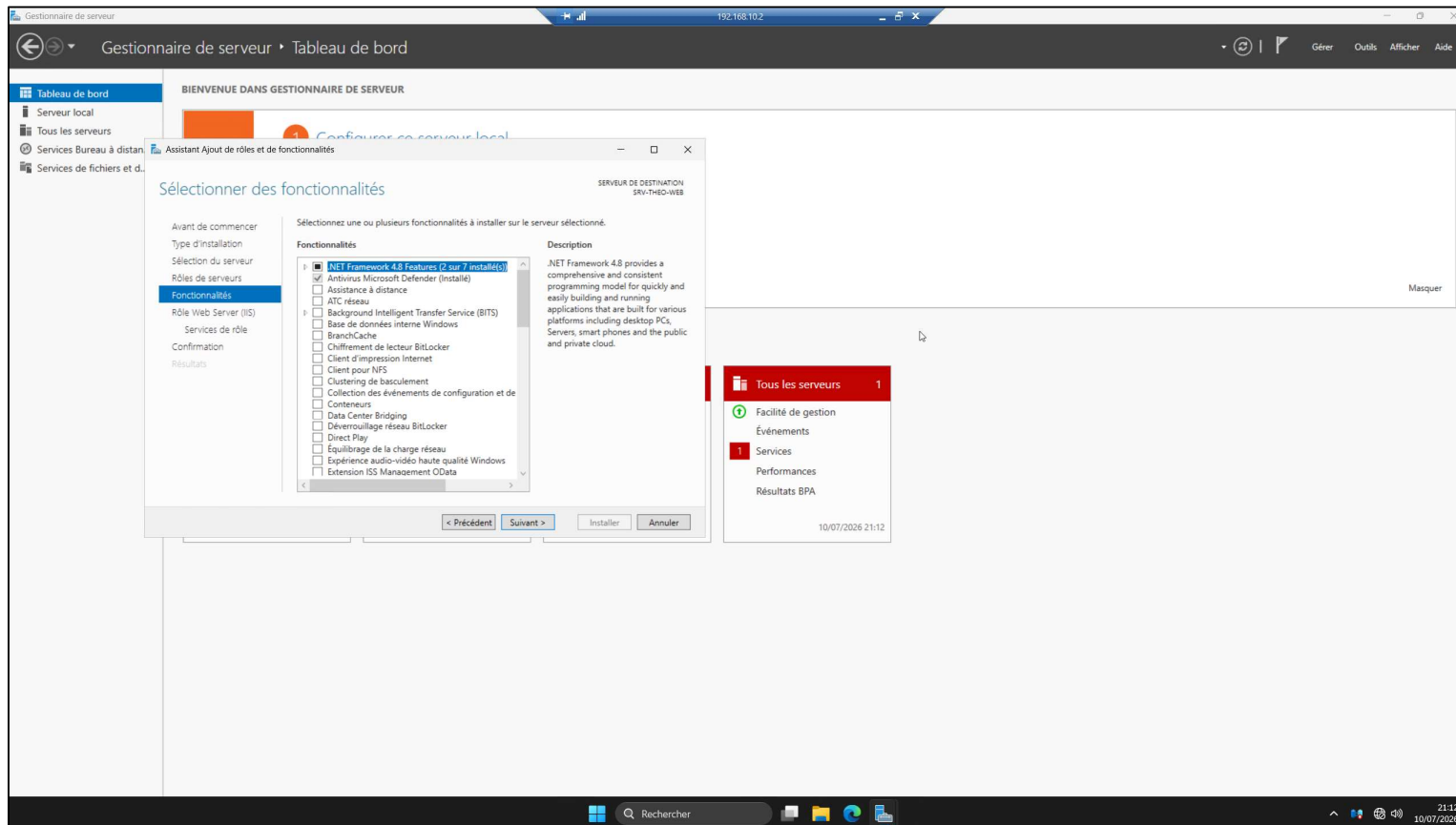
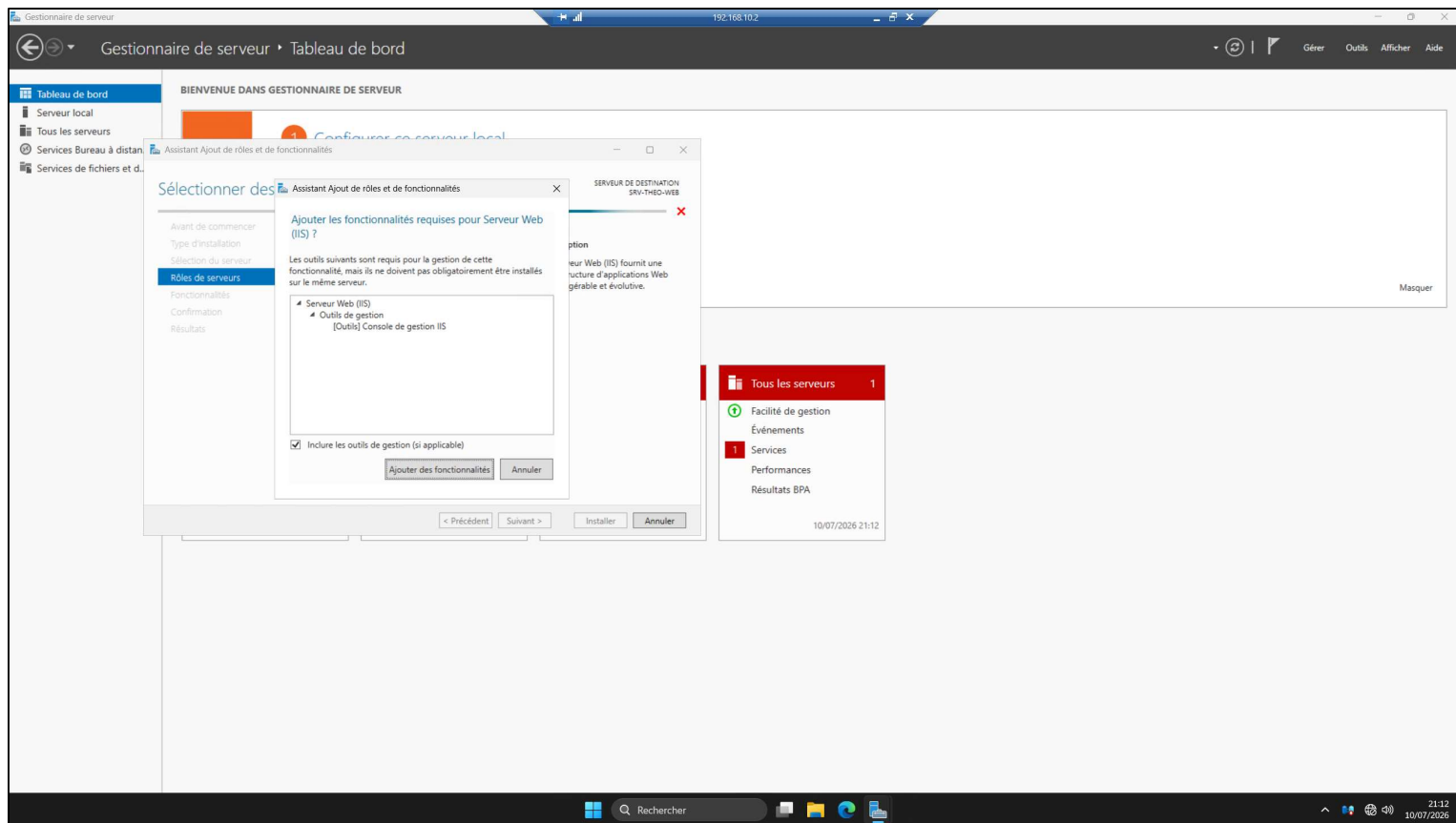
```
SW-HOMELAB_THEO(config)#vlan 10
SW-HOMELAB_THEO(config-vlan)#name Reseau_SiteWeb
SW-HOMELAB_THEO(config-vlan)#exit
```

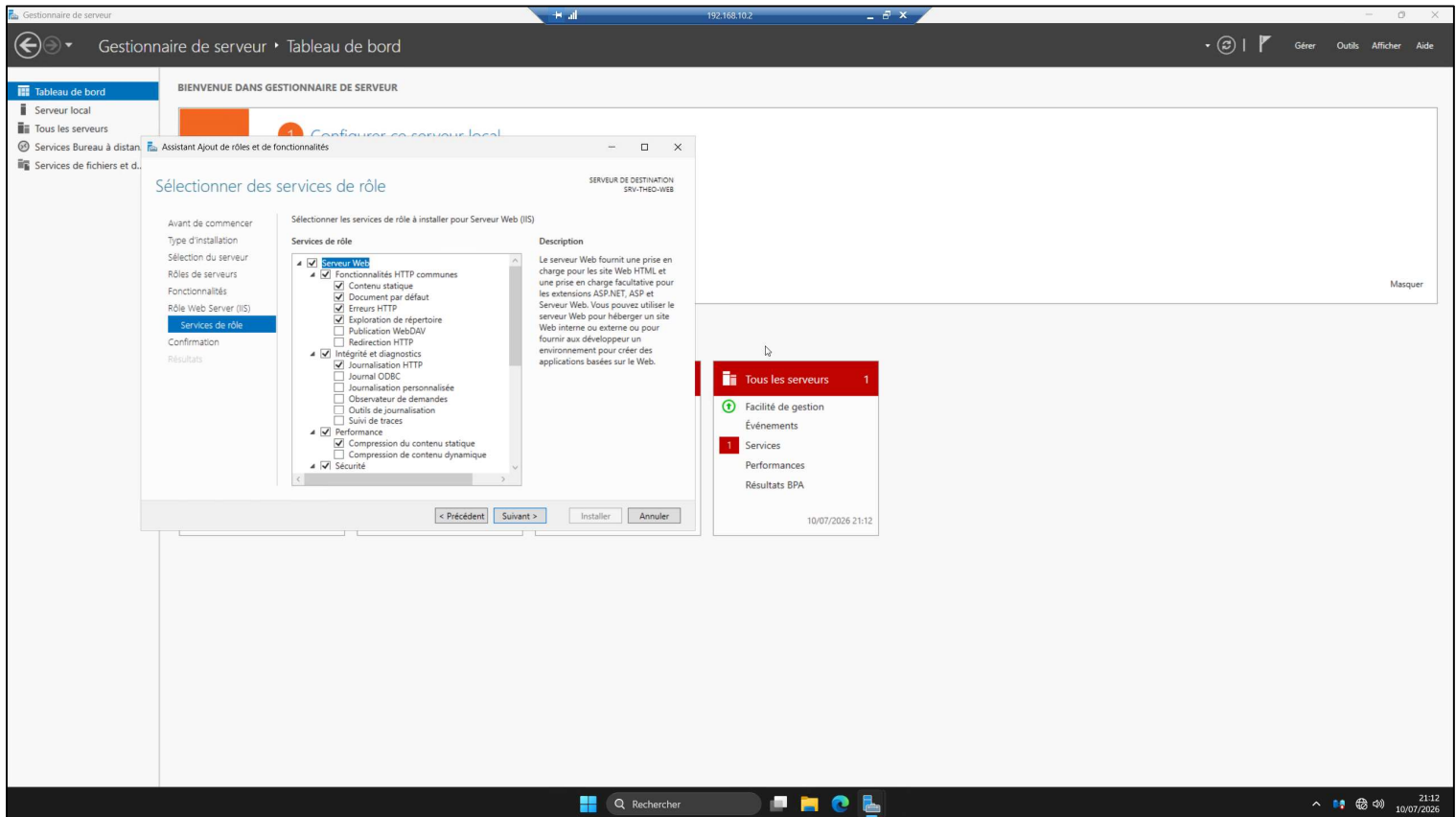
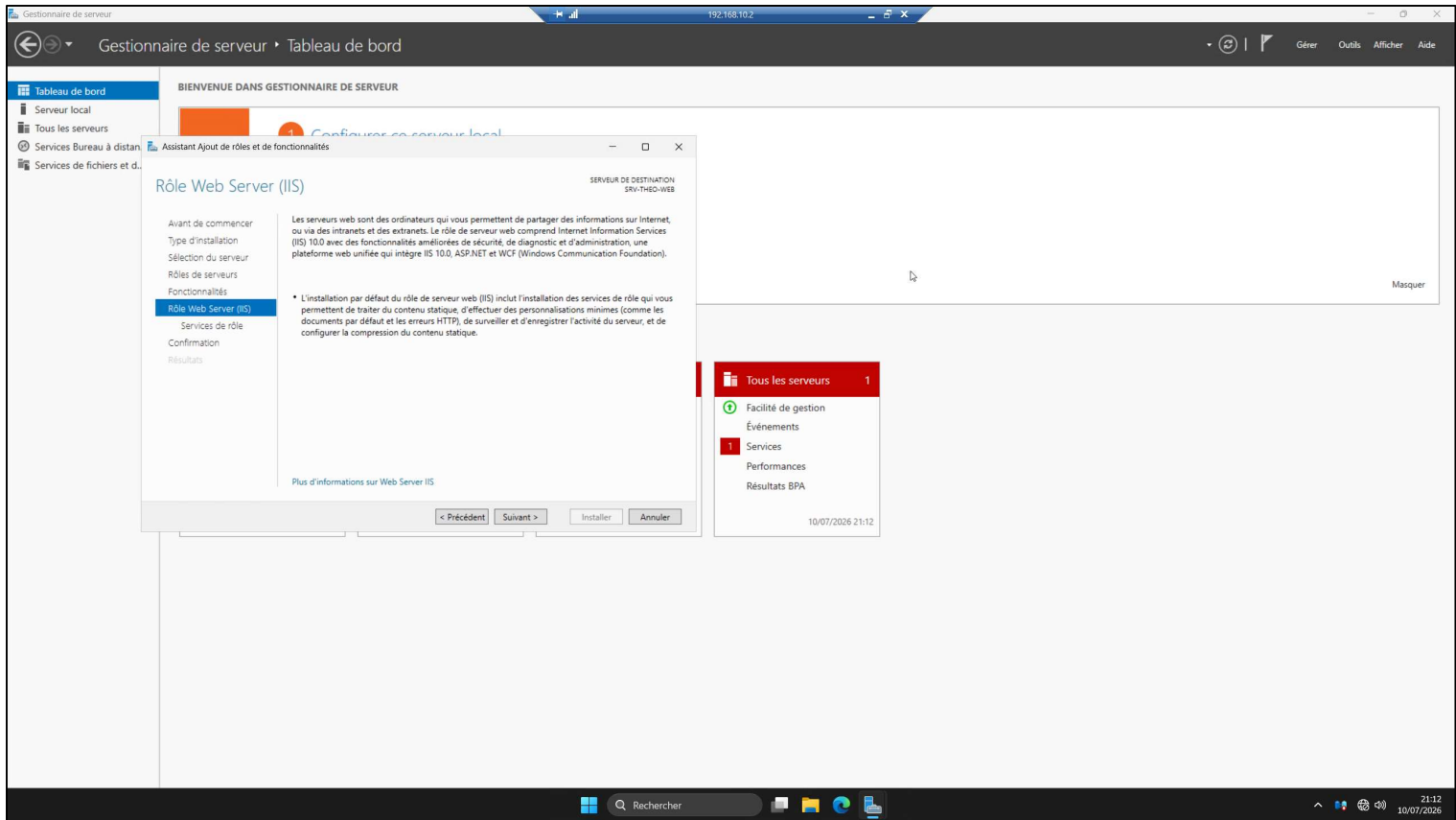
Etape 4 : Installation de la machine virtuelle sur Proxmox

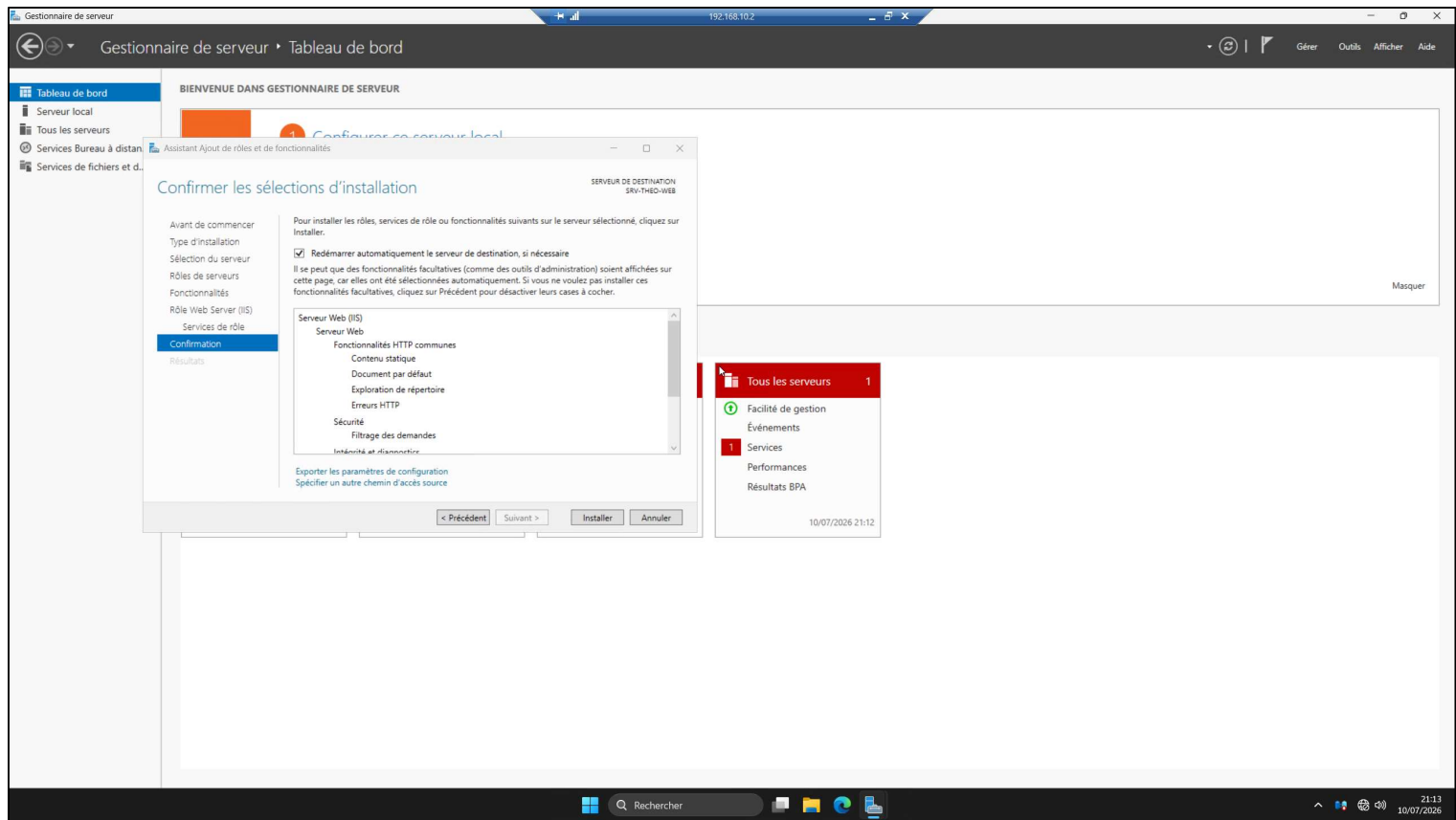


Etape 5 : Installation du rôle IIS pour serveur

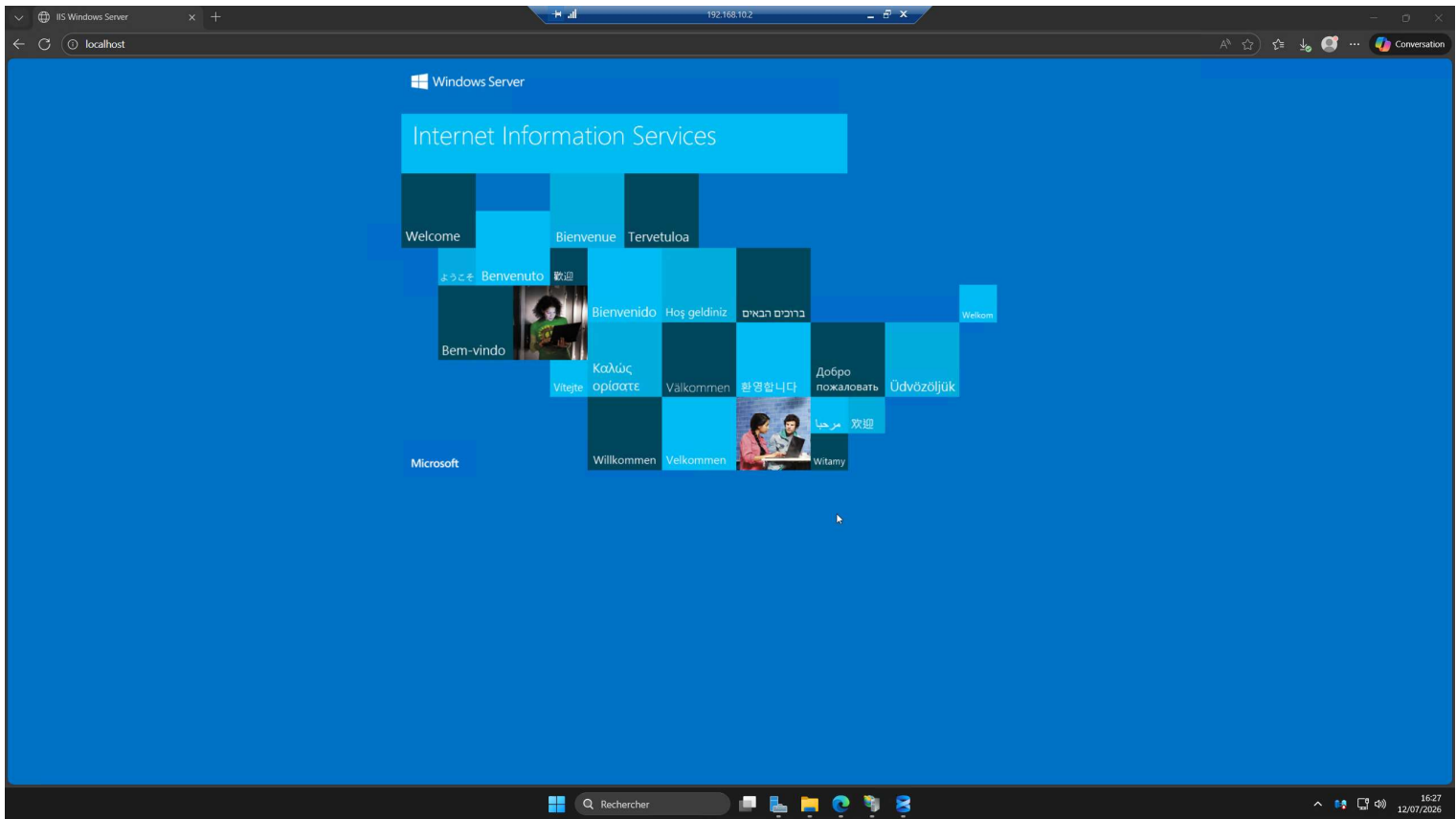








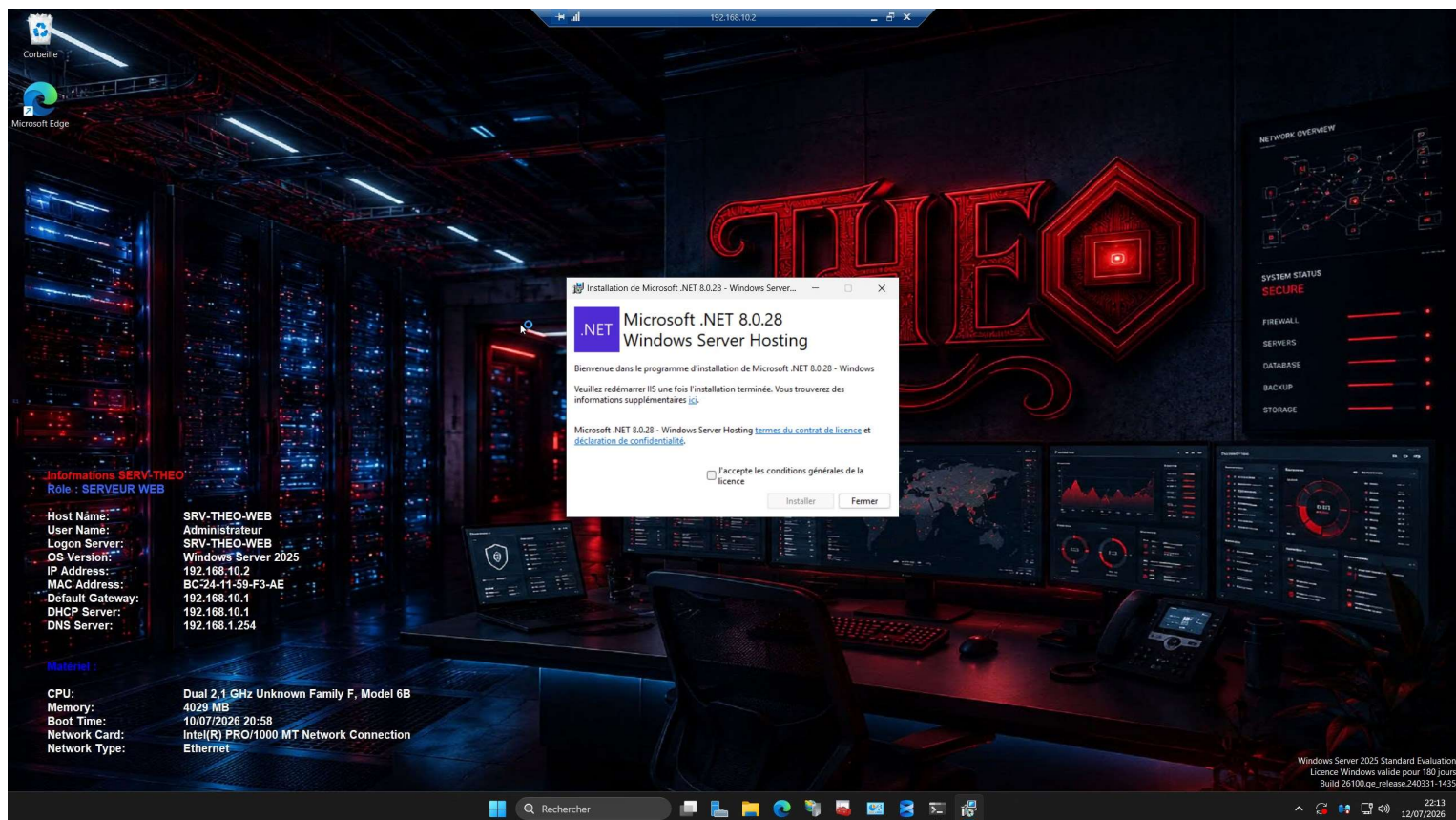
Le rôle IIS fonctionne parfaitement la page par défaut fonctionne :

