Nguyễn Thái Dương – duongtn1512@gmail.com

## Exerise day 4

Task 1: Create a named volume called "data_volume" and mount it to a container. Verify that data is persisted even after the container is removed.

Dockerfile U | powershell | compose.yaml ...\Nginx_t

docker > Day_04 > task1 > Dockerfile > ...

```
1   FROM node:16-alpine3.17 AS build
2   WORKDIR /usr/app
3   COPY package.json ./
4   RUN npm install
5   COPY . .
6   RUN npm run build
7
8   FROM nginx:1.22 AS deploy
9   WORKDIR /app
10  COPY --from=build /usr/app/dist /app/
11  COPY nginx.conf /etc/nginx/nginx.conf
12  CMD [ "nginx", "-g", "daemon off;" ]
13
```

```
PS D:\Devops_FPT_Foudations> docker volume ls
PS D:\Devops_FPT_Foudations> docker volume ls
PS D:\Devops_FPT_Foudations> docker volume ls
DRIVER    VOLUME NAME
local     a001
local     a002
local     a003
PS D:\Devops_FPT_Foudations> docker volume inspect a001
[
    {
        "CreatedAt": "2023-08-11T12:40:28Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/a001/_data",
        "Name": "a001",
        "Options": null,
        "Scope": "local"
    }
]
PS D:\Devops_FPT_Foudations>
```

Using dockerfile we create image duongtn1512/devops_fpt_learn:day2_task3 and use volume a001

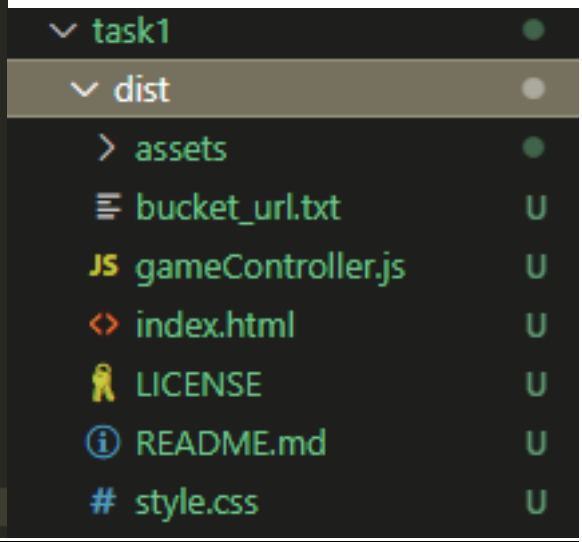
From docker compose we mount a001 to new image we created and name container bb1

The file in dist folder will get put into folder app in container bb1 folow docker file in deploy stage

compose.yaml U | Dockerfile U | powershell | pow

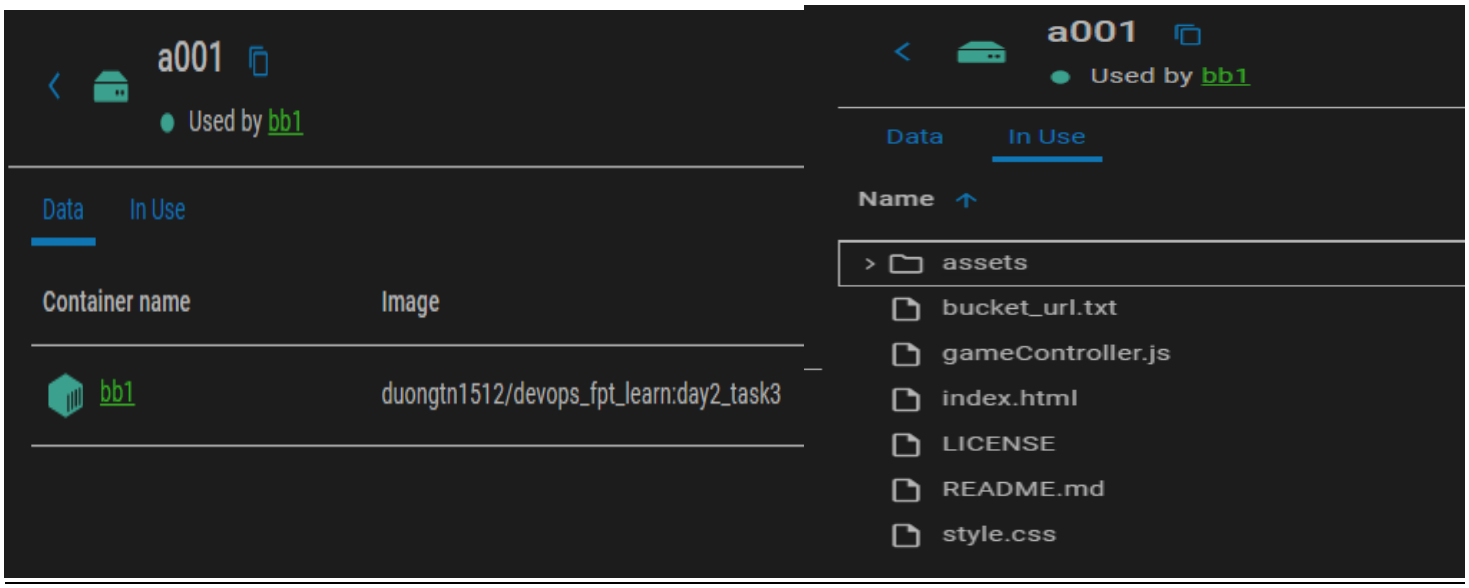
docker > Day_04 > task1 > compose.yaml

```
1   version: '3'
2   services:
3
4     chess:
5       image: duongtn1512/devops_fpt_learn:day2_task3
6       container_name: bb1
7       volumes:
8         - a001:/app
9
10  volumes:
11    a001:
12      external: true
13
```




```
PS D:\Devops_FPT_Foudations\docker\Day_04\task1> docker compose up -d
[+] Running 2/2
 ✓ Network task1_default  Created
 ✓ Container bb1          Started
PS D:\Devops_FPT_Foudations\docker\Day_04\task1>
```

