# Docker - 1

### Difference between Virtualization vs Containerization vs Bare Metal.

**Bare Metal:**

Refers to physical hardware (servers) without any layer of virtualization or abstraction. Applications run directly on the hardware using an operating system. It's very fast and efficient but lacks flexibility—each server is dedicated to a specific task, and scaling is harder.

**Virtualization:**

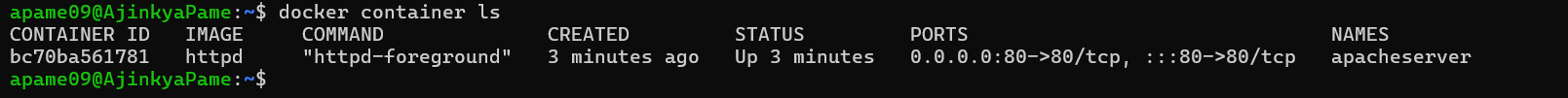
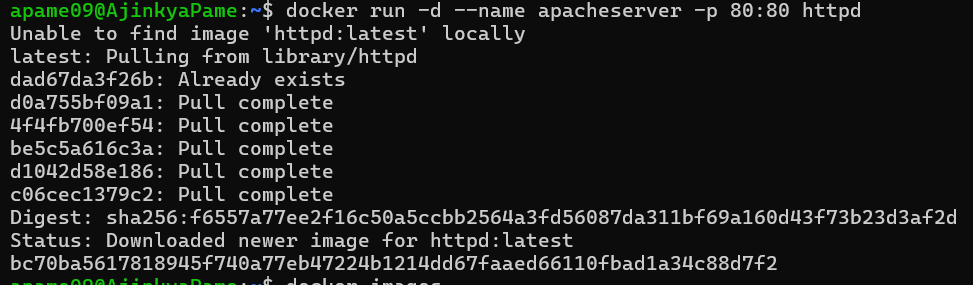
Uses a hypervisor (like VMware, Hyper-V, or KVM) to run multiple virtual machines (VMs) on a single physical server. Each VM has its own OS, which makes it more flexible than bare metal. However, running full OSs on each VM requires more resources (CPU, memory, disk).

**Containerization:**

Runs multiple lightweight applications (containers) on a shared OS using tools like Docker. Containers are faster and use fewer resources than VMs because they share the same operating system kernel, making them ideal for microservices and scalable cloud applications.

### **Task 01:** Create an httpd (apache) container, inside that container create 2 more html pages (eg. home.html and about.html) which will be accessible from the browser.

* Pull the image from DockerHub and create a httpd container using “docker run” command:  
  docker run -d --name apacheserver -p 80:80 httpd

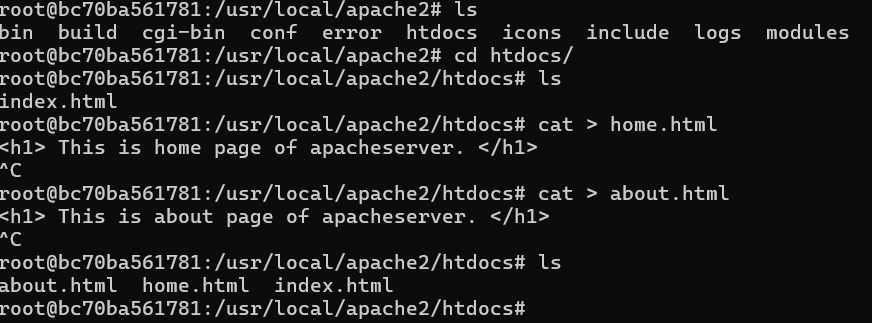


* Enter the container:

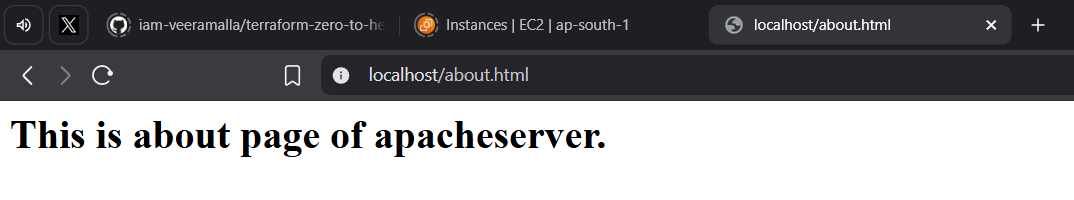
apame09@AjinkyaPame:~$ docker exec -it apacheserver /bin/bash

root@bc70ba561781:/usr/local/apache2#

* Create home.html and about.html page:



* Access the pages on browser:

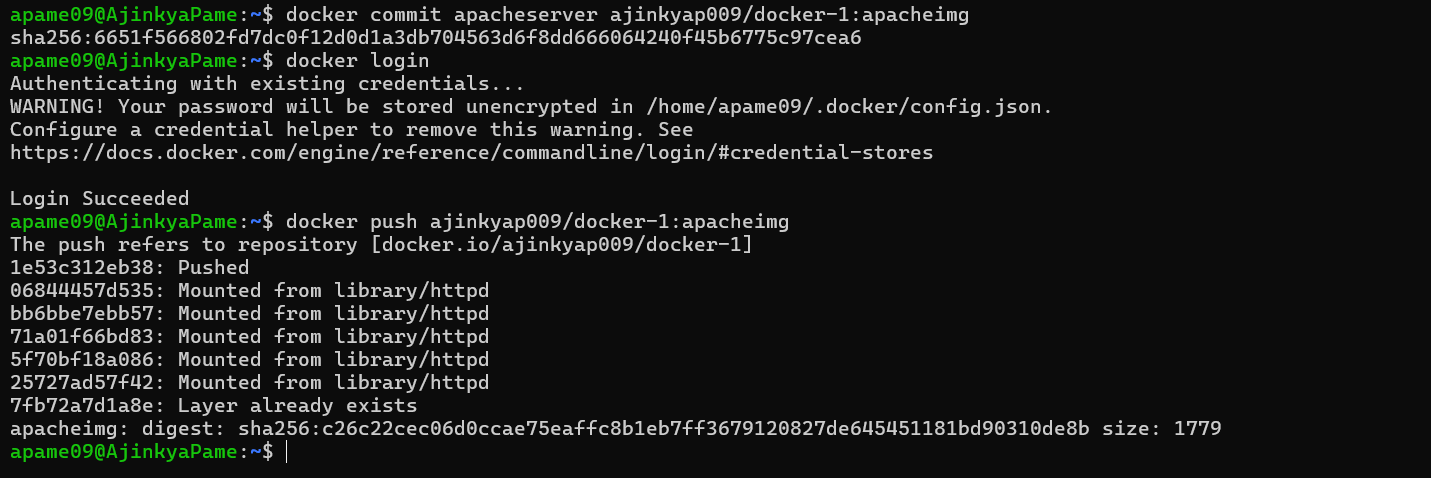


### **Task 02:** Create image from httpd container created from above question and push that image to DockerHub and Amazon ECR.

