

Sauvegarde et restauration carte SD

Mise en place d'un script de sauvegarde de la carte SD d'un Raspberry vers un disque dur ou un emplacement réseau et mode opératoire pour la restauration sur une nouvelle carte SD.

Emplacement de la sauvegarde

Commencer par créer un emplacement pour la sauvegarde de la carte SD.

L'emplacement peut être sur un disque dur local ou sur un disque réseau monté sur le Raspberry à sauvegarder.

Sur la machine hébergeant le disque dur de sauvegarde :

```
mkdir /home/partage/Sauvegarde_SD
```

Si le disque dur est en local sur la machine à sauvegarder, passer directement au chapitre suivant "script de sauvegarde". Sinon réaliser les étapes ci-dessous pour monter le disque dur en réseau sur la machine à sauvegarder.

Pré-requis : le disque dur de sauvegarde doit être partagé sur le réseau local.

Dans cet exemple le dossier **/home/partage** est partagé via samba avec les droits en lecture/écriture.

Exemple : [Voir § Partage réseau avec Samba](#)

Sur le Raspberry à sauvegarder, créer le répertoire pour le montage du disque réseau :

```
sudo mkdir /mnt/Sauvegarde_SD
```

Editer le fichier **/etc/fstab** pour monter le répertoire au démarrage :

```
sudo nano /etc/fstab
```

Ajouter la ligne suivante, penser à renseigner le username et password :

```
//192.168.1.50/partage/Sauvegarde_SD /mnt/Sauvegarde_SD cifs  
rw,username=xxx,password=xxx,uid=1000,gid=1000,_netdev 0 0
```

Avec **192.168.1.50** = ip locale de la machine hébergeant le disque dur et **partage** = [nom du partage] configuré dans **/etc/samba/smb.conf**.

Monter le disque :

```
sudo mount /mnt/Sauvegarde_SD
```

Vérifier si le montage a fonctionné :

```
mount | grep /mnt/Sauvegarde_SD
```

Script de sauvegarde

Créer le script **Save_SD.sh** dans le répertoire **/home/pi** du Raspberry à sauvegarder.

```
nano Save_SD.sh
```

Copier - coller le script ci-dessous et enregistrer. Penser à ajuster la variable **DOSSIER**. Si nécessaire, dé-commenter les lignes pour l'arrêt et le redémarrage des services Nginx, PHP et MySQL. Le log des sauvegardes se fait dans **/home/pi/Save_SD.log**.

```
#!/bin/bash
exec 3>&1 1>>/home/pi/Save_SD.log 2>&1
DATE=$(date +"%Y-%m-%d %H%M")
DOSSIER=/mnt/Sauvegarde_SD
BOOT=Partition_1.tar.gz
SYSTEME=Partition_2.tar.gz

if grep -qs $DOSSIER /proc/mounts; then
    echo "==== Début de la sauvegarde $(date +"%Y-%m-%d %H%M") =====" | tee /dev/fd/3
    #echo "Arrêt des services nginx, php, mysql" | tee /dev/fd/3
    #sudo service nginx stop
    #sudo service php5-fpm stop
    #sudo service mysql stop
    echo "$(date +"%Y-%m-%d %H%M") : Début de la sauvegarde de /boot/" | tee /dev/fd/3
    sudo tar -cpf $DOSSIER/$BOOT /boot/ --one-file-system
    RESULT=$?
    if [ $RESULT = 0 ]; then
        echo "$(date +"%Y-%m-%d %H%M") : Sauvegarde de /boot/ effectuée dans $BOOT" | tee
/dev/fd/3
    elif [ $RESULT = 1 ]; then
        echo "$(date +"%Y-%m-%d %H%M") : Sauvegarde de /boot/ effectuée dans $BOOT mais
certains fichiers peuvent différer" | tee /dev/fd/3
    else
        echo "$(date +"%Y-%m-%d %H%M") : Erreur lors de la sauvegarde de /boot/" | tee /dev/fd/3
    fi
    echo "$(date +"%Y-%m-%d %H%M") : Début de la sauvegarde de /" | tee /dev/fd/3
    sudo tar -cpf $DOSSIER/$SYSTEME / --one-file-system
```

```

RESULT=$?
if [ $RESULT = 0 ]; then
    echo "$(date +"%Y-%m-%d %H%M") : Sauvegarde de / effectuée dans $SYSTEME" | tee
/dev/fd/3
elif [ $RESULT = 1 ]; then
    echo "$(date +"%Y-%m-%d %H%M") : Sauvegarde de / effectuée dans $SYSTEME mais
certains fichiers peuvent différer" | tee /dev/fd/3
else
    echo "$(date +"%Y-%m-%d %H%M") : Erreur lors de la sauvegarde de /" | tee /dev/fd/3
fi
# echo "Redémarrage des services nginx, php, mysql" | tee /dev/fd/3
# sudo service mysql start
# sudo service nginx start
# sudo service php5-fpm start
    echo "==== Fin de la sauvegarde $(date +"%Y-%m-%d %H%M") =====" | tee /dev/fd/3
exit 0
else
    echo "$(date +"%Y-%m-%d %H%M") : Erreur, le repertoire de sauvegarde n'est pas monté" |
tee /dev/fd/3
exit 0
fi

