St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

A.Y. 2023-2024

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

Experiment – 9: To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.

1. **Aim:** To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.
2. **Objectives:** Aim of this experiment is that, the students will learn:
   - Introduction to Docker Architecture
   - To use Docker to Build, ship and manage applications using containerization
3. **Outcomes:** After study of this experiment, the students will learn following:
   - Introduction to Docker Architecture
   - Container Life Cycle
   - Understanding images and containers ● Publishing image on Docker Hub.
4. **Prerequisite:** None
5. **Requirements:** Docker Desktop, JDK, Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

Brief Theory: Refer shared material

7. **Laboratory Exercise**

A. Procedure:

a. Answer the following:

1. What are docker containers and docker images?

Ans:

Docker Containers:

Docker containers are lightweight, standalone, and executable packages that include everything needed to run a piece of software, including the code, runtime, libraries, and system tools. Containers are isolated from each other and the host system, making it easy to deploy and manage applications consistently across different environments.

Docker Images:

Docker images are read-only templates used to create containers. They are essentially snapshots of a file system with an application's code and dependencies.

Images can be shared and reused to create multiple containers with the same configuration and application setup.

In short, Docker containers are the running instances of Docker images, and Docker images are the templates used to create containers with specific application configurations. Docker simplifies application deployment and ensures consistency across different environments.

2. Explain docker architecture with diagrams.

Ans: Docker's architecture is designed around a client-server model, and it consists of three main components:

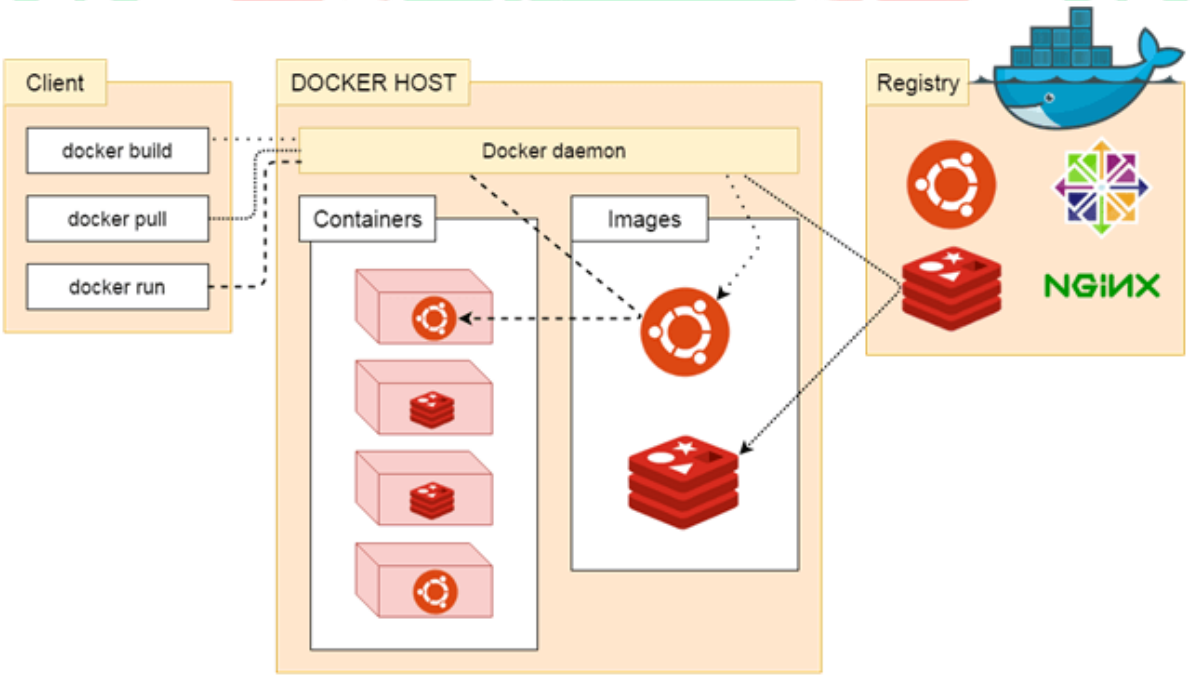
1. Docker Client:

- The Docker client is the command-line tool or API that allows users to interact with Docker. Users issue commands to the client, which communicates with the Docker server to execute those commands.

2. Docker Server (Docker Daemon):
   - The Docker server, also known as the Docker daemon, is a background service responsible for building, running, and managing Docker containers.
   - It listens for Docker API requests from the Docker client and handles container operations, such as creating, starting, and stopping containers.

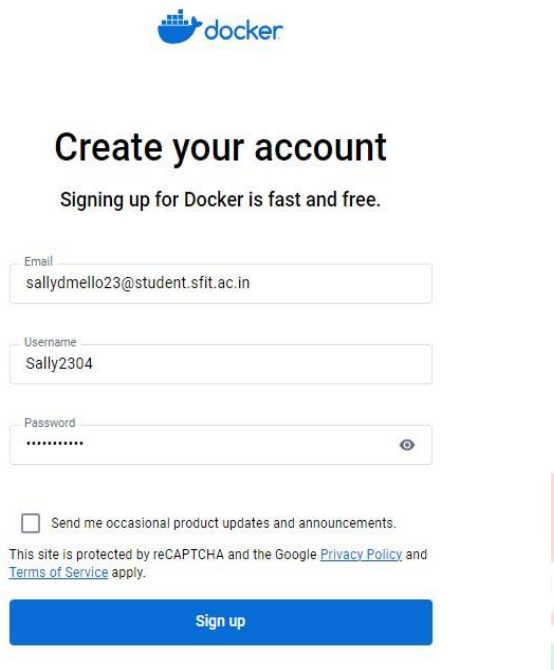
3. Docker Registry:
   - Docker images are stored in registries, which are repositories for sharing and distributing container images.
   - Docker Hub is a popular public registry, but you can also set up private registries for your organization's images.

In summary, Docker's architecture comprises a client that communicates with a server (Docker daemon) to manage and run containers, with container images stored in registries for distribution. This architecture provides a flexible and efficient way to package, deploy, and run applications in isolated containers.



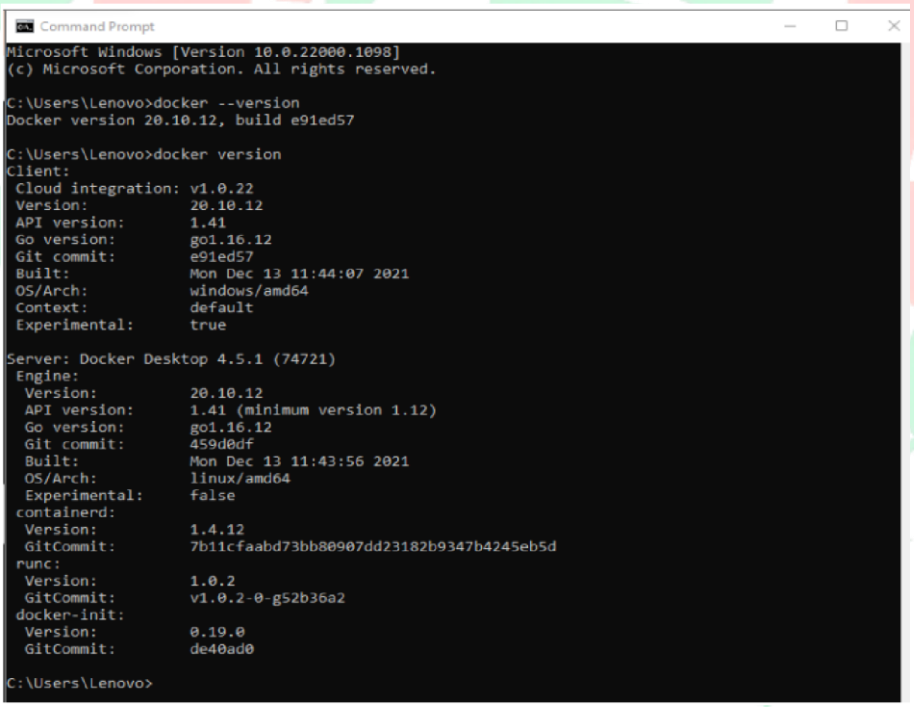
b. Execute following (Refer the shared material) and attach screenshots:

- Create Docker Hub account – screenshot of steps related to account creation

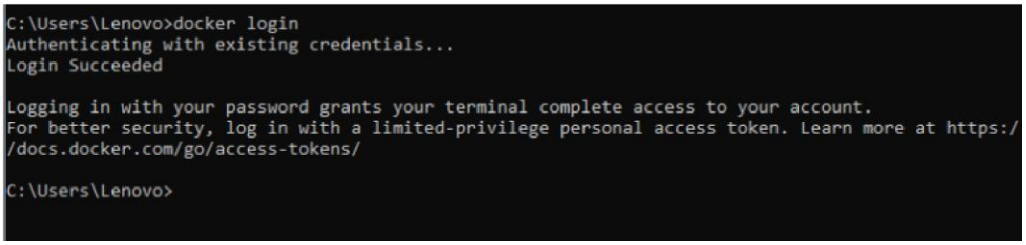


- Download and install Docker Desktop – screenshots of installation steps
- Execute following docker commands and take screenshots

1. Docker version



2. Docker login



3. Docker images

```
C:\Users\Lenovo>docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
ubuntu                  latest    c6b84b685f35   5 weeks ago     77.8MB
docker/getting-started  latest    cb90f98fd791   17 months ago   28.8MB

C:\Users\Lenovo>
```

4. Docker pull image

```
C:\Users\Lenovo>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:aabed3296a3d45cede1dc866a24476c4d7e093aa806263c27ddaadbdce3c1054
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest
```

```
C:\Users\Lenovo>docker pull ubuntu:lunar
lunar: Pulling from library/ubuntu
10fb01f4f619: Pull complete
Digest: sha256:f1090cfa89ab321a6d670e79652f61593502591f2fc7452fb0b7c6da575729c4
Status: Downloaded newer image for ubuntu:lunar
docker.io/library/ubuntu:lunar
```

5. Docker pull image-tag

```
C:\Users\Lenovo>docker images -a
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
ubuntu                  latest    c6b84b685f35   5 weeks ago     77.8MB
ubuntu                  lunar     21098a29e034   5 weeks ago     70.3MB
docker/getting-started  latest    cb90f98fd791   17 months ago   28.8MB
```

6. Docker images help

```
C:\Users\Lenovo>docker images --help

Usage:  docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:
  -a, --all             Show all images (default hides intermediate images)
      --digests         Show digests
  -f, --filter filter   Filter output based on conditions provided
      --format string   Pretty-print images using a Go template
      --no-trunc        Don't truncate output
  -q, --quiet           Only show image IDs

C:\Users\Lenovo>
```

```
C:\Users\Lenovo>docker images -q
c6b84b685f35
21098a29e034
cb90f98fd791
```

```
C:\Users\Lenovo>docker images -f
flag needs an argument: 'f' in -f
See 'docker images --help'.

C:\Users\Lenovo>docker images -f "dangling=true"
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

```
C:\Users\Lenovo>docker images -f "dangling=false"
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
ubuntu                    latest    c6b84b685f35   5 weeks ago      77.8MB
ubuntu                    lunar     21098a29e034   5 weeks ago      70.3MB
docker/getting-started    latest    cb90f98fd791   17 months ago    28.8MB
```

7. Docker run commands

```
C:\Users\Lenovo>docker run ubuntu

C:\Users\Lenovo>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Users\Lenovo>docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED             STATUS                     PORTS     NAMES
987dcd75172f   ubuntu    "/bin/bash"   About a minute ago  Exited (0) 59 seconds ago            interesting_easley
9830afaa7023   ubuntu    "bash"        26 hours ago        Exited (137) 25 hours ago            aditi
```

8. Docker ps
9. Docker start container

```
C:\Users\Lenovo>docker start interesting_easley
interesting_easley
```

10. Docker pause container
11. Docker stop container

```
C:\Users\Lenovo>docker stop sania
sania
```

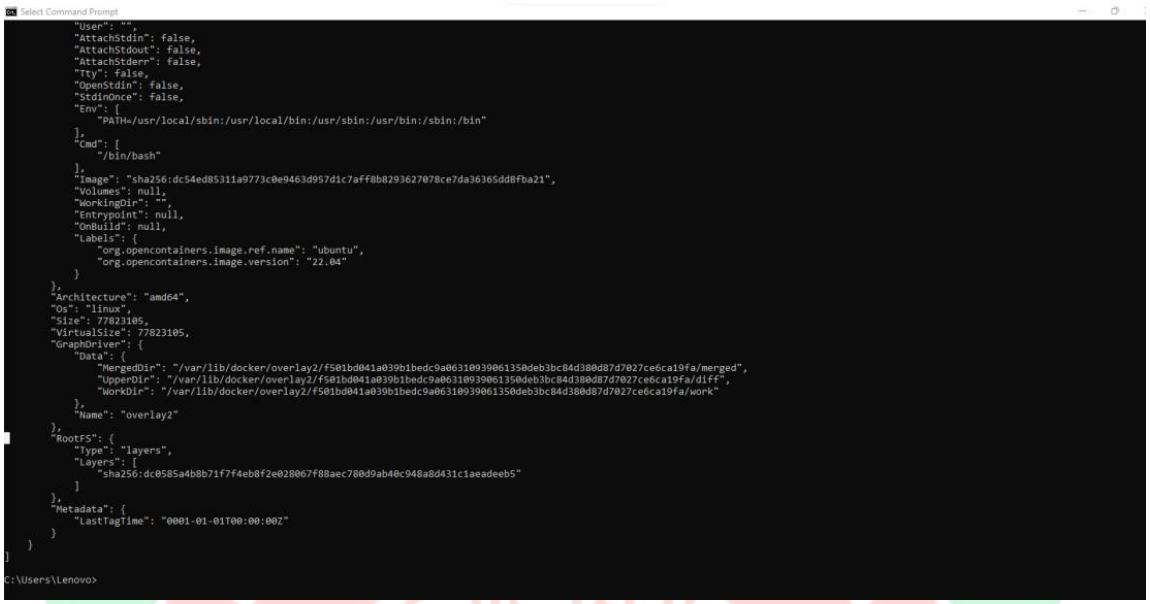
12. Docker rm container

