Docker Volume

1. Bind Mount

Use a local folder as a mount point for Bind Mount
Start a container as an Nginx web server, mapping the local folder to Nginx folder inside container.

Stage 1

I created a folder on my local machine. I cloned the website repo inside the folder by using ‘git clone https://github.com/talfik2/javascript-projects'

```
muhammedumit@Umits-MacBook ~ % mkdir website && cd website
muhammedumit@Umits-MacBook website % ls
muhammedumit@Umits-MacBook website % git clone https://github.com/talfik2/javascr
ipt-projects


Cloning into 'javascript-projects'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), 71.06 KiB | 1.82 MiB/s, done.
Resolving deltas: 100% (1/1), done.
muhammedumit@Umits-MacBook website % ls
javascript-projects
muhammedumit@Umits-MacBook website %
```

I change the directory into reviews folder where the index.html and other files are

```
muhammedumit@Umits-MacBook website % cd javascript-projects
muhammedumit@Umits-MacBook javascript-projects % ls
README.md       counter         reviews
muhammedumit@Umits-MacBook javascript-projects % cd reviews
muhammedumit@Umits-MacBook reviews % ls
app.js          index.html      person-1.jpeg   styles.css
muhammedumit@Umits-MacBook reviews %
```

Stage 2

I run the Nginx container in detached mod, name it as web_server, map the local folder to the Nginx folder inside the container

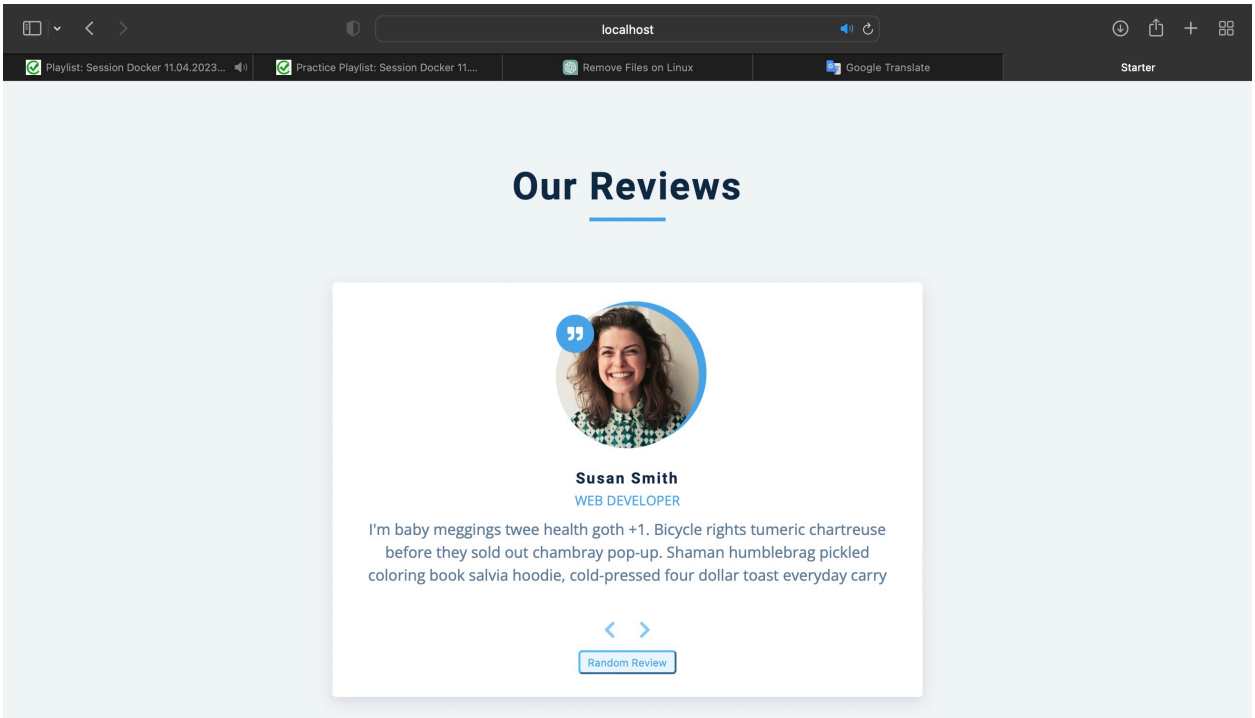
docker run -d --name web_server --rm -p 8080:80 --mount type=bind,source=/local folder which is my local ‘/Users/muhammedumit/website/javascript-projects/reviews’target=/usr/share/nginx/html nginx:alpine

