



--

Academic Year: 2025-26  
Subject: DevOPs Lab (DL)  
Subject Lab In-charge: Prof. Sujata Oak

Semester: V Class / Branch: TE IT

## EXPERIMENT NO. 08

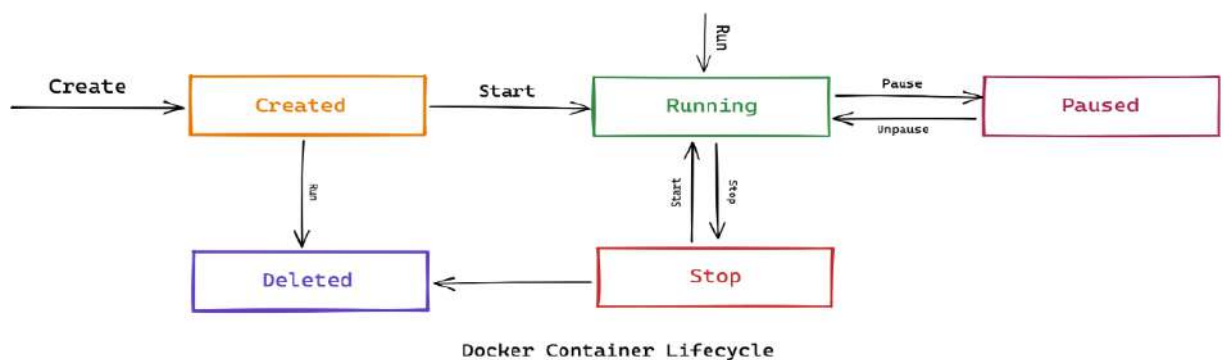
**Aim:** To demonstrate container lifecycle using various docker commands.

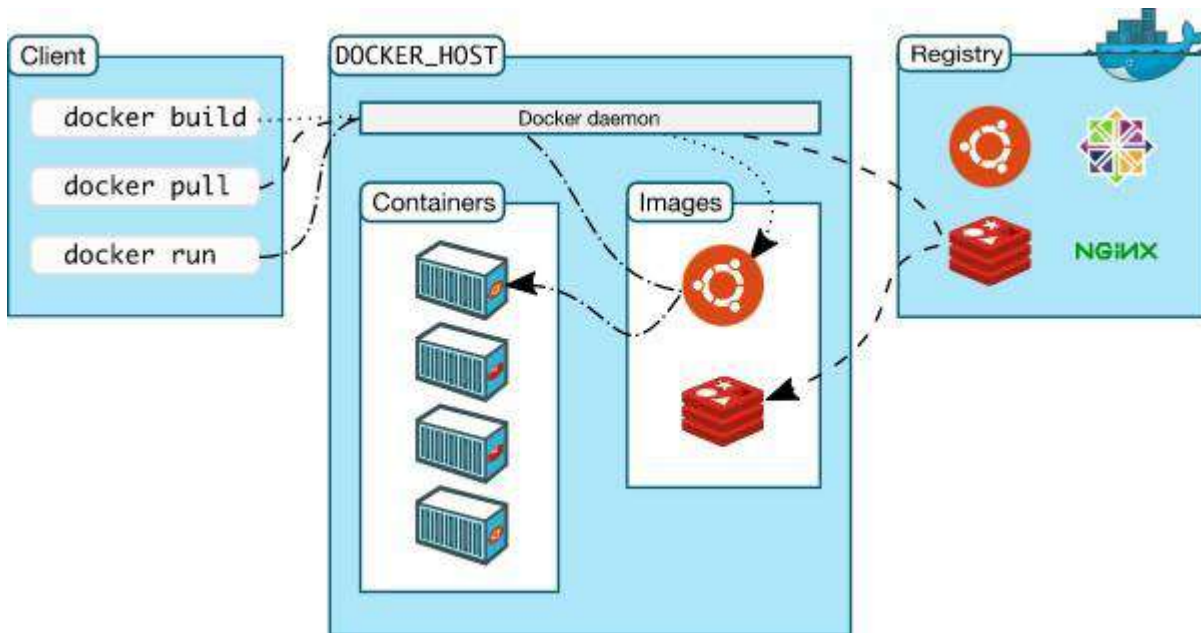
### Theory:

**Docker** is an open-source platform that helps developers **build**, **ship**, and **run** applications in **containers**. A **container** is a lightweight, standalone executable package that includes everything needed to run a piece of software — **code, runtime, libraries, and dependencies**.

