**Deploying a Full Web Application on AWS Using Amplify, Elastic Beanstalk, and EKS**

**Process:**

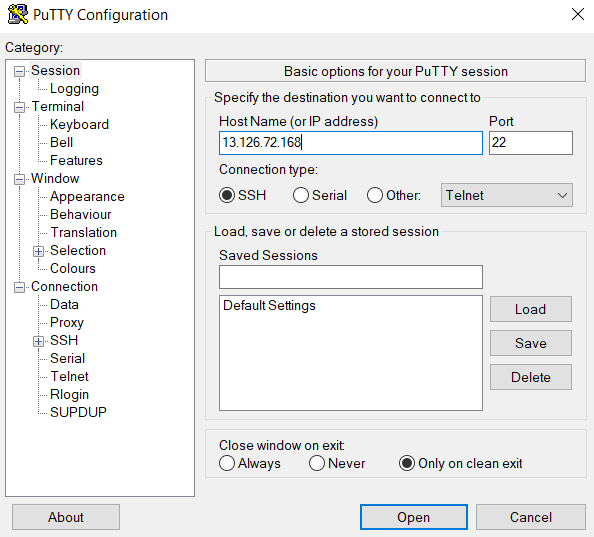
**Set Up AWS Resources**

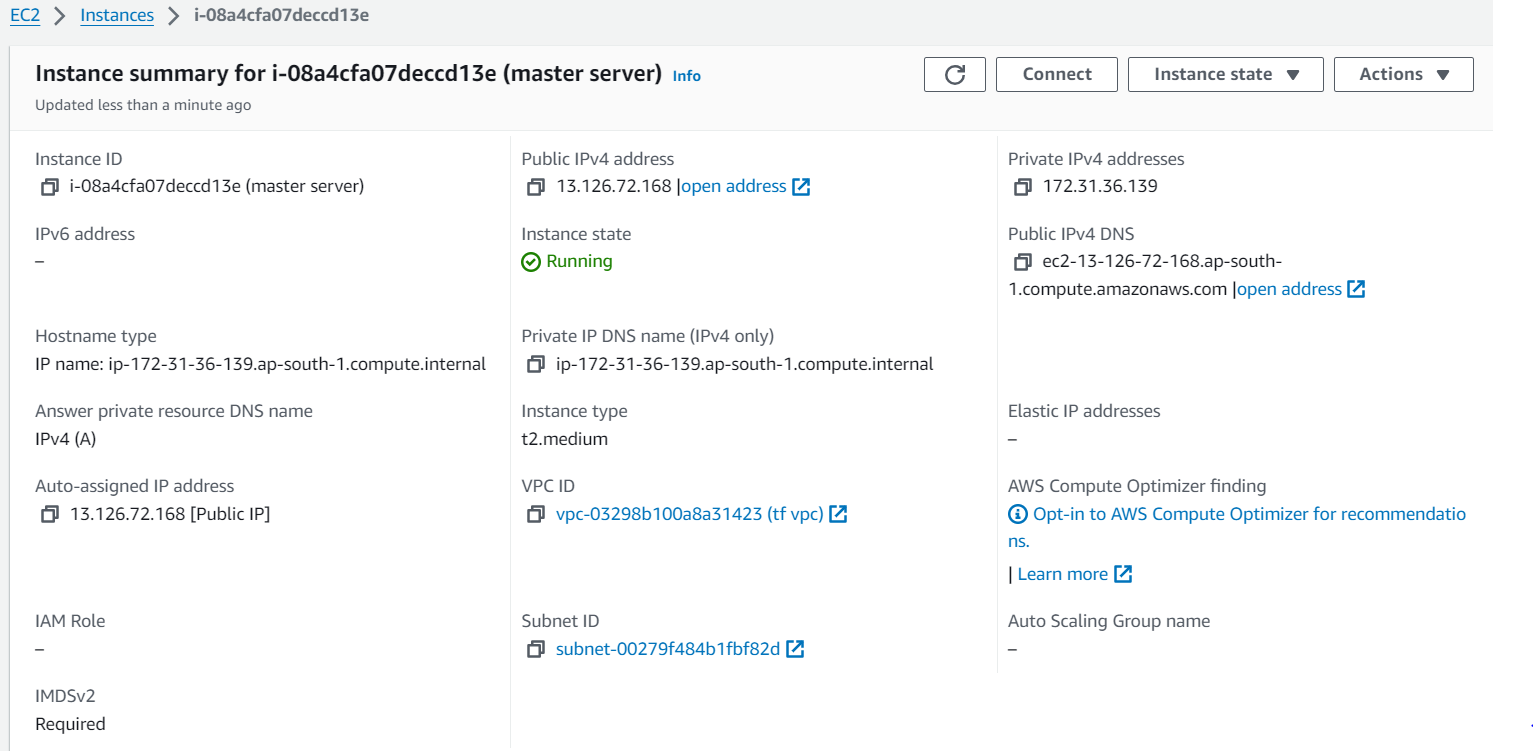
Logged in to AWS Management Console.  
• Launched EC2 Instance: (virtual machine - Ubuntu).

And security groups to allowed SSH (port 22) traffic with instance type - T2.medium and 20Gb disk size and created PPK file.

