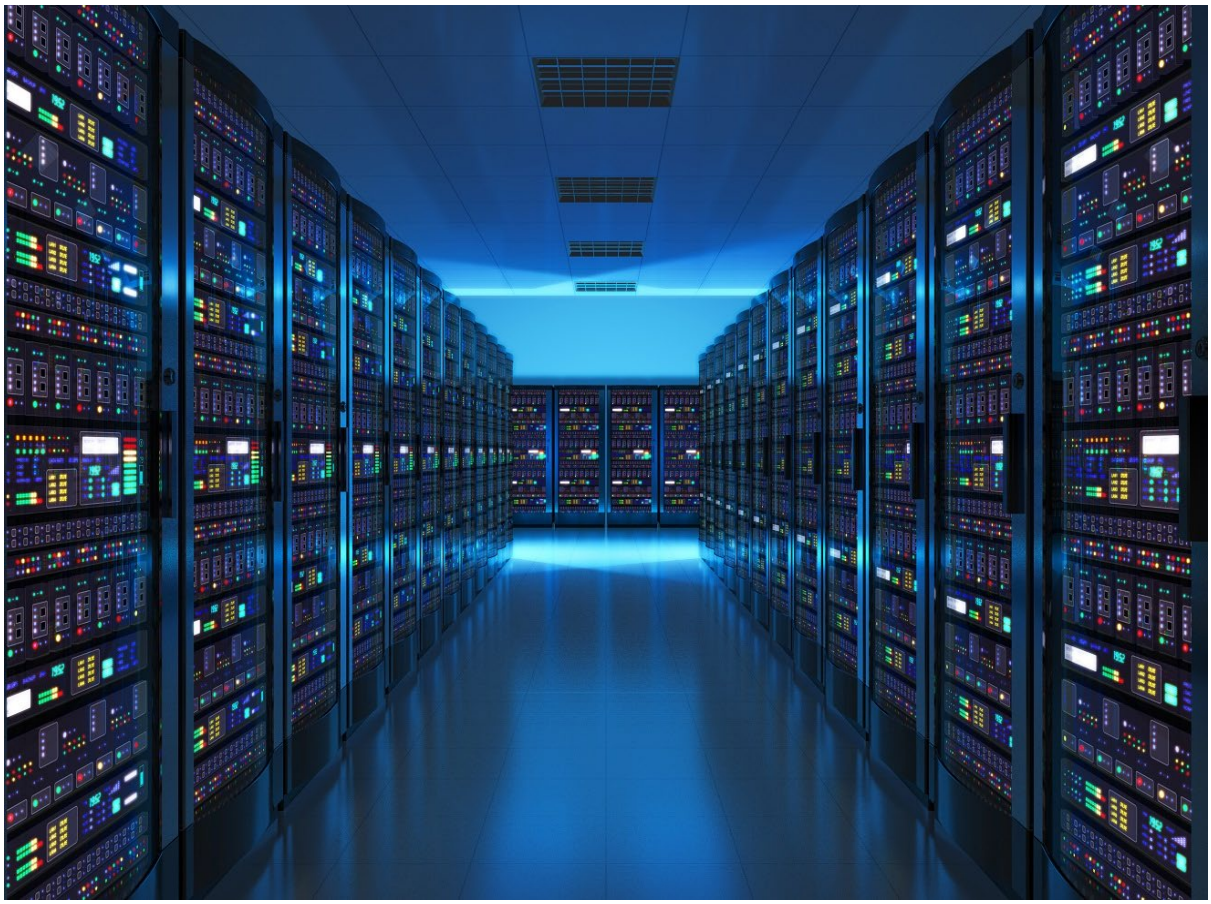
Royaume du Maroc
Université Hassan II de Casablanca
Ecole Nationale Supérieure d'Électricité et de Mécanique

Mini Projet :Script de sauvegard des systémes de fichiers d'un serveur

Administration Linux



Réaliser par :

Khalil Belhaj
Mohamed Oumessaoud
1 ére année GI 2021

Encadré par :

Pr.Meriyem Chergui

Sommaire :

Ⅰ)-Introduction:

La sauvegarde des données informatiques est le seul moyen qui permet à l'entreprise de faire face aux différents types de risques pouvant le mettre en danger. Outre le vol et l'incendie, une mauvaise manipulation du terminal notamment l'ordinateur peut aussi entraîner une perte de données.

