EXPERIMENT : 8

Date of Performance	
Date of Submission	

AIM

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

PROBLEM DEFINITION

Gain insight into Docker architecture and container lifecycle, including installing Docker and managing images and containers.

THEORY

Jenkins is an open-source automation tool created with Java. It is extensively used as a Continuous Integration (CI) and Continuous Deployment (CD) tool.

Maven:

- Maven primarily provides developers with:
    1. A comprehensive, reusable, and easily maintainable model for projects.
    2. Plugins or tools to interact with and operate within this model.
- Maven is a POM (Project Object Model)-based build automation and project management tool written in Java. However, it is compatible with projects written in C#, Python, Ruby, etc.

A few Maven features worth mentioning are:

1. Maven can be used to build projects into predefined output types like .jar, .war, metadata, etc.
2. Maven can automatically download necessary files from the repository when building a project.

Selenium Using Maven in Jenkins

Selenium is a widely used test automation framework for validating web applications across different combinations of browsers, platforms, and devices (or emulators). It is extensively used for testing areas such as functional testing, end-to-end testing, and more.

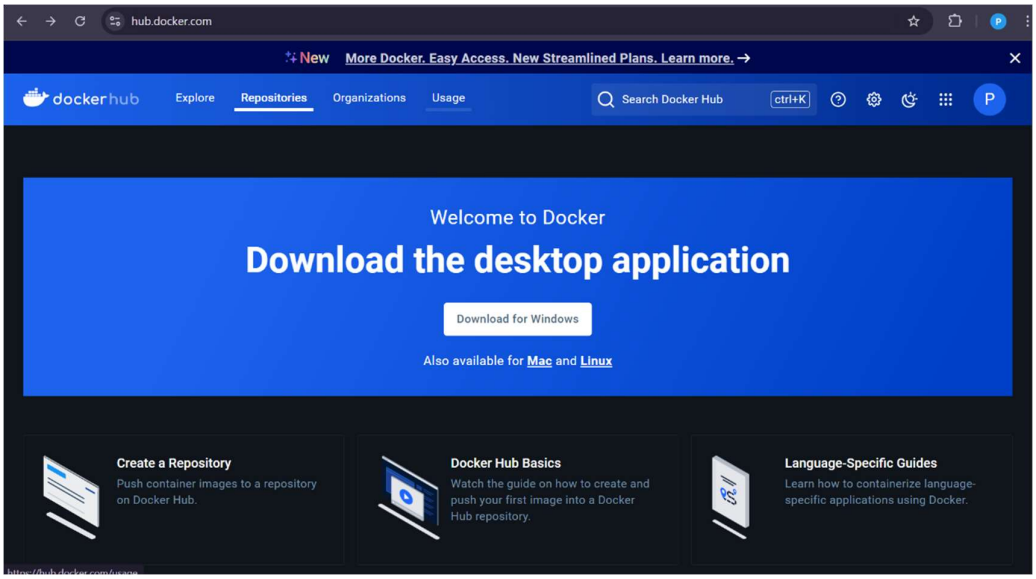
Why Jenkins and Selenium?