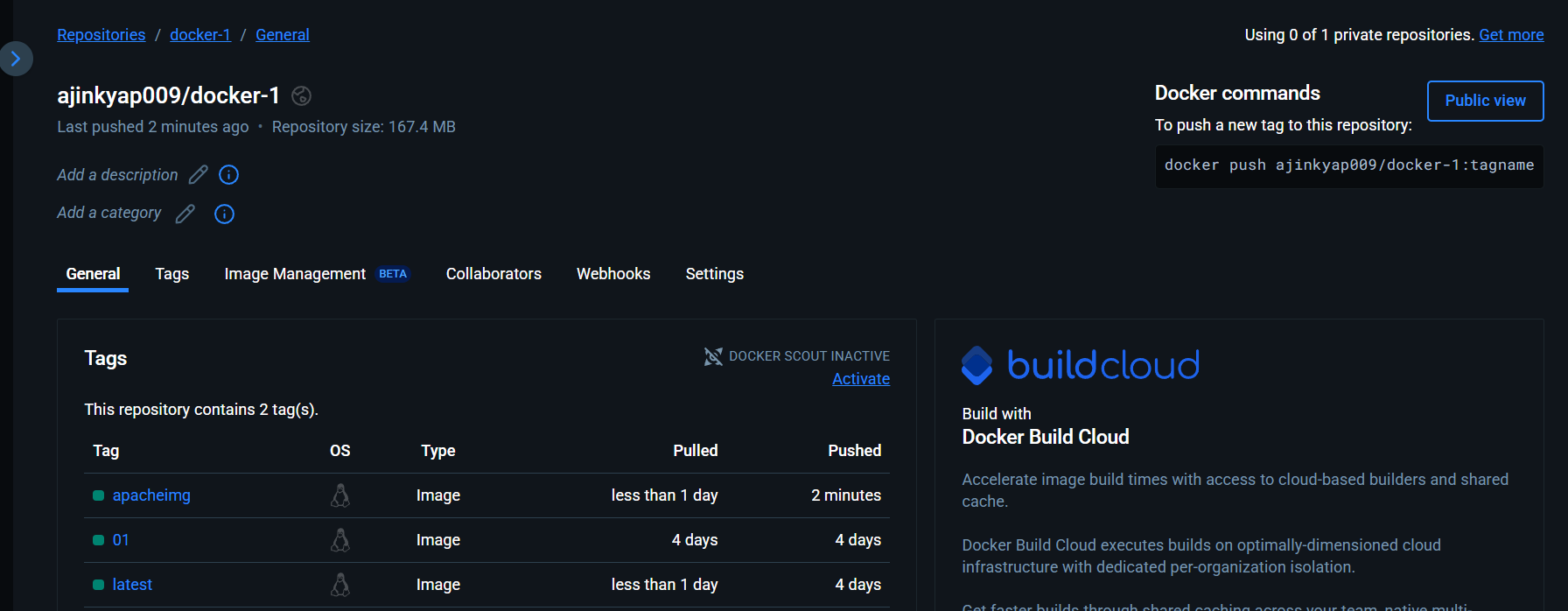
**a. Docker hub:**

* Use docker commit to create image from container. Use image name in format as “username/reponame:tag”.
* Check if you are logged in to DockerHub from terminal using docker login. “Login Succeeded means login is done.
* Push the image to repository at DockerHub using docker command:

docker push imagename

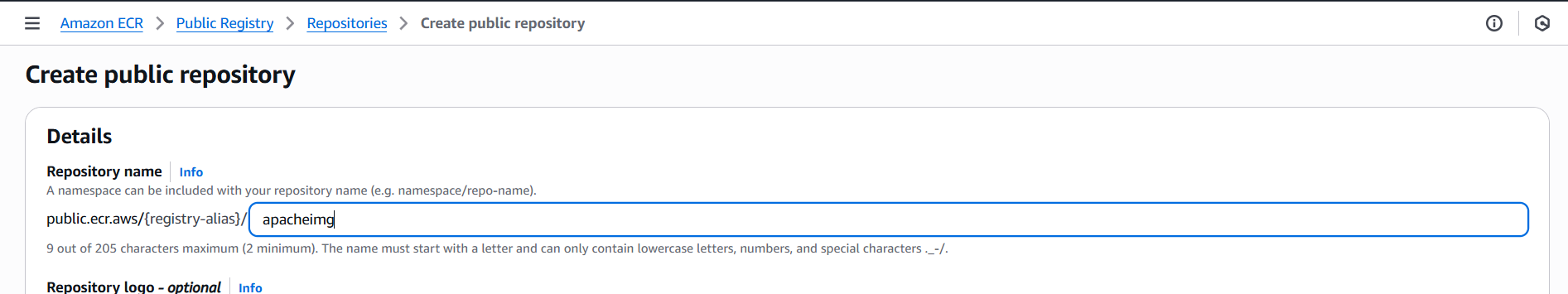


* The image with the tag is present in the repository at DockerHub.

