

1)

```
C:\python\docker\docker_ht>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
11da671f7fc4	nginx:mainline-alpine	"/docker-entrypoint..."	About a minute ago	Exited (0) 16 seconds ago		nginx_3
f3639a029da7	nginx:mainline-alpine	"/docker-entrypoint..."	About a minute ago	Up About a minute	80/tcp	nginx_2
08464ab1b2dc	nginx:mainline-alpine	"/docker-entrypoint..."	About a minute ago	Up About a minute	80/tcp	nginx_1

2)

```
C:\python\docker\docker_ht>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
C:\python\docker\docker_ht>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

3)

```
C:\python>docker exec -it serene_margulis bash
root@b260c8f9181a:/usr/app# tail -f ./logs.txt
Secret message is:
"Docker is easy"
Thu, 24 Mar 2022 10:17:26 GMT
Thu, 24 Mar 2022 10:17:29 GMT
Thu, 24 Mar 2022 10:17:32 GMT
Thu, 24 Mar 2022 10:17:35 GMT
Secret message is:
"Docker is easy"
```

4)

```
C:\python>docker run -d devopsdockeruh/exec_bash_exercise
b260c8f9181a654639fbffba989947500390fa51b7cf21f1f5c08fd73f06fe5
```

```
1 FROM devopsdockeruh/overwrite_cmd_exercise
2 CMD ["-c"]
```

5)

```
D:\docker\t4>docker build -t docker-sequence .
[+] Building 1.7s (5/5) FINISHED
=> [internal] load build definition from Dockerfile 0.7s
=> => transferring dockerfile: 91B 0.0s
=> [internal] load .dockerignore 0.4s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/devopsdockeruh/overwrite_cmd_exercise:latest 0.0s
=> CACHED [1/1] FROM docker.io/devopsdockeruh/overwrite_cmd_exercise 0.0s
=> exporting to image 0.7s
=> => exporting layers 0.0s
=> => writing image sha256:2e188bf896689ee3cd2c0198e5cdbc2af25827990c10a28b11f137fbb7dc4a26 0.1s
=> => naming to docker.io/library/docker-sequence 0.0s

D:\docker\t4>docker run docker-sequence
1
2
3
4
5
6
7
8
9

D:\docker\t4>docker run -v D:\docker\t5\logs.txt:/usr/app/logs.txt: devopsdockeruh/first_volume_exercise
Wrote to file /usr/app/logs.txt
Wrote to file /usr/app/logs.txt
Wrote to file /usr/app/logs.txt
Wrote to file /usr/app/logs.txt
Wrote to file /usr/app/logs.txt
```

```
D:\docker\t5>docker run -p 80:80 devopsdockeruh/ports_exercise
```

```
> ports_exercise@1.0.0 start /usr/app  
> node index.js
```

Listening on port 80, this means inside of the container. Use -p to map the port to a port of your local machine.

6)

← → ↻ ⓘ localhost

Ports configured correctly!!

7)

The screenshot displays a web application interface with a dark theme. The top navigation bar includes links for 'Upgrade', 'Settings', 'Sign in', and 'Logout'. The main content area is divided into two sections: a left sidebar with a 'PREVIEW' button and a right pane showing server logs.

The logs pane, titled 'loving_engelbart deploy_flask:latest RUNNING', contains the following text:

```
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [01/Apr/2022 21:56:19] "GET /list/city HTTP/1.1" 200 -
172.17.0.1 - - [01/Apr/2022 21:57:10] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [01/Apr/2022 21:57:17] "GET /login HTTP/1.1" 302 -
172.17.0.1 - - [01/Apr/2022 22:05:14] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [01/Apr/2022 22:05:35] "GET /list/city HTTP/1.1" 200 -
```

Below the logs, a browser window is open to 'localhost:5000'. The browser's address bar shows 'localhost:5000'. The page content includes a list of links: [Google Login](#), [About](#), [Weather on the weekend](#), [Weather on this day](#), [User-agent](#), and [Logout](#).