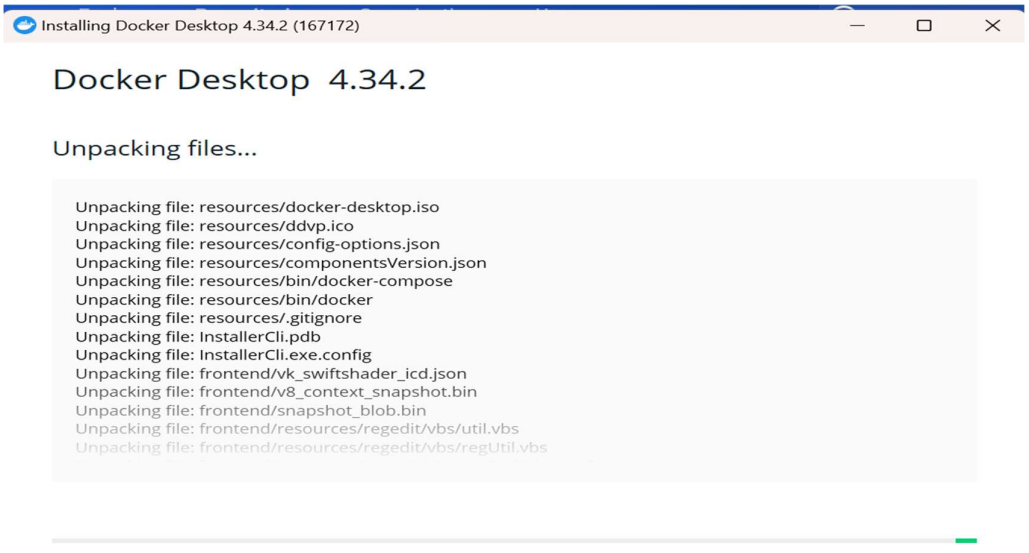
- Running Selenium tests in Jenkins allows you to execute your tests every time your software changes and deploy the software to new environments when the tests pass.

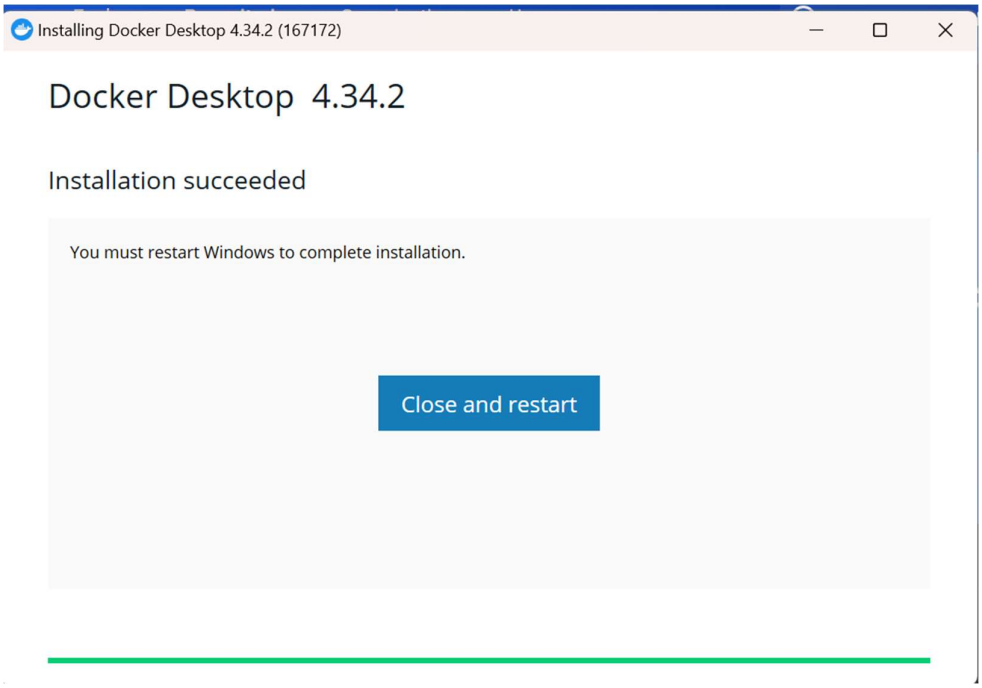
Advantages of Using Maven and Jenkins with Selenium:

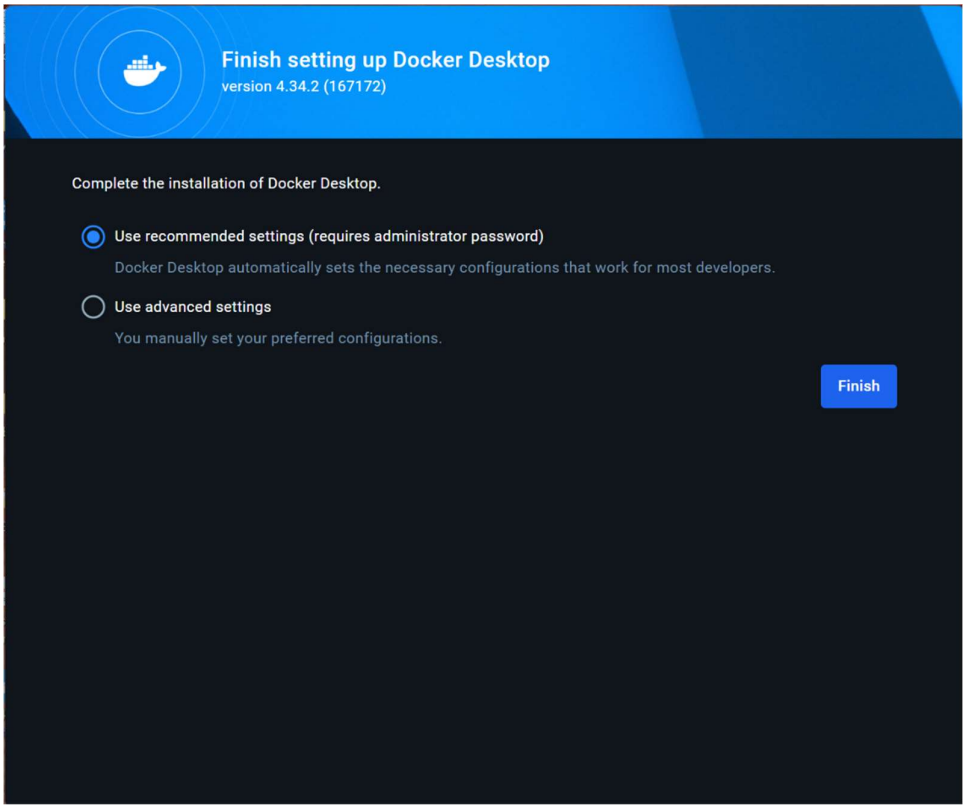
- Whenever a change is made in the implementation, the changes are deployed in the test environment. Automation testing is performed continuously, and developers are kept informed about the build and test stage results.
- Test suites comprising many test scenarios might take a longer duration for testing. In such cases, a nightly build run can be scheduled for build and execution on the Jenkins server.

