**Experiment No. 09**

# AIM: To study and implement Docker basics

**THEORY:**

**Theory:**

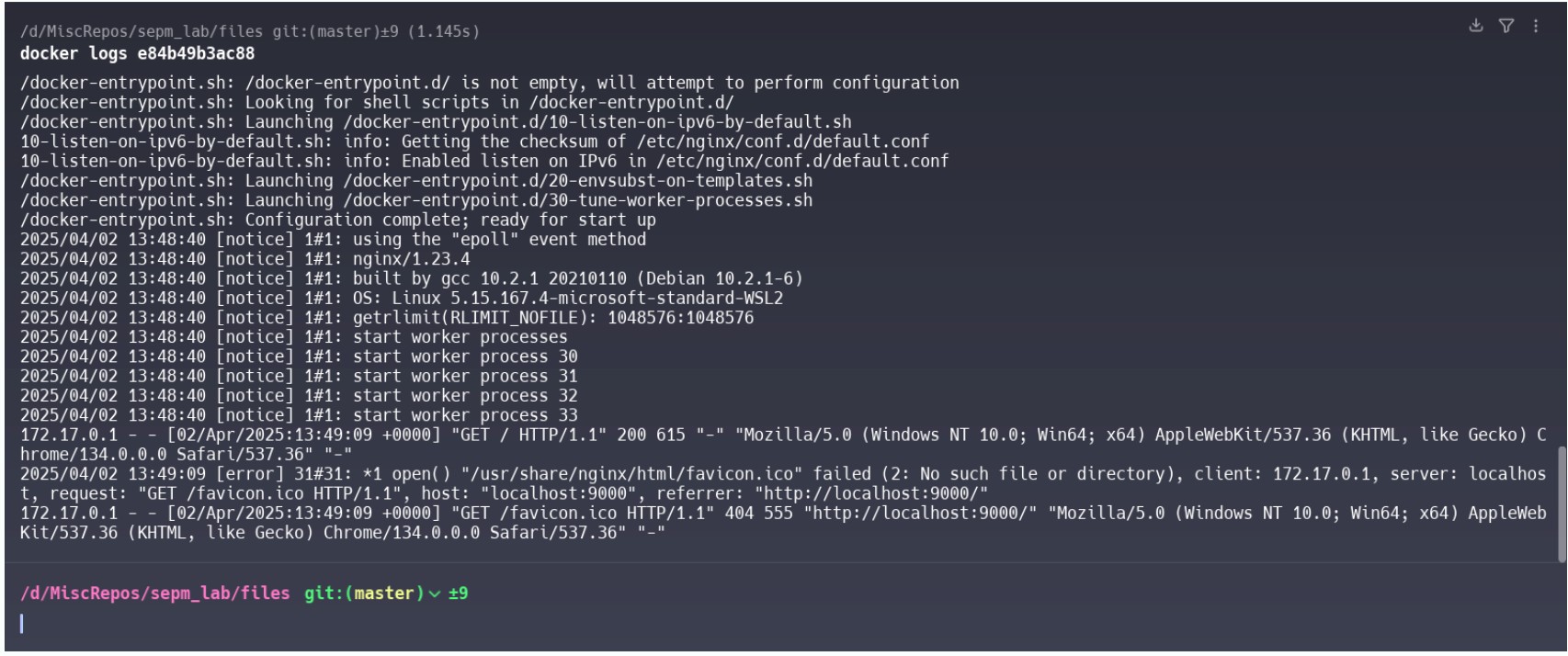
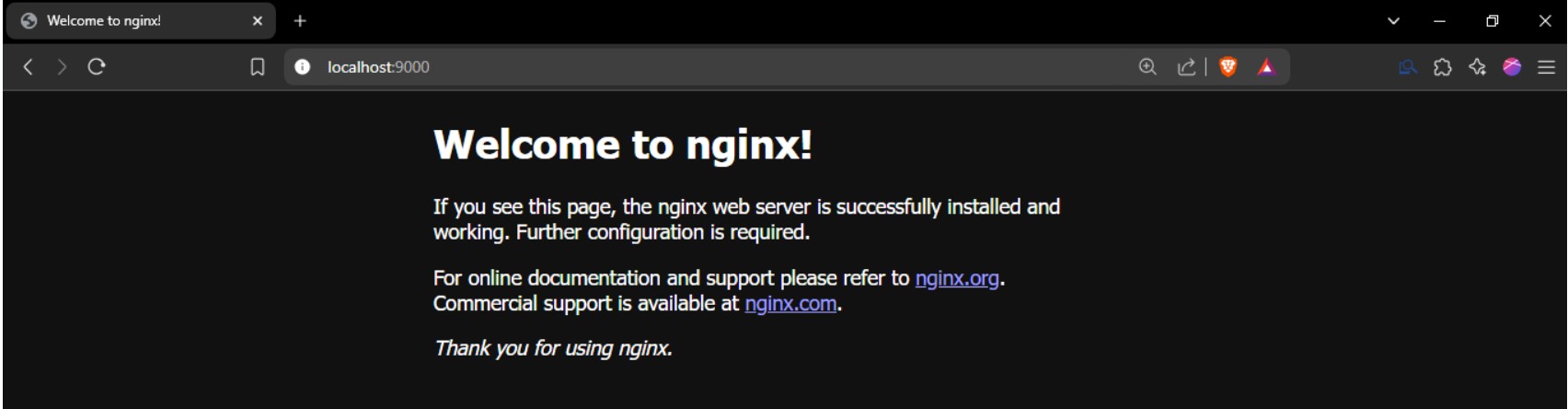
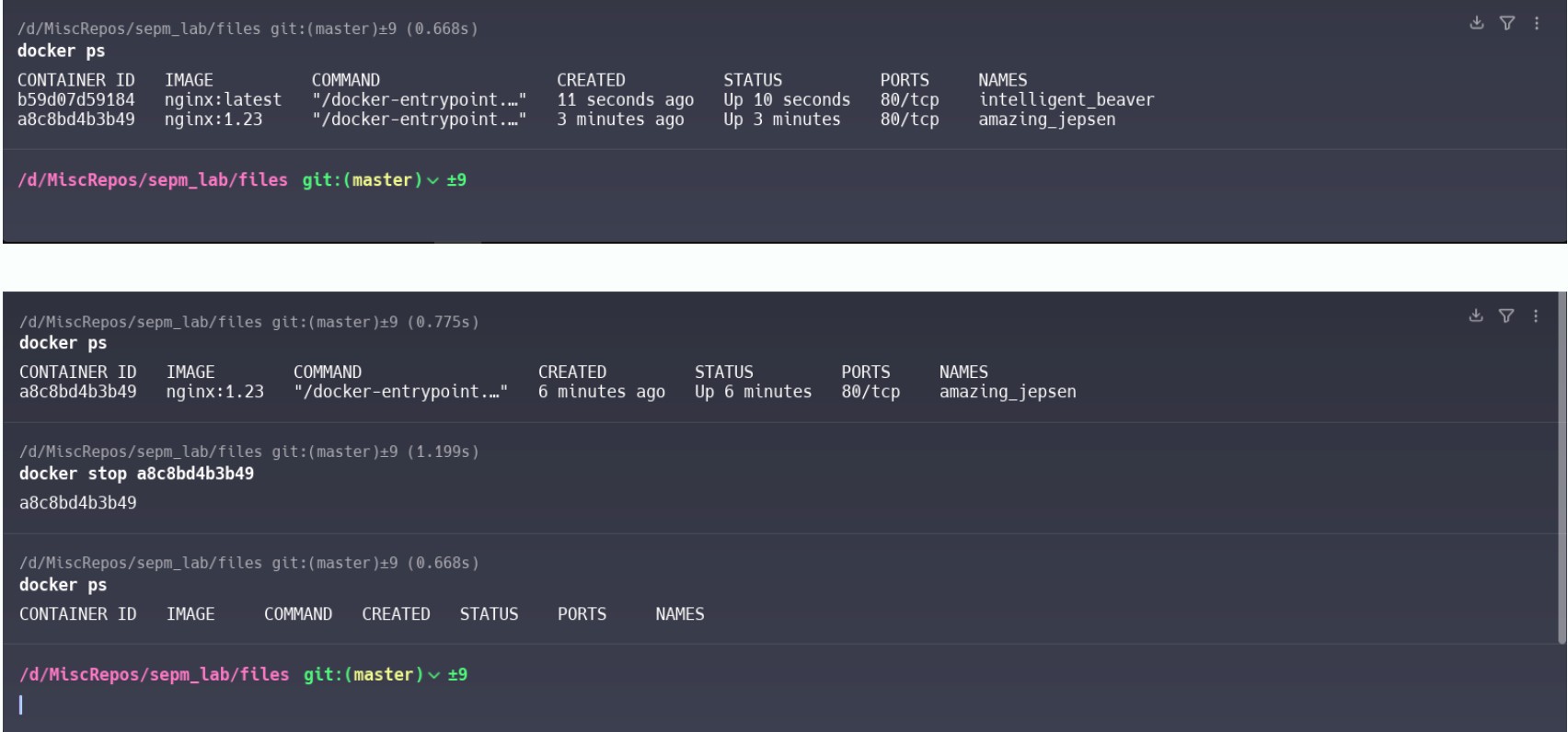
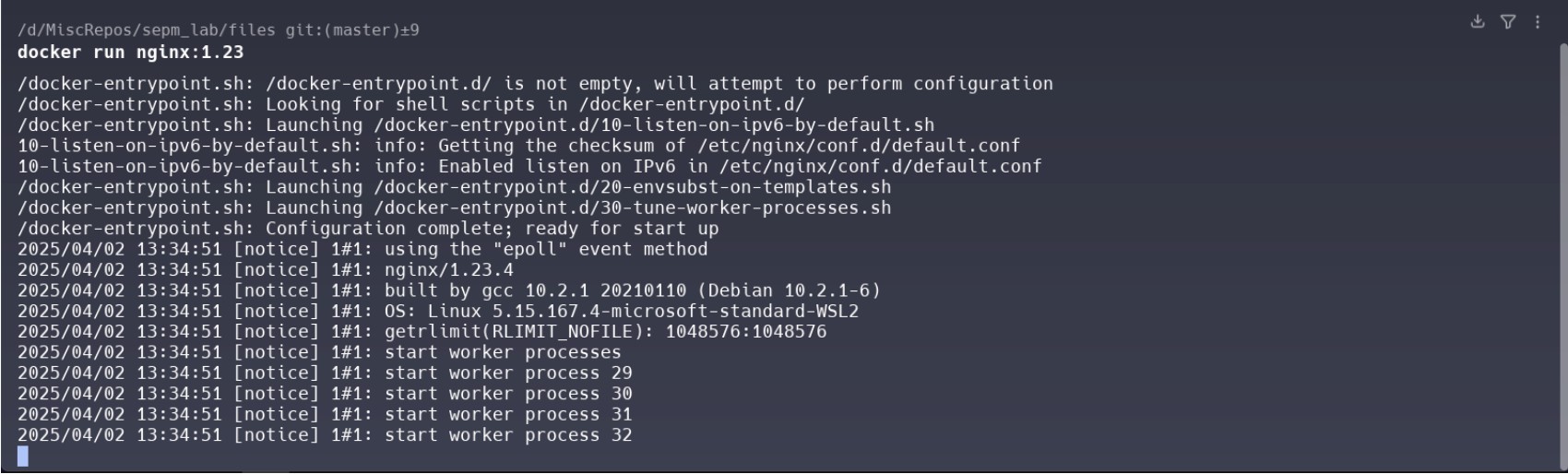
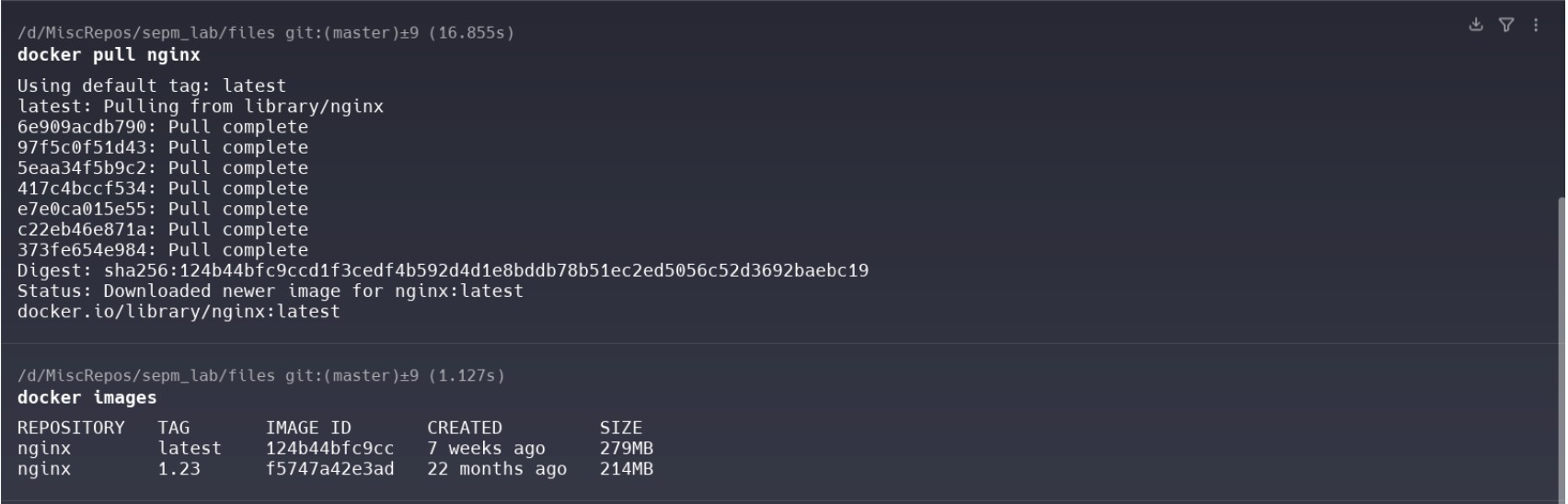
