Docker Hands-on Task

Objective

You will build and run a simple Docker-based application while applying different network modes, volume types, and a basic Dockerfile.

Part 1 – Basic Dockerfile

1. Create a directory named `docker_task`.

2. Inside it, create a file named `Dockerfile` that:

  - Uses **`alpine`** as the base image.

  - Installs `curl`

  - Runs `cat hello from container` when the container starts.

3. Build the image and tag it as `my-basic-image:v1`.

Part 2 – Volumes

You must demonstrate **three types of Docker volumes**:

1. Bind Mount

- Create a local directory `data_bind` and put a file `bind_note.txt` inside it.

- Run the container so that `/app/data` inside the container is linked to `data_bind` on your local machine.
- Verify that changes in the container are reflected locally.

mkdir data_bind
touch bind_note.txt
sudo docker run -d --name t1 -v ./data_bind:/app/data nginx

sudo docker exec -it t1 /bin/bash
cd /app/data
touch myfile
exit
cd data_bind/
ls

```
tasneem@DESKTOP-0VT5601:~$ ls
backup.tar  index.html    my_local_image  task   tasneem.pem                 test.txt
data_bind   my_image.tar  mytest_vol      task1  tasneem.pem:Zone.Identifier
tasneem@DESKTOP-0VT5601:~$ ls data_bind/
bind_note.txt
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d --name t1 -v ./data_bind:/app/data nginx
[sudo] password for tasneem:
cfb02cb48d4fdff01142ee8069820a6d198628d6563f79f9e009c80b25ffb7c5
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it t1  /bin/bash
root@cfb02cb48d4f:/# ls /app/data
bind_note.txt
```

```
root@cfb02cb48d4f:/# cd /app/data
root@cfb02cb48d4f:/app/data# touch myfile
root@cfb02cb48d4f:/app/data# exit
exit
tasneem@DESKTOP-0VT5601:~$ cd data_bind/
tasneem@DESKTOP-0VT5601:~/data_bind$ ls
bind_note.txt  myfile
tasneem@DESKTOP-0VT5601:~/data_bind$
```

2. Named Volume

- Create a named volume called `my_named_volume`.

- Run the container using this named volume mounted at `/app/named`.

- Create a file inside `/app/named` from inside the container and check it persists after container deletion.

sudo docker run -d --name t2 -v my_named_volume:/app/named nginx
sudo docker exec -it t2 /bin/bash
touch /app/named/file1
exit
sudo su
cd /var/lib/docker/volumes/my_named_volume/_data
ls

```
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d --name t2 -v my_named_volume:/app/named nginx
[sudo] password for tasneem:
ce5a2657f86a43ed9ae922799eeeb927f273d426510c46c64797eec196076b92
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it t2  /bin/bash
root@ce5a2657f86a:/# touch file1
root@ce5a2657f86a:/# ls /app/named
root@ce5a2657f86a:/# touch /app/named/file1
root@ce5a2657f86a:/# ls /app/named
file1
root@ce5a2657f86a:/# exit
```

```
tasneem@DESKTOP-0VT5601:~$ sudo su
root@DESKTOP-0VT5601:/home/tasneem# cd /var/lib/docker
root@DESKTOP-0VT5601:/var/lib/docker# cd volumes
root@DESKTOP-0VT5601:/var/lib/docker/volumes# ls
2f541c6be9223a4240a87e3c2ecd623c7b2987cb78f7ef9b1b1fd4052443bb6c  6aa4d60ab0cd5796bbd19a22440a2b611fc1b7475d90b73675cd5f8b2bc44d0c  backingFsBlockDev  metadata.db  my_named_volume  shared_vol  test_vol1
root@DESKTOP-0VT5601:/var/lib/docker/volumes# cd my_named_volume
root@DESKTOP-0VT5601:/var/lib/docker/volumes/my_named_volume# ls
_data
root@DESKTOP-0VT5601:/var/lib/docker/volumes/my_named_volume# cd _data
root@DESKTOP-0VT5601:/var/lib/docker/volumes/my_named_volume/_data# ls
file1
root@DESKTOP-0VT5601:/var/lib/docker/volumes/my_named_volume/_data#
```