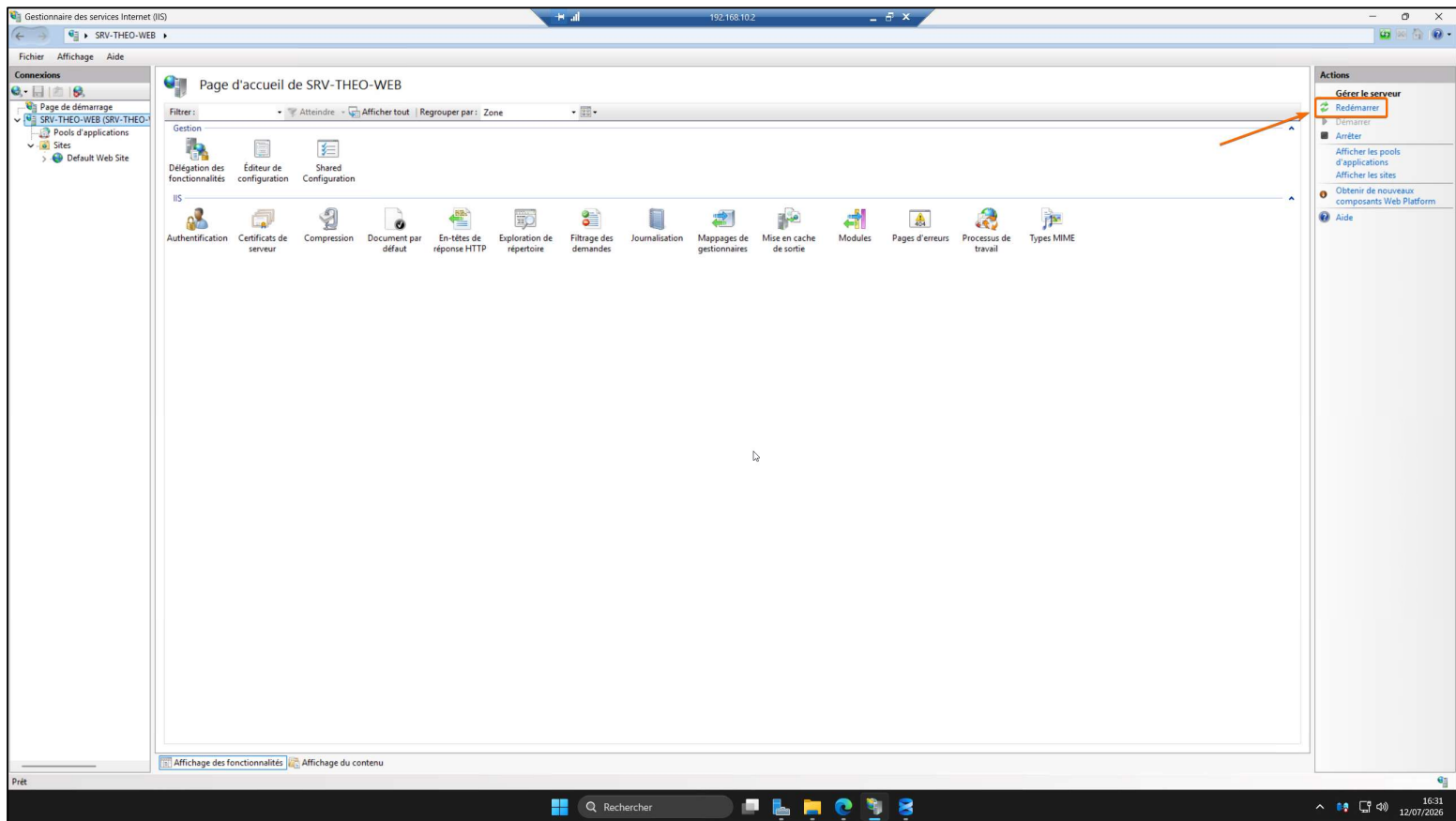
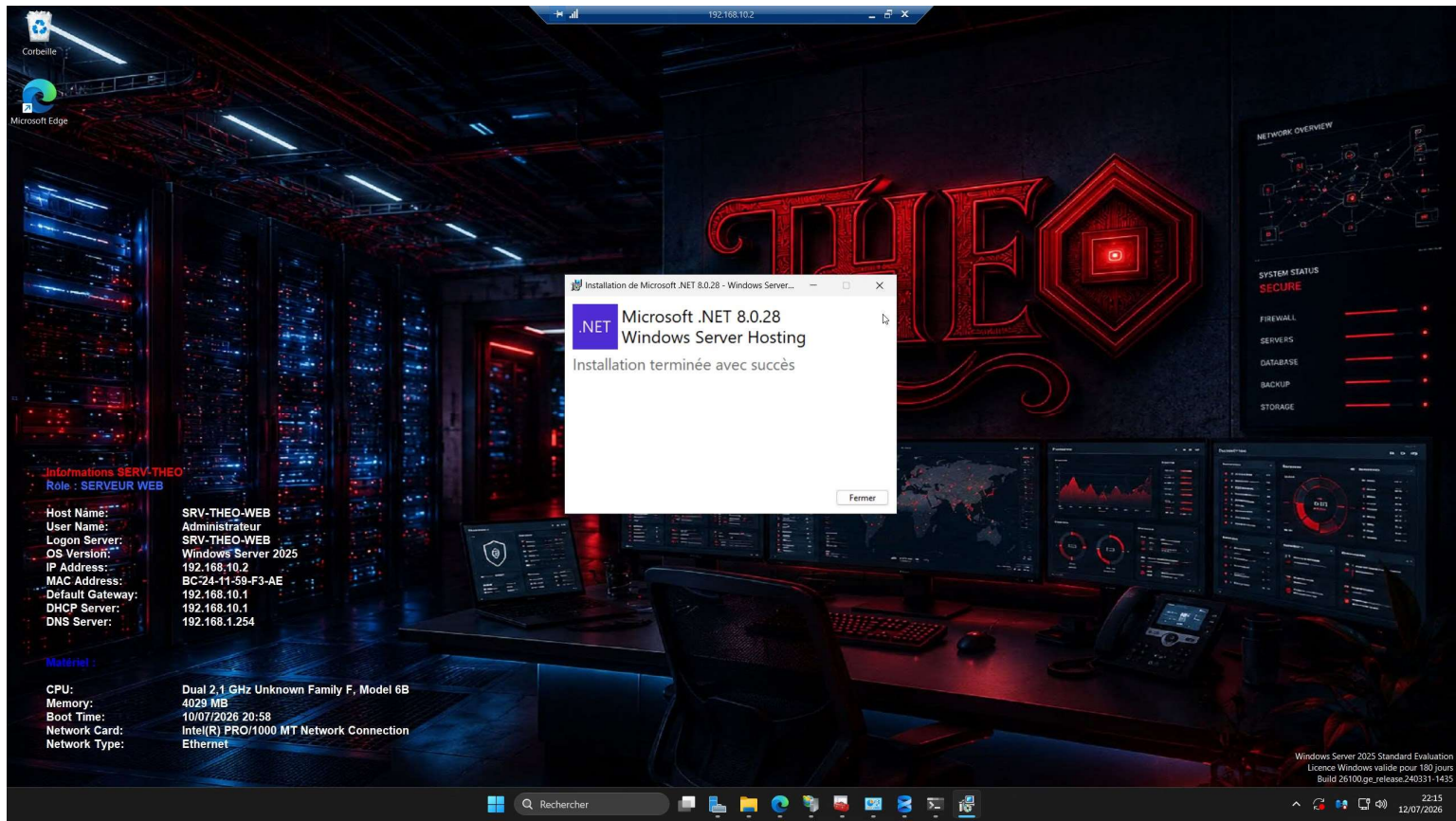


Etape 6 : Installation de .NET 8 Hosting Bundle

Le rôle IIS seul ne sait pas exécuter une application ASP.NET Core : il faut le module ASP.NET Core (ANCM), fourni par le Hosting Bundle. C'est différent du simple « SDK .NET » ou « Runtime .NET » — c'est un package spécifique pour IIS.

1. Sur le serveur, ouvrir <https://builds.dotnet.microsoft.com/dotnet/aspnetcore/Runtime/8.0.28/dotnet-hosting-8.0.28-win.exe>
2. Dans la section « ASP.NET Core Runtime 8.0.x », télécharger le Hosting Bundle pour Windows (fichier dotnet-hosting-8.x.x-win.exe).
3. Exécuter l'installateur (clic droit → Exécuter en tant qu'administrateur).
4. Redémarrer IIS pour que le module soit chargé :



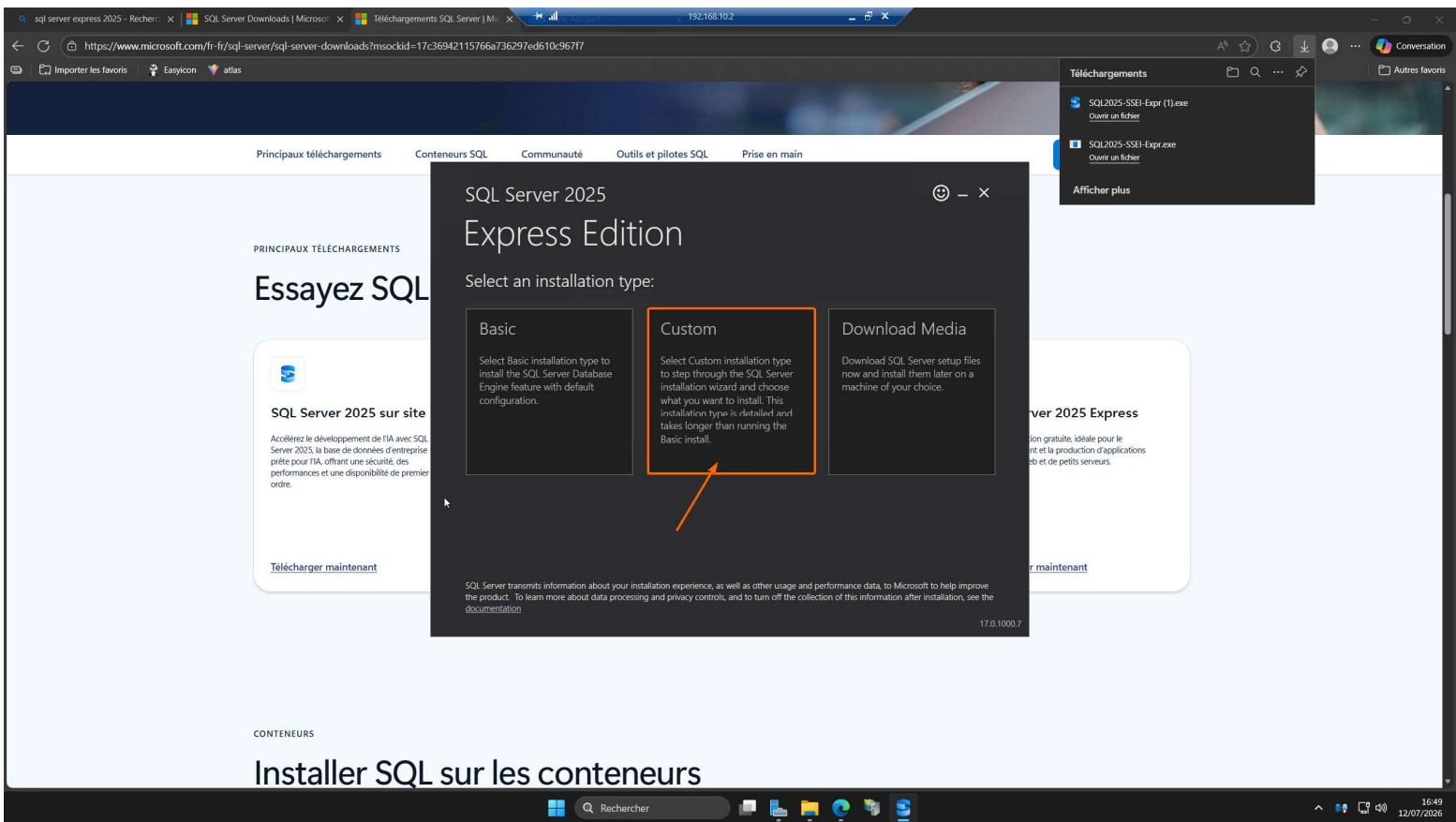
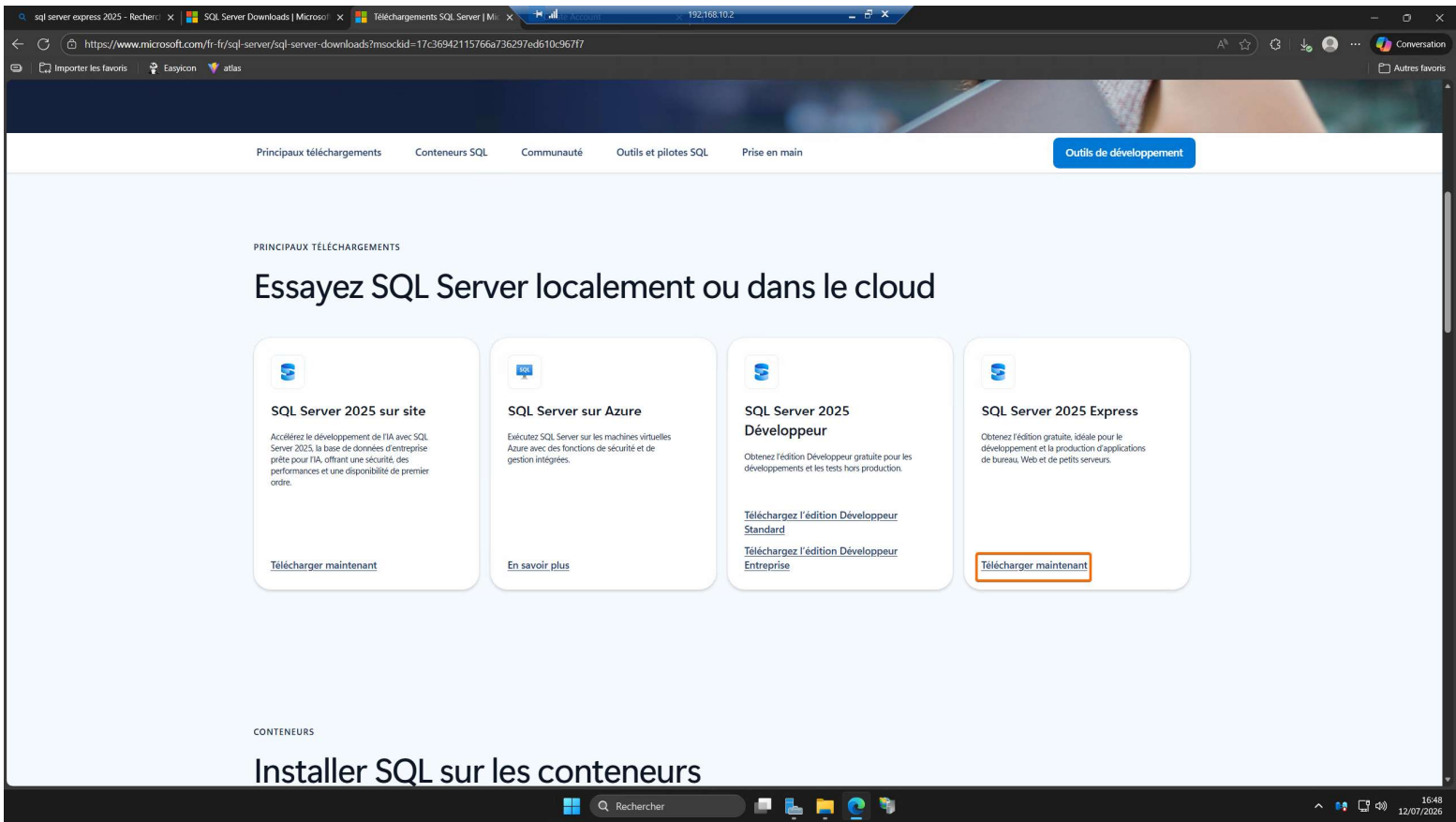


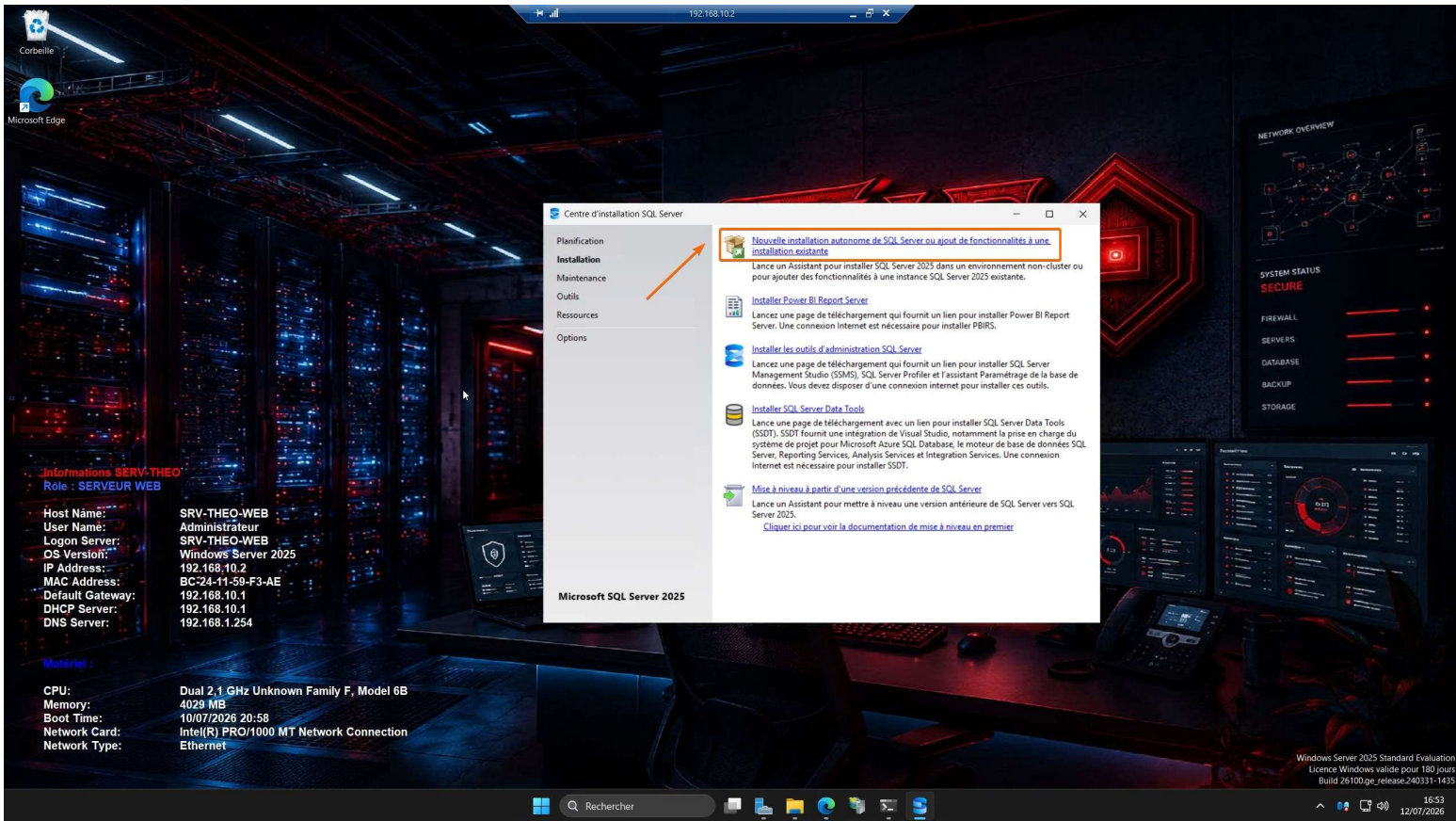
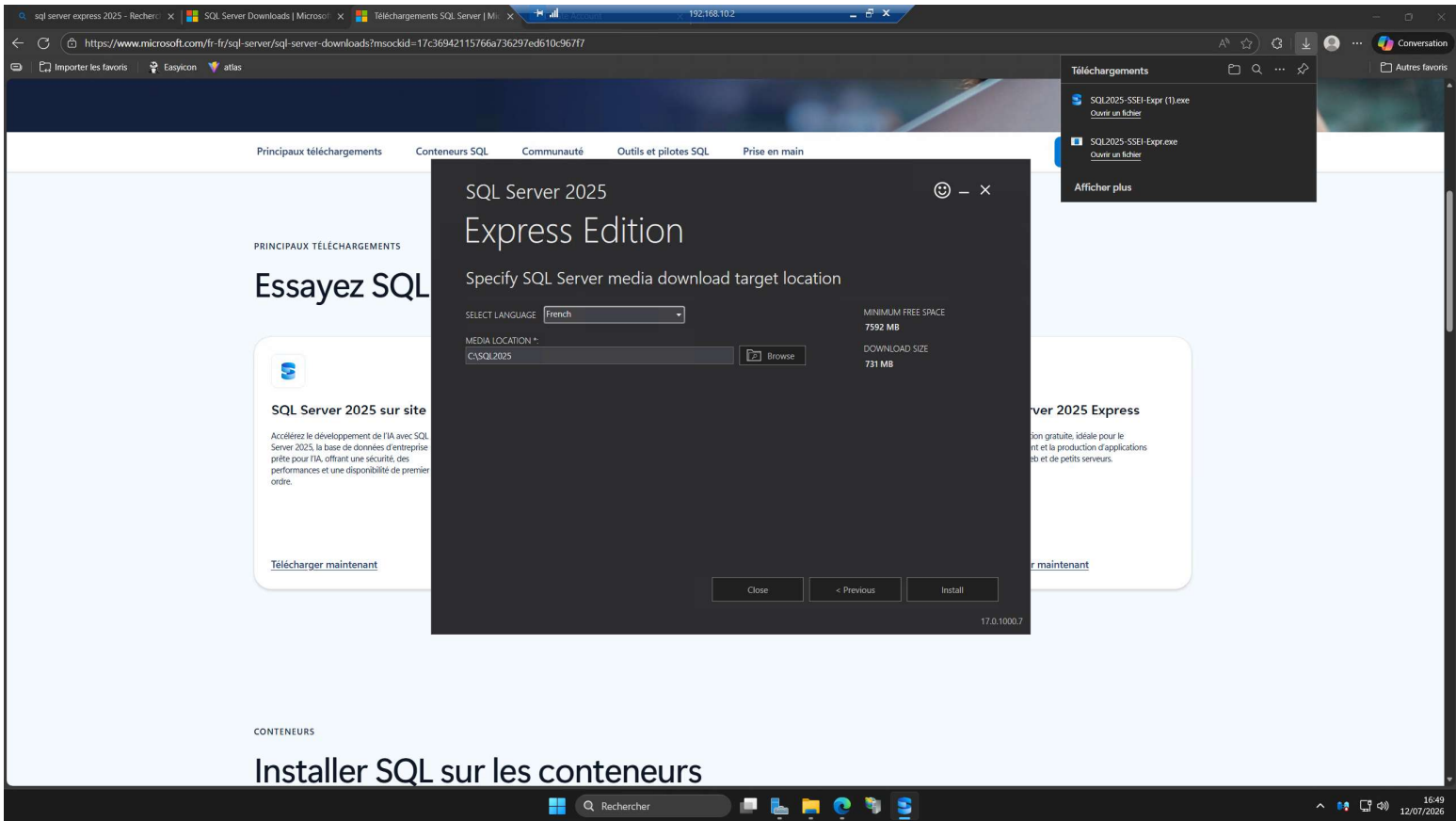
Etape 7 : Installation de SQL Server