OUTPUT

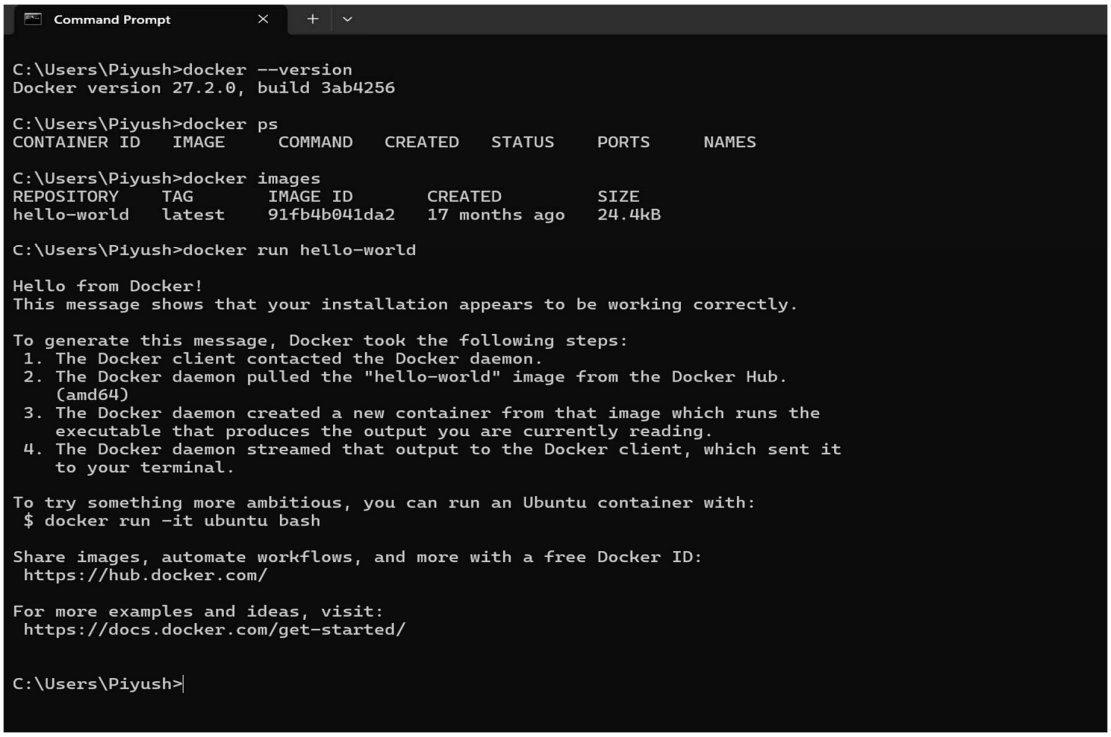




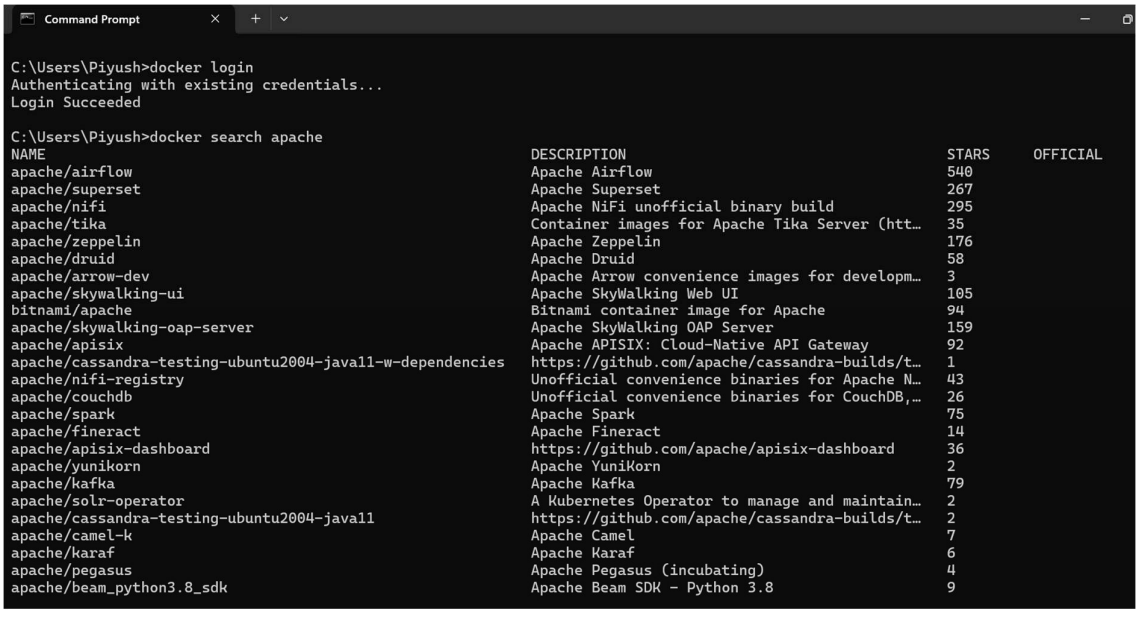

Installing Docker Desktop 4.34.2 (167172)
Docker Desktop 4.34.2
Installation succeeded
You must restart Windows to complete installation.
Close and restart


