Network Attached Storage

Installation et Configuration de l'OS (Debian)

Installation sans interface graphique de Debian

Mise à jour du système :

```
sudo apt update && sudo apt upgrade -y
```

Mettre à jour la distribution :

```
apt dist-upgrade
```

Configuration de l'administrateur :

```
useradd administrateur --home /home/administrateur --create-home --shell /bin/bash
passwd administrateur
apt install sudo
usermod -G sudo administrateur
```

Installation du serveur ssh pour la connection terminal à distance :

```
sudo apt install openssh-server
```

Configuration ssh pour éviter la connection via l'utilisateur root :

```
sudo nano /etc/ssh/sshd_config
```

Il faut que cette ligne soit présente et décommentée :

```
PermitRootLogin no
```

Configuration du Pare-Feu (FirewallD)

Installation du paquet :

```
sudo apt install -y firewalld
```

Afficher toutes les règles du pare-feu :

```
sudo firewall-cmd --list-all
```

Afficher les services qui peuvent être activés ou désactivés :

```
sudo firewall-cmd --get-services
```

Autoriser un service :

```
sudo firewall-cmd --permanent --add-service="service_name"
```

Interdire un service :

```
sudo firewall-cmd --permanent --remove-service="service_name"
```

Recharger le pare-feu :

```
sudo firewall-cmd --reload
```

Autoriser un service via un port spécifique :

```
sudo firewall-cmd --permanent --add-port=2222/tcp
```

Interdire un service via un port spécifique :

```
sudo firewall-cmd --permanent --remove-port=2222/tcp
```

Recharger le pare-feu :

```
sudo firewall-cmd --reload
```

Configuration d'une interface de gestion (Webadmin)

Installation des paquets :

```
sudo wget http://www.webmin.com/download/deb/webmin-current.deb
sudo dpkg --install webmin-current.deb
sudo apt -f install
```

Pour se connecter à l'interface sur le navigateur :

```
http://{ip_du_serveur}:10000
```

Configuration d'un RAID (RAID5)

Installation de mdadm

```
sudo apt install mdadm
```

Création du RAID 5

Pour identifier les disques qui seront utilisés dans le RAID, la commande lsblk permet de lister les disques connectés à notre machine (exemple /dev/sda)

```
sudo mdadm --create --verbose /dev/md5 --level=5 --raid-devices=3 /dev/sd2 /dev/sd3 /dev/sd4
```

Nous montons le raid 5 pour créer un nouveau système (raid5) défini par md5 à partir des disques /dev/sd2 /dev/sd3 /dev/sd4. Level définit le niveau du raid, ici level=5 pour un raid 5.

Création d'un système de fichiers

Une fois le RAID 5 créé, on configure un système de fichiers sur le périphérique RAID

```
sudo mkfs.ext4 /dev/md5
```

Avec cette commande les fichiers seront configurés en ext4.

Montage du système de fichiers

Créez un répertoire où l'on souhaite monter notre RAID :

```
sudo mkdir /mnt/raid
```

On récupère l'UUID du raid puis on le monte dans /mnt/raid/.

```
sudo blkid /dev/md5
>>> /dev/md5: UUID="2d365442-a61a-4236-bc94-4f886c17ab98" BLOCK_SIZE="4096" TYPE="ext4"
sudo mount /dev/md5 /mnt/raid
```

Configuration automatique du montage

Pour que le système de fichiers RAID soit monté automatiquement au démarrage, nous devons ajouter l'entrée suivante dans le fichier /etc/fstab :

```
UUID={md5_UUID}         /mnt/raid         ext4        defaults    0     2
```

Vérification

Nous pouvons vérifier l'état du RAID à tout moment en utilisant la commande mdadm :

```
sudo mdadm --detail /dev/md5
```

