
1 - Mise en place serveur NAS

- installation vm terminal debian => 4go RAM + 3 disque 8Go (Raid 5)
- inet static
- partition disk with gdisk for gpt partition (fdisk for mbr)
 - > nettoyer partition table in /dev/sdb : `wipefs -a /dev/sdb`
 - > créer partitions
 - > copier les partitions sur /dev/sdc et /dev/sdd
 - > randomize GUID pour qu'il soit différent pour tous les disques

2 - Raid (5,6 ou 10)

- mdadm
 - > créer un raid 5 avec 3 disk

```
mdadm --create --level=5 --raid-devices=3 /dev/md0 /dev/sdb1 /dev/sdc1 /dev/sdd1
```
 - > vérifier raid status

```
cat /proc/mdstat
```
 - > enregistrer paramètre raid + lancer au démarrage (évite de devoir refaire la config à chaque démarrage)
 - > créer un système de fichier sur /dev/md0

```
mkfs.ext4 /dev/md0
```
 - > monter le disque /dev/md0 sur le répertoire /data

```
mount /dev/md0 /data
```
 - > vérifier que l'opération à marcher avec `df -h`
 - > sauvegarder la config raid pour qu'elle se réassemble automatiquement au démarrage

```
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```
 - > mettre à jour initramfs pour qu'il prenne en compte la nouvelle config

```
update-initramfs -u
```
 - > rajouter les paramètre nécessaire à /etc/fstab pour monter le RAID au démarrage:

```
echo " /dev/md0 /data ext4 rw,nofail,relatime,x-systemd.device-timeout=20s,defaults 0 2" >> /etc/fstab
```

3 - Accès sécurisé via WebDav

- > installation apache2
 - > site web = <http://192.168.64.100> (adresse ip vm)
 - > désactiver page par défaut

```
sudo a2dissite 000-default
```

```
sudo service apache2 reload
```
 - > configuration hôte virtuel

```
cd /etc/apache2/sites-available/  
sudo nano webdav.local.conf  
(<VirtualHost *80>
```

....

```
</VirtualHost>)
```

-> créer un le répertoire /var/www/webdav/ et la page index

```
sudo mkdir /var/www/webdav
```

```
sudo sh -c 'echo "Welcome from WebDAV.local" >  
/var/www/webdav/index.html'
```

```
sudo chown www-data:www-data /var/www/webdav
```

-> activer le site et recharger apache2

```
sudo a2ensite webdav.local
```

```
sudo service apache2 reload
```

-> module webdav

-> activer le module webdav

```
sudo a2enmod dav_fs
```

-> créer un répertoire pour les données WebDAV et un fichier de donnée pour tester

```
sudo mkdir /var/www/webdav/svn
```

```
sudo touch /var/www/webdav/svn/linuxconfig.txt
```

```
sudo chown www-data:www-data /var/www/webdav/svn
```

-> modifier la configuration du <VirtualHost> pour inclure les paramètres WebDAV

```
Alias /svn /var/www/webdav/svn
```

```
<Location /svn>
```

```
    DAV On
```

```
</Location>
```

-> redémarrer apache2 pour appliquer les modifications

```
-> sudo service apache2 restart
```

-> test

<http://192.168.64.100/svn>

-> configurer auth utilisateurs

-> création fichier mdp + ajout utilisateur

```
sudo mkdir /usr/local/apache2/
```

```
sudo htpasswd -c /usr/local/apache2/webdav.passwords linuxconfig
```

-> inclure paramètre authentification dans <VirtualHost>