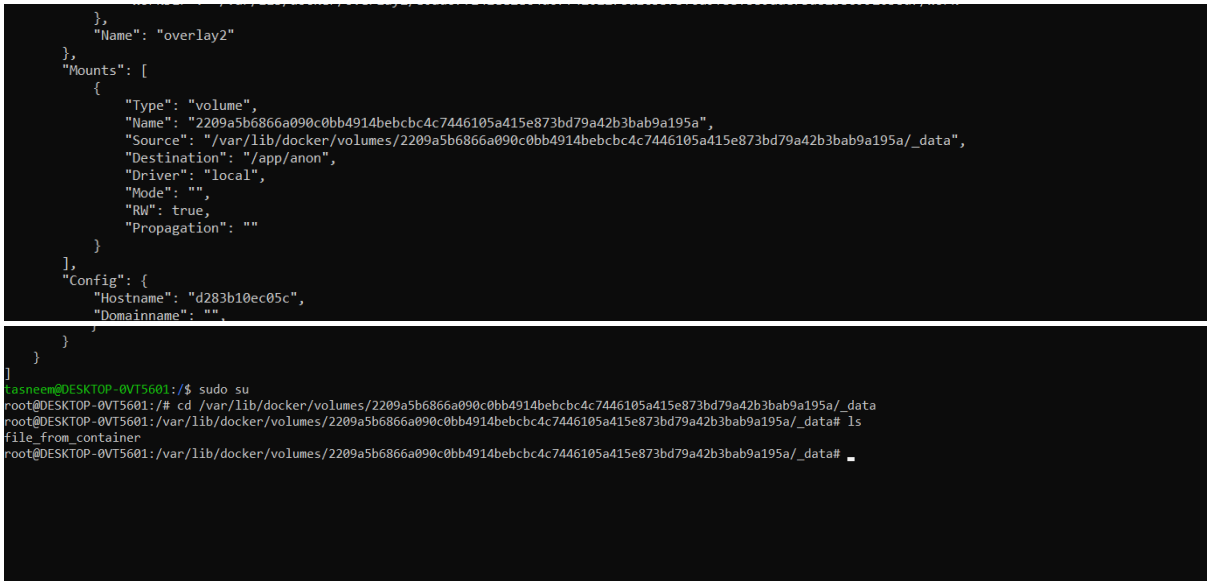
3. Anonymous Volume

- Run the container with an anonymous volume mounted at `/app/anon`.

- Verify the anonymous volume is created by listing all volumes after the container starts.

sudo docker run -d -v /app/anon nginx
sudo docker exec -it d283b /bin/bash
touch /app/anon/file_from_container
exit
sudo docker inspect d283b
sudo su
cd
/var/lib/docker/volumes/2209a5b6866a090c0bb4914bebcbc4c7446105a415e873bd79a42b3bab9a195a/_data
ls

```
tasneem@DESKTOP-0VT5601:/$ sudo docker run -d -v /app/anon nginx
[sudo] password for tasneem:
d283b10ec05cda00c1783bbadae97984a06d5d22f7c0d822521dba05d224dda6
tasneem@DESKTOP-0VT5601:/$ sudo docker exec -it d283b  /bin/bash
root@d283b10ec05c:/# ls /app/anon
root@d283b10ec05c:/# touch /app/anon/file_from_container
root@d283b10ec05c:/# exit
exit
tasneem@DESKTOP-0VT5601:/$ sudo docker volume ls
DRIVER    VOLUME NAME
local     2f541c6be9323a4249a87e3c2ecd623c7b2987cb78f7ef9b1b1fd4052443bb6c
local     6aa4d60ab0cd5796bbd19a22440a2b611fc1b7475d90b73675cd5f8b2bc44d0c
local     2209a5b6866a090c0bb4914bebcbc4c7446105a415e873bd79a42b3bab9a195a
local     my_named_volume
local     shared_vol
local     test_vol1
tasneem@DESKTOP-0VT5601:/$ sudo docker inspect d283b
[
    {
        "Id": "d283b10ec05cda00c1783bbadae97984a06d5d22f7c0d822521dba05d224dda6",
        "Created": "2025-08-12T22:11:38.331444006Z",
        "Path": "/docker-entrypoint.sh",
        "Args": [
            "nginx",
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "running",
```

