# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

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

## Theory:

Docker is an open-source platform that enables developers to automate the deployment of applications inside lightweight, portable, and self-sufficient containers. Containers package the application along with its dependencies and environment, ensuring consistent behavior across different systems.

### Key Features of Docker:

* Lightweight: Shares the host OS kernel, reducing overhead compared to VMs.
* Portability: Runs the same container on any environment (dev, test, prod).
* Isolation: Applications run in isolated environments with their own file systems and dependencies.
* Fast startup: Containers start in seconds, enabling rapid deployment.
* Version control: Docker images can be versioned and managed easily.
* Microservices-friendly: Perfect for deploying individual services in a microservices architecture.

### Docker as a Containerization Tool

Docker uses containerization to simplify software development, testing, and deployment. It provides tools to build, ship, and run applications inside containers.

### Key Docker Components:

Docker Engine Core service for building and running containers Docker Image Read-only template used to create containers Docker Container A runnable instance of a Docker image Dockerfile A script with instructions to build a Docker image Docker Hub Cloud-based registry to share and download images

Docker Compose Tool for defining and running multi-container apps Docker Volume Persistent storage for containers

Docker Network Manage communication between containers

### Demonstration of Docker (Theoretical Steps)

1. Install Docker
   1. Download Docker Desktop from https://[www.docker.com.](http://www.docker.com/)
   2. Install and start Docker Engine on your system (Linux, Windows, or macOS).
   3. Verify installation by running:

*docker --version*

1. Pull a Docker Image
   1. Use Docker Hub to pull a pre-built image:

*docker pull nginx*

TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

1. Run a Docker Container
   1. Start a container using the pulled image:

*docker run -d -p 8080:80 nginx*

* 1. Access the web server by navigating to http://localhost:8080 in your browser.

1. Build a Custom Docker Image
   1. Create a file named Dockerfile:

*FROM python:3.10*

*COPY app.py /app/app.py WORKDIR /app*

*CMD ["python", "app.py"]*

* 1. Build the image using:

*docker build -t my-python-app .*

* 1. Run the container:

*docker run -d -p 5000:5000 my-python-app*

1. Manage Containers
   1. View running containers:

*docker ps*

* 1. Stop a container:

*docker stop <container\_id>*

* 1. Remove a container:

*docker rm <container\_id>*

1. Use Docker Compose (Optional)
   1. Create a docker-compose.yml file to define multiple services.
   2. Start all services using:

*docker-compose up*

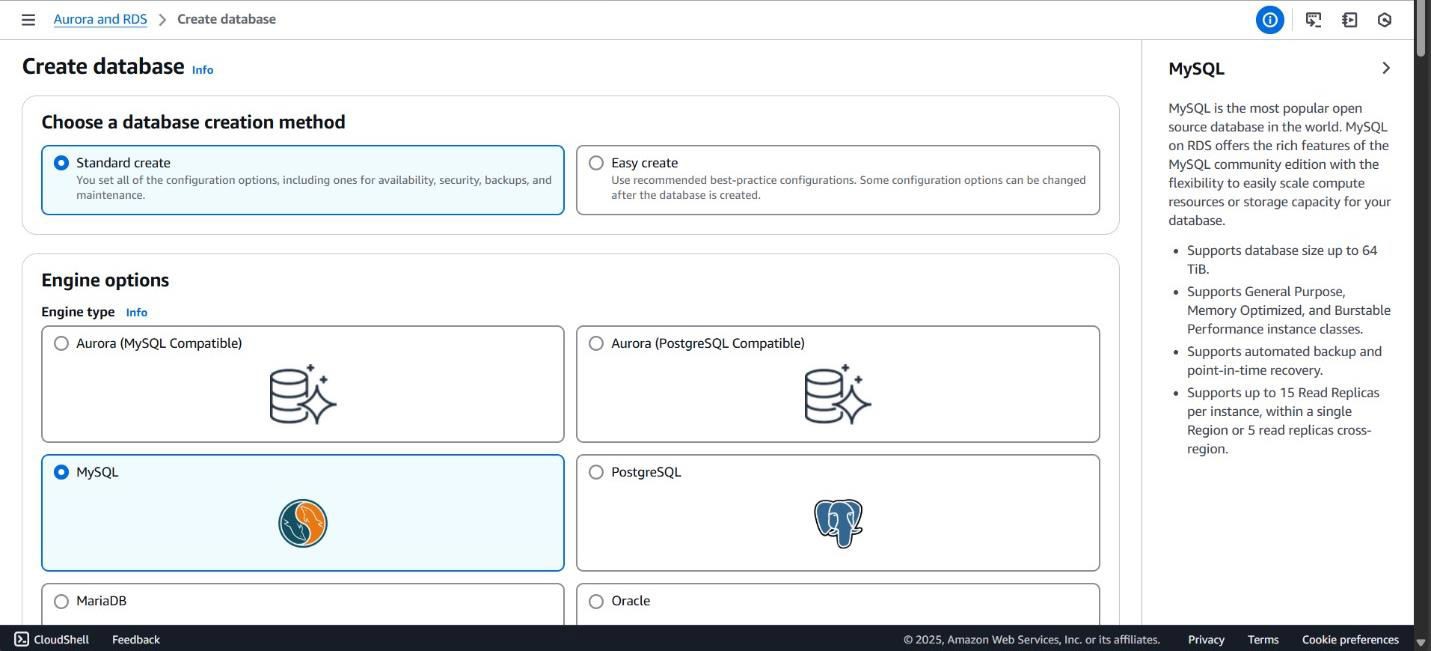
### Use Case Example:

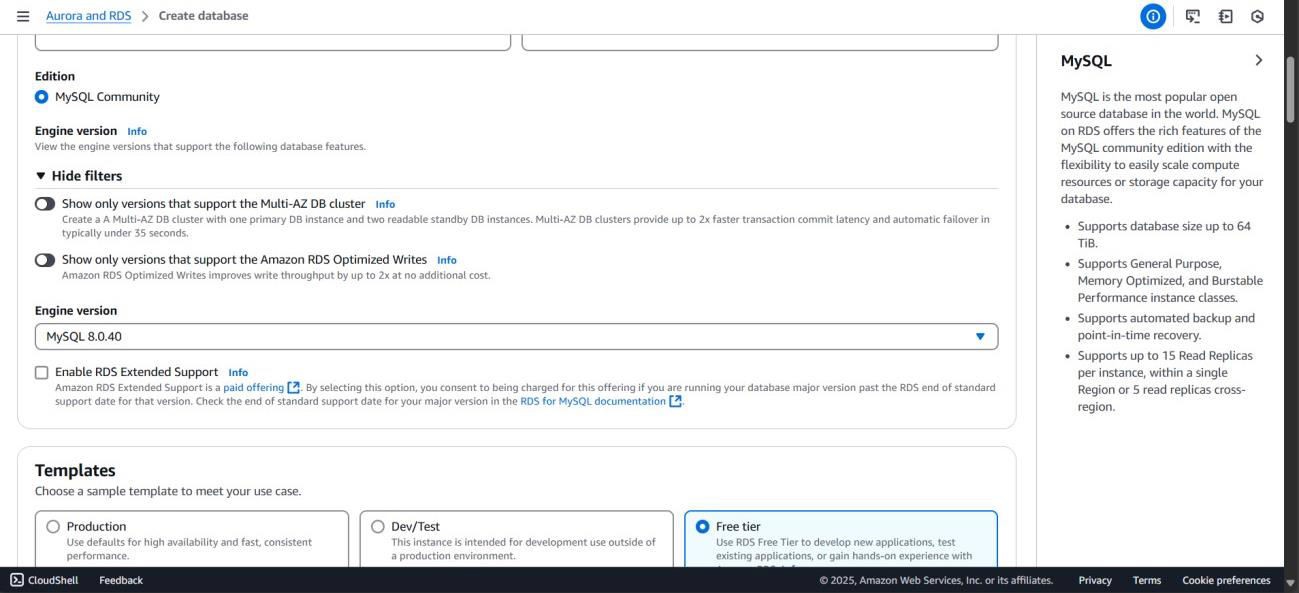
* Containerizing web applications for consistent deployment
* Running microservices architecture in isolated containers
* Developing and testing across different environments
* Deploying scalable applications on cloud or on-premises
* Creating reproducible development environments
* CI/CD pipeline integration for automated testing and delivery
* Education and experimentation with different tech stacks

**Implementation:**

TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

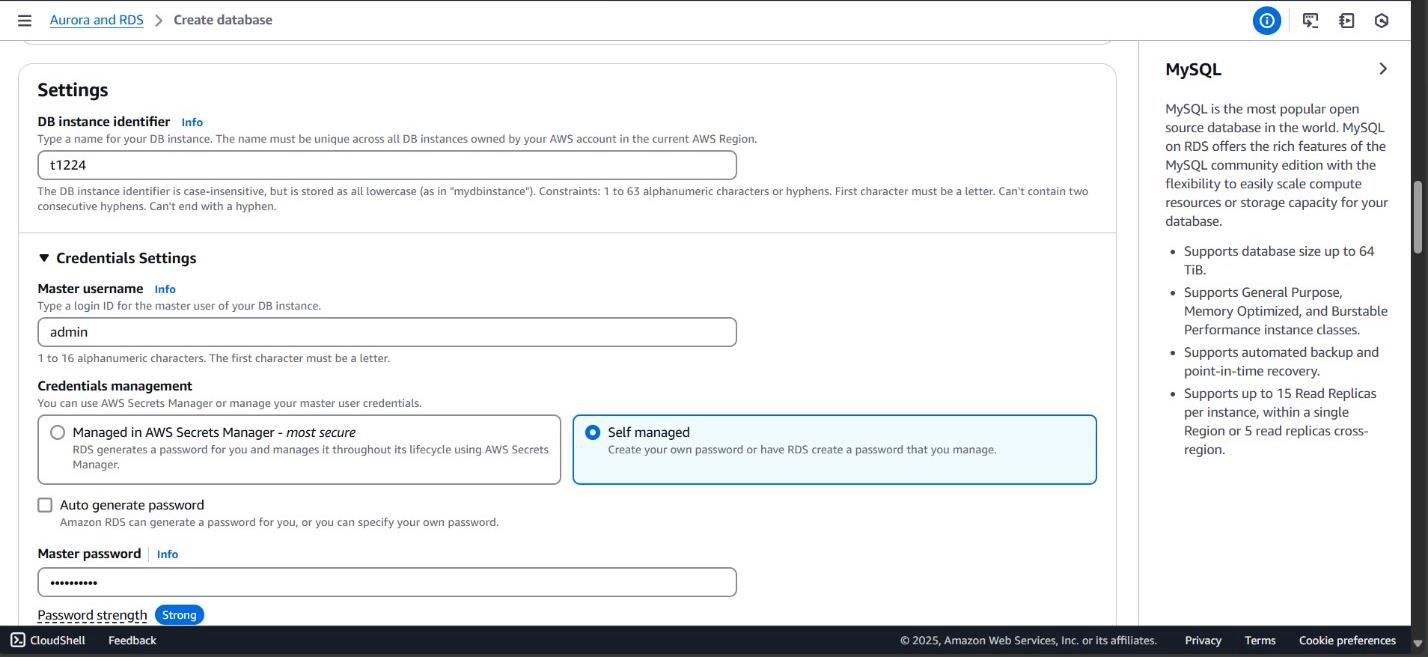
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