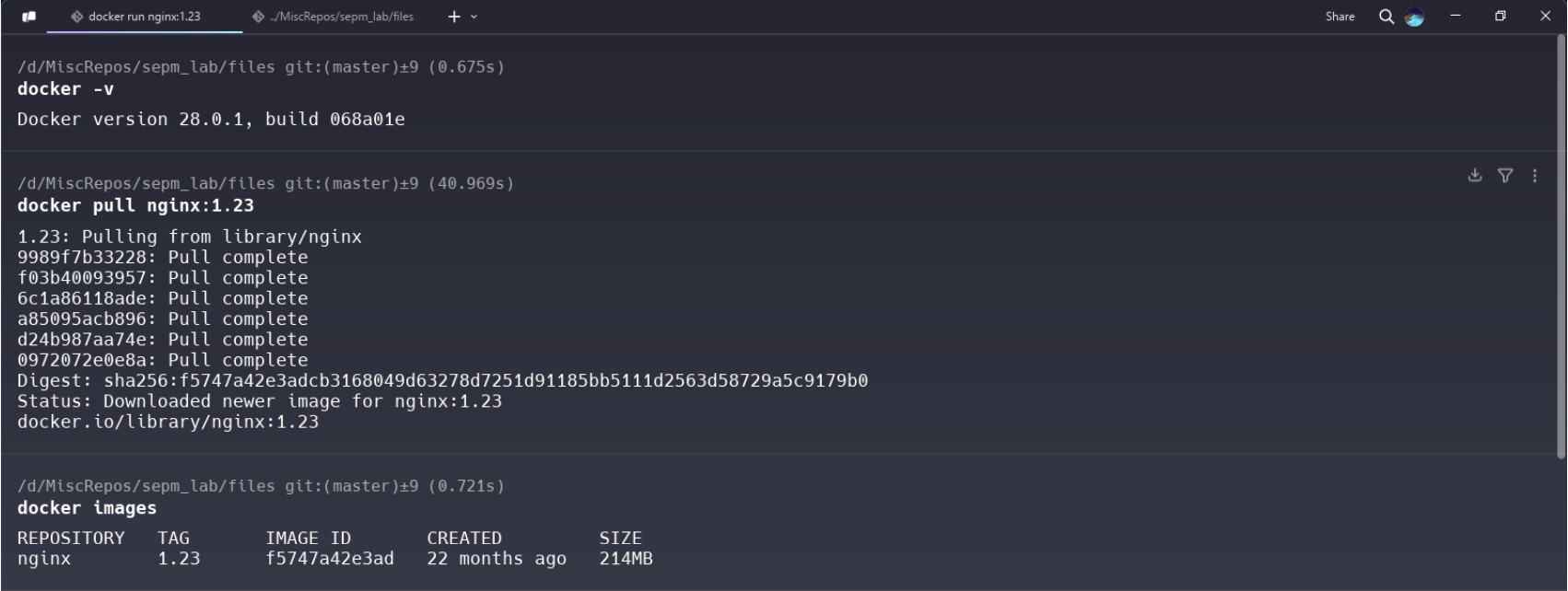
Containerization is a lightweight virtualization technique that allows applications to run in isolated environments called containers. Unlike traditional virtual machines, containers share the host operating system’s kernel, making