

# LSAP HW2 Report

## 1 謎樣的硬碟

### (1) 下載與解壓縮

首先使用 curl 下載題目提供的壓縮檔，接著用 unzip 解壓縮，得到四個硬碟映像檔 disk1.img disk4.img。

```
curl -L -o hw2_imgs.zip "https://www.dropbox.com/scl/fi/tnal4lumtu8wzp972r69k/hw2_imgs.zip?rlkey=563ypklncjbm6tifyq14drv7&st=ekgyz4wn&dl=1"
unzip hw2_imgs.zip
```

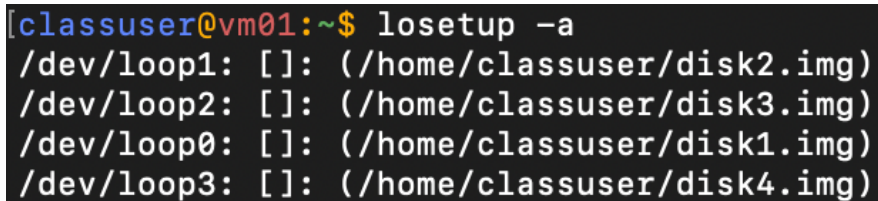
執行後可以看到四個.img 檔案。

### (2) 掛載到 loop device

接著用 losetup 把四個 img 檔掛載成虛擬硬碟 /dev/loop0 /dev/loop3。

```
sudo losetup -fP disk1.img
sudo losetup -fP disk2.img
sudo losetup -fP disk3.img
sudo losetup -fP disk4.img
losetup -a
```

這樣我們就能把它們當成真實硬碟來操作。



```
[classuser@vm01:~$ losetup -a
/dev/loop1: []: (/home/classuser/disk2.img)
/dev/loop2: []: (/home/classuser/disk3.img)
/dev/loop0: []: (/home/classuser/disk1.img)
/dev/loop3: []: (/home/classuser/disk4.img)
```

Figure 1: 確認每個 disk 的位置

### (3) 檢查 RAID metadata

執行 sudo mdadm --examine /dev/loop<id> 檢查四顆硬碟的輸出結果

```
sudo mdadm --examine /dev/loop0
sudo mdadm --examine /dev/loop1
sudo mdadm --examine /dev/loop2
sudo mdadm --examine /dev/loop3
```

執行後得到 /dev/loop1 沒有 superblock，判斷為壞掉的硬碟。其餘三顆 (/dev/loop0、/dev/loop2、/dev/loop3) 屬於同一個 RAID5。

註：損壞的硬碟：/dev/loop1

```

/dev/loop0:
  Magic : a92b4efc
  Version : 1.2
  Feature Map : 0x0
  Array UUID : b908d87f:bfd47d06:8174af4e:b6522d58
  Name : vm01:2 (local to host vm01)
  Creation Time : Mon Sep 22 06:33:04 2025
  Raid Level : raid5
  Raid Devices : 3

  Avail Dev Size : 405504 sectors (198.00 MiB 207.62 MB)
  Array Size : 405504 KiB (396.00 MiB 415.24 MB)
  Data Offset : 4096 sectors
  Super Offset : 8 sectors
  Unused Space : before=4016 sectors, after=0 sectors
  State : clean
  Device UUID : 68efcdd4:d05d83ac:fc684d51:6da60ec9

  Update Time : Mon Sep 22 06:37:06 2025
  Bad Block Log : 512 entries available at offset 16 sectors
  Checksum : 74f640e7 - correct
  Events : 18

  Layout : left-symmetric
  Chunk Size : 512K

  Device Role : Active device 0
  Array State : AAA ('A' == active, '.' == missing, 'R' == replacing)

```

Figure 2: 正常硬碟輸出

```
mdadm: No md superblock detected on /dev/loop1.
```

Figure 3: 壞掉硬碟輸出

#### (4) 組合 RAID

確認壞掉的硬碟後，用剩下三顆來組 RAID，嘗試 assemble 時發生衝突

```

sudo mdadm --assemble --readonly /dev/md0 /dev/loop0 /dev/loop2 /dev/loop3
cat /proc/mdstat

```

系統已有一組 md0，因此改用 /dev/md/10 名稱避免衝突。

```

classuser@vm01:~$ sudo mdadm --assemble --readonly /dev/md0 /dev/loop0 /dev/loop2 /dev/loop3
mdadm: Fail to create md0 when using /sys/module/md_mod/parameters/new_array, fallback to creation via node
mdadm: /dev/md0 is already in use.

