Industry Based Project 1 Assignment Report

Business Challenge/Requirement

ABC technologies is a leading online retail store. ABC has recently acquired a large retails offline business store. The business store has large number of stores across the globe but is following conventional pattern of development and deployment. As a result, it has landed to great loss and are facing below challenges.

- low available
- low scalable
- low Performance
- Hard built and maintained
- Developed and deployed is time consuming

ABC will acquire the data from all these storage systems and plan to use it for analytics and prediction of the firm's growth and sales prospect. In the first phase ABC has to create the servlets to Add a product and Display product details. Add servlet dependencies required to compile the servlets. Create an HTML page which will be used to add a product. Team is using git to keep all the source code. ABC has decided to use DevOps model and Once source code is available in Github, we need to integrate it with Jenkins and provide continuous build generation for continuous Delivery, integrate with Ansible and Kubernetes for deployment. Use docker hub to pull and push images between ansible and Kubernetes.

The Goal of the Project

Implement CI/CD such that *ABC* Company can be

1. Highly available
2. Highly scalable
3. Highly Performance
4. Easily built and maintained
5. Developed and deployed quickly

System Requirement

Hardware Required

- Student Local Machine (PC → MacBook Air M2)
- Virtual Machine (Edureka Cloud Lab)

AUTHORED BY STUDENT: Christian Onyeukwu (Canada)

Software Required :

- Java (for source code implementation)
- maven (build library for maven based project → java)
- Git (version control system)
- jenkins (ci/cd automation tool)
- Docker (containerization tool).
- Ansible (configuration management tool → YAML scripting or playbooks)
- Kubernetes (automated management, scaling, running containers across clusters machines)
- Grafana (visualization and monitoring tool to create an interactive dashboard from source like Prometheus)
- Prometheus (monitoring and alerting tool for storing, collecting and alerting on the system metrics)

Problem Statement/Tasks

We need to develop a CICD pipeline to automate the software development, testing, package, deploy reducing the time to market of app and ensuring good quality service is experienced by end users. In this project we need to

1. Push the code to out github repository
2. Create a continuous integration pipeline using Jenkins to compile,test and package the code present in git hub
3. Write docker file to push the war file to tomcat server
4. Integrate docker with Ansible and write playbook
5. Deploy artifacts to Kubernetes cluster
6. Monitor resources using grafana

Solution Report

Task 1

Step 1: Retrieve source file

The zip file of the source file for the project was downloaded from my classroom to my local machine, unzipped, and moved to the project folder I created on my machine.

Step 2:Prepare source file in local machine

Opened the file from IntelliJ IDE as a Maven-based project, rebuild with maven command (*mvn clean install*), installed apache tomcat in my local machine to view the webapp locally at. See image
→ *http://localhost:8080/ABCTechnologies-1.0/* (optinal)

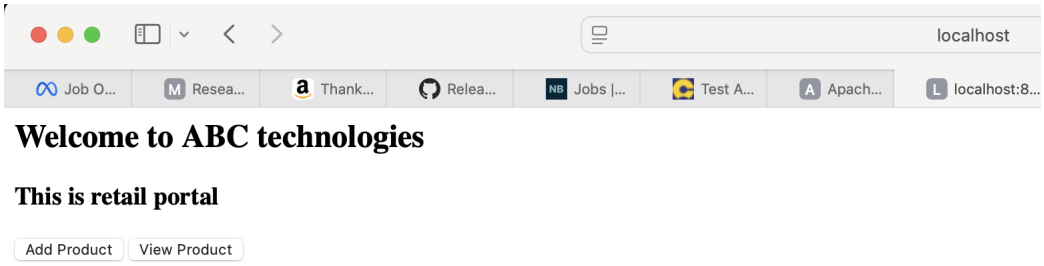

Figure 1: ABC Retail Data Entry & View webpage snapshot

Step 3:Push source file to VCS or Branch for the DevOps

Initialized a git repository, logged in to my github account and created a new repository for this assignment, and then go back to my local terminal window and add, commit and push all my source code into my github

Step 4:Accessing the Virtual Machine (Edureka Cloud Lab)

Logged in to my Cloud Lab, accessed my virtual desktop via *Master Web Desktop* access link. After that I ran some command line code to confirm that all the required tools such as Java, maven, git, docker, kubernetes, jenkins, ansible, grafana and prometheus are pre-installed in the virtual lab. See GitHub repository link:
→ *https://github.com/christianRibig5/industry-based-devops-proj.git*

Step 5: Building codes using maven commands

Created a folder on desktop in my lab (project-required-file-v2), cloned the file from my github and build with command