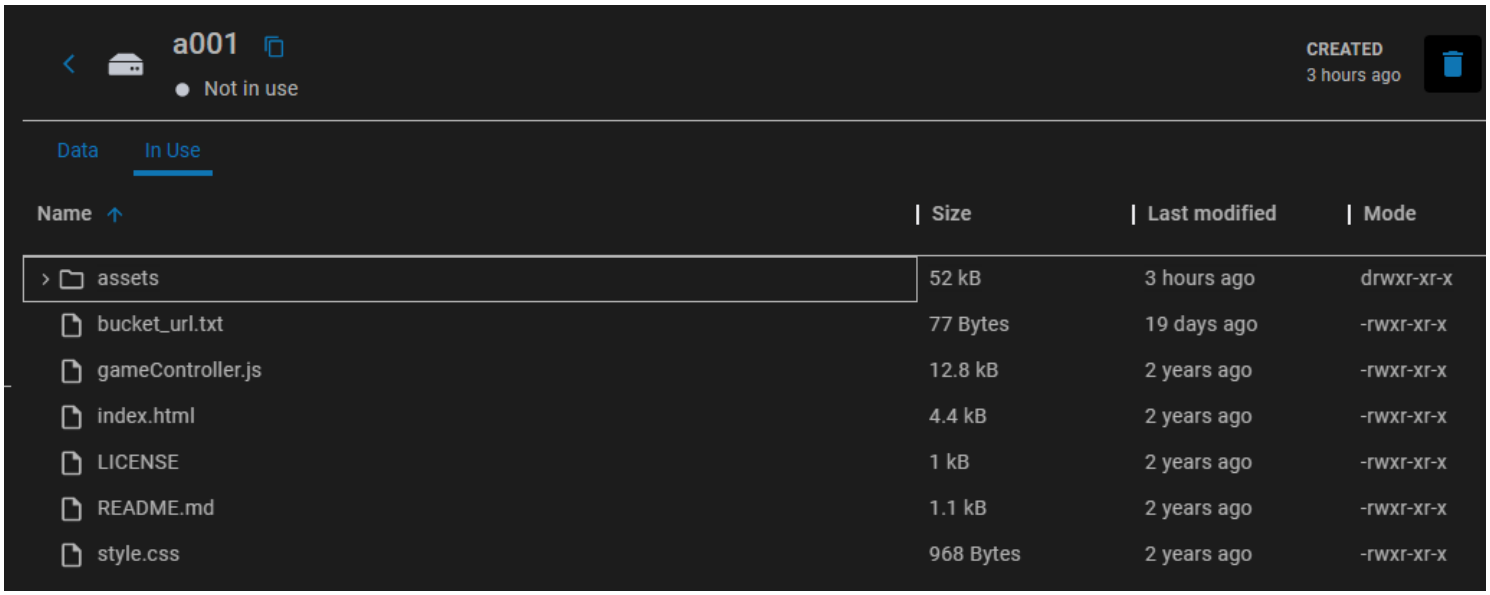
The file in app folder will get mount to storage a001




We delete container bb1

```
PS D:\Devops_FPT_Foudations\docker\Day_04\task1> docker compose up -d
[+] Running 2/2
 ✓ Network task1_default  Created
 ✓ Container bb1          Started
PS D:\Devops_FPT_Foudations\docker\Day_04\task1> docker compose down
[+] Running 2/2
 ✓ Container bb1          Removed
 ✓ Network task1_default  Removed
PS D:\Devops_FPT_Foudations\docker\Day_04\task1>
```

And then we check data in storage a001 when not in use by any docker containers

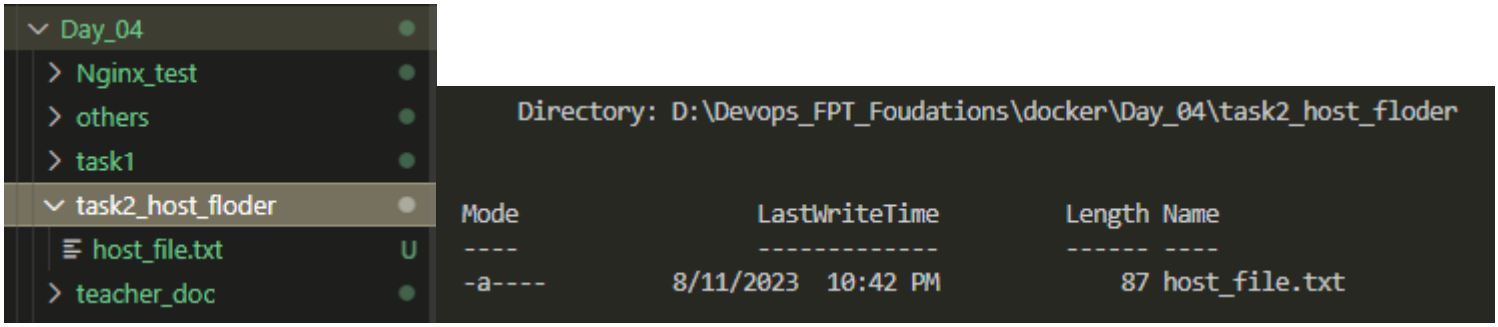



Its still here.

