

Name: Abdurrahman Qureshi

Roll No: 242466

Practical No: 10

Date Of Performance: 10/09/2025

Aim: To understand Docker Hub's purpose, install Docker on an AWS EC2 Ubuntu instance, work with container images, and learn essential Docker commands for container management.

1. What is hub.docker.com?
2. What is docker hub used for?
3. Install docker on AWS EC2 –Ubuntu by using curl

```
#curl -fsSLhttps://get.docker.com -o get-docker.sh
```

```
#sh get-docker.sh
```
4. Run hello-world from docker hub and explain the steps
5. Pull 3 or 4 images ,one of the python , run " Hello World " inside container.
6. Demonstrate any 15 docker command and explain its uses

[Terminate the resources after performing the practical-
terminate environment and application both]

ANS.1:

hub.docker.com, commonly known as Docker Hub, is the world's largest public repository and community for container images. Think of it as:

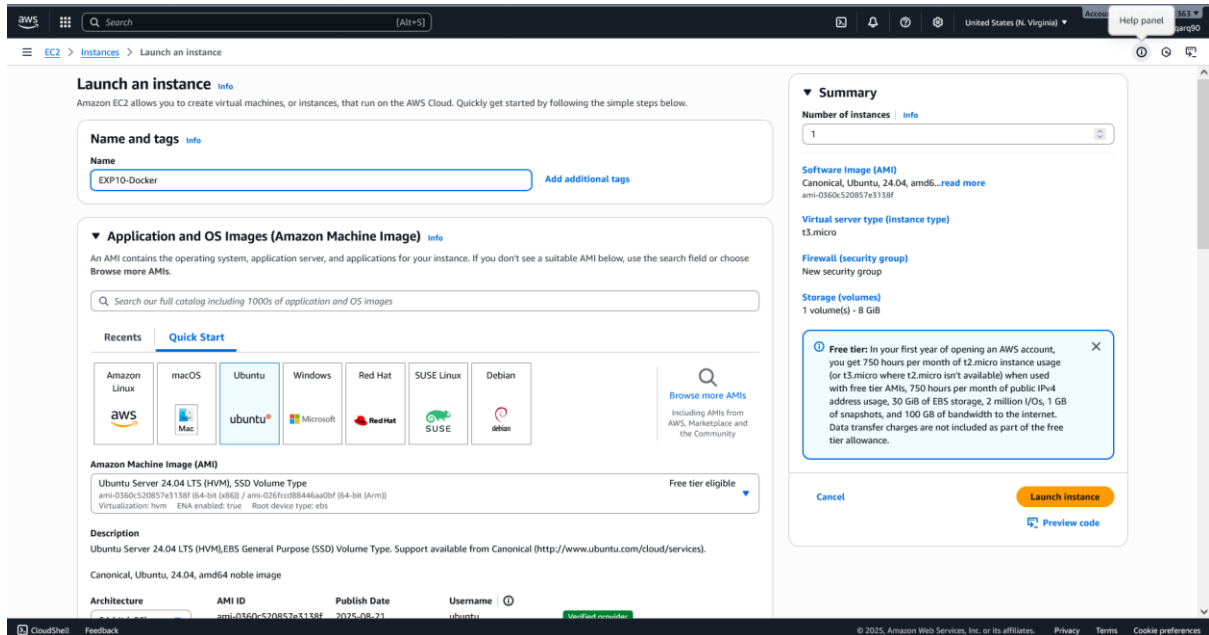
- GitHub for Docker Images: Just as GitHub hosts code, Docker Hub hosts pre-built, ready-to-run application packages called container images.
- A Public Library: It's a central, cloud-based registry where developers and companies can store, share, and distribute their container images.
- A Source of Official Software: Major software vendors like Ubuntu, Python, Node.js, Nginx, Redis, and MySQL publish their official, verified images on Docker Hub. This ensures you get a trusted, standard version of the software.