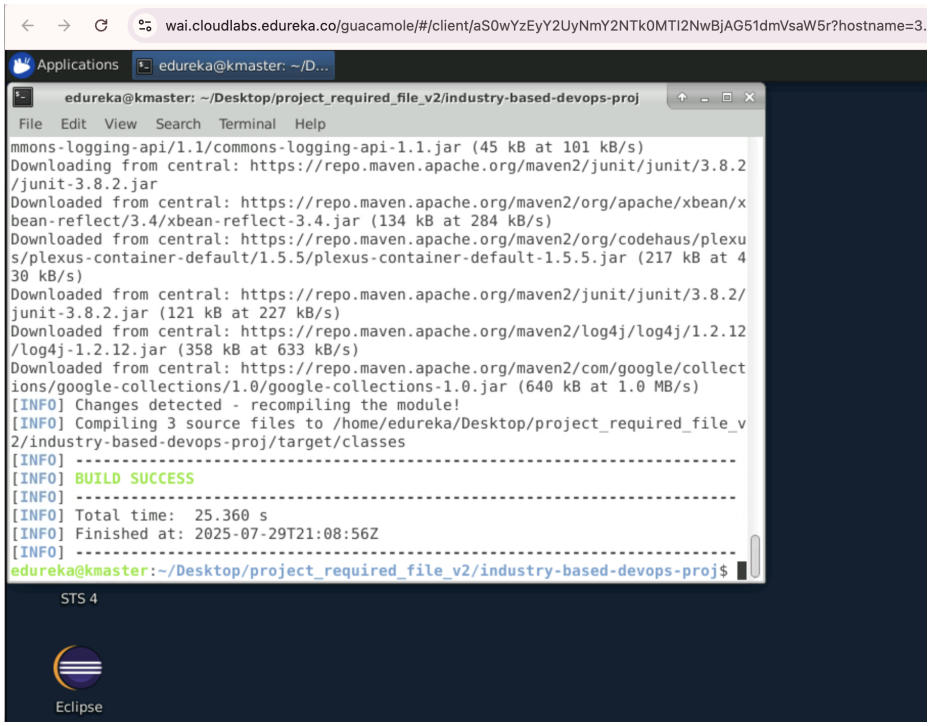

Figure 2: → */opt/maven/bin/mvn compile*:Compile Interface snapshot

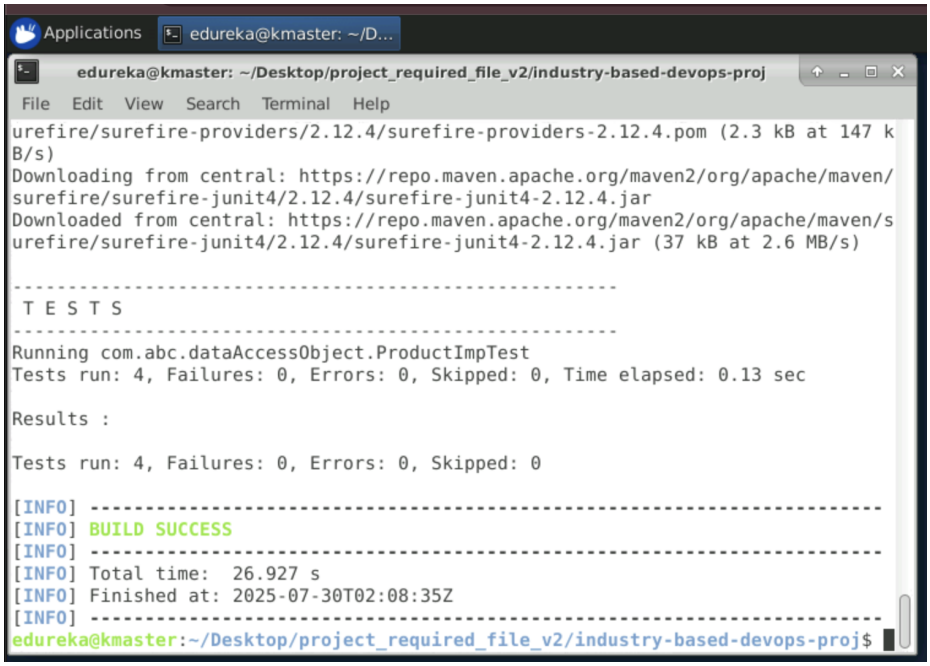

Figure 3: → */opt/maven/bin/mvn test*:Test Interface snapshot

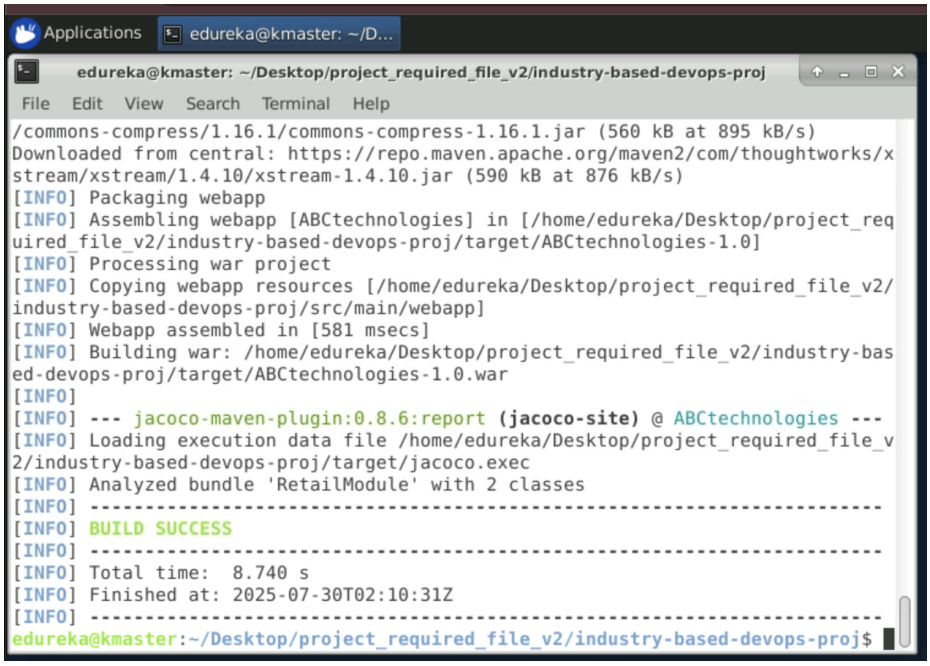

Figure 4: →*/opt/maven/bin/mvn package*:Package Interface snapshot

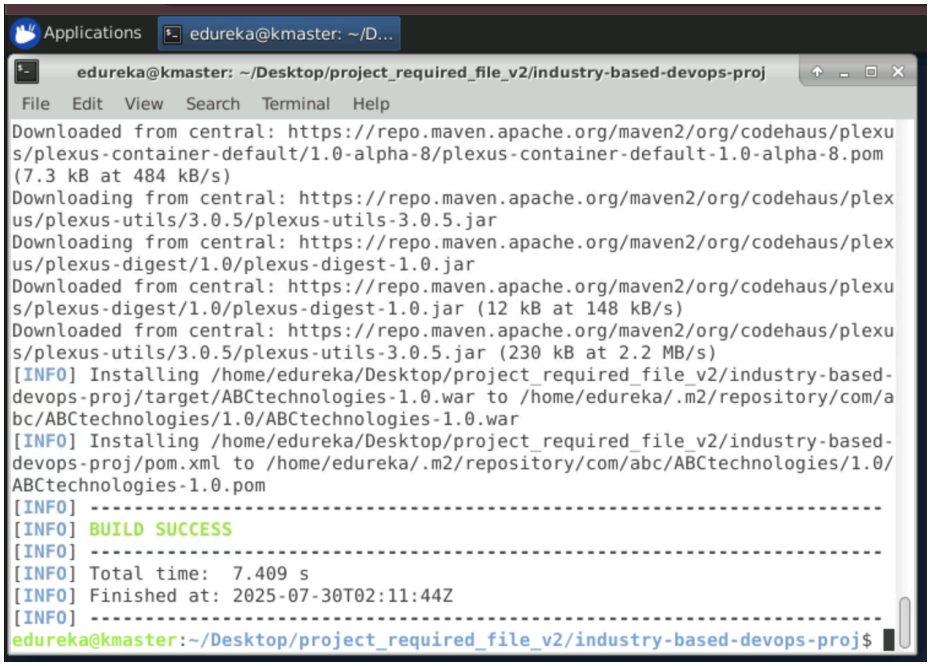