Un système de fichiers est une façon de stocker les informations et de les organiser dans des fichiers Notre travail consiste donc à la conception et l'implémentation d'un script qui fait un sauvegarde des systèmes de fichiers d'un serveur ,Il doit prendre en compte tous les types de systèmes de fichiers spécifiés(RAID ,LVM,SCSI,IDE).

Ⅱ)-Sujet :

1-Enoncé de sujet :

L'objectif est de concevoir un script de sauvegarde des systèmes de fichiers d'un serveur.

Les différentes opérations à réaliser doivent se faire à partir d'un script unique. Ce script doit également gérer toutes les anomalies, une attention toute particulière sera donc portée sur les différentes incohérences possibles.

La stratégie de sauvegarde du système de fichiers dépend de son type et de son mode de gestion (LVM, Raid, etc). Le script déterminera automatiquement la liste des systèmes de fichiers à prendre en compte à partir du fichier système /etc/fstab.

Ce script s'exécute à partir d'une crontab (ou tout séquenceur ou ordonnanceur). Une exécution manuelle permet un affichage d'informations sur la sortie standard. Dans tous les cas, un rapport est fourni par mail, par un fichier de journalisation et par le mécanisme SYSLOG.

La version proposée du script prend en considération différents cas de figure, il vous sera aisé de l'adapter à vos besoins spécifiques. Vos modifications seront, en réalité, des suppressions de séquences d'instructions ou l'initialisation de variables.

Il a été choisi volontairement de ne pas fournir un script qui puisse gérer tous les cas de figure possibles par une configuration avec des variables ou des paramètres .

2-Cahier de charge :

a. Le service et l'environnement

Le service informatique doit gérer un parc de serveurs Linux. Les sauvegardes de l'environnement, et donc du système d'exploitation, sont fondamentales pour assurer un plan de reprise d'activités (PRA). L'équipe technique gère quotidiennement les sauvegardes des données applicatives par l'intermédiaire d'un applicatif spécifique. Ils souhaitent rajouter un processus de sauvegarde complète d'un serveur, indépendant de tous produits du marché, en utilisant uniquement les commandes Linux. Les sauvegardes sont stockées sur des bandes (DAT).

b. Les types de systèmes de fichiers :

Ces serveurs ont une configuration hétérogène des disques. Une analyse du parc a identifié les types suivants de gestion des systèmes de fichiers :

+des systèmes de fichiers standard sur des disques SCSI et IDE (/dev/sdX et /dev/hdX) ;

+des systèmes de fichiers sous la gestion de Logical Volume Manager (LVM) ;

+des systèmes de fichiers sous une configuration RAID comprenant des concaténations(Raid0), des miroirs(Raid1) et des Raid5.

Le script doit être un fichier uniquement positionné sur le serveur à sauvegarder. Il doit prendre en compte tous les types de systèmes de fichiers spécifiés ci-dessus.

3-Conception et les solutions techniques :

a- stratégie de Sauvegarde :

le script doit prendre en compte tous les types de systèmes de fichiers spécifiés . avant de sauvegarder les systèmes de fichiers on va prend une copie de quelque fichier qui contient les informations importantes comme les partitions de disques, les MBR des disques . et des fichiers systèmes comme les fichiers de « /etc/fstab » (File System table) qui contient les informations sur le montage de systèmes de fichiers et la configuration de l'amorce boot « boot/grub/grub.cfg» .L'intérêt de ces fichiers de faciliter la procédure de restauration .

la sauvegarde en général dépend les caractéristiques de chaque type de système de fichiers . en tout cas on va utiliser la commande « dump » dans toute la procédure de sauvegarde sauf le cas de LVM . Alors les stratégies de sauvegardes sont :

+Pour les configurations SCSI et IDE (/dev/sdX & /dev/hdX),Raid0,Raid5 on va faire une sauvegarde classique.