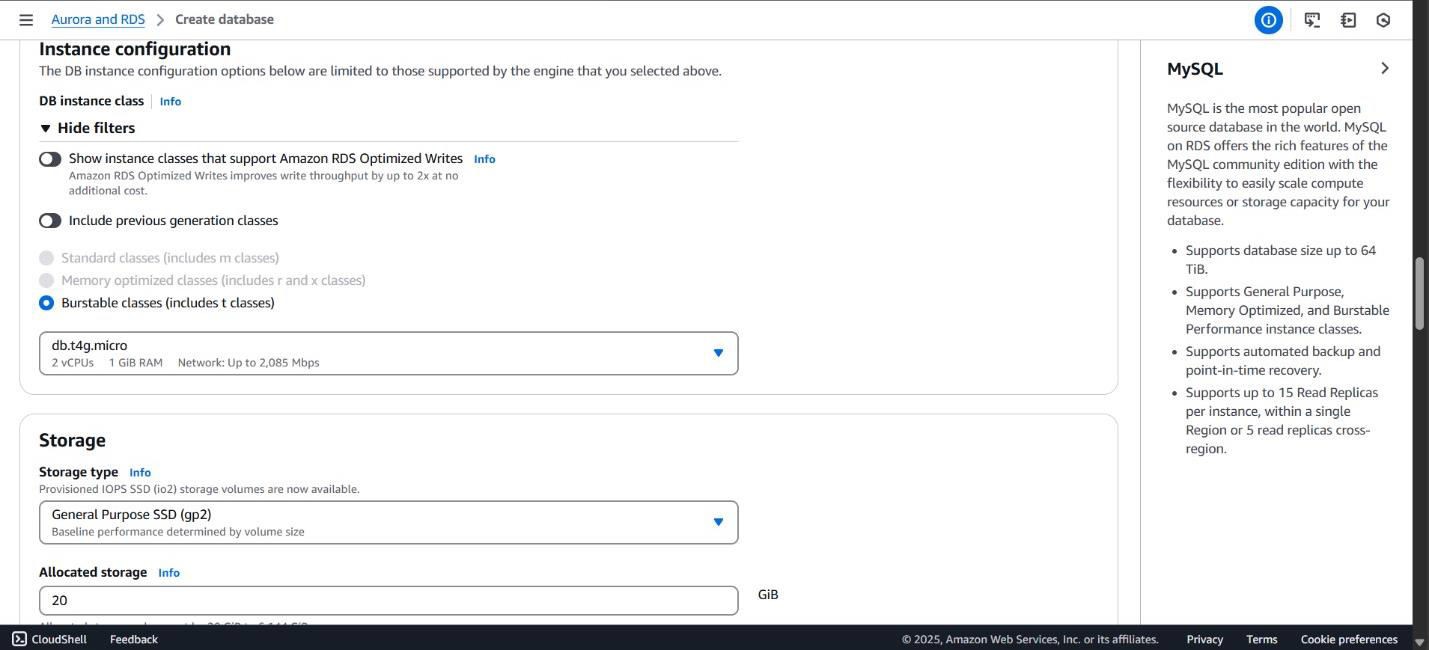




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

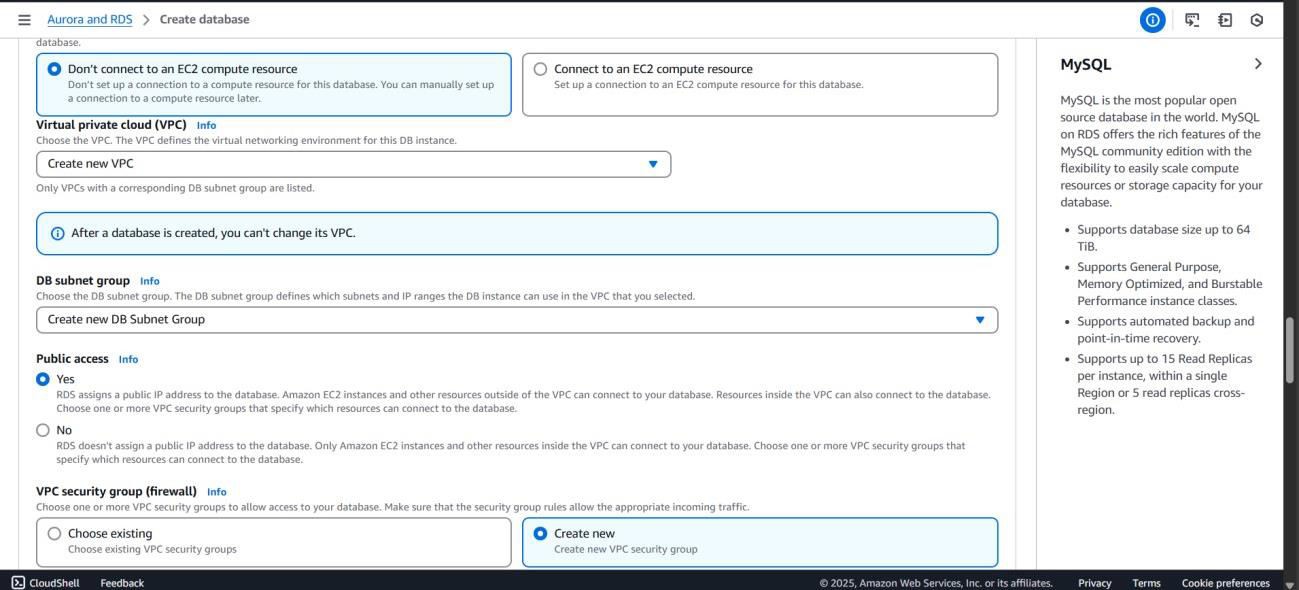
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