There are mainly five states that a container can be in during its lifecycle -

- Created state
- Running state
- Paused state/ Unpaused state
- Stopped state
- Killed/Deleted state





**Fig. Architectural overview of Docker**

Command	Description
<code>docker --version</code>	Check Docker version installed
<code>docker info</code>	Shows detailed info about Docker system
<code>docker login</code>	Login to Docker Hub registry
<code>docker pull &lt;image&gt;</code>	Download an image from Docker Hub (e.g., <code>docker pull nginx</code> )
<code>docker images</code>	List all locally available Docker images
<code>docker rmi &lt;image&gt;</code>	Remove a Docker image from local system
<code>docker build -t &lt;name&gt;</code>	Build an image from a Dockerfile (-t tags the image)
<code>docker run &lt;image&gt;</code>	Create and start a container from an image
<code>docker run -it &lt;image&gt;</code>	Run container in interactive mode with terminal
<code>docker run -d &lt;image&gt;</code>	Run container in detached mode (background)
<code>docker run -p 8080:80 &lt;image&gt;</code>	Map port 80 in container to port 8080 on host
<code>docker ps</code>	List running containers
<code>docker ps -a</code>	List all containers (running + stopped)
<code>docker stop &lt;container&gt;</code>	Stop a running container
<code>docker start &lt;container&gt;</code>	Start a stopped container
<code>docker restart &lt;container&gt;</code>	Restart a container
<code>docker rm &lt;container&gt;</code>	Remove a stopped container
<code>docker logs &lt;container&gt;</code>	View logs of a container
<code>docker exec -it &lt;container&gt; bash</code>	Execute command inside a running container (open terminal)



docker inspect <container/image> Get low-level info (JSON) about a container or image  
docker top <container> Display processes running inside a container

**Check the version of OS: # cat /etc/os-release**

**Step1: Install Docker using the convenience script**  
**Take sudo privileges.**

**# curl -fsSL https://get.docker.com -o get-docker.sh**

```
devasc@labvm:~/Desktop/DOCKER_LAB$ sudo su
root@labvm:/home/devasc/Desktop/DOCKER_LAB# curl -fsSL https://get.docker.com -o get-docker.sh
```

**# sudo sh get-docker.sh**

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# sudo sh get-docker.sh
# Executing docker install script, commit: 0d6f72e671ba87f7aa4c6991646a1a5b9f9dae84
Warning: the "docker" command appears to already exist on this system.

If you already have Docker installed, this script can cause trouble, which is why we're displaying this warning and provide the opportunity to cancel the installation.

If you installed the current Docker package using this script and are using it again to update Docker, you can safely ignore this message.

You may press Ctrl+C now to abort this script.
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# ls
get-docker.sh
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# cat get-docker.sh
```

**Step2: To verify docker is installed or not:**

**# docker version**

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker -v
Docker version 27.1.2, build d01f264
```