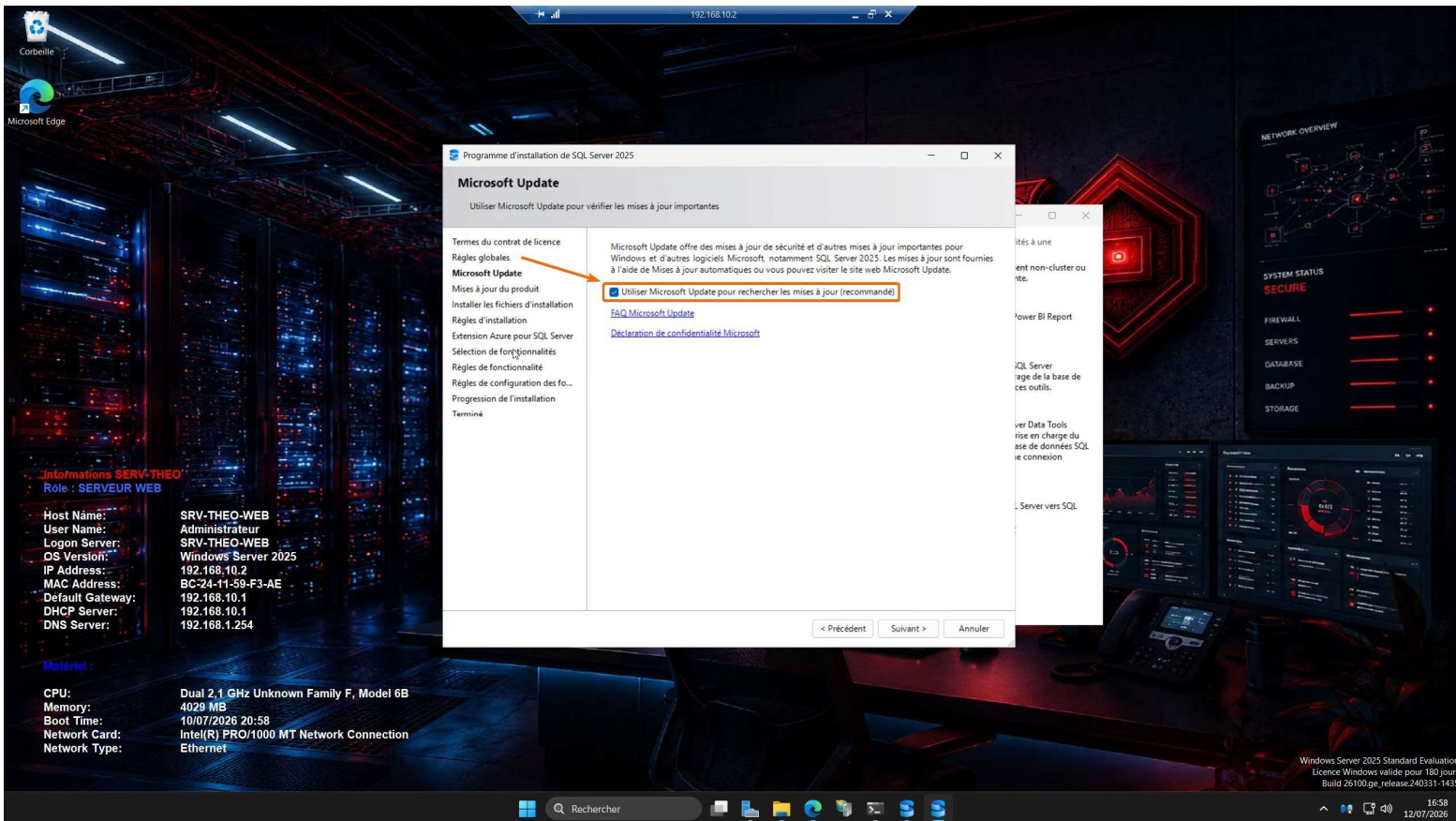
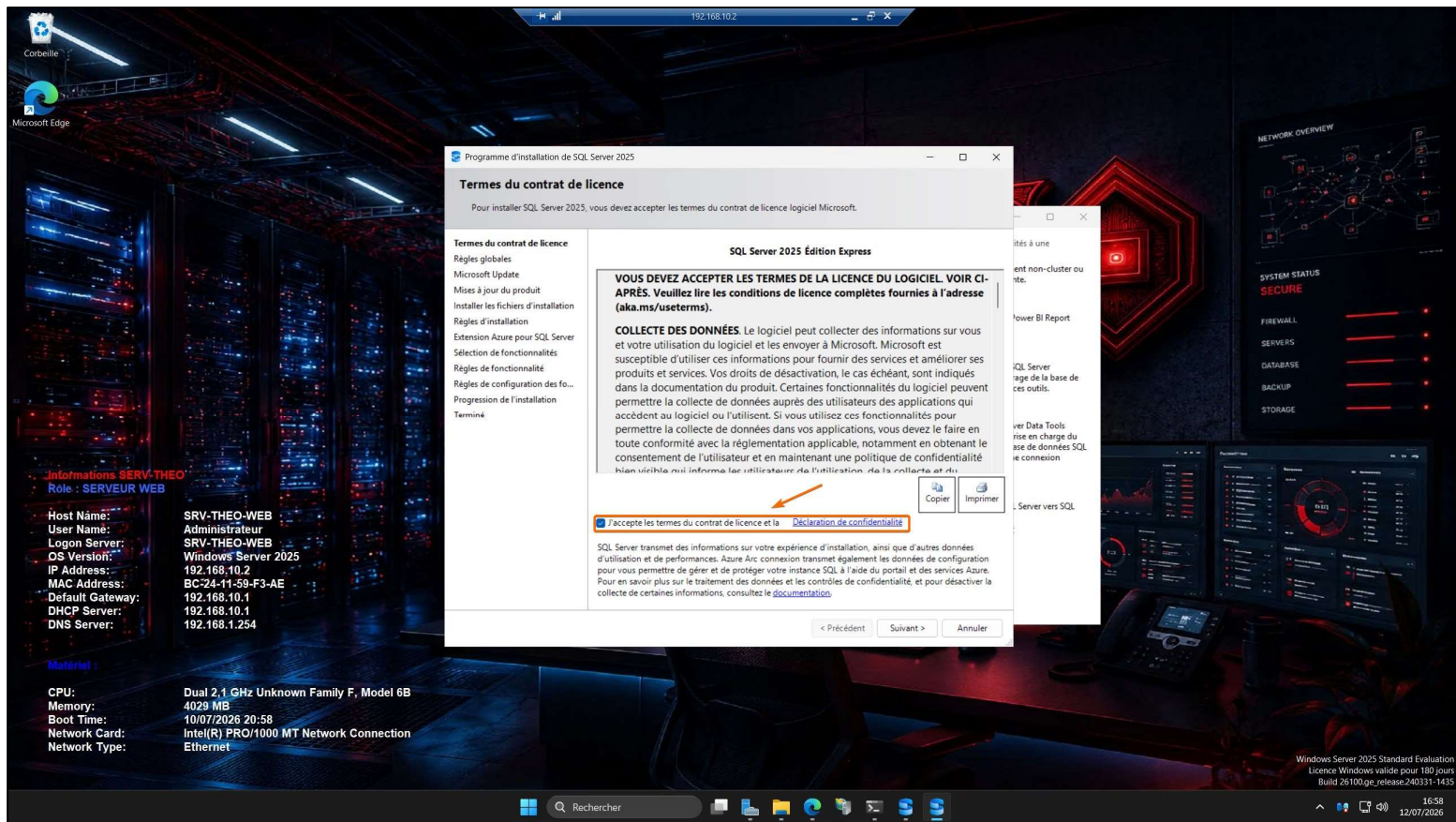


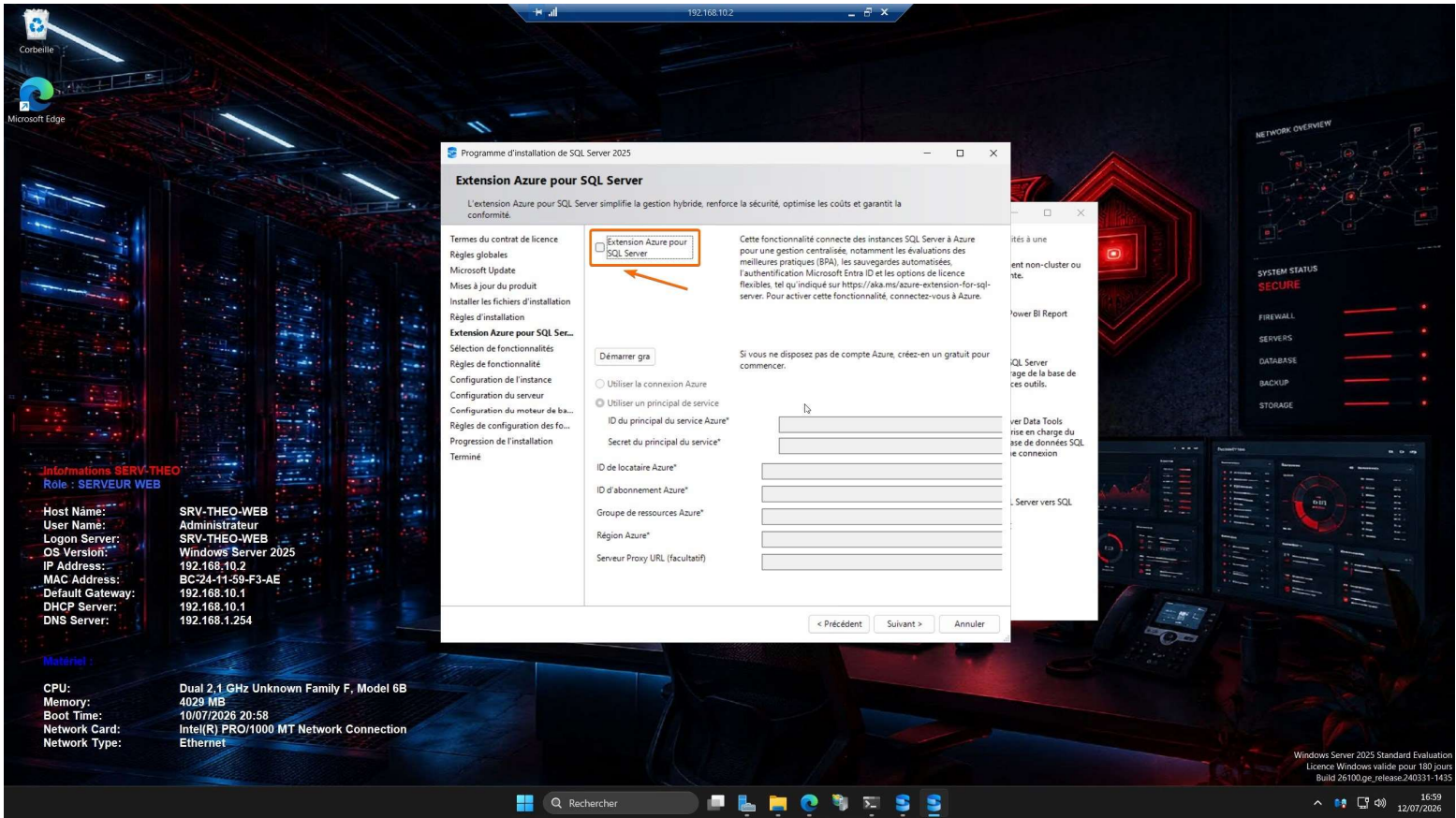
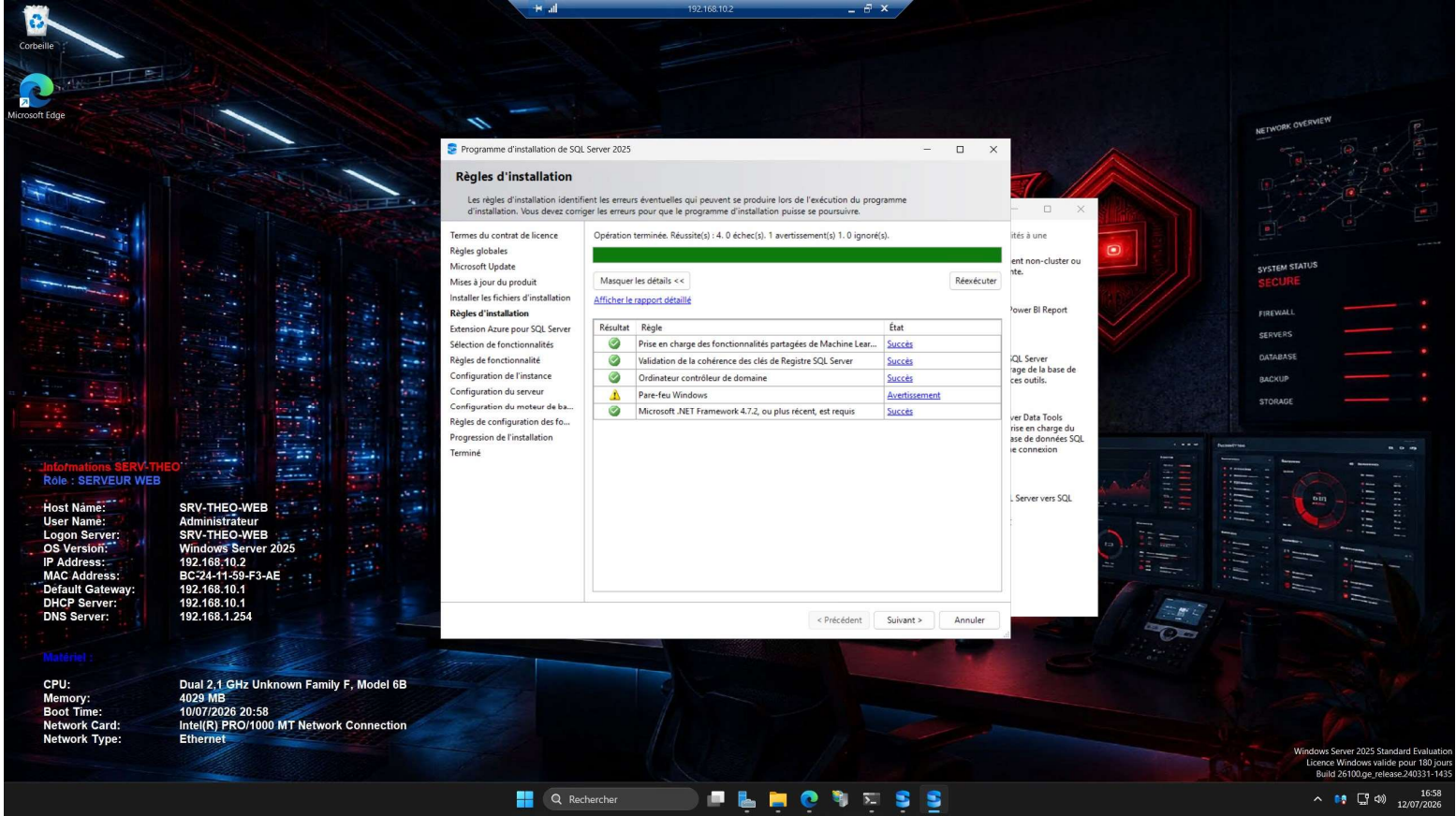
PROJET PERSONNEL – Théo BRETON

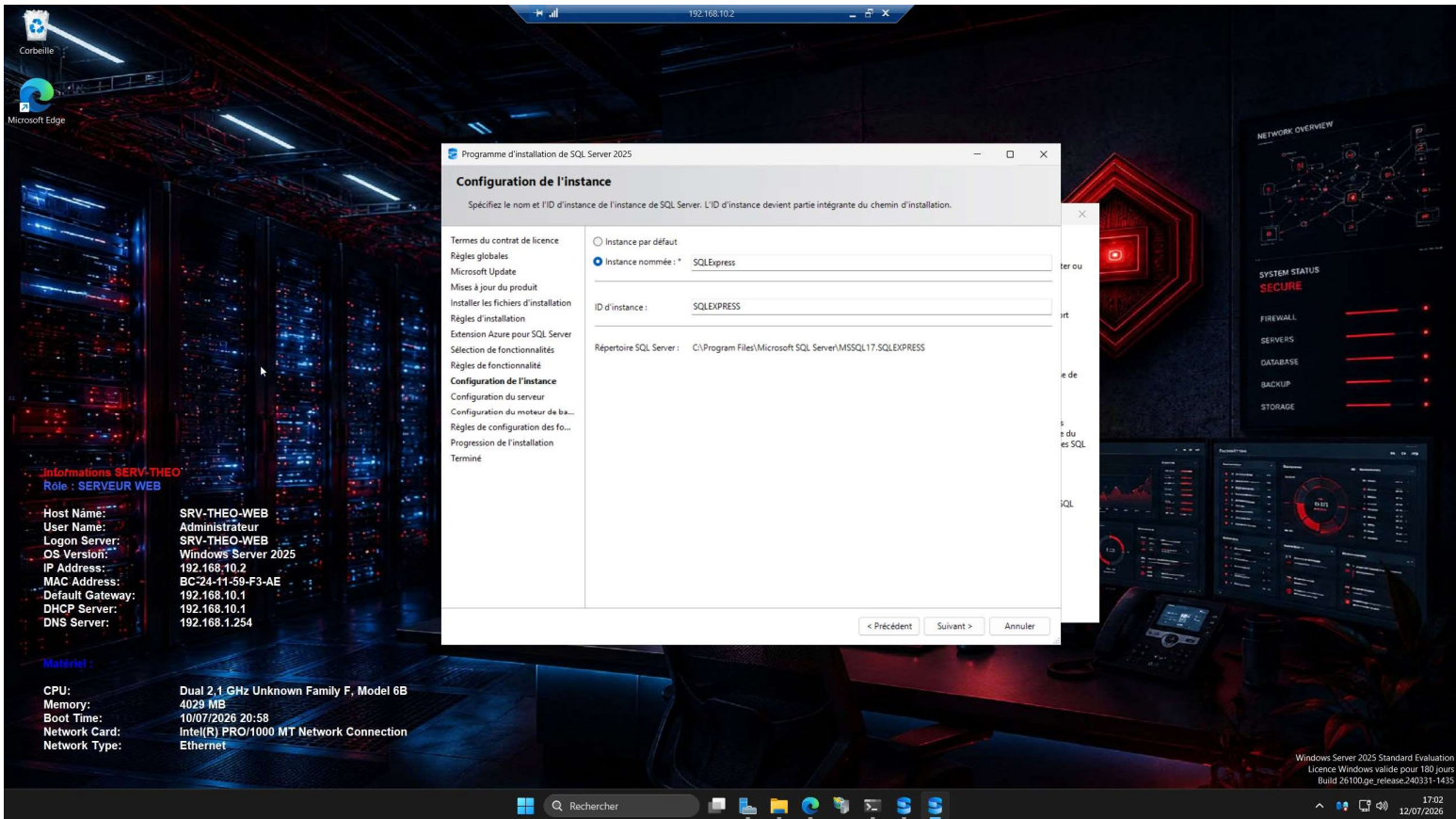
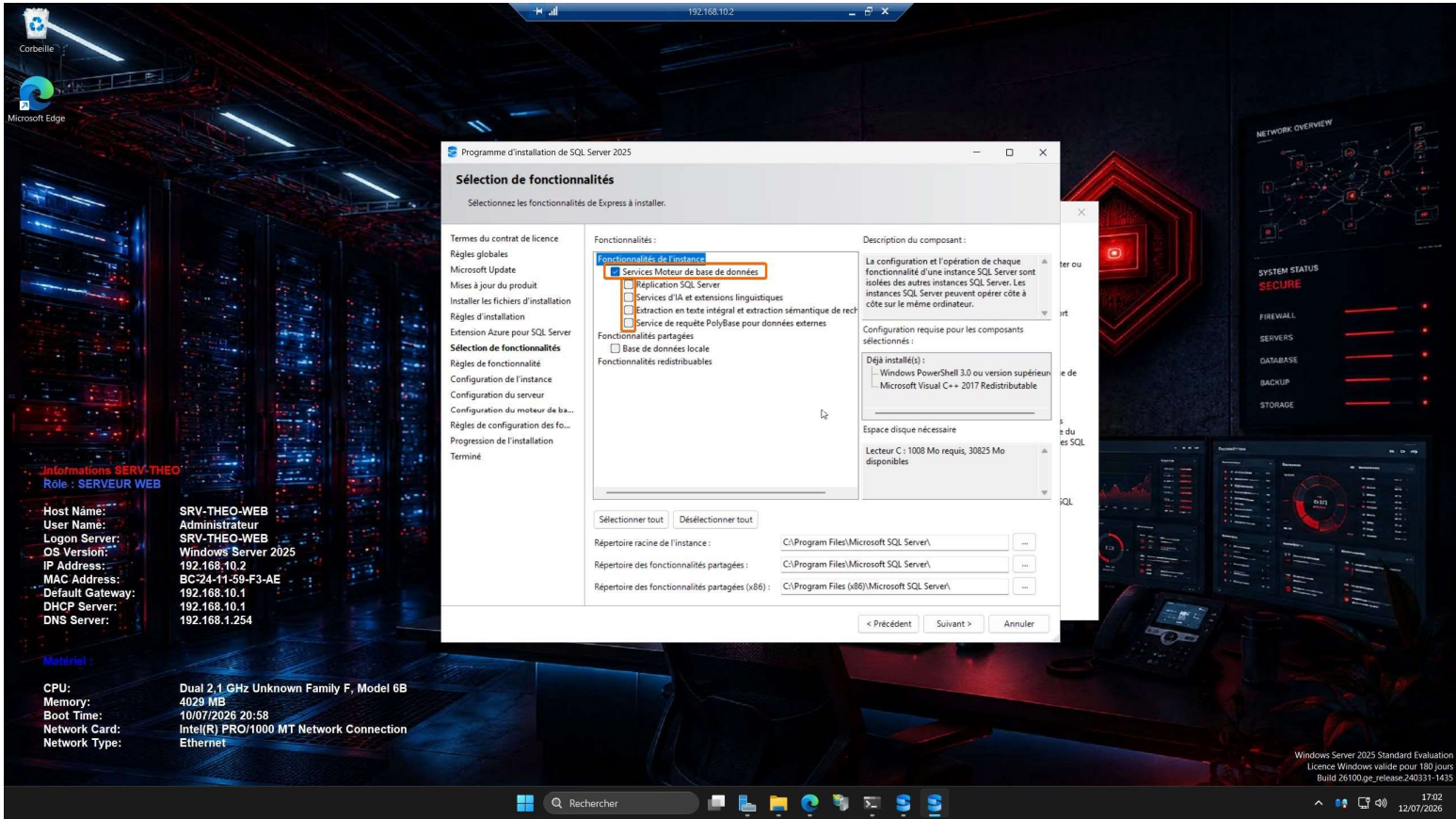
PROJET N°9 : HEBERGEMENT DE MA PLATEFORME, AVEC DIFFUSION DE MES SERVICES DOCKER

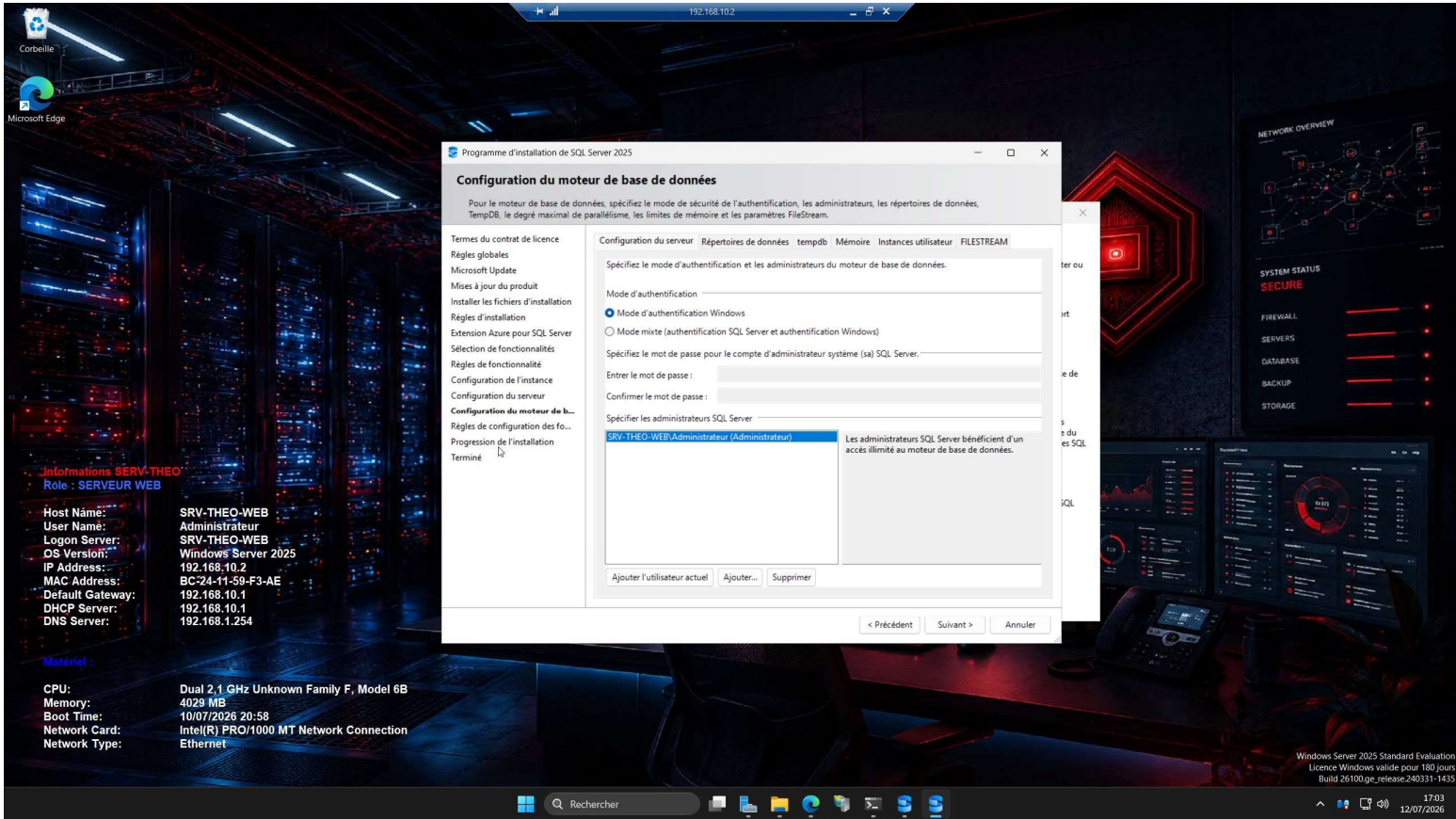
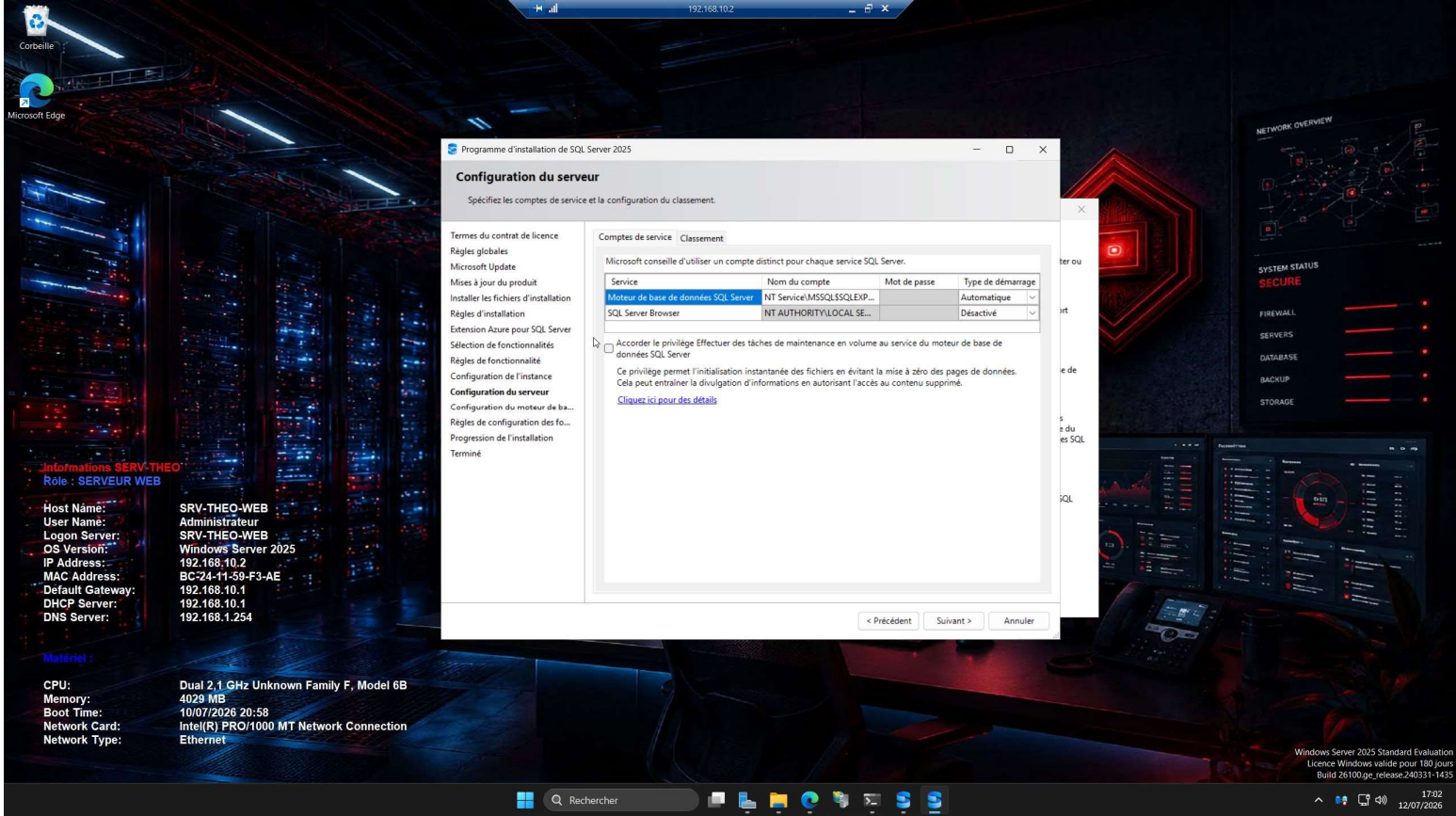