Partage de Fichiers avec SFTP

Liste des commandes utiles en SFTP :

```
bye                                 Quit sftp

cd path                             Change remote directory to 'path'
chgrp [-h] grp path                 Change group of file 'path' to 'grp'
chmod [-h] mode path                Change permissions of file 'path' to 'mode'
chown [-h] own path                 Change owner of file 'path' to 'own'

df [-hi] [path]                     Display statistics for current directory
                                    filesystem containing 'path'

exit                                Quit sftp
get [-afpR] remote [local]          Download file
help                                Display this help text
lcd path                            Change local directory to 'path'
lls [ls-options [path]]             Display local directory listing
lmkdir path                         Create local directory
ln [-s] oldpath newpath             Link remote file (-s for symlink)
lpwd                                Print local working directory
ls [-1afhlnrSt] [path]              Display remote directory listing
lumask umask                        Set local umask to 'umask'
mkdir path                          Create remote directory
progress                            Toggle display of progress meter
put [-afpR] local [remote]          Upload file
pwd                                 Display remote working directory
quit                                Quit sftp

reget [-fpR] remote [local]         Resume download file
rename oldpath newpath              Rename remote file
reput [-fpR] local [remote]         Resume upload file
rm path                             Delete remote file
rmdir path                          Remove remote directory
symlink oldpath newpath             Symlink remote file

version                             Show SFTP version
!command                            Execute 'command' in local shell
!                                   Escape to local shell
```

Création du groupe sftp

```
sudo groupadd sftp
```

On crée le répertoire qui va contenir le dossier commun et le /home de tous nos utilisateurs qui utilisent le SFTP, le propriétaire de ce fichier doit être root et lui seul possède tous les droits, notre groupe sftp peut lire et executer, et les autres rien

```
sudo mkdir /mnt/raid/sftp
sudo mkdir /mnt/raid/sftp/common
sudo chmod -R 750 /mnt/raid/sftp
```

Modification de la fin de notre fichier de configuration ssh /etc/ssh/sshd_config :

```
# indiquer au démon sshd d'utiliser les commandes intégrées
Subsystem sftp internal-sftp

# appliquer les lignes qui suivent à tous les utilisateurs faisant partie du
# groupe sftp
Match Group sftp

# interdire la transmission de l'affichage entre le serveur et le client
X11Forwarding no

# interdire les redirections TCP
AllowTcpForwarding no

# interdire l'utilisation du shell
PermitTTY no

# interdire l'utilisation des commandes SSH pour créer des tunnels chiffrés
PermitTunnel no

# obliger les utilisateurs à n'utiliser QUE les commandes SFTP, l'option -d
# /upload spécifie le répertoire dans lequel l'utilisateur arrive lors de la
# connexion, ceci pour éviter qu'il n'arrive à sa racine, car il n'aura pas les
# droits d'écriture
ForceCommand internal-sftp -d /perso

# indiquer au démon que les utilisateurs doivent croire que la racine est
# /sftp/nomdelutilisateur
ChrootDirectory /mnt/raid/sftp/%u
```

Création de l'utilisateur bob

```
sudo useradd -s /usr/bin/false -d /mnt/raid/sftp/bob -G sftp --create-home bob
```

-s spécifie le shell à utiliser, on ne veut pas qu'il en utilise donc on lui indique un « faux » shell (en fait, c'est plutôt une commande qui ne fait rien et se termine avec un code d'erreur)
-d indique quel répertoire sera celui de l'utilisateur
-G ajoute l'utilisateur au groupe sftp

Mot de passe pour l'utilisateur bob

```
sudo passwd bob
```

Modification du propriétaire et du groupe propriétaire du dossier + modification des droits :

```
sudo chown root:sftp /mnt/raid/sftp/bob
sudo chmod 750 /mnt/raid/sftp/bob
```

Création du répertoire qui sera utilisé par l'utilisateur

```
sudo mkdir /mnt/raid/sftp/bob/perso
sudo chown -R bob:bob /mnt/raid/sftp/bob/perso
sudo chmod -R 750 /mnt/raid/sftp/bob/perso
```

