COMPTE RENDU

Haute disponibilité
GSB

Par Alexandre BEROT-ARMAND
BTS SIO 2
Lycée Louis PERGAUD

TABLE DES MATIERES

INTRODUCTION

Dans cette mission, nous allons augmenter la haute disponibilité de l'infrastructure de GSB. Cela permet d'augmenter la capacité des services à travailler sans les interruptions causées par des pannes.

Pour cela, nous allons mettre en place différentes actions :

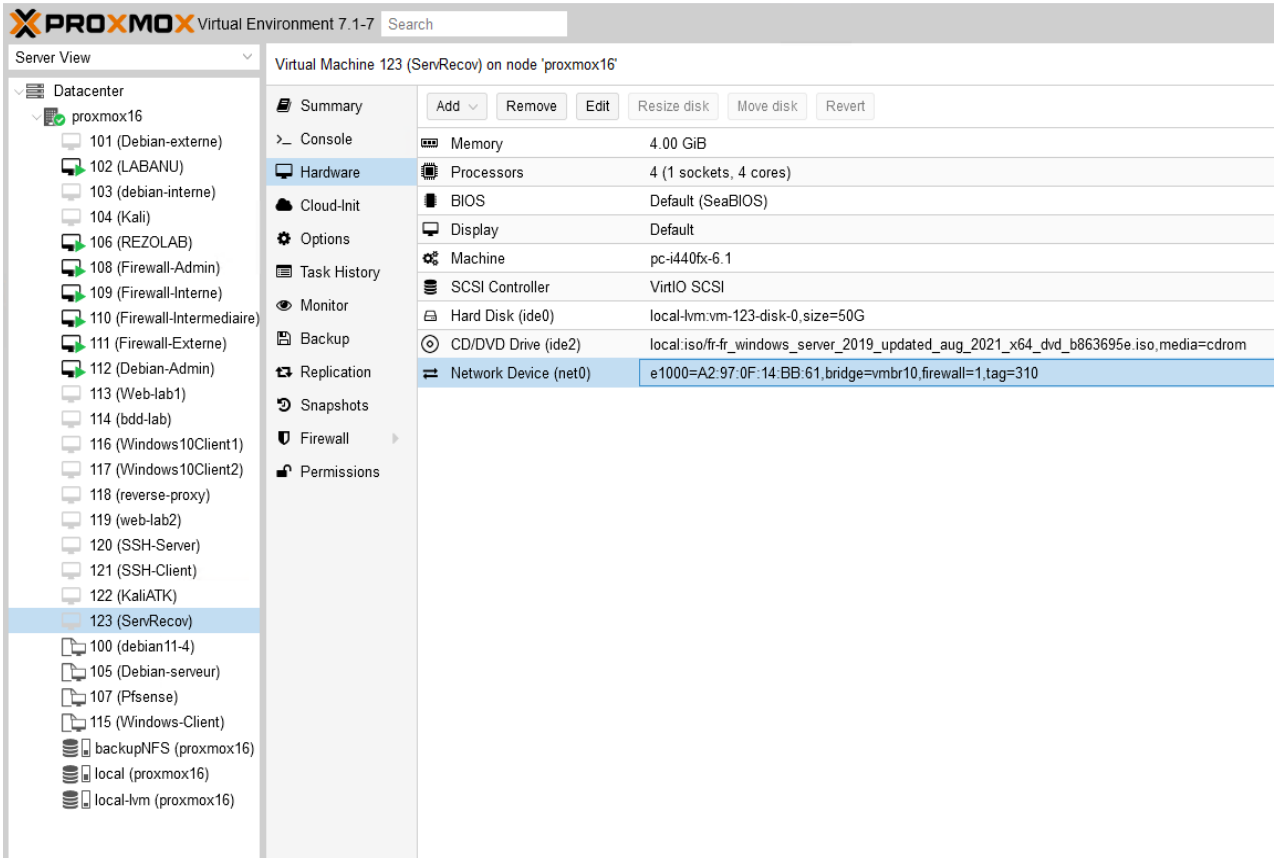
- déploiement d'un serveur de secours pour répartir la charge du serveur DHCP et du serveur AD/DNS et avoir les services concernés disponibles en cas de panne,
- mise en place d'un VRRP pour permettre la redondance de routeur,
- mise en place d'un deuxième lien trunk physique entre le routeur HPE et le switch CISCO en cas de rupture d'un des deux liens,
- déploiement d'un deuxième serveur Web dans la zone des services exposés pour permettre au site de GSB d'avoir un uptime plus élevé (ainsi qu'une répartition de charge de travail),
- création d'un script PowerShell pour créer un backup des dossiers utilisateurs et équipes présents sur le serveur AD/DNS.