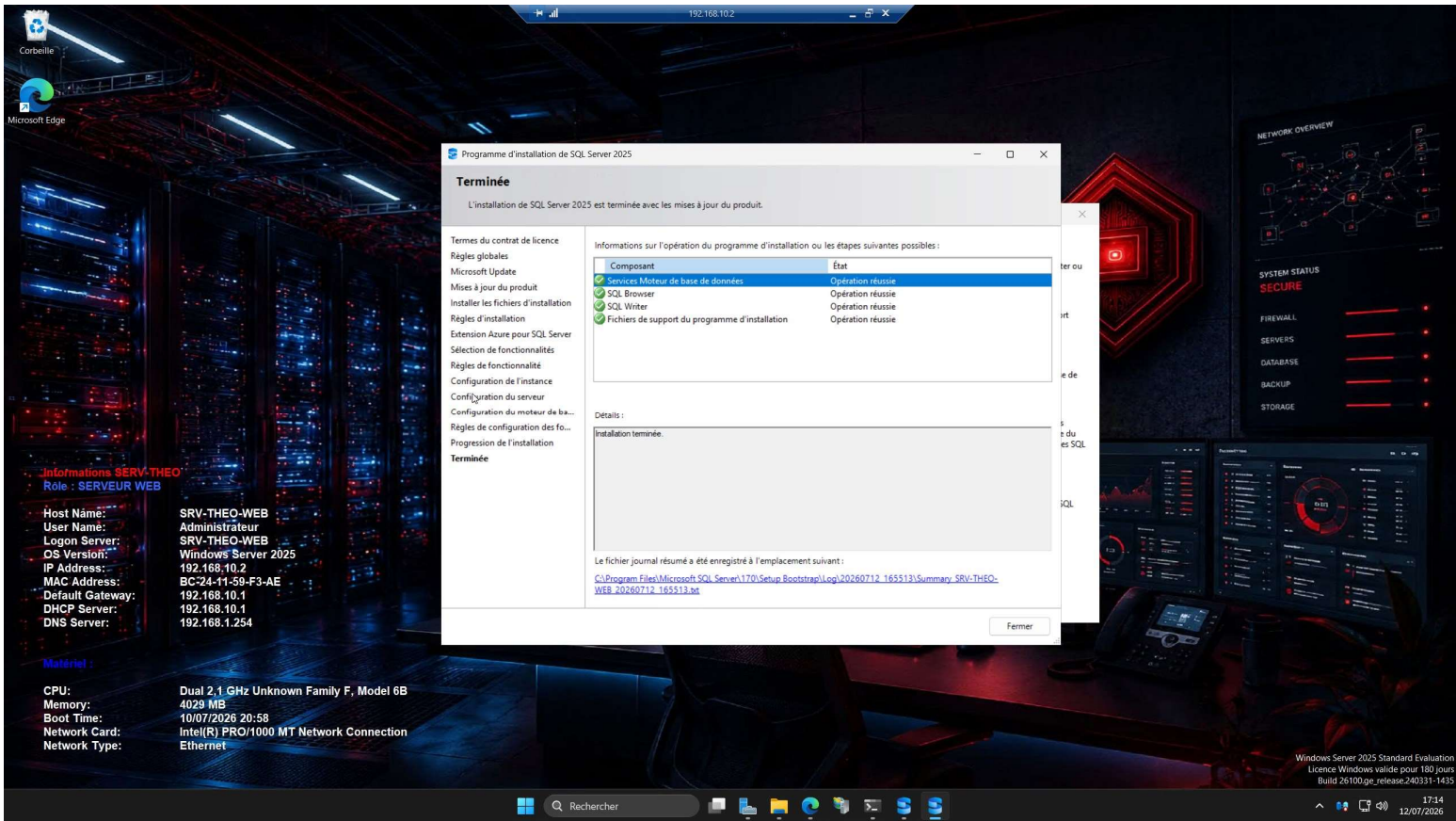
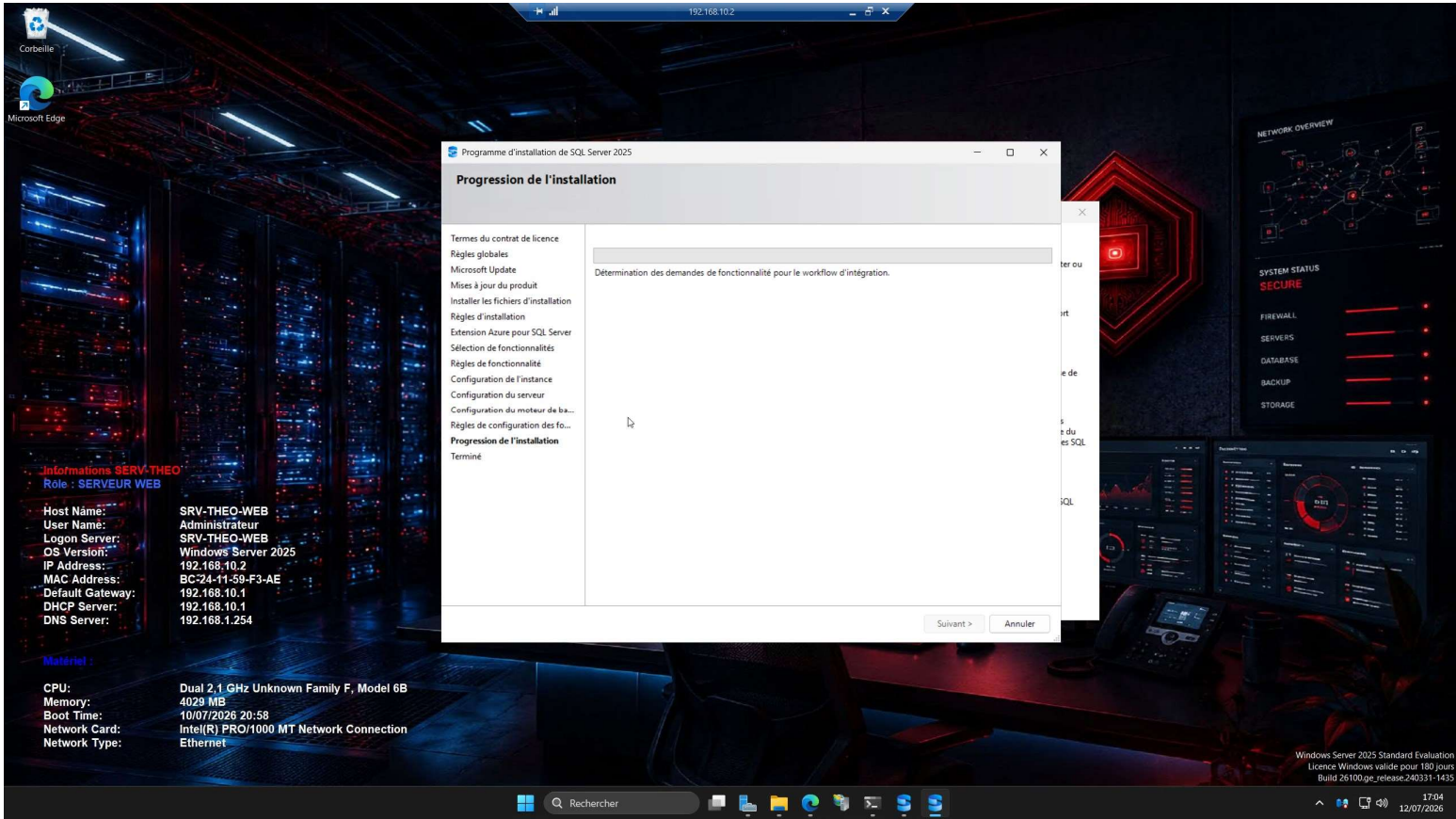




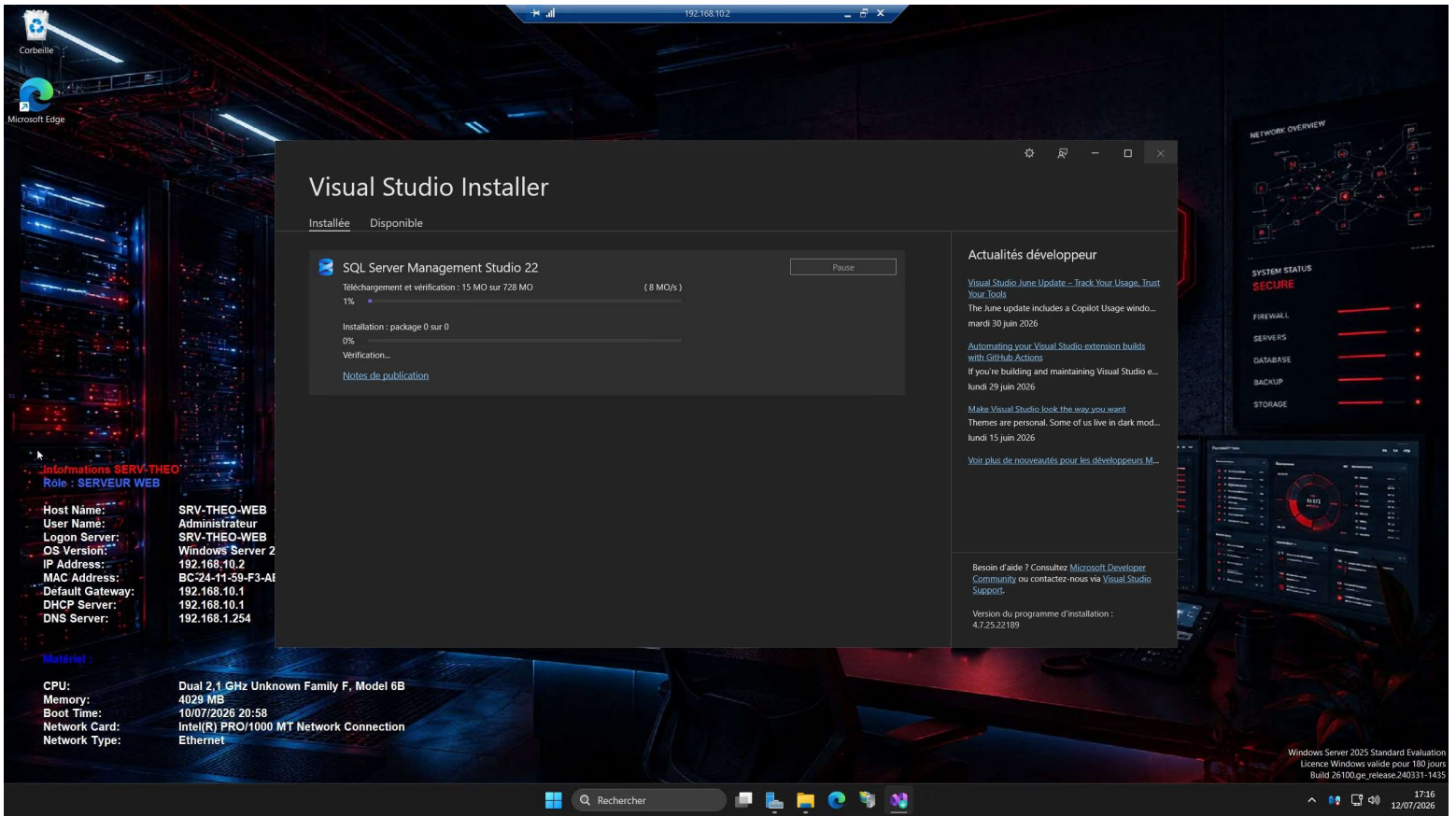
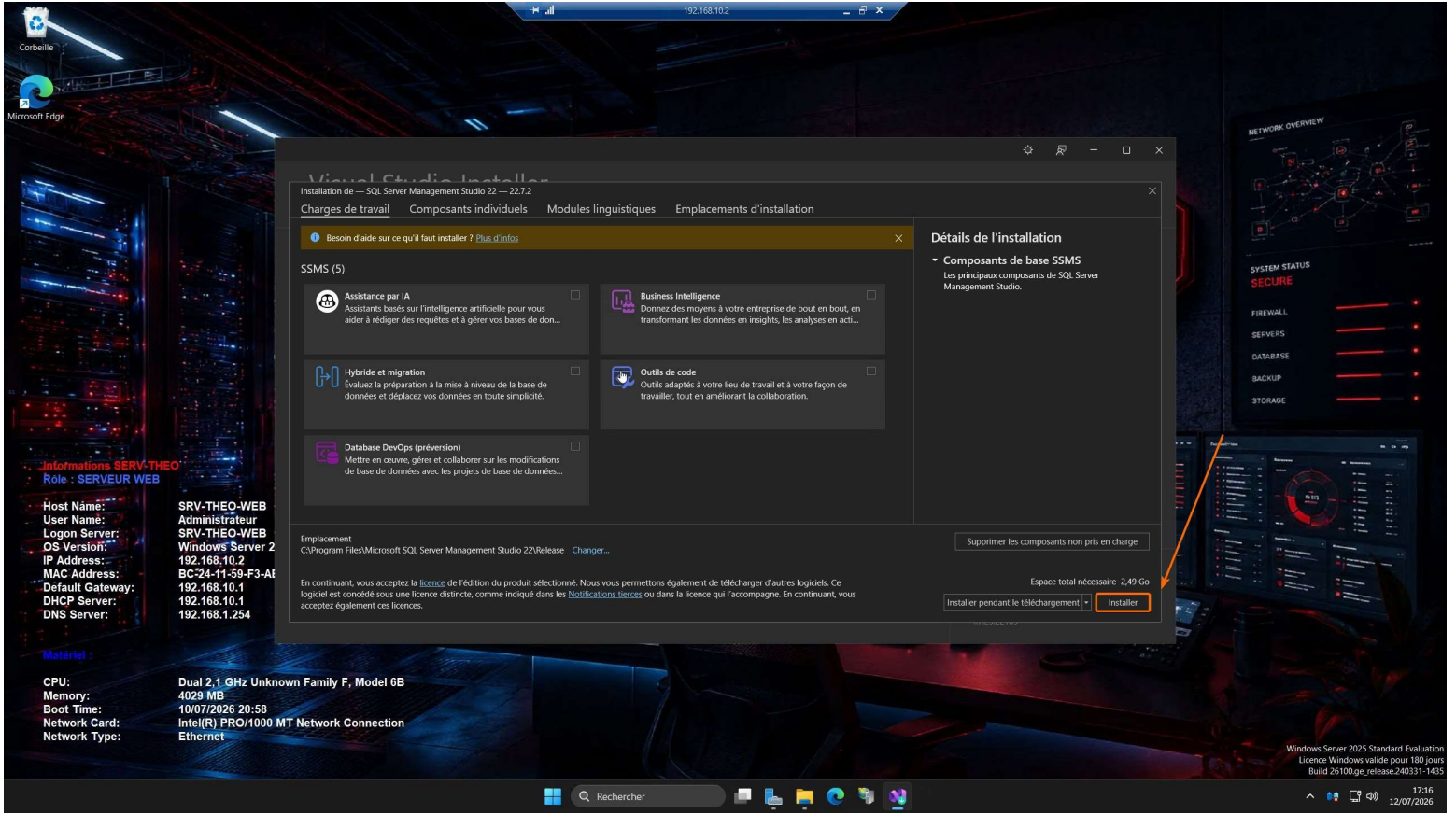


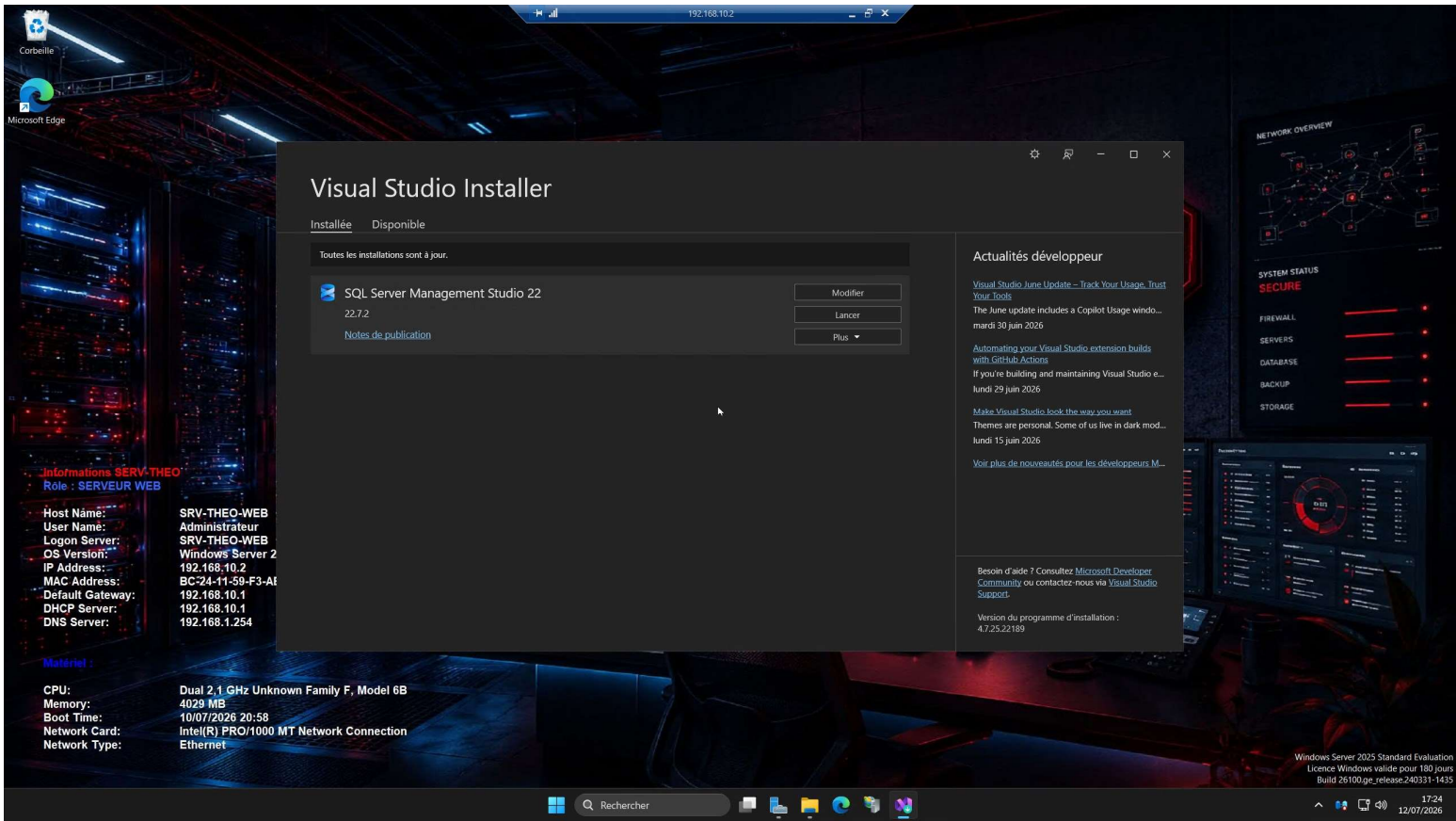






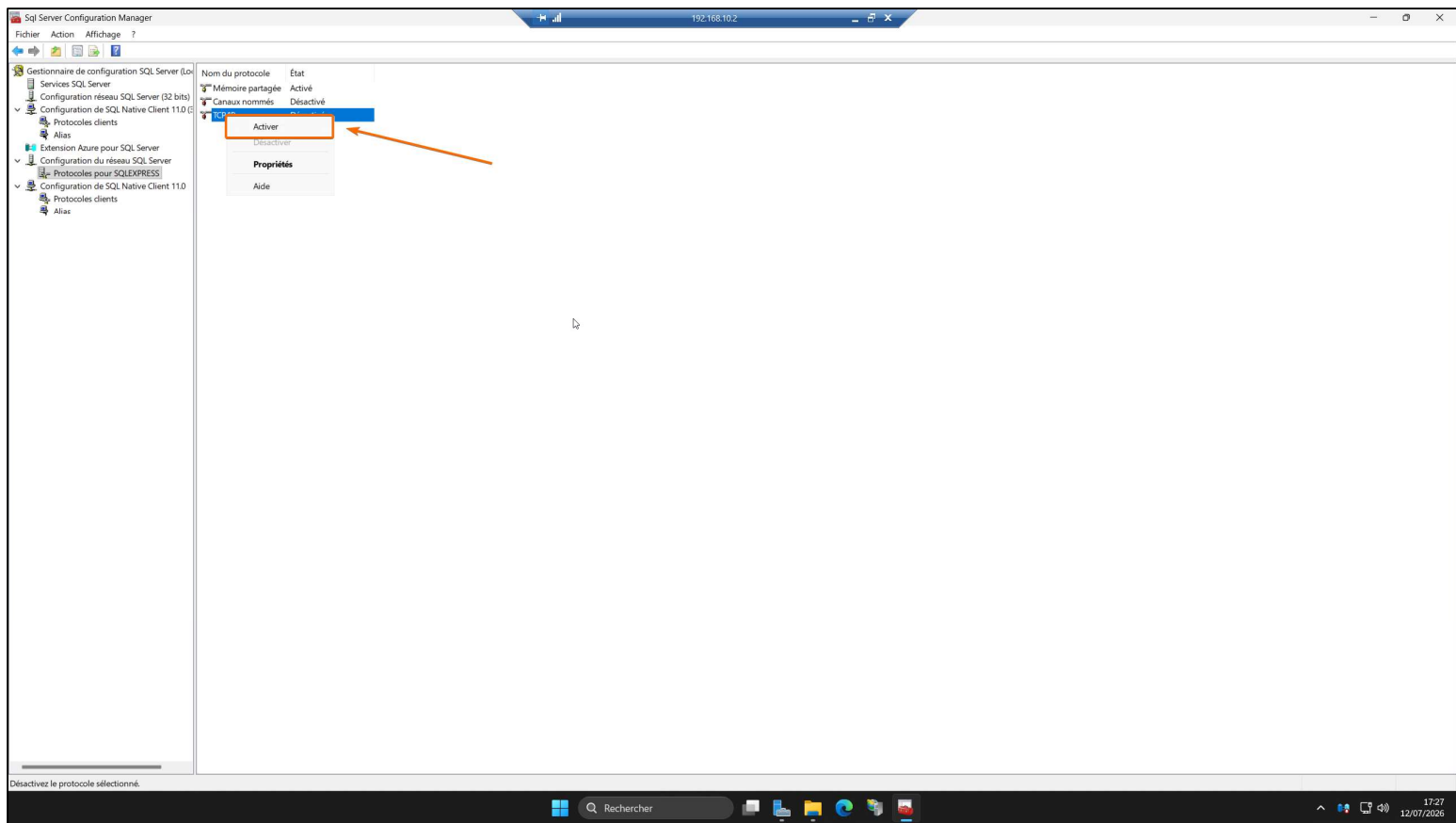
Etape 8 : Installation de SQL Server Management Studio 2025 :