+ Pour LVM une sauvegarde par «Snapshot » pour éviter l'interruption des activités sur ses système de fichiers .

+Pour Raid1 on va choisi l'un des sous miroirs et le retirer de configuration, faire une sauvegarde puis le rajouter .

b-fichier de journalisation et le mécanisme SYSLOG :

le fichier de journalisation contient l'ensemble de déroulement de la sauvegarde système(les opérations et les messages de succès et d'échec de chaque phase) . ce fichier est très important pour envoyer un rapport après pour les prioritaires et les responsables de l'administration de serveur .

SYSLOG c'est un démon(un processus qui s'exécute ne arrière plan) qui permet d'envoyer des messages générés par le système

dans les fichiers log en fonction de fichier de configuration /etc/syslog.conf .

Ce derniers et composé de deux champs , un champ sélecteur et un champ action. Ces deux champs sont séparés par un ou plusieurs espaces ou tabulations. Le champ sélecteur précise un modèle de facility et priorité correspondant à l'action précisée.Les lignes commençant par un dièse («#») et les lignes vides sont ignorées.

Ce qui nous intéresse c'est le champ facility de mot clés localX(X={1,2,3….})

.chaque message a un niveau (Level) par exemple : error,debug , Emergency..

Pour envoyer un message dans le fichier de désiré en utilise la commande logger , exemple syntaxe :

```
logger -p local0.err " le message d'erreur  "
```

c-la commande dump :

On sait qu'il y a plusieurs programmes qui font les sauvegardes comme : dump,restore,tar,cpio,pax …

Alors pourquoi le choix de dump ?

La command dump est la plus forte . en effet : un test fait par Elizabeth D. Zwicky en 1991,elle a crée des système de fichiers avec grande a créé des systèmes de fichiers avec une grande variété de particularités inhabituelles (et quelques unes pas tellement inhabituelles) et a testé chacun des programmes en faisant une sauvegarde et une restauration de ces systèmes de fichiers. Parmi les spécificités testées: fichiers avec des trous, fichiers avec des trous et des blocs de caractères "null", fichiers dont les noms comportent des caractères inhabituels, les fichiers illisibles ou impossible à modifier, les périphériques, fichiers dont la taille change pendant la sauvegarde, fichiers créés ou détruits en cours de sauvegarde et bien plus. Ce test a montré que dump est la méthode de sauvegarde la plus efficace dans toutes les situations .

Syntaxe général : ***dump [-level#] [-a autosize] [-A file] [-B records] [-b blocksize]***
[-d density] [-D file] [-e inode numbers] [-E file] [-f file]
[-F script] [-h level] [-I nr errors] [-jcompression level] [-L label]
[-Q file] [-s feet] [-T date] [-y] [-zcompression level] files-to-dump
dump [-W | -w]

Exemple : sudo dump -0uf /dev/nst0 /dev/sda6

-level # : Le niveau de vidage qui est un entier compris entre 0 et 9. Si l'utilisateur a demandé d'effectuer une sauvegarde complète ou une sauvegarde des seuls nouveaux fichiers ajoutés après le dernier vidage d'un niveau inférieur.

– f fichier : Ceci spécifie le fichier dans lequel la sauvegarde sera écrite. Le fichier peut être un lecteur de bande, une disquette, un fichier ordinaire ou une sortie standard.

-u : Ceci enregistre et met à jour la sauvegarde dans le fichier /etc/dumpdates

Remarque : option -0uf est l'option qu'on va utiliser pour la sauvegarde .

d- Lecture des bandes(DAT) :

Un périphérique à bande est une bande magnétique sur laquelle les données sont stockées et accessibles de manière séquentielle. Les données sont écrites sur ce périphérique de bande à l'aide d'un lecteur de bande. Il n'est pas nécessaire de créer un système de fichiers pour stocker des données sur un périphérique de bande. Les lecteurs de bande peuvent être connectés à un ordinateur hôte avec diverses interfaces telles que SCSI, FC, USB, SATA et d'autres interfaces.

Voici les différents types de périphériques de bande :

/dev/st : est un périphérique de rembobinage de bande.

/dev/nst est un périphérique de bande sans rembobinage.
Utilisez des périphériques sans rembobinage pour les sauvegardes quotidiennes.
Dans notre projet on va faire la sauvegarde plusieurs fois (pour chaque type de système de fichier)est la stockée sur une même DATE alors on va utiliser un périphérique de bande sans

rebobinage /dev/nst et à la fin de sauvegarde on va faire un rembobinage en utilisant la commande mt rewind .

Remarque : on a pas de lecture de band pour tester le programme on va faire un sauvegarde sur la sortie standard mais le script va contient les commandes de ce périphérique .

Ⅲ)-Script :

Avant l'utilisation de script il faut :

-avoir permission root.

-installation de SSMTP (ajouter un mail et son mot de passe dans le fichier de configuration)

-vérifier les variables de script avant l'exécuter.

a-Phase1 : déclaration des variables de script :

Le script constitué des variables à changer par l'administrateur selon les choix (répertoire de sauvegarde, périphérique de bonde ….) et ne contient pas des fonctions , un seul programme principal .

Script :

```
#!/bin/bash
###Les variables
#les variables modifiables par l'administrateur
TAPE="/dev/nst0"
USERMAIL="root khalil.belhaj.etu20@ensem.ac.ma"
Rep_sauve="/sauvegardes"
###############
#les variables du script
FicLog="${Rep_sauve}/sauvesys.log"
Date="`date +%d-%m-%y`"
SERVER="`uname -n`"
PATH=$PATH:/user/sbin:/sbin
fstab="/etc/fstab"
```

```
ListeFS=""
ListeRaid1=""
ListeLVM=""
ListeOrdre=""
MesPB=""
```

Le script a des variables modifiables par l'administrateur qui sont :

USERMAIL : Variable contient les destinataires de rapport par mail soit local (root par exemple) ou par mail externe (en utilisant SSMTP) .

Rep_sauve : Variable contient le répertoire dans lequel on va stocker les informations sur la configuration de serveur.

TAPE : variable défini le périphérique de lecture de bandes . il faut utiliser un périphérique sans rembobinage.

Les variables de scripts sont :

Ficlog : définit le fichier de journalisation qui contient les opérations de script et les erreurs rencontrées

SERVER :le nom du serveur.

Date :la date d'exécution .

fstab : le fichier fstab contient les système de fichiers ,leurs points de montages, leurs types.

PATH : variable définit répertoires des commandes d'administration .exemple par la commande echo $PATH :

/home/thor/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games

MesPB: la liste des anomalies rencontrées déroulement sauvegarde .

ListeOrdre :variable contient la liste de montages des système de fichiers sauvegardés .

ListeFS : variable contient la liste des systèmes de fichiers à sauvegarder à l'exception des LVM,Raid1.

ListeRaid1 : : variable contient la liste des systèmes de fichiers Raid1 à sauvegarder .

ListeLVM : : variable contient la liste des systèmes de fichiers LVM à sauvegarder .

b-Phase 2 : Phase préliminaire .

Cette phase est une phase préparatoire , dans cette partie on va préparer les taches suivantes :

1-Création du répertoire s'il n'existe pas .
2-initialisations de fichier de journalisation.
3-L'affichage de message qui indique le début de la sauvegarde
4initialisations de fichier de journalisation et la redirection des me ssages .

Script :

```
###La phase préparatoire
#partie 1
#creation du répertoire de sauvegard
[ -d $Rep_sauve ] || mkdir -p $Rep_sauve
#partie 2
#initialisation du fichier de journalisation
cat /dev/null > ${FicLog}

#partie 3
#affichage de message vers la sortie standard
#si le script est exécuté manuellement
#ce message n apparait pas lors d un exécution par
#contrab(ou tout ordonnanceur ou séquenceur )
if [ "`tty | cut -c 1-8`" = "/dev/tty" ]
then
    echo -
e "\n Debut de la sauvegarde systéme par dump du serveur
 $SERVER \
    \n Merci de partienter, la procedure est longue.\
      \n Vous pouvez consulter le fichier de log pour su
ivre \nl'avancement de la sauvgarde\n"
    echo -e "Lancement de la SAUVEGARD SYSTEME \
              \nServeur; $Server \
```

```
    \n$(date +'DATE: %d/%m/%y%tHEURE: %H:%M:%S') \n" >>
${FicLog}
fi
```

-descriptif de la commande :

```
[ -d $Rep_sauve ] || mkdir -p $Rep_sauve
```

[-d $Rep_sauve] -d c'est un opérateur de test de fichiers qui vérifier est ce que $Rep_sauve est un répertoire qui existe ou non . si non la commande « mkidr -p $Rep_sauve » crée le répertoire .

-descriptif de la commande :

```
if [ "`tty | cut -c 1-8`" = "/dev/tty" ]
```

la commande tty est une commande Unix qui affiche sur la sortie standard le nom du fichier connecté sur l'entrée standard. Ce test vérifier est ce que le script est exécuté manuellement ou bien en utilisant Crontab. La sortie de la commande ne donne pas toujours un texte qui commence par « /dev/tty » parfois en trouve « /dev/pts » .

c-Phase 3: le départ de sauvegarde :

Après la phase de déclaration des variables , et la préparation de répertoire de sauvegarde on passe maintenant aux taches simples de sauvegardes, dans cette phase on va faire les tâches suivantes :

+Prend une copie de configuration d'amorce boot grub.conf et le fichier fstab.

+les partitionnements et les tables MBR des disques

Script :

```
####La Phase 3
#partie1
#quelque fichiers systémes
cp -p /etc/fstab ${Rep_sauve}/fstab.$Date
cp /boot/grub/grub.cfg ${Rep_sauve}/grub.cfg.$Date
#partie 2
Liste="`ls /dev/sd[a-z] 2>/dev/null` `ls /dev/hd[a-
z] 2>/dev/null` "
for disk in $Liste
do
    fdisk -l $disk > /dev/null 2>&1
    if [ $? -eq 0 ]
    then
        fdisk -
l ${disk} > ${Rep_sauve}/vtoc.$Date.`echo ${disk} | cut -c 6-8`
    dd if=${disk} of=${Rep_sauve}/mbr.$Date.`echo ${disk} | cut -
c 6-8` bs=512 count=1 > /dev/null 2>&1
    fi
done
```

d-Phase 4 : La liste de systéme de fichiers

```
######Phase 4 Fichier Système
#partie1 : remplissage de La liste ListeFS
tab=('')
let "i=0"
let "c=0"
twc="`cat /etc/fstab | grep -v '^#' | grep 'ext'`"
for p in $twc
do
    let "b=i%6"
    if [ $b -eq 0 ]
    then
        t="`echo $p | cut -c 1-5`"
        if [ $t = "UUID=" ] || [ $t = "LABEL" ]
        then
            z="`findfs $p`"
            tab[$c]=$z
            let "c+=1"
        else
                tab[$c]=$p
            let "c+=1"
        fi
    fi
    let "i+=1"
done
ListeFS="${tab[*]}"

#partie2
#recup les raid1 dans ListRaid1
for element in $ListeFS
do
    if (mdadm --detail $element |grep "raid1" ) > /dev/null 2>&1
    then
        nbsm=$(mdadm --
detail $element|grep "active"|awk '{print $NF}'|wc -l)
        if (($nbsm)>1);then
            ListeRaid1=${ListeRaid1}" $element "$(mdadm --
detail $element|grep "active"|awk '{print $NF}'|tail -1)
```

```
        else
            echo -e "PROBLEME pour $element \nLe nombre \
                de sous-miroirs est insuffisant." >> ${Ficlog}
            MesPB=$MesPB"\nSauvegarde de $element non realisee: \
                probleme sur sa configuration."
        fi
    fi
done


#Le supprime de ListeFS les raid1
tmp=""
tabt=("")
let "k=0"
let "i=0"
for liste in $ListeFS
do
    let "k=0"
    for raid in $ListeRaid1
    do
        if [ $raid == $liste ];then
            let "k=1"
        fi
    done
    if [ $k -eq 0 ];then
        tabt[$i]=$liste
        let "i+=1"
    fi
done
ListeFS="${tabt[*]}"

###ListeLVM le supprime de ListeFS
newLVM=("")
newFS=("")
let "q=0"
let "p=0"
for fs in $ListeFS
do
    if [ `echo $fs |cut -c 1-11` == "/dev/mapper" ]
    then
        newLVM[$q]=$fs
        let "q+=1"
```

```
    else
        newFS[$p]=$fs
        let "p+=1"
    fi
done
ListeFS="${newFS[*]}"
ListeLVM="${newLVM[*]}"
```