```

Figure 4: 組合時發生衝突

```

sudo mdadm --assemble --readonly /dev/md10 /dev/loop0 /dev/loop2 /dev/loop3
cat /proc/mdstat

```

#### (5) 掛載 RAID 並讀取檔案

組合成功後，將 RAID 掛載到 /mnt/raid，然後查看裡面的內容。

```

sudo mkdir -p /mnt/raid
sudo mount /dev/md10 /mnt/raid

```

```
ls /mnt/raid
cat /mnt/raid/treasure.txt
```

在目錄中找到 treasure.txt，打開後讀到題目要求的 secret code。

註：Secret code：{IL0V3L5AP\_HAPPYHOLIDAY9/28}

```
classuser@vm01:~$ cat /mnt/raid/treasure.txt
Congratulations! You've found the treasure of this homework!
The secret code is {IL0V3L5AP_HAPPYHOLIDAY9/28}
Remember to write the secret code down in your report so that I know you found it :)
```

Figure 5: 找到 secret code

## (6) 清理

將 RAID 解除掛載，並移除 loop device

```
sudo umount /mnt/raid
sudo mdadm --stop /dev/md10
sudo losetup -d /dev/loop0
sudo losetup -d /dev/loop1
sudo losetup -d /dev/loop2
sudo losetup -d /dev/loop3
```

再次確認 /proc/mdstat 與 losetup -a，系統中已經沒有殘留的 RAID 或 loop 裝置。

## (7) 結論

壞掉的硬碟：disk2.img (/dev/loop1 無 md superblock)。

使用 /dev/loop0 /dev/loop2 /dev/loop3 成功組回 RAID5 (/dev/md/10)。

檔案系統為 ext4，成功讀到 treasure.txt 並取得 secret code {IL0V3L5AP\_HAPPYHOLIDAY9/28}。

## 2 NFS auto mounting

- OS: Ubuntu 20.04 / 22.04 (VM)
- Mode: 同一台 VM 同時作為 Server 與 Client
- autofs timeout: indirect map 60 秒、direct map 預設 300 秒

### (1) NFS Server Setup

在 VM 中安裝必要的套件

```
sudo apt update
sudo apt install -y nfs-kernel-server nfs-common autofs
```

### (2) 建立目錄結構

依照題目要求，建立 /srv/nfsroot 下的完整結構，並在每個目錄放入測試檔案

```

sudo mkdir -p /srv/nfsroot/shared/public
sudo mkdir -p /srv/nfsroot/shared/projects/group12
sudo mkdir -p /srv/nfsroot/shared/projects/HW2
sudo mkdir -p /srv/nfsroot/users/{alice,bob,charlie}
sudo mkdir -p /srv/nfsroot/services/{webdata,backups}

echo "Welcome to public" | sudo tee /srv/nfsroot/shared/public/README.txt
echo "Task for group12" | sudo tee /srv/nfsroot/shared/projects/group12/task.md
echo "HW2 TODO items" | sudo tee /srv/nfsroot/shared/projects/HW2/todo.txt
echo "Alice notes" | sudo tee /srv/nfsroot/users/alice/notes.txt
echo "Bob plan" | sudo tee /srv/nfsroot/users/bob/plan.txt
echo "Charlie secret" | sudo tee /srv/nfsroot/users/charlie/secret.txt
echo "<h1>Hello Web</h1>" | sudo tee /srv/nfsroot/services/webdata/index.html
echo "Nightly snapshot" | sudo tee /srv/nfsroot/services/backups/snapshot.txt

```

執行 `tree /srv/nfsroot` 顯示整個目錄與檔案結構

```

[classuser@vm01:~$ tree /srv/nfsroot
/srv/nfsroot
├── services
│   ├── backups
│   │   └── snapshot.txt
│   └── webdata
│       └── index.html
├── shared
│   ├── projects
│   │   ├── group12
│   │   │   └── task.md
│   │   └── HW2
│   │       └── todo.txt
│   └── public
│       └── README.txt
└── users
    ├── alice
    │   └── notes.txt
    ├── bob
    │   └── plan.txt
    └── charlie
        └── secret.txt

13 directories, 8 files

```

Figure 6: 目錄與檔案結構

### (3) 匯出 NFS Pseudo Root

在 `/etc/exports` 中設定，將 `/srv/nfsroot` 當作 NFSv4 pseudo root

```

/srv/nfsroot *(rw,fsid=0,no_subtree_check,no_root_squash,crossmnt)

```

重新套用設定

```

sudo exportfs -rav

```

```
sudo systemctl restart nfs-kernel-server
showmount -e localhost
```

顯示 /srv/nfsroot 已經成功被 export，任何 client (\*) 都可以掛載。

## (4) Client Setup

建立 /etc/auto.master.d/hw2.autofs，指定 autofs 的 map 檔案

```
/mnt/nfs /etc/auto.nfs --timeout=60 --browse
/- /etc/auto.direct
```