Pour des raisons de sécurité et de façon tout à fait optionnel, j'ai décidé d'ajouter un système de surveillance basique que j'ai développé en PowerShell.

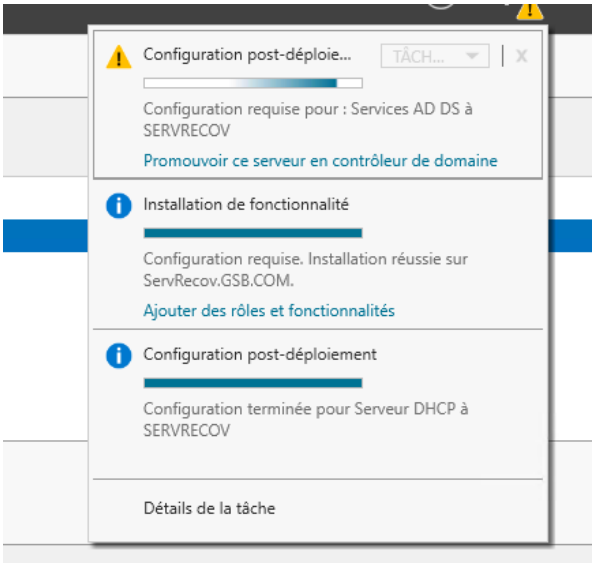
DEPLOIEMENT D'UN SERVEUR DE SECOURS

La première action à faire pour augmenter la haute disponibilité est de déployer un deuxième serveur AD/DNS et DHCP pour permettre la redondance des services et ainsi augmenter la disponibilité des services si le serveur DHCP (REZOLAB) ou AD/DNS (LABANU) venait à tomber en panne.

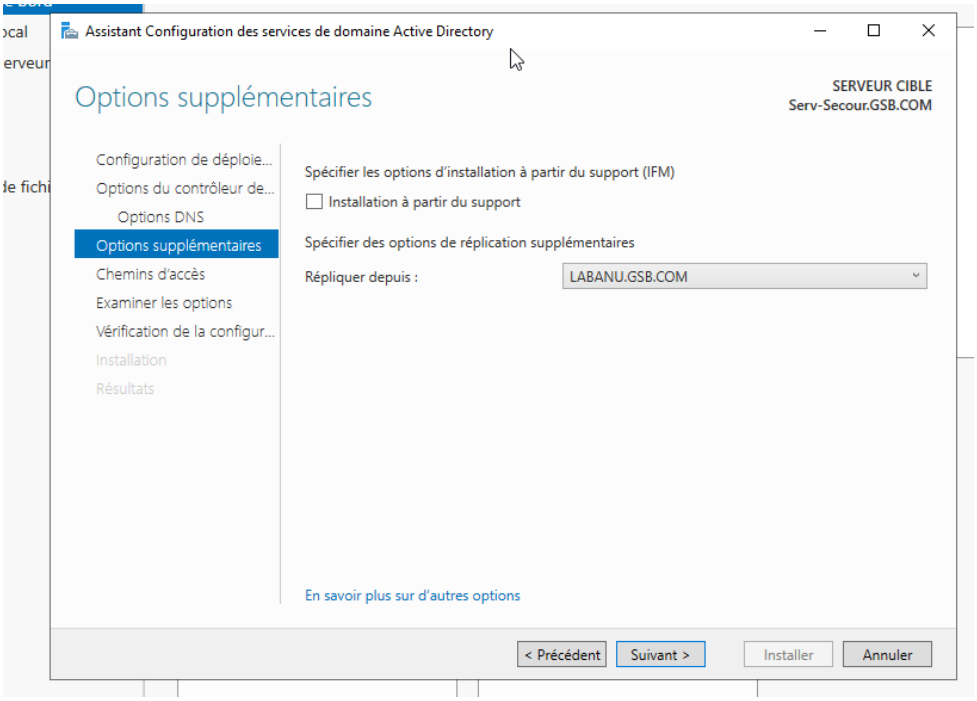
Nous allons donc créer une nouvelle VM (ServRecov) en Windows server 2019.