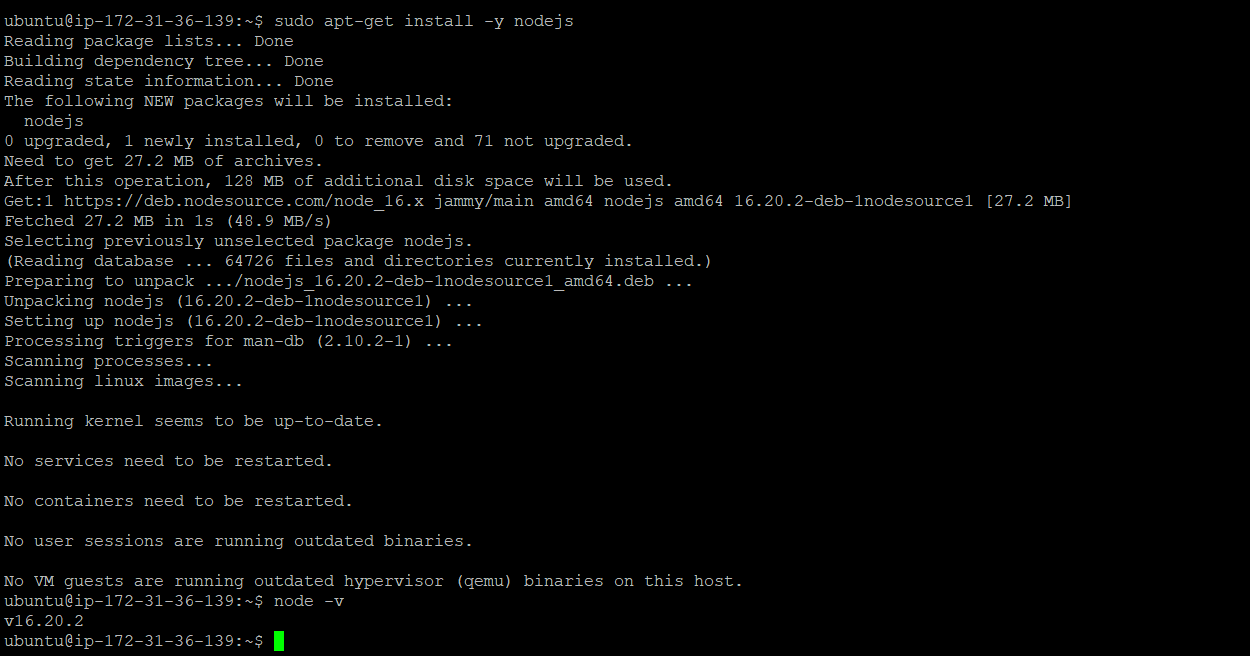
**Connect to Your EC2 Instance**

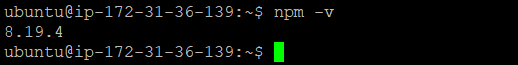
•    Used SSH to connect to EC2 instance using the PPK file via Putty as below.