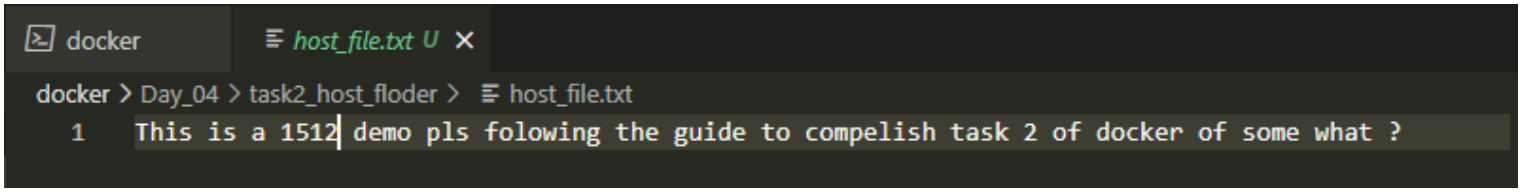
Task 2: Mount a directory from your host machine to a container using a bind mount. Modify files from both the host and the container to observe changes.

Prepare Directory and Files:

With host folder is task2_host_floder and host file is host_file.txt




Data in file host_file.txt




Create an contaner name cc1 with bind mount  task2_host_floder to it contaner path /app:

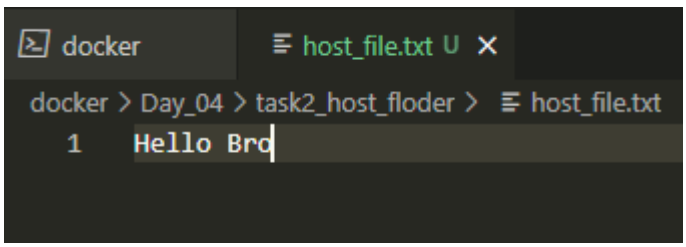
```
docker run -it --name cc1 -v D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder:/app nginx bash
```

We then enter container and check for the file host_file.txt

```
root@8940c4f8a4f1:/# ls
app  boot  docker-entrypoint.d   etc   lib    lib64   media  opt   root  sbin  sys  usr
bin  dev   docker-entrypoint.sh  home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@8940c4f8a4f1:/# ls app
host_file.txt
root@8940c4f8a4f1:/# cd app
root@8940c4f8a4f1:/app# cat host_file.txt
This is a 1512 demo pls folowing the guide to compelish task 2 of docker of some what ?r
```

We had success bring file from windows host to docker nginx container

Modify from Host




And then we enter container to see if it change the txt file

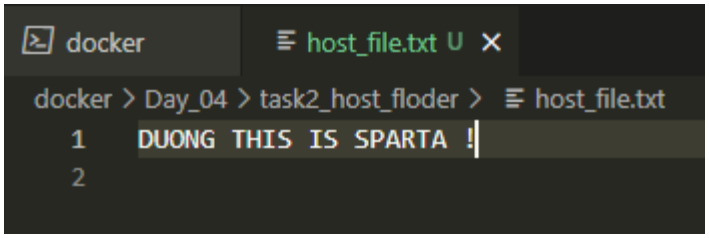
```
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker exec -it cc1 bash
root@8940c4f8a4f1:/# ls
app  boot  docker-entrypoint.d   etc   lib    lib64   media  opt   root  sbin  sys  usr
bin  dev   docker-entrypoint.sh  home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@8940c4f8a4f1:/# cd app
root@8940c4f8a4f1:/app# ls
host_file.txt
root@8940c4f8a4f1:/app# cat host_file.txt
Hello Broroot@8940c4f8a4f1:/app#
```

Yes it dose change the data in host_file.txt

Modify from Container

```
root@8940c4f8a4f1:/app# echo "DUONG THIS IS SPARTA !" > host_file.txt
root@8940c4f8a4f1:/app# cat host_file.txt
DUONG THIS IS SPARTA !
root@8940c4f8a4f1:/app#
```

And in the host is changed




So both ways are ok and can make change to the file its bind mouth

Task 3: Use the docker volume inspect command to view metadata and configuration details of a volume.

```
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker volume ls
DRIVER    VOLUME NAME
local     a001
local     a002
local     a003
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker volume inspect a001
[
    {
        "CreatedAt": "2023-08-11T12:40:28Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/a001/_data",
        "Name": "a001",
        "Options": null,
        "Scope": "local"
    }
]
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder>
```

Task 4: Identify and remove volumes that are no longer in use to free up storage space.

```
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker ps -a
CONTAINER ID   IMAGE                                    COMMAND                  CREATED              STATUS              PORTS     NAMES
a555a88d1b26   duongtn1512/devops_fpt_learn:day2_task3   "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp    bb1
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker volume ls
DRIVER    VOLUME NAME
local     a001
local     a002
local     a003
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker inspect bb1 | Select-String "Mounts" -Context 3

            "CpuPercent": 0,
            "IOMaximumIOps": 0,
            "IOMaximumBandwidth": 0,
>           "Mounts": [
                {
                    "Type": "volume",
                    "Source": "a001",
            },
            "Name": "overlay2"
        },
>       "Mounts": [
            {
                "Type": "volume",
                "Name": "a001",


PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder>
```