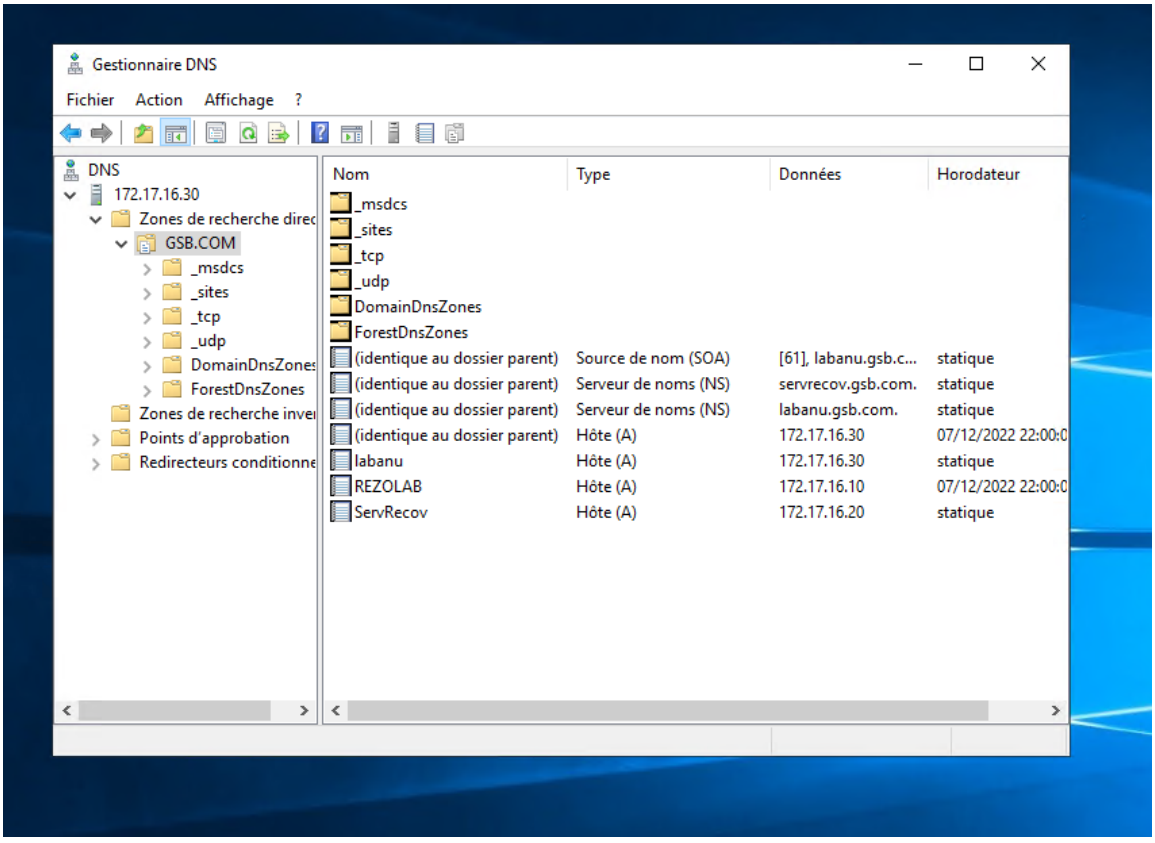
Puis nous ajoutons les services AD, DNS et DHCP dessus :



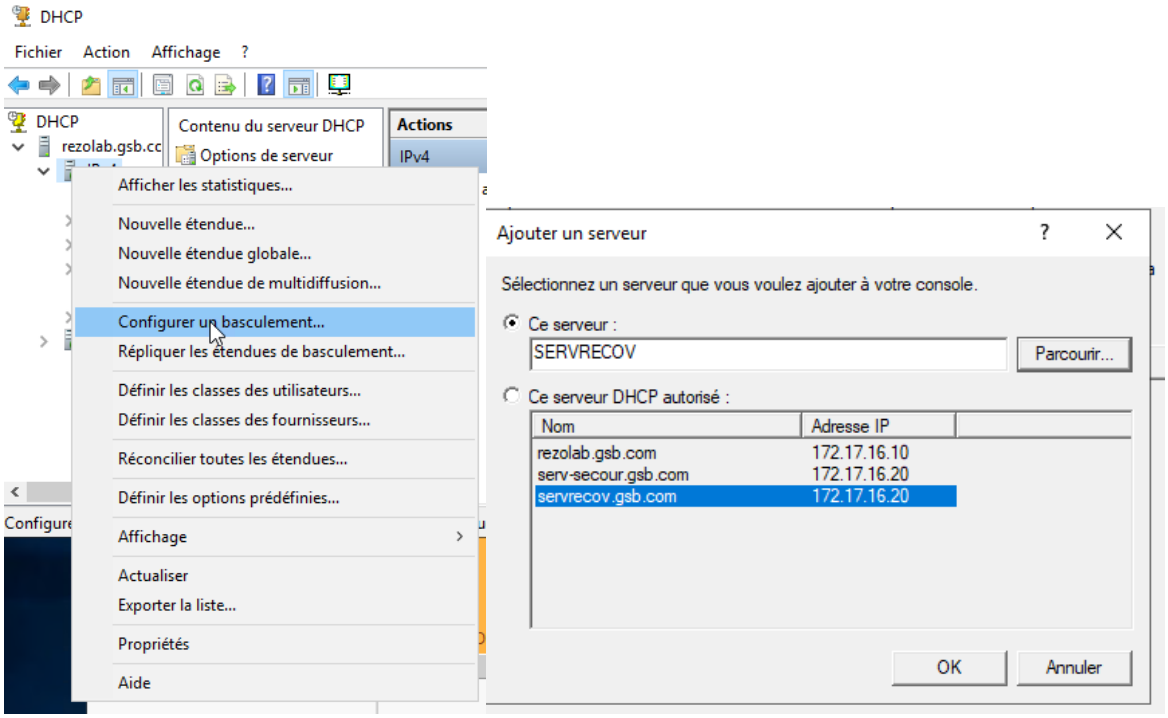
Pour l'AD, nous ferons simplement une réplication depuis le serveur AD principal LABANU, cela permettra d'avoir une disponibilité malgré une panne mais sans possibilité de modification de l'AD :