**Installed Node js and npm on to the machine as below:**

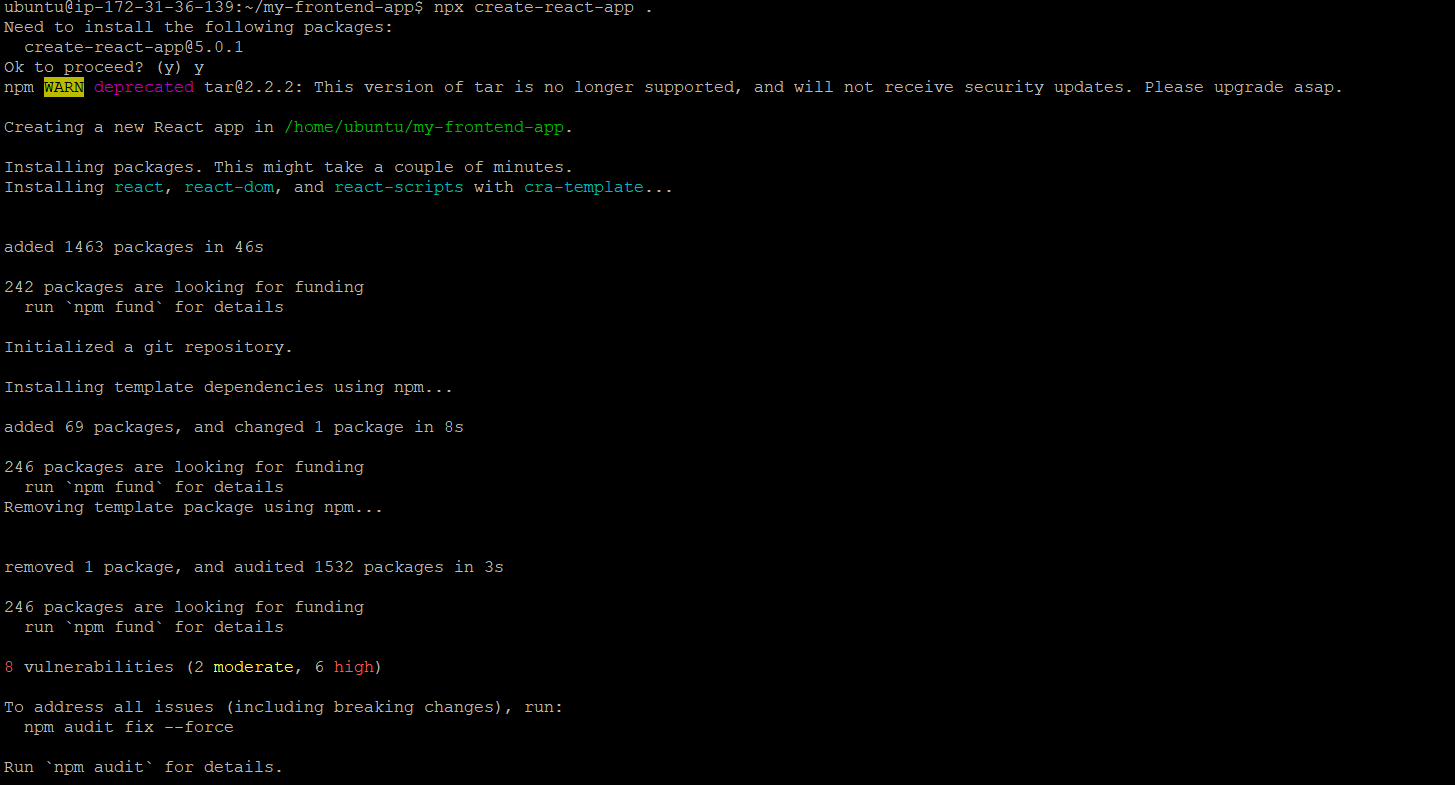




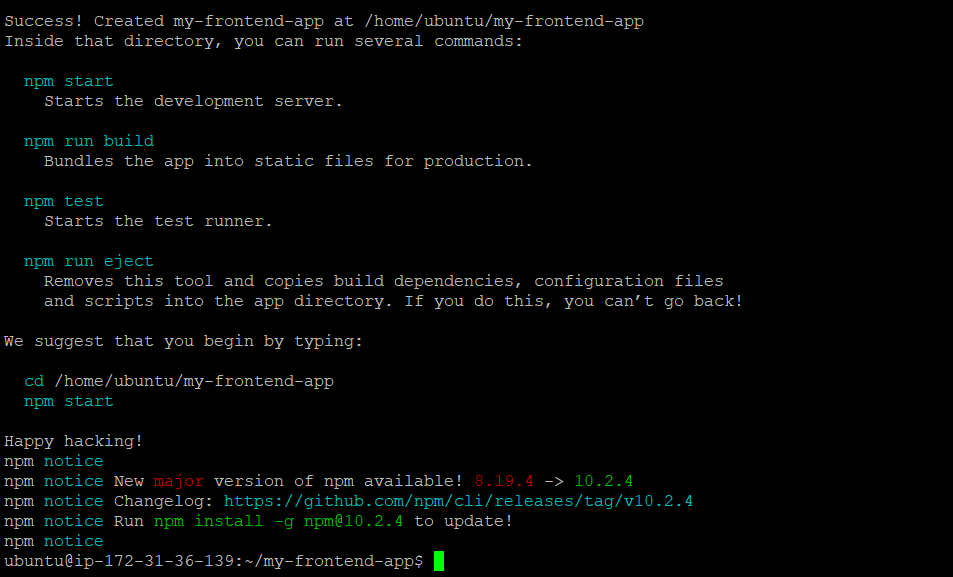
**Frontend Development and Amplify Deployment:**