On monte le fichier commun dans qui mène au dossier commun :

```
sudo mkdir /mnt/raid/sftp/bob/common
sudo mount -B /mnt/raid/sftp/common /mnt/raid/sftp/bob/common
```

Partie client linux

```
sudo apt install -y sshfs
```

Script Ajout d'Utilisateur

```
#!/bin/bash

echo "--------------------------------------"
echo "------CREATION DE L'UTILISATEUR-------"
echo "--------------------------------------"
echo " "

sudo useradd -s /usr/bin/false -d /mnt/raid/sftp/$1 -G sftp -p $(openssl passwd -1
$2) --create-home $1

echo "--------------------------------------"
echo "--------CREATION DES DOSSIERS---------"
echo "--------------------------------------"
echo " "

sudo chown root:sftp /mnt/raid/sftp/$1
sudo chmod 750 /mnt/raid/sftp/$1

sudo mkdir /mnt/raid/sftp/$1/perso
sudo chown -R $1:$1 /mnt/raid/sftp/$1/perso
sudo chmod -R 750 /sftp/$1/perso

sudo mkdir /mnt/raid/sftp/$1/common
sudo mount -B /mnt/raid/sftp/common /mnt/raid/sftp/$1/common

sudo rm /mnt/raid/sftp/$1/.bash_logout /mnt/raid/sftp/$1/.bashrc
sudo rm /mnt/raid/sftp/$1/.profile

echo "--------------------------------------"
echo "-----------PROCESS TERMINE------------"
echo "--------------------------------------"
echo " "
```

Partage de fichiers avec NFS

1. Installation de NFS

```
sudo apt install nfs-kernel-server
```

Partage de fichiers avec Samba

1. Installation de Samba

```
sudo apt install samba
```

2. Configuration de Samba

On crée à l'intérieur de notre raid5 les 3 dossiers partagés suivants :

```
          /mnt/raid/partage1      /mnt/raid/partage2      /mnt/raid/perso
```

On crée un lien symbolique qui mène au dossier qui contient tous nos dossier paratgés:

```
ln -s /mnt/raid /srv/shares
```

Ajout des sections de partage suivantes à la fin de /etc/samba/smb.conf :

```
[shares]
comment = Pool of Shares
path = /srv/shares
browseable = yes
read only = no
guest ok = no
valid users = @sambausers
```

3. Création de l'utilisateur Samba

On crée un utilisateur Samba avec le nom d'utilisateur 'sambauser' :

```
sudo smbpasswd -a sambauser
```

Il sera demandé de définir un mot de passe pour cet utilisateur.

4. Configuration des autorisations de partage

Il est nécessaire que les répertoires de partage aient les permissions appropriées pour que l'utilisateur `sambauser` puisse y accéder. Ici l'utilisateur `sambauser` possède l'accès complet aux répertoires :

```
sudo chown -R sambauser:sambauser /mnt/raid/partage1 /mnt/raid/partage2
/mnt/raid/perso
sudo chmod -R 770 /mnt/raid/partage1 /mnt/raid/partage2 /mnt/raid/perso
```

5. Redémarrage de Samba

```
sudo systemctl restart smbd
```

Service et module pour la configuration de webdav

Installation d'Apache et des modules WebDAV

On commence par installer Apache :

```
sudo apt install apache2
```

Activation des modules dav et dav_fs pour webdav :

```
sudo a2enmod dav
sudo a2enmod dav_fs
```

Configuration d'Apache pour WebDAV

On crée un fichier de configuration pour le site WebDAV dans le répertoire /etc/apache2/sites-available/. Par exemple :

```
sudo nano /etc/apache2/sites-available/webdav.conf
```

