

## # 🧑 Task: Docker Images & Networking (Up to Bridge + Port Forwarding)

### ## 💡 Part 1: Working with Docker Images

#### 1. \*\*Pull the Alpine and Nginx images:\*\*

##### ## ❓ Additional Questions – Docker Images (No Dockerfile / Build)

**sudo docker pull nginx**

**sudo docker pull alpine**

```
tasneem@DESKTOP-0VT5601:~$ sudo docker pull nginx
[sudo] password for tasneem:
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
tasneem@DESKTOP-0VT5601:~$ sudo docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
Digest: sha256:4bcff63911fcbb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1
Status: Image is up to date for alpine:latest
docker.io/library/alpine:latest
tasneem@DESKTOP-0VT5601:~$ -
```

#### 1. \*\*What is the difference between an image and a container?\*\*

► \*Explain in terms of mutability, state, and lifecycle.\*

**Docker image** contains all the libraries, dependencies, and files that the container needs to run. A Docker image is shareable and portable, so you can deploy the same image in multiple locations at once—much like a software binary file.

**Mutability:** immutable template used to create containers and does not change once built.

**State:** It has no runtime state

**lifecycle:** Stored on disk or registry and can be reused in multiple container

**Docker container** is a running (or stopped) instance of a Docker image. It packages the application along with its runtime environment and can be started, stopped, moved, or deleted .

**Mutability:** Ephemeral and modifiable so files, processes, and configurations inside the container can change during runtime.

**State:** Has a runtime state (running, paused, stopped)

**lifecycle: Created from an image, runs in memory until stopped or removed. It can be restarted from the same image**

**2. \*\*What happens if you run ` docker run nginx` twice without removing the first container? Why?\*\***

**Docker will create two separate containers from the same image, each with its own isolated Nginx instance. By default, each container is assigned a random name if no --name is provided.**

```
2025/08/10 19:38:49 [notice] 1#1: worker process 35 exited with code 0
2025/08/10 19:38:49 [notice] 1#1: exit
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d nginx
d292d0e84cc8433926d7cf22aa9dbcc0ca149af4f524cd0559b43807c72f2df
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d nginx
d2916eaa8050f8c0ba795e518bf91b3576d0ed49b6f8753ccfc897bfa8af009a
tasneem@DESKTOP-0VT5601:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d2916eaa8050 nginx "/docker-entrypoint..." 19 seconds ago Up 19 seconds 80/tcp agitated_haslett
df52d0e84cc8 nginx "/docker-entrypoint..." 24 seconds ago Up 23 seconds 80/tcp beautiful_rosalind
tasneem@DESKTOP-0VT5601:~$
```

**3. \*\*Can two containers be created from the same image at the same time? What happens to their file systems?\*\***

**Yes, multiple containers can be created from the same image at the same time. Each container gets its own isolated writable file system layer , read-only image layers. Changes made inside one container's file system do not affect the image or other containers created from it.**

```
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it agitated_haslett sh
# echo "hello" > /usr/share/nginx/html/index.html
# exit
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it beautiful_rosalind sh
# echo "hello" > /usr/share/nginx/html/index.html
# exit
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it agitated_haslett cat /usr/share/nginx/html/index.html
hello
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it beautiful_rosalind cat /usr/share/nginx/html/index.html
hello
```

**4. \*\*What's the difference between ` docker image ls` and ` docker ps` ? When would you use each?\*\***

**docker image ls**

**Shows all the images stored on your machine.**

**Doesn't matter if any container is using them or not.**

**I use it when I want to see what images I have or check if I pulled something.**

**docker ps**

**Shows the containers that are running right now (add -a to see running + stopped ones).**

**I use it when I want to know what containers are up and get details like ports and names.**

```
tasneem@DESKTOP-0VT5601:~$ sudo docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
alpine          latest   9234e8fb04c4  3 weeks ago  8.31MB
nginx           latest   2cd1d97f893f  3 weeks ago  192MB
ubuntu          latest   65ae7a6f3544  3 weeks ago  78.1MB
hello-world     latest   74cc54e27dc4  6 months ago 10.1kB
tasneem@DESKTOP-0VT5601:~$ sudo docker ps
CONTAINER ID    IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
d2916eaa8050   nginx     "/docker-entrypoint..."  20 minutes ago  Up 20 minutes  80/tcp    agitated_haslett
df52d0e84cc8   nginx     "/docker-entrypoint..."  20 minutes ago  Up 20 minutes  80/tcp    beautiful_rosalind
tasneem@DESKTOP-0VT5601:~$
```

**5. \*\*What's the purpose of tagging an image (e.g., `myapp:1.0`)? What happens if you don't specify a tag?\*\***

**Tagging gives an image a specific version and** useful when you update your app but still want to keep older versions instead of coding in files to fix the error.