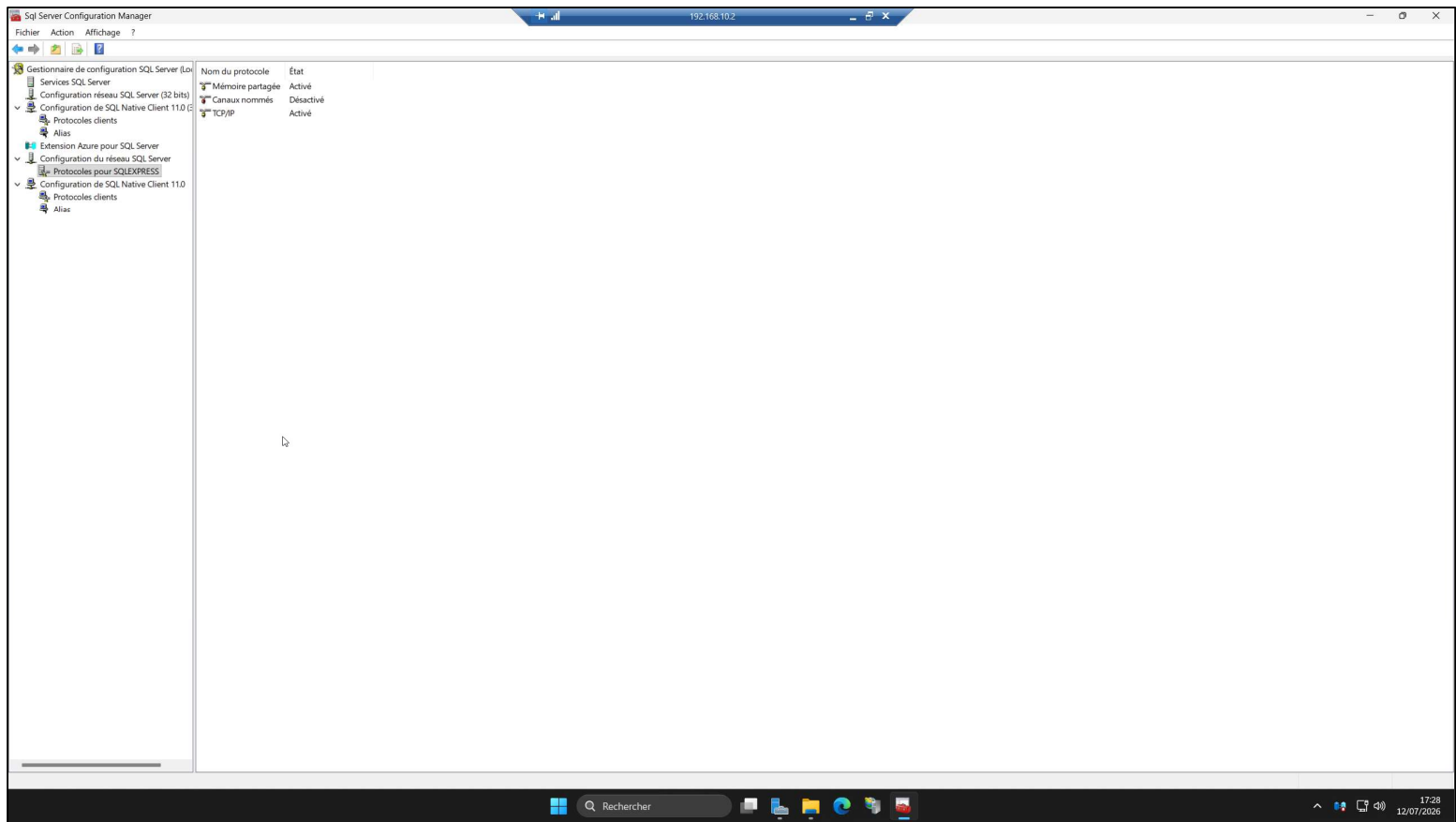




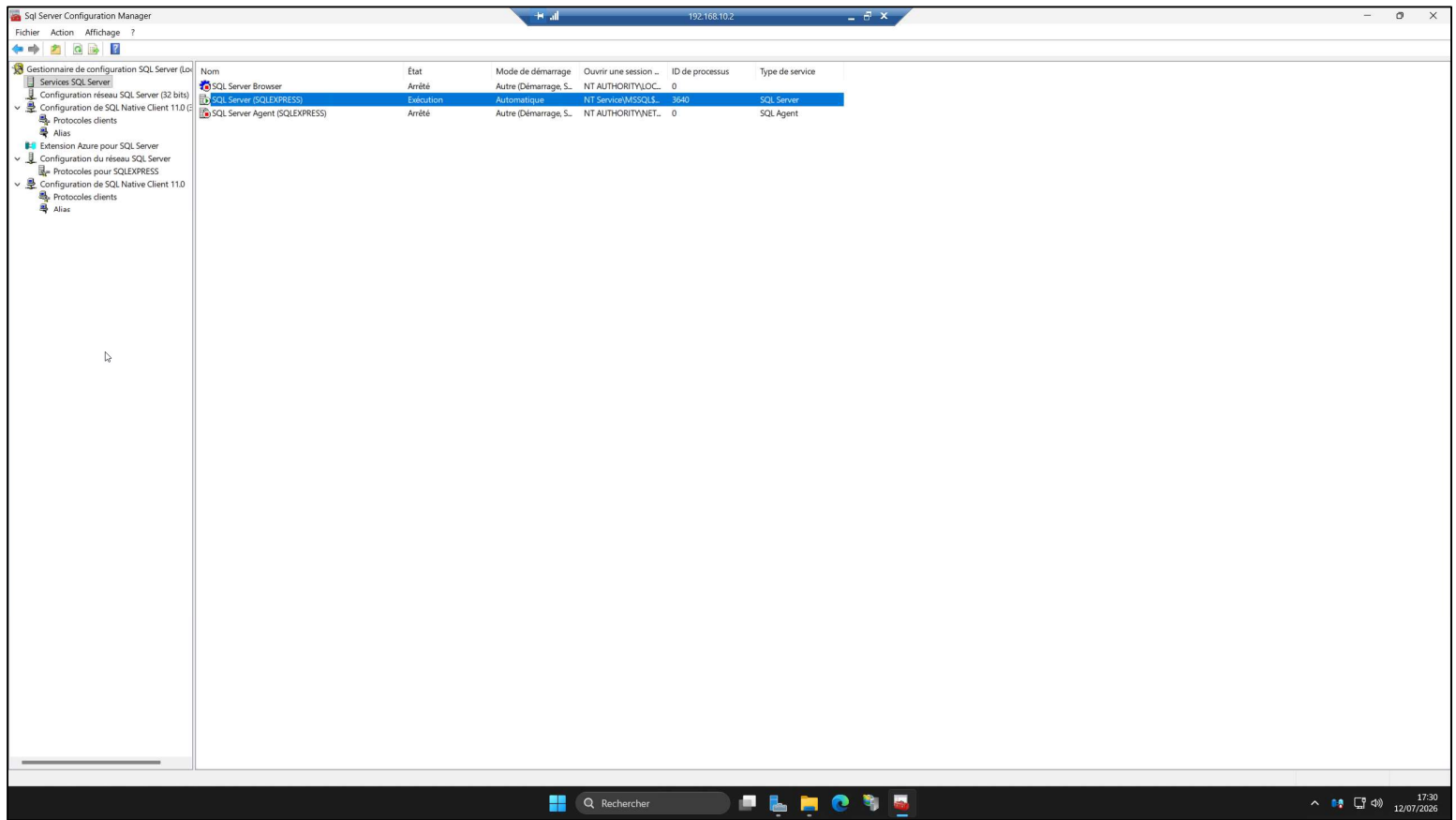
Etape 9 : Configuration de l'instance SQL Server

Ouvrir SQL Server Configuration Manager → Protocoles pour SQLEXPRESS → activer TCP/IP si un accès distant est nécessaire (redémarrage du service SQL Server).



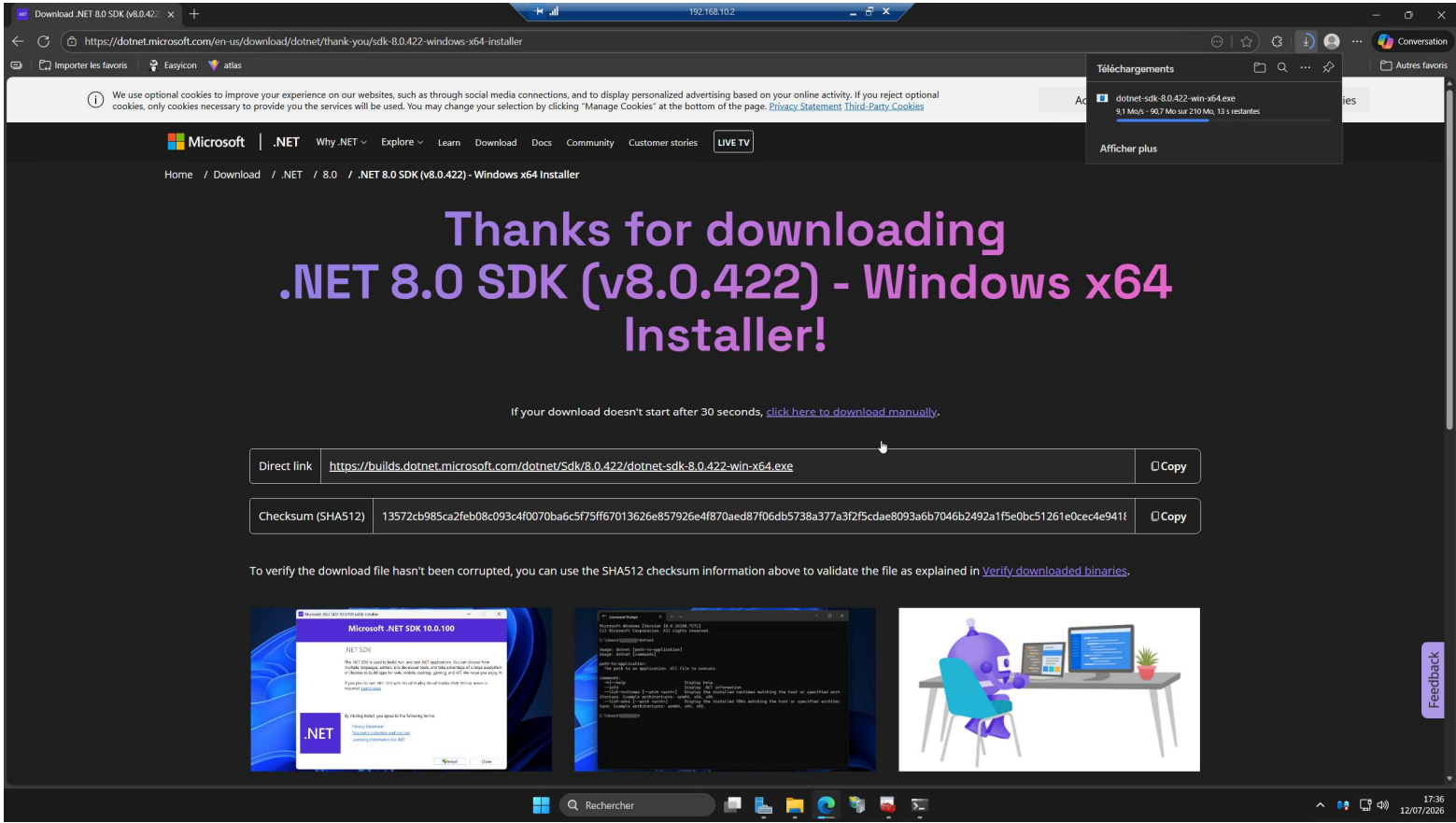


Dans SQL Server Configuration Manager → Services SQL Server, vérification que le service SQL Server (SQLEXPRESS) est en démarrage Automatique.



Etape 10 : Publier l'application (poste de développement)

Installation de **SDK 8.0** :



```
cd TheoLabSphere.Api
dotnet publish -c Release -r win-x64 --self-contained false -o .\publish
```

Cette commande génère un dossier publish contenant TheoLabSphere.Api.dll, TheoLabSphere.Api.exe, web.config (généré automatiquement, indispensable pour IIS), le dossier wwwroot et toutes les dépendances.

```
Administrateur : invite de con X + 192.168.10.2 - X
Microsoft Windows [version 10.0.26100.32230]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Administrateur>cd C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api

C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api>dotnet publish -c Release -r win-x64 --self-contained false -o .\publish

Bienvenue dans .NET 8.0 !
-----
Version du kit SDK : 8.0.422

Télémetrie
-----
Les outils .NET collectent des données d'utilisation qui nous aident à améliorer votre expérience utilisateur. Elles sont collectées par Microsoft et partagées par la communauté. Vous pouvez refuser l'adhésion à la télémétrie en affectant la valeur '1' ou 'true' à la variable d'environnement DOTNET_CLI_TELEMETRY_OPTOUT via l'interpréteur de commandes de votre choix.

Pour plus d'informations sur la télémétrie des outils CLI .NET, accédez à https://aka.ms/dotnet-cli-telemetry
-----

Installation d'un certificat de développement HTTPS ASP.NET Core.
Pour faire confiance au certificat, exécutez 'dotnet dev-certs https --trust'
En savoir plus sur HTTPS : https://aka.ms/dotnet-https
-----

écrivez votre première application : https://aka.ms/dotnet-hello-world
Découvrez les nouveautés : https://aka.ms/dotnet-whats-new
Explorez la documentation : https://aka.ms/dotnet-docs
Signaler les problèmes et trouver une source sur GitHub : https://github.com/dotnet/core
Utilisez « dotnet --help » pour afficher les commandes disponibles ou visitez : https://aka.ms/dotnet-cli
-----

Identification des projets à restaurer...
Restauration effectuée de C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api\TheoLabSphere.Api.csproj (en 29,77 sec).
TheoLabSphere.Api -> C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api\bin\Release\net8.0\win-x64\TheoLabSphere.Api.dll
TheoLabSphere.Api -> C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api\publish\

C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api>
```

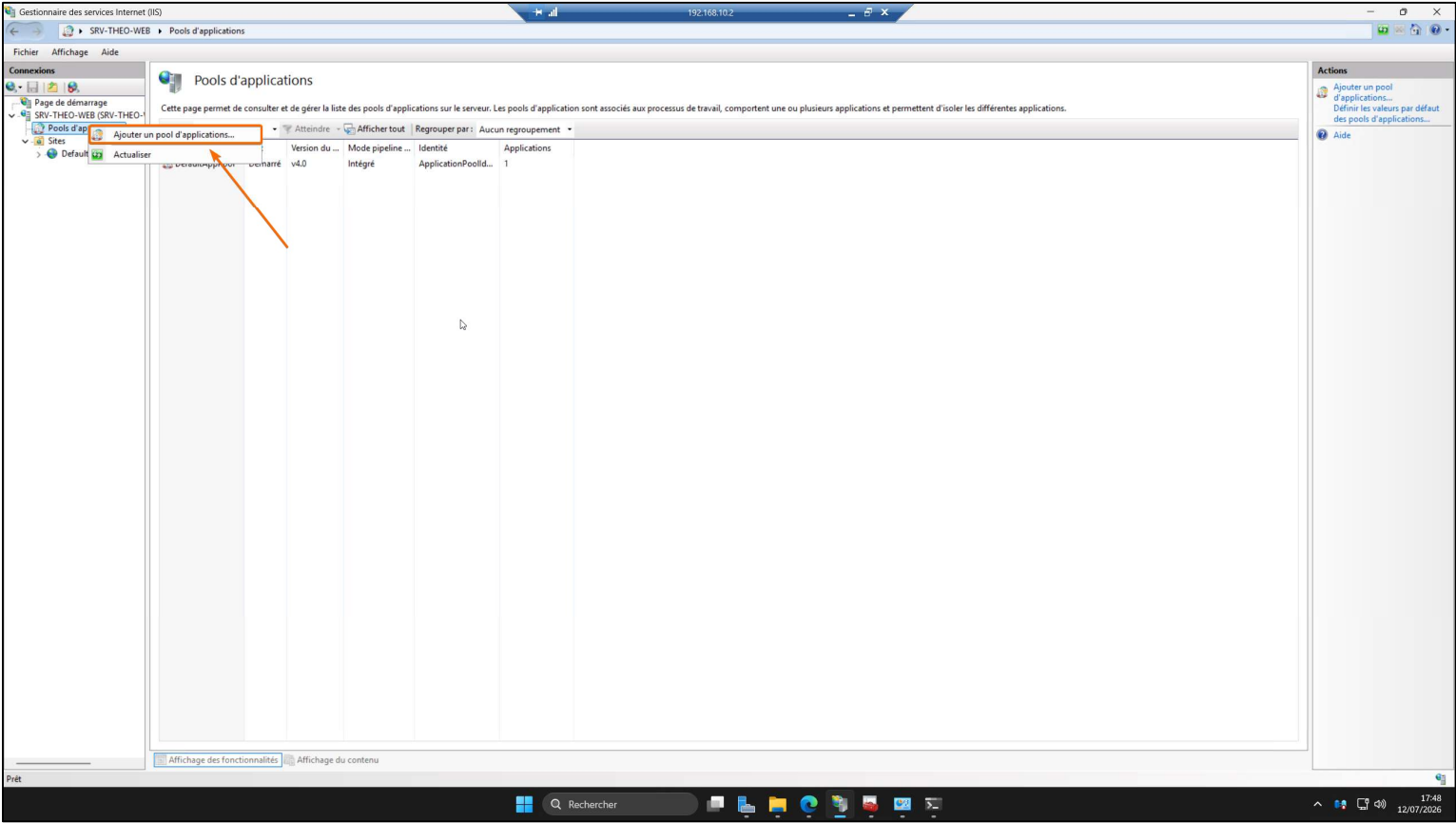
La publication a réussi : le dossier
C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api\publish\ contient
maintenant **TheoLabSphere.Api.dll**, **web.config** et tout le nécessaire.

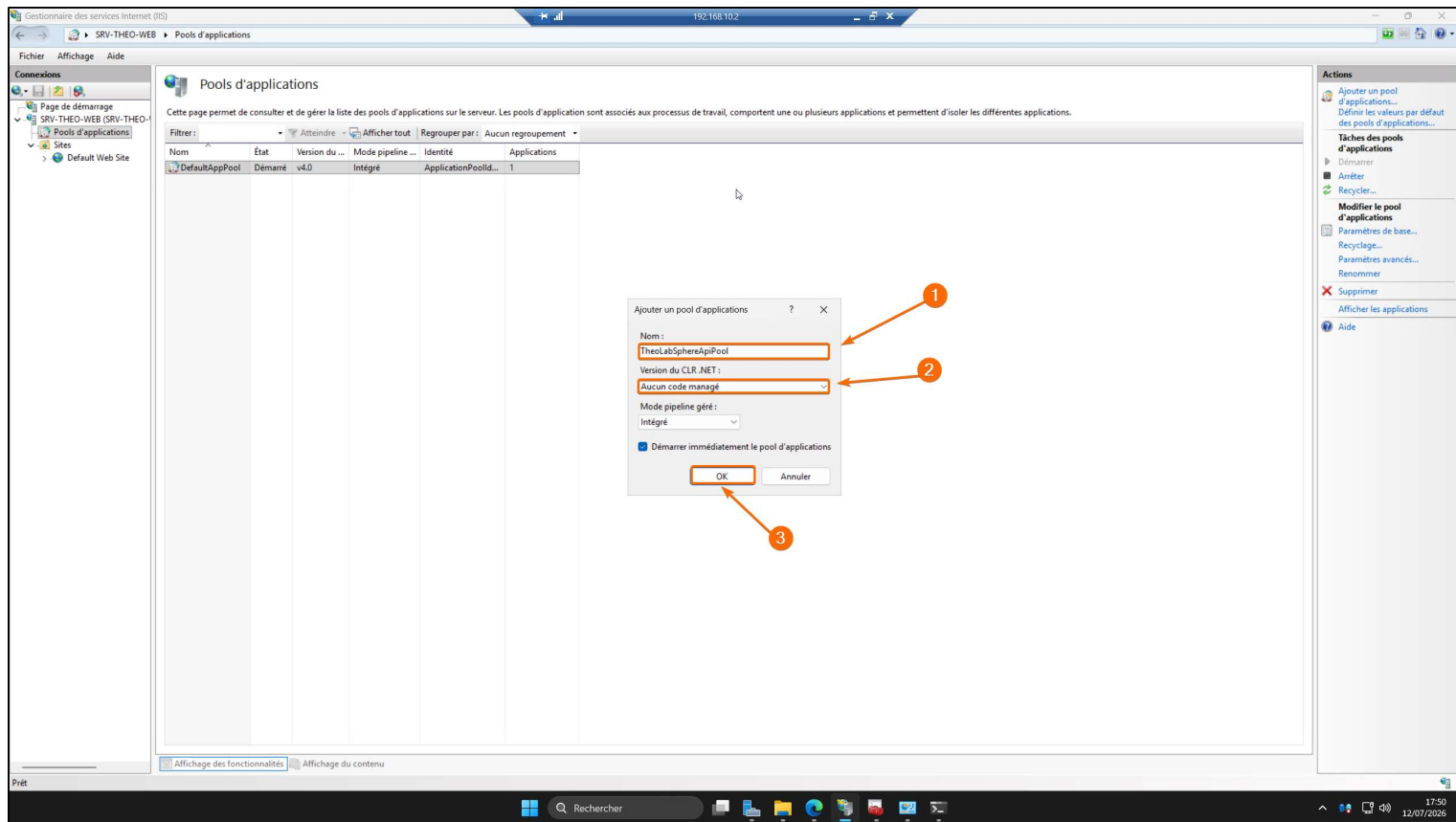


Etape 11 : Créer le site dans IIS :

Partie 1 : Créer un pool d'application dédié :