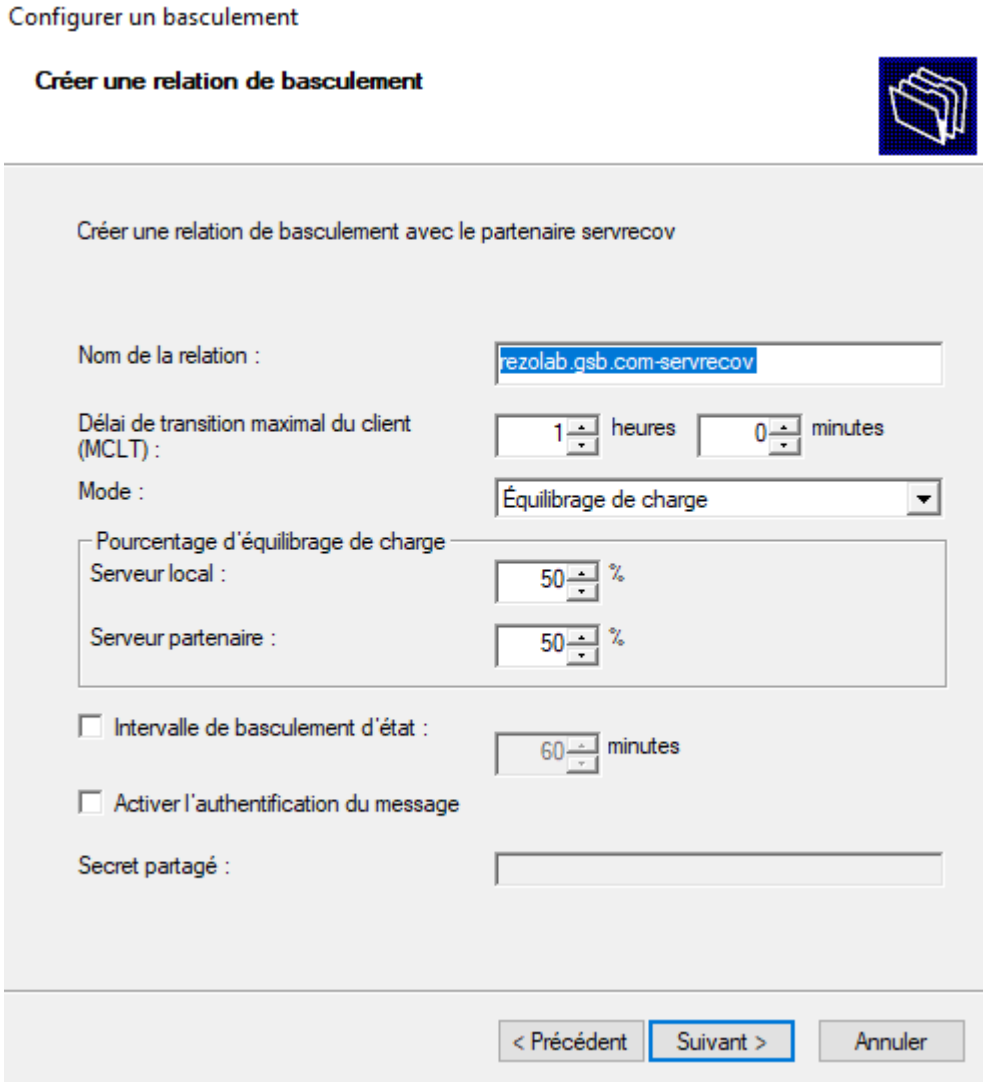


Pour le DNS, il suffit de bien le mettre en serveur secondaire. Il ne faut pas le placer en serveur primaire, sinon cela peut donner des résultats aléatoires qui varient entre conflit entre les deux serveurs et coexistence aux effets hasardeux.