Figure 5: → */opt/maven/bin/mvn clean install*:clean install Interface snapshot

Task 2.1 Setting Jenkins CI/CD pipeline

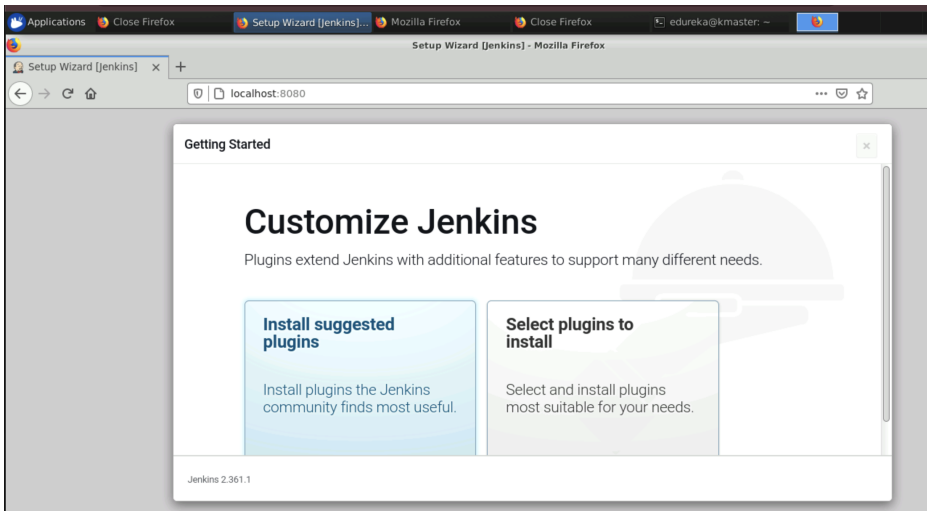

Figure 6: → *http:localhost:8080/login*:jenkins Interface snapshot

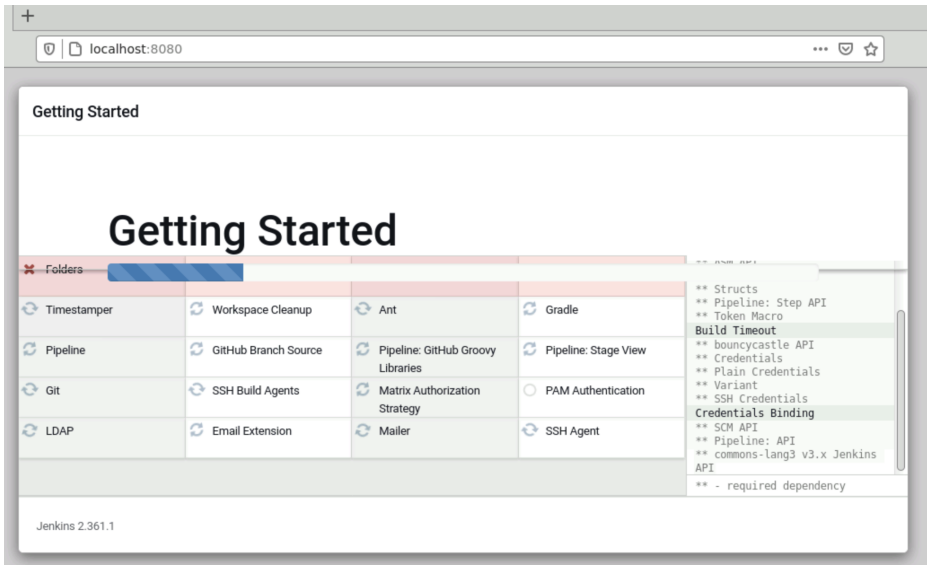

Figure 7: → Selected plugins installation snapshot

Exception:

The plugin installation stopped and I made extra effort to fix the plugin issue but no success except the jenkins version is updated to the latest:

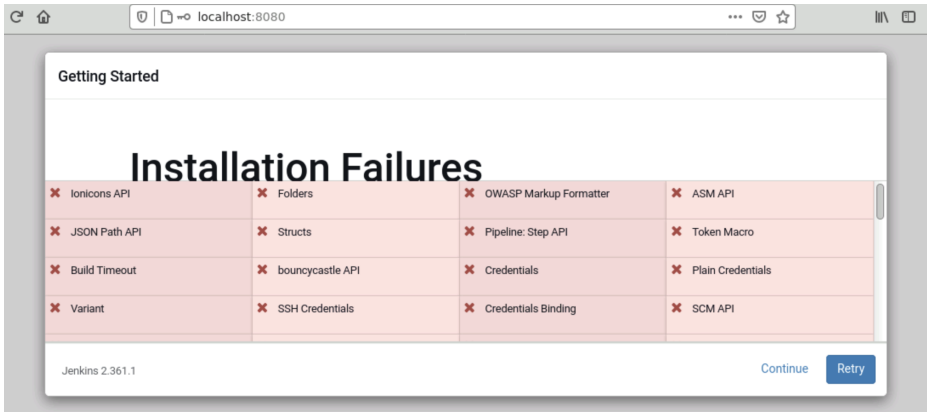

Figure 8: → plugin Installation Error

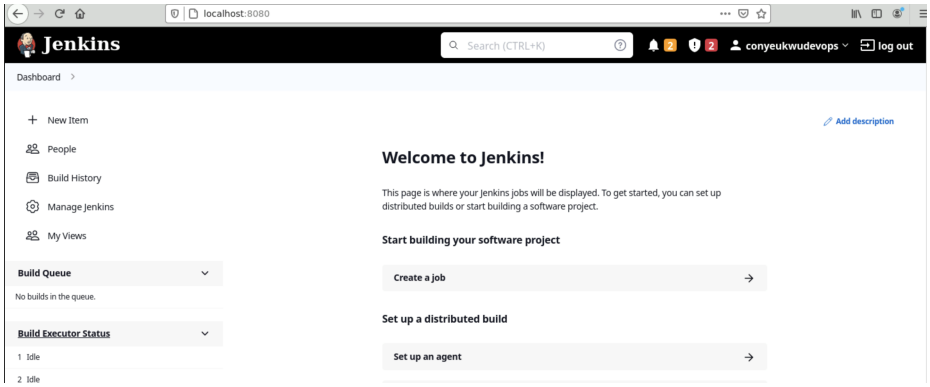

Figure 9: → Welcome Jenkins Interface snapshot

Exception handlled

Resolved my exception by moving my devops vitual platform to Amazon EC2 Cloud t2.large instance at cost 0.126 dollar/hour. And after success installation i continued to set global tool configuration:

Figure 10: → Git Installation configuration Interface snapshot

Task 2.2 Create build pipeline containing 3 jobs

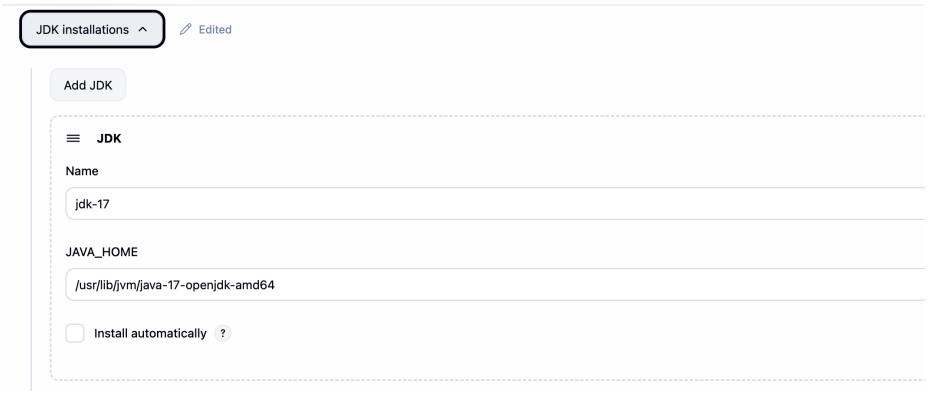

Figure 11: → JDK Installation configuration Interface snapshot

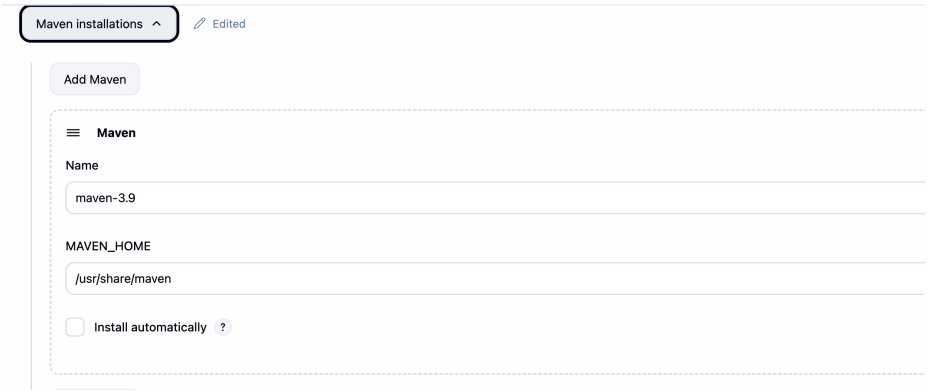

Figure 12: → Maven Installation configuration Interface snapshot

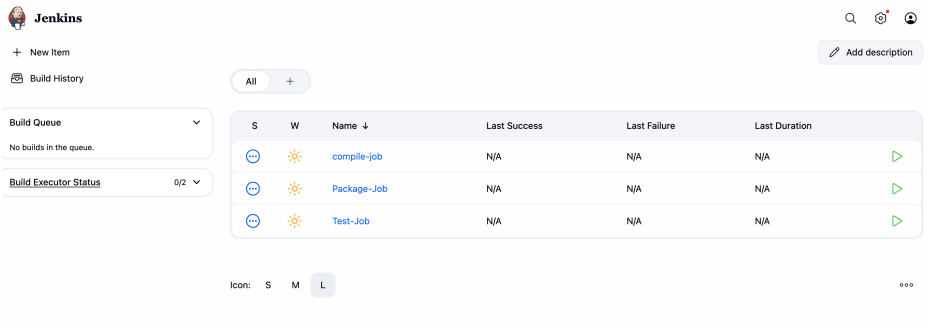

Figure 13: → 3 Jobs schedules Interface snapshot

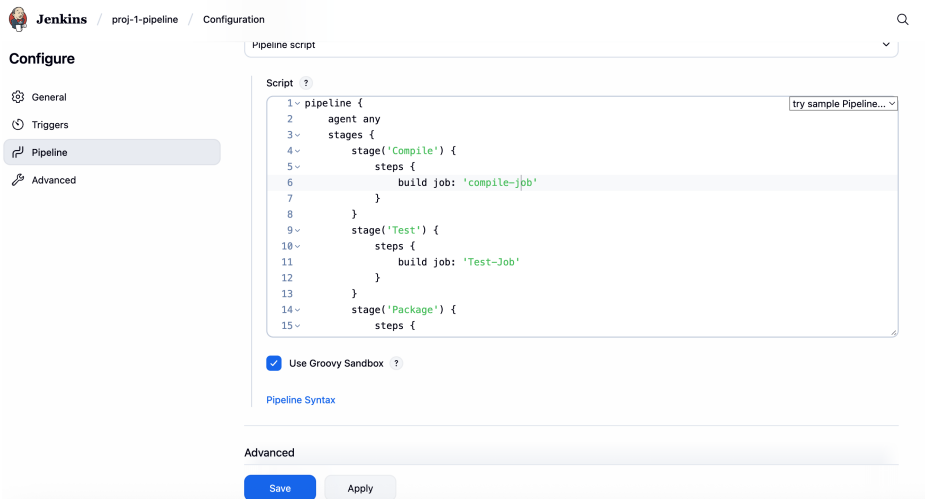

Figure 14: → Pipeline Code