On ajoute le contenu suivant pour configurer le site WebDAV :

```
<VirtualHost *:80>
        ServerName 172.16.0.25
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        Alias /bob /var/www/webdav/sftp/bob
        <Directory /var/www/webdav/sftp/bob>
                AuthType Basic
                AuthName "Restricted Content"
                AuthUserFile /etc/apache2/.htpasswd
                Require user bob
                Dav On
                Options Indexes
                AllowOverride None
                Options Indexes FollowSymLinks
        </Directory>


        Alias /leo /var/www/webdav/sftp/leo
        <Directory /var/www/webdav/sftp/leo>
                AuthType Basic
                AuthName "Restricted Content"
                AuthUserFile /etc/apache2/.htpasswd
                Require user leo
                Dav On
                Options Indexes
                AllowOverride None
                Options Indexes FollowSymLinks
        </Directory>

</VirtualHost>
```

Création du répertoire WebDAV

On crée le répertoire qui servira de point de montage pour le service WebDAV :

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

Redémarrage d'Apache

On active le site WebDAV et on redémarre Apache pour appliquer les modifications de configuration :

```
sudo a2ensite webdav.conf
sudo systemctl restart apache2
```

Configuration de l'accès

On définit les utilisateurs autorisés à accéder au service WebDAV en utilisant les fichiers de mots de passe Apache. On crée un fichier de mot de passe pour les utilisateurs autorisés :

```
sudo htpasswd /etc/apache2/.htpasswd username
```

On remplace username par le nom d'utilisateur de son choix. On sera invité à saisir un mot de passe pour cet utilisateur.

Accès au service WebDAV

On pourra accéder au service WebDAV en utilisant un client WebDAV tel que Cyberduck, cadaver ou simplement en montant le partage dans son système de fichiers via davfs2. Par exemple, si on utilise cadaver, on peut exécuter :

```
cadaver http://server_ip/webdav
```

Index Custom :

```
<!DOCTYPE html>
<html>
<head>
    <style>
        h1 {
        text-align: center;
        color: #392007;
        font-size: 5em;
}

    h2 {
        text-align: center;
        color: #392007;
        font-size: 4em;
    }

    html {
        background-color: antiquewhite;
    }

    a {
        color: #5f350b;
    }

    </style>
</head>
<body>

<h1>Webdav</h1>

<h2><a href="http://172.16.0.25/bob">Bob</a></h2>
<h2><a href="http://172.16.0.25/leo">Leo</a></h2>

</body>
</html>
```

Installation et configuration service icsci

Configuration d'un service iSCSI de manière à ce que le nouveau disque iSCSI soit monté dans le RAID 5 existant dans `/mnt/md5/syscsi` :

Installation du logiciel open-iscsi

On commence par installer le logiciel open-iscsi :

```
sudo apt install open-iscsi
```

Configuration de l'initiateur iSCSI

On édite le fichier `/etc/iscsi/iscsid.conf` :

```
sudo nano /etc/iscsi/iscsid.conf
```

On s'assure que la ligne `node.startup` est définie sur `automatic` :

```
node.startup = automatic
```

Découverte et connexion aux cibles iSCSI

On utilise la commande `iscsiadm` pour découvrir et se connecter à la cible iSCSI. On remplace `target_ip` par l'adresse IP de la cible iSCSI :

```
sudo iscsiadm -m discovery -t st -p target_ip
sudo iscsiadm -m node --targetname target_name --portal target_ip --login
```

Vérification de la connexion iSCSI

On vérifie que la connexion iSCSI est établie avec succès :

```
dmesg | grep sd
```

On s'assure que le nouveau disque iSCSI est détecté.

Configuration du montage automatique des disques iSCSI

On formate le nouveau disque iSCSI et on le monte dans `/mnt/md5/syscsi` :

```
sudo mkfs.ext4 /dev/sdX
sudo mount /dev/sdX /mnt/md5/syscsi
```

Configuration automatique du montage iSCSI

On ajoute une entrée dans `/etc/fstab` pour monter automatiquement le disque iSCSI lors du démarrage :

```
/dev/sdX /mnt/md5/syscsi ext4 defaults,_netdev 0 0
```

Redémarrage du système

On redémarre le système pour appliquer les modifications :

```
sudo reboot
```

Après le redémarrage, le nouveau disque iSCSI devrait être monté dans le RAID 5 existant dans `/mnt/md5/syscsi`. On remplace `target_ip` et `target_name` par les valeurs appropriées pour sa configuration iSCSI.

