Documentation : Mise en place de serveurs de sauvegarde (Backup 01 et Backup 02)

Cloner une VM, réserver des adresses IP sur le serveur DHCP, configurer les interfaces réseau en DHCP, et renommer les serveurs de sauvegarde.

Étapes

1. **Clonage des VMs** Je commence par cloner la VM de base pour créer mes deux serveurs de sauvegarde Backup 01 et Backup 02. Je réalise cette action directement depuis l'hyperviseur.

2. **Réservation des adresses IP sur le serveur DHCP** Ensuite, je vais sur le serveur DHCP et je récupère les adresses MAC des deux serveurs clonés. Je fais une réservation d'adresse IP pour chacun : - **Backup 01** : 10.31.208.73 - **Backup 02** : 10.31.208.74

3. **Configuration des interfaces réseau** Je me connecte sur chacun des serveurs clonés et je modifie le fichier suivant pour que les serveurs utilisent DHCP. J'édite le fichier `/etc/network/interfaces` comme suit :

```
auto eth0
iface eth0 inet dhcp
```

Ensuite, je redémarre le service réseau avec la commande suivante :

```
sudo systemctl restart networking
```

4. **Renommage des serveurs** Une fois les adresses IP correctement attribuées, je renomme les serveurs avec la commande suivante :

Pour **Backup 01** :

```
sudo hostnamectl set-hostname backup01
```

Pour **Backup 02** :

```
sudo hostnamectl set-hostname backup02
```

Enfin, je redémarre chaque serveur pour appliquer les modifications :

```
sudo reboot
```

Après avoir redémarré les serveurs Backup 01 et Backup 02, je passe à la mise à jour des serveurs de

sauvegarde en faisant :

```
apt update && apt upgrade
```

Puis j'exécute la commande suivante pour installer BackupPC :

```
apt install backuppc
```

Comme BackupPC utilise un serveur web pour l'interface de gestion, il faut s'assurer qu'Apache2 est installé sur le serveur. Dans notre cas, c'est déjà fait, sinon faire :

```
apt install apache2
```

Ensuite, je redémarre le service Apache2 avec :

```
systemctl restart apache2
```

5. **Configuration des modes de connexion pour la sauvegarde** Choisir un mode de connexion : BackupPC supporte plusieurs protocoles comme SSH, rsync, SMB. Pour un serveur Linux, SSH/rsync est recommandé.

Configurer l'accès SSH : Il faut générer une clé SSH pour BackupPC sur le serveur :

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa
# Lorsque le système demande une phrase de passe pour la clé, il faut juste appuyer sur Entrée pour ne pas se connecter sans phrase de passe aux machines dont on voudra faire des sauvegardes.
```

Cette commande va créer deux fichiers : -

```
~/.ssh/id_rsa
```

: la clé privée (à garder secrète). -

```
~/.ssh/id_rsa.pub
```

: la clé publique à distribuer aux machines cibles.

6. **Déploiement de la clé publique SSH sur les machines cibles (avec notre script)** Notre script va déployer la clé publique `id_rsa.pub` du compte `backuppc` sur les machines cibles, de

manière à ce que BackupPC puisse s'y connecter via SSH sans mot de passe.

```
import os
import re

######
###    Deploy
###  - the content of a copy id_rsa.pub from backuppc server (user backuppc)
###  - to the authorized_keys file of a list of virtual machines (user
backup)
###  - Target directory : /var/backups/.ssh/authorized_keys
######

# Chemin vers le répertoire distant et le fichier d'autorisation
remote_dir = "/var/backups/.ssh/"
remote_file = "authorized_keys"
remote_path = remote_dir + remote_file

# Fonction pour valider les adresses IPv4
def is_valid_ipv4(ip):
pattern = r'^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.' \
          r'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.' \
          r'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.' \
          r'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$'
# Vérifie si l'adresse IP correspond au motif
return re.match(pattern, ip) is not None