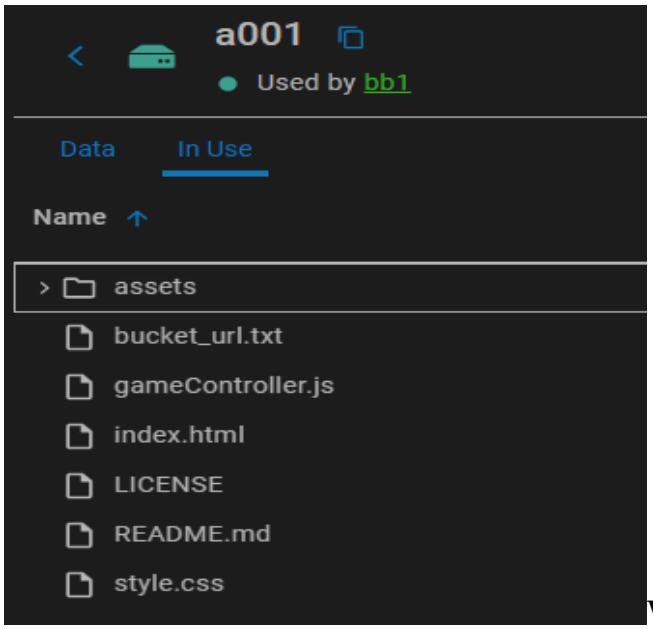
So from the image I get, I belive that only volume a001 in used with container bb1

We rm volume a002 and a003 because it not in use

```
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker volume rm a002 a003
a002
a003
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder> docker volume ls
DRIVER    VOLUME NAME
local     a001
PS D:\Devops_FPT_Foudations\docker\Day_04\task2_host_floder>
```

Task 5: Backup the contents of a volume to your local machine and then restore it to a new volume.

Fist we use volume a001 created before in task 1




We use this comand too backup at host windows

```
docker run --rm -v a001:/data -v C:\local\backup\path:/backup nginx tar cvf /backup/backup.tar /data
```

Check path we backup data C:\local\backup\path using ls command

```
PS C:\local\backup\path> docker run --rm -v a001:/data -v C:\local\backup\path:/backup nginx tar cvf /backup/backup.tar /data
tar: Removing leading `/' from member names
/data/
/data/LICENSE
/data/README.md
/data/assets/
/data/assets/white_king.png
/data/assets/black_rook.png
/data/assets/white_pawn.png
/data/assets/black_bishop.png
/data/assets/black_knight.png
/data/assets/black_king.png
/data/assets/white_bishop.png
/data/assets/white_rook.png
/data/assets/black_pawn.png
/data/assets/black_queen.png
/data/assets/white_knight.png
/data/assets/white_queen.png
/data/style.css
/data/gameController.js
/data/bucket_url.txt
/data/index.html
PS C:\local\backup\path> ls


    Directory: C:\local\backup\path


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         8/12/2023   1:29 AM          61440 backup.tar
```

As we can see we have store volume a001 data to file backup.tar at C:\local\backup\path

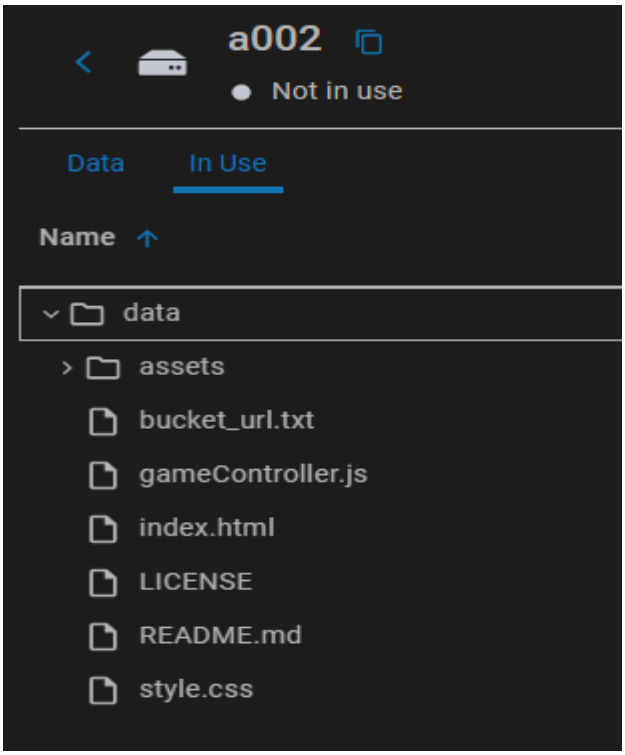
For the restore the data to another volume, we create a new volume and bring data to that volume

```
docker volume create a002

docker run --rm -v a002:/data -v C:\local\backup\path:/backup nginx tar xvf /backup/backup.tar -C /data
```

We inspect the data from docker desktop

```
PS C:\local\backup\path> docker run --rm -v a002:/data
data/
data/LICENSE
data/README.md
data/assets/
data/assets/white_king.png
data/assets/black_rook.png
data/assets/white_pawn.png
data/assets/black_bishop.png
data/assets/black_knight.png
data/assets/black_king.png
data/assets/white_bishop.png
data/assets/white_rook.png
data/assets/black_pawn.png
data/assets/black_queen.png
data/assets/white_knight.png
data/assets/white_queen.png
data/style.css
data/gameController.js
data/bucket_url.txt
data/index.html
```




Task 6: Create a container with a tmpfs mount to store temporary data in memory. Observe how the data is lost when the container stops.

