

# Cloud Computing - Mini Project Report

## **PROJECT 6 - Building an E-commerce Microservices Application on Cloud using Docker, Kubernetes, Jenkins, and Git**

April 2023

Submitted By: **Batch -9**

### **Team Members**

<b>Name</b>	<b>SRN</b>
K N Pragathy Reddy	PES2UG20CS151
Kavya J	PES2UG20CS159
Anjali K	PES2UG20CS160
Manodyna K H	PES2UG20CS187

VI Semester Section : C  
PES University

## **Short Description and Scope of the Project**

The goal of this project is to build an E-commerce microservices application that can be used with Docker, Kubernetes, Jenkins, and Git to deploy it on the cloud. Multiple microservices will make up the application, which will be distributed as Docker containers on a Kubernetes cluster. Git will be utilized for version control, while Jenkins will be used for continuous integration and deployment. The project will also involve testing and debugging the application to ensure that it is functioning correctly.

- Create and put into use the application's microservices architecture.
- For each microservice, create a container using Docker.
- To locally orchestrate the containers, use Kubernetes.
- Automate the deployment process by putting a Jenkins pipeline in place.
- Integrate Jenkins with Git to start the pipeline when the code is changed.

### **Github link**

[https://github.com/pragathy1572/PES2UG20CS151\\_CC-PROJECT.git](https://github.com/pragathy1572/PES2UG20CS151_CC-PROJECT.git)

<https://github.com/Kavyaj554/PES2UG20CS159-CC-PROJECT.git>

## Methodology

- **Design the Microservices Architecture**
- **Develop Microservices**
- **Containerized Microservices using Docker**
- **Orchestrate Microservices using Kubernetes**
- **Implement Continuous Integration and Deployment using Jenkins**
- **Version Control using Git**

This project will be broken down into small, manageable tasks that can be completed within short timeframes. The following steps will be followed:

**Planning and Design:** The first step will involve planning and designing the microservices architecture for the e-commerce application. This will involve identifying the different microservices, their functionalities, and how they will interact with each other.

**Development:** Once the architecture is designed, the next step will be to develop the microservices and create Docker containers for each of them. This will involve writing code, creating Docker images, and testing the containers to ensure that they are functioning correctly.

**Integration:** The next step will be to integrate the microservices using Kubernetes to create a working application. The microservices will be deployed as containers on a Kubernetes cluster, and Kubernetes will be used to orchestrate the containers and ensure that they are working together.

**Continuous Integration and Deployment:** Jenkins will be used to automate the deployment process, and a pipeline will be implemented to ensure continuous integration and deployment. This will involve setting up Jenkins, creating a pipeline, and integrating it with Git to trigger the pipeline on code changes.

## Testing

Testing is an essential part of the development process for any software project, including an e-commerce microservices application developed using Docker, Kubernetes, Jenkins, and Git. **Testing and Debugging:** Once the application is deployed, it will be tested to ensure that it is functioning correctly. Any bugs or issues will be identified and fixed. The following types of testing should be performed:

**Unit Testing:** Unit tests should be performed on each microservice to ensure that it is functioning correctly. This will involve testing each microservice in isolation to verify that it is performing as expected.

**Integration Testing:** Once the microservices are developed and tested, they should be integrated to ensure that they are working together as expected. Integration testing will involve testing the entire application to verify that it is functioning correctly.

**System Testing:** System testing should be performed to ensure that the application meets the requirements and is functioning correctly in the production environment. This will involve testing the entire system, including the application and its infrastructure, to verify that it meets the expected performance criteria.

**Performance Testing:** Performance testing should be performed to ensure that the application can handle the expected workload. This will involve testing the application under simulated loads to verify that it can handle the expected traffic.

