Heramb Vengurlekar
T23 – 120

EXPERIMENT NO. 9

TO UNDERSTAND DOCKER ARCHITECTURE AND CONTAINER LIFE CYCLE, INSTALL DOCKER AND EXECUTE DOCKER COMMANDS TO MANAGE IMAGES AND INTERACT WITH CONTAINERS

Theory:

Introduction to Docker

Docker is an open-source platform that automates the deployment, scaling, and management of applications using containerization. Containers package applications with all dependencies, ensuring consistency across different computing environments.

Key Features of Docker:

- Lightweight and efficient
- Portability across different platforms
- Isolation of applications
- Fast deployment and scaling

Docker Architecture

Docker follows a client-server architecture consisting of the following key components:

1. Docker Client

- The command-line interface (CLI) that communicates with the Docker daemon.
- Commands like docker run, docker pull, and docker stop are executed from the client.

2. Docker Daemon (dockerd)

- The background service that manages containers and images.
- It listens to requests from the Docker client and manages container execution.

3. Docker Images

- Read-only templates used to create containers.
- They include application code, dependencies, libraries, and runtime environment.

4. Docker Containers

- Running instances of Docker images.
- Each container is isolated from the host system and other containers.

5. Docker Registry

- A repository to store and distribute Docker images.
- Examples: Docker Hub, AWS Elastic Container Registry (ECR).

Container Life Cycle

Heramb Vengurlekar
T23 – 120

The Docker container life cycle includes the following stages:

1. **Create** - A container is created from an image but not yet running.
2. **Start** - The container begins execution.
3. **Pause/Unpause** - Temporarily suspends and resumes container processes.
4. **Stop** - Gracefully stops a running container.
5. **Restart** - Stops and then starts the container again.
6. **Kill** - Forcefully stops the container.
7. **Remove** - Deletes the container permanently.

Installing Docker

Follow these steps to install Docker on a Linux system:

1. Update the package repository:
2. sudo apt update
3. Install dependencies:
4. sudo apt install apt-transport-https ca-certificates curl software-properties-common
5. Add the official Docker repository:
6. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
7. echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu focal stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
8. Install Docker:
9. sudo apt update
10. sudo apt install docker-ce docker-ce-cli containerd.io
11. Verify installation:
12. docker --version

Essential Docker Commands

1. Managing Docker Images

- **Pull an image from Docker Hub**
- docker pull <image_name>
- **List available images**
- docker images
- **Remove an image**
- docker rmi <image_id>