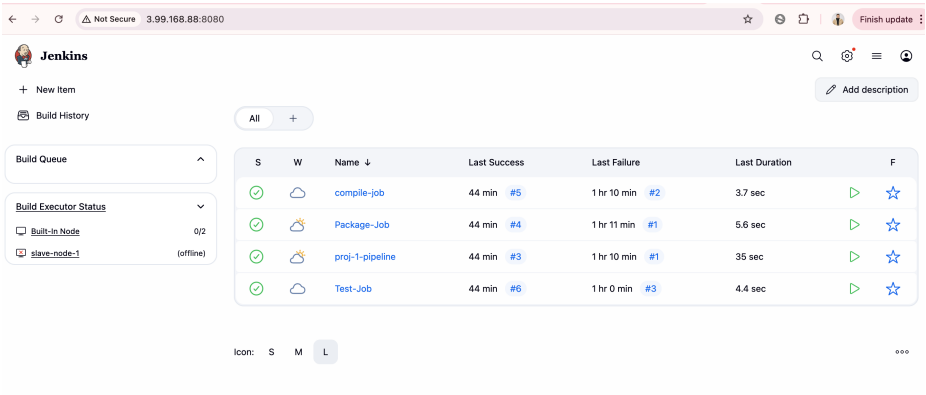

Figure 15: → Execution Interface

Task 2.3: Set up slave node to distribute the tasks in pipeline

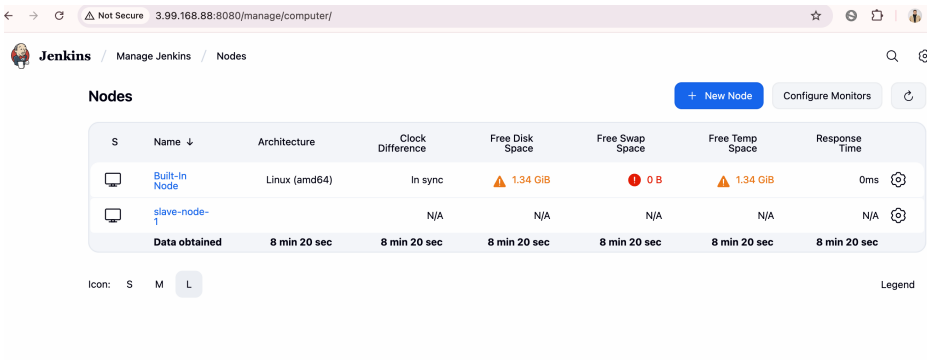

Figure 16: → Slave Node

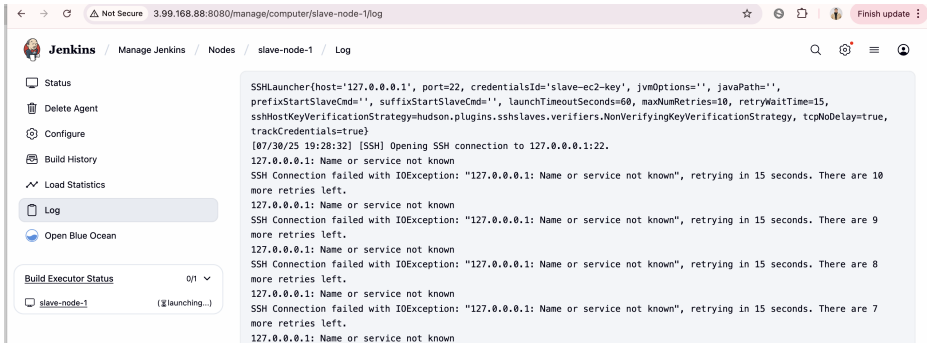

Figure 17: → Lunching Slave Node

Task 3: Write Dockerfile to create Image and container on Docker host

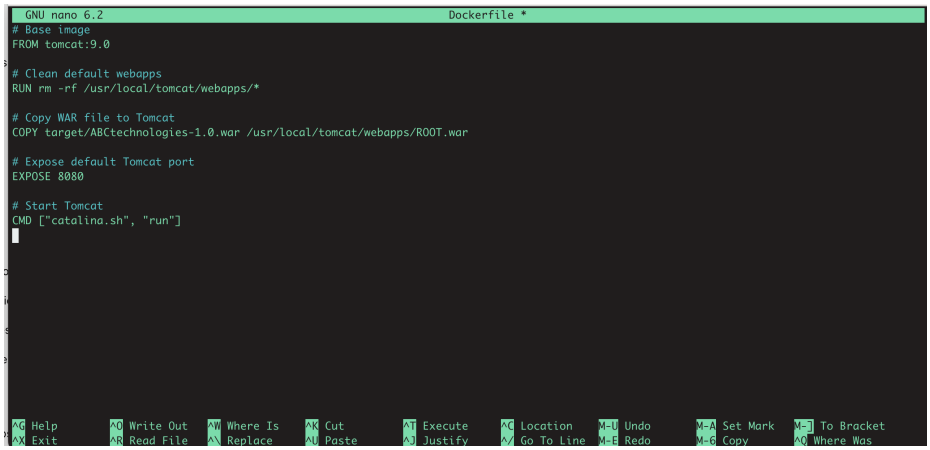

Figure 18: → Docker File Script

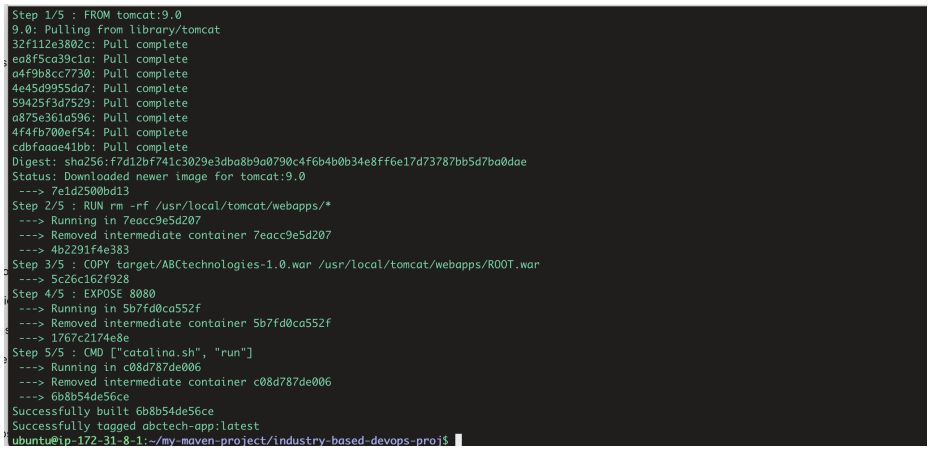

Figure 19: → Docker Script Execution