ANS.2:

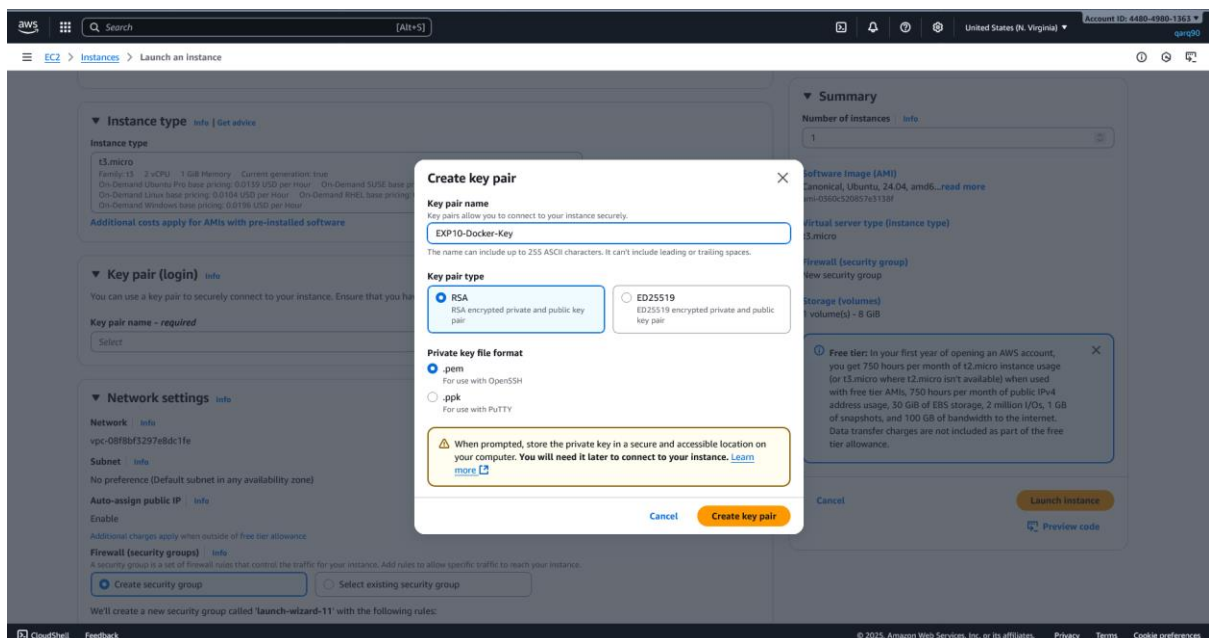
Docker Hub serves several critical purposes in the container ecosystem:

- Image Storage: It provides a centralized place to store your public and private Docker images.
- Image Discovery: Developers can search for and discover pre-built images for almost any software, saving immense time and effort.
- Image Distribution: It facilitates the easy pulling docker pull and pushing docker push of images to and from any machine running Docker.
- Collaboration: Teams can use Docker Hub to share images privately within their organization or publicly with the world.

ANS.3:



Setting up the EC2 Instance



Creating Key Pair


```
ubuntu@ip-172-31-29-53: ~$  
ED25519 key fingerprint is SHA256:EeSc1XFJqiQfPm8Nteg3B6SX3KF4Lv4MF1raarhz0gY.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-184-73-58-25.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Sat Sep 13 04:03:16 UTC 2025  
  
System load: 0.04 Temperature: -273.1 C  
Usage of /: 25.6% of 6.7168 Processes: 116  
Memory usage: 23% Users logged in: 0  
Swap usage: 0% IPv4 address for ens5: 172.31.29.53  
  
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  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-29-53:~$
```

Connected to Instance

```
ubuntu@ip-172-31-29-53:~$ curl -fsSL https://get.docker.com -o get-docker.sh  
ubuntu@ip-172-31-29-53:~$ sudo sh get-docker.sh  
# Executing docker install script, commit: 5c8855edd778525544500337f5ac4ad65a0c168e  
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 ignore this message, but be aware that the  
script resets any custom changes in the deb and rpm repo configuration  
files to match the parameters passed to the script.  
  
You may press Ctrl+C now to abort this script.  
+ sleep 20  
+ sh -c apt-get -qq update >/dev/null  
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/dev/null  
+ sh -c install -m 0755 -d /etc/apt/keyrings  
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" -o /etc/apt/keyrings/docker.asc  
+ sh -c chmod a+r /etc/apt/keyrings/docker.asc  
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu noble stable" > /etc/apt/sources.list.d/docker.list  
+ sh -c apt-get -qq update >/dev/null  
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-c  
e-rootless-extras docker-buildx-plugin docker-model-plugin >/dev/null  
+ sh -c docker version  
Client: Docker Engine - Community  
Version: 28.4.0  
API version: 1.51  
Go version: go1.24.7  
Git commit: d8eb465  
Built: Wed Sep 3 20:57:32 2025  
OS/Arch: linux/amd64  
Context: default  
  
Server: Docker Engine - Community  
Engine:  
Version: 28.4.0  
API version: 1.51 (minimum version 1.24)
```