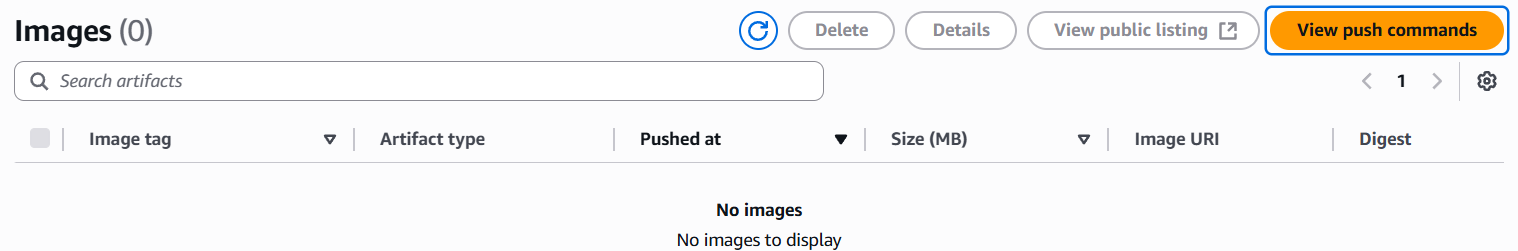


### **b. ECR:**

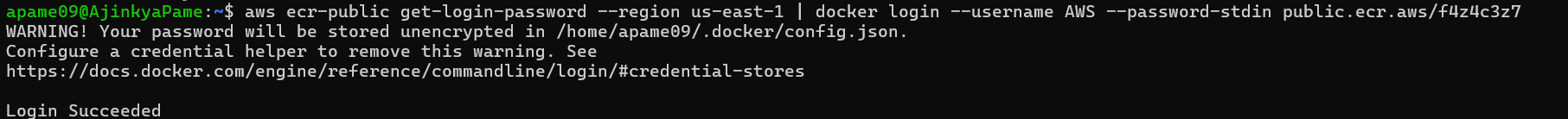
* Install AWS CLI in the terminal.
* Check for the version: aws –version
* Create a repository in AWS ECR.



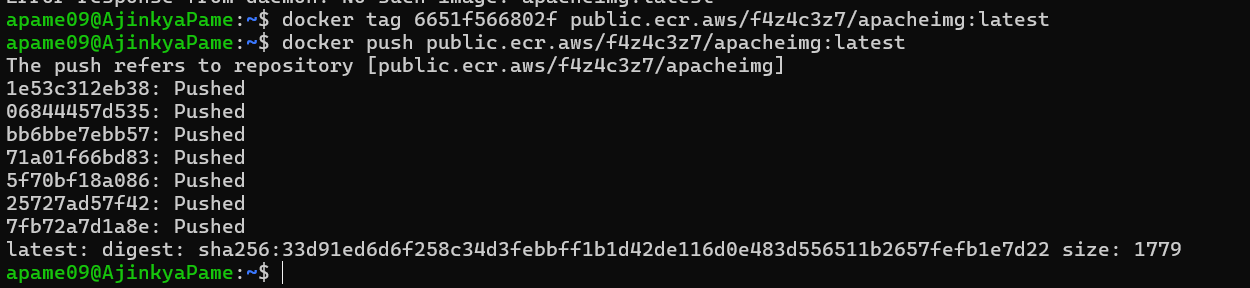
* Check for easy commands to push the image to the public repository we created.



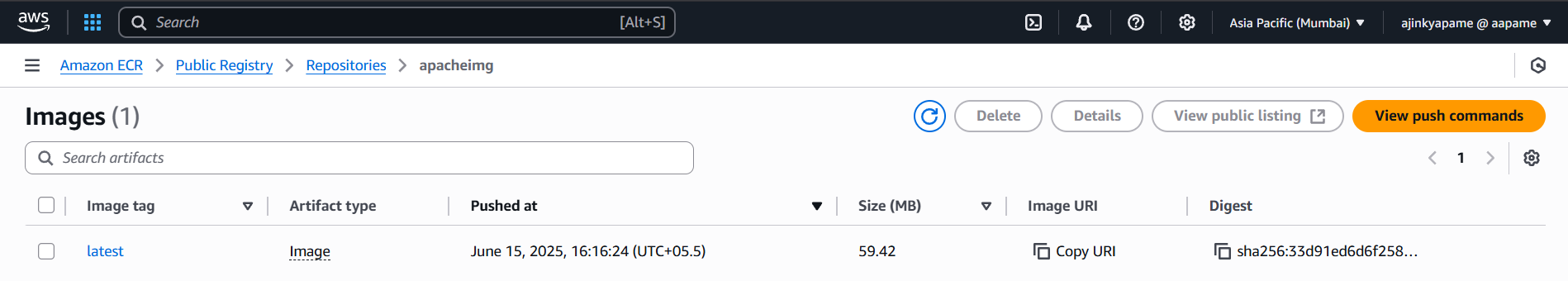
* Run the command to login to the ECR:



* Tag the image and push to the ECR.



* The image is visible at the console.