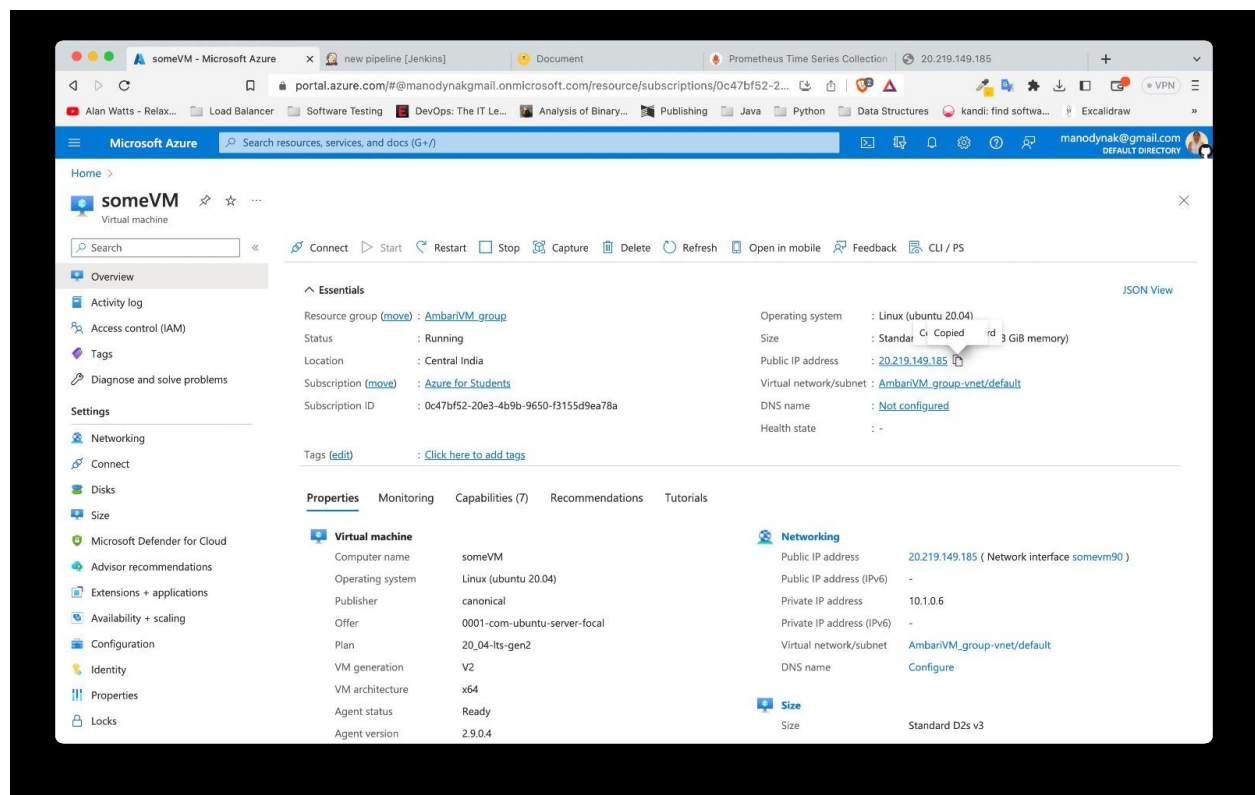
The development team may make sure the application is working properly, satisfies user needs, and is prepared for deployment to the production environment by carrying out several kinds of testing.