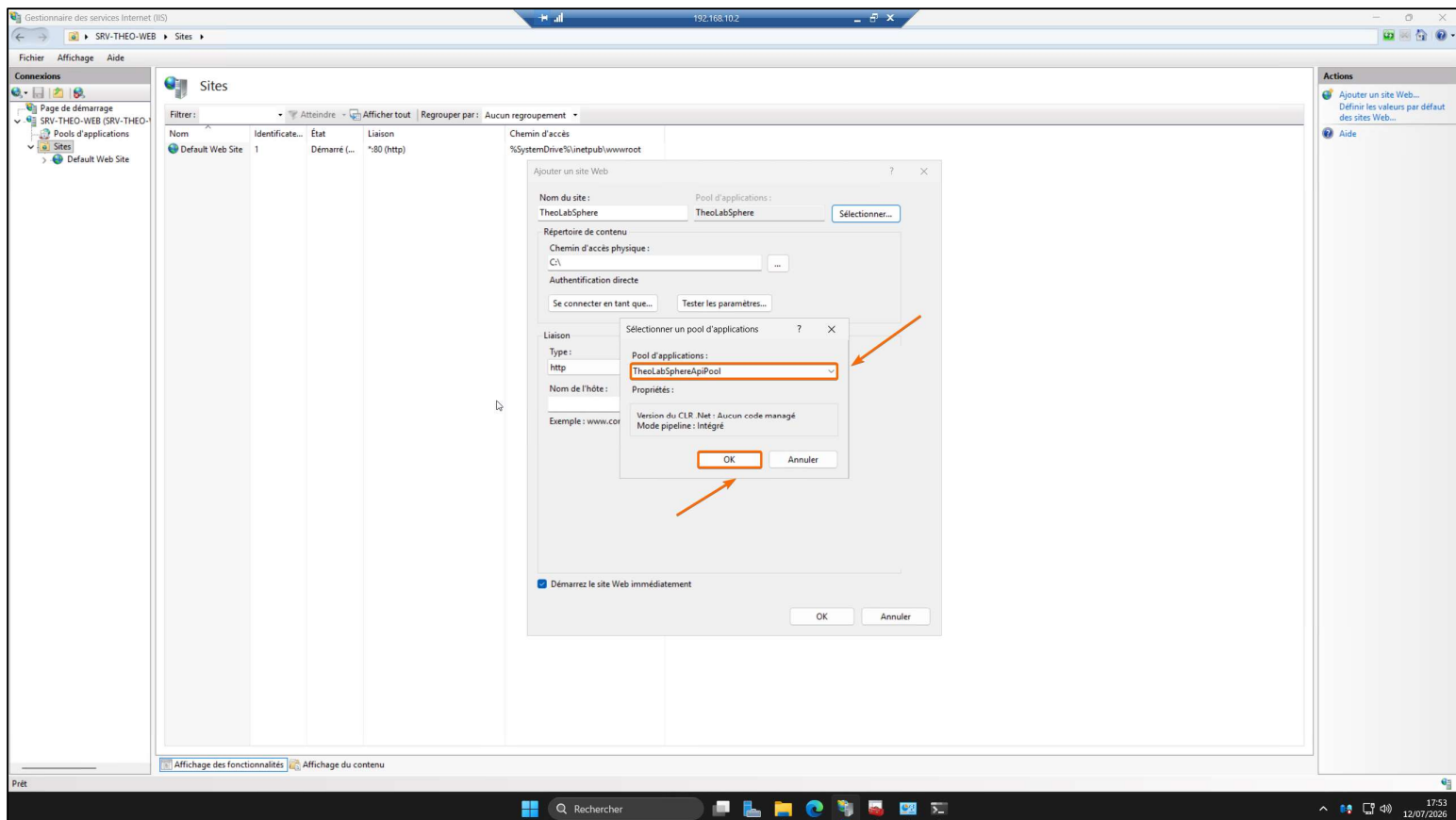
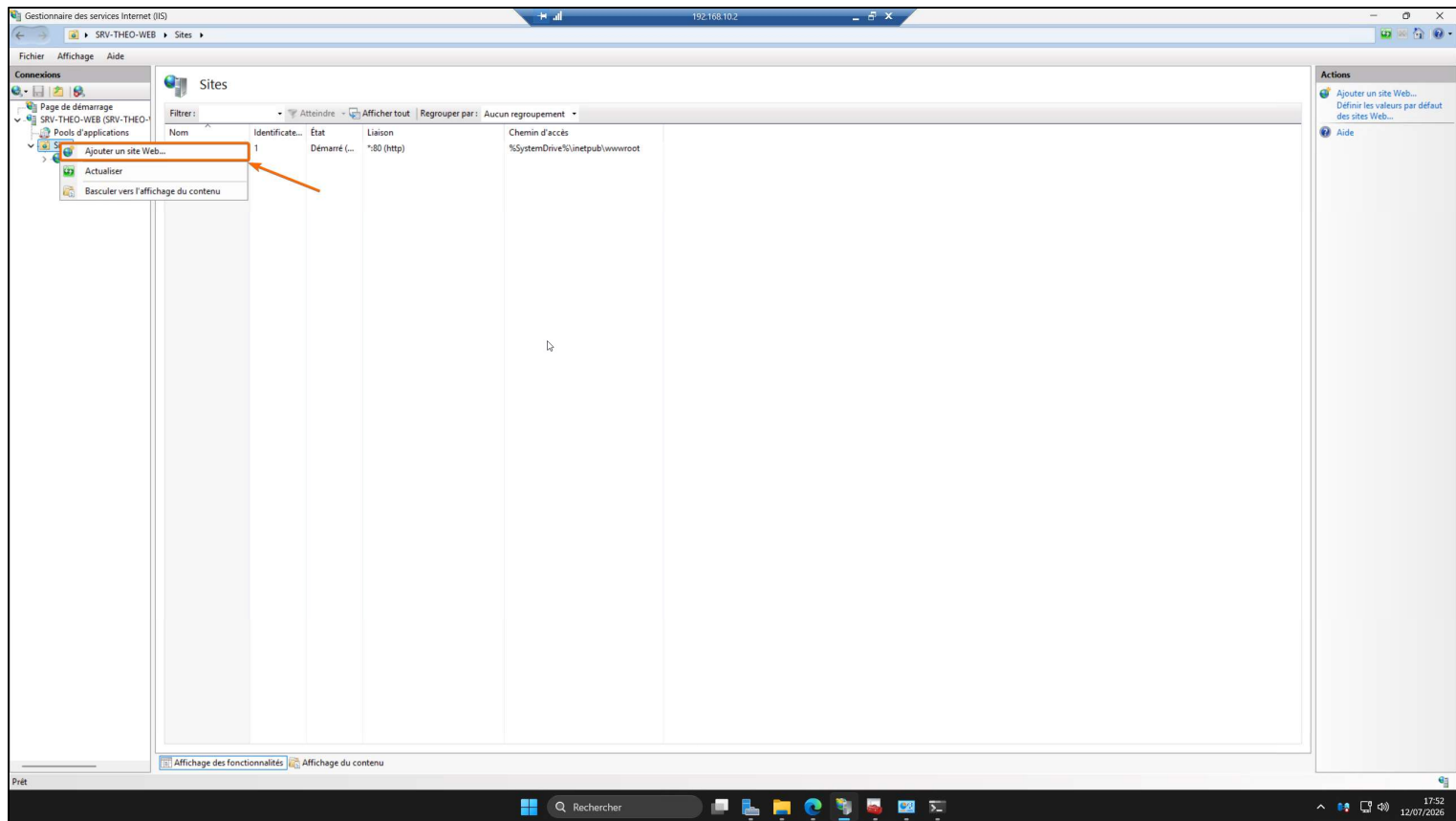
- 1. Clic droit sur Pools d'applications → Ajouter un pool d'applications.
- 2. Nom : TheoLabSphereApiPool.
- 3. Version du .NET CLR : Aucun code managé (No Managed Code) — le module ANCM gère le processus .NET lui-même, IIS ne doit pas essayer de charger le CLR classique.
- 4. Mode de pipeline géré : Intégré.

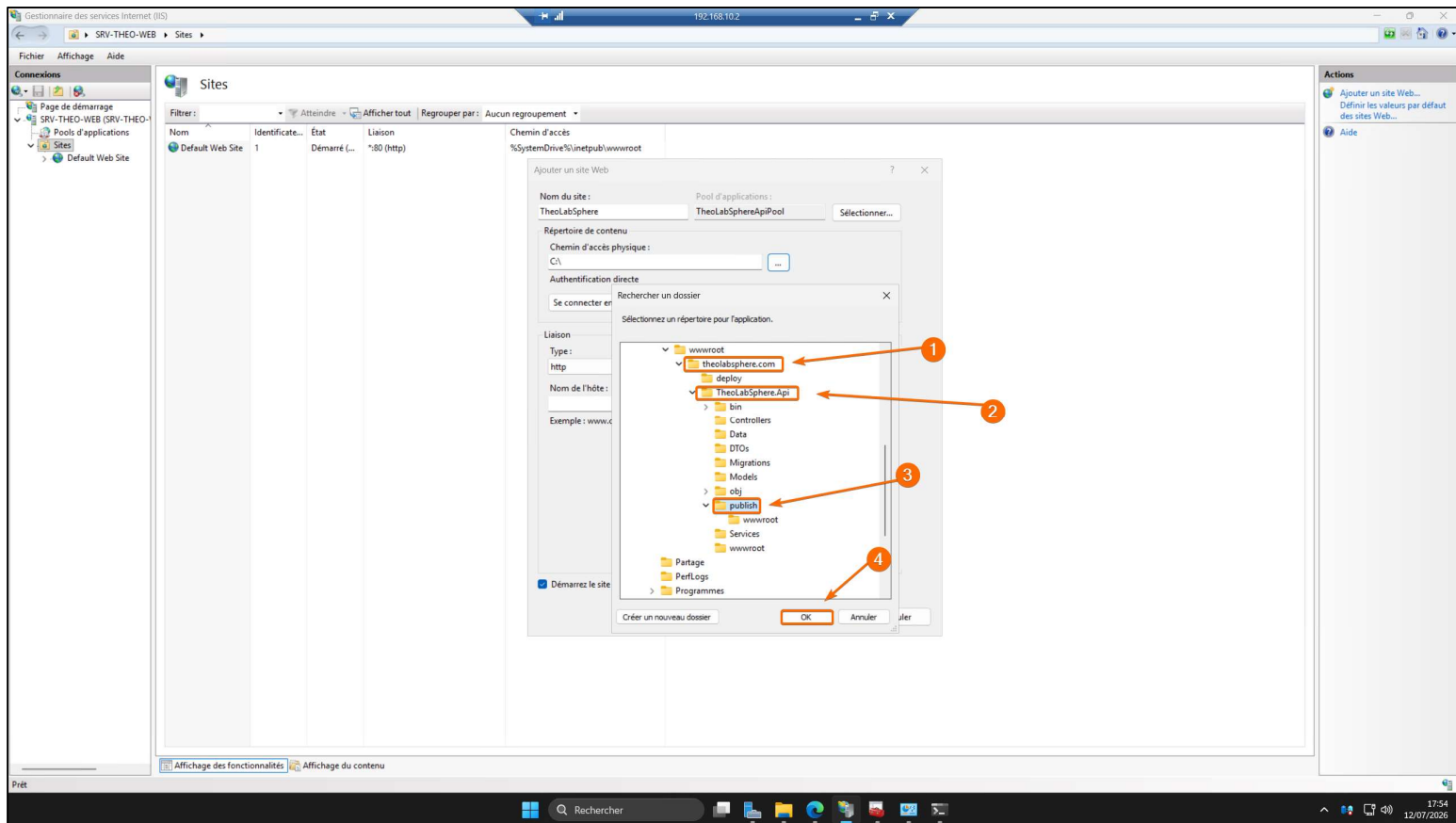




Partie 2 : Créer le site :

1. Clic droit sur Sites → Ajouter un site web.
2. Nom du site : TheoLabSphere.
3. Pool d'applications : sélectionner TheoLabSphereApiPool créé ci-dessus.
4. Chemin d'accès physique : C:\inetpub\theolabsphere\publish (le dossier copié à l'étape 7).
5. Liaison : type http, port 80 (ajouter une liaison https/443 une fois le certificat en place, voir Étape 12). Renseigner le nom d'hôte si un nom de domaine est déjà pointé vers le serveur.
6. Valider avec OK.

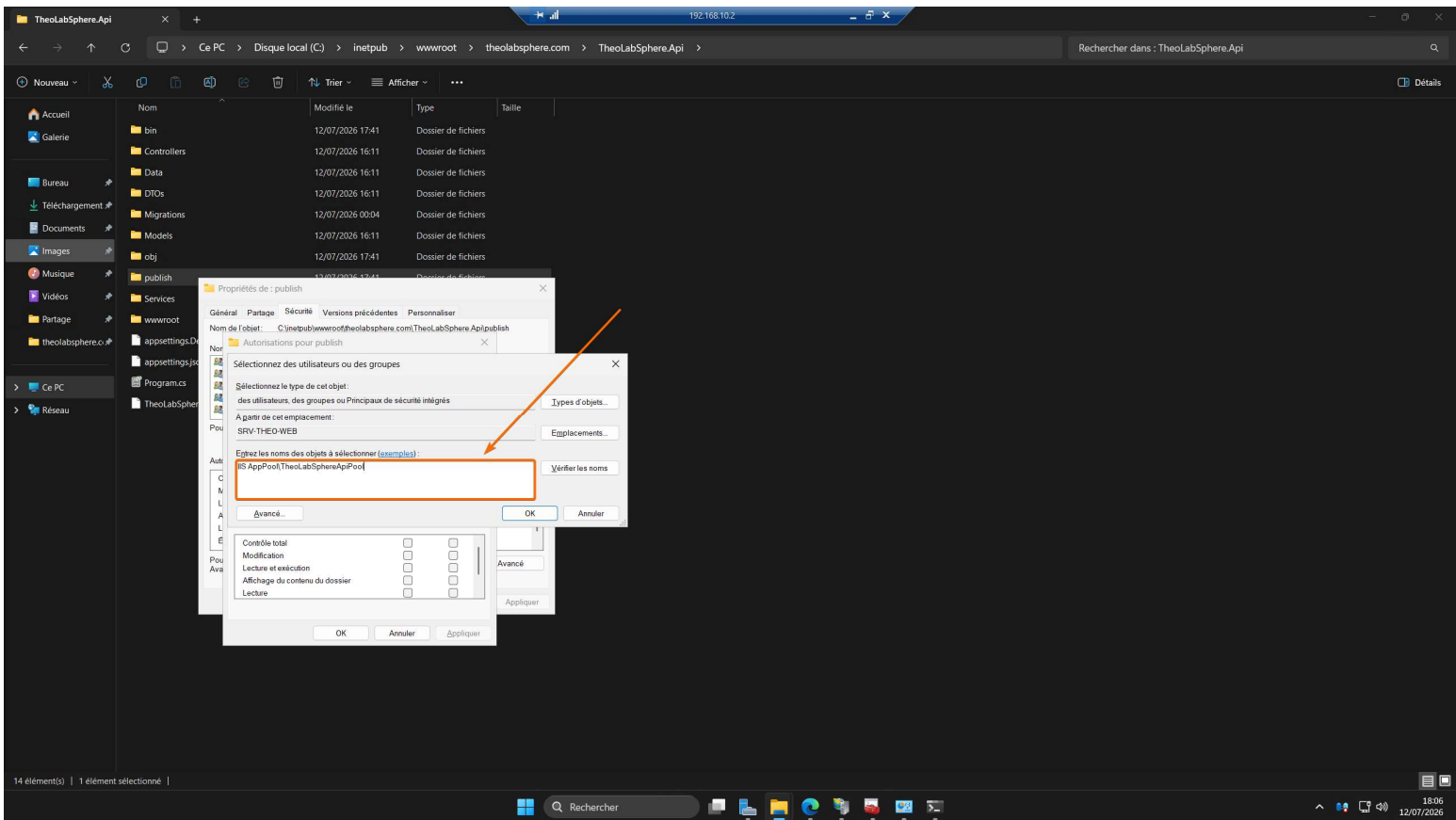
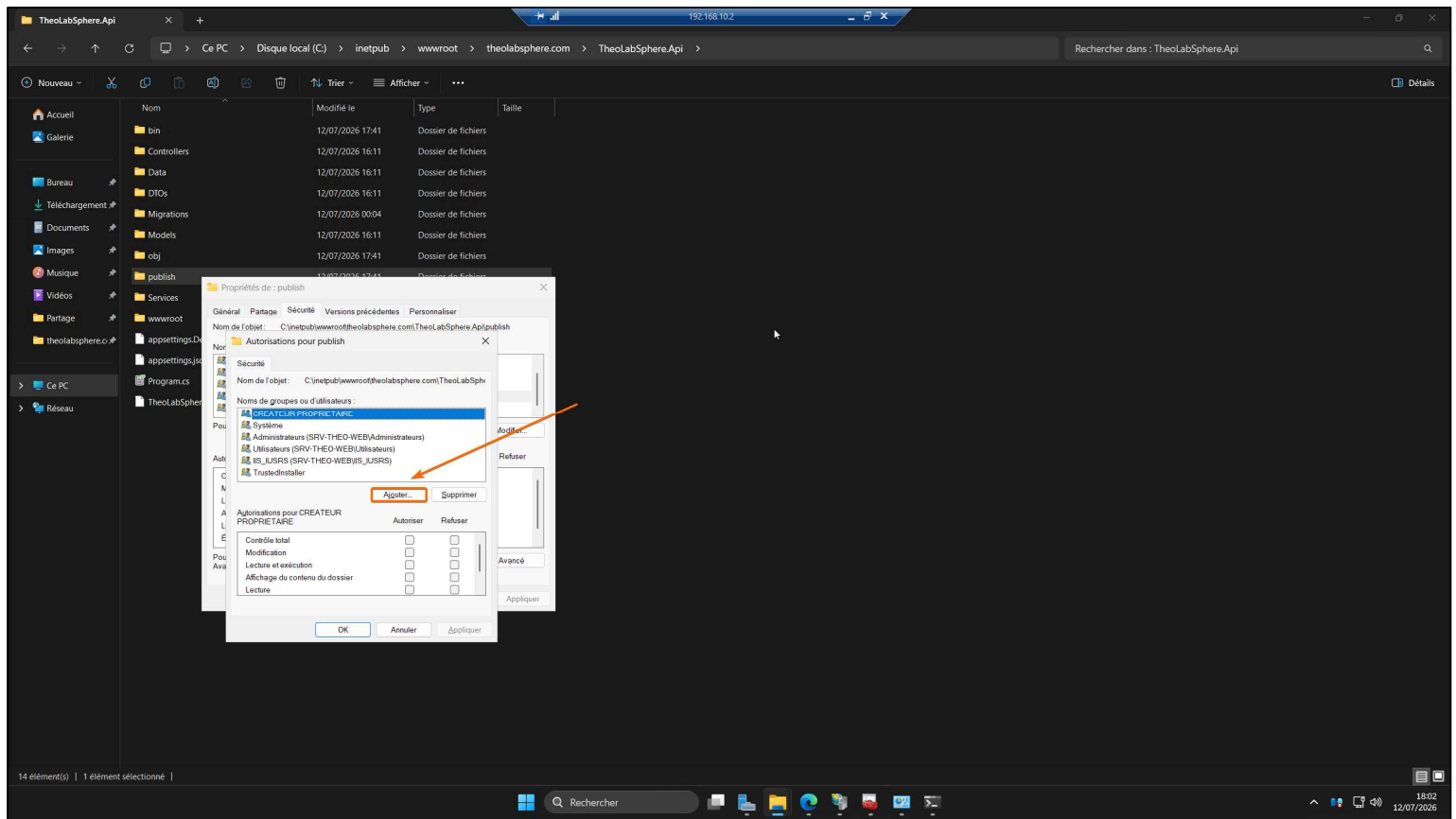


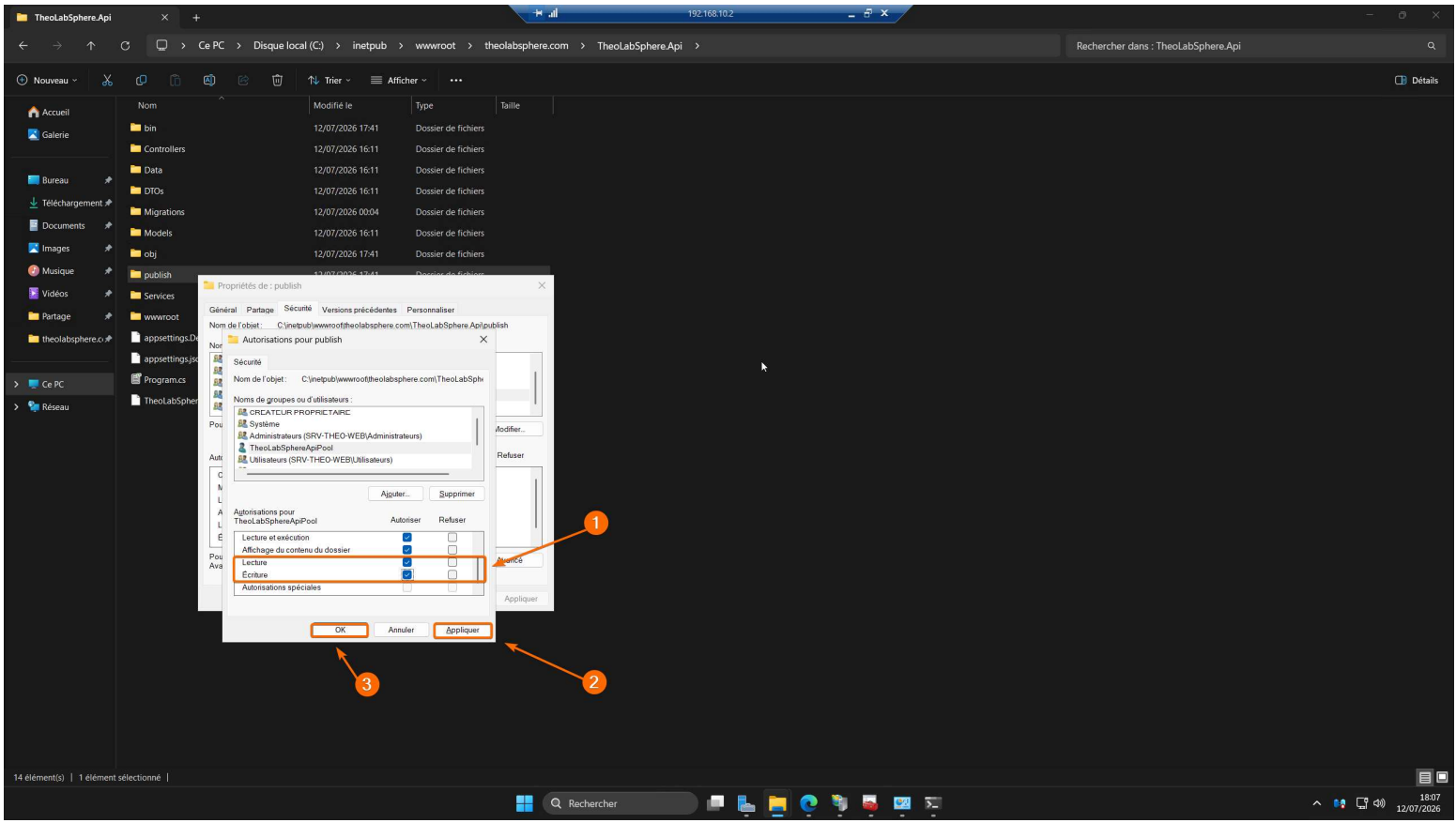


Partie 3 : Les Droit NTFS

L'identité du pool d'applications (IIS AppPool\TheoLabSphereApiPool) doit pouvoir lire les fichiers publiés et écrire dans un éventuel dossier de logs :

1. Ouvre l'Explorateur de fichiers, navigue jusqu'à C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api\.
2. Clic droit sur le dossier **publish** → **Propriétés** → onglet **Sécurité** → **Modifier**.
3. **Ajouter...** → dans le champ, tape exactement IIS AppPool\TheoLabSphereApiPool → **Vérifier les noms** (ça doit se souligner sans erreur) → **OK**.
4. Sélectionne cette entité dans la liste, coche **Lecture et exécution** (et **Modification** si tu veux que l'app puisse écrire des logs dans un dossier logs\).
5. **OK** pour valider.





Etape 12 : Configurer la chaîne de connexion et les secrets en production

Ouvrir **PowerShell en tant qu'administrateur** (clic droit sur l'icône PowerShell dans la barre des tâches → Exécuter en tant qu'administrateur) et colle :

```
Import-Module WebAdministration

$pool = "TheoLabSphereApiPool"
$vars = @(
    @{name="ConnectionStrings__Default";
value="Server=localhost\SQLEXPRESS;Database=TheoLabSphere;Trusted_Connection=True;TrustServerCertificate=True;"},
    @{name="Turnstile__Enabled"; value="false"},
    @{name="AdminAccount__BootstrapPassword"; value="ChangeMe123!"}
)

foreach ($v in $vars) {
    Add-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST" `
        -filter "system.applicationHost/applicationPools/add[@name='$pool']/environmentVariables" `
        -name "." -value $v
}

Restart-WebAppPool -Name $pool
```

Vérifier que les 3 variables sont bien enregistrées avec :

```
Get-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST" -filter
"system.applicationHost/applicationPools/add[@name='$pool']/environmentVariables" -name "."
```



PROJET PERSONNEL – Théo BRETON

**PROJET N°9 : HEBERGEMENT DE MA PLATEFORME, AVEC DIFFUSION DE
MES SERVICES DOCKER**

```
Administrateur : Windows Po... 192.168.10.2
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> Import-Module WebAdministration
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> $pool = "TheoLabSphereApiPool"
PS C:\WINDOWS\system32> $vars = @(
>>     @{name="ConnectionStrings__Default"; value="Server=localhost\SQLEXPRESS;Database=TheoLabSphere;Trusted_Connection=True;TrustServerCertificate=True;"},
>>     @{name="Turnstile__Enabled"; value="false"},
>>     @{name="AdminAccount__BootstrapPassword"; value="ChangeMe123!"}
>> )
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> foreach ($v in $vars) {
>>     Add-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST" `
>>         -filter "system.applicationHost/applicationPools/add[@name='$pool']/environmentVariables" `
>>         -name "." -value $v
>> }
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> Restart-WebAppPool -Name $pool
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> Get-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST" -filter "system.applicationHost/applicationPools/add[@name='$pool']/environmentVariables" -name "."

Collection      : {Microsoft.IIs.PowerShell.Framework.ConfigurationElement, Microsoft.IIs.PowerShell.Framework.ConfigurationElement, Microsoft.IIs.PowerShell.Framework.ConfigurationElement}
ItemXPath       : /system.applicationHost/applicationPools/add[@name='TheoLabSphereApiPool']/environmentVariables
Attributes      : {}
ChildElements   : {}
ElementTagName  : environmentVariables
Methods         :
Schema          : Microsoft.IIs.PowerShell.Framework.ConfigurationElementSchema

PS C:\WINDOWS\system32> |
```



PROJET PERSONNEL – Théo BRETON

PROJET N°9 : HEBERGEMENT DE MA PLATEFORME, AVEC DIFFUSION DE MES SERVICES DOCKER

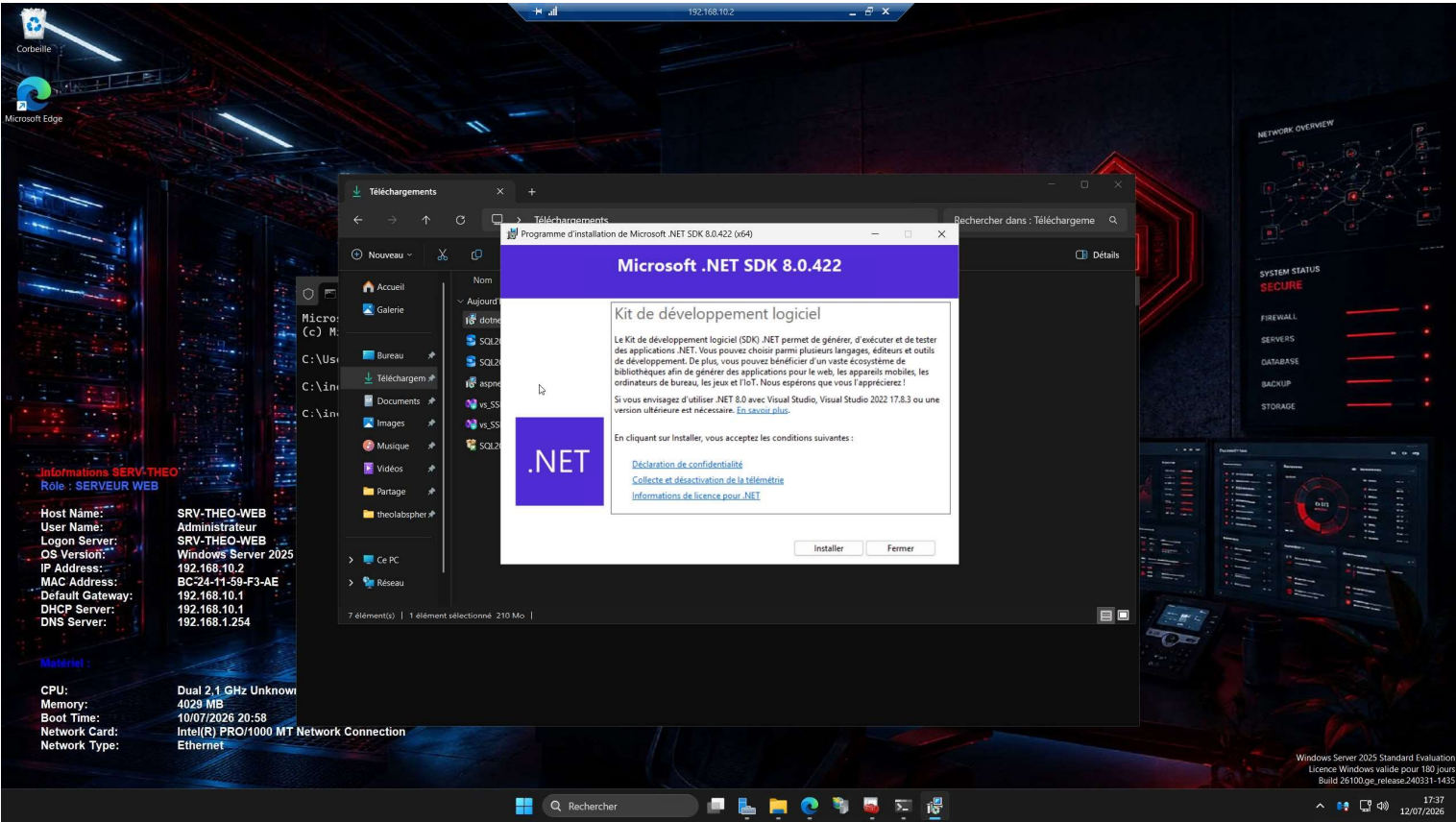
Etape 13 : Créer la base de données :

Le dossier Migrations du projet est vide : aucune migration n'a encore été générée. À faire une seule fois, avant le premier lancement en production.