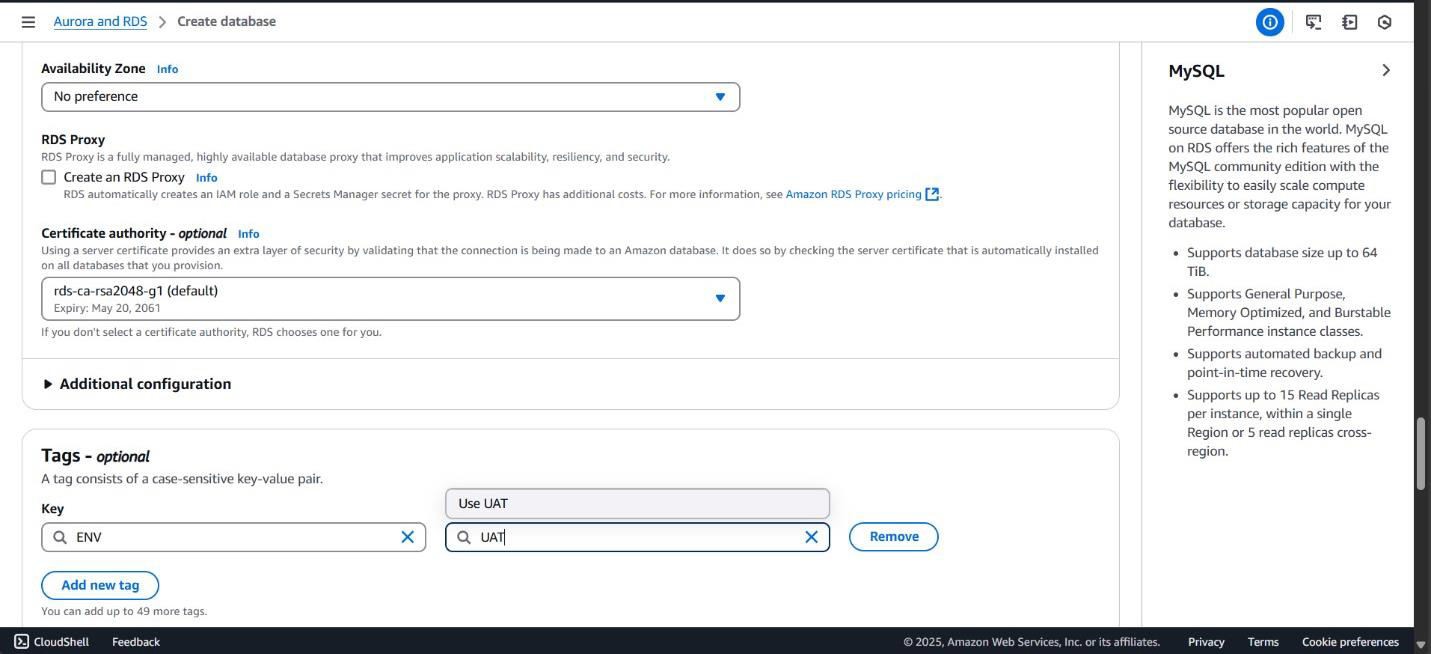




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

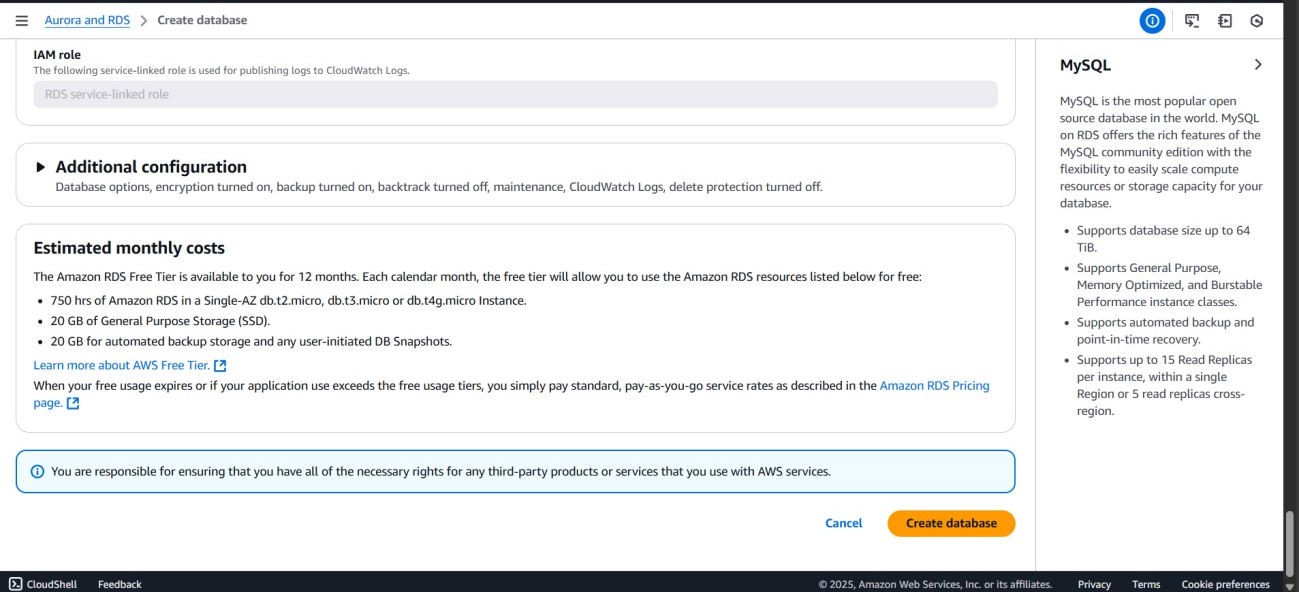
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