### List of some docker commands with description and example usage

#### **1.** *docker -–version***: check docker installation and version.**

> apame09@AjinkyaPame:~$ docker --version

Docker version 27.5.1, build 27.5.1-0ubuntu3~24.04.2

**2.** docker pull imagename**: Pull image from DockerHub if present.**

> apame09@AjinkyaPame:~$ docker pull nginx

Using default tag: latest

latest: Pulling from library/nginx

Digest: sha256:6784fb0834aa7dbbe12e3d7471e69c290df3e6ba810dc38b34ae33d3c1c05f7d

Status: Downloaded newer image for nginx:latest

docker.io/library/nginx:latest

#### **3.** *docker images***: List the present images.**

> apame09@AjinkyaPame:~$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest 1e5f3c5b981a 8 weeks ago 192MB

**4.** docker run -d –-name cont-name -p hostport:containerport imagename: **Create container from image, give it a name and bind** **container’s port to host port in detached mode.**

> apame09@AjinkyaPame:~$ docker run -d --name nginxcont -p 90:80 nginx

8f9a963608be98890fc5b4fdc0d67227248c09984e5d8acc82b477aa5724ed71

**5.** docker exec -it cont-name /bin/bash**: Enter container in interactive mode with a terminal of bash.**

> apame09@AjinkyaPame:~$ docker exec -it nginxcont /bin/bash

root@8f9a963608be:/# service nginx status

nginx is running.

***5.*** *docker ps -a****: List all running containers.***

> apame09@AjinkyaPame:~$ docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

8f9a963608be nginx "/docker-entry” 7 minutes ago Up 7 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp nginxcont

***6.*** *docker stop cont-name****: Stop a running container.***

> apame09@AjinkyaPame:~$ docker stop nginxcont

nginxcont

***7.*** *docker start cont-name****: Start stopped container.***

> apame09@AjinkyaPame:~$ docker start nginxcont

nginxcont

***8.*** *docker rm cont-name****: Remove a stopped container.***

> apame09@AjinkyaPame:~$ docker rm nginxcont

nginxcont

***9.*** *docker rmi imagename****: Remove an image which is not used by any container.***

> apame09@AjinkyaPame:~$ docker rmi nginx:latest

Untagged: nginx:latest

***10.*** *docker volume ls****: List the docker volumes.***

> DRIVER VOLUME NAME

local mysql

***11.*** *docker volume create name****: Create a volume***

> apame09@AjinkyaPame:~$ docker volume create myvolume

myvolume

***12.*** *docker network ls****: List the docker networks.***

> apame09@AjinkyaPame:~$ docker network ls

NETWORK ID NAME DRIVER SCOPE

192ea7819477 app\_db bridge local

b9dcad3f82dc bridge bridge local

***13.*** *docker network create network-name –-driver driver-name****:******Create a network of desired driver type***

> apame09@AjinkyaPame:~$ docker network create pame --driver bridge

6e792ce74a140644eeb50256ff6466ac13b8d954457701b60f16ac5e729a1693

***14.*** *docker inspect cont-name/image/network/volume****: Inspect******any of these.***

> apame09@AjinkyaPame:~$ docker inspect app\_db

[

{

"Name": "app\_db",

"Id": "192ea7819477a199740bc96b44321374331e407f4947bd2b703316daa7faa5d7",

"Created": "2025-06-12T08:16:38.772630642Z",

"Scope": "local",

"Driver": "bridge",

***14.*** *docker tag source-image target-image:tag****: Tag already present image with new desired name and version mostly to push to the repository.***

> apame09@AjinkyaPame:~$ docker tag httpd ajinkyap009/docker-1:httpd

***15.*** *docker push imagename****: Push image to DockerHub***

> apame09@AjinkyaPame:~$ docker push ajinkyap009/docker-1:httpd

The push refers to repository [docker.io/ajinkyap009/docker-1]

06844457d535: Layer already exists

bb6bbe7ebb57: Layer already exists

71a01f66bd83: Layer already exists

5f70bf18a086: Layer already exists

25727ad57f42: Layer already exists

7fb72a7d1a8e: Layer already exists

httpd: digest: sha256:9476a28ed1d67c72b101c29ffea37e0d2c139d540c04be7005c5c45d8030ba25 size: 1572

***16.*** *docker login****: Login to DockerHub***

These are some common docker commands used daily. Docker is very useful and powerful containerization tool.

♣♣♣