**Docker Notes**

**Introduction to Docker:**

* Docker is a containerization tool.
* It enables virtualization at the OS level.
* Allows developers to develop code and ship it into containers.
* Containers ensure that applications run reliably in different environments.

**Difference Between Docker and VMware:**

* **VMware** virtualizes the entire hardware of your PC/laptop.
* **VMware** operates multiple OS like Windows, Linux, Ubuntu, RedHat.
* **VMware** uses a hypervisor to create virtual machines (VMs).
* **Docker** overcomes VMware drawbacks by providing server-level virtualization.

**How Docker Works:**

1. Developers write code.
2. Code is shipped into containers.
3. After building, the application is available to the end user.
4. Containers are deployed on a web server.

**Containers:**

* Run a single process.
* Share the same OS.
* Multiple containers can run on a single hardware system.

**Docker Overview:**

|  |  |
| --- | --- |
| **Type** | **Details** |
| Containerization Tool | Docker |
| Vendor | Docker |
| Open Source | Yes |
| OS | Cross-platform |
| Official Website | Docker |

**Key Features of Docker:**

* Follows client-server architecture.
* Developers and Operationsteams can write Dockerfiles.
* Docker images are created in two ways:
  1. From a Dockerfile.
  2. Using Docker commands.
* Docker images are stored in containers.
* Images and containers are backed up in Docker Hub.
* Docker images can be pulled into multiple environments.
* Each image corresponds to a single container.

**Docker Hub**

* Used for storing both images and containers.
* Enables easy access and sharing of images.

**Steps to Set Up Docker:**

**1. Create a Docker Hub Account**

* Sign up at Docker Hub.
* Create two repositories in Docker Hub.

**2. Set Up Docker on an Instance**

1. Launch an instance (Linux-based system preferred).
2. Switch to root user:

sudo -su

1. Check network configuration:

ifconfig -a

1. Install Docker:

yum install -y docker

1. Verify Docker installation:

docker --version

1. Start Docker service:

service docker start

1. Check Docker root directory:

cd /var/lib/docker

1. List available Docker images:

docker images

1. List running containers:

docker ps

1. List all containers (including stopped ones):

docker ps -a

**3. Pulling a Docker Image**

1. Pull an image from the official online repository:

docker pull <image name>

1. Verify the downloaded image:

docker images

1. Inspect the image for details:

docker inspect <image\_id>

**Steps to Create a Container**

1. Run a container from the pulled image:

docker run -d --name <name> -p 8080:80 <imageid>

1. Check running containers:

docker ps

1. Access the running container:

docker exec -it <name> /bin/bash

1. Stop a container:

docker stop <name>

1. Remove a container:

docker rm <name>

1. Remove an image:

docker rmi <image name>

20-03-2025

1.Open the git bash and connect the aws account with the command

ssh -i "Linux.pem" ec2-user@ec2-13-49-223-96.eu-north-1.compute.amazonaws.com

2. Switch to root user by command

sudo su

3.Use command to docker home location

cd /var/lib/docker

4.To pull the image using the command

docker pull nginx

5.To see the image whether it is pulled or not

docker images

6.To push the images to docker hub account

docker tag nginx:latest umeshindra/qa:ng

docker push umeshindra/ga:ng

7.To create a container to push image

docker run -itd –name Umesh -p 40:80 53a18edff809

8.Check whether container is created or running

docker ps

9.Now check the image is getting or not

<http://13.49.223.96:40/>

10.To stop the running container

Docker stop Umesh

Image 2--tomcat

1.To pull the image using the command

docker pull tomcat:8.0.52

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag tomcat:8.0.52 umeshindra/dev:tom

docker push umeshindra/dev:tom

4.To create a container to push image

docker run -itd –name tomcat-server2 -p 8080:80 b4b762737ed4

5.Check whether container is created or running

docker ps

6.If any containers are running stop them by passing container name

docker stop <container name>

7.Now check the image is getting or not

<http://13.49.223.96:8080/>

Image 3- ubuntu

1.To pull the image using the command

docker pull ubuntu

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag ubuntu:latest umeshindra/dev:ubu

docker push umeshindra/dev:ubu

4.To create a container to push image

docker run -itd –name my-ubuntu -p 8080:80 ubuntu bash

5.To install Apache server

apt update && apt install -y apache2

6.To start the service

Service apache2 start

7.Check whether container is created or running

docker ps

8.If any containers are running stop them by passing container name

docker stop <container name>

9.Now check the image is getting or not

<http://13.49.223.96:8080/>

image 4—httpd

1.To pull the image using the command

docker pull httpd

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag httpd:latest umeshindra/dev:http

docker push umeshindra/dev:http

4.To create a container to push image

docker run -itd –name httpd -p 80:80 httpd

5.Check whether container is created or running

docker ps

6.If any containers are running stop them by passing container name

docker stop <container name>

7.Now check the image is getting or not

<http://13.49.223.96:80/>

image 5—python

1.To pull the image using the command

docker pull python

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag python:latest umeshindra/dev:pyt

docker push umeshindra/dev:pyt

4.To create a container to push image

docker run -itd –name python -p 50:50 python

5.Check whether container is created or running

docker ps

6.If any containers are running stop them by passing container name

docker stop <container name>

7.To run the python commands in docker by using command

docker exec -it python python

8.After completion of python commands want to come out the command is

Exit

image 6—mysql

1.To pull the image using the command

docker pull mysql

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag mysql:latest umeshindra/dev:sql

docker push umeshindra/dev:sql

4.To create a container to push image

docker run –name mysql -e MYSQL\_ROOT\_PASSWORD=Umesh@23 -d mysql

docker exec -it mysql mysql -u root -p

🡪 enter password

5.Check whether container is created or running

docker ps

6.If any containers are running stop them by passing container name

docker stop <container name>

7.After completion of sql commands want to come out the command is

Exit

Image 7- jenkins

1.To pull the image using the command

docker pull jenkins/jenkins

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag Jenkins/jenkins:latest umeshindra/dev:jen

docker push umeshindra/dev:jen

4.To create a container to push image

docker run -d --name jenkins-server -p 8080:8080 -p 50000:50000 jenkins/jenkins

docker logs jenkins-server

image 8-php

1.To pull the image using the command

docker pull php

2.To see the image whether it is pulled or not

docker images

3.To push the images to docker hub account

docker tag php:latest umeshindra/dev:php

docker push umeshindra/dev:php

4.To create a container to push image

docker run -d -p 8080:80 --name my-php-app -v $(pwd):/war/www/html php:apache

5.Now check the image is getting or not

<http://51.20.35.177:8080/>