We create a nginx container  temporary mount it at path /data inside container using this command

```
docker run  -itd --mount type=tmpfs,target=/data nginx
```

We make a test.txt file echo in it “Hello, temporary data!” and save file within peaceful_kalam

```
PS D:\Devops_FPT_Foudations> docker run  -itd --mount type=tmpfs,target=/data nginx
a69c2d3f1843e337c04a63f22d094c9964c0ebb5e14d4b0580873ab04e174373
PS D:\Devops_FPT_Foudations> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
a69c2d3f1843   nginx     "/docker-entrypoint.…"   58 seconds ago   Up 57 seconds   80/tcp    peaceful_kalam
PS D:\Devops_FPT_Foudations> docker exec -it peaceful_kalam bash
root@a69c2d3f1843:/# ls
bin   data  docker-entrypoint.d   etc   lib    lib64   media  opt   root  sbin sys  usr
boot  dev   docker-entrypoint.sh  home  lib32  libx32  mnt    proc  run   srv  tmp  var
root@a69c2d3f1843:/# cd data
root@a69c2d3f1843:/data# ls
root@a69c2d3f1843:/data# echo "Hello, temporary data!" > test.txt
root@a69c2d3f1843:/data# cat test.txt
Hello, temporary data!
root@a69c2d3f1843:/data#
```

Using same command we create another container with docker name it thirsty_nightingale

```
PS D:\Devops_FPT_Foudations> docker run  -itd --mount type=tmpfs,target=/data nginx
c7210a3007f01d2e55248a2ced429f7a5f680f6beb65bbe381da2a70431ac8aa
PS D:\Devops_FPT_Foudations> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
c7210a3007f0   nginx     "/docker-entrypoint.…"   7 seconds ago   Up 6 seconds   80/tcp    thirsty_nightingale
a69c2d3f1843   nginx     "/docker-entrypoint.…"   7 minutes ago   Up 7 minutes   80/tcp    peaceful_kalam
PS D:\Devops_FPT_Foudations> docker exec -it thirsty_nightingale bash
root@c7210a3007f0:/# ls
bin   data  docker-entrypoint.d   etc   lib    lib64   media  opt   root  sbin sys  usr
boot  dev   docker-entrypoint.sh  home  lib32  libx32  mnt    proc  run   srv  tmp  var
root@c7210a3007f0:/# ls data
root@c7210a3007f0:/#
```

As you see we enter container with same type of tmpfs mount and in data folder there is no test.txt

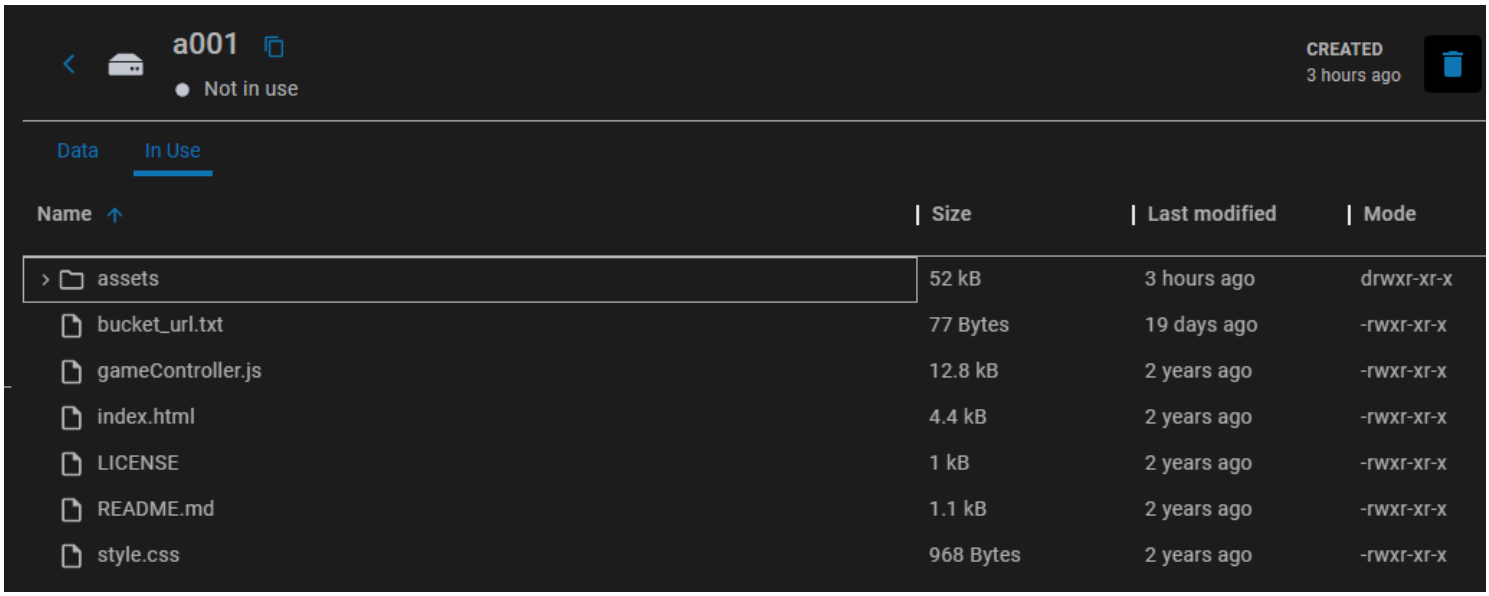
We stop container peaceful_kalam and restart it again

```
PS D:\Devops_FPT_Foudations> docker stop peacefull_kalam
Error response from daemon: No such container: peacefull_kalam
PS D:\Devops_FPT_Foudations> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
c7210a3007f0   nginx     "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes    80/tcp    thirsty_nightingale
a69c2d3f1843   nginx     "/docker-entrypoint.…"   12 minutes ago   Up 12 minutes   80/tcp    peaceful_kalam
PS D:\Devops_FPT_Foudations> docker stop peaceful_kalam
peaceful_kalam
PS D:\Devops_FPT_Foudations> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                       PORTS     NAMES
c7210a3007f0   nginx     "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes                 80/tcp    thirsty_nightingale
a69c2d3f1843   nginx     "/docker-entrypoint.…"   12 minutes ago   Exited (0) 17 seconds ago              peaceful_kalam
PS D:\Devops_FPT_Foudations> docker start peaceful_kalam
peaceful_kalam
PS D:\Devops_FPT_Foudations> docker exec -it peaceful_kalam bash
root@a69c2d3f1843:/# ls
bin   data  docker-entrypoint.d   etc   lib    lib64   media  opt   root  sbin sys  usr
boot  dev   docker-entrypoint.sh  home  lib32  libx32  mnt    proc  run   srv  tmp  var
root@a69c2d3f1843:/# ls data
root@a69c2d3f1843:/#
```

