

問題一：列出所有的 btrfs 的 sub-volume (sudo btrfs subvolume list /)

(1)運行結果

```
~$ sudo btrfs subvolume list /
ID 257 gen 118256 top level 5 path @
ID 258 gen 118242 top level 5 path @home
```

(2)sub-volume 意思

sub-volume 是 file system 的一部分，擁有自己獨立的檔案結構，可共享文件範圍，跟 LVM logical volume(用 block-level snapshot)不同，是 file extent-based；snapshot 也是一種 sub-volume，但是具有 original sub-volume 的原始內容。

問題二：請問/etc/fstab 內各個欄位的意思為何？(sudo vim /etc/fstab)

(1)運行結果

```
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda2 during installation
UUID=a4c86924-9bdb-418b-b55b-9dfd61ab80a6 / btrfs defaults,subvol=@ 0 1
# /boot/efi was on /dev/sda1 during installation
UUID=516E-E8FB /boot/efi vfat umask=0077 0 1
# /home was on /dev/sda2 during installation
UUID=a4c86924-9bdb-418b-b55b-9dfd61ab80a6 /home btrfs defaults,subvol=@home 0 2
# /home/ext4 was on /dev/sda3 during installation
UUID=c4042916-3eb0-407d-aa47-87757417d13a /home/ext4 ext4 defaults 0 2
# swap was on /dev/sda4 during installation
UUID=9cd42d3d-915b-4eff-a4c9-86dd1f9fa288 none swap sw 0 0
```

(2)欄位意思

順序	欄位	描述
1	device	設備的描述，也可以是 UUID。
2	mount point	掛載的目錄。
3	filesystem	該磁區的檔案格式，例如：ext2，vfat，ntfs 等等
4	parameters	檔案格式參數區。
5	dump	是否可以被 dump 指令備份， 0：不要做備份； 1：要做備份； 2：要做備份，重要度比 1 小。
6	fsck	是否於開機時以 fsck 檢驗磁區， 0：不檢驗；1：先檢驗；2：後檢驗。

Note (YOUTUBE 筆記)：

QUID mount 比較好，每顆硬碟也自己的代號，重整線路後仍可重新開機。

問題三：tar 後 a.time 和 b.tme 的內容一樣嗎？

(1)運行結果 不一樣，因 tar 會有時間差

Name	Size	Modified
a.time	57.8 MB	09:21
b.time	58.5 MB	09:22
clocktime	8.9 kB	09:17
clocktime.c	814 bytes	09:15

Note (YOUTUBE 筆記)：


有可能差很多，若 tar 需要一小時會差一小時


問題四：snapshot 備份後解壓縮以後 a.time 和 b.tme 的內容一樣嗎？（可以使用檔案管理員解開即可）



(1)運行結果 一樣，因 snapshot 時間會同時

<

>



Location:  /snapshot/home-2019-09-25-09:27:58/ubuntu/OS_HW/HW1/

Name	Size	Type	Modified
 a.time	135.2 MB	unknown	25 September 2019, 09:27
 b.time	135.2 MB	unknown	25 September 2019, 09:27

Note (YOUTUBE 筆記)：

Snapshot 是奈秒等級，filesystem 仿佛凍結，對目標目錄做快速紀錄，若有快照進行檔案更新，會額外建立一個檔案，原始檔案與更新檔案共享大部分的資料，有一部分資料沒有共享

缺點：記憶體不連續配置，需跳兩次位址，對於硬碟和快取速度會降低

解決方案：btrfs 可以後來把分開的部分變成連續



問題五：參考上面文章，或其他網路文章，說明資料庫的
online backup 和 offline backup 的差別

(1)比較

項目	online backup	offline backup
特點	可允許 client 進來操作並且同時備份，被開啟的資料通常是副本，雖然無法確保被寫入的時間順序完全一致，但是數據庫恢復的過程中，修改容易保持一致	當備份時，雖然 DB2 還是開啟的，所有 client 端無法會連進來，確保所有資料時間一致
優點	1. 透過加密保護資料 2. 可以異地備份 3. 防止物理偷竊與自然災害 4. 便宜(每月 5 至 12 美元) 5. 專業 IT 團隊管理	1. 快速備份和恢復，因硬體傳輸快 2. 實作簡單，USB 即可備份 3. 攜帶性高 4. 不透過網路更安全
缺點	1. 需要透過雲端，無法直接控制資料 2. 網路若故障，備份跟恢復速度急遽下降 3. 上傳與恢復透過網路速度慢	1. 容易被物理損壞(被水用到) 2. 通常未加密，容易被偷就被破解散播 3. 有老化跟壞掉的風險