**If you don't specify a tag: Docker will automatically use the tag latest.**

```
tasneem@DESKTOP-0VT5601:~$ sudo docker pull nginx:1.25
1.25: Pulling from library/nginx
09f376ebb190: Pull complete
a11fc495baf9: Pull complete
933cc8470577: Pull complete
999643392fb7: Pull complete
971bb7f4fb12: Pull complete
45337c09cd57: Pull complete
de3b062c0af7: Pull complete
Digest: sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72af059c
Status: Downloaded newer image for nginx:1.25
docker.io/library/nginx:1.25
tasneem@DESKTOP-0VT5601:~$ sudo docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
alpine          latest   9234e8fb04c4  3 weeks ago  8.31MB
nginx           latest   2cd1d97f893f  3 weeks ago  192MB
ubuntu          latest   65ae7a6f3544  3 weeks ago  78.1MB
hello-world     latest   74cc54e27dc4  6 months ago 10.1kB
nginx           1.25    e784f4560448  15 months ago 188MB
tasneem@DESKTOP-0VT5601:~$
```

**6. \*\*How does Docker know which image to use when you run ` docker run ubuntu` ?\*\***

Docker looks for **ubuntu:latest** **locally** on your machine.

```
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d nginx
77202cac59b49785288665166c45f58c4635c7348cdcb7cdb5b89c53a18fa3a
tasneem@DESKTOP-0VT5601:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
77202cac59b4 nginx "/docker-entrypoint..." 17 seconds ago Up 16 seconds 80/tcp naughty_curie
d2916eaa8050 nginx "/docker-entrypoint..." 37 minutes ago Up 37 minutes 80/tcp agitated_haslett
df52d0e84cc8 nginx "/docker-entrypoint..." 38 minutes ago Up 38 minutes 80/tcp beautiful_rosalind
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d nginx:1.25
57d1da9e4070e0128abd1bf31e22797becbc75fb43046b88b7c4cef28068d29a
tasneem@DESKTOP-0VT5601:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
57d1da9e4070 nginx:1.25 "/docker-entrypoint..." 8 seconds ago Up 7 seconds 80/tcp condescending_hermann
77202cac59b4 nginx "/docker-entrypoint..." 44 seconds ago Up 43 seconds 80/tcp naughty_curie
d2916eaa8050 nginx "/docker-entrypoint..." 38 minutes ago Up 38 minutes 80/tcp agitated_haslett
df52d0e84cc8 nginx "/docker-entrypoint..." 38 minutes ago Up 38 minutes 80/tcp beautiful_rosalind
tasneem@DESKTOP-0VT5601:~$
```

## 7. \*\*If you delete a container, does it delete the image too? Why or why not?\*\*

No — deleting a container does not delete the image.

A container is just a running (or stopped) instance created from an image.

The image stays on your machine so you can create new containers from it again.

You'd have to explicitly remove the image with docker rmi if you want it gone.

```
tasneem@DESKTOP-0VT5601:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
57d1da9e4070 nginx:1.25 "/docker-entrypoint..." 8 seconds ago Up 7 seconds 80/tcp condescending_hermann
77202cac59b4 nginx "/docker-entrypoint..." 44 seconds ago Up 43 seconds 80/tcp naughty_curie
d2916eaa8050 nginx "/docker-entrypoint..." 38 minutes ago Up 38 minutes 80/tcp agitated_haslett
df52d0e84cc8 nginx "/docker-entrypoint..." 38 minutes ago Up 38 minutes 80/tcp beautiful_rosalind
tasneem@DESKTOP-0VT5601:~$ sudo docker rm -f condescending_hermann
condescending_hermann
tasneem@DESKTOP-0VT5601:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
77202cac59b4 nginx "/docker-entrypoint..." 5 minutes ago Up 5 minutes 80/tcp naughty_curie
d2916eaa8050 nginx "/docker-entrypoint..." 43 minutes ago Up 43 minutes 80/tcp agitated_haslett
df52d0e84cc8 nginx "/docker-entrypoint..." 43 minutes ago Up 43 minutes 80/tcp beautiful_rosalind
tasneem@DESKTOP-0VT5601:~$ sudo docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
alpine latest 9234e8fb04c4 3 weeks ago 8.31MB
nginx latest 2cd1d97f893f 3 weeks ago 192MB
ubuntu latest 65ac7a6f3544 3 weeks ago 78.1MB
hello-world latest 74cc54e27dc4 6 months ago 10.1kB
nginx 1.25 e784f4560448 15 months ago 188MB
tasneem@DESKTOP-0VT5601:~$
```

## 8. \*\*What does this command do?\*\*

``` bash

**docker pull ubuntu && docker run -it ubuntu**

```

docker pull ubuntu → downloads the latest Ubuntu image from Docker Hub to your machine.

&& → means “run the next command only if the first one succeeds.”