And then enter container, the file test.txt we make has alredy gone. This is how tmpfs mount work

Task 7: Write a docker-compose.yml file that defines a service using volumes, then launch multiple containers to share data.

Fist off we will use volume a001 to use with multiple container that will share eachother same data



Here is a docker compose file that will make 3 nginx container with it data will be take from volume a001 and then store in /usr/share/nginx/html folder inside container to make an frontend web

docker > Day_04 > Nginx_test > compose.yaml

```yaml
version: '3'
services:
  nginx1:
    image: nginx:latest
    container_name: nginx-1
    ports:
      - "1000:80"
    volumes:
      - a001:/usr/share/nginx/html
    command: [ "nginx", "-g", "daemon off;" ]
  nginx2:
    image: nginx:latest
    container_name: nginx-2
    ports:
      - "1001:80"
    volumes:
      - a001:/usr/share/nginx/html
    command: [ "nginx", "-g", "daemon off;" ]
  nginx3:
    image: nginx:latest
    container_name: nginx-3
    ports:
      - "1002:80"
    volumes:
      - a001:/usr/share/nginx/html
    command: [ "nginx", "-g", "daemon off;" ]
volumes:
  a001:
    external: true
```

We docker compose up -d to make them run and curl each of nginx port to see result

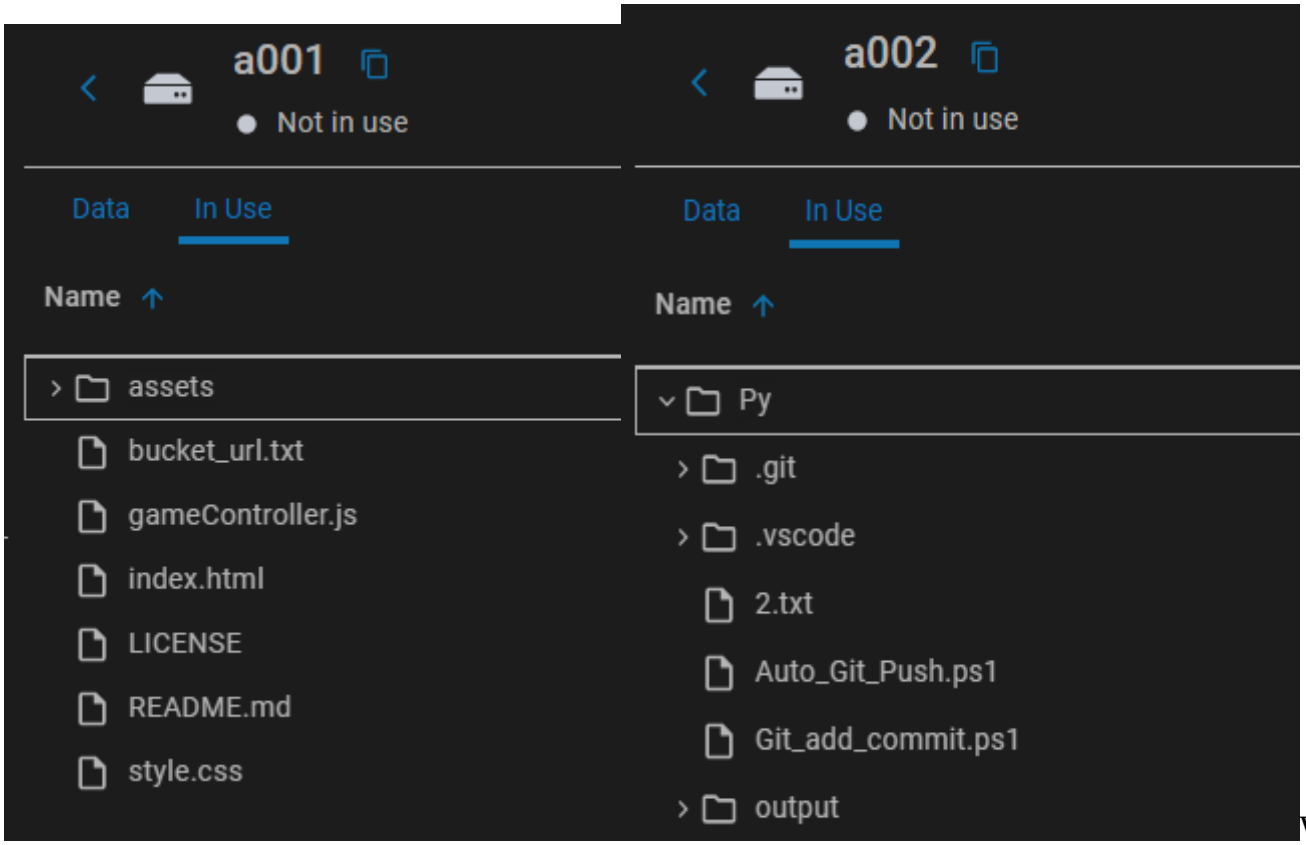
```
PS D:\Devops_FPT_Foudations\docker\Day_04\Nginx_test> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS D:\Devops_FPT_Foudations\docker\Day_04\Nginx_test> docker compose up -d
[+] Running 4/4
 ✓ Network nginx_test_default  Created
 ✓ Container nginx-1           Started
 ✓ Container nginx-2           Started
 ✓ Container nginx-3           Started
PS D:\Devops_FPT_Foudations\docker\Day_04\Nginx_test> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                    NAMES
b19c7cc7a8e7   nginx:latest   "/docker-entrypoint.…"   8 seconds ago   Up 5 seconds   0.0.0.0:1002->80/tcp     nginx-3
8c841ea23d64   nginx:latest   "/docker-entrypoint.…"   8 seconds ago   Up 5 seconds   0.0.0.0:1001->80/tcp     nginx-2
19a554d1386e   nginx:latest   "/docker-entrypoint.…"   8 seconds ago   Up 5 seconds   0.0.0.0:1000->80/tcp     nginx-1
PS D:\Devops_FPT_Foudations\docker\Day_04\Nginx_test>
```