Note (YOUTUBE 筆記)：

online backup：邊用邊備份 例：Oracle

offline backup：要停機備份例：MySQL

問題六：如果使用 Btrfs 的話，可以在不停機的情況下備份 MySQL 嗎？(申論題，無對錯)

- (1) 可以達到類似效果，可先用 btrfs 的 snapshot 的方式在不鎖定任何表的情況下進行快照備份，然後啟動 MySQL 的 Binary Log 的機制，紀錄下從此開始的所有動作，若遇到當機可以從 snapshot 的位置透過 Binary Log 記下的所有操作來進行還原，但是會有時間差的問題。
示意圖：



另外，Mysql 官方有提供 mysqldump 備份機制，針對 InnoDB tables 可以不鎖住任何 table 的狀況下進行 online backup，只要使用 --single-transaction 選項即可。

來源網址：<https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/backup-methods.html>

Note (YOUTUBE 筆記)：

要思考 MySQL 記憶體資料突然不見會怎麼樣

問題七：上述三個指令的時間

- (1) `time cp --reflink=always a.time a1.time`

時間不一樣，雖然一開始用 `--reflink=always` 複製是共用同一份檔案，但是 `a.time` 不斷被修改更新，會把 `a.time` 更新的內容存下來，但是會有記憶體碎裂的問題

Name	Size	Modified
a.time	20.5 MB	11:23
a1.time	8.6 MB	11:22

(2)time cp a.time a1.time

時間不一樣，僅複製當下指令的檔案，並且拆成兩份不同的檔案，
a.time 不斷更新，a1.time 是當下指令的檔案，故不一樣。

```
~/OS_HW/HW1 ➤ time cp a.time a1.time oslab
cp a.time a1.time 0.00s user 0.03s system 32% cpu 0.106 total
```

Name	Size	Modified
a.time	19.2 MB	15:27
a1.time	6.8 MB	15:26

(3)time ln a.time a2.time

時間一樣，因為這是軟連結，其實兩個是相同的檔案，若修改其中一個，
另一個也會更動，因為是 link 同一份檔案。

```
~/OS_HW/HW1 ➤ time ln a.time a2.time lab
zsh: correct 'a2.time' to 'a.time' [nyae]? n
ln a.time a2.time 0.00s user 0.00s system 67% cpu 0.002 total
```

a.time	31.1 MB	15:30
a2.time	31.1 MB	15:30
a.time	45.8 MB	15:31
a2.time	45.8 MB	15:31

問題八：a.time 是否會隨之改變內容

(1)vim a1.time /*任意修改 a1.time*/

不會，因為--reflink=always 複製是共用同一份檔案，但是後面的修改是各自儲存的。

<pre>+ a1.time i love OS course. i love OS course. i love OS course.</pre>	<pre>a.time time=2019-09-25 15:29:17.078311523 time=2019-09-25 15:29:17.092866166 time=2019-09-25 15:29:17.093019537 time=2019-09-25 15:29:17.093093848 time=2019-09-25 15:29:17.093164807 time=2019-09-25 15:29:17.093235487</pre>
--	---

問題九：a.time 是否會隨之改變內容

(2)vim a2.time /*任意修改 a1.time*/

會，因為是軟連結，兩個其實是一樣的檔案，會造成修改其中一個，另一個一起變動。

+ a2.time	a.time
i love OS course.	i love OS course.
i love OS course.	i love OS course.
i love OS course.	i love OS course.
time=2019-09-25 15:29:17.078311523	time=2019-09-25 15:29:17.078311523
time=2019-09-25 15:29:17.092866166	time=2019-09-25 15:29:17.092866166

Note (YOUTUBE 筆記)：

Vim.rc 建立使用者 每個人都會複製一份，用 reflink=always，若讀取 a.time 和 a1.time 是讀取同一份檔案，若修改 a.time，a1.time 不會被更改，類似快照，修改的檔案存在外面，但是會碎裂，需額外資源處理。

延伸閱讀 ZFS Note (YOUTUBE 筆記)：

ZFS 伺服器等級用的檔案系統

家庭用 EXT4(快照功能依賴管理員較弱)

BTRFS 學 ZFS 不到 10 年，太新還需時間穩定