Finish setting up Docker Desktop
version 4.34.2 (167172)
Complete the installation of Docker Desktop.
Use recommended settings (requires administrator password)
Docker Desktop automatically sets the necessary configurations that work for most developers.
Use advanced settings
You manually set your preferred configurations.
Finish

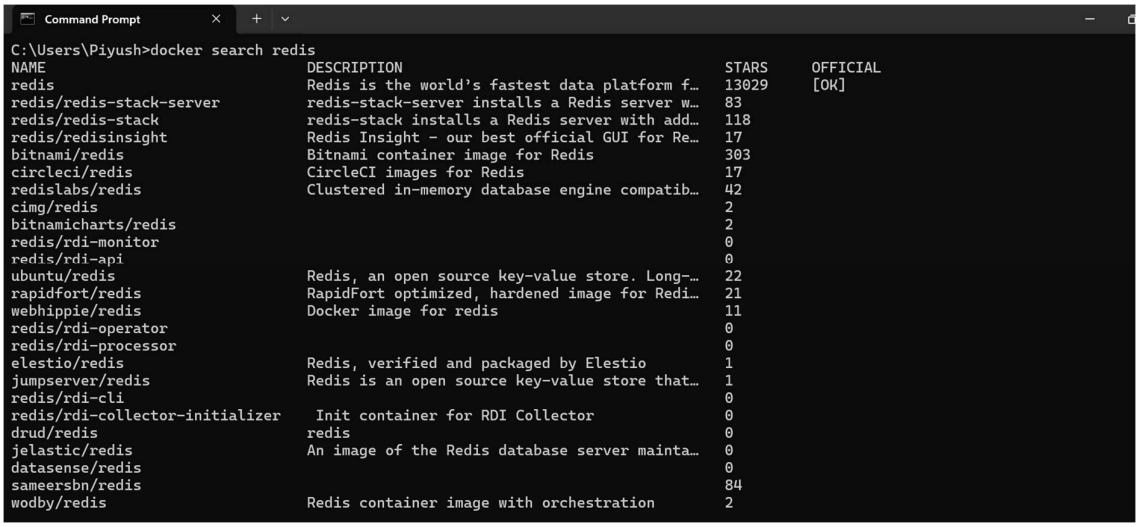
Docker –version, docker ps, docker run



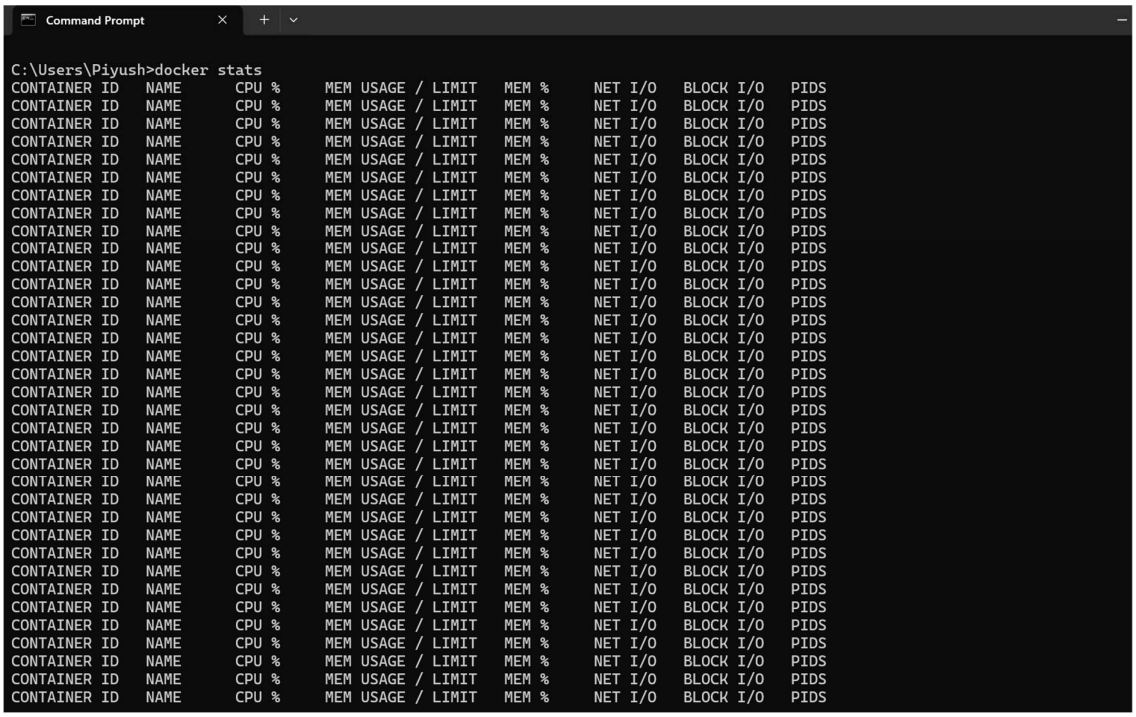
Docker login, docker search apache