`docker run -it ubuntu` → starts a new container from that Ubuntu image in interactive mode with a terminal.

```
Using default tag: latest
latest: Pulling from library/ubuntu
3f112e3802c: Pull complete
Digest: sha256:a08e551cb33859e0740772b38217fc1796a66da2506d312abe51acd354ff061
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
docker permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied

Run 'docker run --help' for more information
lasseen@DESKTOP-4V75601:~$ sudo docker rmi ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:a08e551cb33859e0740772b38217fc1796a66da2506d312abe51acd354ff061
Deleted: sha256:05ae7adf5344dd2db6b19b13bf64c752d76bc592150f874188fcfa40d205a3
Deleted: sha256:107c7bd9eecc042e6b154640c94972c638f4e7fe795902b149e8ce9acbd03d59d7
Deleted: sha256:107c7bd9eecc042e6b154640c94972c638f4e7fe795902b149e8ce9acbd03d59d7
lasseen@DESKTOP-4V75601:~$ sudo docker pull ubuntu && sudo docker run -it ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
3f112e3802c: Pull complete
Digest: sha256:a08e551cb33859e0740772b38217fc1796a66da2506d312abe51acd354ff061
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@935817a79144:~# sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
b212b94acc69      "nginx"             "/docker-entrypoint..."   55 minutes ago    Up 55 minutes
b212b94acc69      "nginx"             "/docker-entrypoint..."   55 minutes ago    Up 55 minutes
d330a6599c78      "nginx"             "/docker-entrypoint..."   58 minutes ago    Exited (0) 56 minutes ago
d330a6599c78      "nginx"             "/docker-entrypoint..."   58 minutes ago    Exited (0) 57 minutes ago
f227108111f       "alpine"            "sh"                24 hours ago     Exited (255) 2 hours ago
b0d65ca4e802       "alpine"            "sh"                24 hours ago     Exited (255) 2 hours ago
a53bf95c29        "alpine"            "sh"                25 hours ago     Exited (255) 2 hours ago
f977a0a034         "alpine"            "sh"                25 hours ago     Exited (255) 2 hours ago
a53bf95c29        "alpine"            "sh"                25 hours ago     Exited (255) 2 hours ago
f4c4295c235       "alpine"            "bin/sh"           25 hours ago     Exited (0) 25 hours ago
fb5fb5a55b        "nginx"             "/docker-entrypoint..."   25 hours ago     Exited (255) 2 hours ago
fb5fb5a55b        "nginx"             "/docker-entrypoint..."   25 hours ago     Created
85675dd822c       "nginx"             "/docker-entrypoint..."   25 hours ago     Exited (0) 25 hours ago
f1136a405b8c       "nginx"             "/docker-entrypoint..."   25 hours ago     Exited (255) 2 hours ago
85675dd822c       "nginx"             "/docker-entrypoint..."   25 hours ago     Exited (255) 2 hours ago
lasseen@DESKTOP-4V75601:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
b212b94acc69      "nginx"             "/docker-entrypoint..."   55 minutes ago    Up 55 minutes
d330a6599c78      "nginx"             "/docker-entrypoint..."   58 minutes ago    Exited (0) 56 minutes ago
f227108111f       "alpine"            "sh"                24 hours ago     Up 24 hours
b0d65ca4e802       "alpine"            "sh"                24 hours ago     Up 24 hours
a53bf95c29        "alpine"            "sh"                25 hours ago     Up 25 hours
f977a0a034         "alpine"            "sh"                25 hours ago     Up 25 hours
a53bf95c29        "alpine"            "sh"                25 hours ago     Up 25 hours
f4c4295c235       "alpine"            "bin/sh"           25 hours ago     Up 25 hours
fb5fb5a55b        "nginx"             "/docker-entrypoint..."   25 hours ago     Up 25 hours
fb5fb5a55b        "nginx"             "/docker-entrypoint..."   25 hours ago     Up 25 hours
85675dd822c       "nginx"             "/docker-entrypoint..."   25 hours ago     Up 25 hours
f1136a405b8c       "nginx"             "/docker-entrypoint..."   25 hours ago     Up 25 hours
85675dd822c       "nginx"             "/docker-entrypoint..."   25 hours ago     Up 25 hours
lasseen@DESKTOP-4V75601:~$
```

9. \*\*You have a local image `nginx:latest` . What happens if you run `docker pull nginx` again? Will it download the image again? Why or why not?\*\*

## Docker will:

**Contact Docker Hub to check the latest digest for nginx:latest**

**Compare that digest to the one you already have locally.**

If they match Docker will not download the image again and if not Docker will download only the changed layers instead of the entire image.

```
tasneem@DESKTOP-0VT5601:~$ sudo docker image ls
[sudo] password for tasneem:
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
alpine          latest        9234e8fb04c4  3 weeks ago   8.31MB
nginx           latest        2cd1d97f893f  3 weeks ago   192MB
ubuntu          latest        65ae7a6f3544  3 weeks ago   78.1MB
hello-world     latest        74cc54e27dc4  6 months ago  10.1kB
nginx           1.25         e784f4560448  15 months ago  188MB
tasneem@DESKTOP-0VT5601:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
tasneem@DESKTOP-0VT5601:~$ sudo docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
alpine          latest        9234e8fb04c4  3 weeks ago   8.31MB
nginx           latest        2cd1d97f893f  3 weeks ago   192MB
ubuntu          latest        65ae7a6f3544  3 weeks ago   78.1MB
hello-world     latest        74cc54e27dc4  6 months ago  10.1kB
nginx           1.25         e784f4560448  15 months ago  188MB
```

10. \*\*What's the difference between these two commands:\*\*

```
```bash
```

```
docker rmi nginx
```

```
docker image prune
```

```
```
```

### ► \*When would each be useful?\*

docker rmi nginx:

Remove the specific image called nginx from your local machine

Will fail if any container (running or stopped) is still using that image, unless you add -f (

When useful:

You know you no longer need a particular image. You want to free up space from one specific app's image.

docker image prune:

Remove all unused images

Doesn't remove images used by existing containers

When useful:

You want to clean up your system from old, leftover images in bulk. Good for freeing space after many experiments or builds.

```
lassemc@DESKTOP-0VT5601:~$ sudo docker ps -a
CONTAINER ID        IMAGE               CREATED             STATUS              PORTS               NAMES
935817a7914d        ubuntu              "bin/bash"         47 minutes ago    Exited (127) 43 minutes ago   funny_cerf
0222e0a1151f        alpine              "sh"               25 hours ago     Exited (255) 2 hours ago    c6
0046cadef2e2        alpine              "sh"               25 hours ago     Exited (255) 2 hours ago    e5
a538f9c5e295        alpine              "sh"               25 hours ago     Exited (255) 2 hours ago    e4
f977d01038dd        alpine              "sh"               26 hours ago     Exited (255) 2 hours ago    e2
9088ba0a52f9        alpine              "sh"               26 hours ago     Exited (255) 2 hours ago    c1
2fc4259cc235        alpine              "/bin/sh"         26 hours ago     Exited (0) 26 hours ago    vigorous.heisenberg
4d91babf620b        hello-world        "/hello"           27 hours ago     Exited (0) 27 hours ago    jovial_herschel
6143e19519ad        hello-world        "/hello"           27 hours ago     Exited (0) 27 hours ago    keen_galois
lassemc@DESKTOP-0VT5601:~$ sudo docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:84e966e61a8c7846f599da7eb081c551d56817448728924a87ab32f12a72fb
Deleted: sha256:2cd1d97f893f70ce86a38b7160c30e5750f3f3edad86c598884ca9ca563a501
Deleted: sha256:3c1159c07ff83ede93fc21302ae30b39b04378b6b1b52451d701d555cc1cb9
Deleted: sha256:221aaeee603b528a74a32189bffff7853e07f6574185366a2272fb5e525e4e99e
Deleted: sha256:701d5f8b1f92711c04588f86fe9b558671bce6ba151a905644dc5c79130a86db9
Deleted: sha256:a8b72be3a456e3035czcaf7d2cbe335802742bd186e3fcfdd7d5228c36f6ef27
Deleted: sha256:5064a019a0ed54289625d07404c88528252b0515dd3426d8d400940e939c5a9
Deleted: sha256:aef100bc4c4630446753dc1c0de8318853d4bea6c897fa4c2aa19fab4d0bbdf4
Deleted: sha256:7cc7fe68eff66ff19872441a1938eecd4ad33746d2baa3abc081c1e6fe25988e
lassemc@DESKTOP-0VT5601:~$ sudo docker image ls
REPOSITORY          TAG      IMAGE ID            CREATED             SIZE
alpine              latest   9234e8fb4c44  3 weeks ago       8.31MB
ubuntu              latest   65ae7af3f544  3 weeks ago       78.1MB
hello-world         latest   74cc54e27dc4  6 months ago      10.1kB
```

```

tasneem@DESKTOP-0VT5601:~$ docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied
tasneem@DESKTOP-0VT5601:~$ sudo docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
tasneem@DESKTOP-0VT5601:~$ sudo docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
alpine latest 9234e8f0d4c4 3 weeks ago 8.31MB
nginx latest 2cd1d97f893f 3 weeks ago 192MB
ubuntu latest 65ae6a6f3544 3 weeks ago 78.1MB
hello-world latest 74c5c4e27dc4 6 months ago 10.1kB
nginx 1.28 e784f4508d4d 15 months ago 188MB

tasneem@DESKTOP-0VT5601:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
935817a791d4 ubuntu "/bin/bash" 35 minutes ago Exited (127) 30 minutes ago
77202ac59b4 nginx "/docker-entrypoint..." 49 minutes ago Up 49 minutes 80/tcp
d2916ea08850 nginx "/docker-entrypoint..." About an hour ago Up About an hour 80/tcp
df52d0e94cc8 nginx "/docker-entrypoint..." About an hour ago Up About an hour 80/tcp
2b21bd94cc69 nginx "/docker-entrypoint..." About an hour ago Exited (0) About an hour ago
d330a2699c78 nginx "/docker-entrypoint..." About an hour ago Exited (0) About an hour ago
u2271081c1f alpine "sh" 25 hours ago Exited (255) 2 hours ago
b0d6c4e42cc2 alpine "sh" 25 hours ago Exited (255) 2 hours ago
u538f9c5e295 alpine "sh" 25 hours ago Exited (255) 2 hours ago
f977da1038da alpine "sh" 25 hours ago Exited (255) 2 hours ago
008b6aa052f9 alpine "sh" 25 hours ago Exited (255) 2 hours ago
f7c427a235b alpine "/bin/sh" 25 hours ago Exited (0) 25 hours ago
c069bae55b5 nginx "/docker-entrypoint..." 20 hours ago Exited (255) 2 hours ago
e54578de6b2b3 nginx "/docker-entrypoint..." 26 hours ago Exited (0) 26 hours ago
f1136495b8c8 nginx "/docker-entrypoint..." 26 hours ago Exited (255) 2 hours ago 0.0.0.0:8083->80/tcp, [::]:8083->80/tcp
856cb4083e14 nginx "/docker-entrypoint..." 26 hours ago Exited (255) 2 hours ago 80/tcp
4d91babf620b hello-world "/hello" 27 hours ago Exited (0) 27 hours ago
6143e19519ad hello-world "/hello" 27 hours ago Exited (0) 27 hours ago
1a252f285083 nginx "/docker-entrypoint..." 13 days ago Exited (255) 2 days ago 80/tcp
92732f0fc1a1 nginx "/docker-entrypoint..." 13 days ago Exited (255) 2 days ago 80/tcp

NAMES
funny_cerf
naughty.curie
agitated.haslett
beautiful.rosalind
affectionate.herschel
angry_carson
c6
c5
c4
c2
c1
vigorous.heisenberg
romantic.jang
nervous.burden
severe.wu
sweet.stonebraker
jovial.herschel
keen.galois
test.nginx
practical.roentgen

```

## 11. \*\*True or False:\*\* Docker images can be shared between different operating systems.

► \*Explain your answer.\*

True

Docker images can be shared between different operating systems if those systems support Docker and the image's architecture

you can save an image on one machine (Linux) using docker save, transfer the image file to another machine , and load it there with docker load

## 12. \*\*Can you save a Docker image as a file and share it without pushing it to Docker Hub? How?\*\*

Yes, you can save a Docker image as a file and share it without pushing to Docker Hub

**docker save -o <file>.tar <image>**

**scp myimage.tar user@remote\_host:/path/on/remote/**

**ssh user@remote\_host**

**docker load -i <file>.tar**

```

tasneem@DESKTOP-0VT5601:~$ sudo docker save -o my_image.tar nginx
[sudo] password for tasneem:
tasneem@DESKTOP-0VT5601:~$ ls
index.html  my_image.tar  my_local_image  task  task1  test.txt

```

```

tasneem@DESKTOP-0VT5601:~$ sudo scp -i /home/tasneem/tasneem.pem my_image.tar ubuntu@35.160.253.225:/home/ubuntu
tasneem@DESKTOP-0VT5601:~$ sudo ssh -i /home/tasneem/tasneem.pem ubuntu@35.160.253.225
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Mon Aug 11 00:20:48 UTC 2025

System load: 0.0 Temperature: -273.1 °C
Usage of /: 28.3% of 6.71GB Processes: 108
Memory usage: 25% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.31.203

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
See "man sudo_root" for details.

Activate Windows
Go to Settings to activate Windows.

CMD - 2:26 AM

```

```

ubuntu@ip-172-31-31-201:~$ sudo snap install docker
docker-28.1.1+1 from Canonical installed
ubuntu@ip-172-31-31-201:~$ docker load -i /home/ubuntu/my_image.tar
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.48/images/load?quiet=0": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-31-201:~$ sudo docker load -i /home/ubuntu/my_image.tar
5d4427064ecc: Loading layer [=====] 77.88MB/77.88MB
fc1cf9ca519: Loading layer [=====] 113.9MB/113.9MB
74fb290ebeb8: Loading layer [=====] 3.584KB/3.584KB
9f4d73e03594: Loading layer [=====] 4.008KB/4.008KB
50ff93014c4d: Loading layer [=====] 2.56KB/2.56KB
7d2fd99c368c: Loading layer [=====] 5.12KB/5.12KB
14773070944d: Loading layer [=====] 7.168KB/7.168KB
Loaded Image: nginx:1.25
ubuntu@ip-172-31-31-201:~$ sudo docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx 1.25 e784f4560448 15 months ago 188MB

Activate Windows
Go to Settings to activate Windows.

```

### **13. \*\*What is the result of this command? Why might you use it?\*\***

``` bash

`docker save -o backup.tar nginx`

```

**It saves the Docker image named nginx into a single tar archive file called backup.tar on your local machine.**

**To create a backup of the image.**

**To transfer or share the image easily without pushing it to a remote registry like Docker Hub.**

**Useful for migrating images between machines or environments where internet access or registry access is limited.**

```

tasneem@DESKTOP-0VT5601:~$ sudo docker save -o backup.tar nginx
[sudo] password for tasneem:
tasneem@DESKTOP-0VT5601:~$ ls
backup.tar index.html my_image.tar my_local_image task task1 tasneem.pem tasneem.pem:Zone.Identifier test.txt
tasneem@DESKTOP-0VT5601:~$
```