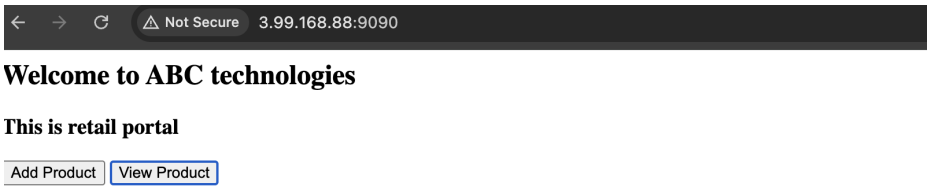

Figure 20: → Image running on Docker container snapshot

Task 4:Integrate Docker host with Ansible

Figure 21: → 1: Deployment artifact snapshot

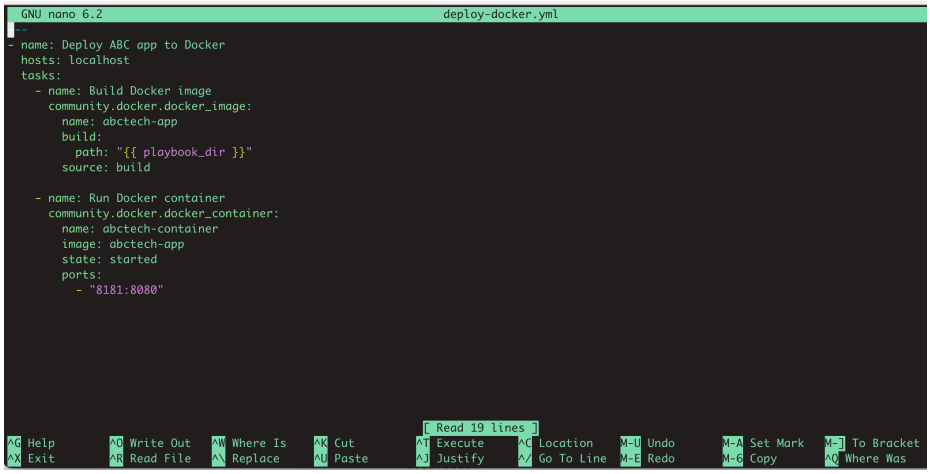

Figure 22: → 2: Ansible Playbook Script snapshot

Figure 23: → 3: pod kube8s playbook snapshot

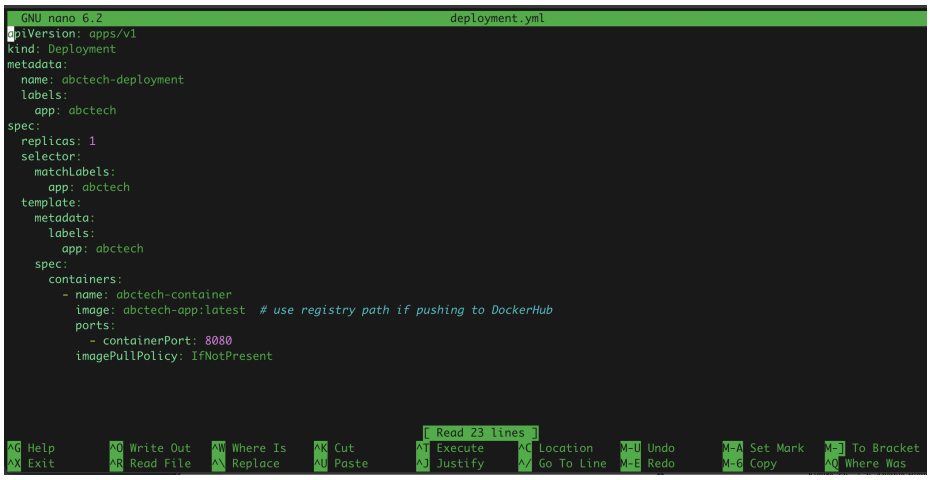

Figure 24: → 3: Kubernetes playbook script snapshot

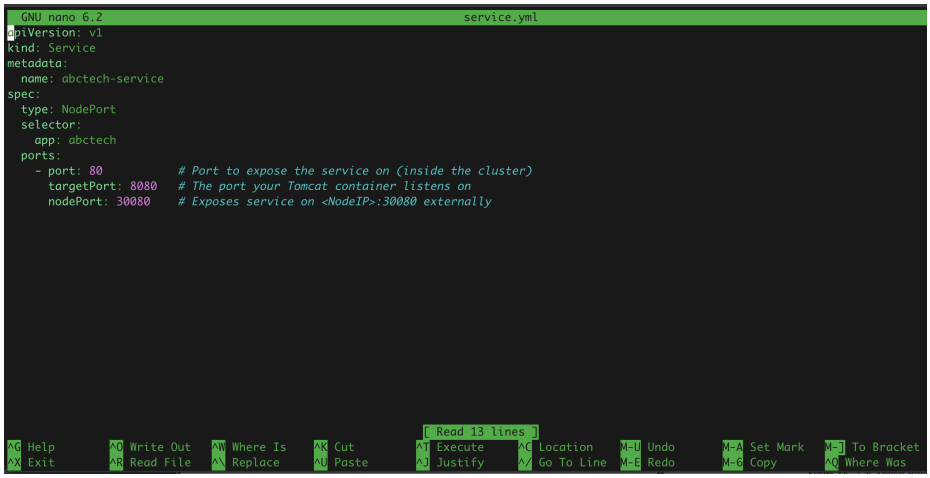

Figure 25: → 3: Service Playbook Script snapshot