Sur le poste de développement

1. Installer le SDK .NET 8.0

- # Télécharger le SDK 8.0.x (pas le Runtime, pas le Hosting Bundle) depuis
- # <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>
- # Installer, PUIS ouvrir une NOUVELLE fenêtre de terminal (PATH mis à jour uniquement dans les nouvelles sessions)



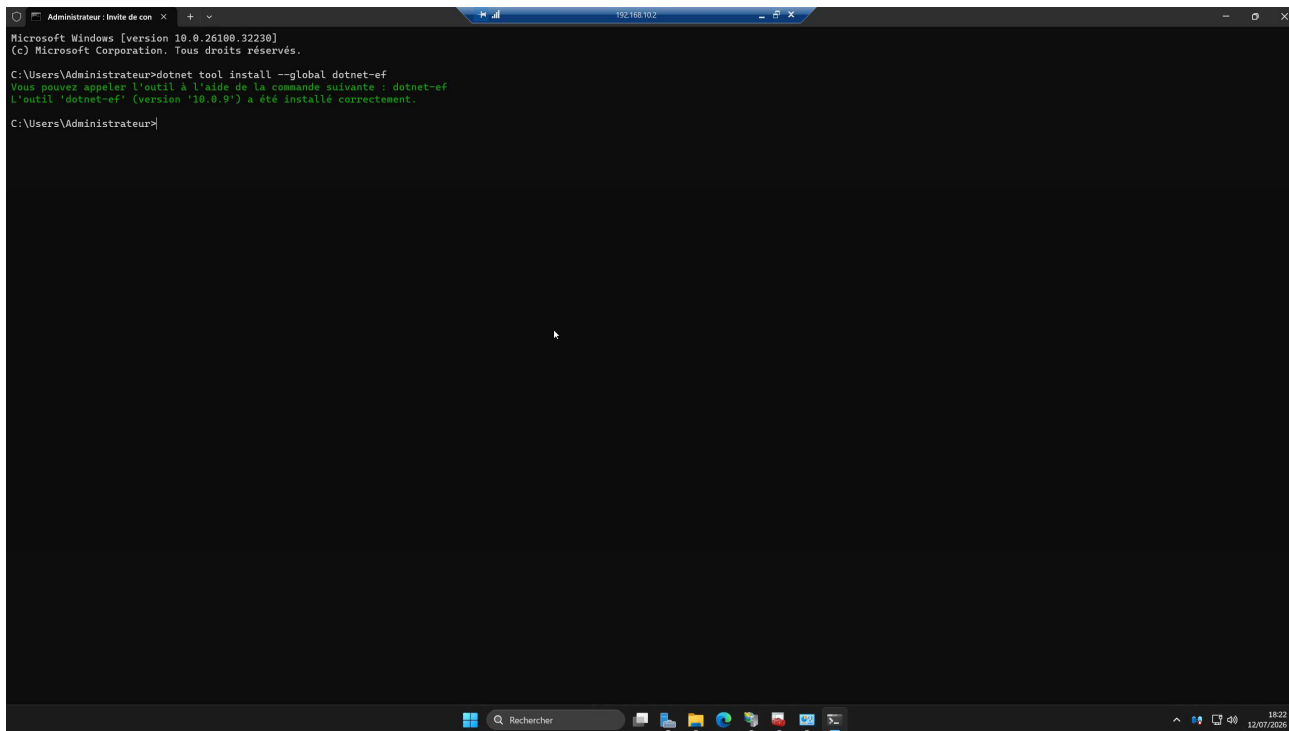
2. Correction du fichier TheoLabSphere.Api.csproj

Avant toute tentative de connexion à la base : changer InvariantGlobalization en false, sinon Microsoft.Data.SqlClient plante avec une CultureNotFoundException dès qu'il essaie d'ouvrir la connexion.

```
<InvariantGlobalization>false</InvariantGlobalization>
```

3. Installer l'outil EF Core si nécessaire : dotnet tool install --global dotnet-ef

```
dotnet tool install --global dotnet-ef
```



```
Administrateur : invite de con x
Microsoft Windows [version 10.0.26100.32230]
(c) Microsoft Corporation. Tous droits réservés.

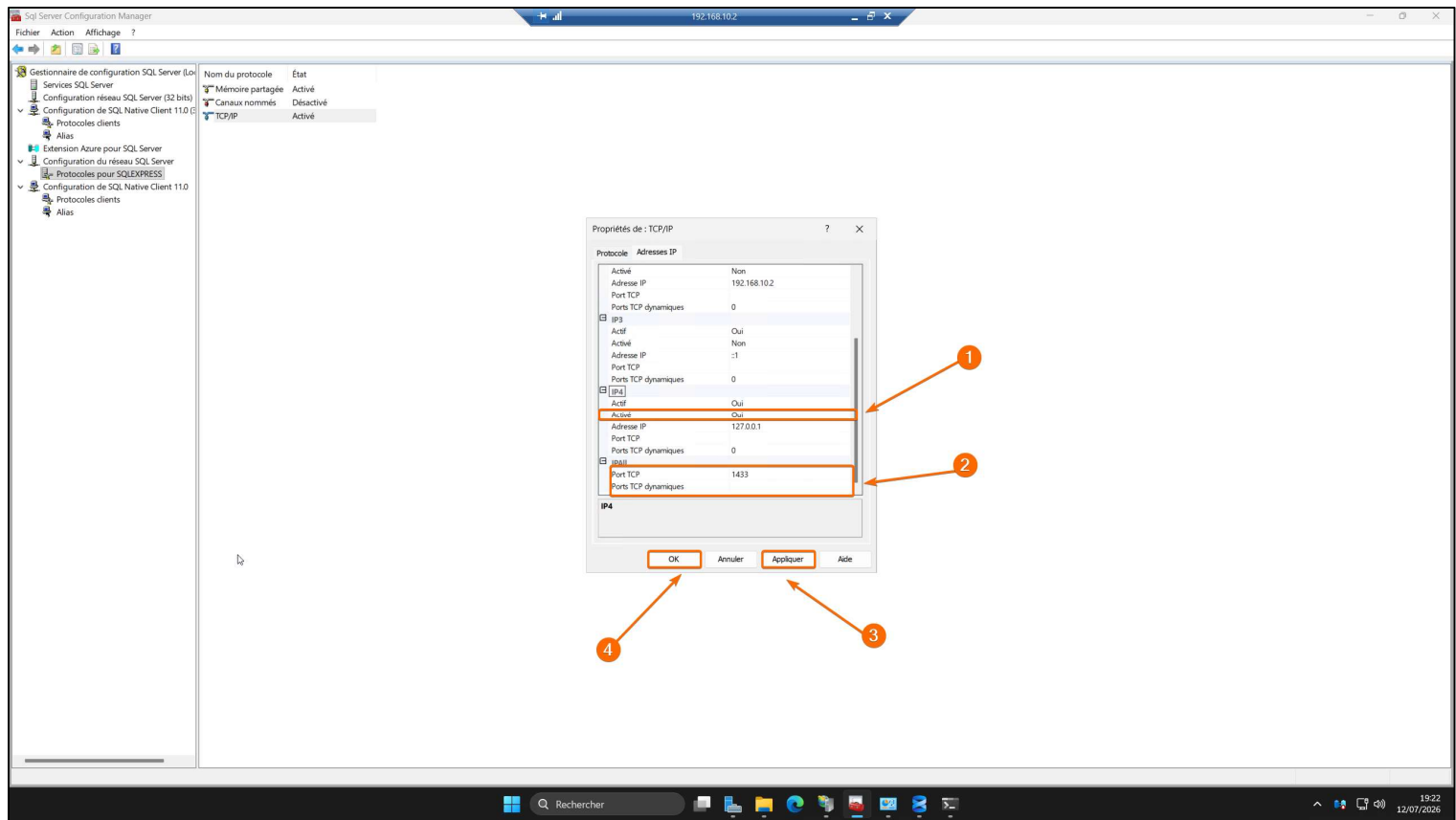
C:\Users\Administrateur>dotnet tool install --global dotnet-ef
Vous pouvez appeler l'outil à l'aide de la commande suivante : dotnet-ef
L'outil 'dotnet-ef' (version '10.0.9') a été installé correctement.

C:\Users\Administrateur>
```

4. Configurer SQL Server pour qu'il écoute réellement en TCP

Le pilote .NET (Microsoft.Data.SqlClient) ne sait utiliser QUE le TCP — ni la mémoire partagée, ni les canaux nommés que SSMS peut utiliser en repli. Il faut donc que TCP/IP soit non seulement "activé" globalement, mais aussi activé sur l'adresse IP utilisée (127.0.0.1 pour une connexion locale).

1. Ouvrir SQL Server Configuration Manager → Configuration du réseau SQL Server → Protocoles pour SQLEXPRESS.
2. TCP/IP → Propriétés → onglet Adresses IP.
3. Repérer IP4 (Adresse IP = 127.0.0.1) → mettre Activé sur Oui (il est à Non par défaut, même quand le protocole global est "Activé").
4. Descendre jusqu'à IPAll (tout en bas) → Port TCP = 1433 → Ports TCP dynamiques = vide (bien effacer toute valeur, y compris un "0").
5. Appliquer → OK.



```
Restart-Service 'MSSQL$SQLEXPRESS'
Set-Service -Name SQLBrowser -StartupType Automatic
Start-Service SQLBrowser
```

Vérifier que ça écoute bien :

```
netstat -ano | findstr 1433
# Doit afficher une ligne "LISTENING" sur 0.0.0.0:1433
```

```
Microsoft Windows [version 10.0.26100.32230]
(c) Microsoft Corporation. Tous droits réservés.

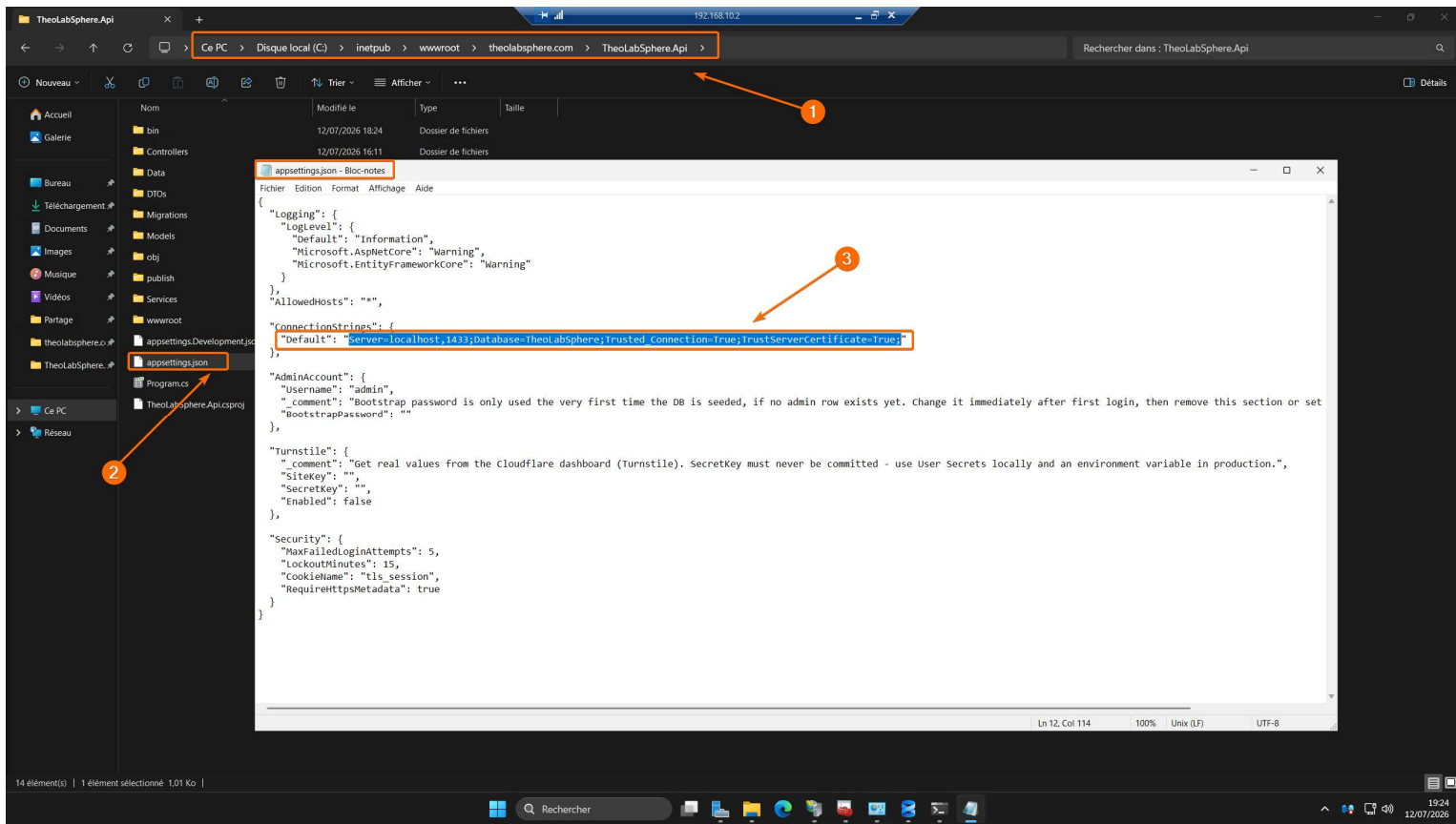
C:\Users\Administrateur>netstat -ano | findstr 1433
TCP    0.0.0.0:1433        0.0.0.0:0          LISTENING        10024
TCP    [::]:1433          [::]:0             LISTENING        10024

C:\Users\Administrateur>
```

5. Utiliser une chaine de connexion par port fixe

Dans appsettings.json, remplacer le nom d'instance par le port statique configuré ci-dessus — plus simple et plus fiable que de dépendre de SQL Server Browser :

```
"Default": "Server=localhost,1433;Database=TheoLabSphere;Trusted_Connection=True;TrustServerCertificate=True;"
```



6. Forcer l'environnement Production pour la commande EF

appsettings.Development.json contient une chaîne de connexion LocalDB ("Server=(localdb)\\MSSQLLocalDB;...") qui écrase celle d'appsettings.json tant que l'environnement n'est pas explicitement Production.

```
$env:ASPNETCORE_ENVIRONMENT = "Production" # PowerShell – PAS "set X=Y"
```

7. Lancement de la migration

```
cd C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api
dotnet ef migrations add InitialCreate
dotnet ef database update
```

Un simple **"Done."** en sortie confirme le succès : la base TheoLabSphere et toutes ses tables sont créées.

8. Republier et resynchroniser IIS

Le dossier publish\ existant date d'avant les corrections (InvariantGlobalization, appsettings.json) — il faut le régénérer, puis remettre à jour les variables d'environnement du pool IIS avec la même chaîne de connexion par port :

```
dotnet publish -c Release -r win-x64 --self-contained false -o .\publish

Clear-WebConfiguration -pspath "MACHINE/WEBROOT/APPHOST" -filter
"system.applicationHost/applicationPools/add[@name='TheoLabSphereApiPool']/environmentVariables"

$pool = "TheoLabSphereApiPool"
$vars = @(
    @{name="ConnectionStrings__Default";
value="Server=localhost,1433;Database=TheoLabSphere;Trusted_Connection=True;TrustServerCertificate=True;"},
    @{name="Turnstile__Enabled"; value="false"},
    @{name="AdminAccount__BootstrapPassword"; value="ChangeMe123!"}
)
foreach ($v in $vars) {
    Add-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST" `
        -filter "system.applicationHost/applicationPools/add[@name='$pool']/environmentVariables" `
        -name "." -value $v
}
Restart-WebAppPool -Name $pool
```

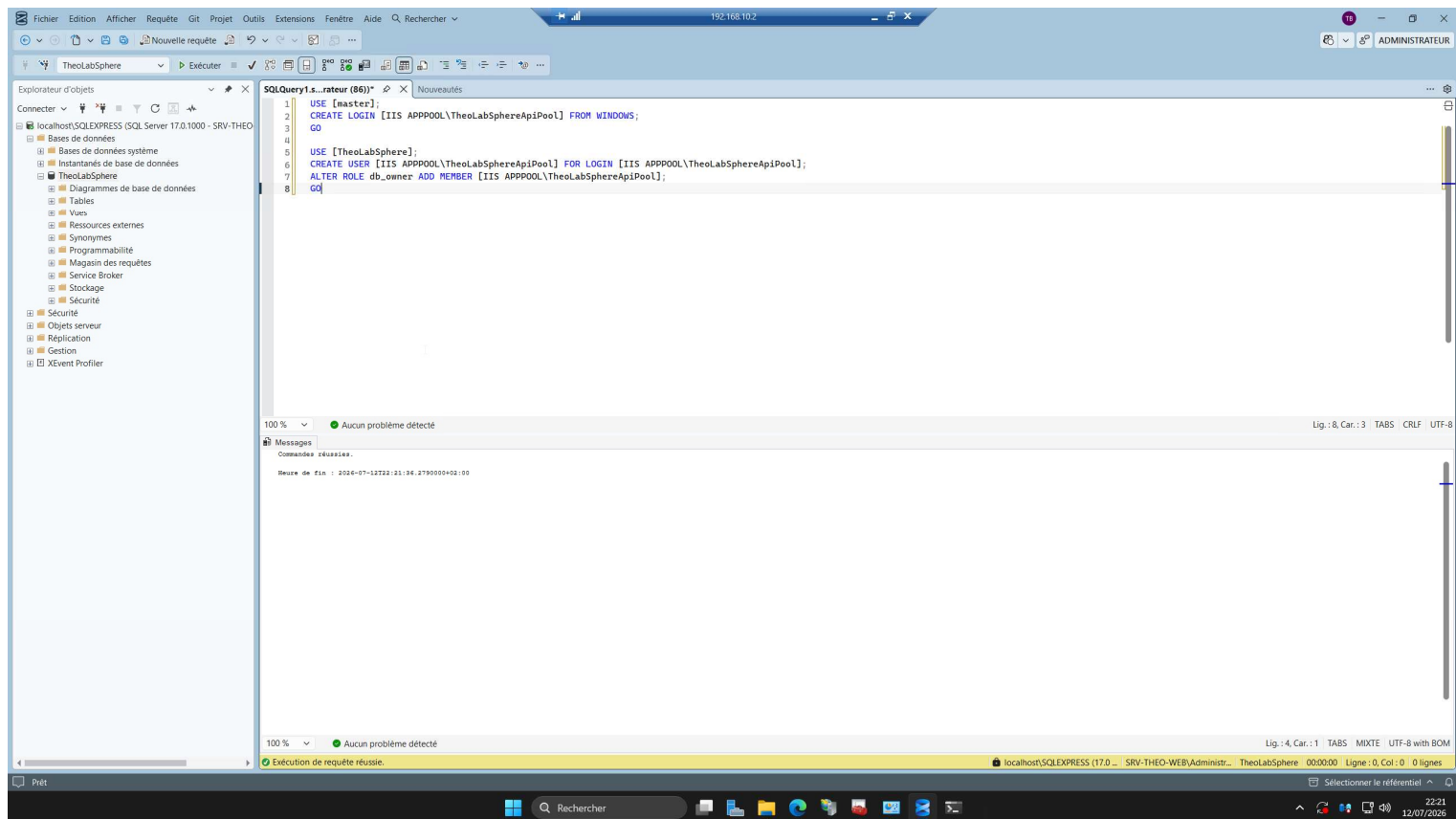
À retenir — Si je décide de reconstruire ce serveur ou en monter un nouveau depuis zéro : je dois appliquer les points 2, 4 et 5 dès l'installation de SQL Server, avant même de publier l'app une première fois — ça évite tout l'aller-retour de dépannage ci-dessus.

