

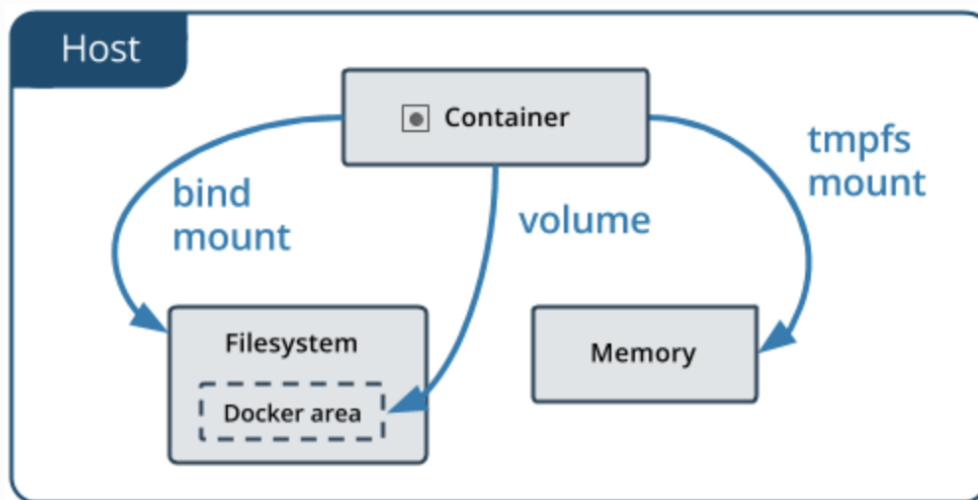
Docker 儲存

默认情况下，在运行中的容器里创建的文件，被保存在一个可写的容器层：

- 如果容器被删除了，则数据也没有了
- 这个可写的容器层是和特定的容器绑定的，也就是这些数据无法方便的和其它容器共享

Docker主要提供了两种方式做数据的持久化

- Data Volume, 由Docker管理，(/var/lib/docker/volumes/ Linux), 持久化数据的最好方式
- Bind Mount，由用户指定存储的数据具体mount在系统什么位置



--mount 與 -v/ --volume

當在使用 docker create 或是 docker run 的指令時，會有兩個選項可以使用，個別是 --mount 以及 -v (或是 --volume)。

注意，千萬不要誤以為使用 -v 就是用 Volume 的方式或者是 --mount 就是使用 Bind Mount 的方式，不是這樣的！！我們先來看看這兩個在 docker create 與 docker run 中的說明是甚麼：

```
-v, --volume list    Bind mount a volume
--mount mount        Attach a filesystem mount to the container
```

但其實是可以利用 --mount 或 -v 來綁定資料到 Volume，也可以用 --mount 或 -v 來綁定資料到 Filesystem。既然都可以，那為什麼要分兩種呢？

這是因為 --mount 其實出現的比較早，但他設定綁定資料的方式比較完整卻麻煩，因此在後續又出現了 -v 的選項來簡化綁定的設定過程。

Data Volume

相關指令

```
# 建立volume
# docker volume create [OPTIONS] [VOLUME]
$docker volume create my-vol

#使用 -v
$ docker run -d --name my-http -v my-vol:/usr/local/apache2 -p 8080:80 httpd

#使用 --mount
```

```
$ docker run -d --name my-http --mount source=my-vol,target=/usr/local/apache2 -p 8080:80 httpd
```

```
# 列出所有volume
[tomcat@docker]$ docker volume ls
DRIVER          VOLUME NAME
local           2db2d2b1a6ceb9f578d2149dc44625e1996dcc33de0639f8644ce2082182b9a8
local           my-vol

# 查詢某volume細節
[tomcat@docke]$ docker volume inspect 2db2d2b1a6ceb9f578d2149dc44625e1996dcc33de0639f8644ce2082182b9a8
[
  {
    "CreatedAt": "2019-06-04T15:40:43+08:00",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/2db2d2b1a6ceb9f578d2149dc44625e1996dcc33de0639f8644ce2082182b9a8/_data",
    "Name": "2db2d2b1a6ceb9f578d2149dc44625e1996dcc33de0639f8644ce2082182b9a8",
    "Options": null,
    "Scope": "local"
  }
]

[tomcat@docke]$ docker volume inspect my-vol
[
  {
    "CreatedAt": "2023-02-07T02:11:18Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/my-vol/_data",
    "Name": "my-vol",
    "Options": {},
    "Scope": "local"
  }
]

# 刪除所有不使用的volume
[tomcat@docker]$ docker volume prune
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
2db2d2b1a6ceb9f578d2149dc44625e1996dcc33de0639f8644ce2082182b9a8
44d6286d1a793ac8c788b0fd364fa8d9430ce3ec079731dbe894dd996cae9adb
925559f4c2399c0c989d8602ca77fb3243dfb83022fe029b09b130c9521e7f89
9484acbd73c53af47f29add893c5125b0cd7f0f96e521c54e643f1aeb8f97eeb
b47202da45060de8f67e48058966b32078d525cadd1c94671a0ffe33f3d5df4b
006d620ae77bf7530ea37abd48b9c20ac264956f4b9ef3397e2c5072f32677c2
15e0fd2e575c2c24a2a137ff420dbf30326f3859d483561efc45e568d294c802
98b5385b04a401e9e870c5a1e807d69c56a1764b2ec91baedd9c13994cb2c863
ffdae0e84d00966afdc0df55d664d06ee99c14d249ad7558848088c464d472ff

Total reclaimed space: 77.8MB

# 指定volume mount 至 /var/www/html
[tomcat@docker]$ docker run -d -v html_doc:/var/www/html httpd
```

