***Github Repository Links***

|  |
| --- |
| **https://github.com/Ram495-ctrl/Container-Orchestration.git** |

**Description**

In this assignment, you are tasked with creating Kubernetes deployment files and a HELM chart for a MERN (MongoDB, Express.js, React.js, Node.js) based application. The provided project consists of separate frontend (FE) and backend (BE) components.

You will need to:

Develop Kubernetes deployment files for both frontend and backend components, ensuring seamless deployment and scalability.

Create a HELM chart to streamline the deployment process, allowing for easy configuration and management.

Write Jenkins Groovy code to automate the build and deployment process, ensuring consistency and efficiency in your CI/CD pipeline.

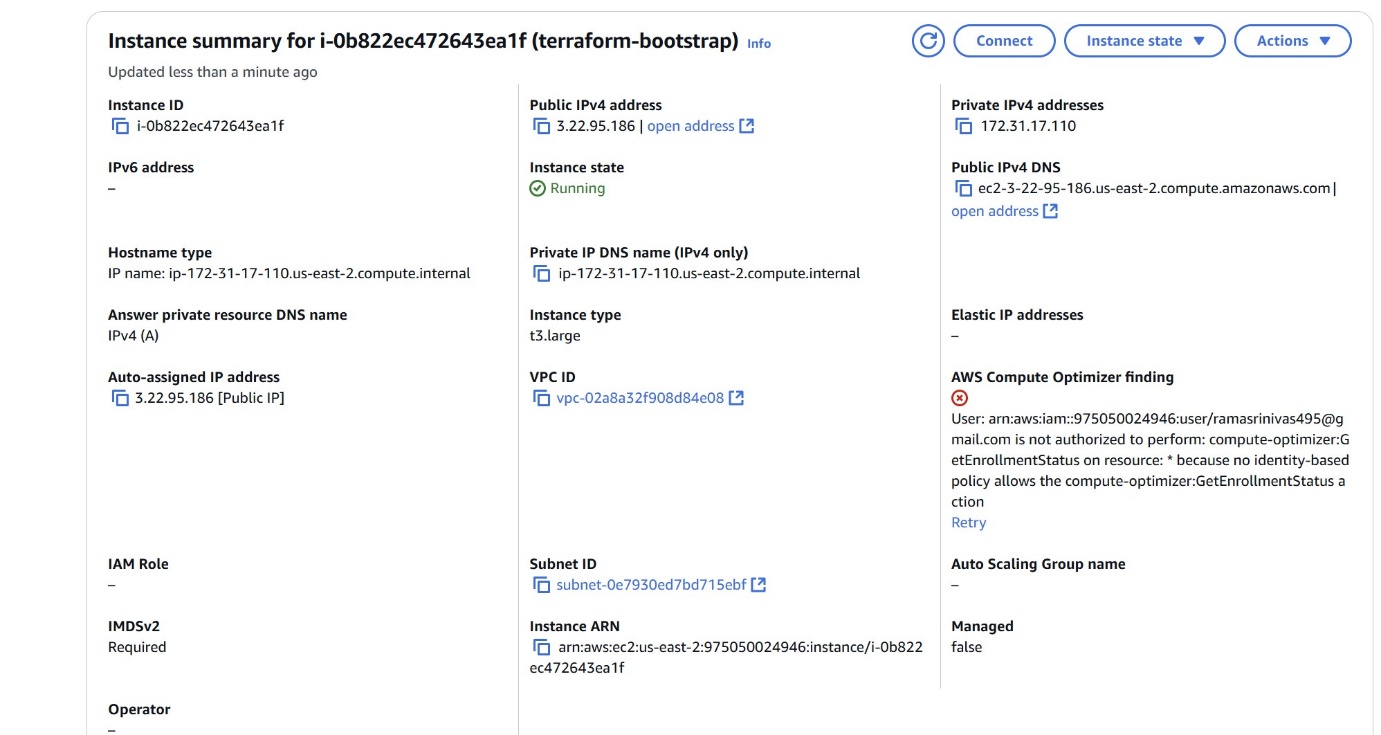
The provided GitHub repositories contain the frontend and backend codebases, which you will use as the basis for your deployment configurations.