Created a new directory for React frontend project **(my-frontend-app)**

**Initialized a React app using Create React App as below :**

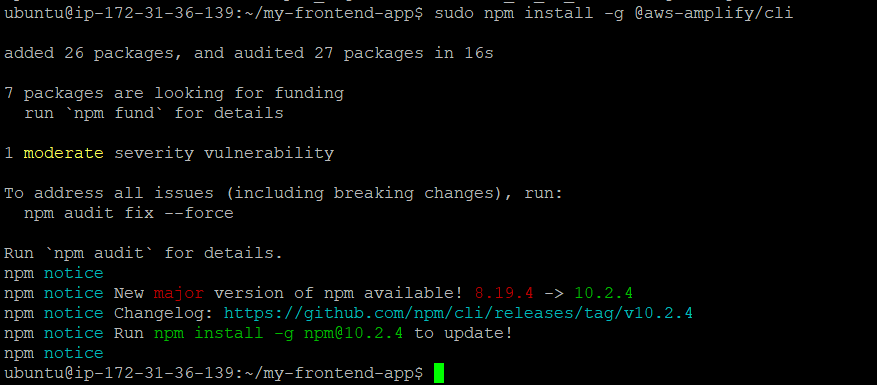


**We got successful result and app is created:**



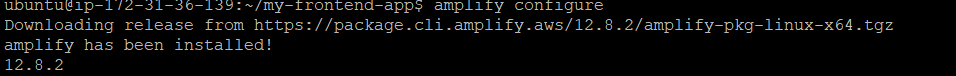
**Step 2: Set Up AWS Amplify**

**Installed the AWS Amplify CLI globally as below:**



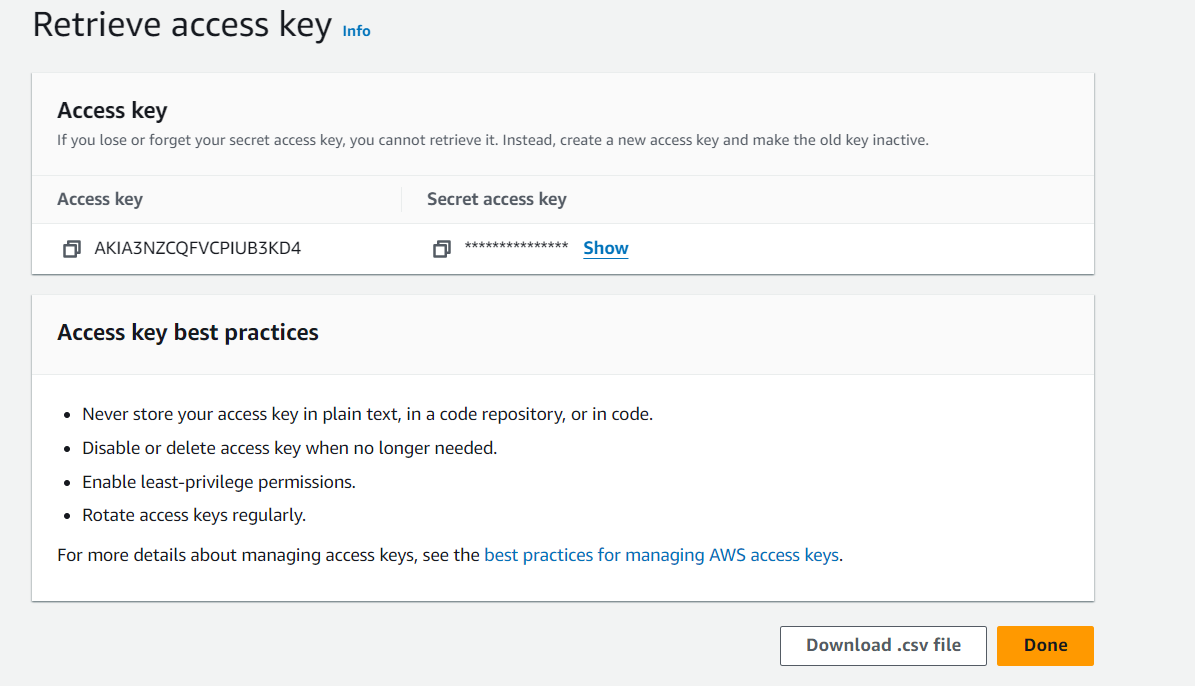
**Configured the Amplify CLI by using**

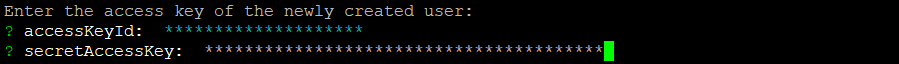
* Amplify configure **command**

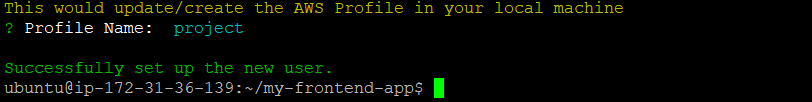


When doing configuration required aws secret key and access key so created it from AWS console:

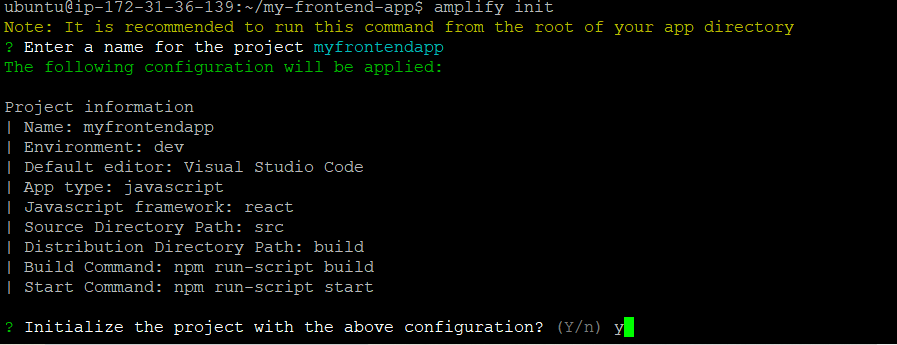
set up AWS account credentials.

