1. docker build –t <image name >

```
E:\Hello World>docker build -t task .
[+] Building 2.8s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                0.1s
 => => transferring dockerfile: 31B                                                                 0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.7                                       2.6s
 => [auth] library/python:pull token for registry-1.docker.io                                       0.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 358B                                                                   0.0s
 => [1/4] FROM docker.io/library/python:3.7@sha256:51dcbb98ba807f3631366d5a0678bc6815ca71901ddab6ed2ed5319da79f1c4d  0.0s
 => CACHED [2/4] COPY . /app                                                                        0.0s
 => CACHED [3/4] WORKDIR /app                                                                       0.0s
 => CACHED [4/4] RUN pip install -r requirements.txt                                                0.0s
 => exporting to image                                                                              0.0s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:610a692540d341c8257a8105c407afdde81f4efbb639ba1ef865bc2fa980d569        0.0s
 => => naming to docker.io/library/task                                                             0.0s
```

The above command used to build a docker file with tag.

2. docker images

```
E:\Hello World>docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
task                    latest    610a692540d3   12 hours ago    1.06GB
docker/getting-started  latest    cb90f98fd791   6 months ago    28.8MB
```

The above command used to list all images that are available.

3. docker rmi <image name>

```
E:\Hello World>docker rmi docker/getting-started
Untagged: docker/getting-started:latest
Untagged: docker/getting-started@sha256:b558be874169471bd4e65bd6eac8c303b271a7ee8553ba47481b73b2bf597aae
Deleted: sha256:cb90f98fd791dd49f09903cef3eb2245646b4d76b093825ea78e0f7bb8fb3403
Deleted: sha256:b6b308c7ce72e0286f9455b9f76ae6cafe55fcc6b068950414165f43bda11fd7
Deleted: sha256:711ca3e1c68e1406fd5b96a71fcf29e4838887b827bd4ee48dfc6e6a62d8fabf
Deleted: sha256:1380ce106a10fac3c312f83ddf8406d187d5c0dd567d9a2454abe6ba563114cd
Deleted: sha256:36e9639dd7f8b2549aba50c0a7d2402510ddb99d3e789515ab6646f21ef392ec
Deleted: sha256:b35646458162a8f3289c0605c02ad46c2a05ae5c977a46e11d56962b373e1e98
Deleted: sha256:e61e5c961a35926efc4df0bcd33aa988c860ba8440ae2bb713084b14b89c9806
Deleted: sha256:f60e2e50f4b58e60ef21034b9d2df92705fa8bb3870b2ca81089de8af45a2e90
Deleted: sha256:4fc242d58285699eca05db3cc7c7122a2b8e014d9481f323bd9277baacfa0628
```

The above command used to delete particular already created image.

4. docker pull <image name>

```
E:\Hello World>docker pull docker/getting-started
Using default tag: latest
latest: Pulling from docker/getting-started
df9b9388f04a: Pull complete
5867cba5fcbd: Pull complete
4b639e65cb3b: Pull complete
061ed9e2b976: Pull complete
bc19f3e8eeb1: Pull complete
4071be97c256: Pull complete
79b586f1a54b: Pull complete
0c9732f525d6: Pull complete
Digest: sha256:b558be874169471bd4e65bd6eac8c303b271a7ee8553ba47481b73b2bf597aae
Status: Downloaded newer image for docker/getting-started:latest
docker.io/docker/getting-started:latest
```

The above command used to download images from docker hub.

5. docker run –d –p network host <image name>

```
(base) E:\Hello World>docker run -d -p 8000:6000 task
656c7f5440463a95f610f13f42c202e9babc13a8ca8f28a57d7bdc47e7d5d8c5
```

The above command use to create a container from an image file. Argument –d is provided to run the container in background as detached mode. The above command launches an *httpd* container and maps the host's port 8000 to port 6000 inside that container.

6. docker ps

```
(base) E:\Hello World>docker ps
CONTAINER ID   IMAGE     COMMAND           CREATED         STATUS         PORTS                    NAMES
656c7f544046   task      "python app.py"   9 seconds ago   Up 7 seconds   0.0.0.0:8000->6000/tcp   determined_varahamihira
```

This command only shows running containers by default. To see all containers, use the –a

7. docker logs <container>

```
(base) E:\Hello World> docker logs 656c7f544046
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:6000
 * Running on http://172.17.0.2:6000
Press CTRL+C to quit
172.17.0.1 - - [16/Oct/2022 23:29:05] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [16/Oct/2022 23:29:06] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [16/Oct/2022 23:29:11] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [16/Oct/2022 23:29:49] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [16/Oct/2022 23:29:49] "GET /favicon.ico HTTP/1.1" 404 -
```

The above command use to fetch the logs of that particular container.

8. docker stop <container>

```
(base) E:\Hello World> docker stop 656c7f544046
656c7f544046

(base) E:\Hello World> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

This command use to stop container service.

9. docker start <container>

```
(base) E:\Hello World>docker start 656c7f544046
656c7f544046

(base) E:\Hello World>docker ps
CONTAINER ID   IMAGE     COMMAND          CREATED         STATUS         PORTS                    NAMES
656c7f544046   task      "python app.py"  24 minutes ago  Up 3 seconds   0.0.0.0:8000->6000/tcp   determined_varahamihira
```

This command use to start the container service.

10. docker rename old name new name <container>

```
(base) E:\Hello World>docker rename determined_varahamihira karthik_task

(base) E:\Hello World>docker ps
CONTAINER ID   IMAGE     COMMAND          CREATED         STATUS         PORTS                    NAMES
656c7f544046   task      "python app.py"  27 minutes ago  Up 2 minutes   0.0.0.0:8000->6000/tcp   karthik_task
```

This command use to change the container name.

11. docker stats

```
CONTAINER ID   NAME          CPU %     MEM USAGE / LIMIT     MEM %     NET I/O       BLOCK I/O   PIDS
656c7f544046   karthik_task  0.02%     49.37MiB / 6.163GiB   0.78%     1.12kB / 0B   0B / 0B     1
```

This command shows the stats of running container.

12. docker search <image name>

```
(base) E:\Hello World>docker search krishnaik06/welcome-app
NAME                       DESCRIPTION   STARS     OFFICIAL   AUTOMATED
krishnaik06/welcome-app                  0
```

This command used to search public image file in docker hub

13. docker top <container name>

```
(base) E:\Hello World>docker top 72781964a76c
UID                 PID                 PPID                C                   STIME               TTY                 TIME
CMD
root                1417                1396                2                   22:35               ?                   00:00:00
python app.py
```

This command use to display the process of container

14. docker image prune

```
(base) E:\Hello World>docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

This command use to remove all dangling images which are unused. A dangling image is one that is not tagged and is not referenced by any container.

15. docker rm <container name>

```
(base) E:\Hello World>docker rm 72781964a76c
72781964a76c

(base) E:\Hello World>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

This command use to delete a container.