# Ouverture du fichier contenant les adresses IP cibles
with open("./target.ip", "r") as inputfile:
for line in inputfile:
ip = line.rstrip()  # Suppression des espaces blancs de fin
# Validation de l'adresse IP
if not is_valid_ipv4(ip):
print(f"\n Skipping {ip} (- Unvalid IPv4 Address -) ")
continue
# Affichage des informations sur l'adresse IP traitée
print("")
print("*" * 25)
print(f"*** {ip}")  # Correction d'affichage
print("*" * 25)
# Lecture de la clé publique depuis le fichier
with open("./id_rsa.pub", "r") as keyfile:
mykey = keyfile.readlines()[0].rstrip()  # Récupération de la première ligne
(clé publique)

# Création du répertoire .ssh s'il n'existe pas déjà
print("-- Updating backup user shell (sh)...")
print("-- Creating /var/backups/.ssh/ directory...")
print("-- Adding pubkey in authorized_keys file...")
```

```
print("-- Updating authorized_keys permissions...")

# Exécution des commandes sur la machine distante via SSH
os.system(f"ssh root@{ip} \"usermod --shell /bin/sh backup && mkdir -p
{remote_dir} && echo {mykey} >> {remote_path} && chown backup
{remote_path}\"")

# Configuration des permissions sudo pour l'utilisateur backup
sudo_cfg = "backup ALL=NOPASSWD: /usr/bin/rsync"

print("-- Adding sudo configuration for backup account")
print(f"---- {sudo_cfg}")

# Ajout de la configuration sudo sur la machine distante
os.system(f"ssh root@{ip} \"echo {sudo_cfg} > /etc/sudoers.d/backup\"")
print("-- DONE")

print("*" * 25)
```

7. **Création du fichier target.ip** Il faut aussi créer le fichier

```
target.ip
```

. Dans ce fichier, on mettra toutes les adresses IP des machines sur lesquelles nous allons envoyer nos clés publiques :

Adresse IP	Description
———————-	———————-
10.31.216.53	DNS 1
10.31.216.54	DNS 2
10.31.208.67	DHCP PRIMAIRE
10.31.208.68	DHCP SECONDAIRE
10.31.216.67	DHCP RELAY
10.31.208.33	BDD1
10.31.208.34	BDD2

À votre avis, où le script et le fichier target.ip doivent-ils être exécutés ?

Le script doit être exécuté sur notre machine Windows, car c'est la seule à pouvoir se connecter en SSH sur toutes nos machines dans le LAN et la DMZ. Le script et le fichier target.ip doivent impérativement être dans le même répertoire, ainsi que la clé publique du compte BackupPC que nous avons générée.

Normalement, si toutes ces conditions sont réunies, il faut exécuter le script dans son terminal Windows. Pour cela, vous devez avoir le module Python installé. Vous vous attendez peut-être à ce que je vous donne la procédure pour installer le module Python sur Windows, mais non, Windows c'est un peu la merde, donc je vous demande gentiment de bien vouloir vous débrouiller pour ça(#humour).

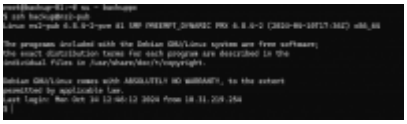
Pour exécuter le script, il faut s'assurer d'être dans le répertoire exact du script et de faire :

```
python script.py
```

Normalement, cela fonctionne et vous donne ça :

Du coup, il va falloir entrer sa phrase de passe pour toutes les adresses IP correspondantes. Oui, c'est chiant : Windows n'a pas de fonction ssh-agent. Donc, après avoir entré les phrases de passe, il faut tester la connexion depuis le compte BackupPC. D'abord, connectez-vous sur BackupPC en faisant :