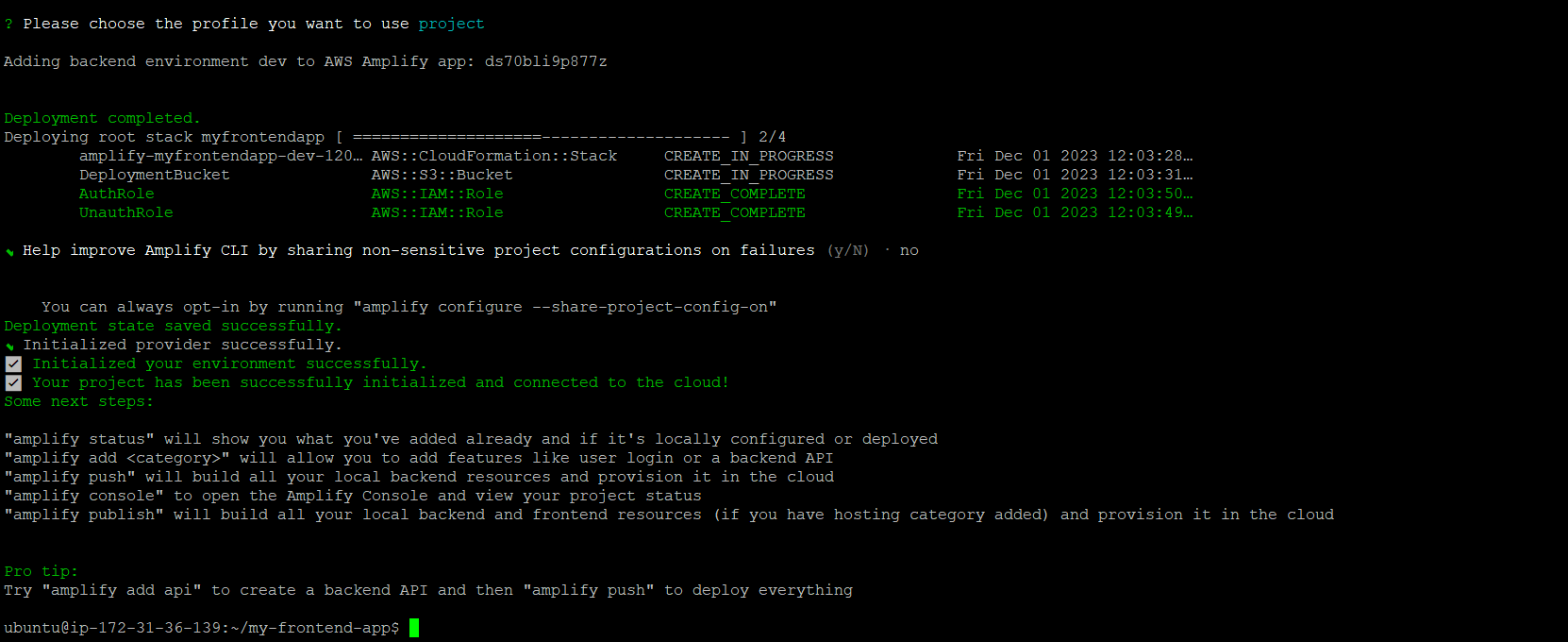






 Initialized an Amplify project by using **amplify init** command as below :

