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
   i. Download Docker Desktop from https://www.docker.com.
   ii. Install and start Docker Engine on your system (Linux, Windows, or macOS).
   iii. Verify installation by running:
   docker --version

2. Pull a Docker Image
   i. Use Docker Hub to pull a pre-built image:
   docker pull nginx

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

3. Run a Docker Container
   i. Start a container using the pulled image:
      docker run -d -p 8080:80 nginx
   ii. Access the web server by navigating to http://localhost:8080 in your browser.

4. Build a Custom Docker Image
   i. Create a file named Dockerfile:
      FROM python:3.10
      COPY app.py /app/app.py
      WORKDIR /app
      CMD ["python", "app.py"]
   ii. Build the image using:
      docker build -t my-python-app .
   iii. Run the container:
   docker run -d -p 5000:5000 my-python-app

5. Manage Containers
   i. View running containers:
      docker ps
   ii. Stop a container:
      docker stop <container_id>
   iii. Remove a container:
      docker rm <container_id>

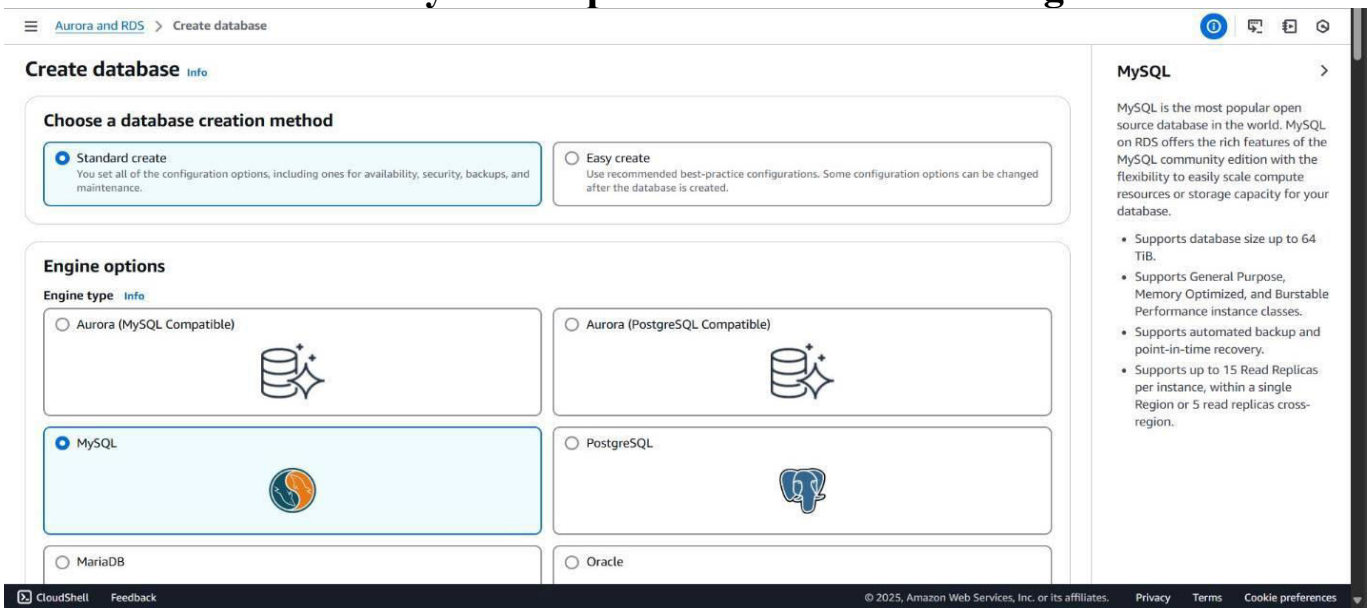
6. Use Docker Compose (Optional)
   i. Create a docker-compose.yml file to define multiple services.
   ii. Start all services using:
      docker-compose up

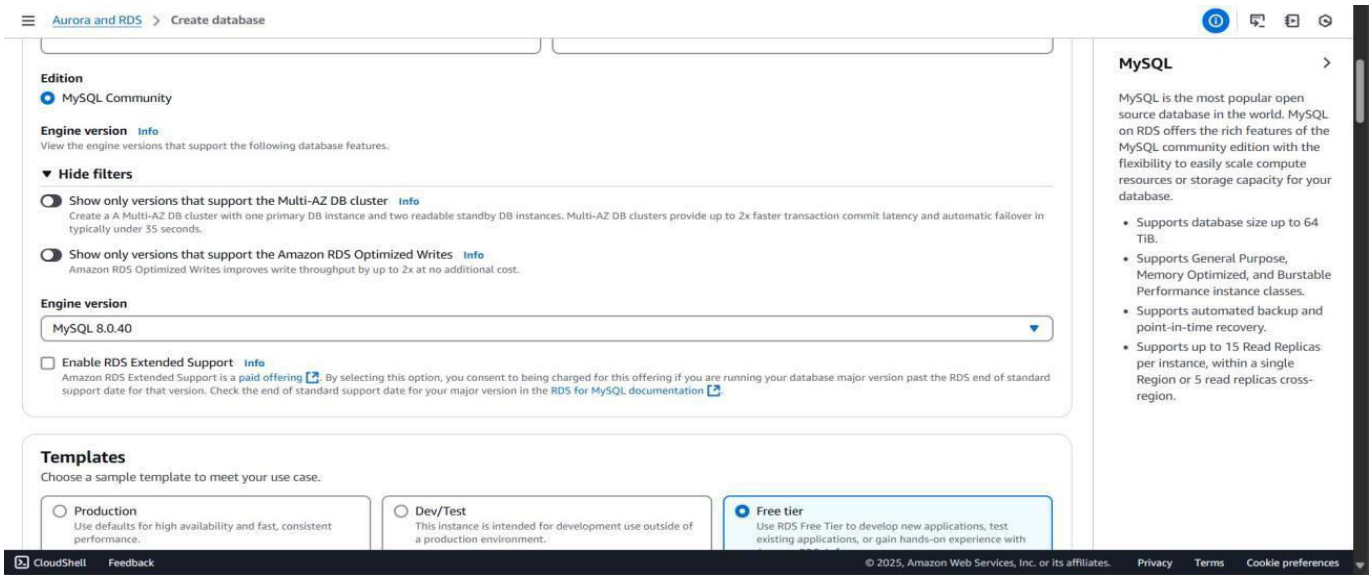
Use Case Example:

- 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

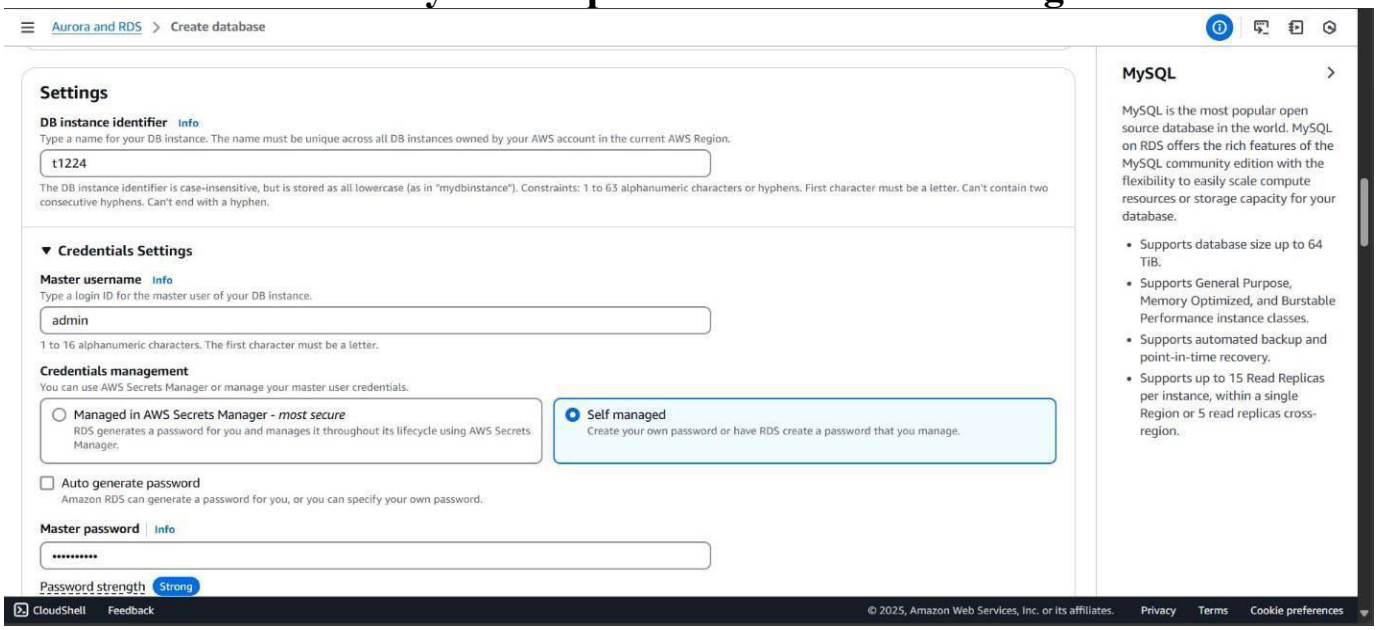
Implementation:

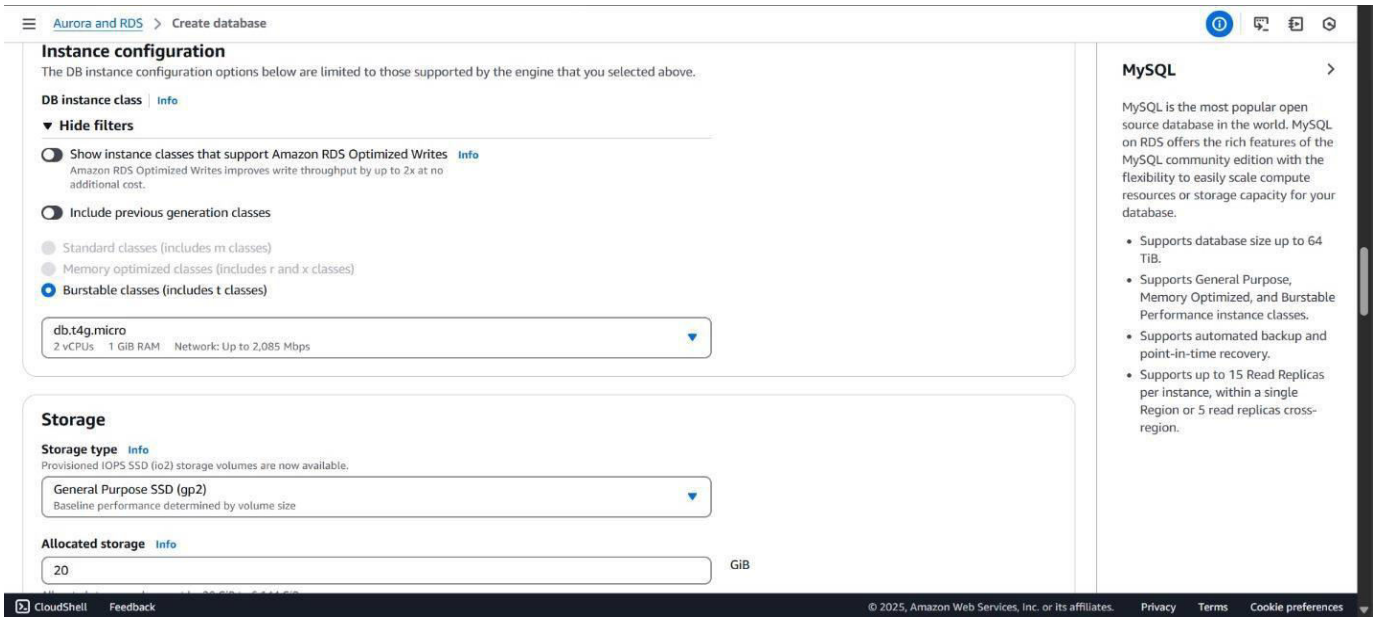
Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.