Installation et configuration de Nextcloud

Téléchargement de Nextcloud sur Debian

Depuis notre serveur, on télécharge l'archive zip de Nextcloud. La. on va sur <https://nextcloud.com/install> pour récupérer le lien de téléchargement et depuis notre terminal, on exécute la commande suivante (à adapter en fonction de la version) :

```
wget https://download.nextcloud.com/server/releases/nextcloud-19.0.1.zip
```

On décompresse l'archive :

```
sudo apt install unzip
sudo unzip nextcloud-19.0.1.zip -d /var/www/
```

Changement des droits pour qu'Apache puisse écrire dans le répertoire :

```
sudo chown www-data:www-data /var/www/nextcloud/ -R
```

Création d'une base de données et d'un utilisateur spécifique pour Nextcloud sur MariaDB

Depuis notre terminal, on lance mysql en admin :

```
sudo mysql
```

On crée ensuite une base de données pour Nextcloud. Nous allons appeler la base de données `nextcloud`. Mais on peut utiliser un autre nom.

```
create database nextcloud;
```

On crée l'utilisateur de la base de données. Encore une fois, vous pouvez utiliser votre nom préféré pour cet utilisateur. On remplace `votre_mot_de_passe` par notre mot de passe souhaité.

```
create user nextclouduser@localhost identified by 'votre_mot_de_passe';
```

On accorde à cet utilisateur tous les privilèges sur la base de données `nextcloud` :

```
grant all privileges on nextcloud.* to nextclouduser@localhost identified by
'your-password';
```

On applique les changements :

```
flush privileges;
```

Création d'un hôte virtuel Apache pour Nextcloud

On va faire les choses propres, donc on va utiliser notre ou un nom de domaine.

Pour cela, on va créer un hôte virtuel Apache avec notre domaine ou sous-domaine. On crée un fichier `nextcloud.conf` dans `/etc/apache2/sites-available/` :

```
sudo nano /etc/apache2/sites-available/nextcloud.conf
```

On copie/colle le texte suivant dans le fichier. On remplace `nextcloud.example.com` par notre propre sous-domaine voulu. On n'oublie pas de créer un enregistrement DNS A pour ce sous-domaine dans votre éditeur de zone DNS.

```
<VirtualHost *:80>
        DocumentRoot "/var/www/nextcloud"
        ServerName nextcloud.example.com

        ErrorLog ${APACHE_LOG_DIR}/nextcloud.error
        CustomLog ${APACHE_LOG_DIR}/nextcloud.access combined

        <Directory /var/www/nextcloud/>
            Require all granted
            Options FollowSymlinks MultiViews
            AllowOverride All

           <IfModule mod_dav.c>
               Dav off
           </IfModule>

        SetEnv HOME /var/www/nextcloud
        SetEnv HTTP_HOME /var/www/nextcloud
        Satisfy Any

       </Directory>

</VirtualHost>
```

On sauvegarde le fichier puis activez ce virtual host via la commande suivante :

```
sudo a2ensite nextcloud.conf
```

Activer les modules Apache nécessaires au bon fonctionnement de Nextcloud :

```
sudo a2enmod rewrite headers env dir mime setenvif ssl
```

On vérifiez notre configuration Apache :

```
sudo apache2ctl -t
```

Si « syntax OK » => redémarrerApache pour que vos modifications prennent effet :

```
sudo systemctl restart apache2
```