So all 3 port take same resource from volume a001 to make web chess game and expose to port

Localhost: 1000, 1001, 1002


Task 8: Launch a container that uses multiple volumes for different parts of its filesystem.

We will use 2 volume a001 and a002 with them own distinctive data




We then run

```
docker run -d --name multi_nginx -v a001:/app1 -v a002:/app2 nginx
```

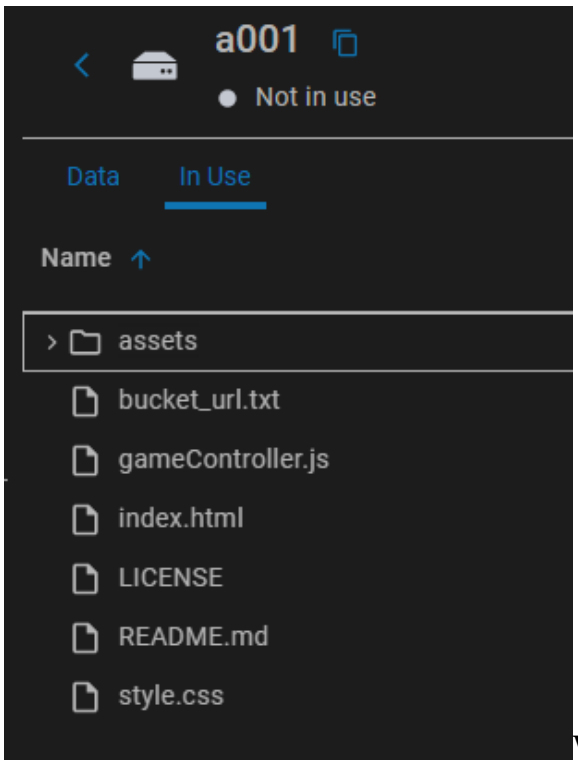
This command line will fist pull and run image nginx:latest to a container name multi_nginx and attach data from volume a001 to app1 folder in container, so on with data from a002 to app2 floder

We then enter container and check if the data from volume a001 and a002 has successfully deploy inside container multi_nginx folder app1, app2.