```
su - backuppc
```

Puis, testez la connexion 

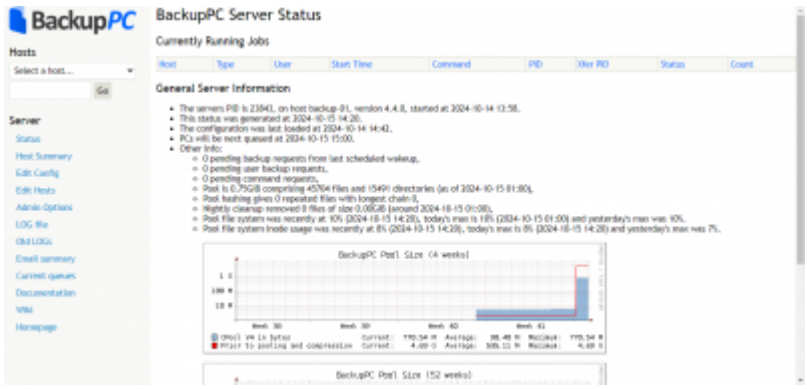
> là on remarque que la connexion fonctionne , on s'est connecté au compte backuppc puis on a testez la connexion ssh vers la machine ns2-pub(les connexions ssh fonctionne pour toutes les adresses ip qui sont de le fichier target.ip.
> Et du coup on peut désormais acceder à l'interface Web de notre compte backuppc les adresses sont :
>
> ```
> 10.31.208.73
> 10.31.208.74
> ```
>
> Voici comment l'interface web du backuppc se présente

> IL faut donc s'identifier pour pouvoir accéder à l'interface du backuppc , cependant avant que l'interface ne fonctionne on avait une message d'erreur en essayat d'acceder à l'interface web du proxmox :
>
> ```
>  FORBIDEN
> ```
>
> cela est due à une ligne non commentée dans le fichier de conf :
>
> ```
> /etc/apache2/conf-available/backuppc.conf
> ```

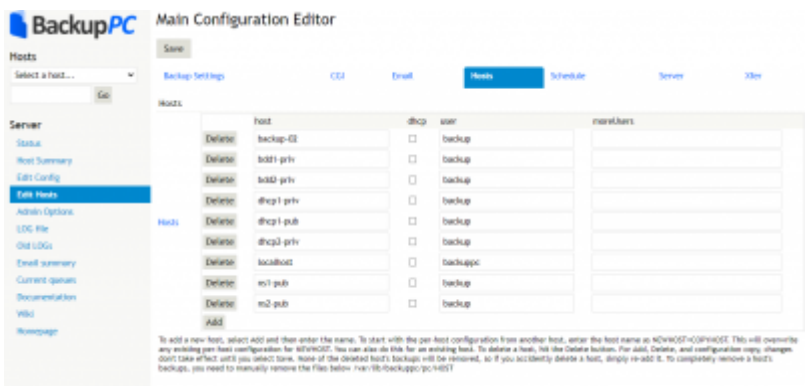
```
Alias /backuppc /usr/share/backuppc/cgi-bin/

<Directory /usr/share/backuppc/cgi-bin/>
        AllowOverride None

        # Uncomment the line below to ensure that nobody can sniff important
        # info from network traffic during editing of the BackupPC config or
        # when browsing/restoring backups.
        # Requires that you have your webserver set up for SSL (https)
access.
        #SSLRequireSSL

        Options ExecCGI FollowSymlinks
        AddHandler cgi-script .cgi
        DirectoryIndex index.cgi
```

```
        AuthUserFile /etc/backuppc/htpasswd
        AuthType basic
        AuthName "BackupPC admin"