Etape 14 : Premier Lancement et vérification

1. Donner à l'identité du pool IIS les droits sur la base SQL

```
USE [master];
CREATE LOGIN [IIS APPPOOL\TheoLabSphereApiPool] FROM WINDOWS;
GO

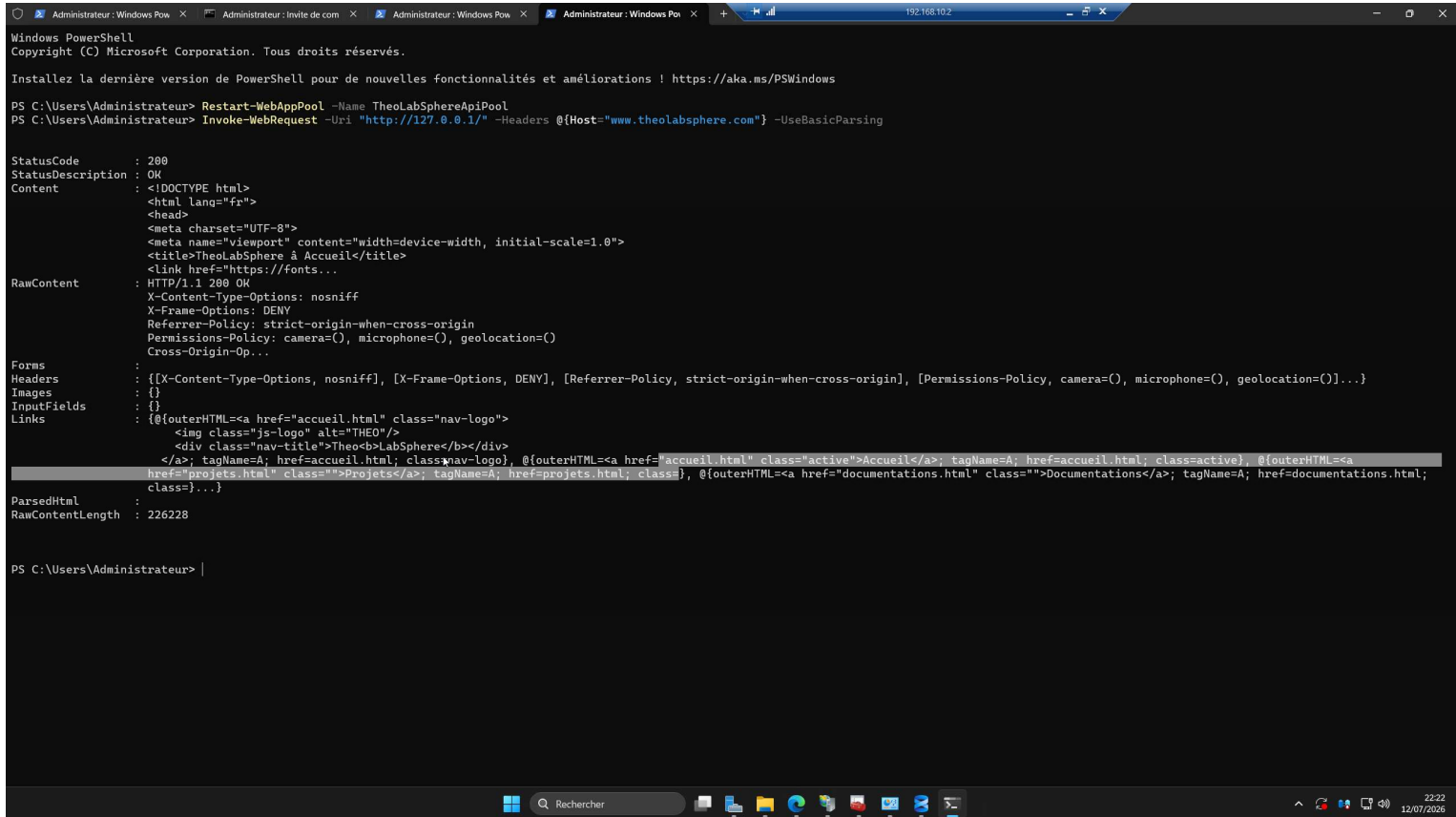
USE [TheoLabSphere];
CREATE USER [IIS APPPOOL\TheoLabSphereApiPool] FOR LOGIN [IIS
APPPOOL\TheoLabSphereApiPool];
ALTER ROLE db_owner ADD MEMBER [IIS APPPOOL\TheoLabSphereApiPool];
GO
```



2. Revalider le backend

```
Restart-WebAppPool -Name TheoLabSphereApiPool
Invoke-WebRequest -Uri "http://127.0.0.1/" -Headers @{Host="www.theolabsphere.com"}
-UseBasicParsing
Invoke-WebRequest -Uri "http://127.0.0.1/api/projects" -Headers
@{Host="www.theolabsphere.com"} -UseBasicParsing
```

Les deux doivent renvoyer *StatusCode 200* (le premier avec le *HTML d'accueil.html*, le second avec du *JSON*, ex. []). C'est seulement à ce stade que le serveur est confirmé fonctionnel de bout en bout.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\Administrateur> Restart-WebAppPool -Name TheoLabSphereApiPool
PS C:\Users\Administrateur> Invoke-WebRequest -Uri "http://127.0.0.1/" -Headers @{Host="www.theolabsphere.com"} -UseBasicParsing

StatusCode      : 200
StatusDescription : OK
Content          : <!DOCTYPE html>
                  <html lang="fr">
                  <head>
                  <meta charset="UTF-8">
                  <meta name="viewport" content="width=device-width, initial-scale=1.0">
                  <title>TheoLabSphere à Accueil</title>
                  <link href="https://fonts...
RawContent       : HTTP/1.1 200 OK
                  X-Content-Type-Options: nosniff
                  X-Frame-Options: DENY
                  Referrer-Policy: strict-origin-when-cross-origin
                  Permissions-Policy: camera=(), microphone=(), geolocation=()
                  Cross-Origin-Op...
Forms            :
Headers          : [[X-Content-Type-Options, nosniff], [X-Frame-Options, DENY], [Referrer-Policy, strict-origin-when-cross-origin], [Permissions-Policy, camera=(), microphone=(), geolocation=())...]
Images           : {}
InputFields      : {}
Links            : {{outerHTML=<a href="accueil.html" class="nav-logo">
                  <img class="js-logo" alt="THEO"/>
                  <div class="nav-title">TheoLabSphere</div>
                  </a> ; tagName=A; href=accueil.html; class=nav-logo}, @{{outerHTML=<a href="accueil.html" class="active">Accueil</a> ; tagName=A; href=accueil.html; class=active}, @{{outerHTML=<a href="projets.html" class="">Projets</a> ; tagName=A; href=projets.html; class=}, @{{outerHTML=<a href="documentations.html" class="">Documentations</a> ; tagName=A; href=documentations.html; class=}}...}}
ParsedHtml       :
RawContentLength : 226228

PS C:\Users\Administrateur> |
```

3. Tester depuis un navigateur sur le serveur en lui-même

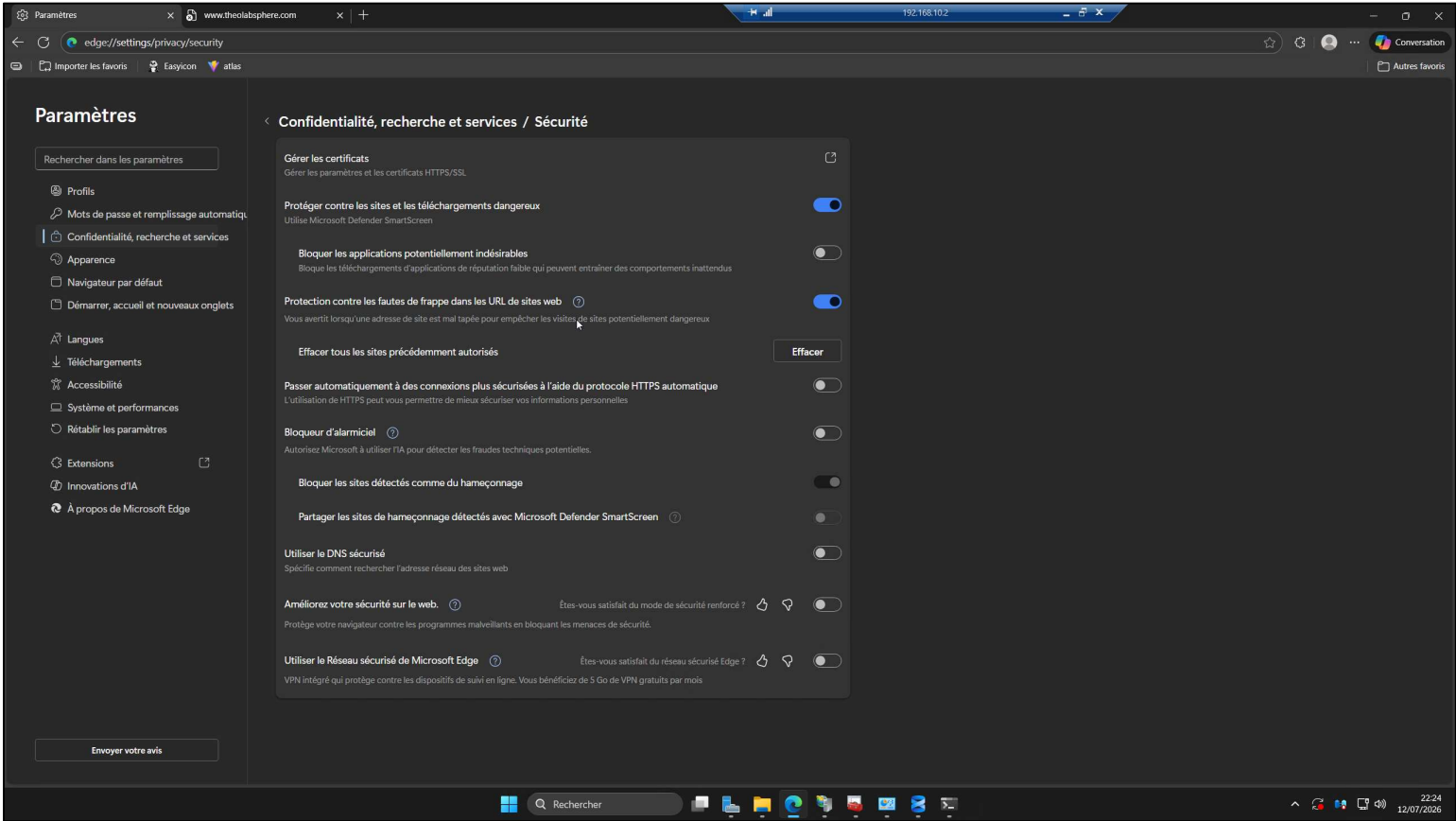
Ajouter l'entrée dans le fichier hosts :

Vérifier que l'entrée est bien présente (ne pas supposer que la commande a réussi) :

Vider le cache DNS :

```
ipconfig /flushdns
```

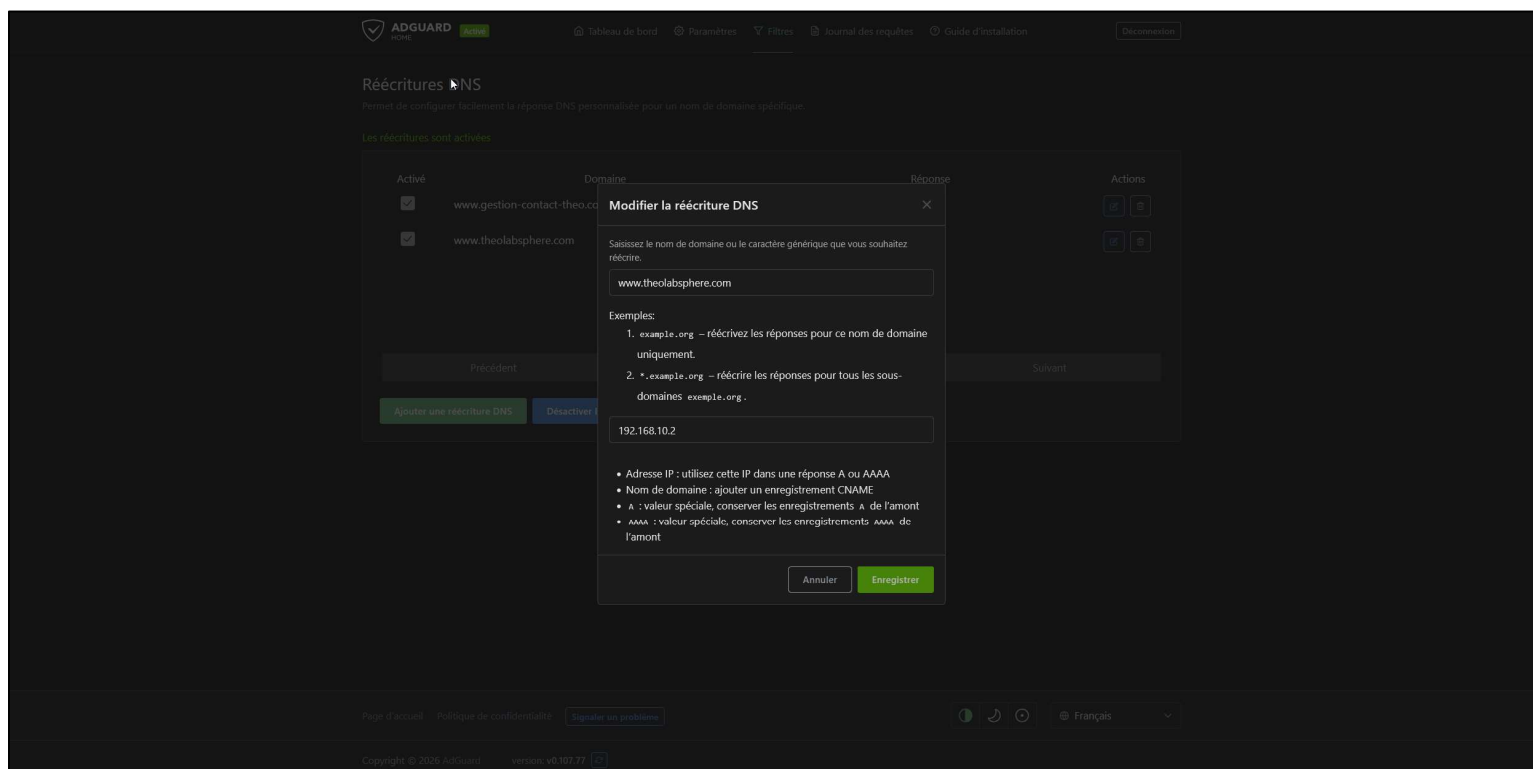
Dans Edge, vérifier que "DNS sécurisé" est désactivé : Paramètres → Confidentialité, recherche et services → Sécurité.



Recharger avec Ctrl+Maj+R (contourne le cache navigateur) sur <http://www.theolabsphere.com/>.



Autre test que j'ai choisis de faire, comme j'ai un serveur DNS Aguard-Home j'ai fais une réécriture DNS sur celui-ci pour y accéder depuis mon pc personnel.



ADGUARDHOME

Tableau de bordParamètresFiltresJournal des requêtesGuide d'installationDéconnexion

Journal des requêtes

www.theolabsphere.com

Toutes les requêtes

Temps	Requête	Réponse	Client
22:30:00 12/07/2026	www.theolabsphere.com Type: A, DNS brut	Réécrit 0.02 ms	192.168.1.39 pc-theo1.home
22:30:00 12/07/2026	www.theolabsphere.com Type: A, DNS brut	Réécrit 0.02 ms	192.168.1.39 pc-theo1.home
22:29:50 12/07/2026	www.theolabsphere.com Type: A, DNS brut	Réécrit 0.02 ms	192.168.1.39 pc-theo1.home
22:29:48 12/07/2026	www.theolabsphere.com Type: A, DNS brut	Réécrit 0.02 ms	192.168.1.39 pc-theo1.home

Page d'accueil

Politique de confidentialité

Signaler un problème

Français

Copyright © 2026 AdGuard version: v0.107.77

Etape 15 : Liés le site Web à Cloudflare tunnel

Arrêt du pool applicatif IIS :

```
Stop-WebAppPool -Name "TheoLabSphereApiPool"
```

Copie des fichiers modifiés dans
C:\inetpub\wwwroot\theolabsphere.com\TheoLabSphere.Api.
dotnet publish -c Release -o .\publish

Exécution du script add-passkeys-table.sql dans SSMS pour créer la table FidoCredentials.

Vérification de la structure de la table via INFORMATION_SCHEMA.COLUMNS : SignCount bien en bigint (point critique pour éviter un plantage EF Core), autres colonnes conformes.

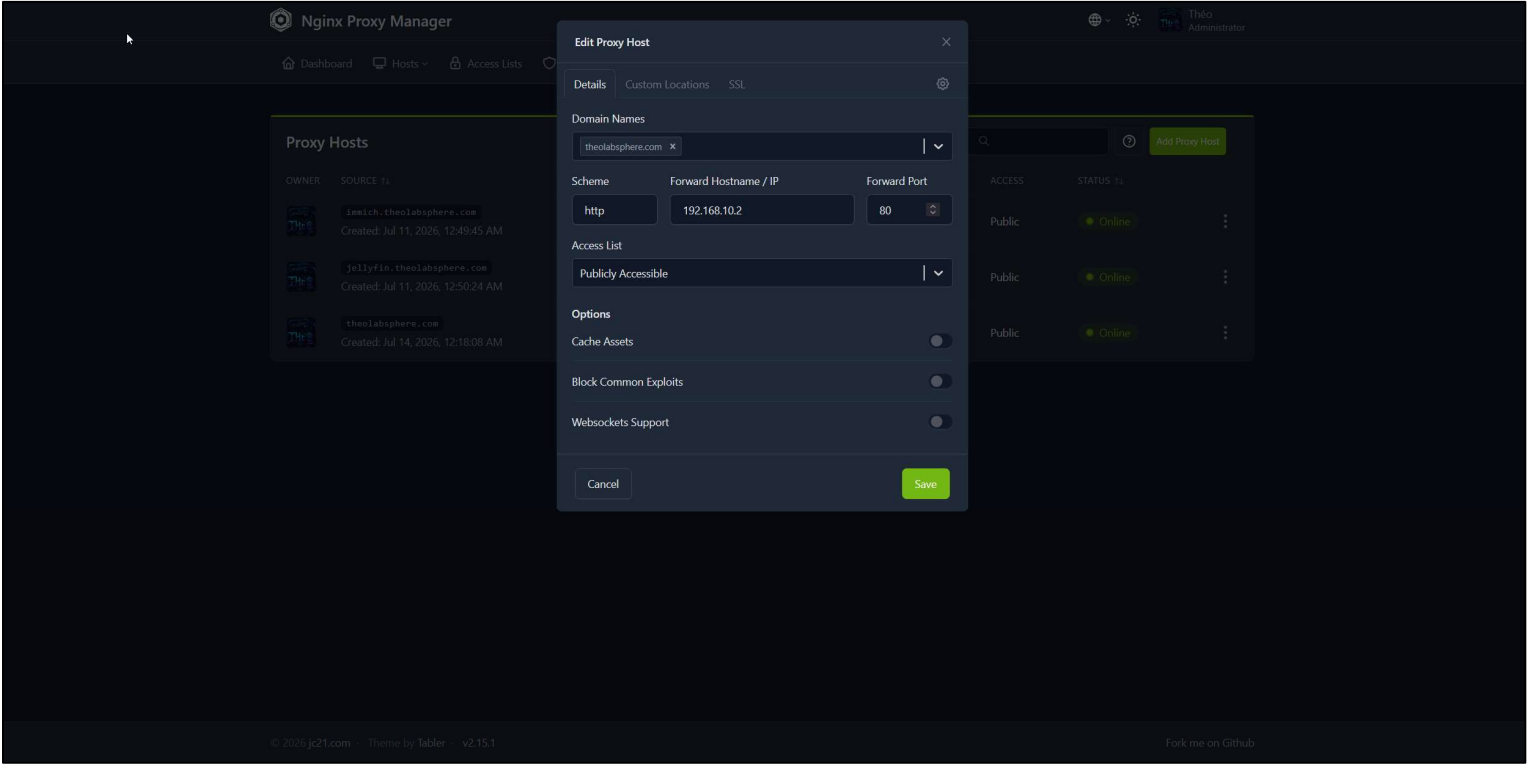
Redémarrage du pool :

```
Stop-WebAppPool
```

NGINX Proxy Manager :

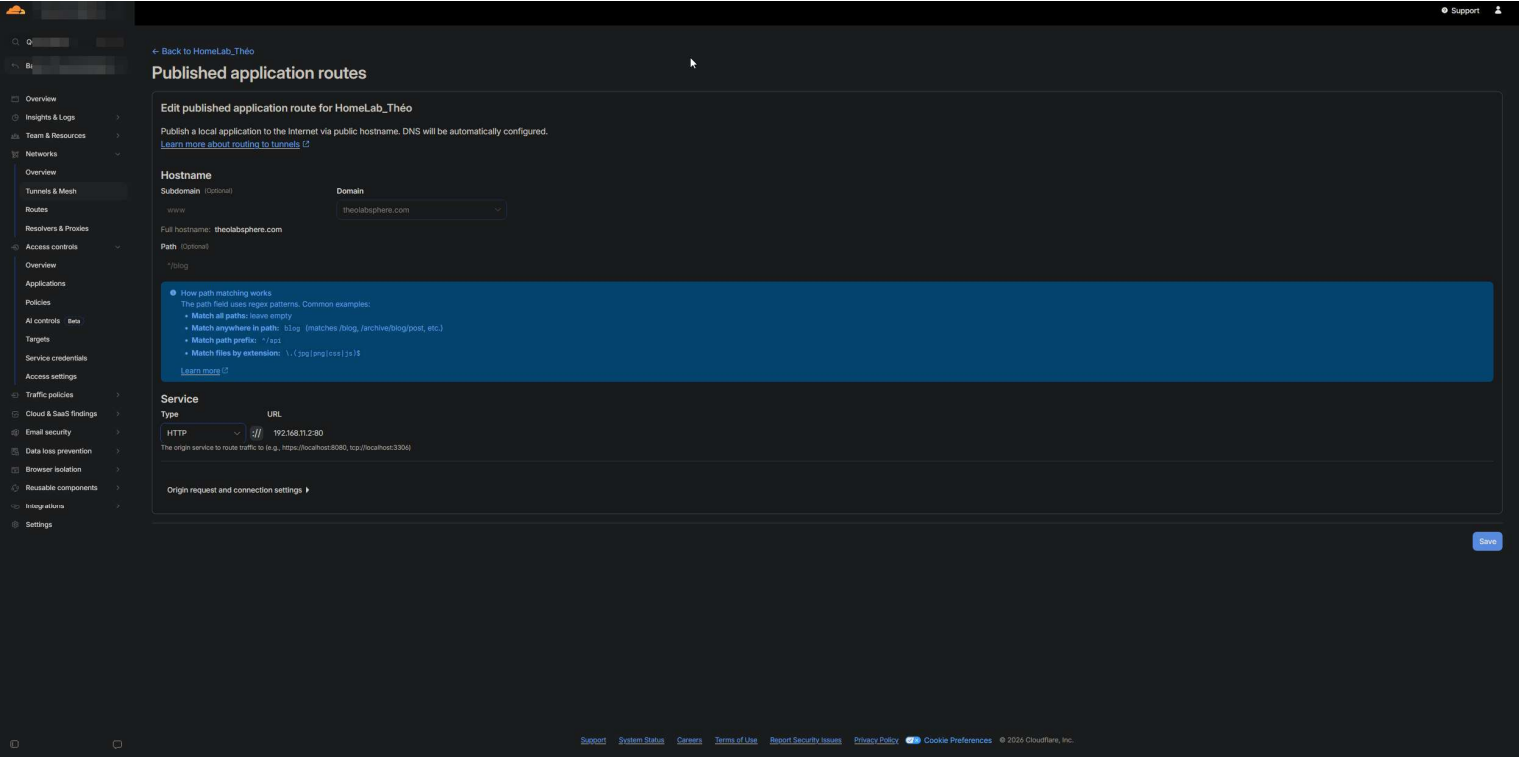
Affectation du serveur Web dans NPM

Domaine theolabsphere.com, Forward Hostname/IP 192.168.10.2 (IIS), port 80, schéma HTTP.



Cloudflare Tunnel :

Une route a été ajoutée via le tableau de bord Zero Trust (Published application route), pointant en HTTP vers 192.168.11.2:80.

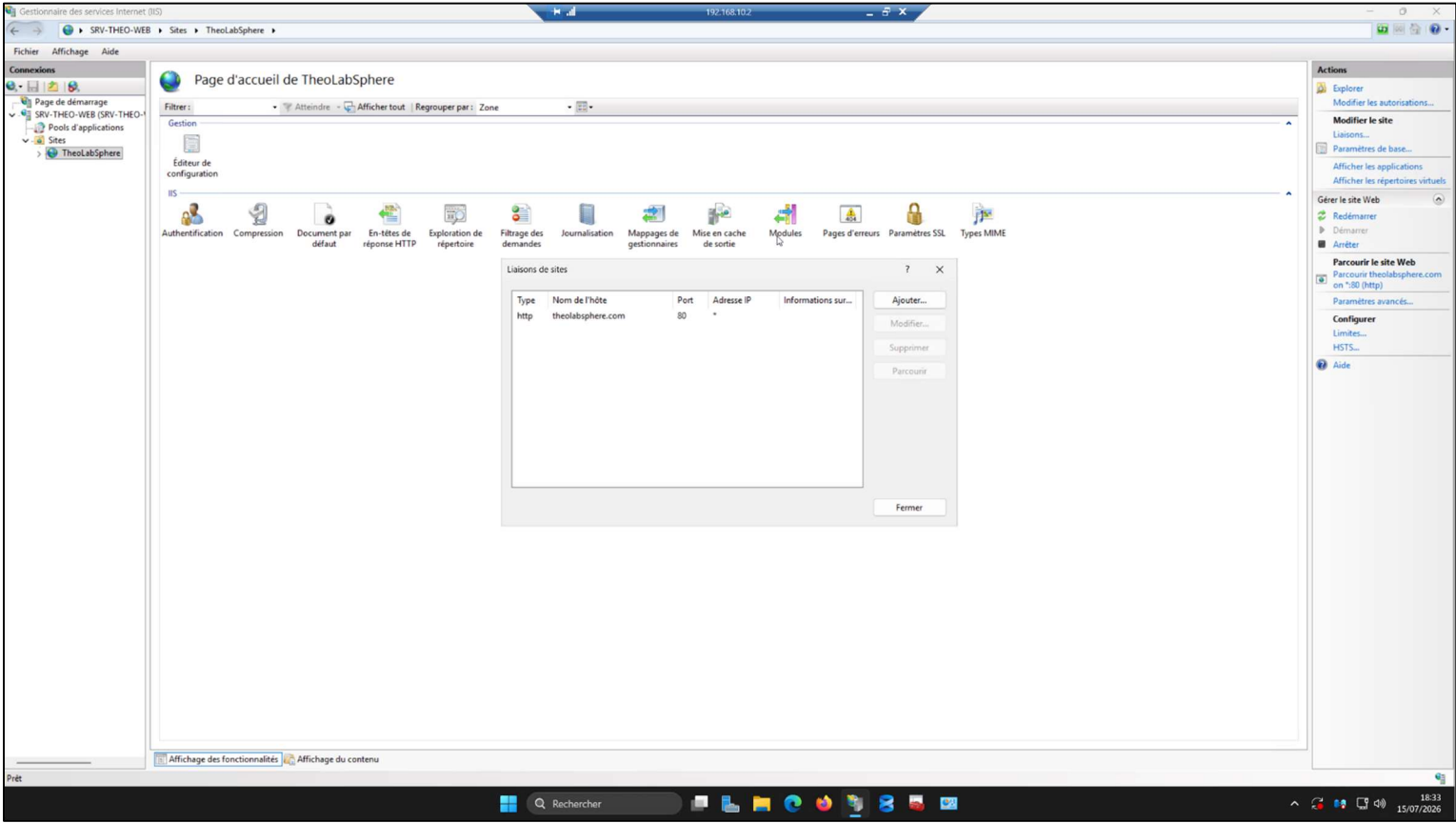


Pare feu Fortinet :

Ajout des règles aller et retour pour établir la communication entre la DMZ et le VLAN10.

IIS – liaisons de site :

Vérification de la liaison HTTP sur le site Web



Cahier des charges

Immich (photothèque)

- Accès refusé à tout le monde par défaut.
- Accès autorisé uniquement via les liens de partage (partage public de galerie/photo).
- Accès autorisé uniquement pour les appareils disposant de l'agent Cloudflare WARP et enrôlés dans l'organisation.

Jellyfin (serveur multimédia)

- Accès refusé à tout le monde par défaut.
- Accès autorisé uniquement pour les appareils disposant de l'agent Cloudflare WARP et enrôlés dans l'organisation.

Exigence de discrétion

Un visiteur non autorisé qui tombe sur ces sous-domaines ne doit pas comprendre facilement qu'un système de connexion protège une application précise (ne pas afficher le nom de l'app, ni un vrai formulaire de connexion menant quelque part).

Architecture présente :

Chaîne complète traversée par une requête vers immich.theolabsphere.com ou jellyfin.theolabsphere.com :

Visiteur → Cloudflare (Access + Tunnel) → Nginx Proxy Manager → IIS / conteneur → Application

Pour des raisons de confidentialité les règle de sécurité présente dans Cloudflare ne seront pas montré dans cette documentation.