Installation et activation des modules PHP

On lance la commande suivante pour installer les modules PHP nécessaires au bon fonctionnement de NextCloud (y'a du monde !):

```
sudo apt install php-imagick php7.4-common php7.4-mysql php7.4-fpm php7.4-gd
php7.4-json php7.4-curl  php7.4-zip php7.4-xml php7.4-mbstring php7.4-bz2
php7.4-intl php7.4-bcmath php7.4-gmp
```

On relance Apache :

```
sudo systemctl reload apache2
```

Activation du certificat HTTPS pour NextCloud

Avant de commencer à entrer des informations sensibles, on va installer un certificat. On Lance la commande suivante :

```
sudo apt install certbot python3-certbot-apache
```

Puis, on réclame votre certificat Let's Encrypt via la commande suivante (en modifiant les informations en rouge par les vôtres) :

```
sudo certbot --apache --agree-tos --redirect --staple-ocsp --email
votre_email@example.com -d nextcloud.example.com
```

Certbot ne peut pas ajouter automatiquement l'en-tête HSTS dans le fichier de configuration Apache pour Nextcloud. Nous devons activer HSTS (HTTP Strict Transport Security), et on modifie le fichier :

```
sudo nano /etc/apache2/sites-enabled/nextcloud-le-ssl.conf
```

Nous pouvons ensuite ajouter la ligne suivante dans le bloc serveur SSL pour activer l'en-tête HSTS :

```
Header always set Strict-Transport-Security "max-age=31536000"
```

On ferme et sauvegarde votre fichier de configuration Apache.

On relance Apache pour que les changements soient pris en compte. Nous devrions maintenant pouvoir accéder à NextCloud en HTTPS sans problème :

```
sudo systemctl reload apache2
```

Terminer l'installation

Maintenant, depuis votre domaine / sous-domaine, vous allez pouvoir terminer l'installation de NextCloud. Nous allons devoir créer un compte admin, entrer le chemin du dossier de données Nextcloud, entrer les détails de la base de données que vous avez créés, etc...

Le dossier de données est l'endroit où les fichiers des utilisateurs sont stockés. Pour des raisons de sécurité, il est préférable de placer le répertoire de données en dehors du répertoire Web Nextcloud. Ainsi, au lieu de stocker les fichiers des utilisateurs sous `/var/www/nextcloud/data/`, nous pouvons le changer en `/var/www/nextcloud-data` que nous allons pouvoir créer avec la commande suivante:

```
sudo mkdir /var/www/nextcloud-data
```

On pose ensuite les droits correctement sur ce répertoire :

```
sudo chown www-data:www-data /var/www/nextcloud-data -R
```

Sécurisation des données dans le système RAID 5

Pour déplacer le stockage de Nextcloud vers le RAID 5, on effectue les étapes suivantes :

On stop le service Nextcloud car il a été lancée auparavant:

```
sudo systemctl stop apache2
```

On copie le dossier de données Nextcloud vers le nouveau répertoire dans le RAID 5 :

```
sudo rsync -av /var/www/nextcloud/data /mnt/md5/syscsi/nextcloud-data/
```

Modifiez les droits de propriété et les permissions sur le nouveau répertoire :

```
sudo chown -R www-data:www-data /mnt/md5/syscsi/nextcloud-data/
sudo chmod -R 750 /mnt/md5/syscsi/nextcloud-data/
```

On modifie le fichier de configuration de Nextcloud pour qu'il pointe vers le nouveau répertoire de données :

```
sudo nano /var/www/nextcloud/config/config.php
```

Recherchez la ligne suivante :

```
'datadirectory' => '/var/www/nextcloud/data',
```

Et remplacez-la par :

```
'datadirectory' => '/mnt/md5/syscsi/nextcloud-data',
```

Enregistrez et fermez le fichier.

On démarre le service Nextcloud :

```
sudo systemctl start apache2
```

On vérifie que Nextcloud fonctionne correctement et que nous pouvons accéder à tous nos fichiers.

Maintenant, on peut rentrer tout ce qu'il faut pour configurer NextCloud.