        <RequireAll>
                # Comment out this line once you have setup HTTPS and
uncommented SSLRequireSSL
                # Require local #c'est cette ligne qui fallait
impérativement commentée pour ne plus avoir le forbiden
                # This line ensures that only authenticated users may access
your backups
                Require valid-user
        </RequireAll>
</Directory>
```

> Donc après authentification on accède donc à cette interface :



> il faut donc configurer les HOSTS dans les edits hosts :



Host	DHCP	User	MoreUsers
backup-02		backup	
bdd1-priv		backup	
bdd2-priv		backup	
dhcp1-priv		backup	
dhcp1-pub		backup	
dhcp2-priv		backup	
localhost		backuppc	

Host	DHCP	User	MoreUsers
ns1-pub		backup	
ns2-pub		backup	

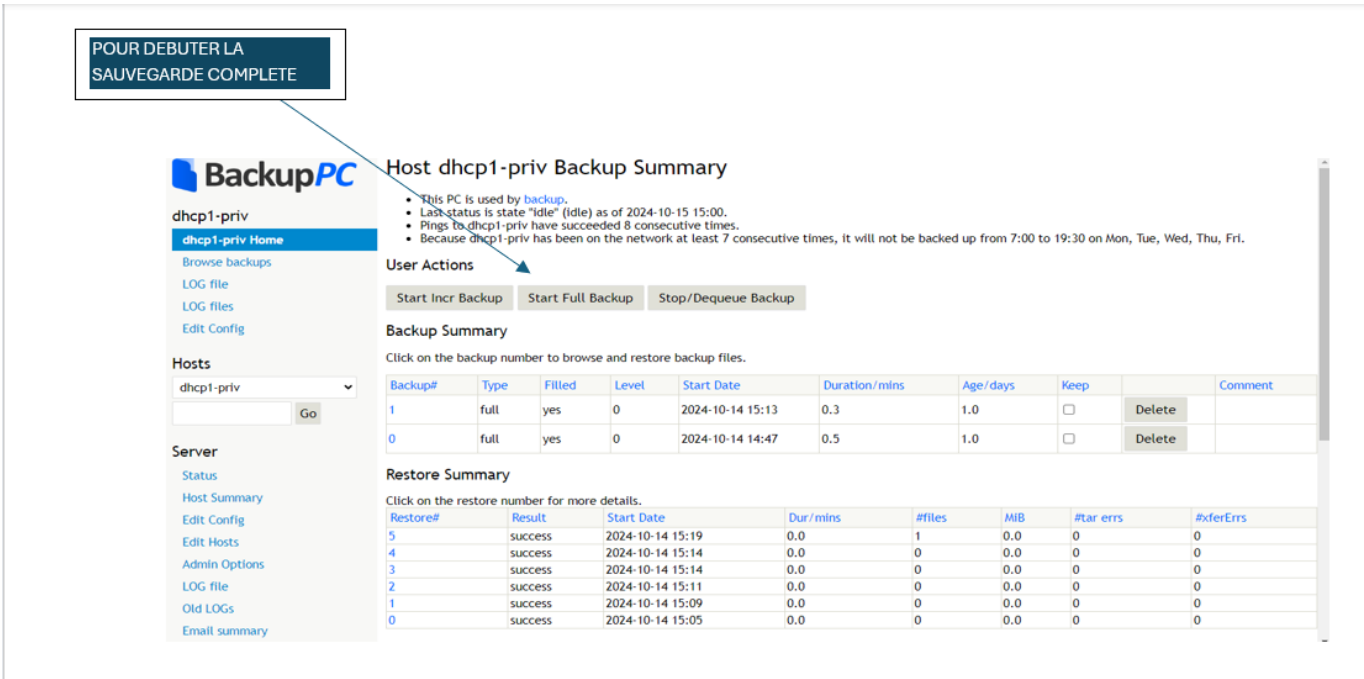
apres avoir paramétrer les hosts il faut aussi configurer les parametres :Xfer Settings

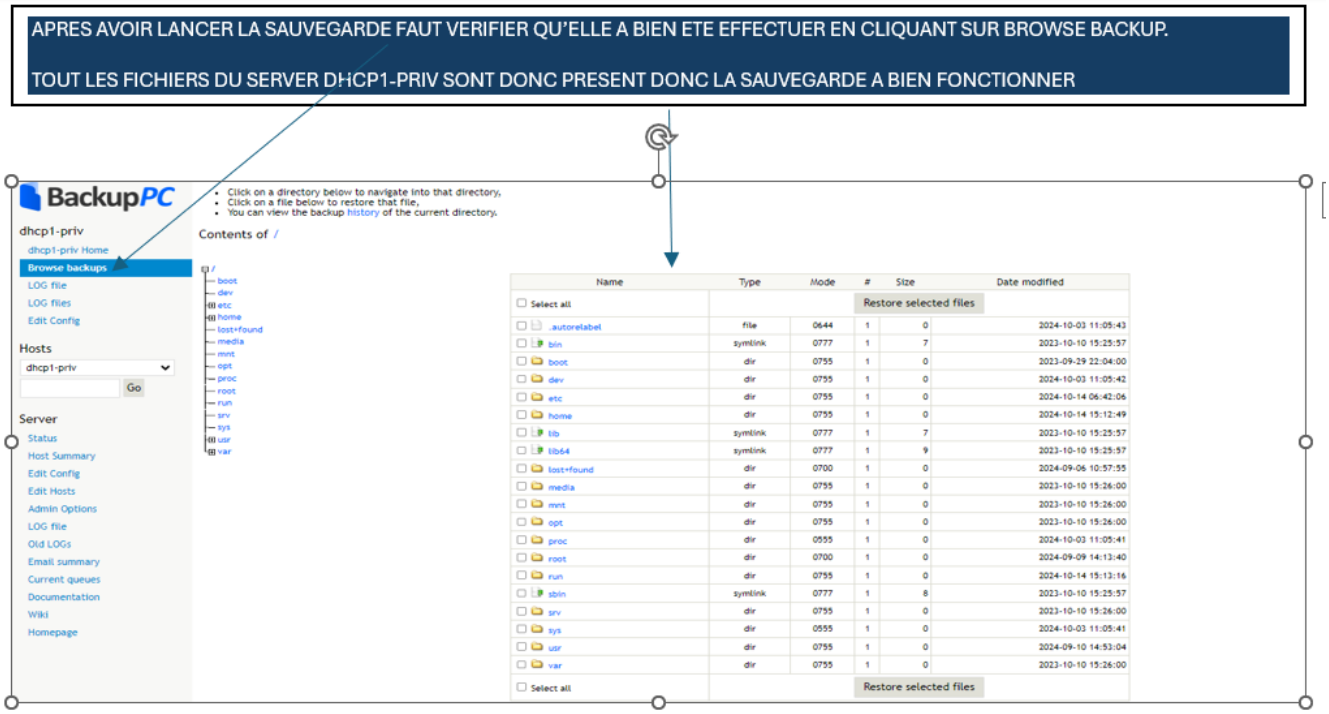
Paramètre	Valeur	Commentaire
XferMethod	rsync	Méthode de transfert choisie, ici c'est `rsync`.
XferLogLevel	1	Niveau de log pour les transferts.
ClientCharsetLegacy	iso-8859-1	Encodage des caractères pour les anciens clients.
