

**BTS Services informatiques aux organisations - SISR****Session 2022****E4 – Support et mise à disposition de services informatiques****Coefficient 4****DESCRIPTION DE LA REALISATION PROFESSIONNELLE****NOM et prénom du candidat : LEDUC Quentin****N° candidat : 02145639104****Contexte de la réalisation professionnelle**

Scani fait parti de plusieurs associations ... Felin pour la partie hébergement de site internet et de données utilisateurs (RP4) et le Maillet de Joigny (Fablab). Au coeur de cette réalisation professionnelle, les ordinateurs solidaires, financé par la Mairie de Joigny pour lutter contre le manque informatique des Joviniens pendant la crise sanitaire. Ces ordinateurs Solidaires sont créés à partir de Raspberry Pi. Avec la création des boîtiers, des fixations etc ... nous fournissons aussi un système d'exploitation modifié par nos soins et il nous faut un système de sauvegarde pour créer des versions.

**Intitulé de la réalisation professionnelle**

Créer une image Raspberry sur Synology

**Période de réalisation : DU 01/07/21 AU 11/07/21****Lieu : Joigny****Modalité : Individuelle****Principale(s) activité(s) concernée(s) :**

- Gérer des sauvegardes

**Conditions de réalisation**

- **Ressources présentes**

**On a des sauvegardes de nos sites internet pour le FTP et la BDD**

- **Résultats attendus**

**La solution que nous voulons mettre en place est de sauvegarder notre carte SD sous forme d'image img chaque jour. Si notre carte cessait de fonctionner nous pourrions alors reprendre cette image et la reflasher sur une nouvelle carte avec etcher ou bien un autre logiciel.**

- **Durée de réalisation**

**3H****Modalités d'accès à cette réalisation professionnelle.****Site internet : [www.netwaze.fr](http://www.netwaze.fr)****Aller dans « Réalisations Professionnelles » Mot de passe : Mr.Robot**

## PRÉREQUIS MATÉRIEL

- Un NAS Synology (ou un partage NFS)
- Un Raspberry PI

## PRÉREQUIS

Identifions d'abord le réseau. Notre Raspberry PI a l'adresse ip 192.168.3.183 et notre NAS a l'adresse IP 192.168.3.246 et 192.168.3.247.

Ce qui va se passer : notre Raspberry PI va envoyer les partitions BOOT et SYSTEME pour en créer une image IMG sur le NAS Synology.

## INSTALLATION DE NFS SUR LE RASPBERRY PI

Tout d'abord nous devons installer le nfs sur notre Raspberry PI. Pour ma part mon Raspberry PI fonctionne avec Rocky Linux, une distribution basée sur une RHEL. Donc ceci ne fonctionnera pas avec un Ubuntu ou une Debian.

Installation du NFS sur Rocky Linux : `$ dnf install nfs-utils`

On l'active au démarrage et on le lance.

```
$ systemctl enable nfs-server
```

```
$ systemctl start nfs-server
```

```
$ systemctl status nfs-server
```

```
[[root@WebServer-Rocky ~]# systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor >
   Active: active (exited) since Sun 2021-07-11 14:57:20 UTC; 1min 15s ago
   Process: 8233 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then s>
   Process: 8221 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
   Process: 8219 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCE>
   Main PID: 8233 (code=exited, status=0/SUCCESS)

Jul 11 14:57:20 WebServer-Rocky systemd[1]: Starting NFS server and services...
Jul 11 14:57:20 WebServer-Rocky systemd[1]: Started NFS server and services.
[[root@WebServer-Rocky ~]# █
```

# CRÉATION D'UN DOSSIER NFS SUR LE SYNOLOGY

Dans notre gestionnaire de stockage, nous créons un nouveau volume BackupPI.

 **Volume 5** - BackupPI 17 Mo / 192 Go 0%  
Sain

## Infos

Système de fichiers :

Btrfs

Puis dans file station nous créons notre dossier partager BackupPI puis créer le NFS.



**BackupPI**

Volume 5 (BackupPI)

## Créer une règle NFS

Nom d'hôte ou IP :  i

Privilège :

Squash :

Sécurité :

- Activer le mode asynchrone
- Permettre les connexions à partir des ports non privilégiés (ports supérieurs à 1024)
- Permettre à des utilisateurs d'accéder aux sous-dossiers montés

Annuler

Sauvegarder

Pendant la création de notre NFS on indique uniquement a notre Raspberry PI d'utiliser notre stockage NFS.