* + - <https://github.com/UnpredictablePrashant/learnerReportCS_frontend>
    - <https://github.com/UnpredictablePrashant/learnerReportCS_backend>

Please ensure to document your process thoroughly, including any challenges faced and solutions implemented. Your submission should demonstrate your understanding of Kubernetes deployment, HELM charts, and Jenkins automation in the context of a MERN stack application.

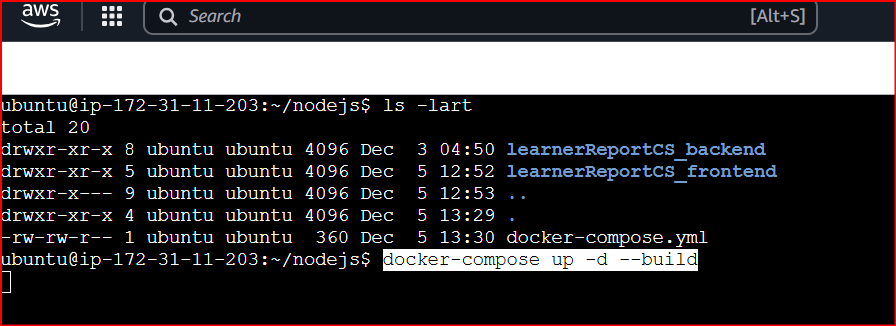
*Step1:*

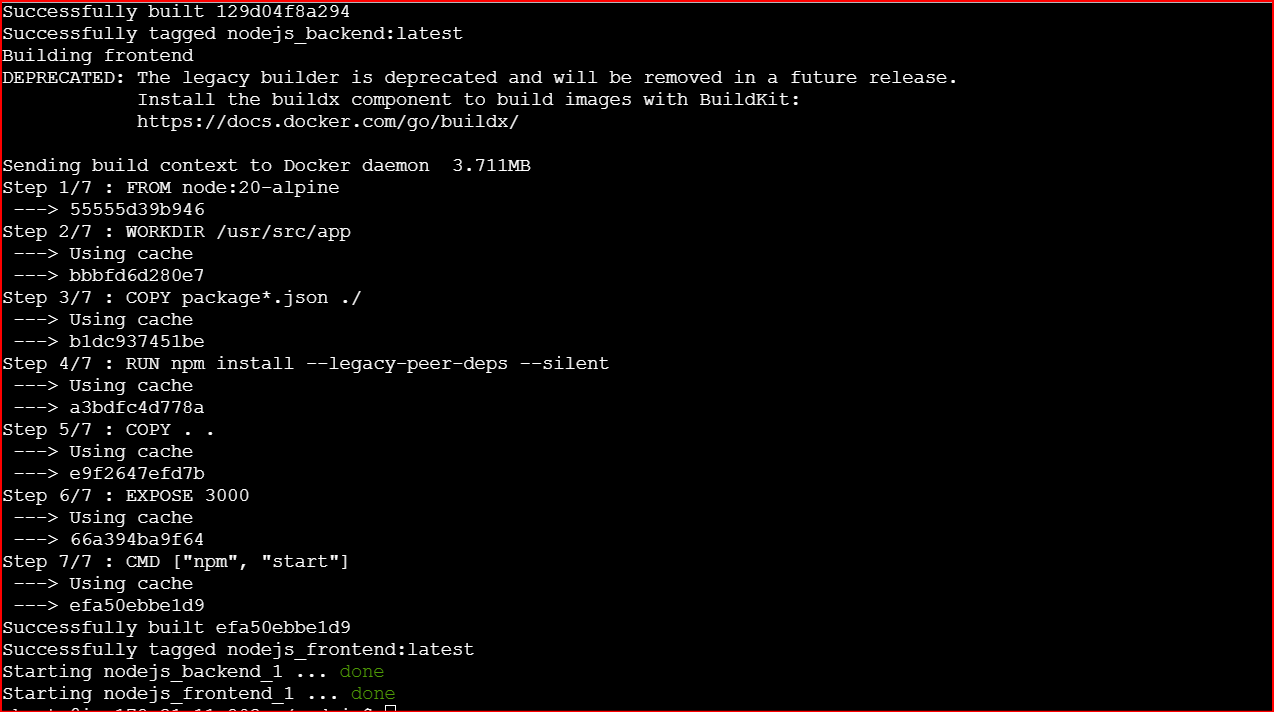
Using Terraform script I created an Ec2 instance where I will install EKSCTL to create the Kubernetes Cluster: This EC2 instance will act like a boot strap instance to proceed further. The Terraform script will automatically download the EKSCTL and will download from the GIT repositories.



