What is Docker Hub Registry

Docker Hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker Cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

Ref URL : https://docs.docker.com/docker-hub/

1. Create a docker hub account in https://hub.docker.com/
2. Pull a docker image

   ```
   docker pull ubuntu
   ```

3. pull a docker image with the old version

   ```
   docker pull ubuntu:16.04
   ```

4. create a custom tag to the docker image

   ```
   docker tag ubuntu:latest valaxy/ubuntu:demo
   ```

5. login to your docker hub registry

6.
   ```
   docker login
   docker push valaxy/ubuntu:demo
   ```

testing

1. Remove all images in docker server

   ```
   docker image rm -f <Image_id>
   ```

2. Pull your custom image from your docker account

   ```
   docker pull valaxy/ubuntu:demo
   ```

Docker Commands

1. how to search a docker image in hub.docker.com

```
docker search httpd
```

2. Download a docker image from hub.docker.com

```
docker image pull <image_name>:<image_version/tag>
```

3. List out docker images from your local system

```
docker image ls
```

4. Create/run/start a docker container from image

```
docker run -d --name <container_Name> <image_name>:<image_version/tag>
```

d - run your container in back ground (detached)

5. Expose your application to host server

```
docker run -d  -p <host_port>:<container_port> --name <container_Name> <image_name>:<Image_version/tag>
```

```
docker run -d --name httpd_server -p 8080:80 httpd:2.2
```

6. List out running containers

```
docker ps
```

7. List out all docker container (running, stpooed, terminated, etc...)

```
docker ps -a
```

8. run a OS based container which interactive mode (nothing but login to container after it is up and running)

```
docker run -i -t --name centos_server centos:latest
```

i - interactive
t - Terminal

9. Stop a container

```
docker stop <container_id>
```

10. Start a container which is in stopped or exit state

```
docker start <container_id>
```

11. Remove a container

```
docker rm <container_id>
```

12. login to a docker container

```
docker exec -it <container_Name> /bin/bash
```

Installing Docker on Amazon Linux server

Pre-requisites

1. Amazon Linux EC2 Instance

Installation Steps

Install docker and start docker services

```
yum install docker -y
docker --version
# start docker services
service docker start
service docker status
```

Create a user called dockeradmin

```
useradd dockeradmin
passwd dockeradmin
```

add a user to docker group to manage docker

```
usermod -aG docker dockeradmin
```

Validation test

1. Create a tomcat docker container by pulling a docker image from the public docker registry

   ```
   docker run -d --name test-tomcat-server -p 8090:8080 tomcat:latest
   ```

Docker Installation on CentOS server

Referance URL : https://docs.docker.com/install/linux/docker-ce/centos/#install-using-the-repository

Pre-requisites

Please follow below steps to install docker CE on CentoOS server instance. For RedHat only Docker EE available

1. Install the required packages.
2. sudo yum install -y yum-utils \
3. device-mapper-persistent-data \
   lvm2
4. Use the following command to set up the stable repository.
5. sudo yum-config-manager \
6. --add-repo \
   https://download.docker.com/linux/centos/docker-ce.repo

INSTALLING DOCKER CE

1. Install the latest version of Docker CE.

   ```
   sudo yum install docker-ce
   ```

   Note: If prompted to accept the GPG key, verify that the fingerprint matches 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35, and if so, accept it.

2. Start Docker.

   ```
   sudo systemctl start docker
   ```

3. Verify that docker is installed correctly by running the hello-world image.

   ```
   sudo docker run hello-world
   ```

DockerFile

1. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.
2. The docker build command builds an image from a Dockerfile and a context.

```
docker build .
docker build -f /path/to/a/Dockerfile .
docker build -t shykes/myapp .
```

3. The Docker daemon runs the instructions in the Dockerfile one-by-one, committing the result of each instruction to a new image if necessary, before finally outputting the ID of your new image.
4. Whenever possible, Docker uses a build-cache to accelerate the docker build process significantly.
5. When you’re done with your build, you’re ready to look into scanning your image with docker scan

```
docker scan hello-world
```

Dockerfile Creation

Docker file format

```
# Comment
INSTRUCTION arguments
```

1. A Dockerfile must begin with a FROM instruction.

2. Docker distributes official versions of the images that can be used for building Dockerfiles under docker/dockerfile repository on Docker Hub.

- FROM - A valid Dockerfile must start with a FROM instruction.
- RUN - The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.
- CMD - The main purpose of a CMD is to provide defaults for an executing container.
  Note: There can only be one CMD instruction in a Dockerfile. If you list more than one CMD then only the last CMD will take effect.
  Note - Do not confuse RUN with CMD. RUN actually runs a command and commits the result; CMD does not execute anything at build time, but specifies the intended command for the image.
- LABEL -The LABEL instruction adds metadata to an image. A LABEL is a key-value pair. T

- EXPOSE
    1. The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime.
    2. To set up port redirection on the host system, see using the -P flag.
    3. The docker network command supports creating networks for communication among containers without the need to expose or publish specific ports, because the containers connected to the network can communicate with each other over any port.
- ENV - The ENV instruction sets the environment variable to the value .
- ADD - The ADD instruction copies new files, directories or remote file URLs from and adds them to the filesystem of the image at the path .

  If is a local tar archive in a recognized compression format (identity, gzip, bzip2 or xz) then it is unpacked as a directory.

- COPY - The COPY instruction copies new files or directories from and adds them to the filesystem of the container at the path .
- ENTRYPOINT - An ENTRYPOINT allows you to configure a container that will run as an executable.
    1. ENTRYPOINT will override all elements specified using CMD
    2. Both CMD and ENTRYPOINT instructions define what command gets executed when running a container. There are few rules that describe their co-operation.
    3. Dockerfile should specify at least one of CMD or ENTRYPOINT commands.
    4. ENTRYPOINT should be defined when using the container as an executable.
    5. CMD should be used as a way of defining default arguments for an ENTRYPOINT command or for executing an ad-hoc command in a container.
    6. CMD will be overridden when running the container with alternative arguments.
- VOLUME - The VOLUME instruction creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers.
- USER - The USER instruction sets the user name to use when running the image
- WORKDIR - The WORKDIR instruction sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions
- ARG - The ARG instruction defines a variable that users can pass at build-time to the builder with the docker build command using the --build-arg = flag.
- SHELL - The SHELL instruction allows the default shell used for the shell form of commands to be overridden.

Install apache Dockerfile

```
FROM
ubuntu:12.04
                    MAINTAINER mohit1talmale@gmail.com
                    LABEL version="1.1.0" \
                        app_name="registration application" \
                        release_date="9-Sep-2021"
                    RUN apt-get update && apt-get install -y apache2 && apt-get clean
                    ENV APACHE_RUN_USER www-data
                    ENV APACHE_RUN_GROUP www-data
                    ENV APACHE_LOG_DIR /var/log/apache2
                    EXPOSE 80
                    COPY index.html /var/www/html
                    CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

Tomcat Dockerfile

```
FROM
centos
            RUN yum install java -y
            RUN mkdir /opt/tomcat/
            WORKDIR /opt/tomcat
            ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.54/bin/apache-tomcat-9.0.54.tar.gz /opt/tomcat
            RUN tar xvfz apache*.tar.gz
            RUN mv apache-tomcat-9.0.54/* /opt/tomcat
            EXPOSE 8080
            CMD ["/opt/tomcat/bin/catalina.sh", "run"]
```