

## Experiment -1.3

**Student Name:** Chayan Gope

**Branch:** AIT CSE (DevOps)

**Semester:** 5<sup>th</sup>

**Subject Name:** Docker & Kubernetes

**UID:** 22BDO10036

**Section/Group:** 22BCD-1 (A)

**Date of Performance:** 12/08/2024

**Subject Code:** 22CSH-343

### 1. Aim/Overview of the practical: To understand Container Creation and Execution:

- Install docker on Linux/windows
- Pull a Docker image from Docker Hub.
- Run a container from the pulled image.
- Explore container logs and check the running processes inside the container.
- Stop and remove the container.

### 2. Apparatus: PC, Web Browser, Docker Engine, DockerHub, Ubuntu Linux

### 3. Steps for experiment/practical:

- **Install Docker on Windows**
  1. Download Docker Desktop: Go to the Docker website and download Docker Desktop for Windows.
  2. Install Docker Desktop: Follow the installation instructions provided by Docker. You may need to enable WSL 2 (Windows Subsystem for Linux) during the installation.
  3. Start Docker Desktop: After installation, launch Docker Desktop and wait for it to start. You should see the Docker icon in your system tray.

- **Pull a Docker image from Docker Hub**

1. We can pull a docker image from the dockerHub repository using the command below.

*docker pull <image\_name>:<tag>*

```
chayan@chayan-virtual-machine:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
chayan@chayan-virtual-machine:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	edbf74c41f8	3 weeks ago	78.1MB
hello-world	latest	d2c94e258dcb	16 months ago	13.3kB
centos	latest	5d0da3dc9764	2 years ago	231MB

- **Run a container from the pulled image.**

1. We can create and run a container of the above pulled docker image using the following commands.

*docker run -dit --name<container\_name> <image\_name>*

OR

*docker run -d --name <container\_name> <image\_name>*

```
chayan@chayan-virtual-machine:~$ docker run -dit --name spy ubuntu
d4c1b90a13dc6d64fe92f4e08bf748eed01e878475210b68e554729baa295c6b
chayan@chayan-virtual-machine:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d4c1b90a13dc	ubuntu	"/bin/bash"	7 seconds ago	Up 5 seconds		spy

- Explore container logs and check the running processes inside the container.

1. Checking the container logs

*docker logs <container\_name>*

2. Entering into a container and checking the running processes inside it.

*docker exec -it <container-name> bash*  
*top*

```
chayan@chayan-virtual-machine:~$ docker exec -it spy bash
root@d4c1b90a13dc:/# top

top - 14:09:45 up 15 min,  0 user,  load average: 0.01, 0.13, 0.15
Tasks:  3 total,   1 running,  2 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.2 sy,   0.0 ni, 99.6 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
MiB Mem :  7894.6 total,  5279.7 free,  1321.6 used,  1577.6 buff/cache
MiB Swap:  2048.0 total,  2048.0 free,    0.0 used.  6573.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
26	root	20	0	8852	5120	3072	R	0.3	0.1	0:00.01	top
1	root	20	0	4588	3584	3072	S	0.0	0.0	0:00.17	bash
18	root	20	0	4588	3712	3200	S	0.0	0.0	0:00.07	bash

- Stop and remove the container.

*docker stop <container\_name> OR docker stop <container\_id>*

*docker rm <container\_name> OR docker rm <container\_id>*

```
chayan@chayan-virtual-machine:~$ docker stop spy
spy
```

#### 4. Result/Output/Writing Summary:

1. Docker images: lightweight, standalone, and executable package that includes everything needed to run a piece of software. This includes the code, runtime, libraries, environment variables, and configuration files.
2. Docker containers: runtime instances of Docker daemon which are used to run a docker image in an isolated environment.
3. Docker repository: storage location where docker images are stored and shared. It allows users to upload, manage, and distribute docker images.
4. Docker commands: Docker commands are essential for interacting with Docker, allowing users to build, manage, and run containers, images, networks, and more.

#### Learning outcomes (What I have learned):

1. learned the concept of containerization.
2. learned to configure Docker to work with different environments.
3. learned how to build docker images using Dockerfile.
4. learned the purpose of Docker volumes and their role in data persistence.
5. learned how to use Docker Hub to pull and push Docker images.

#### Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			