## CRÉATION DU POINT DE MONTAGE NFS

Sur le Raspberry PI on créer un nouveau dossier pour le point de montage. Je vais le mettre dans le dossier /mnt.

```
$ mkdir /mnt/NAS
```

On le monte a chaque démarrage du système. Pour se faire on ajoute dans notre fstab.

```
$ nano /etc/fstab
```

Puis on ajoute ces 2 lignes :

```
192.168.3.246:/volume5/BackupPI /mnt/NAS nfs rw,user 0 0
```

```
192.168.3.247:/volume5/BackupPI /mnt/NAS nfs rw,user 0 0
```

(On indique nos 2 IP local, au cas où notre port switch crame nous avons toujours notre NFS en fonctionnement)

On monte le tout avec la commande : `$ mount -a`

Et on regarde que notre NFS soit bien monté avec la commande : `$ df -h`

```
[root@WebServer-Rocky ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/root                  29G       2.7G   27G  10% /
devtmpfs                   1.9G         0   1.9G   0% /dev
tmpfs                      1.9G         0   1.9G   0% /dev/shm
tmpfs                      1.9G       25M   1.9G   2% /run
tmpfs                      1.9G         0   1.9G   0% /sys/fs/cgroup
/dev/mmcblk0p1             488M       64M   424M  14% /boot
tmpfs                      380M         0   380M   0% /run/user/0
192.168.3.246:/volume5/BackupPI 192G       17M   192G   1% /mnt/NAS
192.168.3.247:/volume5/BackupPI 192G       17M   192G   1% /mnt/NAS
[root@WebServer-Rocky ~]# █
```

## CRÉATION DU SCRIPT

Nous devons créer un script en bash qui créera une image entière du Raspberry PI directement dans notre dossier NFS créer précédemment. Le problème, notre image fera 32GB car nous sauvegarderons toute notre carte SD. Pour contrer ce problème nous utiliseront un deuxième script pishring qui redimensionnera notre image selon sa réelle taille. Il est aussi pratique car ce script redimensionne ensuite notre image sur n'importe qu'elle taille de stockage. Ces images seront ensuite ranger dans un dossier IMG par date.

```
#!/bin/bash
#Chemin du log
LOGFILE=/mnt/NAS/BackupPI.log
#Définition du format de la date
DATE=$(date +"%d-%m-%Y-%H-%M-%S")
#Rempli le fichier log avec la date de la sauvegarde
echo $DATE Debut de la sauvegarde >> $LOGFILE
#Format du fichier image de sortie
FileName=BackupPI_$(date +"%d-%m-%Y-%H-%M-%S").img
#Suppression des images a partir de la 10
RETENTION=10
#Dossier de sortie de l'image
File=/mnt/NAS/$FileName
#Création de l'image
dd if=/dev/mmcblk0 bs=1M of=$File >> $LOGFILE && sync >> $LOGFILE
#Déplacement dans notre dossier NFS
cd /mnt/NAS
#Lancement du script de pishrink qui rezize notre image
bash ./pishrink.sh BackupPI_$(date +"%d-%m-%Y-%H-%M-%S").img
#Déplacement de notre Image dans le dossier IMG
mv BackupPI_$(date +"%d-%m-%Y-%H-%M-%S").img IMG/
#Suppression des images
sudo find /mnt/NAS/IMG/* -mtime 7 -exec rm {} \;
```

## TEST DU SCRIPT

Maintenant que le script est créé nous devons mettre à la racine de notre NAS tous les éléments dont nous avons besoin, notre script, le script de pishrink et notre dossier IMG puis de lancer le script manuellement pour voir si il fonctionne avec la commande : `$ bash /mnt/NAS/script.sh`

 #recycle	Dossier	12/07/2021 12:41:14
 IMG	Dossier	12/07/2021 12:39:25
 BackupPI_12-07-2021-12-41-20.img	22.63 Go IMG Fichier	12/07/2021 12:49:56
 BackupPI.log	 <b>BackupPI.log - Éditeur de texte</b>	
 pishrink.sh	Fichier ▾    Modifier ▾    Recherche ▾    Préférences	
 script.sh		