## Results and Conclusions

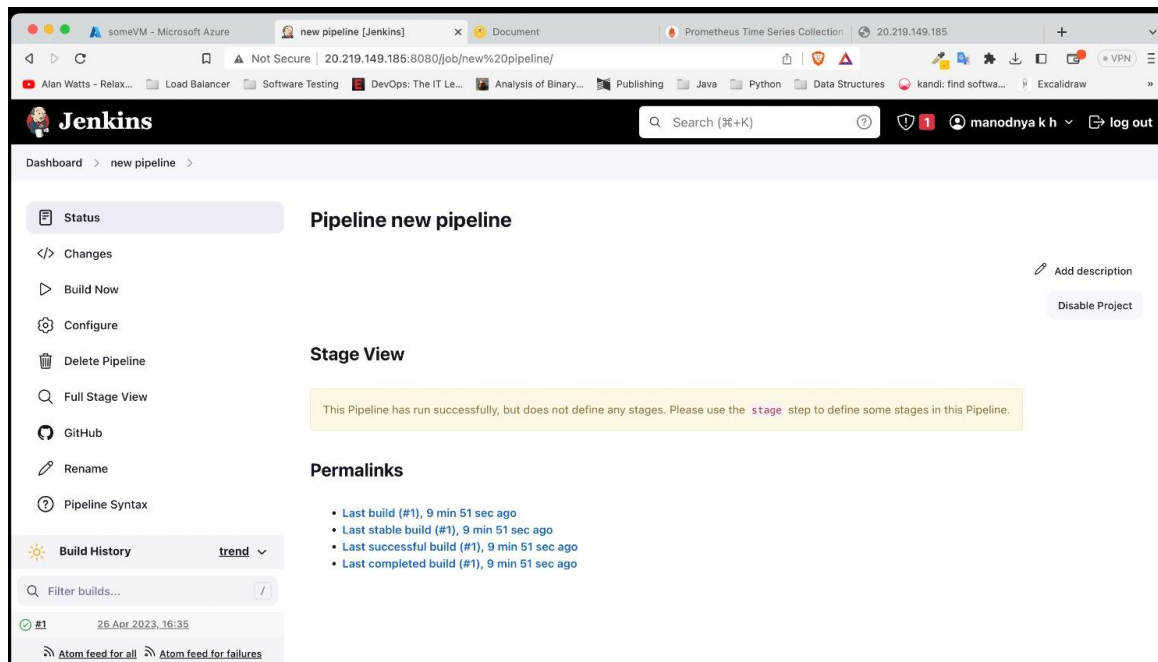
The conclusion of this project will be that an e-commerce microservices application can be developed and deployed on the cloud using Docker, Kubernetes, Jenkins, and Git. The use of microservices architecture provides greater flexibility and scalability, while the use of Docker and Kubernetes simplifies deployment and management. Continuous integration and deployment using Jenkins and Git ensures that the application is always up-to-date and functioning correctly. By following an agile methodology and performing rigorous testing, the development team can ensure that the application is of high quality and meets the requirements.

## Screenshots Of the Project Output

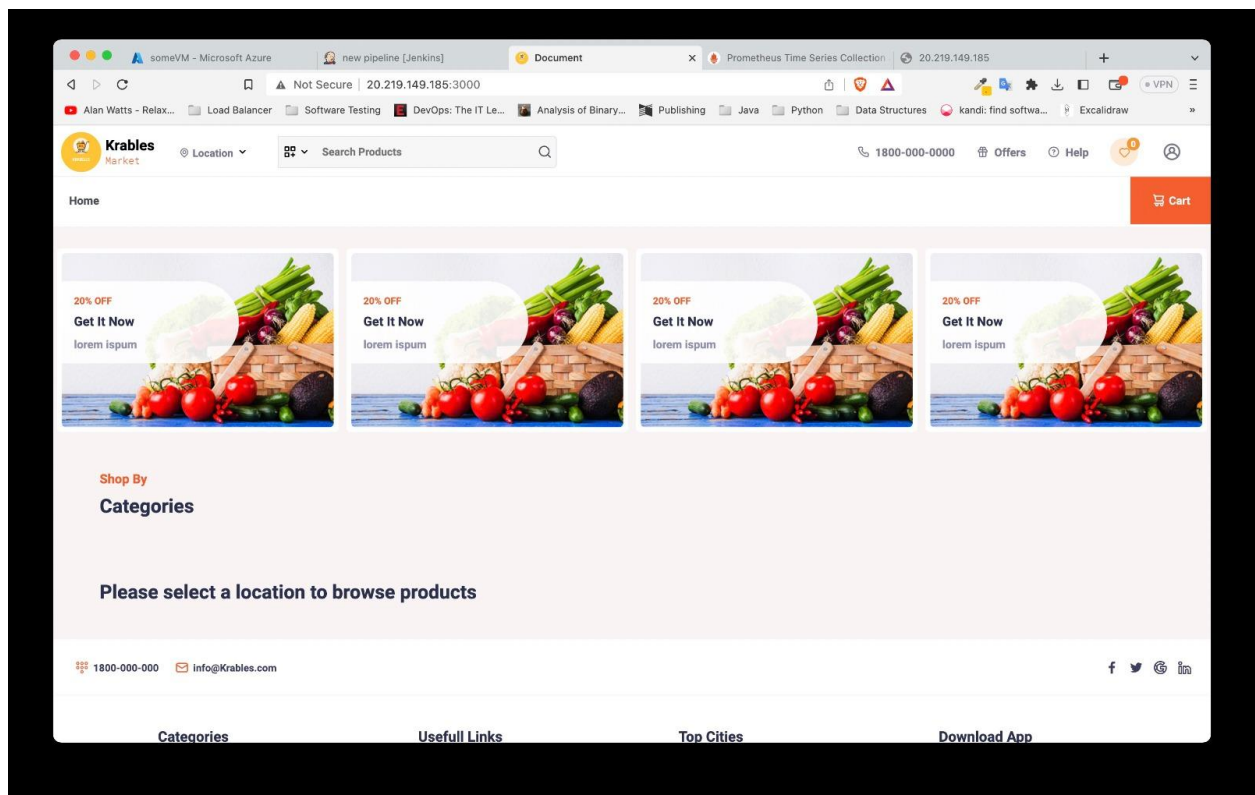
Setting up Microsoft Azure:



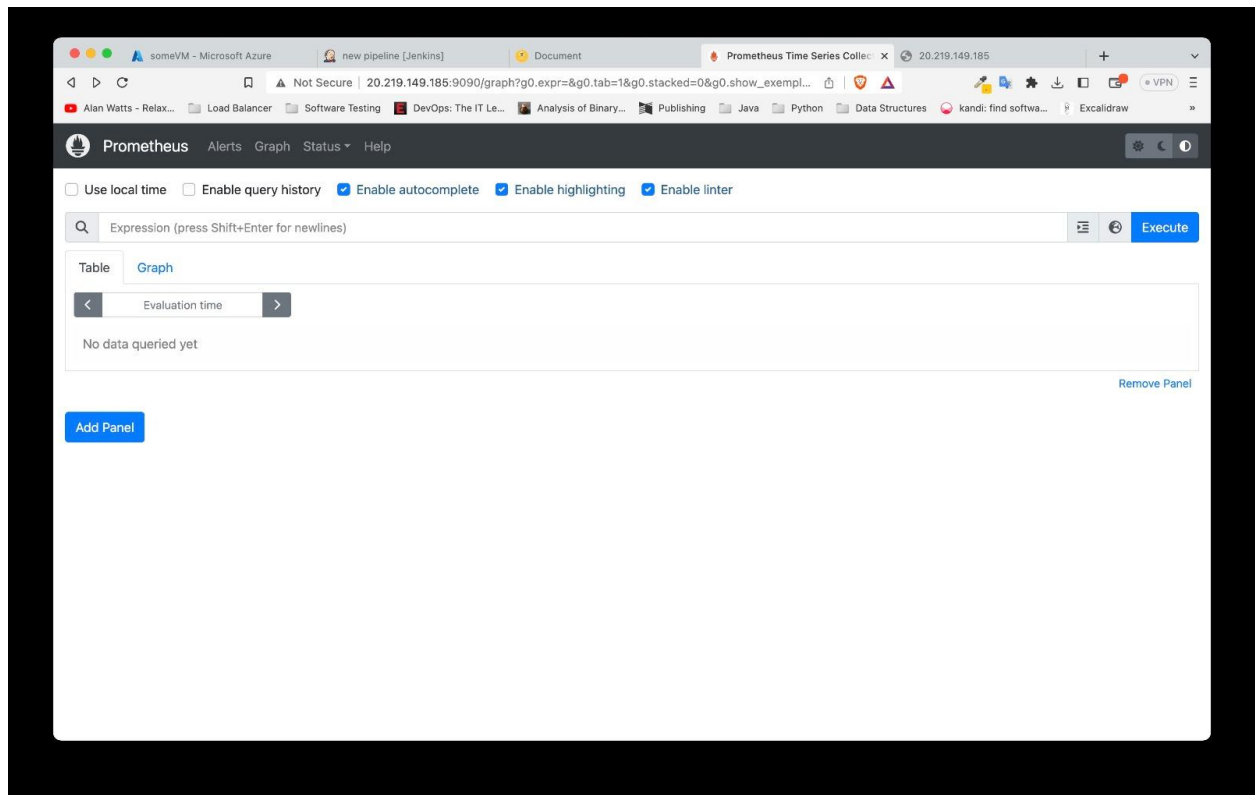
## Setting up Jenkins:



## E commerce website output:



## Prometheus output:



## Running docker ps:

