

# µhackathon 5-6

## Gestió de fitxers + temporització

Volem temporitzar un sistema de backup remot mensual on cada més es realitza una còpia de nivell 0 (completa) , cada setmana una copia de nivell 1 (incremental setmanal sobre el nivell 0) i cada dia es fa una còpia de nivell 2 (incremental diari sobre el nivell 1) que compleixin les següents condicions:

### a) connexió entre màquines

- La connexió ha de ser segura (ssh).
- Per tal de poder temporitzar el backup es necessari que cada vegada que ens connectem a la maquina remota no ens demani el password

### b) Còpies de seguretat

- El directori a copiar es: /var/log
- Els arxius .gz han de quedar exclosos.
- Els arxius copiats han de tenir els següents permisos al directori destí:
  - owner: rwx, group: - - - , i others: - - -.
- (Nota: mireu al man les opcions de rsync)
- La copia s'ha de realitzar en una màquina remota (clon) amb adreça IP 172.16.1.2 Aquesta màquina ha de tenir una partició exclusiva per als backups muntada a /home/aso/backups.
- Al directori de backup es guardaran solament les tres còpies mes recents (rotació), identificades amb backup-<dir\_bck>-nivell<x>-<any mes dia hora minut segon>. Ex.: backup-log-nivell0-202212011030.
- Les ordres de rotació es donaran des de la maquina host cap a la màquina remota

### c) Temporització de les còpies

- Es realitzarà una copia nivell 0 (completa) cada primer dimarts del mes a les 00:05h
- Es realitzarà un copia incremental nivell 1 cada diumenge a les 00:15h.
- Es realitzarà un copia incremental nivell 2 cada dia a les 02:30h.

Entrega:

Especificacions

Llista de tasques amb responsable

Conexio entre maquines : Marc1, Marc2, Adri

Copies de seguretat: Mariona, Marc3, Adri

Temporitzacio: Francesco, Pau

Documentar: Adri

Algoritme

Descripció de la implementació amb captures de pantalla de cada responsable

El primer que hem de fer, és generar una key per ssh d'una **maquina clon**. Utilitzem la comanda "ssh-keygen".

```
root@MarcPerez (Mon Dec 18):<~># ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
```

Com utilitzem una maquina clon dintre del virtualbox, afegim dos adreces dintre de la mateixa red a /etc/network/interfaces.

Clon:

```
File Edit View Search Terminal Help
GNU nano 7.2 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
    address 172.16.1.2
    netmask 255.255.255.0

[ Read 11 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Host:

```
File Edit View Search Terminal Help
GNU nano 7.2 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
    address 172.16.1.1
    netmask 255.255.255.0

[ Read 11 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

```

root@MarcR (Mon Dec 18):</># ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2e+yB6E8Wl8QlnPZcAff5huFzkpm5zRgHKgKzgQ50ts root@aso-client
The key's randomart image is:
+---[RSA 3072]-----+
|           .+=0..|
|      . .   =ooo.+|
|    . =     ..++ . =|
|    . = . +o. + o.|
|    . E.S..o+ *..|
|      +  = .=. = .o|
|      oo o oo .. |
|      .  o..  |
|      .+.  |
+----[SHA256]-----+
root@MarcR (Mon Dec 18):</>#

```

Aquesta key generada la ficarem al fitxer de la màquina principal `/.ssh/authorized_keys` amb la comanda `ssh-copy-id`:

```

root@MarcPerez (Mon Dec 18):<~># ssh-copy-id -p 3022 aso@10.192.59.147
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
aso@10.192.59.147's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -p '3022' 'aso@10.192.59.147'"
and check to make sure that only the key(s) you wanted were added.

root@MarcPerez (Mon Dec 18):<~>#

```

```

root@MarcPerez (Mon Dec 18):<~># ssh -p 3022 aso@10.192.59.147
Linux aso-client 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 18 08:29:45 2023 from 10.0.2.2
aso@MarcP (Mon Dec 18):<~>$

```

b) Script de la copia de seguretat:

Creem un txt per excloure els arxius `.gz`.