2. Working with Containers

- **Run a container**

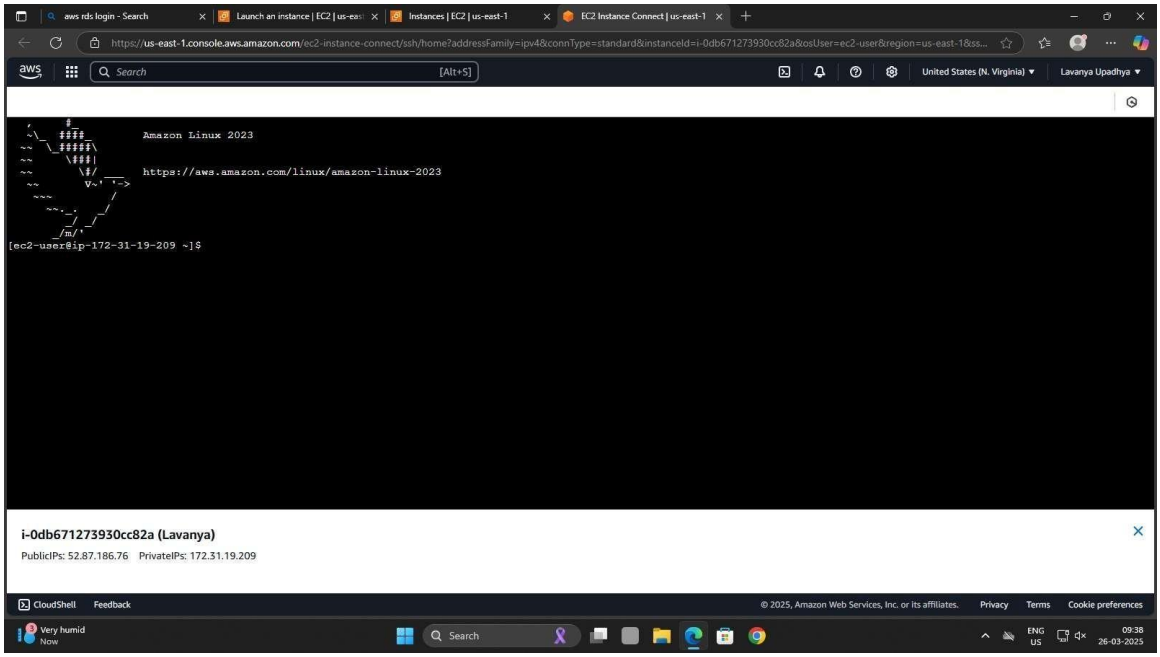
Heramb Vengurlekar
T23 – 120

- docker run -d -p 80:80 nginx
- **List running containers**
- docker ps
- **List all containers (including stopped ones)**
- docker ps -a
- **Stop a running container**
- docker stop <container_id>
- **Remove a container**
- docker rm <container_id>

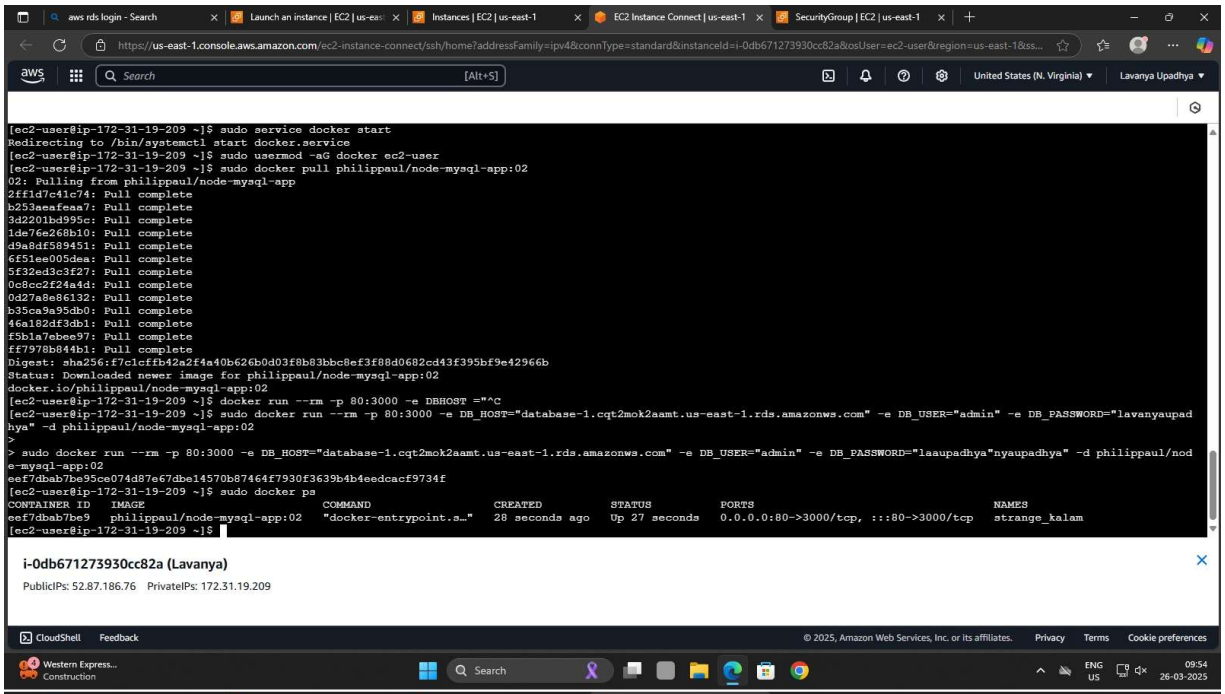
3. Interacting with Containers

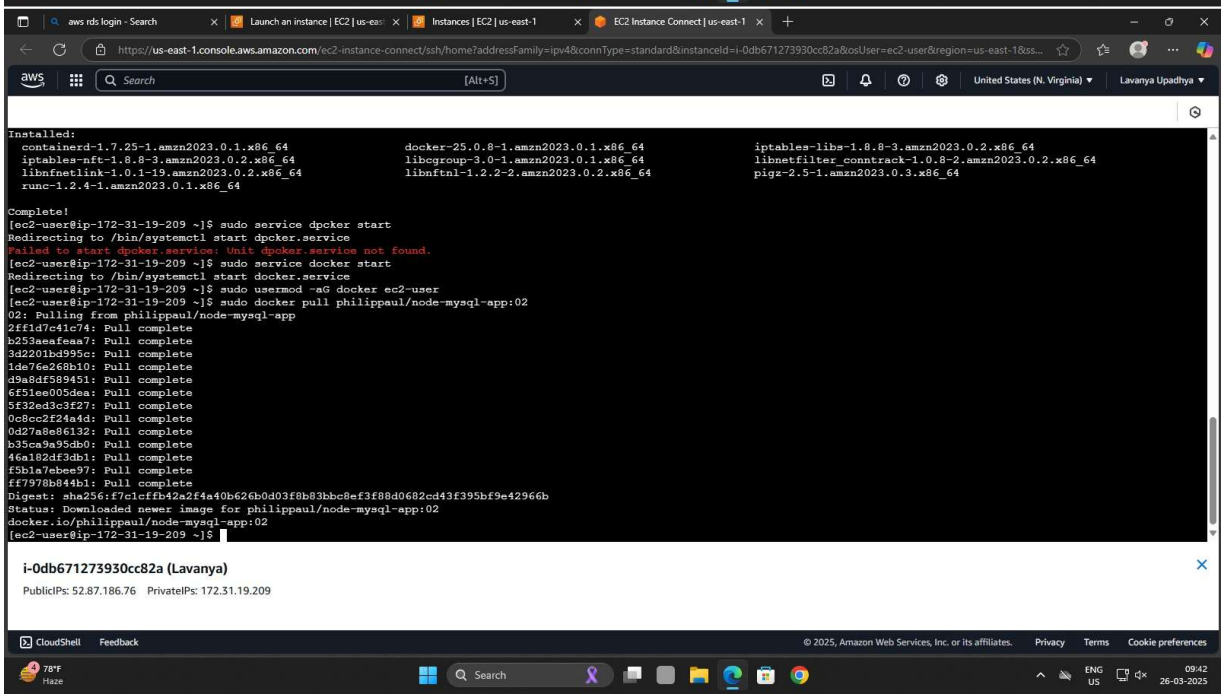
- **Access a running container's shell**
- docker exec -it <container_id> /bin/bash
- **View logs of a container**
- docker logs <container_id>

Output:

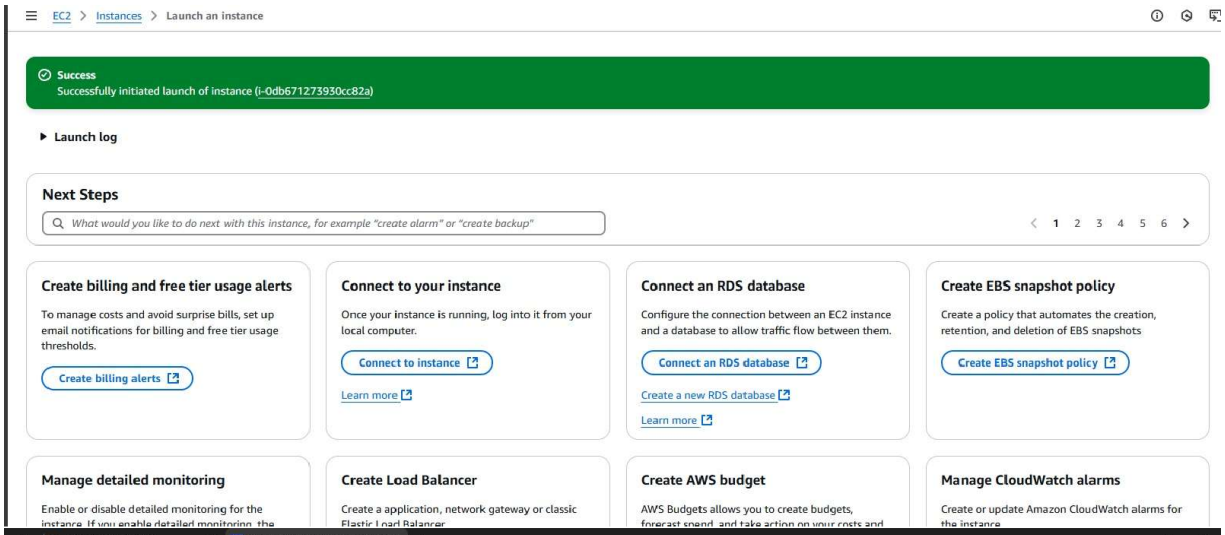


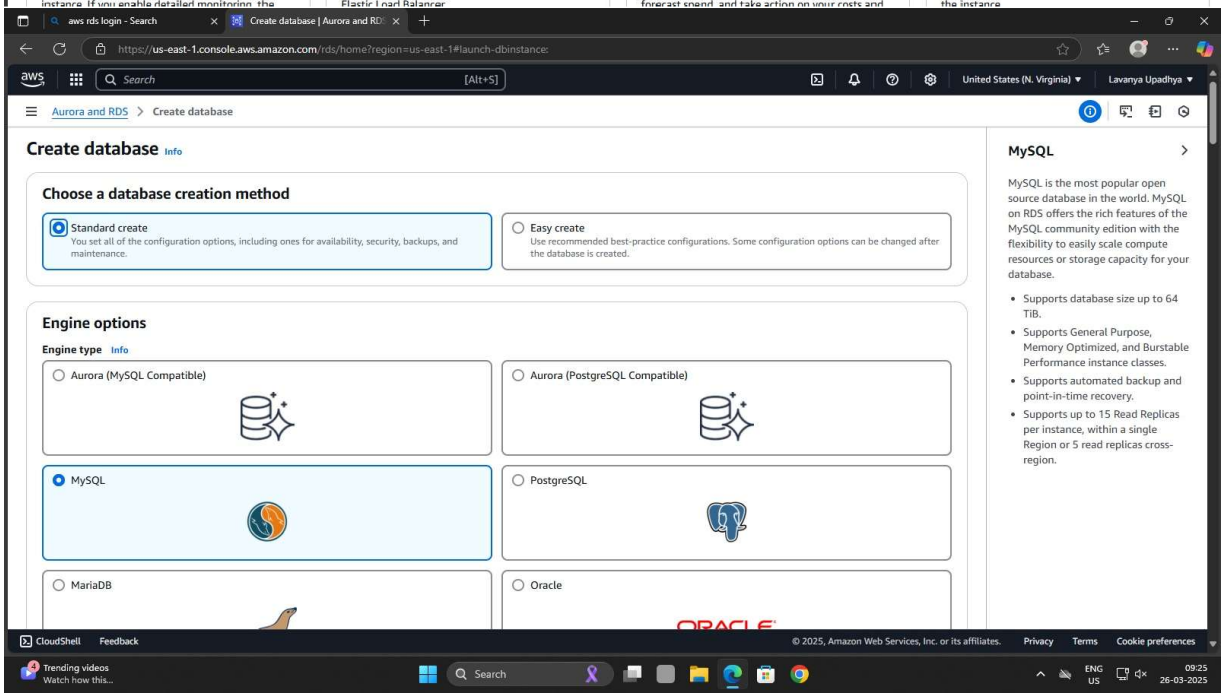
Heramb Vengurlekar
T23 – 120



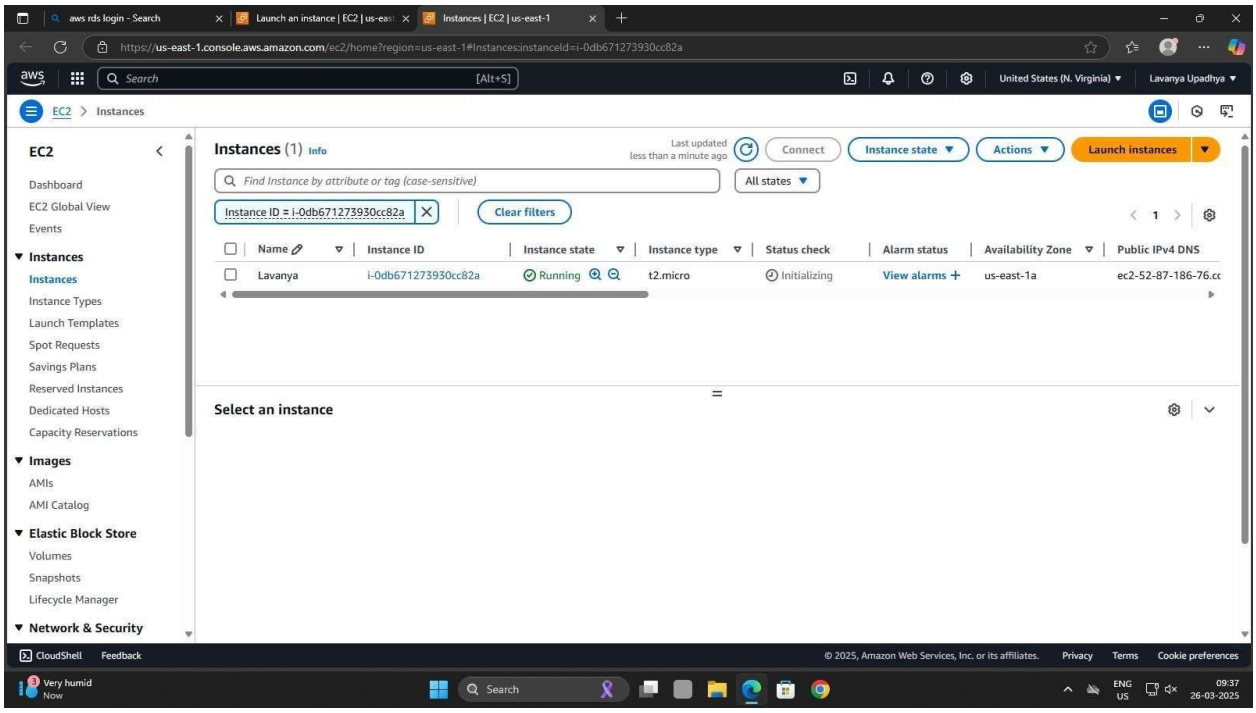


Heramb Vengurlekar
T23 – 120





Heramb Vengurlekar
T23 – 120



Conclusion

In this lab, we explored Docker's architecture, the container life cycle, and performed practical operations such as installing Docker, pulling images, and managing containers. These fundamental concepts are essential for deploying and managing applications in a containerized environment.