## DOCKER COMMANDS:

1] **#docker info:** Give details about docker

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker info
Client: Docker Engine - Community
Version: 27.1.2
Context: default
Debug Mode: false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version: v0.16.2
    Path: /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version: v2.29.1
    Path: /usr/libexec/docker/cli-plugins/docker-compose
Server:
  Containers: 4
    Running: 0
    Paused: 0
    Stopped: 4
  Images: 8
```

2] **#systemctl status docker:** To Check docker engine is active or not

**#systemctl start docker:** To start the docker engine

**#systemctl stop docker :** To stop the docker engine

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-07-28 15:28:05 UTC; 5h 14min left
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 1401 (dockerd)
   Tasks: 11
   Memory: 107.2M
   CGroup: /system.slice/docker.service
           └─1401 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 28 15:28:04 labvm dockerd[1401]: time="2025-07-28T15:28:04.329132221Z" level=info msg="Starting Docker Application Container Engine"
Jul 28 15:28:04 labvm dockerd[1401]: time="2025-07-28T15:28:04.671783428Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm dockerd[1401]: time="2025-07-28T15:28:05.230562423Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm dockerd[1401]: time="2025-07-28T15:28:05.372599545Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm dockerd[1401]: time="2025-07-28T15:28:05.372929307Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm dockerd[1401]: time="2025-07-28T15:28:05.603900488Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm dockerd[1401]: time="2025-07-28T15:28:05.603932240Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm dockerd[1401]: time="2025-07-28T15:28:05.605881468Z" level=info msg="Docker daemon"
Jul 28 15:28:05 labvm systemd[1]: Started Docker Application Container Engine.
```

3] **How you login into your Docker Hub Account from CLI?**

**#docker login**



```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker login
Log in with your Docker ID or email address to push and pull images from Docker
Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to crea
te one.
You can log in with your password or a Personal Access Token (PAT). Using a limi
ted-scope PAT grants better security and is required for organizations using SSO
. Learn more at https://docs.docker.com/go/access-tokens/

Username: 18061977
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
```

4] **docker run:** It helps you to run a container on top of your docker engine, but the ingredients that it needs is image name.

# **docker container run ubuntu cat /etc/os-release**

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container run ubuntu cat /etc/os-rel
ease
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
32f112e3802c: Pull complete
Digest: sha256:a08e551cb33850e4740772b38217fc1796a66da2506d312abe51acda354ff061
Status: Downloaded newer image for ubuntu:latest
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

5] **#docker images :** Lists all images

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    65ae7a6f3544   2 weeks ago   78.1MB
```

6] **# docker container run ubuntu cat /etc/os-release** #Again Hit the same command.





```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container run ubuntu cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
root@labvm:/home/devasc/Desktop/DOCKER_LAB#
```

#### 7) To list docker images: #docker images

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    65ae7a6f3544   2 weeks ago    78.1MB
```

#### 8) To List Container: #docker container ls

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
```

Here the container started and exited at same time.

#### 9) To check container have exited or NOT

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
530ef56a0a90   ubuntu    "cat /etc/os-release"   8 minutes ago   Exited (0) 8 mi
```

Open another terminal:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
baad61dfa5c7   ubuntu    "sleep 60"   28 seconds ago   Up 27 se   stupefied_kepler
101cc947a3e8   ubuntu    "sleep 60"   4 minutes ago    Exited (0) 3 minutes ago   fervent_pike
a1a3a06f8b5e   ubuntu    "cat /etc/os-release..."   5 minutes ago    Exited (0) 1 minutes ago   awesome_jones
530ef56a0a90   ubuntu    "cat /etc/os-release"   18 minutes ago   Exited (0) 18 minutes ago   sharp_perlman
499fb530aa8e   ubuntu    "cat /etc/os-release"   22 minutes ago   Exited (0) 22 minutes ago   boring_herschel
```



### #docker run hello-world

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:53cc4d415d839c98be39331c948609b659ed725170ad2ca8eb36951288f81b75
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

The command used to access the running container is:  
You Can run a ubuntu container with following command:  
#docker run -it ubuntu bash

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
31e907dcc94a: Pull complete
Digest: sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee
Status: Downloaded newer image for ubuntu:latest
root@1d1a286d11ea:/#
```