them more efficient and faster to start. Docker is one of the most popular platforms for containerization.

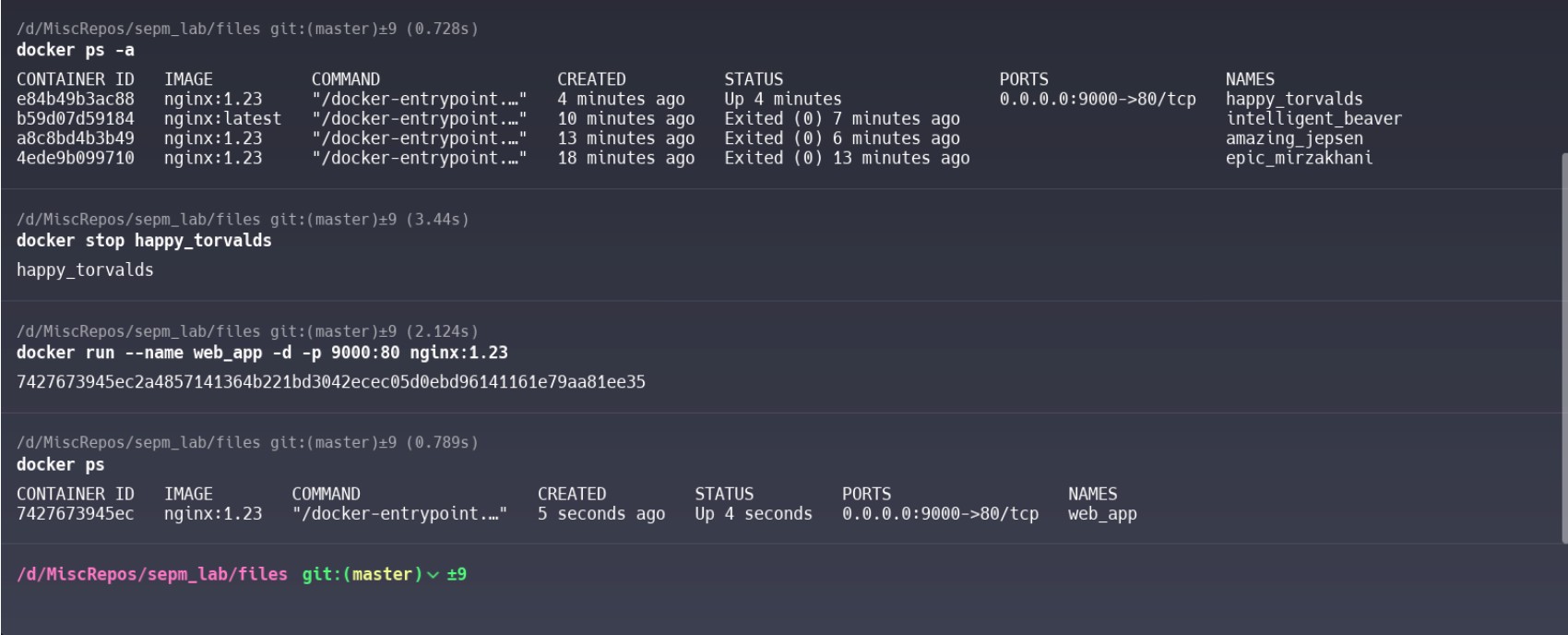
Docker allows developers to package an application along with all its dependencies, libraries, and configuration files into a single unit called a Docker container. This ensures the application runs consistently across different environments—whether it’s development, testing, or production. Docker uses a file called a *Dockerfile* to automate the creation of images, which are then used to spin up containers.