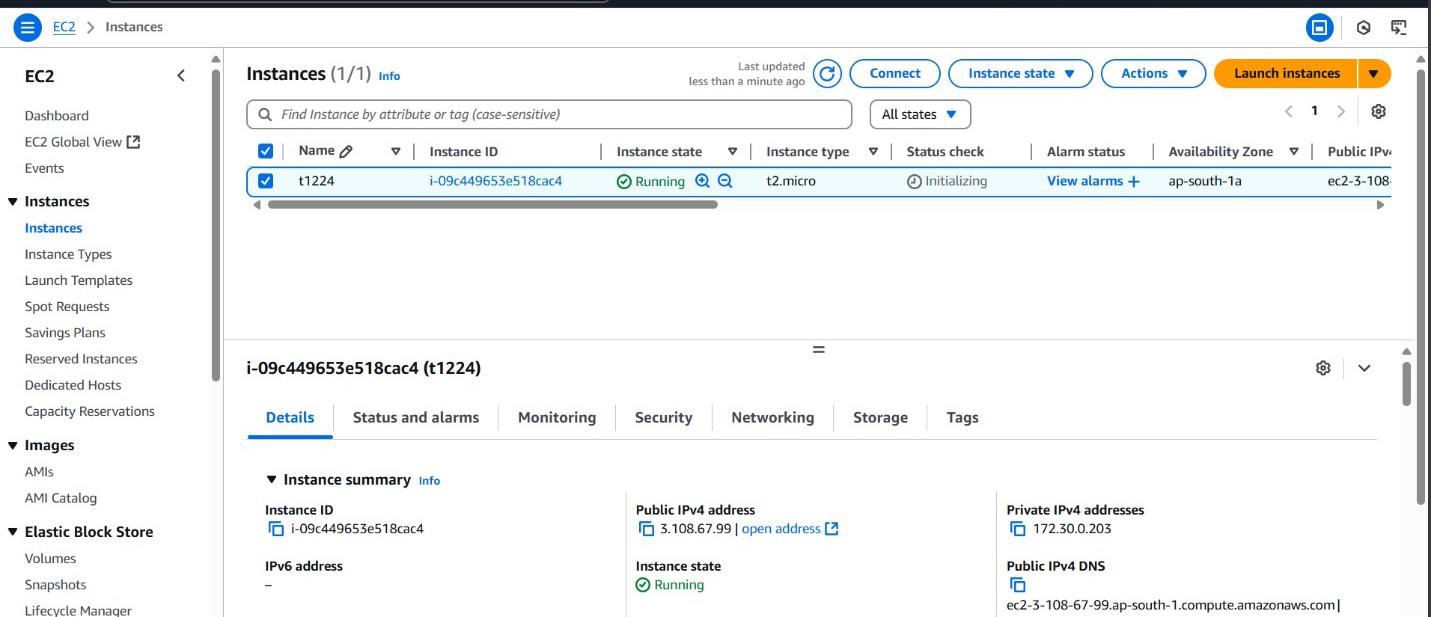




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

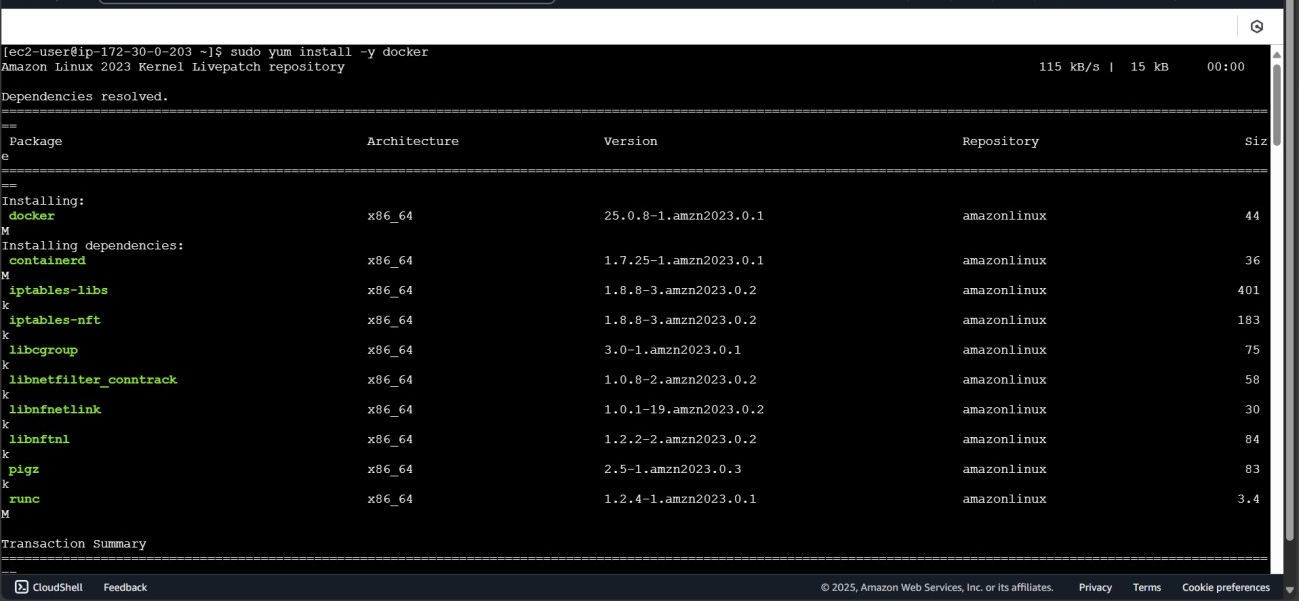
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