```
C:\Users\Lenovo>docker rm interesting_easley
interesting_easley

C:\Users\Lenovo>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED          STATUS                      PORTS     NAMES
301f4ff765f2   ubuntu    "bash"    12 minutes ago   Up 12 minutes                         jerin
9830afaa7023   ubuntu    "bash"    26 hours ago     Exited (137) 26 hours ago             aditi
```

13. Docker inspect

```
C:\Users\Lenovo>docker inspect ubuntu
[
    {
        "Id": "sha256:c6b84b685f35f1a5d63661f5d4aa662ad9b7ee4f4b8c394c022f25023c907b65",
        "RepoTags": [
            "ubuntu:latest"
        ],
        "RepoDigests": [
            "ubuntu@sha256:aabed3296a3d45cede1dc866a24476c4d7e093aa806263c27ddaadbdce3c1054"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2023-08-16T06:01:54.322918183Z",
        "Container": "ccf77fc533a6d0610b8f9f15b47d1b42172a8feedf293e1c5bc8c1a37bbf2f37",
        "ContainerConfig": {
            "Hostname": "ccf77fc533a6",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh",
                "-c",
                "#(nop) ",
                "CMD [\"/bin/bash\"]"
            ],
            "Image": "sha256:dc54ed85311a9773c0e9463d957d1c7aff8b8293627078ce7da36365dd8fba21",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": {
                "org.opencontainers.image.ref.name": "ubuntu",
                "org.opencontainers.image.version": "22.04"
            }
        },
        "DockerVersion": "20.10.21",
        "Author": "",
        "Config": {
            "Hostname": "",
            "Domainname": "",
```



14. Docker rmi

```
C:\Users\Lenovo>docker rmi 21098a29e034
Untagged: ubuntu:lunar
Untagged: ubuntu@sha256:f1090cfa89ab321a6d670e79652f61593502591f2fc7452fb0b7c6da575729c4
Deleted: sha256:21098a29e034a8f6f1952d1d8a49b5732b70e532c31f0e88e1ff499c6540c57c
Deleted: sha256:3dbc2b38e067be4bebab4fd0141d3891d539822277f707f7467957abe097fc96
```

15. Docker commit

```
C:\Users\Lenovo>docker commit 301f4ff765f2 jerin99/new:latest
sha256:0d09d9e97163ff54d53d875fb4faeb144bdeb76717d8580799750737faad656c
```

16. Docker push

```
C:\Users\Lenovo>docker push jerin99/new:latest
The push refers to repository [docker.io/jerin99/new]
6e148bf6b915: Pushed
dc0585a4b8b7: Mounted from library/ubuntu
latest: digest: sha256:d1052379632c302f9b136457e7ec1dfbde6c442f6e1414628f457eab6bd6eaeb size: 736
```

17. Docker history image

```
C:\Users\Lenovo>docker history ubuntu
IMAGE          CREATED       CREATED BY                                      SIZE      COMMENT
c6b84b685f35   5 weeks ago   /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      5 weeks ago   /bin/sh -c #(nop) ADD file:aa9b51e9f0067860c…   77.8MB
<missing>      5 weeks ago   /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      5 weeks ago   /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      5 weeks ago   /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      5 weeks ago   /bin/sh -c #(nop)  ARG RELEASE                  0B
```

8. Post-Experiments

Exercise A. Extended

Theory: Nil **B. Questions:**

1. Write all Docker commands with syntax and example
2. Explain differences between VMs and docker containers **C. Conclusion:**
1. Write what was performed in the experiment.
2. Write the significance of the topic studied in the experiment. **D. References:**

https://www.youtube.com/watch?v=zJ6WbK9zFpI
https://www.simplilearn.com/tutorials/docker-tutorial
https://www.edureka.co/blog/docker-explained/