### **14. \*\*How can you copy an image from one machine to another without using Docker Hub or a registry?\*\* saving the image to a file, transferring the file, and loading it on the other machine**

```

tasneem@DESKTOP-0VT5601:~$ sudo docker save -o my_image.tar nginx
[sudo] password for tasneem:
tasneem@DESKTOP-0VT5601:~$ ls
index.html  my_image.tar  my_local_image  task  task1  test.txt

tasneem@DESKTOP-0VT5601:~$ sudo scp -i /home/tasneem/tasneem.pem my_image.tar ubuntu@35.160.253.225:/home/ubuntu
my_image.tar
tasneem@DESKTOP-0VT5601:~$ sudo ssh -i /home/tasneem/tasneem.pem ubuntu@35.160.253.225
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/pro

System information as of Mon Aug 11 00:26:48 UTC 2025

System load: 0.0          Temperature:           -273.1 °C
Usage of /: 28.3% of 6.71GB Processes:           108
Memory usage: 25%          Users logged in:     0
Swap usage:  0%          IPv4 address for ens5: 172.31.31.203

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Activate Windows
Go to Settings to activate Windows.

100% 183MB 319.3KB/s 09:46

ubuntu@ip-172-31-31-203:~$ sudo snap install docker
docker 28.1.1+1 from Canonical installed
ubuntu@ip-172-31-31-203:~$ docker load -i /home/ubuntu/my_image.tar
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://<FVar%2FrUn%2Fdocker.sock>/v1.48/images/load?quiet=0": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-31-203:~$ sudo docker load -i /home/ubuntu/my_image.tar
5d4d427064ecc: Loading layer [=====] 77.88MB/77.88MB
fc1cf9ca5139: Loading layer [=====] 113.9MB/113.9MB
f47b290eb0b1: Loading layer [=====] 3.384MB/3.384MB
9f4f230f111c: Loading layer [=====] 6.608MB/6.608MB
66f9feadd0: Loading layer [=====] 5.568B/2.56KB
7a2fd59c368c: Loading layer [=====] 5.128B/5.12KB
147730709094d: Loading layer [=====] 7.168kB/7.168kB
Loaded image: nginx:1.25
ubuntu@ip-172-31-31-203:~$ sudo docker image ls
REPOSITORY  TAG      IMAGE ID      CREATED      SIZE
nginx       1.25     e784f4560448   15 months ago  188MB

```

## 15. \*\*How do you inspect the internal metadata of an image? What kind of information can you find?\*\*

**sudo docker image inspect ubuntu**

**Id:** Unique identifier of the image.

**Created:** Timestamp when the image was built.

**Author:** Who created the image (if specified).

**Architecture:** CPU architecture (e.g., amd64, arm64).

**Os:** Operating system (e.g., linux, windows).

**Size:** Size of the image.

**ContainerConfig & Config:** Configuration settings like environment variables, default commands (CMD), entrypoint (ENTRYPOINT), exposed ports, volumes, etc.

**Layers:** Information about the image layers (history).

**Labels:** Custom metadata tags set on the image.

```

tasneem@DESKTOP-0VT5601:~$ sudo docker image inspect ubuntu
[{"Id": "sha256:65ae7a6f3544bd2d2b6d19b13bfc64752d776bc92c510f874188bf404d205a3",
 "RepoTags": [
     "ubuntu:latest"
 ],
 "RepoDigests": [
     "ubuntu@sha256:a08e551cb33850e4740772b38217fc1796a66da2506d312abe51acd354ff061"
 ],
 "Parent": "",
 "Comment": "",
 "Created": "2025-07-14T16:40:23.439496836Z",
 "DockerVersion": "24.0.7",
 "Author": "",
 "Architecture": "amd64",
 "Os": "linux",
 "Size": 7818997,
 "GraphDriver": {
     "Data": {
         "MergedDir": "/var/lib/docker/overlay2/835451b1fleef1ca9ba75385f8ee6404c3391b4af95a9159d3d1536736e5ab5d/merged",
         "UpperDir": "/var/lib/docker/overlay2/835451b1fleef1ca9ba75385f8ee6404c3391b4af95a9159d3d1536736e5ab5d/diff",
         "WorkDir": "/var/lib/docker/overlay2/835451b1fleef1ca9ba75385f8ee6404c3391b4af95a9159d3d1536736e5ab5d/work"
     },
     "Name": "overlay2"
 },
 "RootFS": {
     "Type": "layers",
     "Layers": [
         "sha256:107cbdaec042e6154640c94972c638f4e2fee795902b149e8ce9acbd03d59d7"
     ]
 },
 "Metadata": {
     "LastTagTime": "0001-01-01T00:00:00Z"
 },
 "Config": {
     "Cmd": [
         "/bin/bash"
     ],
     "Entrypoint": null,
     "Env": [
         "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
     ],
     "Labels": {
         "org.opencontainers.image.ref.name": "ubuntu",
         "org.opencontainers.image.version": "24.04"
     },
     "OnBuild": null,
     "User": ""
 }
}]

```

Activate Windows  
Go to Settings to activate Windows.

## ## 🌐 Part 2: Networking and Bridge Mode

### 4. \*\*Run two containers without specifying a network:\*\*

```
sudo docker run -dit --name first_container alpine sh
```

```
sudo docker run -dit --name second_container alpine sh
```

```

tasneem@DESKTOP-0VT5601:~$ sudo docker run -dit --name first_container alpine sh
0aec578af6b64f915b5d2e1af1e259a367d88bd3f20c45d8c4eadba72373471d
tasneem@DESKTOP-0VT5601:~$ sudo docker run -dit --name second_container alpine sh
78752a57024820a5942403db3400897b09c3d29259d489da4bca7630207d8504

```

### 5. \*\*Try to ping `container1` from `container2` :\*\*

- What happens? Why?