1	12-07-2021-12-41-20	Debut de la sauvegarde
2		
3		

Nous pouvons alors voir la création de notre image sur notre NAS puis se redimensionner avec le script de pishrink puis aller dans le dossier IMG avec une taille beaucoup plus intéressante que celle d'origine.

```

_/: ***** FILE SYSTEM WAS MODIFIED *****
_/: 56956/1838640 files (0.3% non-contiguous), 810857/7689216 blocks
resize2fs 1.45.6 (20-Mar-2020)
pishrink.sh: Shrinking filesystem ...
resize2fs 1.45.6 (20-Mar-2020)
Resizing the filesystem on /dev/loop0 to 986857 (4k) blocks.
Begin pass 2 (max = 62494)
Relocating blocks          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 3 (max = 235)
Scanning inode table      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 4 (max = 8515)
Updating inode references  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/loop0 is now 986857 (4k) blocks long.

pishrink.sh: Shrinking image ...
pishrink.sh: Shrunk BackupPI_12-07-2021-12-41-20.img from 30G to 4.3G ...
[root@WebServer-Rocky ~]#

```

Nom ^	Taille	Type de fichier	Date de modification
 BackupPI_12-07-2021-12-41-20.img	4.25 Go	IMG Fichier	12/07/2021 12:54:04

## AUTOMATISER LE SCRIPT

Pour automatiser le lancement du script nous allons créer une tâche CRON qui lancera le script chaque jours a minuit.

Pour créer cette automatisation, nous allons utiliser une tâche cron.

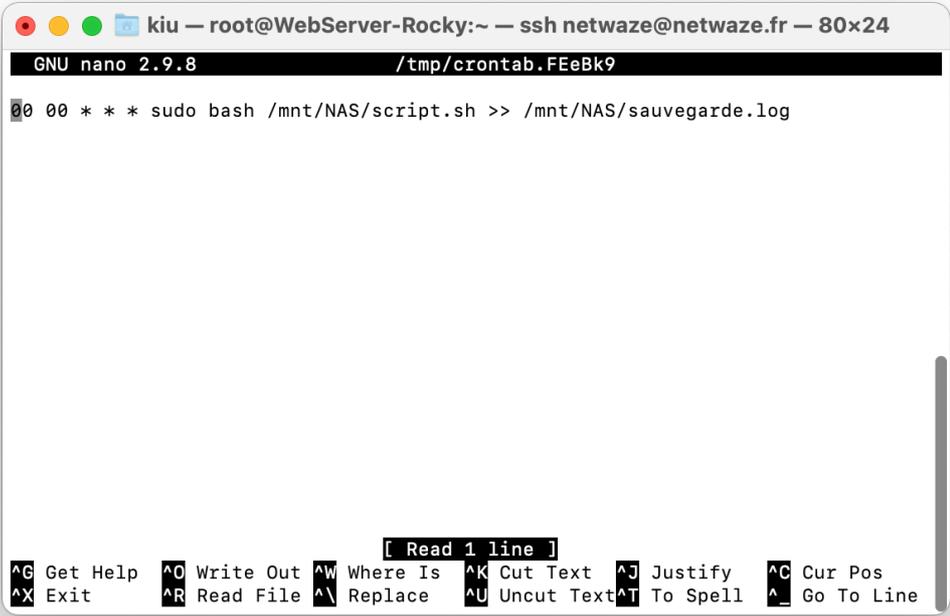
Pour installer cron : `dnf install crond`

Pour le lancer a chaque démarrage et l'activer :

`systemctl enable cron`

`systemctl start cron`

Pour créer une nouvelle tâche cron nous devons lancer la commande : `crontab -e` puis de rentrer : `00 00 * * * sudo bash /mnt/NAS/script.sh >> /mnt/NAS/sauvegarde.log`