### 3] How to see the image that I just downloaded whether it is available on my machine or not?

docker images: This command is used to show all the pulled images from docker

# docker images

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
ubuntu              latest         edbfe74c41f8   3 weeks ago    78.1MB
mysql               latest         a82a8f162e18   4 weeks ago    586MB
hello-world         latest         d2c94e258dcb   16 months ago  13.3kB
```

Try to launch a docker image for testing purpose, you can find the images in docker public repository at <https://hub.docker.com>

4] **docker pull:** This command is used to pull images from the docker repository(hub.docker.com)  
Usage: docker pull <image name>

#docker pull mysql





PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

Department of Information Technology

(NBA Accredited)



```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
6e839ac3722d: Pull complete
ad912193ad5f: Pull complete
25d13d87fd8d: Pull complete
004d383c75ef: Pull complete
6d9bbc82a0b8: Pull complete
81fec07ea550: Pull complete
83357cb2d3a5: Pull complete
8ffe968b82c1: Pull complete
30dfd9a7ed57: Pull complete
35844ae33cbe: Pull complete
Digest: sha256:86cdf8e832c81e39a89cfb63c3fde1683c41cc00ef91e67653c9c1df0ba80f454
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

OR

**#docker container run nginx**

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container run 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
/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: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/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
```

**Open another terminal:**

**#docker container run nginx**





```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
3739e1fd10a8	nginx	"/docker-entrypoint..."	34 seconds ago	Up 31 se
conds				
baad61dfa5c7	ubuntu	"sleep 60"	8 minutes ago	Exited (
0) 7 minutes ago				
101cc947a3e8	ubuntu	"sleep 60"	12 minutes ago	Exited (
0) 11 minutes ago				
a1a3a06f8b5e	ubuntu	"cat /etc/os-release..."	13 minutes ago	Exited (
1) 12 minutes ago				
530ef56a0a90	ubuntu	"cat /etc/os-release"	26 minutes ago	Exited (
0) 26 minutes ago				
499fb530aa8e	ubuntu	"cat /etc/os-release"	29 minutes ago	Exited (
0) 29 minutes ago				
81cf4991575a	59ab366372d5	"cat /etc/os-release"	38 minutes ago	Exited (
0) 38 minutes ago				
41aa6f73a6bd	f1e530d55abc	"nginx -g 'daemon of..."	9 months ago	Exited (
255) 6 months ago	0.0.0.0:4032->80/tcp, :::4032->80/tcp			
659dd47038a0	d2c94e258dcb	"/hello"	9 months ago	Exited (
0) 9 months ago				
bc8a1f0b194f	d2c94e258dcb	"/hello"	11 months ago	Exited (
0) 11 months ago				
fbd148039aee	d2c94e258dcb	"/hello"	11 months ago	Exited (

Ctrl C to come out of the current screen and again check whether nginx image container is exited.

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
3739e1fd10a8	nginx	"/docker-entrypoint..."	4 minutes ago	Exited (
0) 13 seconds ago				
baad61dfa5c7	ubuntu	"sleep 60"	12 minutes ago	Exited (

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	2cd1d97f893f	13 days ago	192MB
ubuntu	latest	65ae7a6f3544	2 weeks ago	78.1MB

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB#
```

5) **docker ps** : This command lists the running containers on my system

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps
```

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

6) **docker ps -a** : This command list all containers running or exited from the system.

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
1d1a286d11ea	ubuntu	"bash"	12 minutes ago	Exited (0) 8 minutes ago
o	sleepy_cerf			
fbd148039aee	hello-world	"/hello"	22 minutes ago	Exited (0) 22 minutes ago
go	infallible_cerf			

Now to get the container running : **#docker run -it ubuntu bash**

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it ubuntu bash
root@a0baee026da8:/#
```



In New terminal: docker ps -a

```
devasc@labvm:~/Desktop/DOCKER_LAB$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES
a0baee026da8   ubuntu    "bash"                  39 seconds ago   Up 39 seconds
1d1a286d11ea   ubuntu    "bash"                  18 minutes ago   Exited (0) 13 minutes a
go             sleepy_cerf
fbd148039aee   hello-world "/hello"                27 minutes ago   Exited (0) 27 minutes a
go             infallible_cerf
```

In First Terminal: ls

You will see the lists of directories available in ubuntu container

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it ubuntu bash
root@19151f20756a:/#
root@19151f20756a:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
```

Now get exit from ubuntu container

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it ubuntu bash
root@a0baee026da8:/# exit
exit
```

In New terminal: docker ps -a

So You see the container has exited 3 minutes ago.