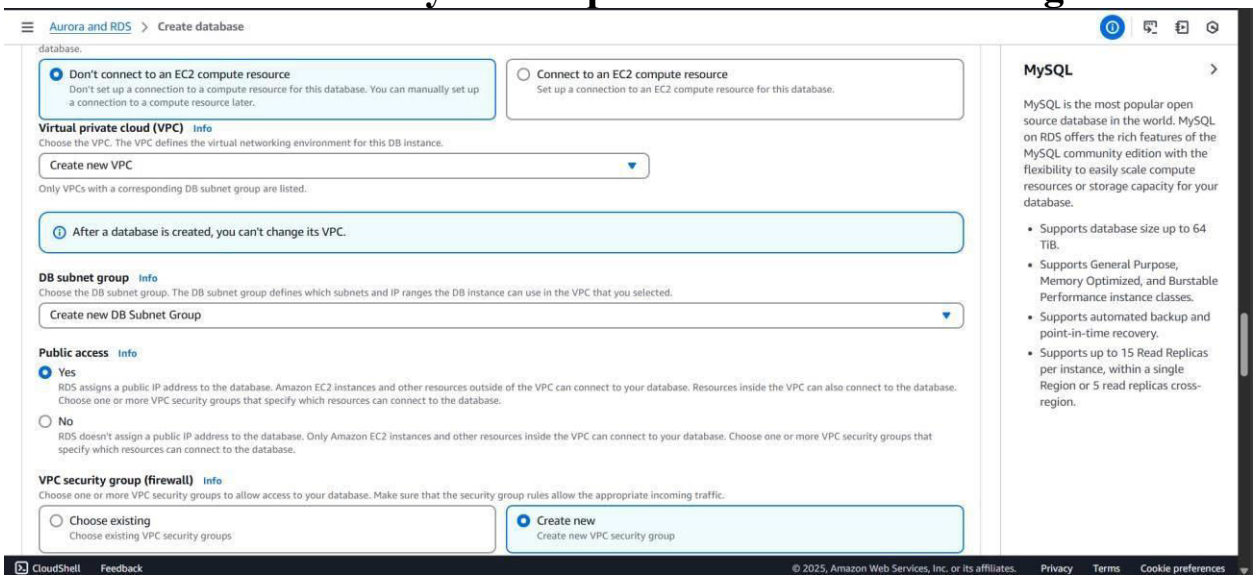


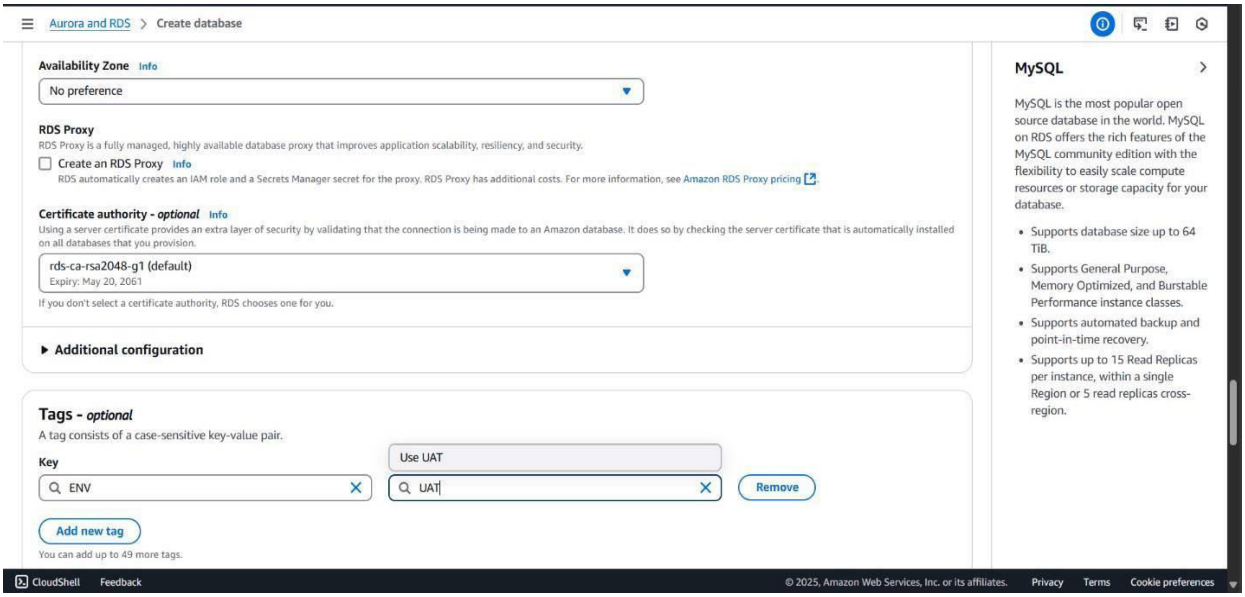
Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.