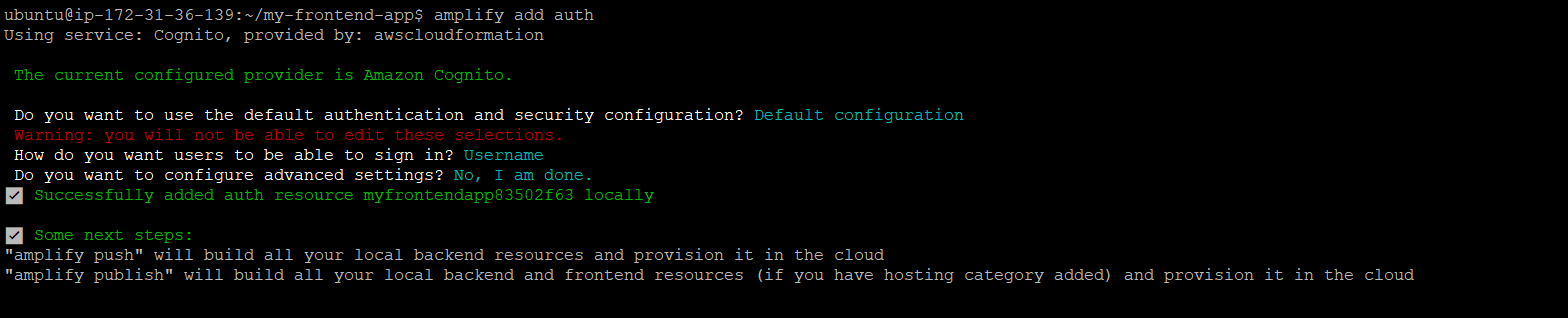




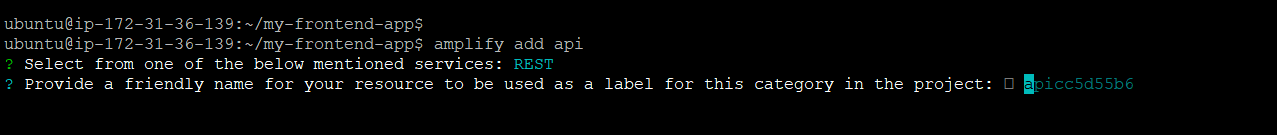
For configuring amplify project used

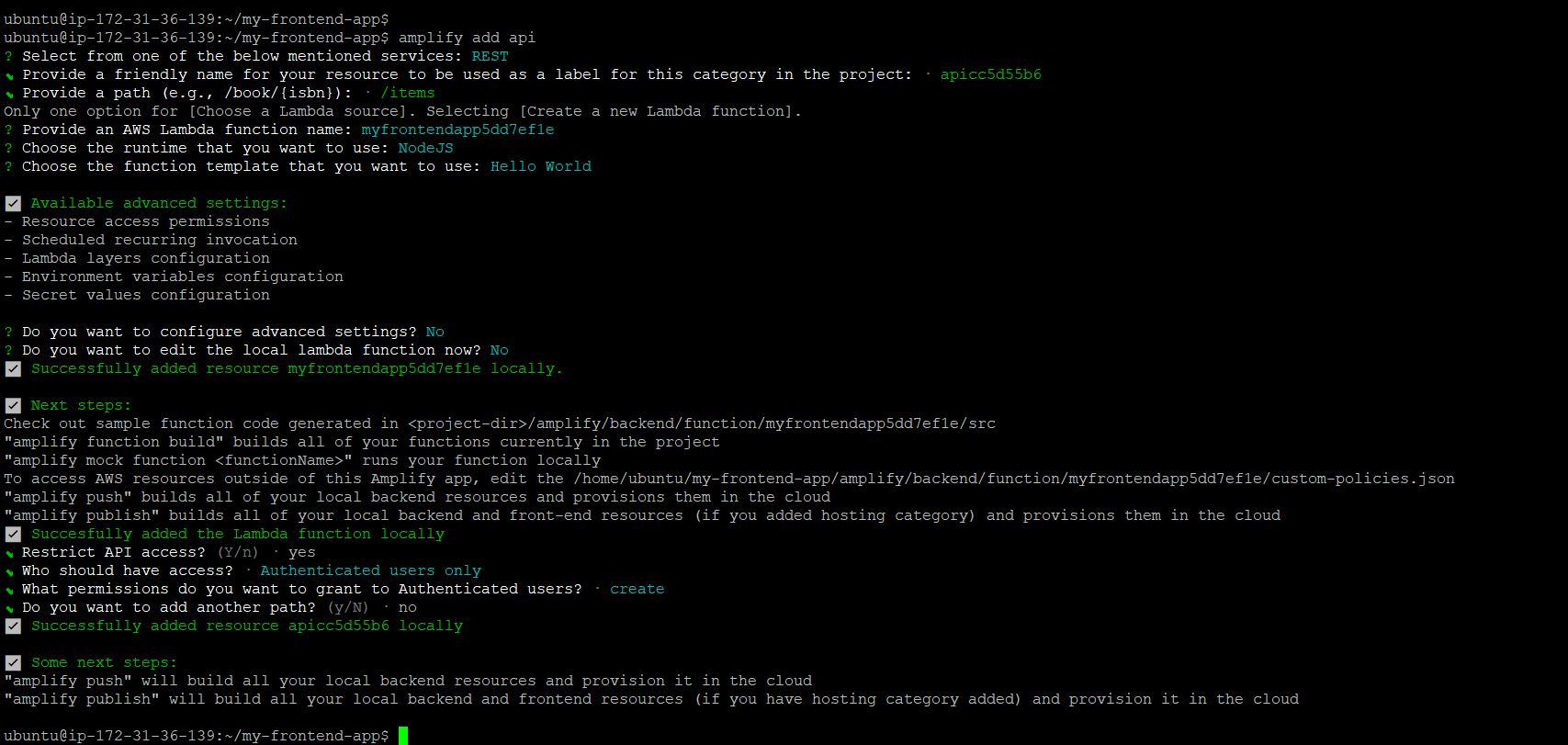
**Add backend resources like authentication, API, and storage using the Amplify CLI:**