```
devasc@labvm:~/Desktop/DOCKER_LAB$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES
a0baee026da8   ubuntu    "bash"                  3 minutes ago   Exited (0) 6 seconds ag
o             friendly_easley
```

NOTE: Every container created has a unique container id. That is, from a single image multiple containers can be created. Also, every container will be independent of itself, will be isolated from other container

NOTE: If you don't provide name to your container , the docker-engine gives fancy name to your container.

7] docker exec: This command is used to executes the container.

Usage: `docker exec -it <container id> bash`

8] To delete the container: `#docker rm <container-name/container-id>`

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker rm 191
191
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker rm a0b 1d1
a0b
1d1
```





```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
fbfd148039aee	hello-world	"/hello"	48 minutes ago	Exited (0) 48 minutes ago
go	infallible/cerf			

9] To delete the image: `#docker rmi <image-name/image-id>`

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	edbfe74c41f8	3 weeks ago	78.1MB
mysql	latest	a82a8f162e18	4 weeks ago	586MB
hello-world	latest	d2c94e258dcb	16 months ago	13.3kB

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker rmi edb
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee
Deleted: sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a
Deleted: sha256:f36fd4bb7334b7ae3321e3229d103c4a3e7c10a263379cc6a058b977edfb46de
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	latest	a82a8f162e18	4 weeks ago	586MB
hello-world	latest	d2c94e258dcb	16 months ago	13.3kB

10] Commands related to containers:

`docker start <container-id>` : Start the stop container  
`docker stop <container-id>` : Stop the running container  
`docker pause <container-id>` : Pause the processes in running container  
`docker kill <container id>` : Kill the container.

## Task 2: HOW TO SETUP AND CONFIGURE MYSQL DATABASE INSIDE DOCKER CONTAINER?

MySQL is the single most popular relational database tool.

MySQL is popular because it is simple yet powerful. Here are its best features:

- **Relational:** follows the relational model and uses SQL to manage databases.
- **Open-source (GNU license):** the community loves it. Companies love it.
- **Scalable:** can handle applications from small-sized to enterprise-level.
- **Secure:** offers user authentication, access management, and encryption.
- **High-performance:** known for its speed and efficiency in handling complex queries and large volumes of data.
- **Replication and backup:** it has options for data replication and backup, allowing for disaster recovery strategies.

**Step 1: To pull the image of mysql from docker hub**





#docker pull mysql:latest

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
6e839ac3722d: Pull complete
ad912193ad5f: Pull complete
25d13d87fd8d: Pull complete
004d383c75ef: Pull complete
6d9bbc82a0b8: Pull complete
81fec07ea550: Pull complete
83357cb2d3a5: Pull complete
8ffe968b82c1: Pull complete
30dfd9a7ed57: Pull complete
35844ae33cbe: Pull complete
Digest: sha256:86cdf8e832c81e39a89cfb63c3fde1683c41cc00ef91e67653c9c1df0ba80f454
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

**EXPLANATION:** The code is a Docker command, not SQL.

- It's used to download the latest version of the MySQL Docker image.
- "docker pull" is a command that tells Docker to download an image from Docker Hub.
- "mysql:latest" specifies the image to download.
- "mysql" is the name of the image and "latest" is the tag.

**Step 2: List the mysql images**

#docker images

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
mysql                latest       a82a8f162e18     4 weeks ago     586MB
hello-world          latest       d2c94e258dcb     16 months ago   13.3kB
```

Docker images are blueprints for building containers. Just like a blueprint allows you to build a house, a Docker image contains all the necessary instructions and components to create a running instance of an application or service.