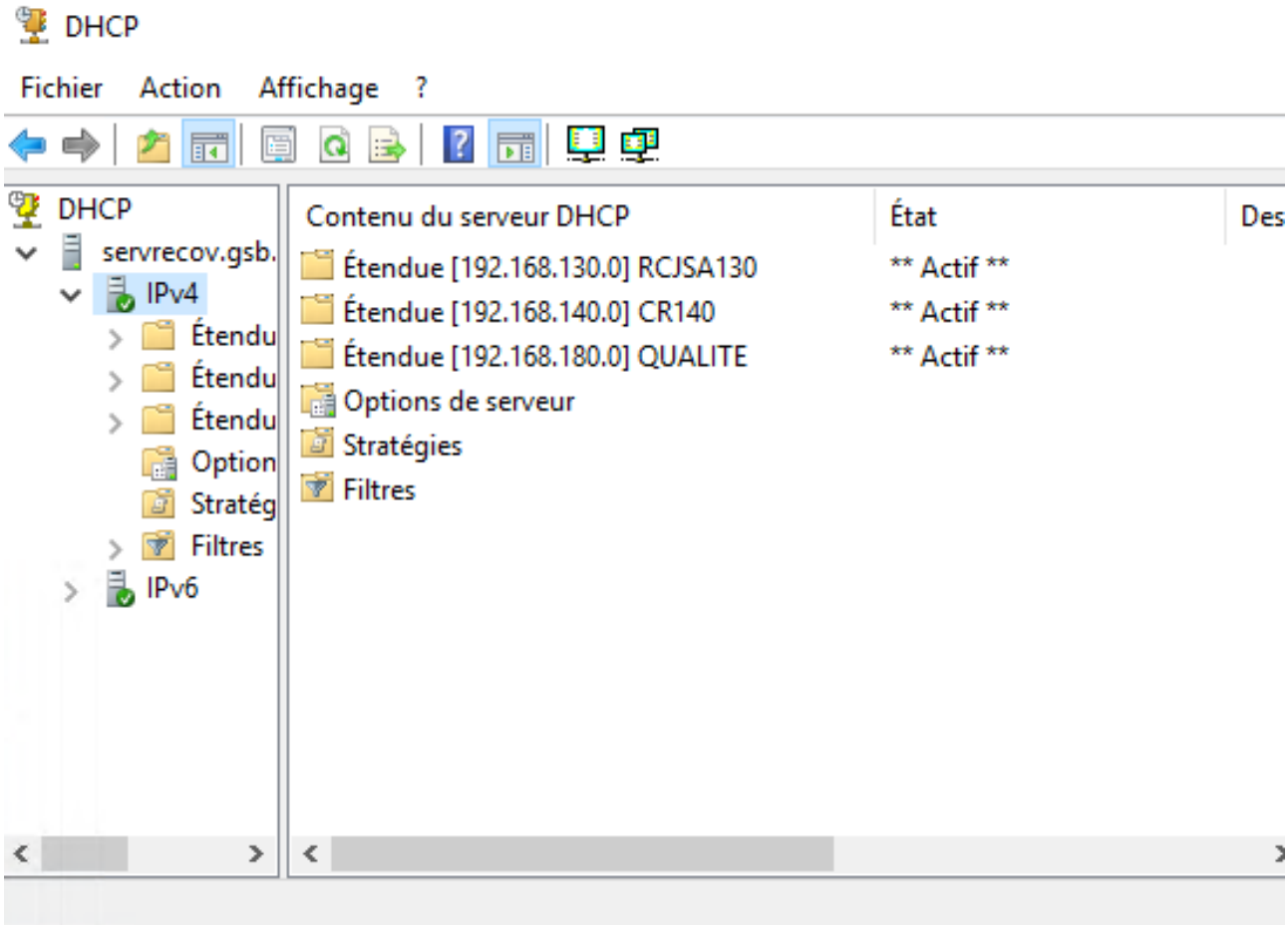


Pour le DHCP, il suffit de configurer un basculement DHCP :





Ainsi le service DHCP aura un partage de charge de 50/50 entre le serveur DHCP principal Rezolab et le serveur de secours ServRecov pour les mêmes plages d'adresses :



MISE EN PLACE D'UN VRRP

Pour augmenter la disponibilité, il est possible de mettre en place un VRRP (Virtual Router Redundancy Protocole).
Le principe est simple : pour sortir sur internet, le flux doit forcément passer par une interface de routeur. Or s'il tombe, le réseau ne peut plus avoir accès à internet jusqu'à son remplacement (ou sa réparation). Pour contrer ça, on peut attribuer une adresse IP Virtuelle en tant que sortie mais reliée à un groupe de routeur.
Cela permet d'utiliser un groupe de routeur au lieu d'une seule machine.
Il est à noter que le protocole VRRP permet aussi le Load Balancing pour améliorer la fluidité en plus de la disponibilité.