Installing Docker

```
ubuntu@ip-172-31-29-53: ~  
Server: Docker Engine - Community  
Engine:  
  Version:      28.4.0  
  API version:  1.51 (minimum version 1.24)  
  Go version:   go1.24.7  
  Git commit:   249d679  
  Built:        Wed Sep 3 20:57:32 2025  
  OS/Arch:      linux/amd64  
  Experimental: false  
containerd:  
  Version:      1.7.27  
  GitCommit:    85044ec0a9a75232cad458027ca83437aae3f4da  
runc:  
  Version:      1.2.5  
  GitCommit:    v1.2.5-0-g59923ef  
docker-init:  
  Version:      0.19.0  
  GitCommit:    de40ad0  
  
=====
```

To run Docker as a non-privileged user, consider setting up the Docker daemon in rootless mode for your user:

```
dockerd-rootless-setuptool.sh install
```

Visit <https://docs.docker.com/go/rootless/> to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root users access, refer to <https://docs.docker.com/go/daemon-access/>

WARNING: Access to the remote API on a privileged Docker daemon is equivalent to root access on the host. Refer to the 'Docker daemon attack surface' documentation for details: <https://docs.docker.com/go/attack-surface/>

```
=====
```

ubuntu@ip-172-31-29-53:~\$

Configuring Docker

```
ubuntu@ip-172-31-29-53:~$ sudo usermod -aG docker ubuntu  
ubuntu@ip-172-31-29-53:~$ newgrp docker  
ubuntu@ip-172-31-29-53:~$ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
17eec7bb9d7: Pull complete  
Digest: sha256:54e66c1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31  
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.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
  
ubuntu@ip-172-31-29-53:~$
```

Configuration Complete

```
ubuntu@ip-172-31-29-53:~$ docker pull python:3.9-slim
3.9-slim: Pulling from library/python
ce1261c6d567: Pull complete
1d454ace0e38: Pull complete
41dc2499d8fe: Pull complete
7fcd9369fa9: Pull complete
Digest: sha256:71b97a8eed69cddd132431327ff7c204fd6ea3d38e0c5d11d52f9661482bc8a3
Status: Downloaded newer image for python:3.9-slim
docker.io/library/python:3.9-slim
ubuntu@ip-172-31-29-53:~$ docker pull nginx:alpine
alpine: Pulling from library/nginx
9824c27679d3: Pull complete
6bc572a340ec: Pull complete
403e3f251637: Pull complete
9adfbae99cb7: Pull complete
7a8a46741e18: Pull complete
c9ebe2ff2d2c: Pull complete
a992fbc61ecc: Pull complete
cb1ff4086f82: Pull complete
Digest: sha256:42a516af16b852e33b7682d5ef8acbd5d13fe08fecadc7ed98605ba5e3b26ab8
Status: Downloaded newer image for nginx:alpine
docker.io/library/nginx:alpine
ubuntu@ip-172-31-29-53:~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
9824c27679d3: Already exists
Digest: sha256:4bcff63911fcb4448bd4fdacc207030997caf25e9bea4045fa6c8c44de311d1
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
ubuntu@ip-172-31-29-53:~$ docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
13b7e930469f: Pull complete
Digest: sha256:8feb4d8ca5354def3d8fca243717141ce31e2c428701f6682bd2fafa15388214
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
ubuntu@ip-172-31-29-53:~$
```

Pulling some images

```
ubuntu@ip-172-31-29-53:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                alpine             4a86014ec699       4 weeks ago        52.5MB
hello-world          latest             1b44b5a3e06a       5 weeks ago        10.1kB
python               3.9-slim           56cea0119ab6       5 weeks ago        120MB
alpine               latest             9234e8fb04c4       8 weeks ago        8.31MB
ubuntu               20.04              b7bab04fd9aa       5 months ago       72.8MB
ubuntu@ip-172-31-29-53:~$
```

Images installed so far

```
ubuntu@ip-172-31-29-53:~$ docker run python:3.9-slim python -c 'print("Hello World from inside a Python container!")'
Hello World from inside a Python container!
ubuntu@ip-172-31-29-53:~$
```

Running a simple python code

Docker Commands:

```
ubuntu@ip-172-31-29-53:~$ docker run -d -p 8080:80 --name my_web_server nginx:alpine
00dc418b372394e97a52eebb1aaffc2a10e4dc7bf13fd70c2d3ba46692f5bf5a
```

Run a container in the background

```
ubuntu@ip-172-31-29-53:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAME
00dc418b3723	nginx:alpine	"/docker-entrypoint..."	8 seconds ago	Up 8 seconds	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	my_web_server

List running containers