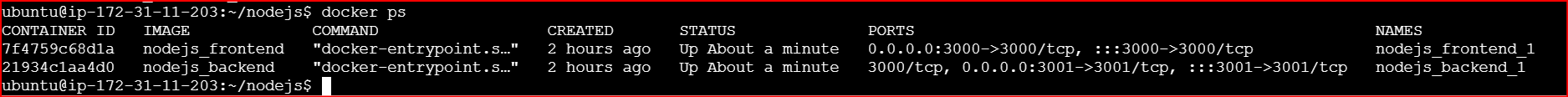
*Step2:*

*Create the required Docker files using Docker compose*

**

**

*Docker containers are successfully running now*

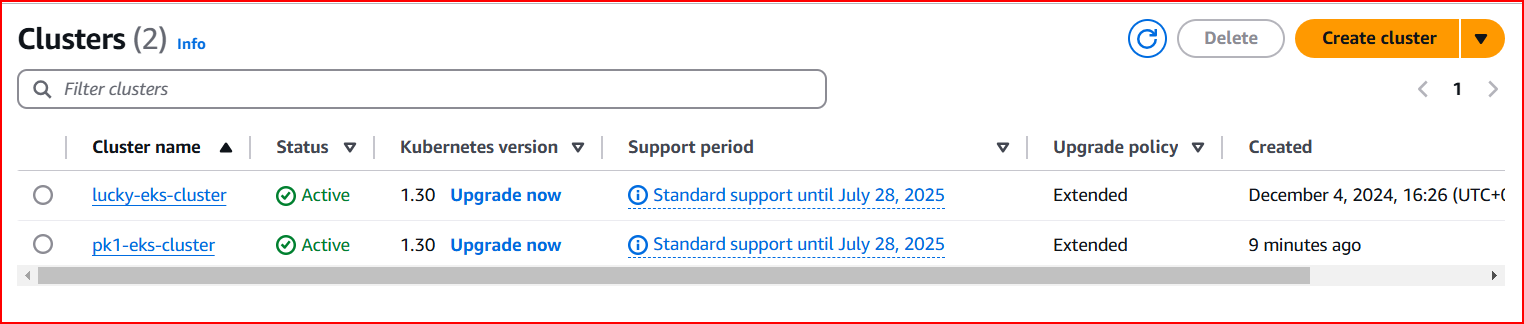
**

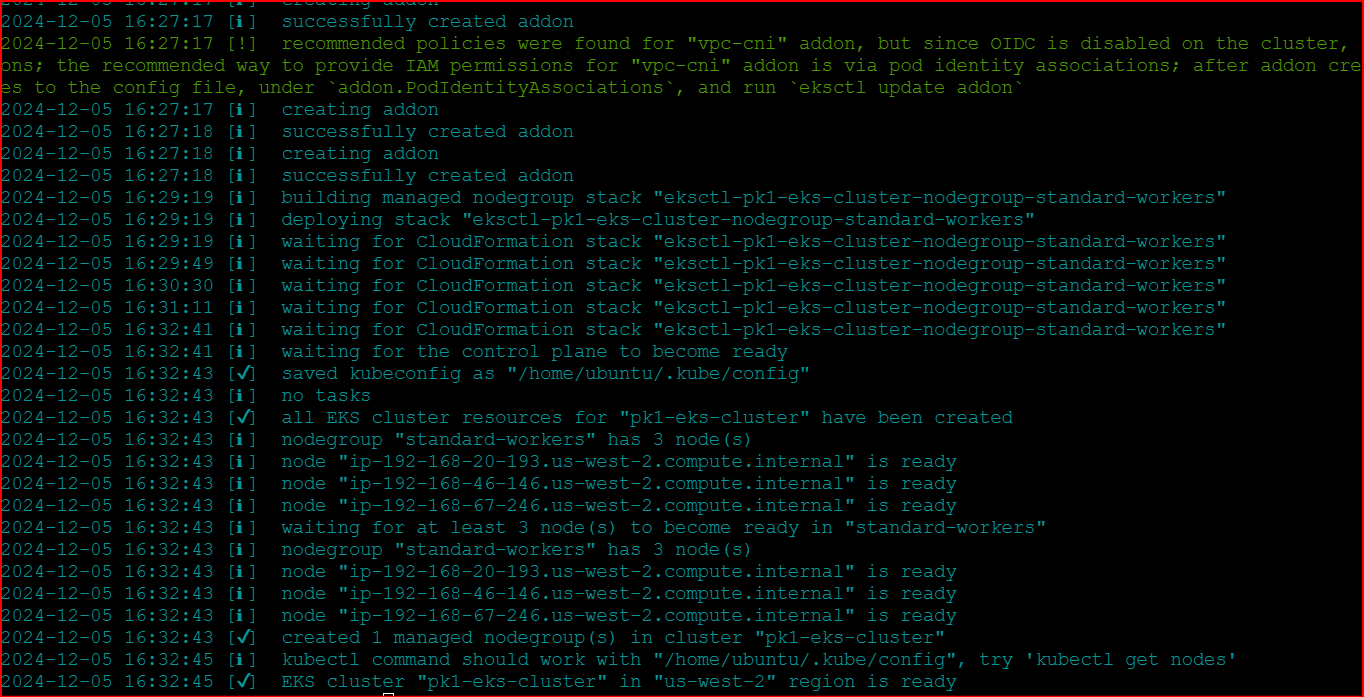
*Step3:*

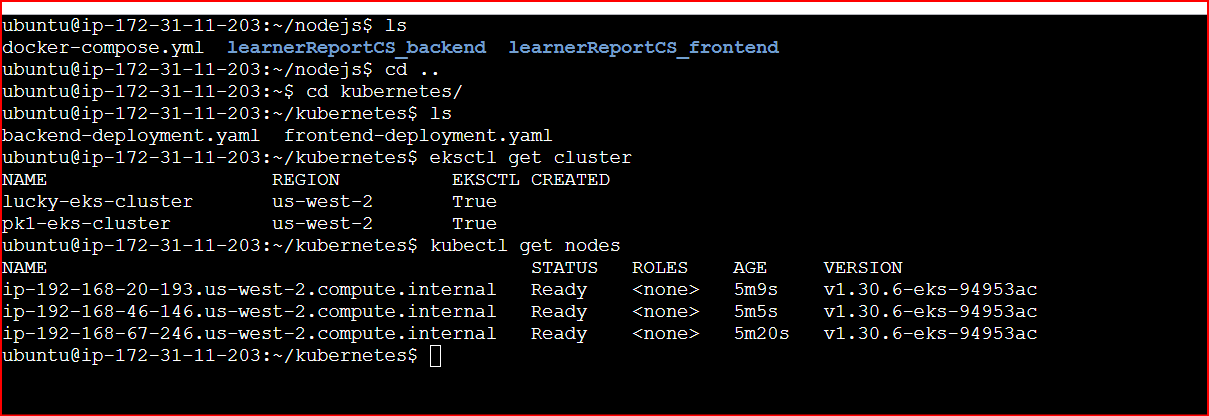
*Create the cluster using EKSCTL – 3 Nodes*

|  |
| --- |
| *eksctl create cluster \*  *--name pk1-eks-cluster \*  *--region us-west-2 \*  *--nodegroup-name standard-workers \*  *--node-type t3.medium \*  *--nodes 3 \*  *--nodes-min 3 \*  *--nodes-max 3 \*  *--managed* |