Dans la mission 2, il n'était pas possible de mettre en place le VRRP à cause des VLANs mis en place dans la mission 1. Si le protocole gère les VLANs, nous avions des répétitions dans les VLANs entre les groupes d'étudiants, il était donc impossible de savoir où envoyer un paquet entre 2 VLANs qui portent le même identifiant mais qui concernent des machines différentes puisque nous disposions d'un routeur pour deux étudiants et de deux routeurs par baie.

MISE EN PLACE D'UN SECOND LIEN TRUNK

La mise en place d'un second lien trunk entre le Switch et le routeur permet d'augmenter aussi la disponibilité des services.
En effet, si l'un des liens venait à tomber (panne, sabotage, accident…), le deuxième serait présent pour assurer le trafic.
Dans la configuration actuelle, il est impossible de mettre en place un second lien. En effet, les adresses IP des passerelles étant liées à la sous-interface et non au VLAN, il est impossible de dupliquer le lien sans devoir changer la passerelle.
Il n'est pas non plus possible de créer un VRRP au sein du même routeur pour bypasser le problème des sous interfaces.

MISE EN PLACE D'UN SECOND SERVEUR WEB

Dans une activité transverse, nous avons mis en place un second serveur web pour l'hébergement des sites web des entreprises.
Cette mise en place a été réalisée dans le but d'avoir une copie du serveur web d'origine au cas où ce dernier tomberait (attaque ou panne).
Comme les serveurs sont cachés derrière un Reverse Proxy, il y a un Load Balancing configuré uniquement en cas de non-réponse du serveur d'origine, mais il est tout à fait possible de changer simplement pour faire un 50/50 sur la charge.

```
SSLEngine on
SSLCertificateFile /etc/apache2/tls/extra/extra.gsb.fr.crt
SSLCertificateKeyFile /etc/apache2/tls/extra/extra.gsb.fr.key

<Proxy balancer://pool1>
        BalancerMember https://web-lab1.gsb.fr:443 loadfactor=1
        BalancerMember https://web-lab2.gsb.fr:444 loadfactor=1
        ProxySet lbmethod=bytraffic
        ProxySet stickysession=COOKIE_LB
</Proxy>
SSLProxyEngine on
```

Ainsi, le reverse proxy alternera entre les deux serveurs pour que le trafic soit plus fluide et plus stable (dans tous les cas, il n'utilisera que les serveurs up ; donc en cas de panne, il n'en utilisera qu'un).

CREATION D'UN SCRIPT POWERSHELL DE BACKUP DES DOSSIERS

Avec la création et la mise en place du serveur de secours ServRecov, nous avons dupliqué l'AD, le DNS et le DHCP.
Le problème est qu'en cas de panne grave du serveur AD/DNS (LABANU), les dossiers des équipes et des utilisateurs seraient inutilisables et perdus.
La possibilité d'avoir une sorte de VRRP pour l'accès aux ressources étant compliquée, il est tout à fait possible de faire un script PowerShell pour copier (à minima) les dossiers utilisateurs et équipes. Cela permettrait en quelques lignes de récupérer les données sur les scripts de mise en place des utilisateurs et de rendre manuellement l'accès à tous les utilisateurs.

```
1   $src1 = "\\LABANU\PERSO\"
2   $src2 = "\\LABANU\EQUIPES\"
3
4   $dst = "C:\BACKUPS"
5
6   while ($true) {
7       $time = (Get-Date).ToString("yyyyMMdd-HHmm")
8       $backup_path = "$dst\$time"
9       New-Item -ItemType Directory -Path $backup_path
10
11      Robocopy "$src1" "$backup_path\PERSO" /E /MIR
12      Robocopy "$src2" "$backup_path\EQUIPES" /E /MIR
13
14      $delete_before = (Get-Date).AddDays(-7)
15      Get-ChildItem $dst | Where-Object {$_.LastWriteTime -lt $delete_before} | Remove-Item
16
17      Start-Sleep -Seconds 3600
18  }
```