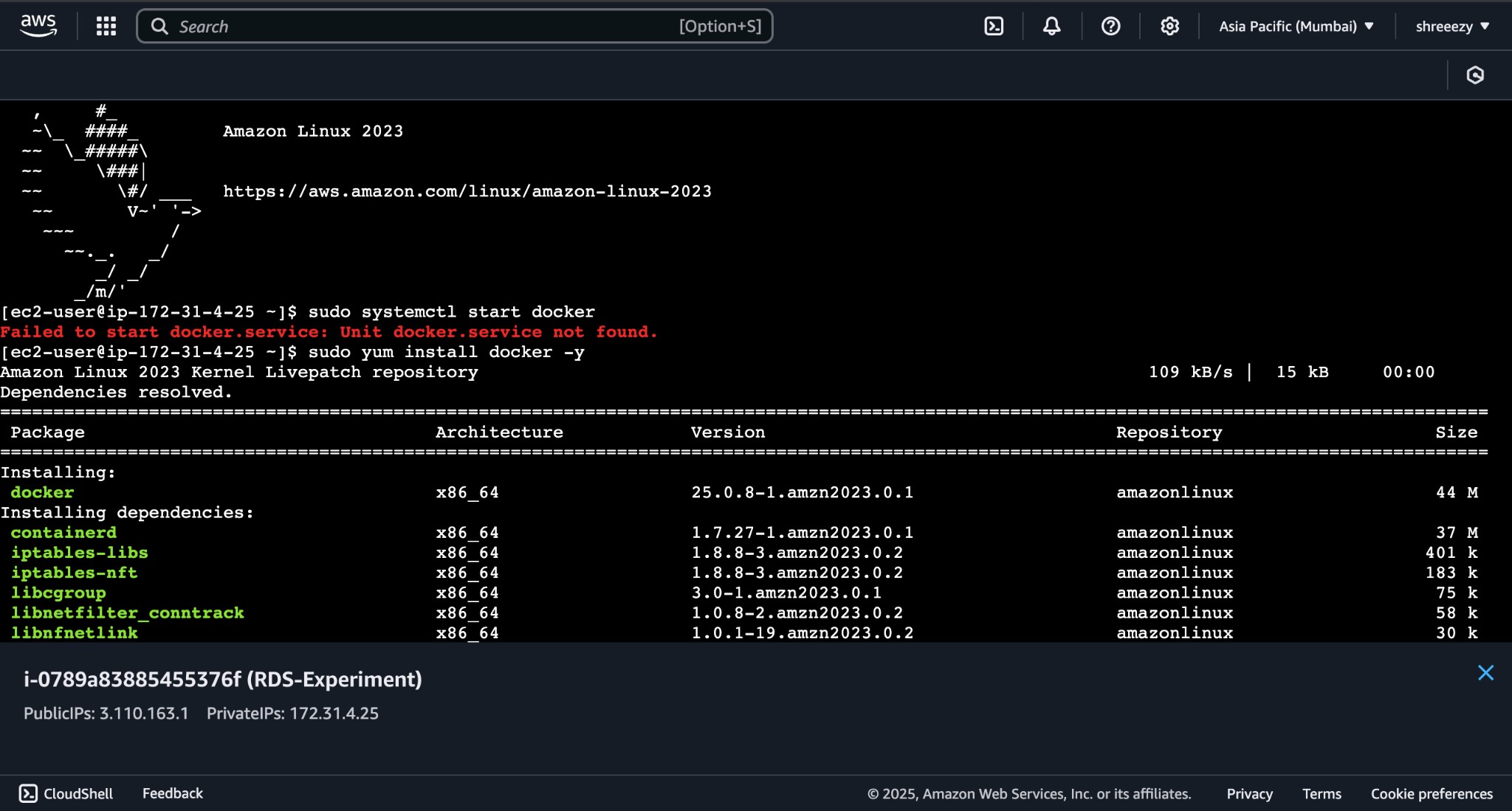
With Docker, scaling applications becomes simpler, and deployment is faster and more reliable. It also helps in reducing conflicts between environments and improves CI/CD pipelines.