**failed to ping**

**On the default bridge network, Docker does not provide automatic DNS resolution of container names so we need ips to ping.**

```

tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it first_container ping second_container
ping: bad address 'second_container'
tasneem@DESKTOP-0VT5601:~$ 

```

### 6. \*\*Inspect the `docker0` bridge network and check container IPs:\*\*

```
sudo docker network inspect bridge
```

```

[asnew@DESKTOP-0VT5601:~]$ sudo docker network ls
NETWORK ID      NAME        DRIVER      SCOPE
6429b337377b   be_fe_intg  bridge      local
81698d697f4b   bridge      bridge      local
3816aa03ae    host        host        local
b69bf7c6a459   my_net     bridge      local
0683d58b0384   my_netl    bridge      local
34b3b2629abd   none       null       local
[asnew@DESKTOP-0VT5601:~]$ sudo docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "81698d697f4bec58bae2465d1957bb160695488b2e709a4ae76c8f06ebb3c373",
    "Created": "2025-08-10T21:53:58.393198802+03:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": null,
    "Network": null,
    "ConfigOnly": false,
    "Containers": {
      "0aec578af6b64f915b5d2e1af1e259a367d88bd3f20c45d8c4eadba72373471d": {
        "Name": "first_container",
        "EndpointID": "b91844aee393a8478fe5e9d0f01091877987bd717dbbd28d1c4ef9f6787fc7d6",
        "MacAddress": "8e:bd:f2:6e:02:0a",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
      },
      "78752a57024820a5942403db3400897b09c3d29259d489da4bca7630207d8504": {
        "Name": "second_container",
        "EndpointID": "66a6049776742d1e91b490f3a5de828ca84db4f73fadad5fa493c95c9ed76bf0",
        "MacAddress": "a2:ed:9e:42:d1:c2",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
      }
    }
  }
]

```

Activate Windows  
Go to Settings to activate Windows

```

  "Network": null,
},
"ConfigOnly": false,
"Containers": {
  "0aec578af6b64f915b5d2e1af1e259a367d88bd3f20c45d8c4eadba72373471d": {
    "Name": "first_container",
    "EndpointID": "b91844aee393a8478fe5e9d0f01091877987bd717dbbd28d1c4ef9f6787fc7d6",
    "MacAddress": "8e:bd:f2:6e:02:0a",
    "IPv4Address": "172.17.0.2/16",
    "IPv6Address": ""
  },
  "78752a57024820a5942403db3400897b09c3d29259d489da4bca7630207d8504": {
    "Name": "second_container",
    "EndpointID": "66a6049776742d1e91b490f3a5de828ca84db4f73fadad5fa493c95c9ed76bf0",
    "MacAddress": "a2:ed:9e:42:d1:c2",
    "IPv4Address": "172.17.0.3/16",
    "IPv6Address": ""
  }
}

```

## 7. \*\*Now try pinging `container1` from `container2` using IP address.\*\*

```
sudo docker exec -it first_container ping 172.17.0.3
```

```
tasneem@DESKTOP-0VT5601:~$ sudo docker exec -it first_container ping 172.17.0.3
PING 172.17.0.3 (172.17.0.3): 56 data bytes
64 bytes from 172.17.0.3: seq=0 ttl=64 time=44.389 ms
64 bytes from 172.17.0.3: seq=1 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=2 ttl=64 time=0.153 ms
64 bytes from 172.17.0.3: seq=3 ttl=64 time=0.144 ms
64 bytes from 172.17.0.3: seq=4 ttl=64 time=0.143 ms
64 bytes from 172.17.0.3: seq=5 ttl=64 time=0.164 ms
64 bytes from 172.17.0.3: seq=6 ttl=64 time=0.146 ms
64 bytes from 172.17.0.3: seq=7 ttl=64 time=0.153 ms
64 bytes from 172.17.0.3: seq=8 ttl=64 time=0.150 ms
64 bytes from 172.17.0.3: seq=9 ttl=64 time=0.154 ms
64 bytes from 172.17.0.3: seq=10 ttl=64 time=0.151 ms
64 bytes from 172.17.0.3: seq=11 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=12 ttl=64 time=0.153 ms
64 bytes from 172.17.0.3: seq=13 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=14 ttl=64 time=0.061 ms
64 bytes from 172.17.0.3: seq=15 ttl=64 time=0.105 ms
64 bytes from 172.17.0.3: seq=16 ttl=64 time=0.154 ms
64 bytes from 172.17.0.3: seq=17 ttl=64 time=0.155 ms
64 bytes from 172.17.0.3: seq=18 ttl=64 time=0.395 ms
64 bytes from 172.17.0.3: seq=19 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=20 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=21 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=22 ttl=64 time=0.152 ms
64 bytes from 172.17.0.3: seq=23 ttl=64 time=0.157 ms
64 bytes from 172.17.0.3: seq=24 ttl=64 time=0.153 ms
64 bytes from 172.17.0.3: seq=25 ttl=64 time=0.156 ms
64 bytes from 172.17.0.3: seq=26 ttl=64 time=0.078 ms
64 bytes from 172.17.0.3: seq=27 ttl=64 time=0.088 ms
64 bytes from 172.17.0.3: seq=28 ttl=64 time=0.177 ms
64 bytes from 172.17.0.3: seq=29 ttl=64 time=0.080 ms
64 bytes from 172.17.0.3: seq=30 ttl=64 time=0.160 ms
64 bytes from 172.17.0.3: seq=31 ttl=64 time=0.162 ms
64 bytes from 172.17.0.3: seq=32 ttl=64 time=0.162 ms
64 bytes from 172.17.0.3: seq=33 ttl=64 time=0.227 ms
64 bytes from 172.17.0.3: seq=34 ttl=64 time=0.150 ms
64 bytes from 172.17.0.3: seq=35 ttl=64 time=0.164 ms
```

