

# Docker Basics

# What is docker image

- Images are read-only templates containing instructions for creating a container.
- A Docker image creates containers to run on the Docker platform.
- An image is composed of multiple stacked layers.
  - For example, to build a web server image, start with an image that includes Ubuntu Linux (a base OS). Then, add packages like Apache and PHP on top.
  - We can manually build images using a Dockerfile, a text document containing all the commands to create a Docker image.

# What is container

- A container is an isolated place where an application runs without affecting the rest of the system and without the system impacting the application.
- They are isolated, containers are well-suited for securely running software like databases or web applications that need access to sensitive resources without giving access to every user on the system.

# Docker registry

- The Registry is a stateless, highly scalable server side application that stores and lets you distribute Docker images. The Registry is open-source, under the permissive [Apache license](#). You can find the source code on [GitHub](#).
  - tightly control where your images are being stored
  - fully own your images distribution pipeline
  - integrate image storage and distribution tightly into your in-house development workflow

# Docker basic commands

`docker search <name of the image>`: To search the public image on docker hub

```
root@ubuntu20:/etc# docker search nginx
```

NAME	DESCRIPTION	STARS
nginx	Official build of Nginx.	17366
linuxserver/nginx	An Nginx container, brought to you by LinuxS...	177
bitnami/nginx	Bitnami nginx Docker Image	140
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	58
bitnami/nginx-ingress-controller	Bitnami Docker Image for NGINX Ingress Contr...	19
rancher/nginx-ingress-controller		11
webdevops/nginx	Nginx container	9
ibmcom/nginx-ingress-controller	Docker Image for IBM Cloud Private-CE (Commu...	4
bitnami/nginx-ldap-auth-daemon		3
rancher/nginx		2

- docker pull <image name:tags> Pull image from docker hub. Tags are used to identify the image inside registry. If no tags provided it will pull the latest image.

```
root@ubuntu20:/etc# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:b95a99feebf7797479e0c5eb5ec0bdfa5d9f504bc94da550c2f58e839ea6914f
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
```

- docker images : This will list the images already available on local host machine

```
root@ubuntu20:/etc# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest   2b7d6430f78d  2 weeks ago   142MB
```

•  
  
docker run <name of the image> : it will run the container

```
root@ubuntu20:/etc# docker run -d nginx
3361799c8c6db16064cb3ecalf9b5365dc2c2b484736f666703d27bdb990aa1
root@ubuntu20:/etc# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
3361799c8c6d   nginx    "/docker-entrypoint..." 22 seconds ago Up 22 seconds 80/tcp      eloq
```



•  
**docker ps : list all the running container**

```
root@ubuntu20:/etc# docker run -d nginx
3361799c8c6db16064cb3eca1ff9b5365dc2c2b484736f666703d27bdb990aa1
root@ubuntu20:/etc# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
3361799c8c6d   nginx    "/docker-entrypoint..." 22 seconds ago Up 22 seconds 80/tcp       eloq
```

`docker stop <container id/container name>` : To stop the container

```
root@ubuntu20:/etc# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3361799c8c6d	nginx	"/docker-entrypoint..."	22 seconds ago	Up 22 seconds	80/tcp	eloquent_tu

```
root@ubuntu20:/etc# docker stop 3361799c8c6d
```

```
3361799c8c6d
```

```
root@ubuntu20:/etc#
```

..

`docker rename <old name> <new name>`

`docker restart <container id>`

```
3361799c8c6a
```

```
root@ubuntu20:/etc# docker run -d nginx
```

```
8ca4b02e84332fc8f14cf8f09096cc68117cb75f3428c71efdc192be34adb101
```

```
root@ubuntu20:/etc# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8ca4b02e8433	nginx	"/docker-entrypoint...."	3 seconds ago	Up 2 seconds	80/tcp	blissful_proskuriakova

```
root@ubuntu20:/etc# docker restart 8ca4b02e8433
```

```
8ca4b02e8433
```

```
root@ubuntu20:/etc# docker rename blissful_proskuriakova mynginx
```

```
root@ubuntu20:/etc# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8ca4b02e8433	nginx	"/docker-entrypoint...."	25 seconds ago	Up 15 seconds	80/tcp	mynginx

```
root@ubuntu20:/etc# █
```

..

## docker exec : Access the running container

```
root@ubuntu20:/etc# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8ca4b02e8433	nginx	"/docker-entrypoint...."	5 minutes ago	Up 5 minutes	80/tcp	mynginx

```
root@ubuntu20:/etc# docker exec -it mynginx bash
```

```
root@8ca4b02e8433:/#
```

••

## docker logs : fetch running container logs

```
root@ubuntu20:/etc# docker logs 8ca4b02e8433
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/09/09 04:52:30 [notice] 1#1: using the "epoll" event method
2022/09/09 04:52:30 [notice] 1#1: nginx/1.23.1
2022/09/09 04:52:30 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
```

..

`docker rm <container id /container name>`

```
2022/09/09 04:52:40 [notice] 1#1: start worker process 25
```

```
root@ubuntu20:/etc#
```

```
root@ubuntu20:/etc# docker rm mynginx
```

```
Error response from daemon: You cannot remove a running container 8ca4b02e84332fc8f14cf8f09096cc68117cb75f3428c71efdc134adb101. Stop the container before attempting removal or force remove
```

```
root@ubuntu20:/etc# docker stop mynginx
```

```
mynginx
```

```
root@ubuntu20:/etc# docker rm mynginx
```

```
mynginx
```

```
root@ubuntu20:/etc#
```