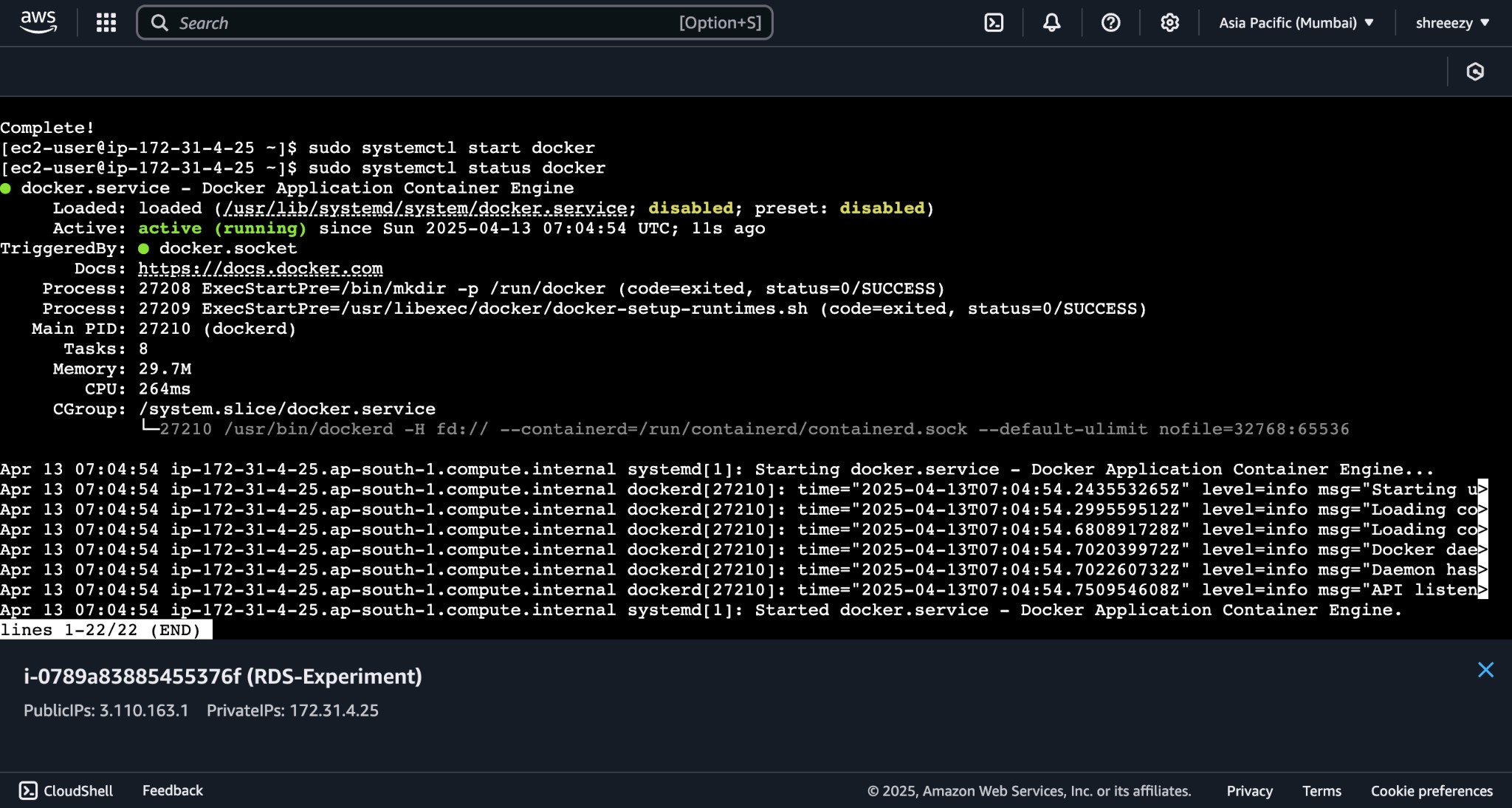
In this experiment, we aim to understand how Docker works, how to build Docker images, and how to run and manage containers. Through hands-on implementation, we’ll explore Docker commands and learn how containerization can streamline development workflows.