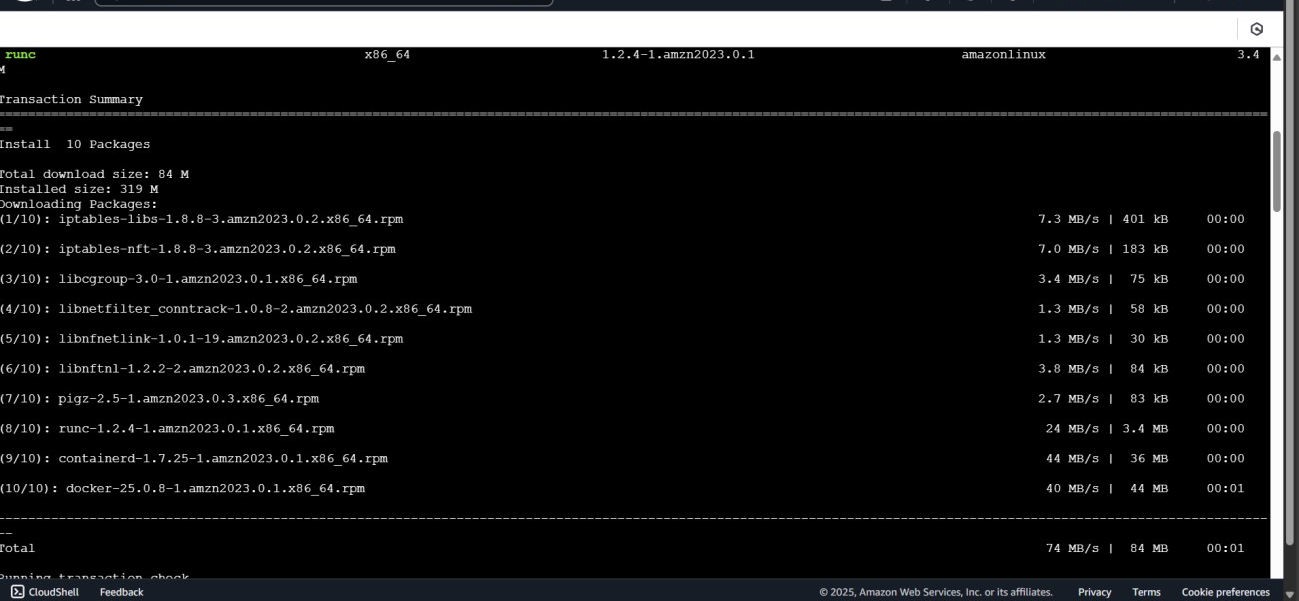




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

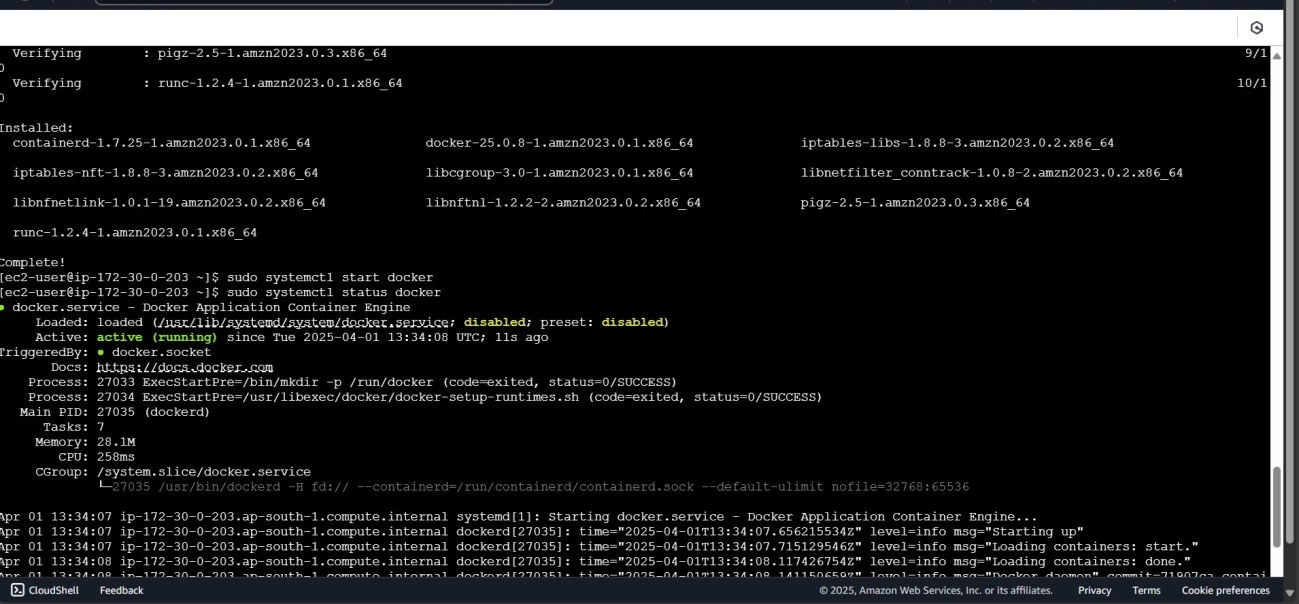
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

