Hands-On Lab (HOL)

Lab Name

HOL-01: Docker Installation and Basic Docker Commands on Ubuntu 22.04

Author : Sandeep Kumar Sharma

Learning Objectives

By the end of this lab, participants will be able to:

Understand what Docker is and why it is used in modern DevOps environments. Install Docker Engine on Ubuntu 22.04 using the APT package manager. Verify Docker installation and Docker service status. Work confidently with basic Docker commands such as searching images, pulling images, creating containers, starting, stopping, and attaching to containers.

Lab Outcome

After completing this lab, you will be able to independently install Docker on an Ubuntu 22.04 system and perform basic container lifecycle operations. You will clearly understand the difference between images and containers and how Docker commands work internally during container creation and execution.

Pre-requisites

Ubuntu 22.04 Server or Desktop OS. Internet connectivity. A user with sudo privileges.

Lab Explanation and Hands-On Steps

Section 1: Docker Installation on Ubuntu 22.04

Before installing any new software on a Linux system, it is always recommended to update the package index so that the system is aware of the latest versions available in the repository.

```
apt update -y
```

This command refreshes the local APT cache by downloading the latest package lists from configured repositories. The `-y` flag automatically answers yes to any prompt.

Now, check whether Docker is already installed on the system.

```
docker --version
```

If Docker is not installed, you will receive a `command not found` message. This confirms that Docker needs to be installed.

Install Docker using the official Ubuntu package repository.

```
apt install docker.io -y
```

This command installs the Docker Engine (`docker.io` package) along with its required dependencies such as containerd and runc.

Verify the Docker installation again.

```
docker --version
```

If Docker is installed successfully, this command will display the Docker version, confirming a successful installation.

Check the status of the Docker service.

```
systemctl status docker
```

This command shows whether the Docker daemon is running, stopped, or failed. You should see `active (running)` in green color. Press `CTRL + C` to exit the status screen.

Section 2: Docker Basic Commands

Switch to the root user for ease of execution during the lab.

```
sudo -i
```

This command provides root-level access, avoiding repeated use of `sudo`.

Searching Docker Images

```
docker search ubuntu
```

This command searches Docker Hub for images related to Ubuntu. Docker Hub is the default public image registry. The output includes image name, description, stars, and official image status.

Listing Running Containers

```
docker ps
```

This command lists only the containers that are currently running.

Listing Docker Images

```
docker images
```

This command displays all Docker images present locally on your system.

Pulling Docker Images

```
docker pull ubuntu
```

This command downloads the latest Ubuntu image from Docker Hub. If the image already exists locally, Docker will reuse it instead of downloading again.

Check the images again:

```
docker images
```

Pull a specific Ubuntu version:

```
docker pull ubuntu:16.04
```

This pulls a tagged version of Ubuntu. Tags represent different versions or variants of an image.

Verify:

```
docker images
```

Creating and Running Containers

```
docker run -it --name c1 ubuntu
```

This command does multiple things internally:

It pulls the image if not available locally. It creates a new container named `c1`. It allocates an interactive terminal using `-it`. It starts the container and attaches you to it.

You will now be inside the Ubuntu container shell.

List running containers from another terminal or after exiting:

```
docker ps
```

List all containers (running and stopped):

```
docker ps -a
```

Exit from the container to return to the host machine:

```
exit
```

Starting Existing Containers

```
docker start c1
```

This starts an already created container without creating a new one.

Verify:

```
docker ps -a
```

Creating Containers with Specific Image Versions

```
docker run -it --name c2 ubuntu:16.02
```

This command will fail because the tag `16.02` does not exist.

Correct command:

```
docker run -it --name c2 ubuntu:16.04
```

This successfully creates and runs the container using Ubuntu 16.04 image.

Attaching to Running Containers

```
docker attach c1
```

This command attaches your terminal to the running container `c1`.

Detach or exit and attach to another container:

```
docker attach c2
```

Stopping Containers

```
docker stop c1
```

```
docker stop c2
```

These commands gracefully stop the running containers.

Verify:

```
docker ps -a
```

Starting Multiple Containers

```
docker start c1 c2
```

Docker allows starting multiple containers using a single command.

Check status:

```
docker ps -a
```

Summary

In this lab, you installed Docker on Ubuntu 22.04 and practiced essential Docker commands. You learned how Docker images are pulled, how containers are created and managed, and how to control container lifecycle operations. This lab builds a strong foundation for advanced Docker concepts such as Dockerfiles, volumes, networks, and container orchestration.