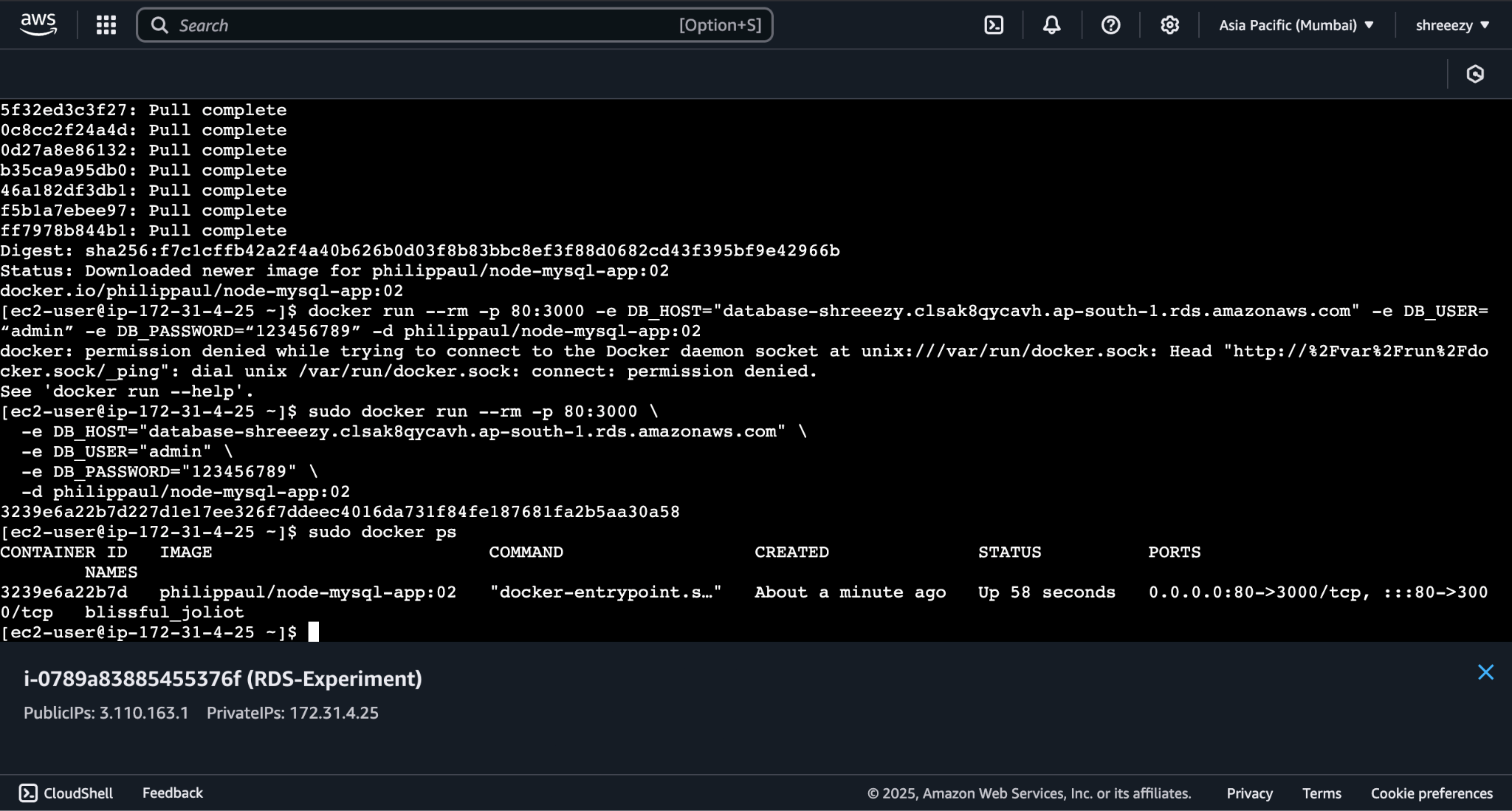
**IMPLEMENTATION:**





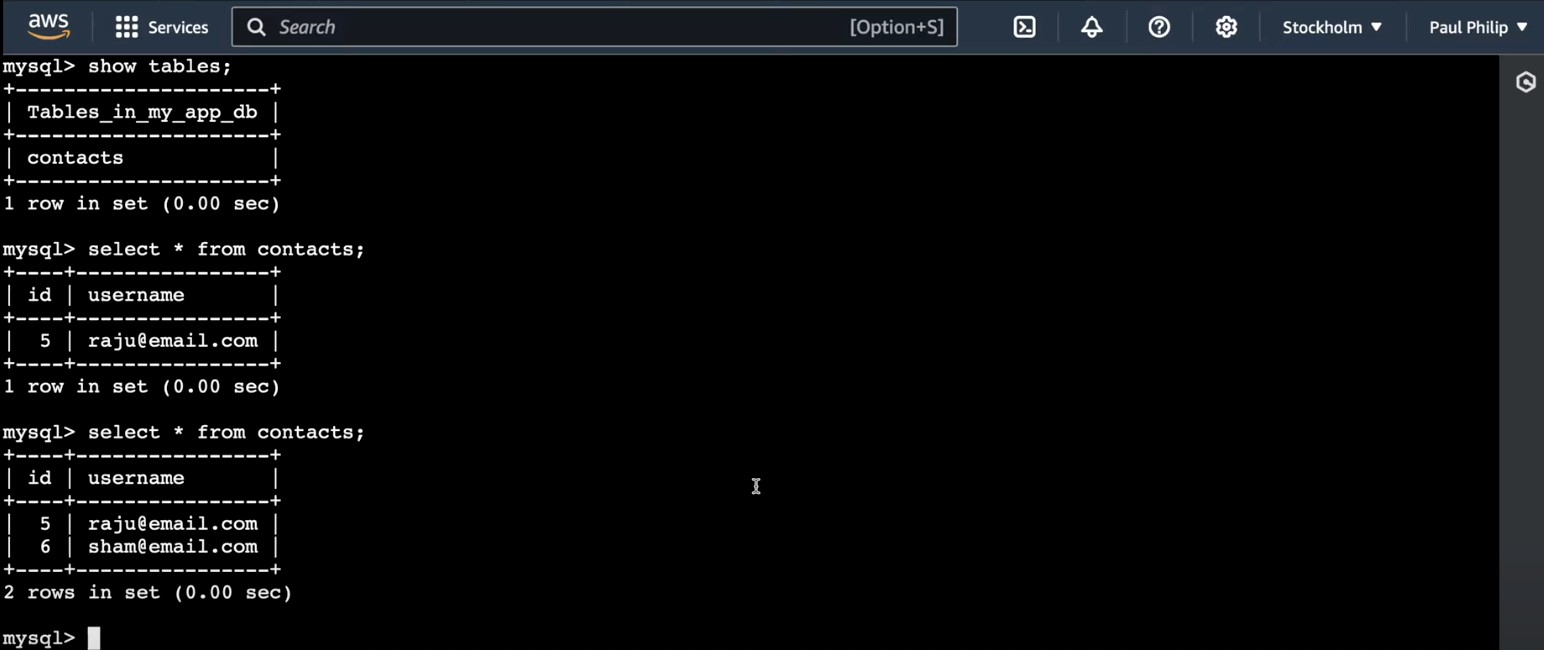