設定 /etc/auto.nfs，對應到 pseudo root 下的 shared 與 users，/mnt/nfs/public、/mnt/nfs/projects 等目錄就會自動掛載對應的 NFS 子目錄

```
public -fstype=nfs4,ro localhost:/shared/public
projects -fstype=nfs4,rw localhost:/shared/projects
alice -fstype=nfs4,rw localhost:/users/alice
bob -fstype=nfs4,rw localhost:/users/bob
charlie -fstype=nfs4,rw localhost:/users/charlie
```

設定 /etc/auto.direct，掛載沒有共同母目錄的服務，/webdata 和 /backups 可以直接透過 autofs 觸發掛載

```
/webdata -fstype=nfs4,ro localhost:/services/webdata
/backups -fstype=nfs4,rw localhost:/services/backups
```

建立掛載點並重啟 autofs

```
sudo mkdir -p /mnt/nfs
sudo mkdir -p /webdata /backups
sudo /etc/init.d/autofs reload
```

```
classuser@vm01:~$ sudo mkdir -p /mnt/nfs
sudo mkdir -p /webdata /backups
sudo /etc/init.d/autofs reload # 或 : sudo systemctl restart autofs
Reloading autofs configuration (via systemctl): autofs.service.
```

Figure 7: 重啟成功

## (5) Auto-mount Proof

一開始檢查，只有 autofs 守門員在等待觸發

```
mount | grep -E '/mnt/nfs|/webdata|/backups'
```

顯示 type=autofs，沒有 type=nfs4

```
[classuser@vm01:~$ mount | grep -E '/mnt/nfs|/webdata|/backups' | echo "目前尚未掛載" ]
/etc/auto.nfs on /mnt/nfs type autofs (rw,relatime,fd=5,pgrp=200794,timeout=60,minproto=5,maxproto=5,indirect,pipe_ino=1340314)
/etc/auto.direct on /webdata type autofs (rw,relatime,fd=10,pgrp=200794,timeout=300,minproto=5,maxp
roto=5,direct,pipe_ino=1340326)
/etc/auto.direct on /backups type autofs (rw,relatime,fd=10,pgrp=200794,timeout=300,minproto=5,maxp
roto=5,direct,pipe_ino=1340326)
```

Figure 8: mount 輸出

使用 ls 或 cat 這些目錄時，autofs 會自動掛載真正的 NFS

```
cat /mnt/nfs/public/README.txt
cat /mnt/nfs/projects/group<id>/task.md
cat /mnt/nfs/projects/HW2/todo.txt
cat /mnt/nfs/alice/notes.txt
cat /mnt/nfs/bob/plan.txt
cat /mnt/nfs/charlie/secret.txt
cat /webdata/index.html
cat /backups/snapshot.txt
```

正確讀取檔案內容，說明 autofs 自動觸發掛載

```
classuser@vm01:~$ ls /mnt/nfs/public
[cat /mnt/nfs/public/README.txt
README.txt
Welcome to public
classuser@vm01:~$ ls /mnt/nfs/projects
[ls /mnt/nfs/projects/group12
cat /mnt/nfs/projects/HW2/todo.txt
group12 HW2
task.md
HW2 TODO items
classuser@vm01:~$ ls /mnt/nfs/alice
[cat /mnt/nfs/alice/notes.txt
notes.txt
Alice notes
classuser@vm01:~$ ls /webdata
[cat /webdata/index.html
index.html
<h1>Hello Web</h1>
classuser@vm01:~$ ls /backups
[cat /backups/snapshot.txt
snapshot.txt
Nightly snapshot
```

Figure 9: autofs 自動觸發掛載

## (6) 存取後掛載情況

再次檢查

```
mount | grep -E '/mnt/nfs|/webdata|/backups'
findmnt -t nfs4
```

結果顯示 localhost:/... type nfs4 的條目，表示 NFS 檔案系統已經掛載上來



```

[classuser@vm01:~]$ findmnt -t nfs4
TARGET SOURCE FSTYPE OPTIONS
/mnt/nfs/public
    localhost:/shared/public
    nfs4 ro,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,time
/mnt/nfs/bob
    localhost:/users/bob
    nfs4 rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,time
/mnt/nfs/charlie
    localhost:/users/charlie
    nfs4 rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,time
/webdata
    localhost:/services/webdata
    nfs4 ro,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,time
/backups
    localhost:/services/backups
    nfs4 rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,time

```

Figure 10: NFS 自動掛載完成

等待超過 60 秒（或 direct map 的 300 秒）後再檢查

```
mount | grep -E '/mnt/nfs|webdata|backups' || echo "已自動卸載"
```

顯示只剩 autofs 控制點，NFS 掛載自動消失，符合 autofs 的設計