Docker search redis

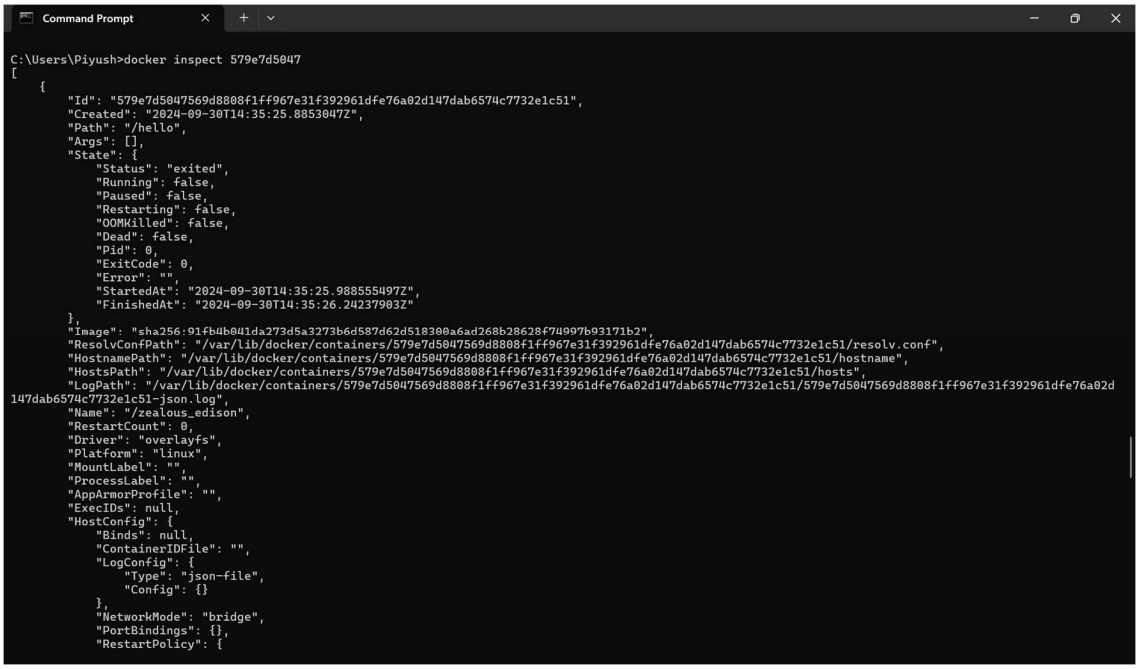


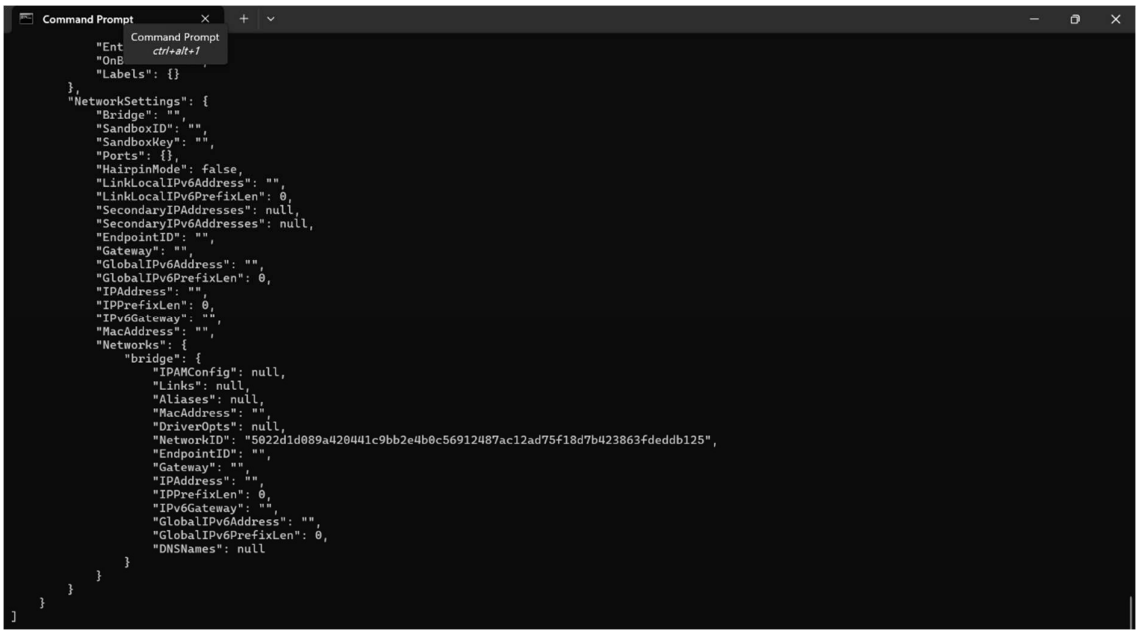
Docker stats



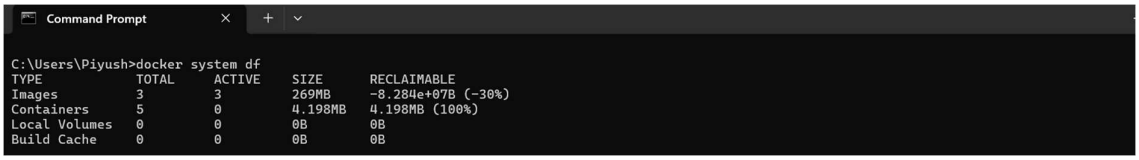
Docker network ls

Docker inspect

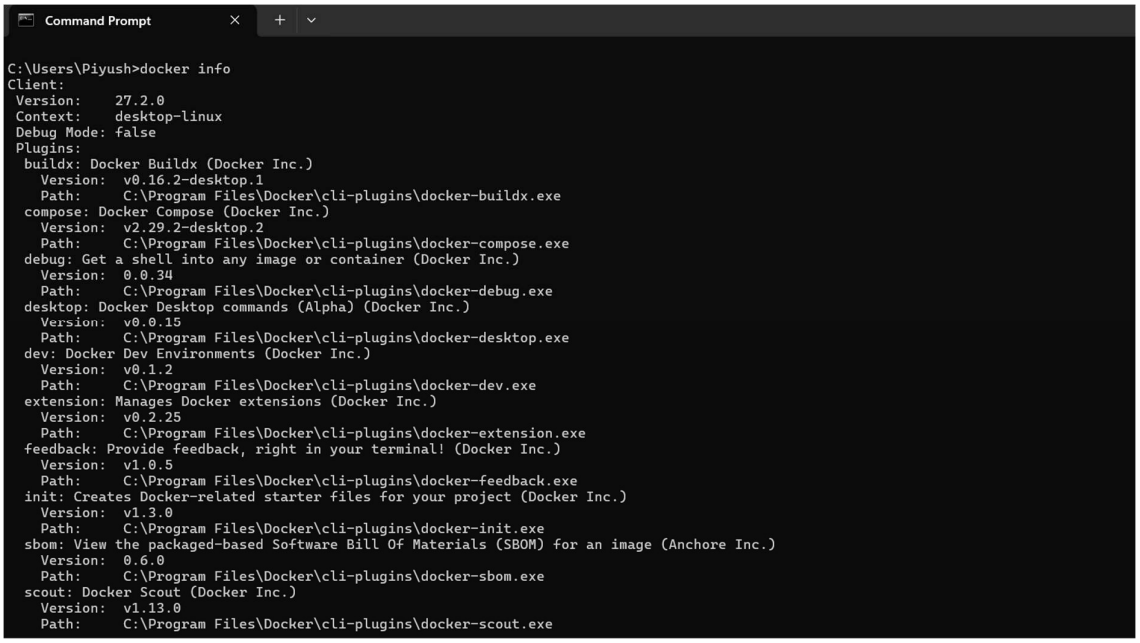