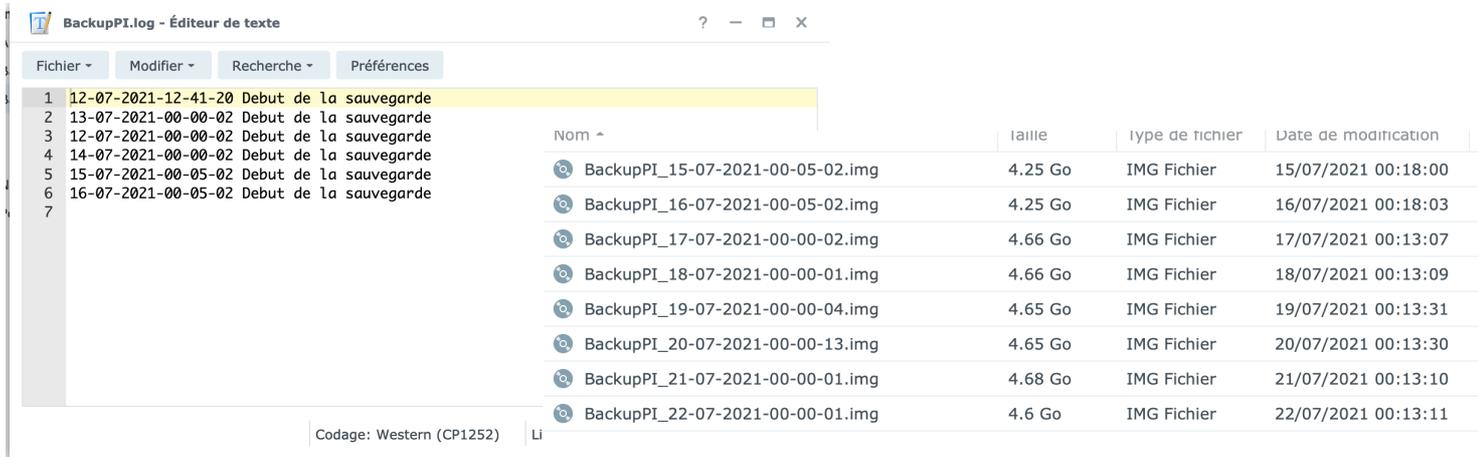


```
kiu — root@WebServer-Rocky:~ — ssh netwaze@netwaze.fr — 80x24
GNU nano 2.9.8 /tmp/crontab.FEeBk9
00 00 * * * sudo bash /mnt/NAS/script.sh >> /mnt/NAS/sauvegarde.log

[ Read 1 line ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

## Partie 2 – Validation.

Après plusieurs jours de sauvegarde nous voyons le bon fonctionnement de notre script qui récupère et supprime les sauvegardes dans un intervalle de 7 jours grâce au fichier log et au dossier IMG dans lequel sont entreposés nos images du Raspberry.



The screenshot shows a text editor window titled "BackupPI.log - Éditeur de texte". The log file contains the following entries:

Line	Timestamp	Event
1	12-07-2021-12-41-20	Debut de la sauvegarde
2	13-07-2021-00-00-02	Debut de la sauvegarde
3	12-07-2021-00-00-02	Debut de la sauvegarde
4	14-07-2021-00-00-02	Debut de la sauvegarde
5	15-07-2021-00-05-02	Debut de la sauvegarde
6	16-07-2021-00-05-02	Debut de la sauvegarde
7		

Below the log, a file manager view displays a list of backup files:

Nom	taille	Type de fichier	Date de modification
BackupPI_15-07-2021-00-05-02.img	4.25 Go	IMG Fichier	15/07/2021 00:18:00
BackupPI_16-07-2021-00-05-02.img	4.25 Go	IMG Fichier	16/07/2021 00:18:03
BackupPI_17-07-2021-00-00-02.img	4.66 Go	IMG Fichier	17/07/2021 00:13:07
BackupPI_18-07-2021-00-00-01.img	4.66 Go	IMG Fichier	18/07/2021 00:13:09
BackupPI_19-07-2021-00-00-04.img	4.65 Go	IMG Fichier	19/07/2021 00:13:31
BackupPI_20-07-2021-00-00-13.img	4.65 Go	IMG Fichier	20/07/2021 00:13:30
BackupPI_21-07-2021-00-00-01.img	4.68 Go	IMG Fichier	21/07/2021 00:13:10
BackupPI_22-07-2021-00-00-01.img	4.6 Go	IMG Fichier	22/07/2021 00:13:11

The text editor also shows the encoding: "Codage: Western (CP1252)".

## Partie 3 – Veille technologique.

Nous aurions eu la possibilité de créer nos images à la main tout les jours avec le logiciel Win32DiskImager ou avec la commande DD sur Linux. Pour le Stockage, à la place d'un serveur Synology, on aurait pu utiliser TrueNAS, Openmediavault ou un Serveur NFS Basic.

Link :

Win32DiskImager : <https://sourceforge.net/projects/win32diskimager/>

TrueNas : <https://www.truenas.com/>

Openmediavault : <https://www.openmediavault.org/>