Per això fem un `"nano /tmp/excludes.txt"` i guardem. Dintre del nano afegim la línia `"*.gz"`

```
as0@AdrianGaso-client: ~
File Edit View Search Terminal Help
GNU nano 7.2 /tmp/exclude.txt *
*.gz
```

```
root@AdrianG (Mon Dec 18):/home/homeB/aso# nano /tmp/exclude.txt
root@AdrianG (Mon Dec 18):/home/homeB/aso#
```

Crear un script per les còpies + donar-li permisos d'execució

```
root@marionaF (Mon Dec 18):<~/Documents># subl copia_seguretat.sh
root@marionaF (Mon Dec 18):<~/Documents># chmod +x copia_seguretat.sh
root@marionaF (Mon Dec 18):<~/Documents># ls
BadUsers.py      copia_incremental.sh  mkht3.py         user_stats.py
BadUsers.sh      copia_seguretat.sh    mkht3.sh         user_stats.sh
BadUsers_t6.py   delete_user.sh        ocupacio_t6.sh   user_stats.sha256sum
asosh.sh         llista_excloure.txt   subprocess       usuaris.txt
```

Creem la carpeta /home/aso/backups en la màquina clon:

```
root@MarcR (Mon Dec 18):</home># mkdir /home/aso/backups
root@MarcR (Mon Dec 18):</home>#
```

A continuació muntem la nostre partició extra en aquesta carpeta:

```
root@MarcR (Mon Dec 18):</home># umount /dev/sda4
root@MarcR (Mon Dec 18):</home># mount -t ext4 /dev/sda4 /home/aso/backups/
root@MarcR (Mon Dec 18):</home>#
```

```
root@MarcR (Mon Dec 18):</home># lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   32G  0 disk
├─sda1       8:1    0    8G  0 part /
├─sda2       8:2    0    1K  0 part
├─sda3       8:3    0    2G  0 part
├─sda4       8:4    0    6G  0 part /home/aso/backup
├─sda5       8:5    0    6G  0 part /usr/local
└─sda6       8:6    0    9G  0 part /home/homeB
sdb          8:16   0   16G  0 disk
├─sdb1       8:17   0    5G  0 part /home/homeA
└─sdb3       8:19   0    2G  0 part [SWAP]
sr0         11:0    1 1024M  0 rom
```

Hem de guardar els backups en un fitxer .tar

## Script nivell 0:

```
#!/bin/bash

# Directori d'origen per fer el backup
SOURCE_DIR="/var/log"

# Directori de destinació per guardar el backup
DEST_DIR="/home/aso/backups"

# Fitxer que conté la llista d'exclusions
EXCLUDES="/tmp/exclude.txt"
echo "*.gz" >$EXCLUDES

# Nom o adreça IP del servidor de backup
BSERVER="127.0.0.1"

# Comanda per generar la data actual amb format d'any, mes, dia, hora, minut i segon
BACKUP_DATE=$(date +"%Y-%m-%d_%H-%M-%S")

BASE_NAME="backup-log-nivell0"
BK_NAME="${BASE_NAME}-${BACKUP_DATE}"

# Opcions per a rsync
OPTS="-av --exclude-from=$EXCLUDES --exclude=$BASE_NAME --exclude=$BK_NAME
--chmod=u=rwx,g=,o="

# Transferència actual amb rsync
rsync $OPTS $SOURCE_DIR root@$BSERVER:$DEST_DIR
```

## Fer l'execució de la còpia de seguretat:

```
root@marionaF (Mon Dec 18):<~/Documents># ./copia_seguretat.sh
root@127.0.0.1's password:
sending incremental file list
log/
log/README -> ../../usr/share/doc/systemd/README.logs
log/Xorg.0.log
log/Xorg.0.log.old
log/Xorg.1.log
log/Xorg.1.log.old
```