**STEP 3: Running and Managing a MySQL Server Container**

Now, let's create our first container from the mysql image. Here is the command we will use:

```
$ docker run --name test-mysql -e MYSQL_ROOT_PASSWORD=strong_password -d mysql
```

**EXPLANATION:**

- ✓ **run:** creates a new container or starts an existing one
- ✓ **--name CONTAINER\_NAME:** gives the container a name. The name should be readable and short. In our case, the name is test-mysql.
- ✓ **-e ENV\_VARIABLE=value:** the -e tag creates an environment variable that will be accessible within the container. It is crucial to set **MYSQL\_ROOT\_PASSWORD** so that we can run SQL commands later



from the container. Make sure to store your strong password somewhere safe (not your brain).

- ✓ -d: short for detached, the -d tag makes the container run in the background. If you remove this tag, the command will keep printing logs until the container stops.
- ✓ image\_name: the final argument is the image name the container will be built from. In this case, our image is mysql.
- ✓

If the command returns a long string of gibberish (the container ID), it means the container has started. You can check its status with docker ps:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run --name test-mysql -e MYSQL_ROOT_PASSWORD=sujata -d mysql
0e2b27de5979ae2bfa7c4e464841157796500458dfb194c18bd4716b63440de4
```

In New Terminal:

```
devasc@labvm:~/Desktop/DOCKER_LAB$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
0e2b27de5979	mysql	"docker-entrypoint.s..."	35 seconds ago	Up 34 seconds
3306/tcp		test-mysql		

Step 4: To access the terminal inside your container, you can use the following command:

```
$ docker exec -it container_name bash
```

This will launch a bash session.

Connecting to the MySQL Server Container Locally :

All MySQL containers launch a MySQL server that includes everything to create and manage databases using SQL. To connect to the server, containers also come with a MySQL client that lets us run SQL queries. The client is just a fancy name for the mysql terminal command. Let's use it inside test-mysql's terminal:

1. Open the bash terminal of test-mysql:

```
$ docker exec -it test-mysql bash
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker exec -it test-mysql bash
bash-5.1#
```

2. Connect to the client as a root user:



We are using the -u tag to specify the username (root) and adding the -p tag to enter the password when prompted.

```
$ mysql -u root -p
Enter password: ...
mysql>
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker exec -it test-mysql bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

```
mysql> SELECT 'hello-world!!!';
```

```
mysql> SELECT 'hello-world!!!';
+-----+
| hello-world!!! |
+-----+
| hello-world!!! |
+-----+
1 row in set (0.00 sec)
```

```
mysql> show databases;
```





```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

```
mysql> SELECT DATABASE();
+-----+
| DATABASE() |
+-----+
| mysql |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SHOW TABLES;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| component |
| db |
| default_roles |
| engine_cost |
| func |
| general_log |
| global_grants |
| gtid_executed |
+-----+
```



```
mysql> DESCRIBE db;
```

Field	Type	Null	Key	Default	Extra
Host	char(255)	NO	PRI		
Db	char(64)	NO	PRI		
User	char(32)	NO	PRI		
Select_priv	enum('N','Y')	NO		N	
Insert_priv	enum('N','Y')	NO		N	
Update_priv	enum('N','Y')	NO		N	
Delete_priv	enum('N','Y')	NO		N	
Create_priv	enum('N','Y')	NO		N	
Drop_priv	enum('N','Y')	NO		N	
Grant_priv	enum('N','Y')	NO		N	
References_priv	enum('N','Y')	NO		N	
Index_priv	enum('N','Y')	NO		N	

```
mysql> SELECT NOW() ;
```

NOW()
2024-08-25 18:10:04

1 row in set (0.00 sec)

**Conclusion:** In this experiments student have learnt how to deal with containerization technology using various docker commands.