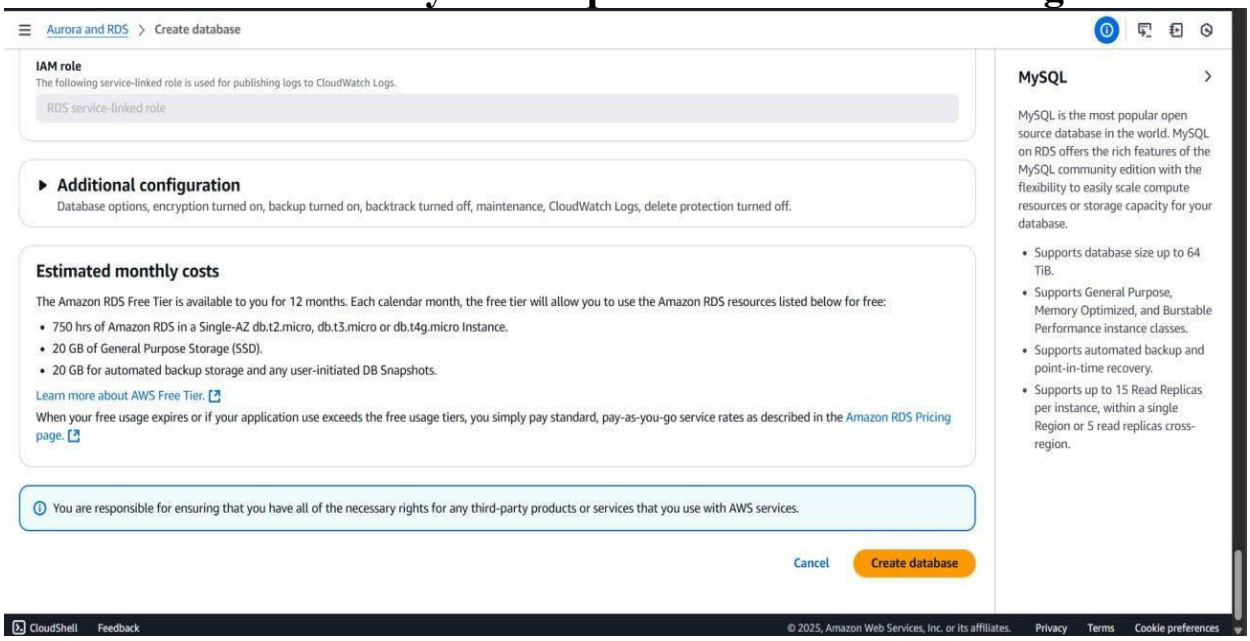


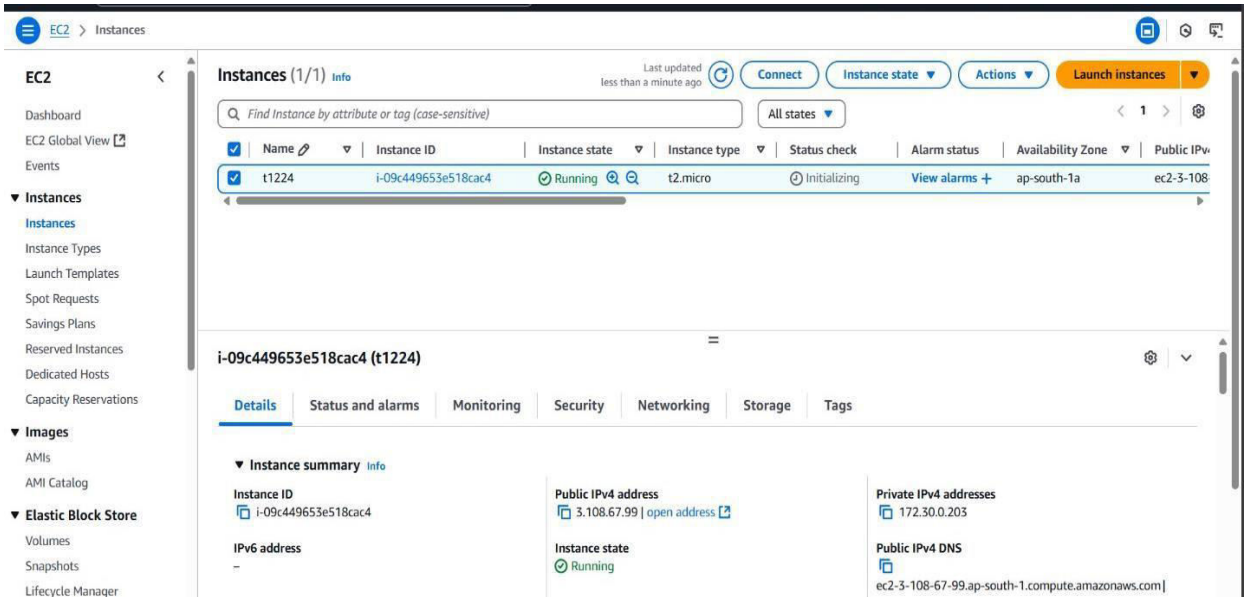
Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.