Bind Mount to FileSystem

```
# 直接mount 外部資料夾 至 /var/www/html
# 相對路徑
[tomcat@docker-qc html-doc]$ docker run -d -v ${PWD}/html-doc:/var/www/html httpd
# 絕對路徑
[tomcat@docker-qc html-doc]$ docker run -d -v /home/tomcat/docker/html-doc:/var/www/html httpd
```

Bind Mount to Memory

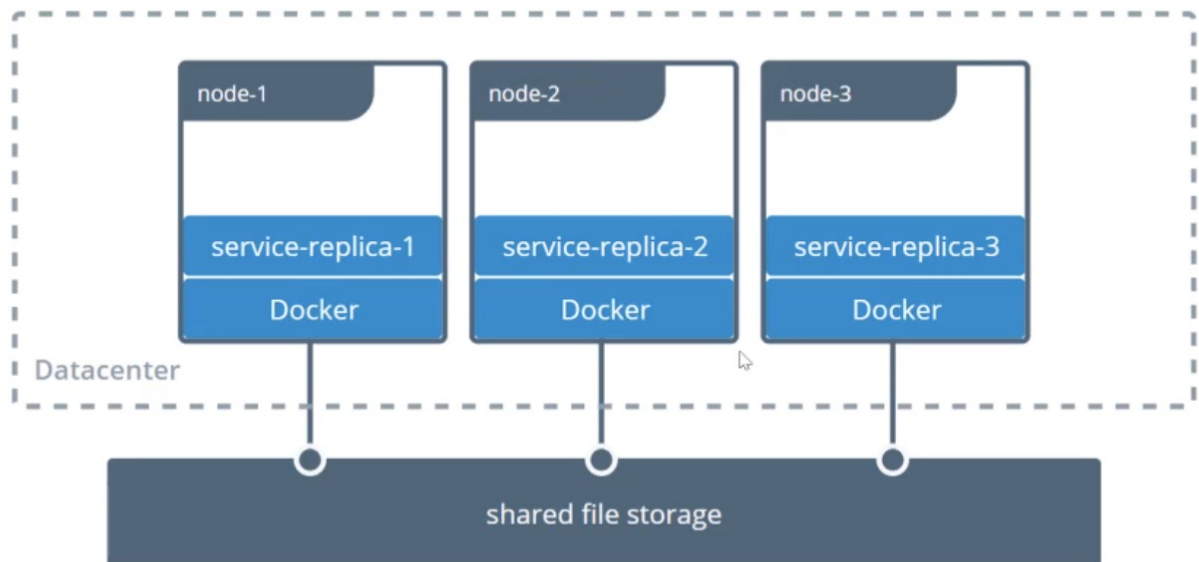
Tmpfs，全名 Temporary File System，是在 Linux 暫存檔儲存空間的常見名稱，通常以掛載檔案系統方式實現。他可以將資料儲存在記憶體中而不是存在硬碟內，所以只要一重新開機，資料則會被抹除。

Tmpfs 如其名，如果使用 Tmpfs Mount 則資料會存在於記憶體上，在重啟 Container 的時候就會 Reset 資料。

特別注意的是，使用 Tmpfs Mount 並不是使用 `-v` 的選項，而是使用 `--tmpfs`，`--tmpfs` 就如同 `-v` 一樣簡化了使用 `--mount` 時所需的設定。

```
使用 --tmpfs
$ docker run -d --name my-http --tmpfs /usr/local/apache2/htdocs -p 8080:80 httpd
使用 --mount
$ docker run -d --name my-http --mount type=tmpfs,destination=/usr/local/apache2/htdocs -p 8080:80 httpd
```

多个机器之间的容器共享数据



官方参考链接 <https://docs.docker.com/storage/volumes/#share-data-among-machines>

官方参考链接 <https://docs.docker.com/storage/volumes/#share-data-among-machines>

常用volume mapping (log)

```
# nginx
docker run -p 80:80 -itd -v /tmp/log/nginx/docker-nginx-log:/var/log/nginx nginx
```

使用綁定的不同時機比較

既然 Docker 提供了這麼多的方式讓我們儲存資料，那麼我們應該在甚麼情況下使用哪一種方式來儲存呢？

- 使用 Volume 的時機

- 想要多個 Container 中共用資料的時候

- 在無法確保存放資料路徑或結構的時候
- 當 Docker 不在本機端而是在雲端或遠端的時候
- 當有需要資料的備份、還原、轉移的時候
- 當資料有大量 I/O 的情況，Docker 建議你可以把資料放在 Docker VM 內，會有更好的性能表現
- 當你的應用程式需要完整且原生的 File System 的時候

- 使用 Bind Mount 的時機

- 想要將 Config 設定檔同時分享給 Container 與本機端的時候
- 想要同時分享程式碼或其他來源文件給 Container 與本機端的時候
- 當非常確定所綁定的路徑底下，其資料夾與結構等等都非常確定且保持一致不會改變的時候

- 使用 Tmpfs Mount 的時機

- 不想讓資料被永久儲存的時候
- 需要每次重啟都有乾淨的環境時
- 當想要讀寫性能非常好時，因為它掛載的空間是在記憶體上

[Docker筆記 - 讓資料遠離Container，使用 Volume、Bind Mount 與 Tmpfs Mount](#)

🕒修訂版本 #22

★由 treeman 建立於 18 🕒@🕒🕒 2022 22:47:25

✍由 treeman 更新於 5 🕒🕒🕒🕒 2023 10:14:58