```
ubuntu@ip-172-31-29-53:~$ docker logs my_web_server
/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
2025/09/13 04:11:12 [notice] 1#1: using the "epoll" event method
2025/09/13 04:11:12 [notice] 1#1: nginx/1.29.1
2025/09/13 04:11:12 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/09/13 04:11:12 [notice] 1#1: OS: Linux 6.14.0-1011-aws
2025/09/13 04:11:12 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/09/13 04:11:12 [notice] 1#1: start worker processes
2025/09/13 04:11:12 [notice] 1#1: start worker process 30
2025/09/13 04:11:12 [notice] 1#1: start worker process 31
ubuntu@ip-172-31-29-53:~$
```

View container logs

```
ubuntu@ip-172-31-29-53:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAME
00dc418b3723	nginx:alpine	"/docker-entrypoint..."	About a minute ago	Up About a minute	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	my_web_server
6f009dd6d9ed	python:3.9-slim	"python -c 'print(\"H..."	2 minutes ago	Exited (0) 2 minutes ago		zen_raman
4373f6319c41	hello-world	"/hello"	6 minutes ago	Exited (0) 6 minutes ago		beautiful_herschel

List all containers, including stopped ones


```
ubuntu@ip-172-31-29-53:~$ docker run -it --name demo-container ubuntu:20.04 /bin/bash
root@d3cbea410a67:/# apt-get update
```

Create a new container

```
ubuntu@ip-172-31-29-53:~$ docker start demo-container
demo-container
```

Start a container

```
ubuntu@ip-172-31-29-53:~$ docker --version
Docker version 28.4.0, build d8eb465
```

Check Docker Version

```
ubuntu@ip-172-31-29-53:~$ docker pull nginx:alpine
alpine: Pulling from library/nginx
Digest: sha256:42a516af16b852e33b7682d5ef8acbd5d13fe08fecadc7ed98605ba5e3b26ab8
Status: Image is up to date for nginx:alpine
docker.io/library/nginx:alpine
```

Download an image from a registry like Docker Hub without running it.

```
ubuntu@ip-172-31-29-53:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
c7158f8a12db	ubuntu:20.04	"sleep 300"	54 seconds ago	Up 54 seconds	
my-test-container					
d3cbea410a67	ubuntu:20.04	"/bin/bash"	8 minutes ago	Up 7 minutes	
demo-container					
00dc418b3723	nginx:alpine	"/docker-entrypoint..."	11 minutes ago	Up 11 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
my_web_server					

```
ubuntu@ip-172-31-29-53:~$ docker stop my-test-container
my-test-container
ubuntu@ip-172-31-29-53:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
c7158f8a12db	ubuntu:20.04	"sleep 300"	About a minute ago	Exited (137) 6 seconds ago	
my-test-container					
d3cbea410a67	ubuntu:20.04	"/bin/bash"	9 minutes ago	Up 8 minutes	
demo-container					
00dc418b3723	nginx:alpine	"/docker-entrypoint..."	12 minutes ago	Up 12 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
my_web_server					
6f009dd6d9ed	python:3.9-slim	"python -c 'print(\"H..."	13 minutes ago	Exited (0) 13 minutes ago	
zen_raman					
4373f6319c41	hello-world	"/hello"	16 minutes ago	Exited (0) 16 minutes ago	
beautiful_herschel					

Stop a Container

```
ubuntu@ip-172-31-29-53:~$ docker rm my-test-container
my-test-container
ubuntu@ip-172-31-29-53:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
d3cbea410a67	ubuntu:20.04	"/bin/bash"	9 minutes ago	Up 9 minutes	
demo-container					
00dc418b3723	nginx:alpine	"/docker-entrypoint..."	12 minutes ago	Up 12 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
my_web_server					
6f009dd6d9ed	python:3.9-slim	"python -c 'print(\"H..."	13 minutes ago	Exited (0) 13 minutes ago	
zen_raman					
4373f6319c41	hello-world	"/hello"	17 minutes ago	Exited (0) 17 minutes ago	
beautiful_herschel					

Remove a Container

```
ubuntu@ip-172-31-29-53:~$ docker rmi alpine
Untagged: alpine:latest
Untagged: alpine@sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1
Deleted: sha256:9234e8fb04c47cfe0f49931e4ac7eb76fa904e33b7f8576aec0501c085f02516
```

Remove an Image