```

Rendre le script exécutable :

```
chmod +x Save_SD.sh
```

Lancer la sauvegarde avec :

```
sudo ./Save_SD.sh
```

Une fois la sauvegarde terminée, vérifier la présence des archives **Partition_1.tar.gz** (~20 Mo) et **Partition_2.tar.gz** (>1,80 Go).

```
ls -l /mnt/Sauvegarde_SD/
```

Programmer la sauvegarde

Programmer la sauvegarde automatique de la carte SD tous les 6 mois avec crontab :

```
sudo crontab -e
```

Ajouter la ligne ci-dessous à la fin et enregistrer.

```
0 4 1 6,12 * sudo sh /home/pi/Save_SD.sh
```

La sauvegarde se lancera les 1er juin et 1er décembre à 4h00.

Pour personnaliser la programmation : <https://crontab.guru/>

Restauration de la carte SD

Utiliser **gparted** ou **fdisk** pour créer 2 partitions, la première de 50 Mo qui sera de type W95 et formatée en FAT32 et la seconde du reste de l'espace disponible de type Linux et formatée en ext4.

Repérer la carte SD et lancer fdisk (penser à démonter les partitions)

```
sudo fdisk -l
sudo fdisk /dev/sdd
```

- Lister les partitions : commande "p"
- Supprimer les partitions existantes : commande "d"
- Créer une nouvelle table vide de partitions DOS : commande "o"
- Ajouter une première partition : commande "n", p (primaire), numéro 1, premier secteur par défaut, +50M (taille)
- Ajouter une seconde partition : commande "n", p (primaire), numéro 2, premier secteur par défaut, deuxième secteur par défaut
- Modifier le type de la partition 1 : commande "t", numéro 1, code hexa "c"
- Sauvegarder et quitter : commande "w"

Formater et nommer les partitions.

Label première partition : **boot**

Label seconde partition : **rootfs**

```
sudo mkfs.vfat -F 32 -n boot /dev/sdd1
sudo mkfs.ext4 -b 4096 -L rootfs /dev/sdd2
```

Monter les partitions avec l'explorateur de fichier ou manuellement.

```
sudo mkdir /media/sdcard/boot
sudo mkdir /media/sdcard/rootfs
sudo mount /dev/sdd1 /media/sdcard/boot
sudo mount /dev/sdd2 /media/sdcard/rootfs
```

Extraire les archives de sauvegarde **Partition_1.tar.gz** et **Partition_2.tar.gz** respectivement sur les partitions **boot** et **rootfs**.

```
sudo tar -xvpf /chemin/vers/Partition_1.tar.gz -C /media/sdcard/boot/ --numeric-owner
sudo tar -xvpf /chemin/vers/Partition_2.tar.gz -C /media/sdcard/rootfs/ --numeric-owner
```

Une fois l'extraction terminée, déplacer tous les fichiers dans **/media/sdcard/boot/boot** vers **/media/sdcard/boot**.

```
mv /media/sdcard/boot/boot/* /media/sdcard/boot/
rm -R /media/sdcard/boot/boot
```

Démonter les partitions et retirer la carte SD.

```
sudo umount /media/sdcard/*
```

Il ne reste plus qu'à booter sur la nouvelle carte SD depuis le Raspberry.

Revision #3

Created 2025-12-31 20:40:16 UTC by Choukajohn

Updated 2026-01-01 00:03:07 UTC by Choukajohn