Al entrar a dins de /home/aso/backups podem veure que s'ha fet correctament la còpia dels logs:

```
aso@MarcP (Mon Dec 18):<~/backup/backup-log-nivell0-2023-12-18_08-37-03>$ cd log/
aso@MarcP (Mon Dec 18):<~/backup/backup-log-nivell0-2023-12-18_08-37-03/log>$ ls
README      auth.log    debug      lastlog     syslog.1
Xorg.0.log   auth.log.1  debug.1    lightdm     user.log
Xorg.0.log.old boot.log    dpkg.log   messages    user.log.1
Xorg.1.log   bootstrap.log faillog     messages.1  wtmp
Xorg.1.log.old btmp       fontconfig.log mkht6.txt
alternatives.log cron.log    journal    private
apparmor     daemon.log kern.log    runit
apt          daemon.log.1 kern.log.1  syslog
aso@MarcP (Mon Dec 18):<~/backup/backup-log-nivell0-2023-12-18_08-37-03/log>$
```

Si ho comprovem amb la carpeta a copiar podem veure que NO s'han copiat els fitxers .gz:

```
root@marionaF (Mon Dec 18):</var/log># ls
README                boot.log              dpkg.log             private
Xorg.0.log            boot.log.1           dpkg.log.1           runit
Xorg.0.log.old        boot.log.2           dpkg.log.2.gz        syslog
Xorg.1.log            boot.log.3           exim4                syslog.1
Xorg.1.log.old        boot.log.4           faillog              syslog.2.gz
alternatives.log      bootstrap.log        fontconfig.log       syslog.3.gz
alternatives.log.1    bttmp               journal              syslog.4.gz
alternatives.log.2.gz bttmp.1             kern.log             user.log
apparmor              cron.log             kern.log.1           user.log.1
apt                   cron.log.1          kern.log.2.gz        user.log.2.gz
auth.log              cron.log.2.gz       kern.log.3.gz        user.log.3.gz
auth.log.1            cron.log.3.gz       kern.log.4.gz        user.log.4.gz
auth.log.2.gz         cron.log.4.gz       lastlog              wtmp
auth.log.3.gz         daemon.log           lightdm
auth.log.4.gz         debug               messages
root@marionaF (Mon Dec 18):</var/log>#
```

Si entrem a ASO podem veure que shan fet correctament els backups:

```
aso@MarcP (Mon Dec 18):<~>$ ls
Desktop  Downloads  Pictures  Templates  backup
Documents Music      Public    Videos
aso@MarcP (Mon Dec 18):<~>$ cd backup/
aso@MarcP (Mon Dec 18):<~/backup>$ pwd
/home/aso/backup
aso@MarcP (Mon Dec 18):<~/backup>$ ls
aso@MarcP (Mon Dec 18):<~/backup>$ ls
backup-log-nivell0-2023-12-18_08-37-03
aso@MarcP (Mon Dec 18):<~/backup>$ █
```

Podem veure que només es guarden els backups més recents, al anar creant còpies, es borran els backups més antics, en aquest cas

```
aso@MarcP (Mon Dec 18):<~/backup>$ ls
backup-log-nivell0-2023-12-18_08-37-03  backup-log-nivell0-2023-12-18_08-42-10
aso@MarcP (Mon Dec 18):<~/backup>$ ls
backup-log-nivell0-2023-12-18_08-37-03  backup-log-nivell0-2023-12-18_08-42-17
backup-log-nivell0-2023-12-18_08-42-10
aso@MarcP (Mon Dec 18):<~/backup>$ ls
backup-log-nivell0-2023-12-18_08-42-10  backup-log-nivell0-2023-12-18_08-42-29
backup-log-nivell0-2023-12-18_08-42-17
aso@MarcP (Mon Dec 18):<~/backup>$
```