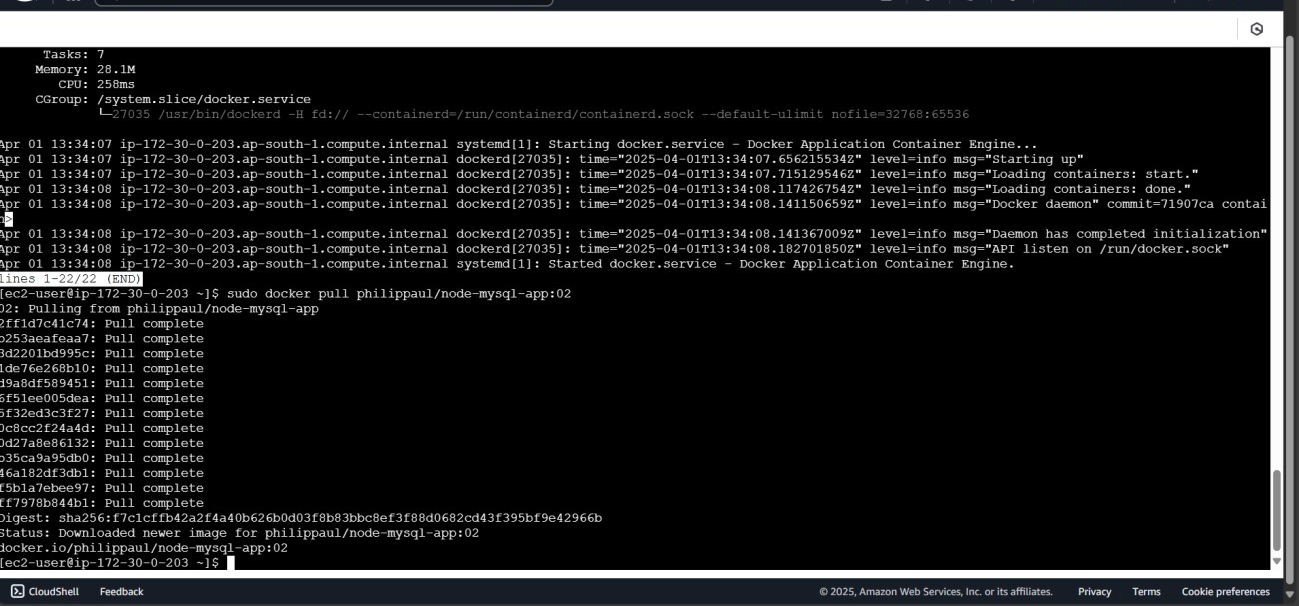




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

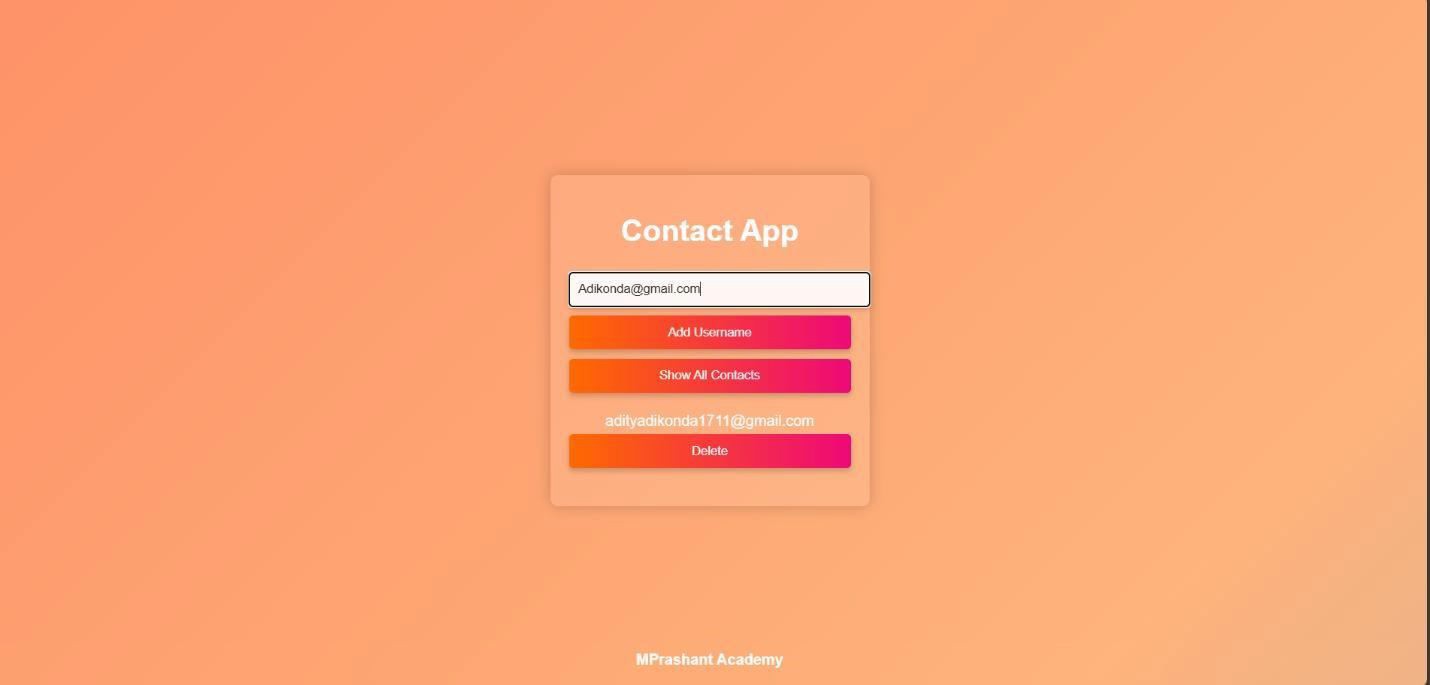
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