*Cluster is created with 3 Nodes and in active pk1-eks-cluster*

**

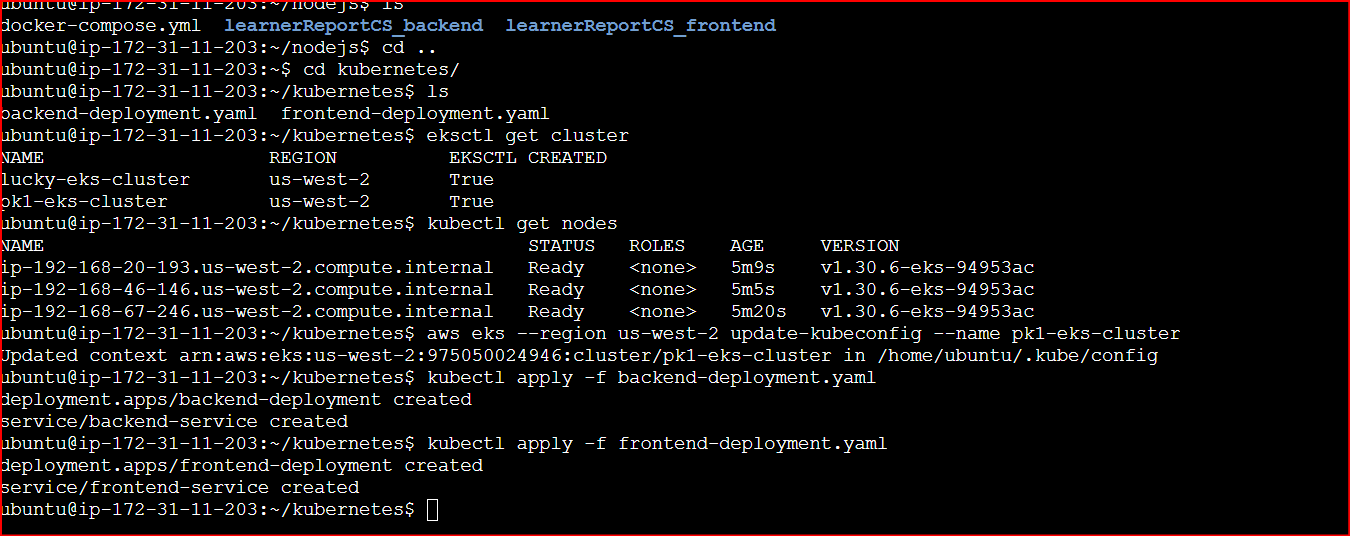
**

**

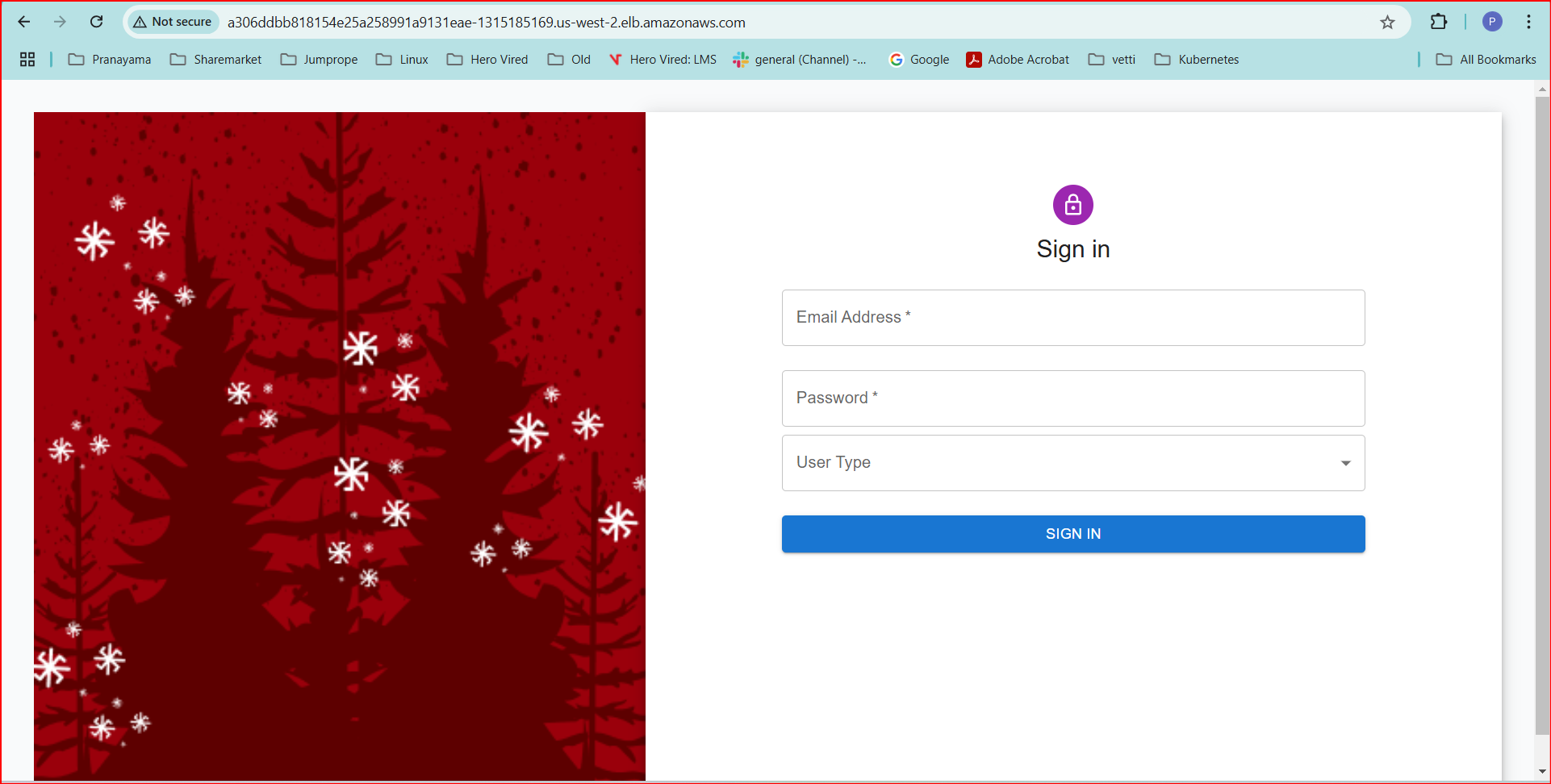
*Step4:*

*Installing the nodejs app into the cluster and trying to run it manually. After creating the deployment files.*

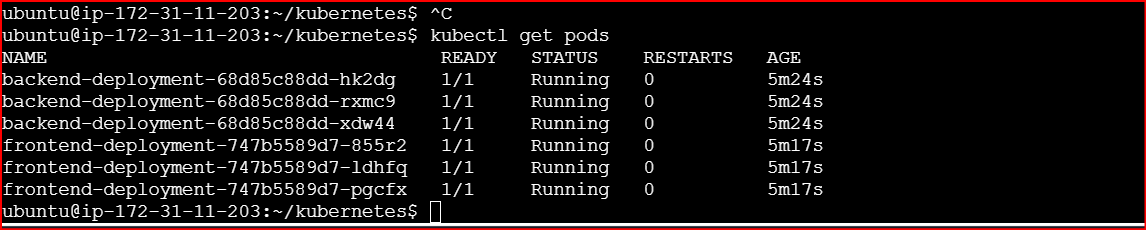
*Frontend and the Backend YAML are deployed successfully*

**

*When I give the external IP of the load balancer front end service the application is displayed as expected*

**

*All the 6 Pods are running as expected*

**

*Step5*

*Using the HELM for the package deployment:*

*mern-helm-chart/*

*├── Chart.yaml*

*├── values.yaml*

*├── templates/*

*├── frontend-deployment.yaml*

*├── backend-deployment.yaml*

*├── mongo-deployment.yaml*