**Script incremental 1:**

```
root@marionaF (Mon Dec 18):<~/Documents># subl incremental1.sh
root@marionaF (Mon Dec 18):<~/Documents># chmod +x incremental1.sh
```

```
#!/bin/bash
```

```
# Directori d'origen per fer el backup
SOURCE_DIR="/var/log"
```

```
# Directori de destinació per guardar el backup
DEST_DIR="/home/aso/backup"
```

```
# Fitxer que conté la llista d'exclusions
EXCLUDES="/tmp/exclude.txt"
echo "*.gz" >$EXCLUDES
```

```
# Nom o adreça IP del servidor de backup
BSERVER="10.192.59.147"
```

```
# Num PORT
PSERVER="3022"
```

```
# Comanda per generar la data actual amb format d'any, mes, dia, hora, minut i segon
BACKUP_DATE=$(date +"%Y-%m-%d_%H-%M-%S")
```

```
# Nom base per als fitxers de backup
BASE_NAME="backup-log-nivell1"
```

```
# Nivell de backup (0 per complet, 1 per incremental)
BACKUP_LEVEL=1
```

```
BK_NAME="${BASE_NAME}${BACKUP_LEVEL}-${BACKUP_DATE}"
```

```
# Fitxer de referència per backups incrementals
REF_FILE="${DEST_DIR}/${BASE_NAME}0-latest"
```

```
# Comprova si existeix el fitxer de referència per a backups incrementals
if [ ! -f "$REF_FILE" ]; then
    echo "No es troba el fitxer de referència del backup complet. Creant backup complet primer."
    BACKUP_LEVEL=0
    BK_NAME="${BASE_NAME}${BACKUP_LEVEL}-${BACKUP_DATE}"
    REF_FILE="${DEST_DIR}/${BK_NAME}"
fi
```

```
mkdir -p $DEST_DIR/
mkdir -p $DEST_DIR/$BK_NAME
```

```
# Opcions per a rsync
OPTS="-avz --exclude-from=$EXCLUDES --exclude=$BASE_NAME --exclude=$BK_NAME
--chmod=u=rwx,g=,o="
```

```
# Transferència actual amb rsync
rsync $OPTS -e "ssh -p 3022" $SOURCE_DIR root@$BSERVER:$DEST_DIR/$BK_NAME
```

```
# Actualitza el fitxer de referència si es tracta d'un backup complet
```

## Script incremental 2:



```

#!/bin/bash

# Directori d'origen per fer el backup
SOURCE_DIR="/var/log"

# Directori de destinació per guardar el backup
DEST_DIR="/home/aso/backups"

# Fitxer que conté la llista d'exclusions
EXCLUDES="/tmp/exclude.txt"
echo "*.gz" > $EXCLUDES

# Nom o adreça IP del servidor de backup
BSERVER="127.0.0.1"

# Comanda per generar la data actual amb format d'any, mes, dia, hora, minut i segon
BACKUP_DATE=$(date +"%Y-%m-%d_%H-%M-%S")

# Nom base per als fitxers de backup
BASE_NAME="backup-log-nivell2"

# Nivell de backup (0 per complet, 1 per incremental, 2 per incremental diari)
BACKUP_LEVEL=2

# Nom complet del fitxer de backup
BK_NAME="${BASE_NAME}-${BACKUP_DATE}.tar"

# Directori temporal per al backup
TMP_DIR="/tmp/backup-${BACKUP_DATE}"
mkdir -p $TMP_DIR

# Fitxer de referència per backups incrementals de nivell 1
REF_FILE="${DEST_DIR}/backup-log-nivell1-latest"

# Comprova si existeix el fitxer de referència per a backups incrementals de nivell 1
if [ ! -f "$REF_FILE" ]; then
    echo "No es troba el fitxer de referència del backup de nivell 1. Creant backup de nivell 1 primer."
    exit 1
fi

# Crear un arxiu tar del backup
tar -cvpf $TMP_DIR/$BK_NAME --exclude-from=$EXCLUDES --listed-incremental=$REF_FILE $SOURCE_DIR

# Transferència del arxiu tar amb rsync
rsync -av $TMP_DIR/$BK_NAME root@$BSERVER:$DEST_DIR/

# Neteja del directori temporal
rm -rf $TMP_DIR

# Actualitza el fitxer de referència de nivell 2
ln -fns $BK_NAME "${DEST_DIR}/backup-log-nivell2-latest"

```

c) Fitxer crontab amb les comandes per temporitzar les 3 còpies de seguretat:

Comandes a escriure:

- 5 0 1-7 \* \* [ "\$(date '+\%a')" = "Tue" ] && ./script\_nivell0.sh
- 15 0 \* \* Sun ./script\_nivell1.sh
- 30 2 \* \* \* ./script\_nivell2.sh

```
root@francesco0 (Mon Dec 18):<~># █
```

```
5 0 1-7 * * [ "$(date '+\%a')" = "Tue" ] && script_nivell0.sh
15 0 * * Sun script_nivell1.sh
30 2 * * * script_nivell2.sh
```