-d as in detached mode

-p 8080:80 as in publish local port and container port
—rm delete the container after using
—mount as in match, type is bind, source is local host which is local folder, target is the folder structure that Nginx publishes inside of the container. With this code I match the container with my local host.

```
muhammedumit@Umits-MacBook reviews % pwd
/Users/muhammedumit/website/javascript-projects/reviews
muhammedumit@Umits-MacBook reviews % docker run -d --name web_server --rm -p 8080
:80 --mount type=bind,source=/Users/muhammedumit/website/javascript-projects/revi
ews,target=/usr/share/nginx/html nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
edb6bdbacee9: Pull complete
e7b9f8564496: Pull complete
e6a8c530945d: Pull complete
a31273f49e64: Pull complete
4fbf2e3beac8: Pull complete
467b5f1f71f7: Pull complete
3a6c132398b3: Pull complete
fca8748d6c5b: Pull complete
Digest: sha256:2d194184b067db3598771b4cf326cfe6ad5051937ba1132b8b7d4b0184e0d0a6
Status: Downloaded newer image for nginx:alpine
4f224c083c3e45ec9de946f179db97dca34ea4b8c41ce9b3e90a6b79a68d9e4c
muhammedumit@Umits-MacBook reviews % docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
       PORTS                  NAMES
4f224c083c3e   nginx:alpine   "/docker-entrypoint.…"   22 seconds ago   Up 21 sec
onds   0.0.0.0:8080->80/tcp   web_server
```

In the local host <http://localhost:8080/>, we can see the Nginx website published in my local machine in seconds.



2. Volume

Let's start with run ubuntu by using 'docker run -it —rm ubuntu'

```
muhammedumit@Umits-MacBook reviews % docker run -it --rm ubuntu
root@b87989eb4776:/#
```

I created 50 files by using the code 'touch myfile{1..50}.txt'

```
root@b87989eb4776:/# touch myfile{1..50}.txt
root@b87989eb4776:/# ls
bin           myfile13.txt  myfile24.txt  myfile35.txt  myfile46.txt  root
boot          myfile14.txt  myfile25.txt  myfile36.txt  myfile47.txt  run
dev           myfile15.txt  myfile26.txt  myfile37.txt  myfile48.txt  sbin
etc           myfile16.txt  myfile27.txt  myfile38.txt  myfile49.txt  srv
home          myfile17.txt  myfile28.txt  myfile39.txt  myfile5.txt   sys
lib           myfile18.txt  myfile29.txt  myfile4.txt   myfile50.txt  tmp
media         myfile19.txt  myfile3.txt   myfile40.txt  myfile6.txt   usr
mnt           myfile2.txt   myfile30.txt  myfile41.txt  myfile7.txt   var
myfile1.txt   myfile20.txt  myfile31.txt  myfile42.txt  myfile8.txt
myfile10.txt  myfile21.txt  myfile32.txt  myfile43.txt  myfile9.txt
myfile11.txt  myfile22.txt  myfile33.txt  myfile44.txt  opt
myfile12.txt  myfile23.txt  myfile34.txt  myfile45.txt  proc
root@b87989eb4776:/#
```

After exit, my 50 txt files removed.

```
root@b87989eb4776:/# exit
exit
muhammedumit@Umits-MacBook reviews % docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
       PORTS                  NAMES
4f224c083c3e   nginx:alpine   "/docker-entrypoint.…"   23 minutes ago   Up 23 min
utes   0.0.0.0:8080->80/tcp   web_server
muhammedumit@Umits-MacBook reviews % docker images
REPOSITORY       TAG       IMAGE ID       CREATED       SIZE
ubuntu           latest    37f74891464b   5 weeks ago   69.2MB
nginx            alpine    66bf2c914bf4   7 weeks ago   41MB
jpetazzo/clock   latest    d4c3ce7fdfdb   2 years ago   1.4MB
muhammedumit@Umits-MacBook reviews % docker run -it --rm ubuntu
root@d35703cb2c19:/# ls
bin   dev  home  media  opt   root  sbin  sys  usr
boot  etc  lib   mnt    proc  run   srv   tmp  var
root@d35703cb2c19:/#
```

In order to create a volume the code is docker volume firstvolume

```
muhammedumit@Umits-MacBook reviews % docker volume create firstvolume
firstvolume
muhammedumit@Umits-MacBook reviews % docker volume ls
DRIVER    VOLUME NAME
local     firstvolume
muhammedumit@Umits-MacBook reviews %
```

In order to see its features, I used docker volume inspect firstvolume code;

```
muhammedumit@Umits-MacBook reviews % docker volume inspect firstvolume
[
    {
        "CreatedAt": "2023-08-06T15:17:40Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/firstvolume/_data",
        "Name": "firstvolume",
        "Options": null,
        "Scope": "local"
    }
]
muhammedumit@Umits-MacBook reviews %
```