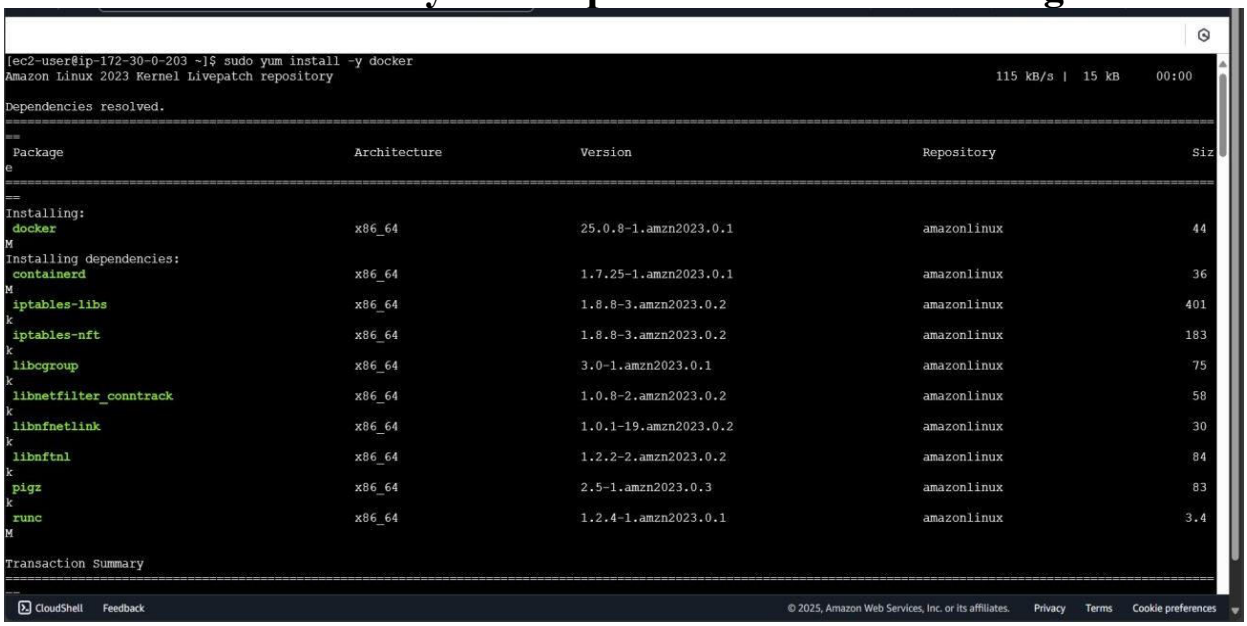


Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.



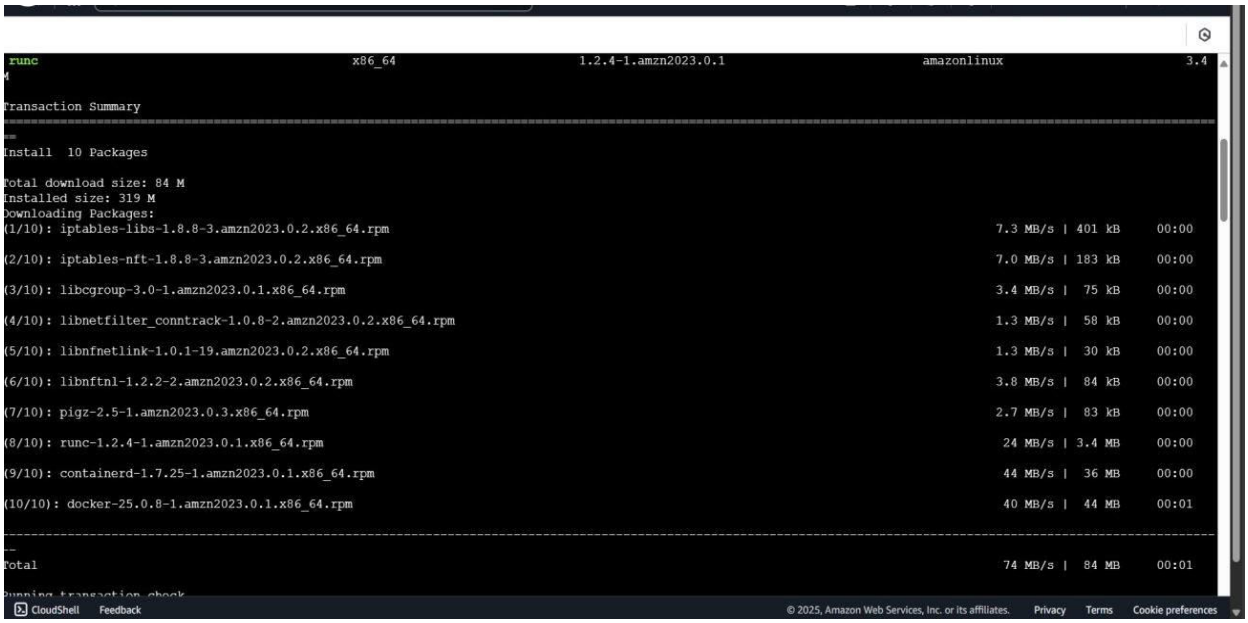


Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.



```
[ec2-user@ip-172-30-0-203 ~]$ sudo yum install -y docker
Amazon Linux 2023 Kernel Livepatch repository                                                    115 kB/s |  15 kB     00:00

Dependencies resolved.
================================================================================
 Package                 Architecture        Version                 Repository                Size
================================================================================
Installing:
 docker                  x86_64              25.0.8-1.amzn2023.0.1   amazonlinux               44 M
Installing dependencies:
 containerd              x86_64              1.7.25-1.amzn2023.0.1   amazonlinux               36 M
 iptables-libs           x86_64              1.8.8-3.amzn2023.0.2    amazonlinux              401 k
 iptables-nft            x86_64              1.8.8-3.amzn2023.0.2    amazonlinux              183 k
 libcgroup               x86_64              3.0-1.amzn2023.0.1      amazonlinux               75 k
 libnetfilter_conntrack  x86_64              1.0.8-2.amzn2023.0.2    amazonlinux               58 k
 libnfnetlink            x86_64              1.0.1-19.amzn2023.0.2   amazonlinux               30 k
 libnftnl                x86_64              1.2.2-2.amzn2023.0.2    amazonlinux               84 k
 pigz                    x86_64              2.5-1.amzn2023.0.3      amazonlinux               83 k
 runc                    x86_64              1.2.4-1.amzn2023.0.1    amazonlinux              3.4 M

Transaction Summary
================================================================================
```