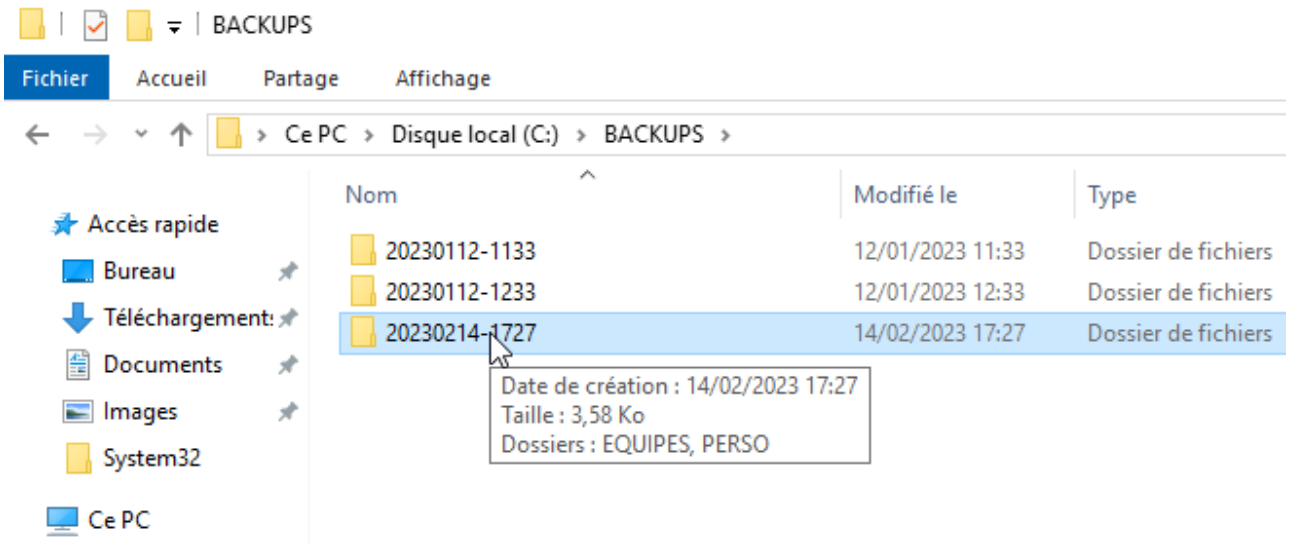
Le programme est simple :
On crée 3 variables : deux pour les sources des dossiers et une pour la destination.

Ensuite on démarre une boucle où l'on crée un dossier sur le serveur de backup ServRecov en fonction de la date puis on démarre la commande Robocopy qui permet de faire des backups.
Dans un souci de clarté, j'ai préféré faire une ligne de Robocopy par dossier.

La fin du script sert à supprimer les fichiers de backup où rien n'a été changé depuis 7 jours.

Bien entendu, la dernière ligne permet de faire cette manipulation de sauvegarde toutes les heures afin d'éviter de surcharger le serveur de backup, et pour économiser sur l'utilisation du processeur au vu du peu de données à copier actuellement.

Résultat des Robocopy :



On voit bien les copies à différentes dates.

La suppression n'a pas eu lieu à cause de l'écart démusuré entre les deux lancements du script que j'ai oublié de relancer après une sauvegarde.

OPTIONNEL : CREATION D'UN SCRIPT POWERSHELL DE SURVEILLANCE

Dans le but d'avoir un système de surveillance minimale, j'ai créé un script de surveillance en PowerShell.

Après la version 0.1, j'ai appris que certaines missions à venir seront à propos de la surveillance donc je n'ai pas poussé le concept plus loin. Mais voici l'ébauche d'un système simple de surveillance :

```
#Chargement de Windows Form
[void][System.Reflection.Assembly]::LoadWithPartialName("System.Windows.Forms")
[void][System.Reflection.Assembly]::LoadWithPartialName("System.Drawing")

#Creation de la forme principale
$form = New-Object Windows.Forms.form

# Titre + no resize + taille
$form.Text = "Système de surveillance du réseau"
$form.FormBorderStyle = [System.Windows.Forms.FormBorderStyle]::FixedDialog
$form.Size = New-Object System.Drawing.Size(600,600)
```

Les deux premiers groupes permettent de charger Windows Form dans PowerShell et de définir la variable $form dans Windows Form.

Le dernier groupe crée la fenêtre du système de surveillance.