```
<Location /svn>
```

```
    DAV On
```

```
    AuthType Basic
```

```
    AuthName "webdav"
```

```
    AuthUserFile /usr/local/apache2/webdav.passwords
```

```
    Require valid-user
```

```
</Location>
```

-> redémarrer apache2

```
sudo service apache2 restart
```

-> monter le répertoire WebDAV

-> apt install davfs2

-> sélectionner si oui ou non les utilisateur non privilégié aurons le droit de monter des ressources webDAV (modifiable avec dpkg-reconfigure => j'ai choisi oui

-> créer un point de montage pour le répertoire
sudo mkdir /mnt/webdav
sudo mount.davfs http://192.168.64.100/svn /mnt/webdav/

-> en + : certificat ssl pour une connexion sécurisée en https (requiert un nom de domaine)

4 - SFTP + Partage dossier (Dossier Public)

- sftp
 - > apt install proftpd
 - > apt install openssh-server
 - > activer module ssh ou ssl pour connexion en sftp
- OU
 - > apt install openssh-server
 - > nano /etc/ssh/sshd_config => config sftp (plus compliqué)
- samba
 - > création utilisateur samba
 - > /!\ chown samba: /data
 - > editer /etc/samba/smb.conf
 - > systemctl restart samba
 - > test: Executer (système) sur windows => //192.168.64.100 => connexion avec id samba
 - > configurer l'accès en partage au dossier
 - > ajouter les lignes suivante au fichier /etc/samba/smb.conf

```
create mask = 0660
directory mask = 0770
force group = share
```
 - > changer groupe propriétaire + droits sur le dossier partager
 - > chgrp -R share /data/share
 - > chmod -R g+rw /data/share

5 - Sauvegarde (rsync + 2eme serveur)

- mise en place d'une 2eme vm (serveur backup)
- installer rsync sur les deux machines
 - > apt -y install rsync
 - >
- rsync
 - > rsync -a /data/share/ rsync@192.168.64.101:/home/rsync/nas_backup/
- crontab sur le serveur nas pour envoyer le backup sur le serveur backup
- > configurer le ssh pour ajouter un certificat et ne plus avoir besoin d'entrer le mdp
 - > ssh-keygen -t rsa -b 2048
 - > ssh-copy-id -i /root/.ssh/id_rsa.pub rsync@192.168.64.101
- > configurer le crontab

```
-> crontab -e
-> * * * * * rsync -avzhe ssh /data/share/
rsync@192.168.64.101:/home/rsync/nas_backup/
```

===== plus loins =====

6 - Virtualisation serveur

- installation hyperviseur linux KVM
 - > sudo apt -y install qemu-kvm libvirt-daemon bridge-utils virtinst libvirt-daemon-system
- charger et lancer le module vhost_net
 - > sudo modprobe vhost_net
 - > echo vhost_net | sudo tee -a /etc/modules
- installation outils gestion machine virtuelles
 - > sudo apt -y install vim libguestfs-tools libosinfo-bin qemu-system virt-manager
- création d'une machine virtuelle via l'outils de ligne de commande virt-install
 - > activer le network 'default'
virsh net-start default
 - > virt-install \
--name deb11 \
--ram 2048 \
--vcpus 2 \
--disk path=/var/lib/libvirt/images/deb12-vm.qcow2,size=20 \
--os-variant debian11 \
--graphics none \
--console pty,target_type=serial \
--location 'http://ftp.debian.org/debian/dists/bookworm/main/installer-amd64/' \
--extra-args 'console=ttyS0,115200n8 serial'

7- Interface gestion

- cockpit
 - > apt install cockpit cockpit-storaged cockpit-machines cockpit-podman
 - > systemctl enable cockpit.socket && systemctl start cockpit.socket
- web: <https://192.168.64.100:9090>

===== sources =====

sources:

raid + samba: https://std.rocks/gnulinux_nas.html

samba accès groupe (config):

<https://www.it-connect.fr/serveur-de-fichiers-debian-installer-et-configurer-samba-4/>

Apache2 WebDAV: <https://fr.linux-console.net/?p=11647>

-> certificat ssl:

<https://linuxconfig.org/how-to-get-free-ssl-tls-certificates-with-lets-encrypt-and-certbot>

<https://linuxconfig.org/apache-web-server-ssl-authentication>

backup nas synology vers serveur rsync:

<https://www.pc-fute.com/sauvegarde-du-nas-synology-vers-un-serveur-rsync/>

(-> à adapter pour serveur nas debian vers rsync, pas encore testé)

FTP (via WebDAV): <https://fr.linux-console.net/?p=20689>

rsync + crontab:

<https://www.ubuntumint.com/rsync-files-between-two-linux-servers-automatically/>

<https://linuxize.com/post/how-to-use-rsync-for-local-and-remote-data-transfer-and-synchronization/>

kvm + cockpit:

<https://fr.linux-console.net/?p=21516>