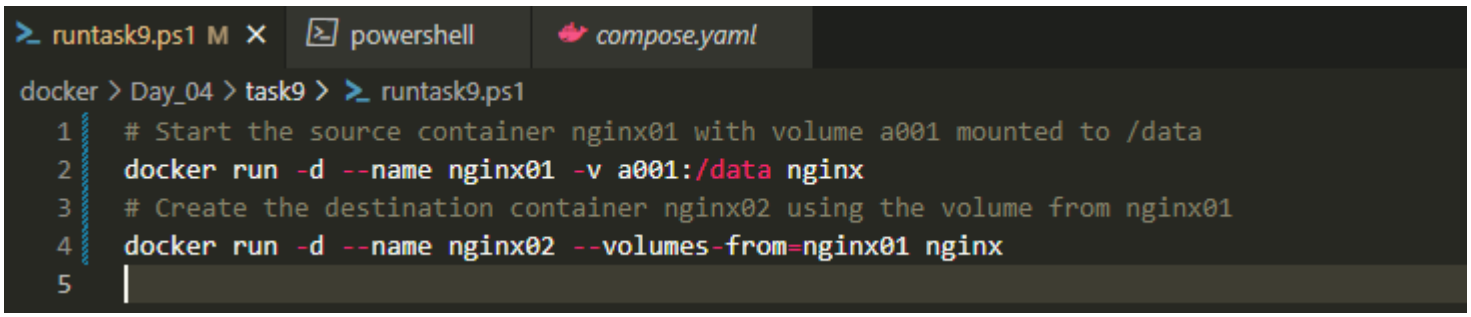
```
PS D:\Devops_FPT_Foudations> docker run -d --name multi_nginx -v a001:/app1 -v a002:/app2 nginx
40ee3e34b37940bfbdee3bc498b359f485f83c6ed7fc0047302a94868de7bdbf
PS D:\Devops_FPT_Foudations> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
40ee3e34b379   nginx     "/docker-entrypoint.…"   9 seconds ago   Up 7 seconds   80/tcp    multi_nginx
PS D:\Devops_FPT_Foudations> docker exec -it multi_nginx bash
root@40ee3e34b379:/# ls
app1  bin   dev                  docker-entrypoint.sh  home  lib32  libx32  mnt  proc  run   srv  tmp  var
app2  boot  docker-entrypoint.d  etc                   lib   lib64  media   opt  root  sbin  sys  usr
root@40ee3e34b379:/# ls app1
LICENSE  README.md  assets  bucket_url.txt  gameController.js  index.html  style.css
root@40ee3e34b379:/# cd app2
root@40ee3e34b379:/app2# ls Py
2.txt              Py_blank.py        Python_Learn_Day2_3_4        Self-trace-history.txt  output
Auto_Git_Push.ps1  Python_Fast        Python_Learn_Day5_and_Beyone  Self-trace.txt
Git_add_commit.ps1 Python_Learn_Day1  README.md                     User_add_to_Git.ps1
root@40ee3e34b379:/app2#
```

Task 9: Create two containers, migrate data from one to the other using volumes, and ensure minimal downtime.

We fist will using volume a001 mount it to new container we will create nginx01




We then using volume-from to let new container take data




We name our new container nginx02 using volume from nginx01 the path to data will be the same when we create nginx01 and mount it to a001

```
runtask9.ps1 M | powershell | compose.yaml
PS D:\Devops_FPT_Foudations\docker\Day_04\task9> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS D:\Devops_FPT_Foudations\docker\Day_04\task9> ./runtask9
f3506d40b281393e8f62c8f52c1fd43b064e8eae828f0ad220f8a7f526565baf
290d03dd809390a8d362584b75ea06310911fabafd5e63a42a8f872e257eb410
PS D:\Devops_FPT_Foudations\docker\Day_04\task9> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
290d03dd8093   nginx     "/docker-entrypoint.…"   4 seconds ago   Up 3 seconds   80/tcp    nginx02
f3506d40b281   nginx     "/docker-entrypoint.…"   5 seconds ago   Up 4 seconds   80/tcp    nginx01
PS D:\Devops_FPT_Foudations\docker\Day_04\task9> docker exec -it nginx02 bash
root@290d03dd8093:/# ls
bin   data  docker-entrypoint.d   etc   lib    lib64   media  opt   root  sbin
ys  usr
boot  dev   docker-entrypoint.sh  home  lib32  libx32  mnt    proc  run   srv
mp  var
root@290d03dd8093:/# ls data
LICENSE  README.md  assets  bucket_url.txt  gameController.js  index.html  style.css
root@290d03dd8093:/#
exit
```

Folow the result we have success transfer data floder from nginx01 mount a001 to nginx02

Task 10: Launch multiple instances of a container and share data using the same volume between them.

```
runtask10.ps1 U | powershell | exercises.md U | runtask9.ps1 U | Dockerfile U
docker > Day_04 > task10 > runtask10.ps1
1  # Create volume that will get shared
2  docker volume create a004
3  # Create 2 nginx container mount to volume a004
4  docker run -d --name nginx01 -v a004:/shared-data nginx
5  docker run -d --name nginx02 -v a004:/shared-data nginx
6  # Wirte text to volume a004 from nginx01 and check result at nginx02
7  docker exec -it nginx01 sh -c "echo 'Hello from Nginx01 Bro u got chose' > /shared-data/demo.txt"
8  docker exec -it nginx02 cat /shared-data/demo.txt
```

We make volume a004 and shared it with 2 continer nginx01 and nginx02

We run the script to auto report us if we have success share data using the volume a004

```
runtask10.ps1 U | powershell | exercises.md U | runtask9.ps1 U
PS D:\Devops_FPT_Foudations\docker\Day_04\task10> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS D:\Devops_FPT_Foudations\docker\Day_04\task10> ls


    Directory: D:\Devops_FPT_Foudations\docker\Day_04\task10


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         8/13/2023   6:58 PM            446 runtask10.ps1


PS D:\Devops_FPT_Foudations\docker\Day_04\task10> ./runtask10
a004
8a10c394f4221cb8c07bed863878aeb9e658b20ba74731e24beac03179e7cea2
3722ede5310013beb90aa212d8c6752959b4c59f658bea5166ff2fe951861e9b
Hello from Nginx01 Bro u got chose
PS D:\Devops_FPT_Foudations\docker\Day_04\task10>
```

As nginx02 cat out text we make from nginx01 to volume a004 I said we has finish task 10