Ensuite on crée un ensemble de paramètres pour chacun des serveurs, je présenterai uniquement LABANU vu que c'est une répétition (seuls les emplacements des objets, les adresses et les noms changent) :

```
#### LABANU

#Texte
$LABANU = New-Object System.Windows.Forms.Label
$LABANU.Text = "LABANU"
$LABANU.Location = New-Object System.Drawing.Point(30,30)
$LABANU.AutoSize = $true
$form.Controls.Add($LABANU)
#Bouton
$CheckButton = New-Object System.Windows.Forms.Button
$CheckButton.Location = New-Object System.Drawing.Size (90,25)
$CheckButton.Size = New-Object System.Drawing.Size (50,20)
$CheckButton.Text = "Teste"
$form.Controls.Add($CheckButton)
##Check du bouton
$CheckButton.Add_click({Test-Connection -ComputerName $LABANU.text -Quiet
$TLABANU.Text = If(Test-Connection -ComputerName $LABANU.text -Count 1){"UP"} Else {"Down"}
})
### Resultat
$TLABANU = New-Object System.Windows.Forms.Label
$TLABANU.Text = if (Test-Connection -ComputerName $LABANU.text -Count 1){"UP"} Else {"Down"}
$TLABANU.Location = New-Object System.Drawing.Point(150,30)
$form.Controls.Add($TLABANU)
```

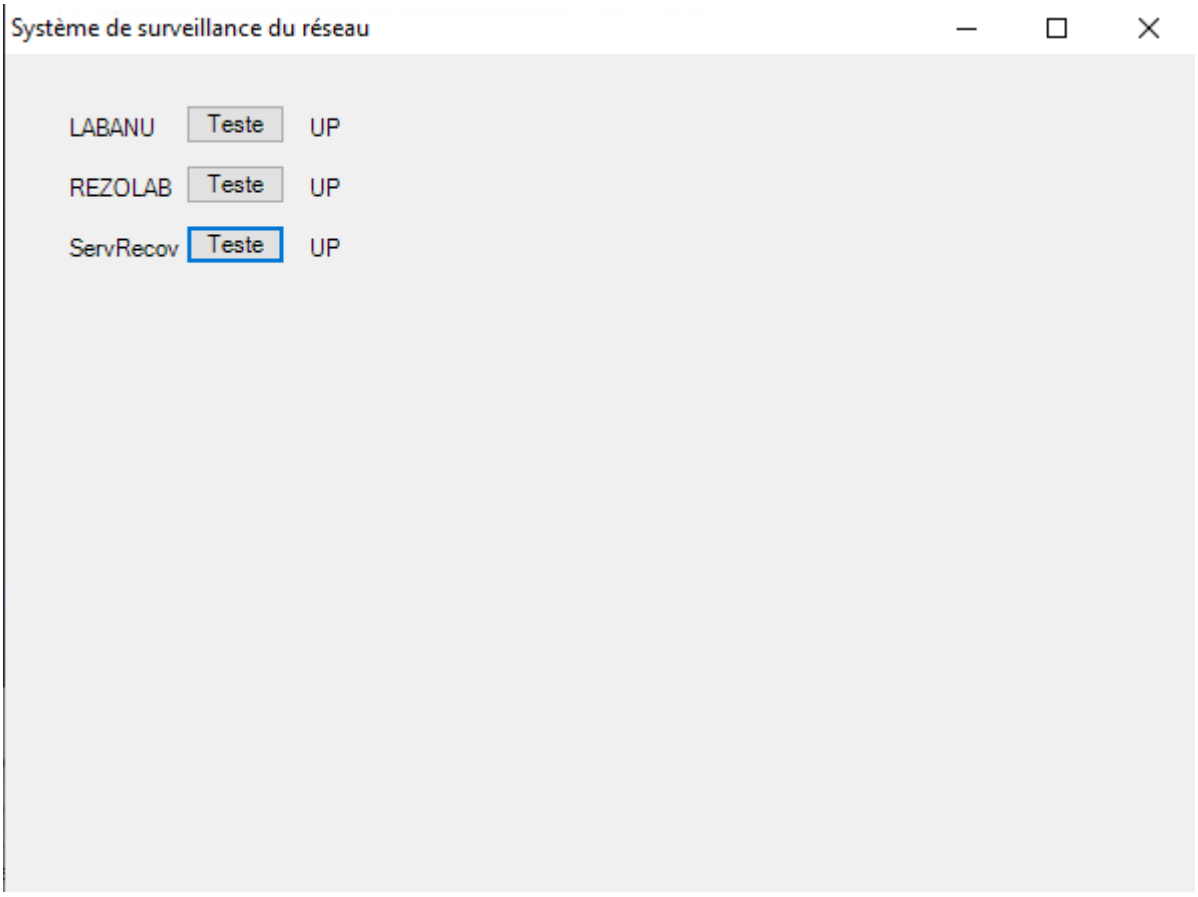
Le groupe #Texte permet de créer une ligne de texte "LABANU " sur la fenêtre principale.

Le groupe #Bouton crée un bouton cliquable "Texte" à côté du nom du serveur.

Le groupe #Check du bouton permet de check le serveur avec un ping.

Le groupe #Resultat affiche UP ou DOWN en fonction du résultat obtenu en cliquant sur le bouton (un test initial est fait).

Résultat final :



C’est basique mais fonctionnel.