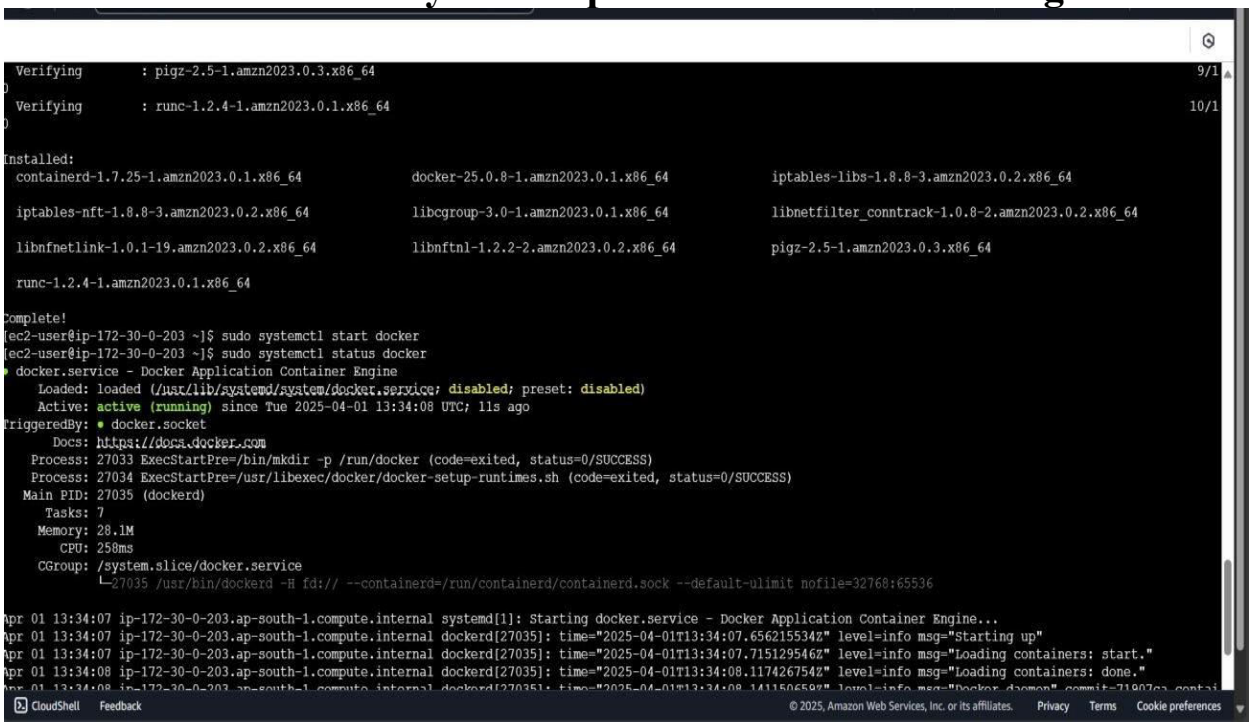
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

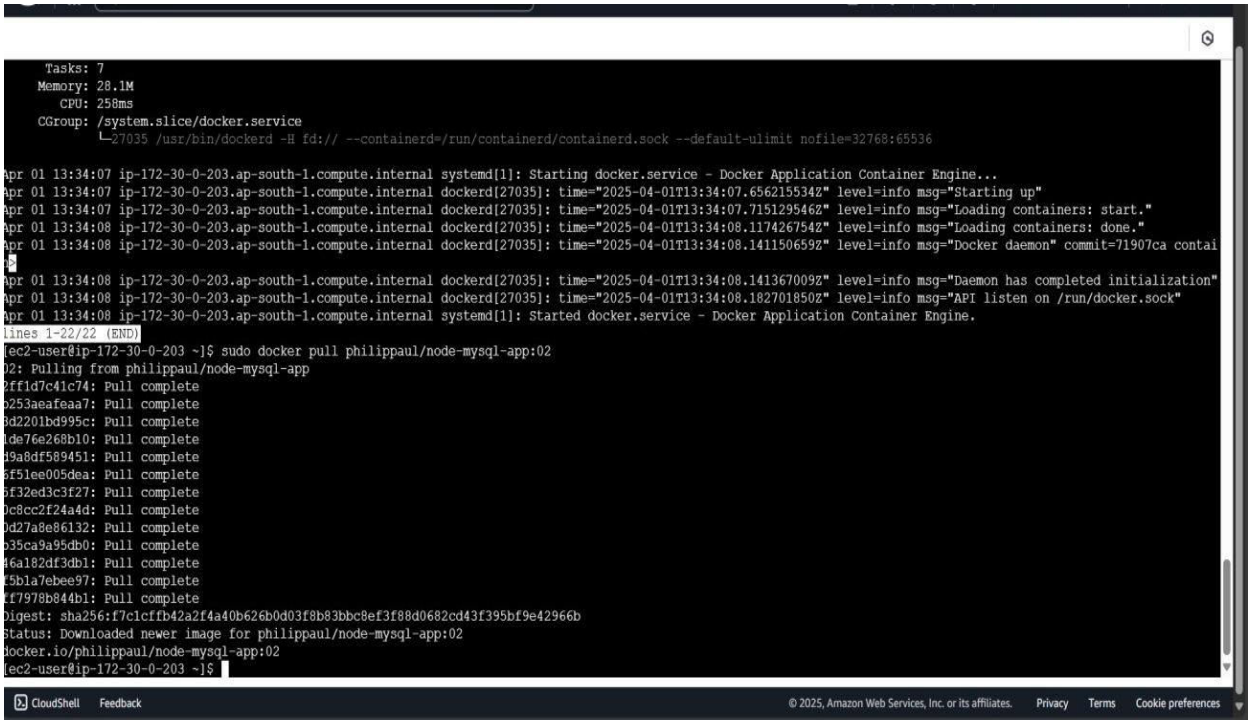


```
 runc                    x86_64              1.2.4-1.amzn2023.0.1    amazonlinux              3.4 M

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 319 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm          7.3 MB/s | 401 kB     00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm           7.0 MB/s | 183 kB     00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm                3.4 MB/s |  75 kB     00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.3 MB/s |  58 kB     00:00
(5/10): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm          1.3 MB/s |  30 kB     00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm               3.8 MB/s |  84 kB     00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                     2.7 MB/s |  83 kB     00:00
(8/10): runc-1.2.4-1.amzn2023.0.1.x86_64.rpm                    24 MB/s | 3.4 MB     00:00
(9/10): containerd-1.7.25-1.amzn2023.0.1.x86_64.rpm             44 MB/s |  36 MB     00:00
(10/10): docker-25.0.8-1.amzn2023.0.1.x86_64.rpm                40 MB/s |  44 MB     00:01
--------------------------------------------------------------------------------
Total                                                           74 MB/s |  84 MB     00:01

Running transaction check
```