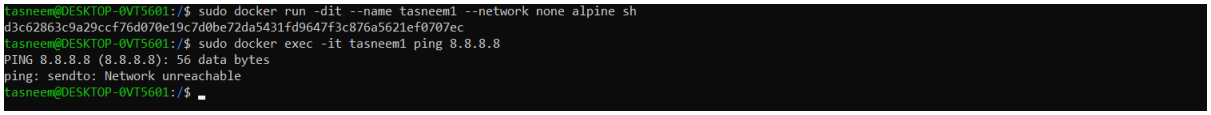


Part 3 – Network Modes

You must run the container with:

1. `network=none`

- Run your image with `--network none` and try to `ping google.com` inside the container (it should fail).
sudo docker run -dit --name tasneem1 --network none alpine sh
sudo docker exec -it tasneem1 ping 8.8.8.8

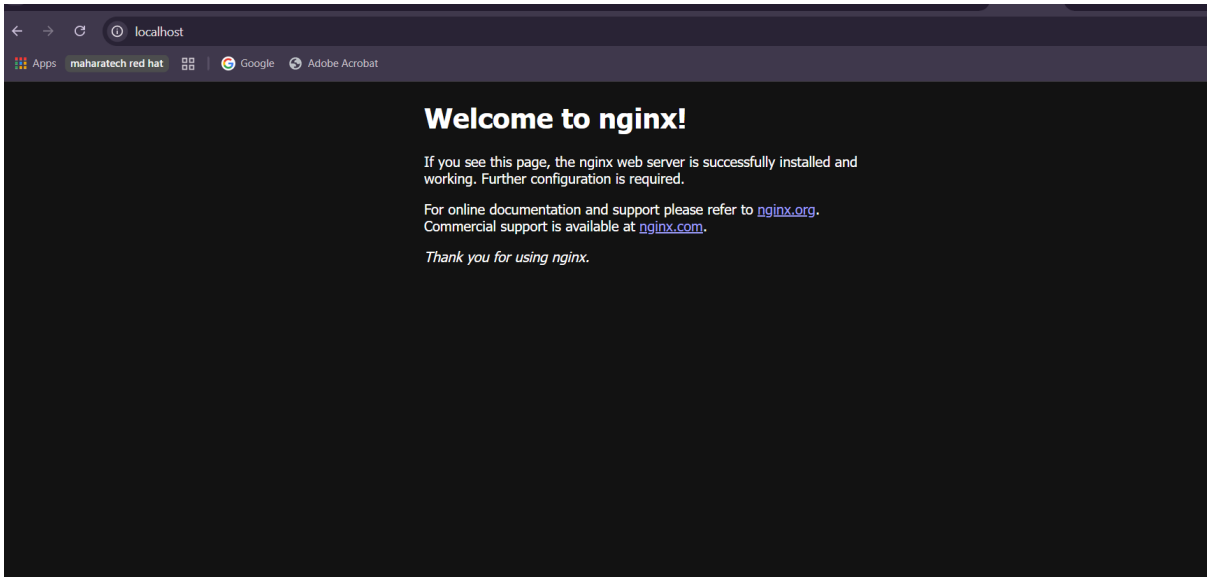


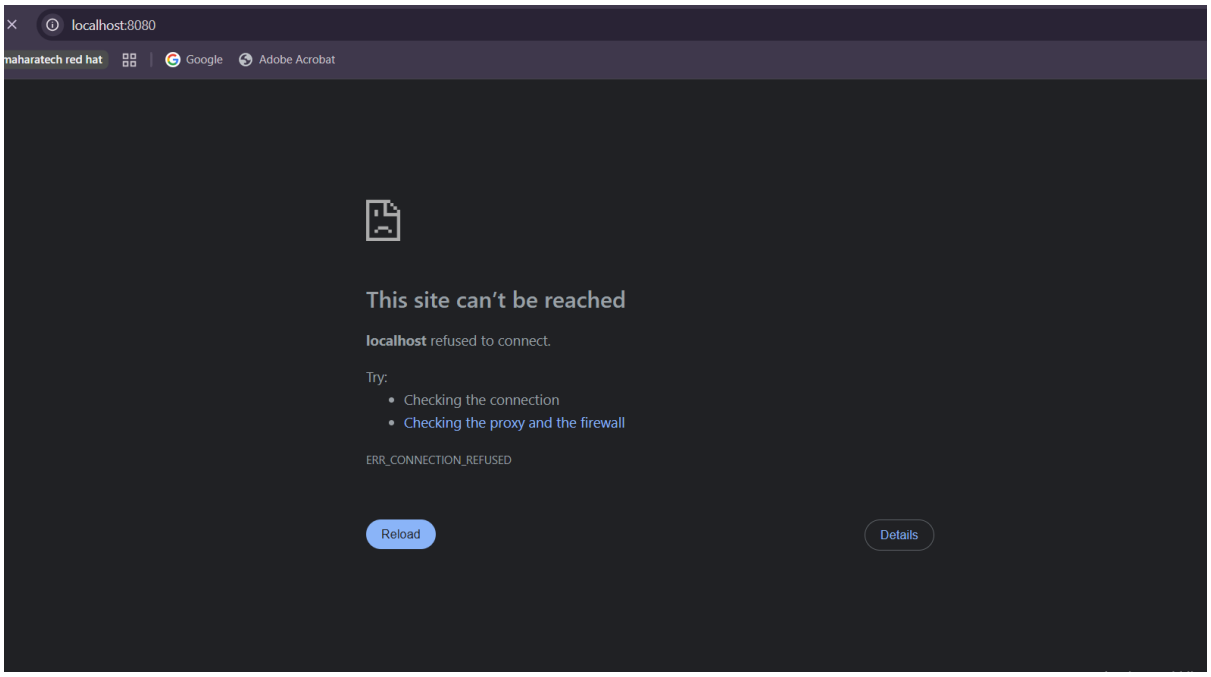
2. `network=host`

- Run a simple web server from your container Use `--network host` and verify you can access the server directly from your host at `http://localhost:8080`.

sudo docker run -d --name tasneem2 --network host nginx

```
tasneem@DESKTOP-0VT5601:/$ sudo docker run -d --name tasneem2 --network host nginx
1f51eaedeae67133f8eee8ea60cf64125ae133fabfe1dca14cb855ea7bbf27df
tasneem@DESKTOP-0VT5601:/$ sudo docker run -dit --name tasneem3 --network host nginx sh
a8d78edf0bd15be07012fe4a5bff0092ae9e634f486dd289bea9751f55575268
tasneem@DESKTOP-0VT5601:/$ sudo docker inspect tasneem2
[
    {
        "Id": "1f51eaedeae67133f8eee8ea60cf64125ae133fabfe1dca14cb855ea7bbf27df",
        "Created": "2025-08-12T22:29:01.671156292Z",
        "Path": "/docker-entrypoint.sh",
        "Args": [
            "nginx",
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "exited",
            "Running": false,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 0,
```





OR: I can use an image that the default port is 8080 like tomcat

sudo docker run -d --name mytomcat --network host tomcat:latest

```
tasneem@DESKTOP-0VT5601:/$ sudo docker run -d --name mytomcat --network host tomcat:latest
Unable to find image 'tomcat:latest' locally
latest: Pulling from library/tomcat
b71466b94f26: Pull complete
f51d8cf18447: Pull complete
7c9e2e04796b: Pull complete
9967391d7d21: Pull complete
9246e450bedd: Pull complete
1db455b62022: Pull complete
4f4fb700ef54: Pull complete
09a0cc3aa763: Pull complete
Digest: sha256:25e357988a706fb8745139d496c43148babd4773dbfbc0afce6dee1c11852ed6
Status: Downloaded newer image for tomcat:latest
941e468b4fedd52c7a2cc3492fb59c92f71ca6926559a89937c35e281fbeddd4
```