Description de la partie 1:

On définit une variable twc = cat /etc/fstab | grep -v '^#' | grep 'ext'`"

, ce variable prend le contenu de fichier fstab (cat /etc/fstab) à l'exception des commentaires qui sont les lignes qui commance par « # » . et on prendre les lignes qui contient 'ext'(on fait le sauvegarde pou les système de fichiers de type ext2 ,ext3 et ext4). Après on prend le premier champ de chaque ligne qui peut être un « UUID » , « Label » ou le nom de système de fichier . si c'est le dernier cas c'est bon sinon en utilise la commande « findfs » pour remplacer chaque Label et UUID par le nom de fichier système correspond.

Après en prend la nouvelle ListeFS , cette liste contient tous les système de fichier à sauvegarder par contre on a énoncé que ListeFS contient tous les système de fichier à l'exception de LVM et Raid1 . alors on doit remplir les deux liste ListeLVM et ListeRaid1 et supprimer ses système de fichier de la liste ListeFS on oublie pas de prend le point de montage pour chaque système de fichier et remplir la liste ListeOrdre .

Description de la partie 2:

On parcours la liste ListeFS cherchons le système de fichier Raid1, si la commande if (mdadm --detail $element |grep "raid1") > /dev/null 2>&1

Est exécuté sans erreur ce qui signifié que le systéme de fichier correspond à un raid1 on a besoin de trouver la liste des sous miroir de ce Raid1 et l'ajouter a cette liste .avant de faire ca on doit

calculer les nombre de sous miroir la variable nbsm=$(mdadm --detail $element|grep "active"|awk '{print $NF}'|wc -l)
Contiet le nombre des sous miroir plus 1 en compte aussi la tète de colonne « active » , la commande awk '{print $NF}' prend le dernier champ telque élement séparateur par défaux et l'espace vide , après en prend le texte retourner et on compte le nombre des lignes .
Si le nombre des miroirs et inférieur a 1 dans ce cas en affiche un message d'erreur et on n'oublier pas de le rediriger vers le fichier de journalisation. et la méme chose pour la liste ListeLVM.
Après on fait parcourir les deux liste ListeRaid1 et ListeFS pour éliminer les Raid1 existent sur ListeFS.
En fin on obtient les 4 listes ListeFs,ListeLVM,ListeRaid1,ListOrdre .

f-Phase 5 : La suavegarde de fichier

Après avoir les listes des différents fichier système . on va commencer notre sauvegarde .