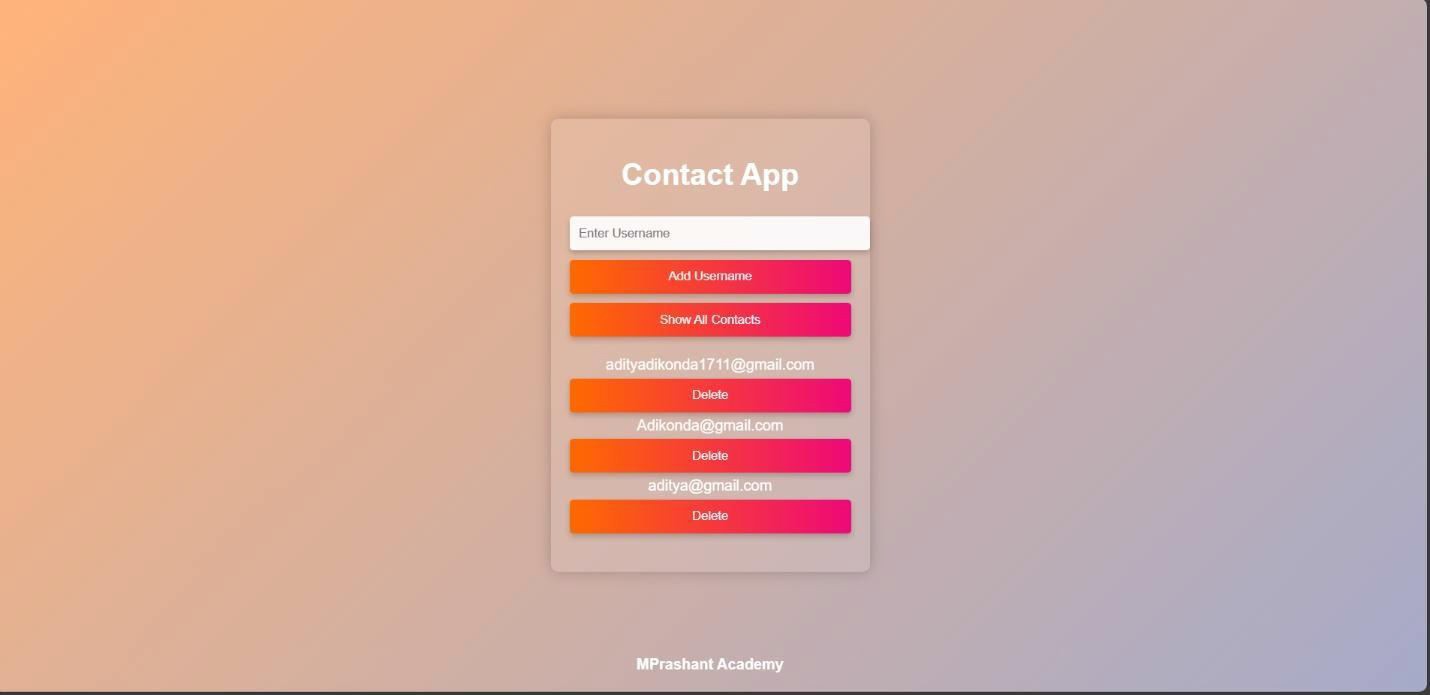




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

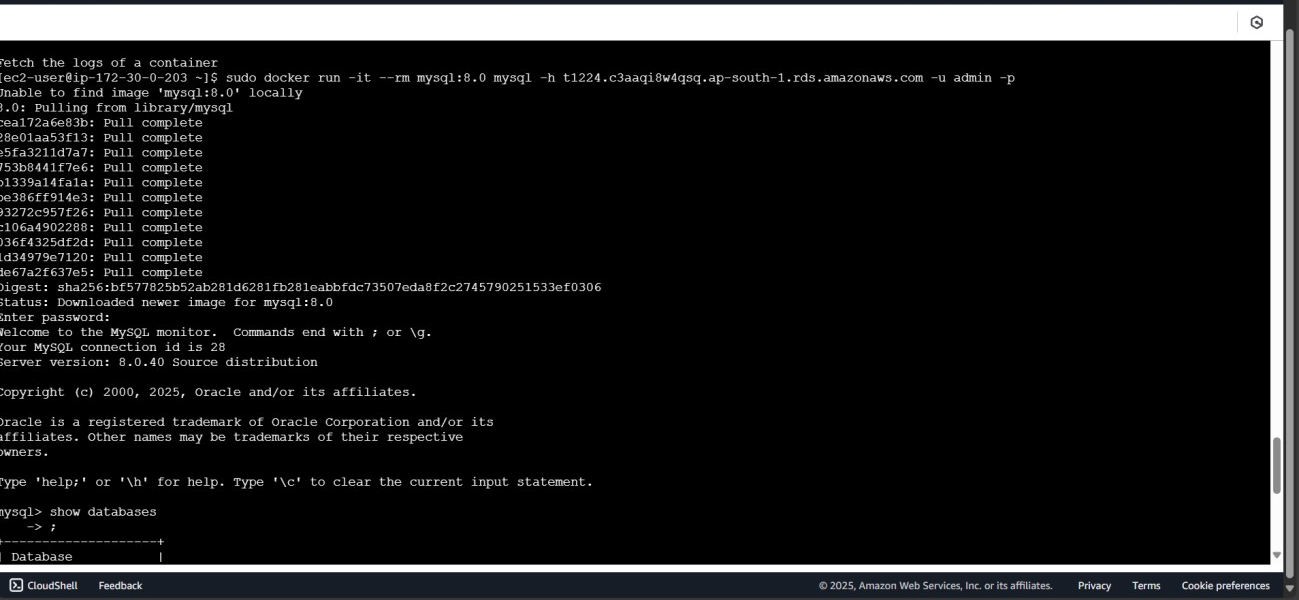
# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

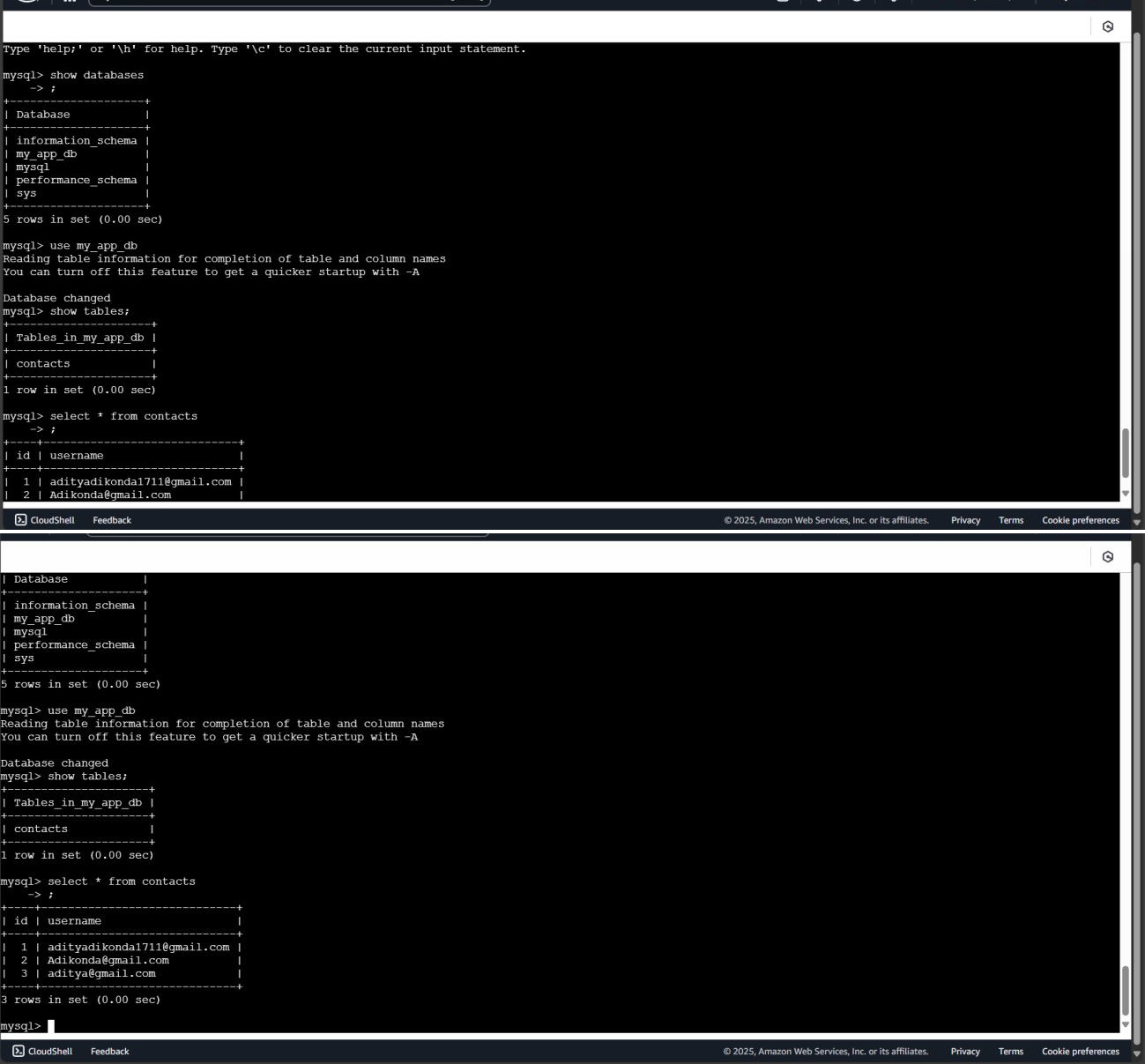




TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.





TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35

# Software Engineering and Project Management Lab Experiment No: - 09 Aim: To Study and Implement a Container using Docker.

**Conclusion:** We have successfully understood Docker Architecture and Container Life Cycle, installed Docker and executed docker commands to manage images and interact with containers.

**LO Mapping:** *LO is mapped*

TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35