ClientShareName2Path	New ShareName or '*'	Définit un nouveau nom de partage ou utilise '*'.
RsyncBackupPCPath	/usr/libexec/backuppc-rsync/rsync_bpc	Chemin pour l'exécutable `rsync_bpc` utilisé par BackupPC.
RsyncClientPath	sudo /usr/bin/rsync	Chemin pour l'exécutable `rsync` du client.
RsyncSshArgs	$sshPath -l backup	Commande `ssh` utilisée pour la connexion sécurisée.

une fois que les configurations des hosts sont terminé on peut désormais tester les sauvegardes

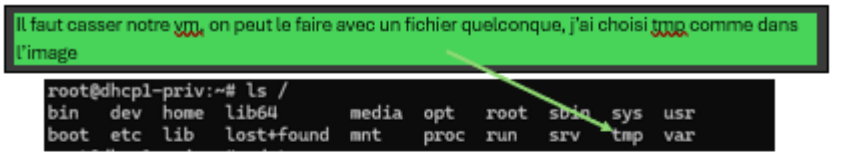
Dans les fichiers d'host on aurait pu mettre les adresses ip , On on a fait des réservations avec des noms , chaque adresse ip correspont à un nom , le fichier de conf du dhcp c'est celui-ci :conf_db.oceanie

TEST DE SAUVEGARDE

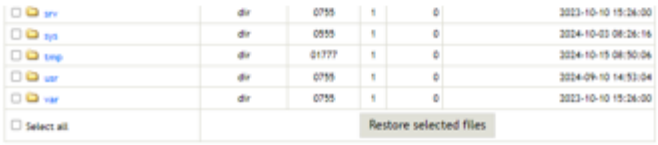




> A PRESENT IL FAUT TESTER LA RESTAURATION (il faut s'assurer de donner les permissions “sudo /usr/bin/rsync ”)



Voici les fichiers sauvegardés du dhcp1-priv, du coup il faut cliquer sur tmp puis appuyer sur RESTORE SELECTED FILES (suivez les instructions puis restorer)



APPUYEZ SUR START RESTORE(et normalement le tour est joué)

MAINTENANT ON VERIFIE SI LA RESTAURATION A ETE EFFECTUER SUR LE SERVEUR

Du coup le fichier tmp a bien été restaurer