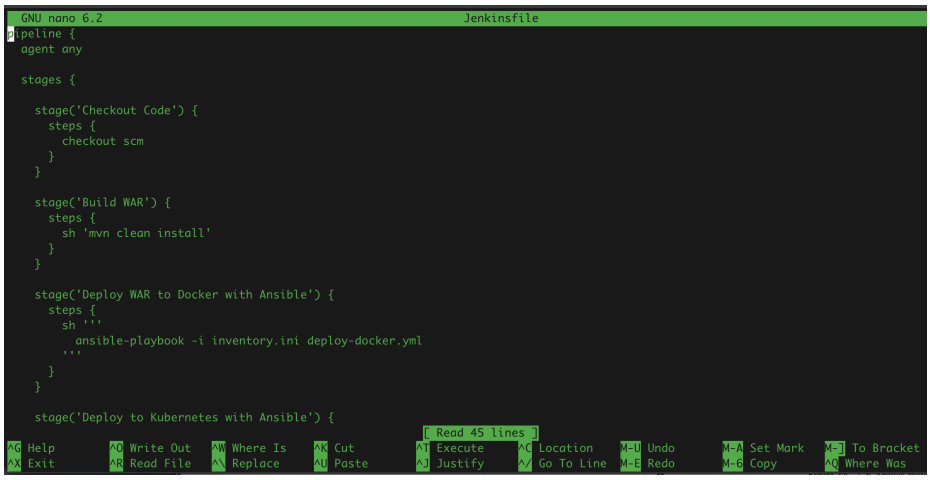

Figure 26: → 4: Pipeline Script snapshot

Task 5: Using Prometheus Monitor and Grafana Dashbord for Metrics Monitor

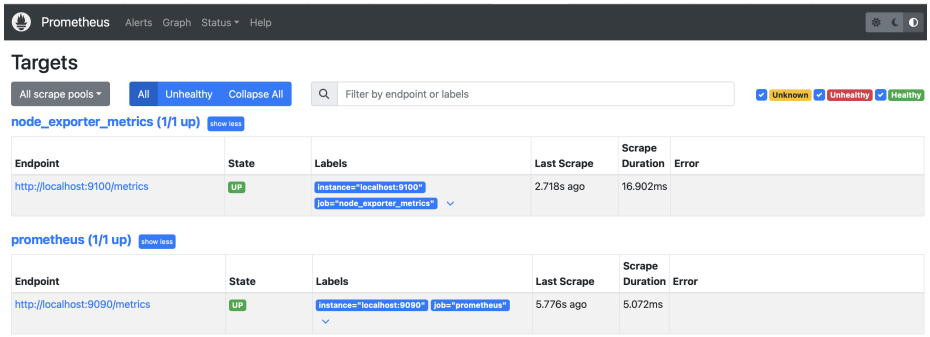

Figure 27: → 1: Prometheus Dashboard snapshot

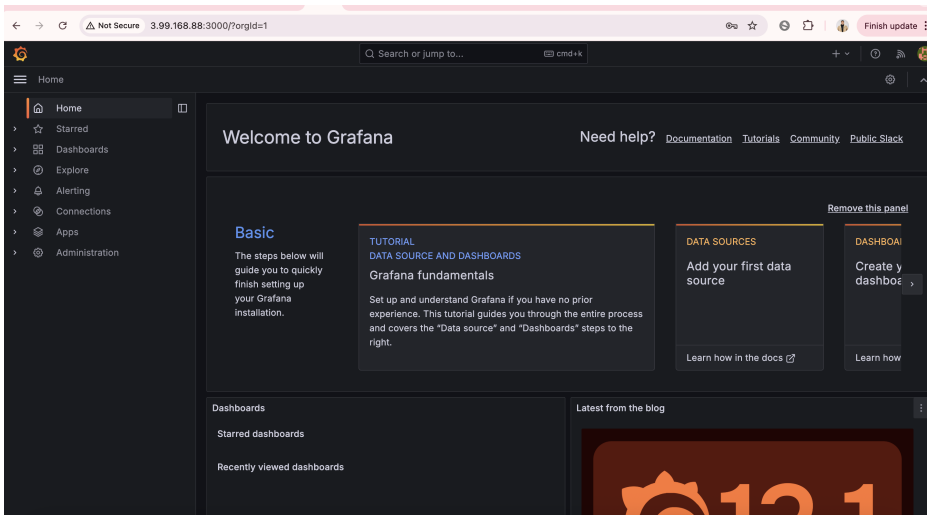

Figure 28: → 2: Grafana Dashboard snapshot

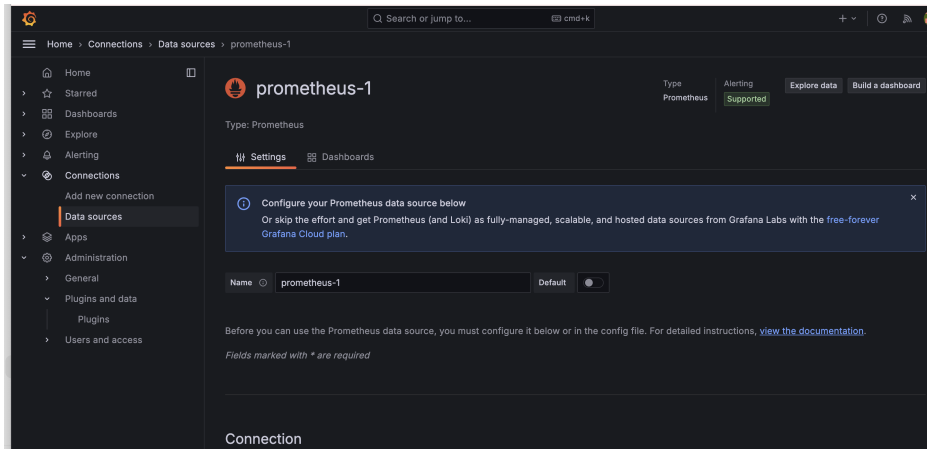

Figure 29: → 3: Added Prometheus as Data Source snapshot

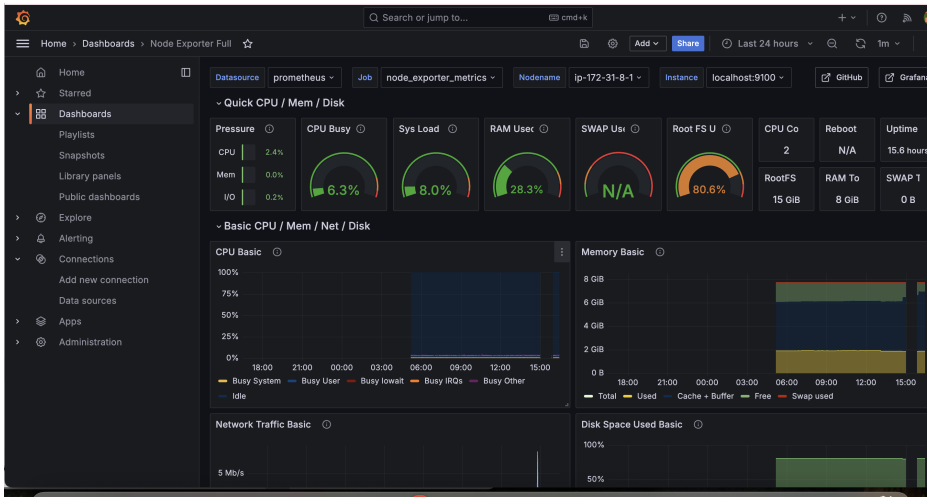

Figure 30: → 4: Monitoring Dashboard snapshot