Script :

```
#####PAHSE 4
#PARTIE1
#SAUVEGARD DE LISTEFS UTILISANT DUMP

set -- $ListeFS
if [ "$#" -ne 0 ]
then
    for liste in $ListeFS
    do
        echo -e "\nENTRAIN DE SAUVEGARDER LE FICHIER DE SYSTEM: $liste "
        dump -0uf ${TAPE} $liste >> /dev/null 2>&1
        if [ "$?" -ne 0 ]
        then
            MesPB=${MesPB}"\nEchec Sauvegard de $liste"
        else
            echo -e "\nSauvegard de $fs realisé avec succes."
            ListeOrdre=${ListeOrdre}"$ (mount |grep $liste| awk '{print $3}'"
        fi
    done;>> ${Ficlog} 2>&1
fi
```

```
###PARTIE 2:SAUVEGARD DE LISTELVM
if [ -n "$ListeLVM" ]
then
    for liste in $ListeFS
    do
        vg=`echo $LVM|awk -F"/" '{print $4}'|awk -F"--" '{print $1}'`
        lv=`echo $LVM|awk -F"/" '{print $4}'|awk -F"--" '{print $2}'`
        lvcreate --size 500M -s -n lesnap $liste
        echo -e "\nENTRAIN DE SAUVEGARDER LE FICHIER DE SYSTEM: $liste "
            dump -0uf ${TAPE} /dev/${vg}/lesnap >> /dev/null 2>&1
            if [ "$?" -ne 0 ]
        then
            MesPB=${MesPB}"\nEchec Sauvegard de $liste"
        else
            echo -e "\nSauvegard de $fs realisé avec succes."

                ListeOrdre=${ListeOrdre}"$ (mount |grep $vg-
$lv| awk '{print $3}'"
        fi
        lvremove -f /dev/${vg}/lensnap
    done
    #>> ${Ficlog} 2>&1
fi

###Partie3:SAUVEGARD DE FS RAID PAR DUMP
if [  -n "$ListeRaid1" ]
then
    set -- $ListeRaid1
    while [ $# -ne 0 ]
    do
        echo -e "\nProcedure RAID1"
        mdadm $1 --fail $2
        mdadm $1 --remove $2
        echo -e "\n+ Sauvegarde RAID1 de $1"
        dump 0uf ${TAPE} $2
        if [ $? -ne 0 ]
        then
            MesPB=$ {MesPB}"\néchec Sauvegarde de $1"
        else
            echo -e "\sauvegarde de $1 réalisée avec succès."
            ListeOrdre=${ListeOrdre}" $(mount |grep "$1"|awk '{print $3}')"
        fi
        mdadm $1 --add $2
        switch 2
    done
    #>> ${Ficlog} 2>&1
fi
```

Description de la partie 1:

Cette partie correspond à la sauvegarde de système de fichier SCSI,IDE,RAID0,RAID5 qui sont les éléments de la liste ListeFS. Premiérement on vérifier est ce que la liste est vide ou non (existence au moins de l'un de ses système de fichiers) on utilisant la commande :

```
set -- $ListeFS
if [ "$#" -ne 0 ]
```

la commande set prend les éléments de la liste ListeFS comme des arguments($1 le première élément …. $n n'éme élément de la liste)
pour la condition « $# » compte le nombre des arguments (la taille de la liste) , « -ne » est un comparateur signé différent de .
Puis Une boucle for permet le parcours de la liste et l'exécution de la commande :
dump -0uf ${TAPE} $liste

Description de la partie 2:

C'est la partie correspond à la sauvegarde de système de fichier LVM.
On vérifier est ce que la liste est vide ou non .

Commande : `if [ -n "$ListeLVM" ]`

en utilisant le comparateur -n (qui vérifier la chaîne de caractères n'est pas « vide »).
Puis on définit deux variable :

```
vg=`echo $LVM|awk -F"/" '{print $4}'|awk -F"--" '{print $1}'`
lv=`echo $LVM|awk -F"/" '{print $4}'|awk -F"--" '{print $2}'`
```

Remarque : On a installer ubuntu server pour le nom de lvm est de la forme indiqué dans l'exemple .
exemple : pour le lvm /dev/mapper/vg—ENSEM-lv—root
en doit prend /dev/mapper/vg-ENSEM/lv-root pour la command lvcreate.
Dans le script on a commenté le choix pour un autre serveur ca dépend la nomenclature des périphérique LVM.