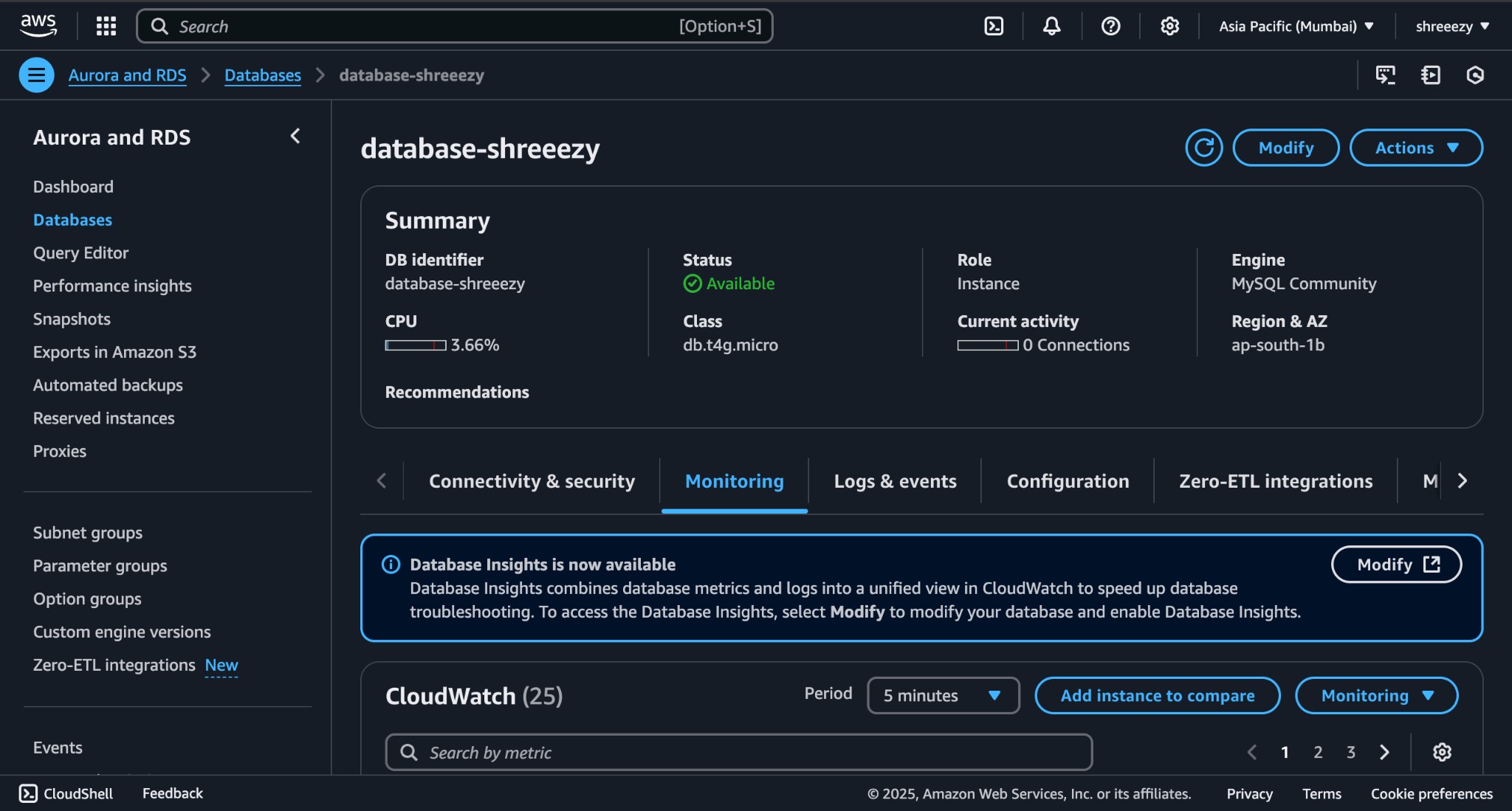












# CONCLUSION: Hence, we have successfully studied and implemented the

**basics of Docker.**