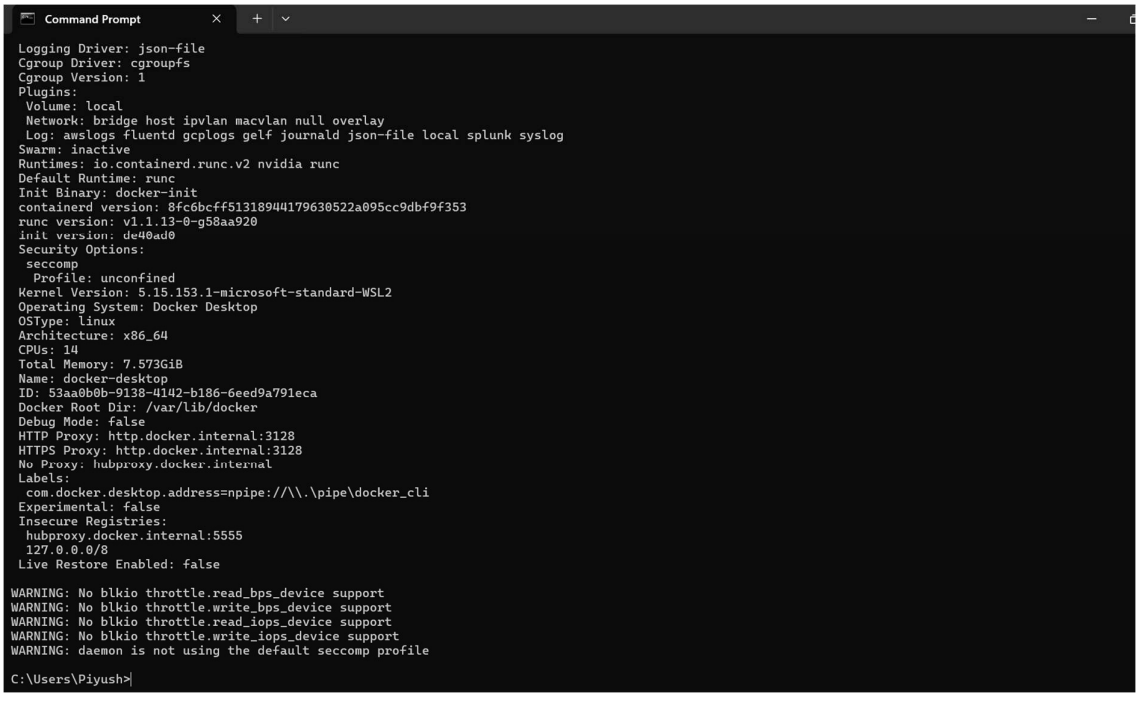


Docker system df



Docker info





CONCLUSION

As a result, we understood Docker architecture and container life cycle by installing Docker and executing commands to manage images and interact with containers effectively.