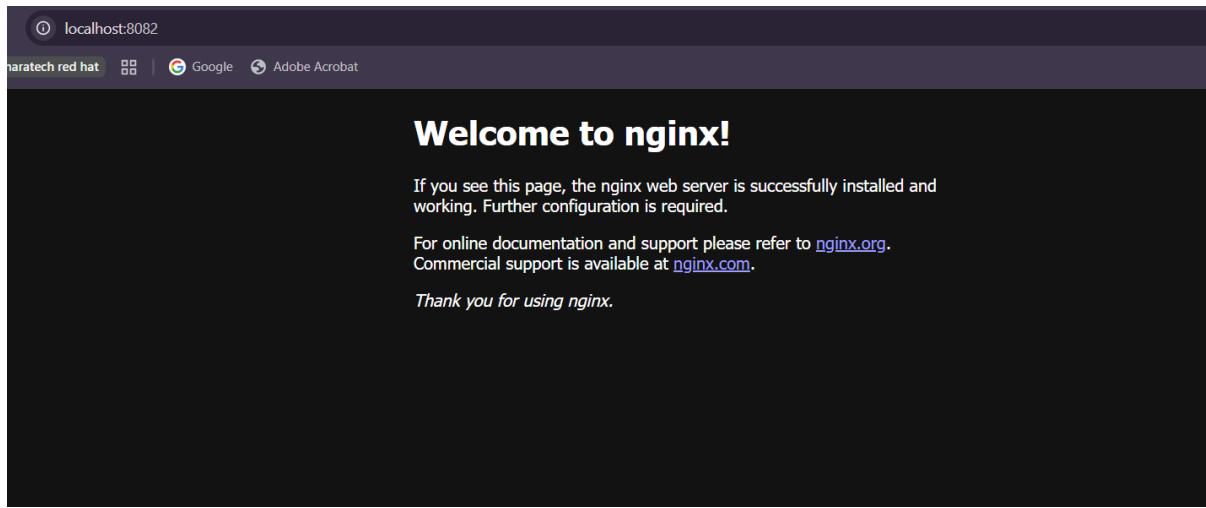
---

## ## Part 3: Port Forwarding

## 8. \*\*Run an Nginx container with port forwarding:\*\*

```
sudo docker run -d -p 8082:80 nginx
```

```
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d -p 8082:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
59e22667830b: Pull complete
140da4f89dcb: Pull complete
96e47e70491e: Pull complete
2ef442a3816e: Pull complete
4b1e45a9989f: Pull complete
1d9f51194194: Pull complete
f30ffbee4c54: Pull complete
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb
Status: Downloaded newer image for nginx:latest
4435e7dfef1d9de0ca4666ba786df5020bca2d8077acd13d1e5a673648f7dedc
+-----+
          Specified: 8082
          Default: 80
```



## 9. \*\*Access the container from the browser or using `curl`:\*\*

**curl http://localhost:8082**

```
tasneem@DESKTOP-0VT5601:~$ sudo docker run -d -p 8082:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
59e22667830b: Pull complete
140da4f89dc9: Pull complete
96e47e70491e: Pull complete
2ef442a3816e: Pull complete
4b1e45a9989f: Pull complete
1d9f51194194: Pull complete
f30ffbee4c54: Pull complete
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb
Status: Downloaded newer image for nginx:latest
4435e7dfe1d9de0ca4666ba786df5020bca2d8077acd13d1e5a673648f7dedc
tasneem@DESKTOP-0VT5601:~$ curl http://localhost:8082
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

## 10. \*\*Try running a second Nginx container with the same port mapping. What happens? Why?\*\*

**sudo docker run -d -p 8082:80 nginx**

I get an error message because the port on my host machine is in use and You cannot bind two different containers to the same host port

```
[Administrator:0.0.0.0:0] ~$ sudo docker run -d -p 8082:80 nginx  
5d84bf6fd45c2400868ed614c720889780027318372e539e5ab80081a39b1585  
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint cool_hellman (56915c2447ae900dd7d543ab40c9bdf3245918105ce51f9ee2f249e9e0  
f6): Bind for :::8082 failed: port is already allocated  
Run 'docker run --help' for more information
```