```
ubuntu@ip-172-31-29-53:~$ docker create --name my_created_container ubuntu:20.04 echo "Hello"
4a71778960eb79ee77f988164389e130c097f3d183d2f3b516525e7c6f477f00
ubuntu@ip-172-31-29-53:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
4a71778960eb	ubuntu:20.04	"echo Hello"	5 seconds ago	Created	
my_created_container					
d3cbea410a67	ubuntu:20.04	"/bin/bash"	11 minutes ago	Up 10 minutes	
demo-container					
00dc418b3723	nginx:alpine	"/docker-entrypoint..."	14 minutes ago	Up 14 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
my_web_server					
6f009dd6d9ed	python:3.9-slim	"python -c 'print(\"H..."	15 minutes ago	Exited (0) 15 minutes ago	
zen_raman					
4373f6319c41	hello-world	"/hello"	19 minutes ago	Exited (0) 19 minutes ago	
beautiful_herschel					

Creates a new container from an image but does not start it

```
ubuntu@ip-172-31-29-53:~$ docker top demo-container
```

UID	CMD	PID	PPID	C	STIME	TTY	TIME
root	/bin/bash	4151	4129	0	04:15	pts/0	00:00:00

Display the running processes of a container

```
ubuntu@ip-172-31-29-53:~$ docker search mysql
```

NAME	DESCRIPTION	STARS	OFFICIAL
mysql	MySQL is a widely used, open-source relation...	15911	[OK]
bitnami/mysql	Bitnami container image for MySQL	141	
circleci/mysql	MySQL is a widely used, open-source relation...	32	
bitnamicharts/mysql	Bitnami Helm chart for MySQL	0	
cimg/mysql		3	
ubuntu/mysql	MySQL open source fast, stable, multi-thread...	70	
google/mysql	MySQL server for Google Compute Engine	26	
linuxserver/mysql	A Mysql container, brought to you by LinuxSe...	48	
elestio/mysql	Mysql, verified and packaged by Elestio	1	
docksal/mysql	MySQL service images for Docksal - https://d...	0	
alpine/mysql	mysql client	6	
eclipse/mysql	Mysql 5.7, curl, rsync	2	
datajoint/mysql	MySQL image pre-configured to work smoothly ...	2	
ilios/mysql	Mysql configured for running Ilios	1	
ddev/mysql	ARM64 base images for ddev-dbserver-mysql-8...	1	
mirantis/mysql		0	
mysql/mysql-server	Optimized MySQL Server Docker images. Create...	1032	
corpusops/mysql	https://github.com/corpusops/docker-images/	0	
vulnhub/mysql		1	
cbioportal/mysql	This repository hosts MySQL database images ...	1	
vitess/mysql	Lightweight image to run MySQL with Vitess	1	
mysql/mysql-router	MySQL Router provides transparent routing be...	28	
mysql/mysql-cluster	Experimental MySQL Cluster Docker images. Cr...	100	
nasqueron/mysql		1	
mysql/mysql-operator	MySQL Operator for Kubernetes	1	

Search Docker Hub for images

```
ubuntu@ip-172-31-29-53:~$ docker history python:3.9-slim
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
56cea0119ab6	5 weeks ago	CMD ["python3"]	0B	buildkit.dockerfile.v0
<missing>	5 weeks ago	RUN /bin/sh -c set -eux; for src in idle3 p...	36B	buildkit.dockerfile.v0
<missing>	5 weeks ago	RUN /bin/sh -c set -eux; savedAptMark="\$(a...	38MB	buildkit.dockerfile.v0
<missing>	5 weeks ago	ENV PYTHON_SHA256=61a42919e13d539f7673cf11d1...	0B	buildkit.dockerfile.v0
<missing>	5 weeks ago	ENV PYTHON_VERSION=3.9.23	0B	buildkit.dockerfile.v0
<missing>	5 weeks ago	ENV GPG_KEY=E3FF2839C048B25C084DEBE9B26995E3...	0B	buildkit.dockerfile.v0
<missing>	5 weeks ago	RUN /bin/sh -c set -eux; apt-get update; a...	3.81MB	buildkit.dockerfile.v0
<missing>	5 weeks ago	ENV LANG=C.UTF-8	0B	buildkit.dockerfile.v0
<missing>	5 weeks ago	ENV PATH=/usr/local/bin:/usr/local/sbin:/usr...	0B	buildkit.dockerfile.v0
<missing>	5 weeks ago	# debian.sh --arch 'amd64' out/ 'trixie' '@1...	78.6MB	debuerreotype 0.16

Show an image' s history