Après la creation de lensnap correspond a chaque LVM :
lvcreate --size 500M -s -n lesnap $liste
on va le sauvegarder en utilisant la commande dump :
dump -0uf ${TAPE} /dev/${vg}/lesnap
après on va vérifier toujours le retour de la commande par la variable « $? »

s'il est différent de 0 alors la sauvegarde est échec et on va rediriger un message à MesPB indiquant le LVM correspondant .
après la sauvegarde on a pas besoin de snap qu'on a fait et pour avoir plus d'espace on va le supprimer :
lvremove -f /dev/${vg}/lensnap
Description de la partie 3:
Dans cette partie on va faire la sauvegarde de Raid1 , comme on a déjà mentionné au début de rapport qu'on va détacher un sous miroir de Raid1 on va faire la sauvegarde en utilisant dump , puis le rajouter .

```
if [  -n "$ListeRaid1" ]
```

vérification est ce que la liste est vide ou non.

```
set -- $ListeRaid1
    while [ $# -ne 0 ]
```

Prendre les éléments de la liste comme arguments pour parcourir la liste.

```
mdadm $1 --fail $2
mdadm $1 --remove $2
```

fail de sous miroir et le détache de Raid1

```
        dump 0uf ${TAPE} $2
```

Sauvegarde par dump de sous miroir et vérification d'existence d'erreur et le rediriger sous forme de message vers le fichiers de journalisation .

```
   mdadm $1 --add $2
```

rajouter le sous mirroir .

```
        switch 2
```

pour éviter boucle infini et pour atteindre la condition d'arrêt .

Remarque : Normalement maintenant on va faire un rembobinage en utilisant la commande
mt -f /dev/nst0 rewind
Malheureusement on a pas un lecture de bonde :(

g-Phase 6 (final) : l'affichage des problémes l'envoi de rapport par mail

Il nous reste que de lister les problèmes rencontrés et envoyer le contenu de fichier de journalisation vers la boite mail des administrateurs .

Script :

```
####Phase final
#partie 1
# La liste final des problémes
echo -e "\n\nListe des problémes:\n" > ${FicLog}
if [ -z "MesPB" ]
then echo -e "Aucun Probléme:" >> ${FicLog}
else echo -e "MesPB" >> ${FicLog}
fi
#Partie 2
#Mail de rapport
cat ${FicLog}}|ssmtp $USERMAIL
#partie 3
# syslog
if [ -z "$MesPB" ];then
    logger -i -
p local1.info "Sauvegarde de systéme est bien realisé!"
else
    logger -i -
p local1.info "Sauveguard systéme en Echec voir $FicLog pour plus
 de détail."
    logger -i -
p local2.err "Sauveguard systéme en Echec voir $FicLog pour plus
de détail."
fi
```

Description de la partie 1 :

si la liste des problèmes est vides en redirige un message vers le fichier de journalisation qui indique qu'il y a pas de problème.
Sinon en ajoute au fichier de journalisation la liste des problèmes.

Description de la partie 2 :

l'envoi de contenu de fichier de journalisation vers la boite mail des administrateurs . n'oublier pas de configurer le ssmtp ajoutant les donné de compte et le service smtp choisi (/etc/ssmtp/ssmtp.conf).

Description de la partie 3 :

Envoyer le message des problème au mécanisme syslog. Le niveau .err pour le message d'erreur et .info pour l'informatio.

Ⅳ)Conclusion

Ce projet à nous aidé pour comprend l'importance de programmation shell surtout ou niveau d'automatisation des taches .au lieu de faire tous ce travail en peut seulement faire un script et utiliser un ordonnanceur comme Crontab pour l'automatiser respectant une agenda donnée par les administrateurs de serveur .
C'était très intéressant malgré les difficultés qu'on a rencontrées. Comme le problème de l'absence d'un matériel comme le lecture de bondes donc on a pas fait le travail parfaitement mais en général c'est bon .

Références :

Syslog : http://marionpatrick.free.fr/man_html/html/syslog.conf_5.html

Chapitre 18. Stockage des donnéesPartie III. Administration Système : https://docs.freebsd.org/fr/books/handbook/disks/#backup-basics

Dump exemple : https://www.geeksforgeeks.org/dump-command-in-linux-with-examples/

Managing tape devices :
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_storage_devices/managing-tape-devices_managing-storage-devices

Opérateur de test des fichiers : https://abs.traduc.org/abs-5.1-fr/ch07s02.html

Tty : https://fr.wikipedia.org/wiki/Tty_(Unix)

Google et youtube.