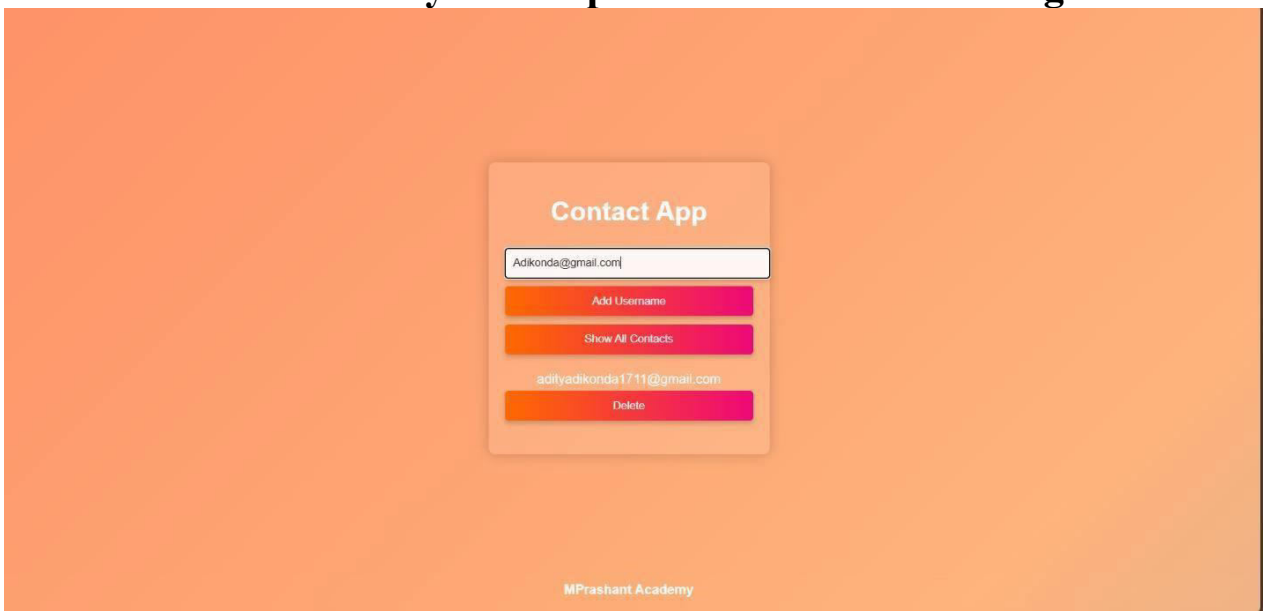
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

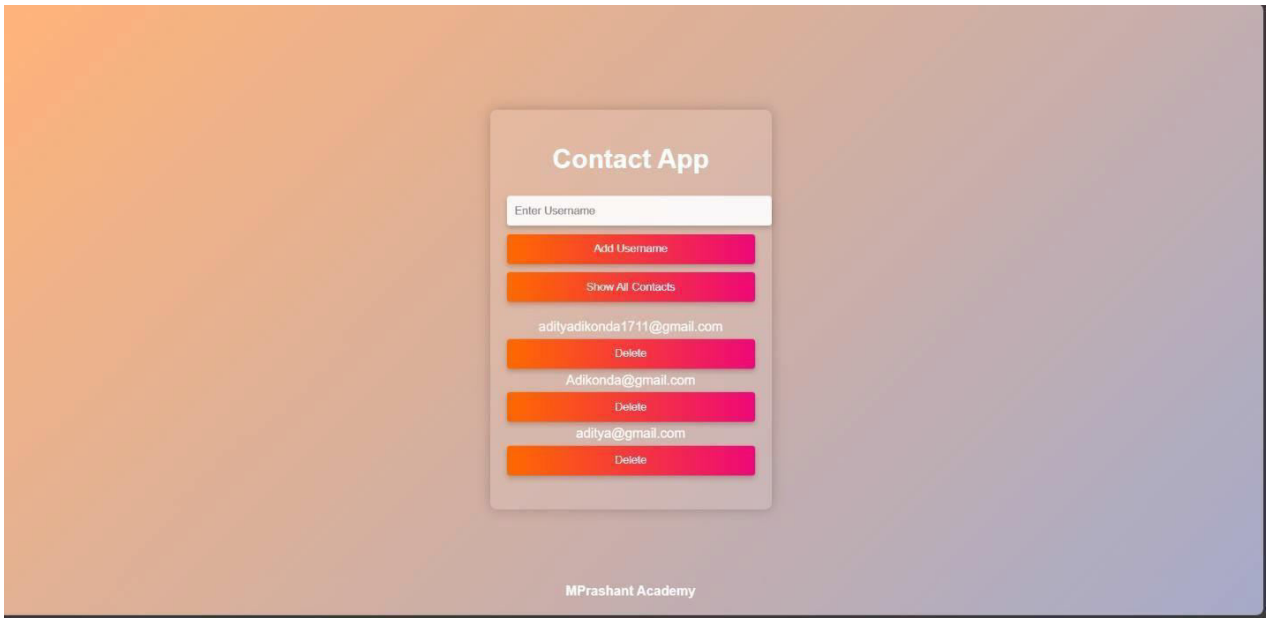
Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.



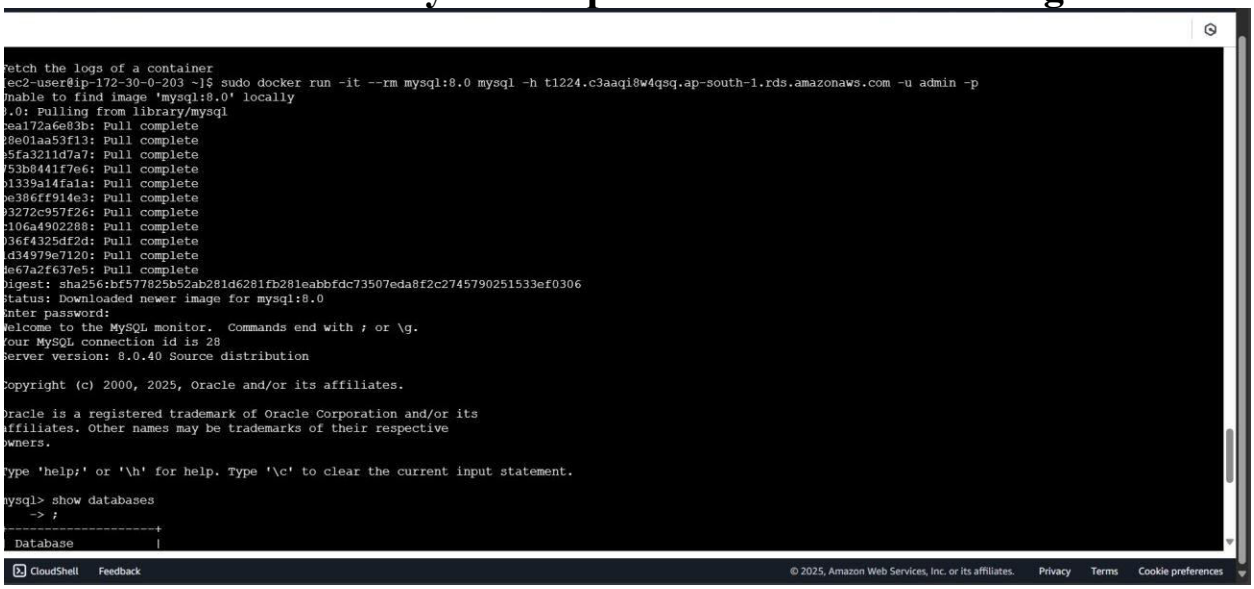


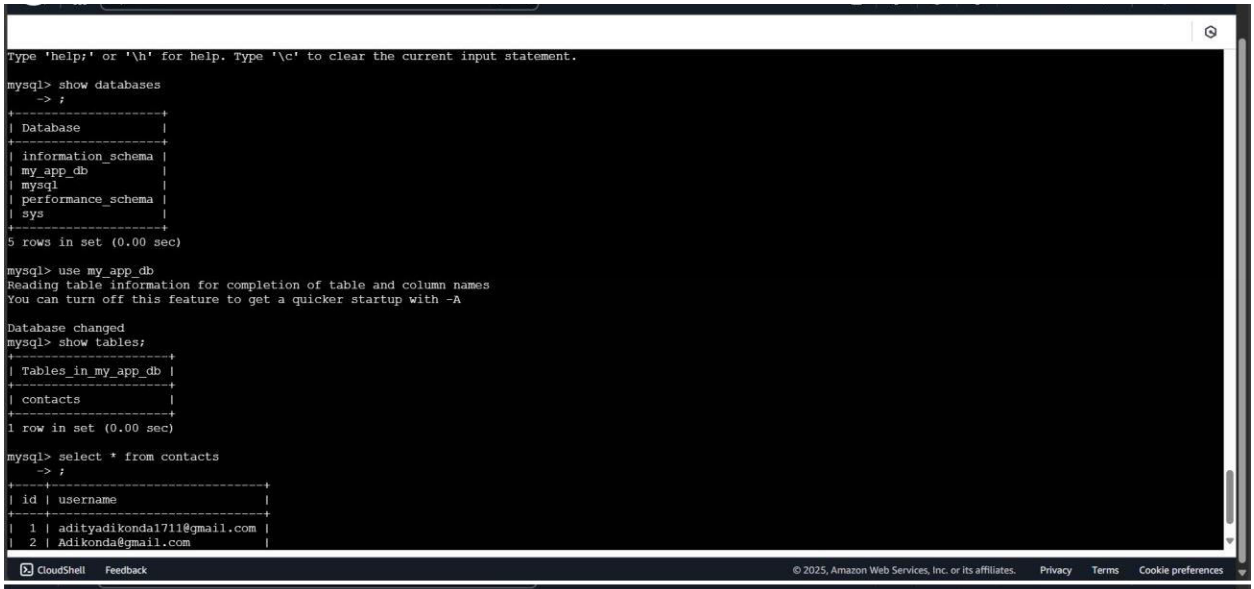
Software Engineering and Project Management Lab Experiment No: - 09
Aim: To Study and Implement a Container using Docker.





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





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*