* Add auth

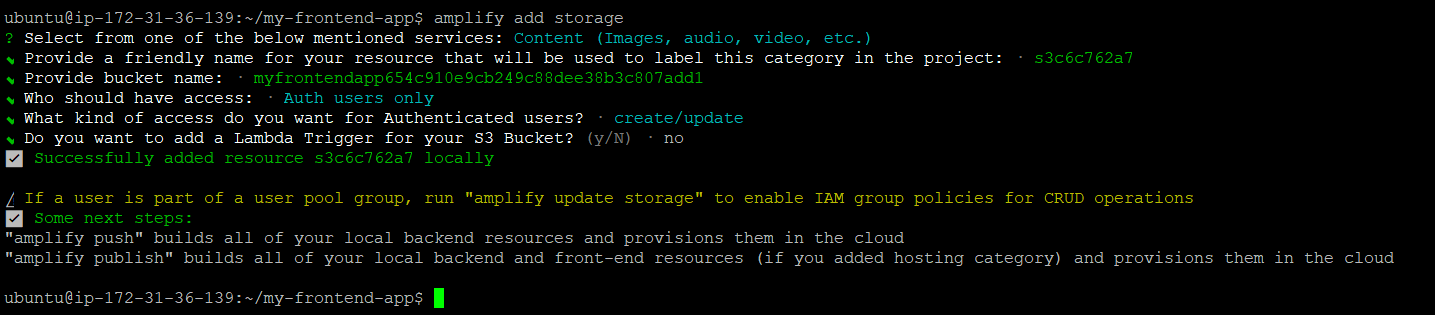


* Add api



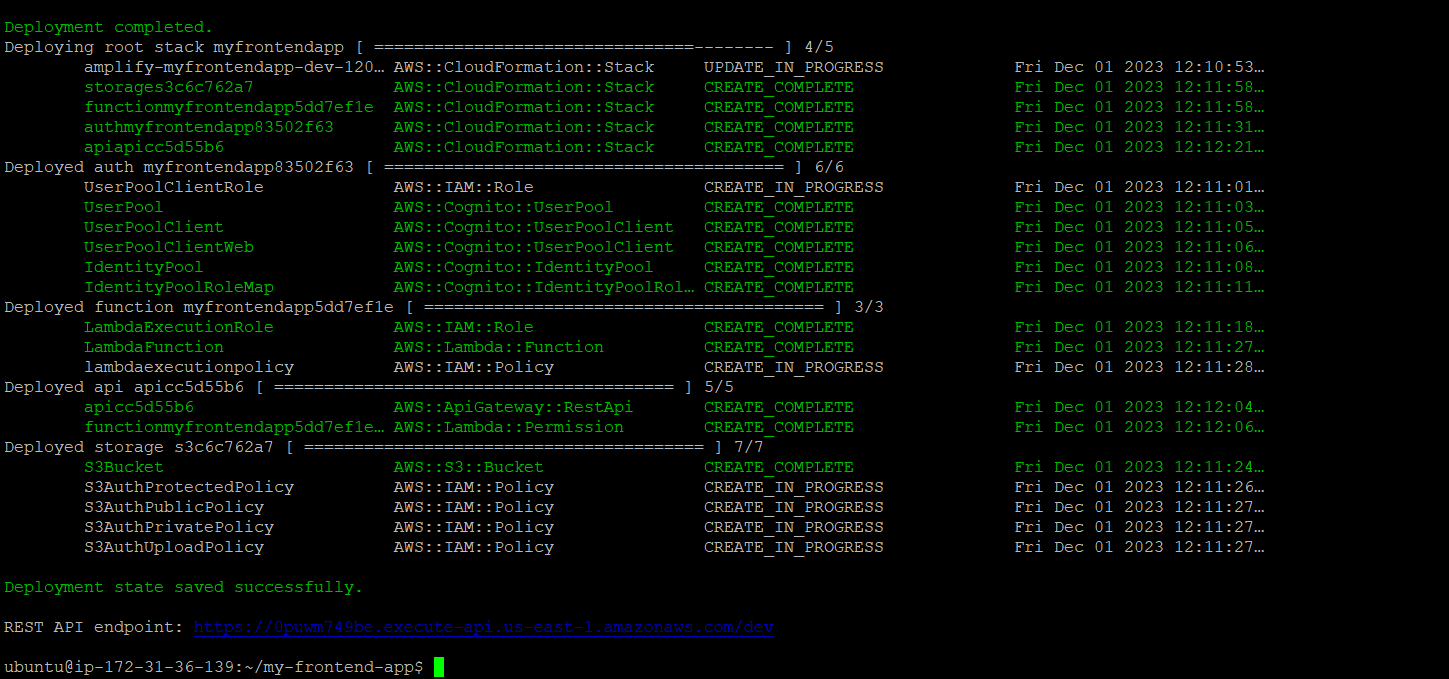


* Add storage



Pushed Amplify project to the AWS cloud as below:

* amplify push

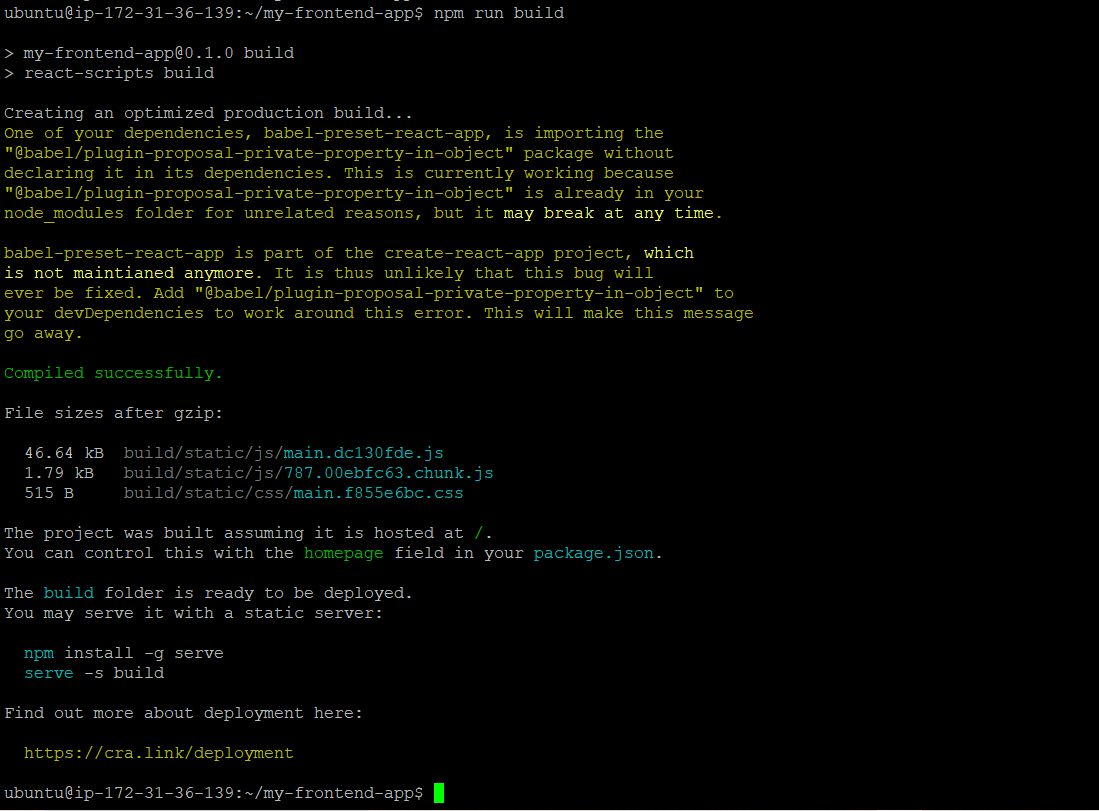


After successful deployment we got the Rest api Endpoint as below:

REST API endpoint: <https://0puwm749be.execute-api.us-east-1.amazonaws.com/dev>

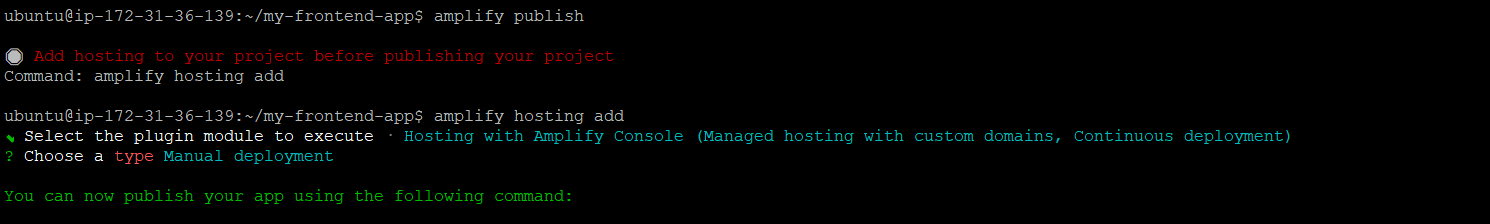
**Deployed Frontend using**

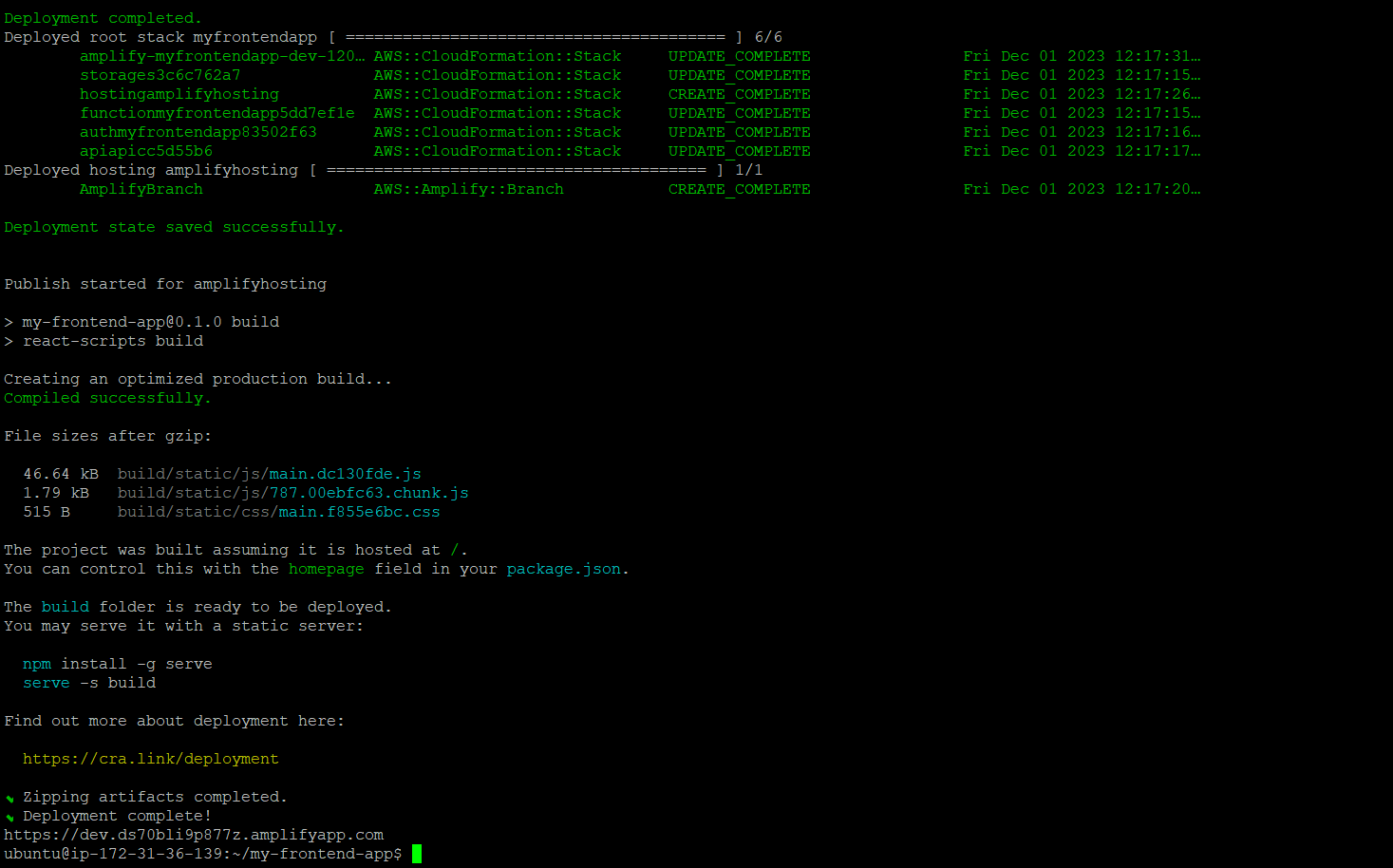
* **npm run build**



After building next step is to publish it

So published as below using below command:

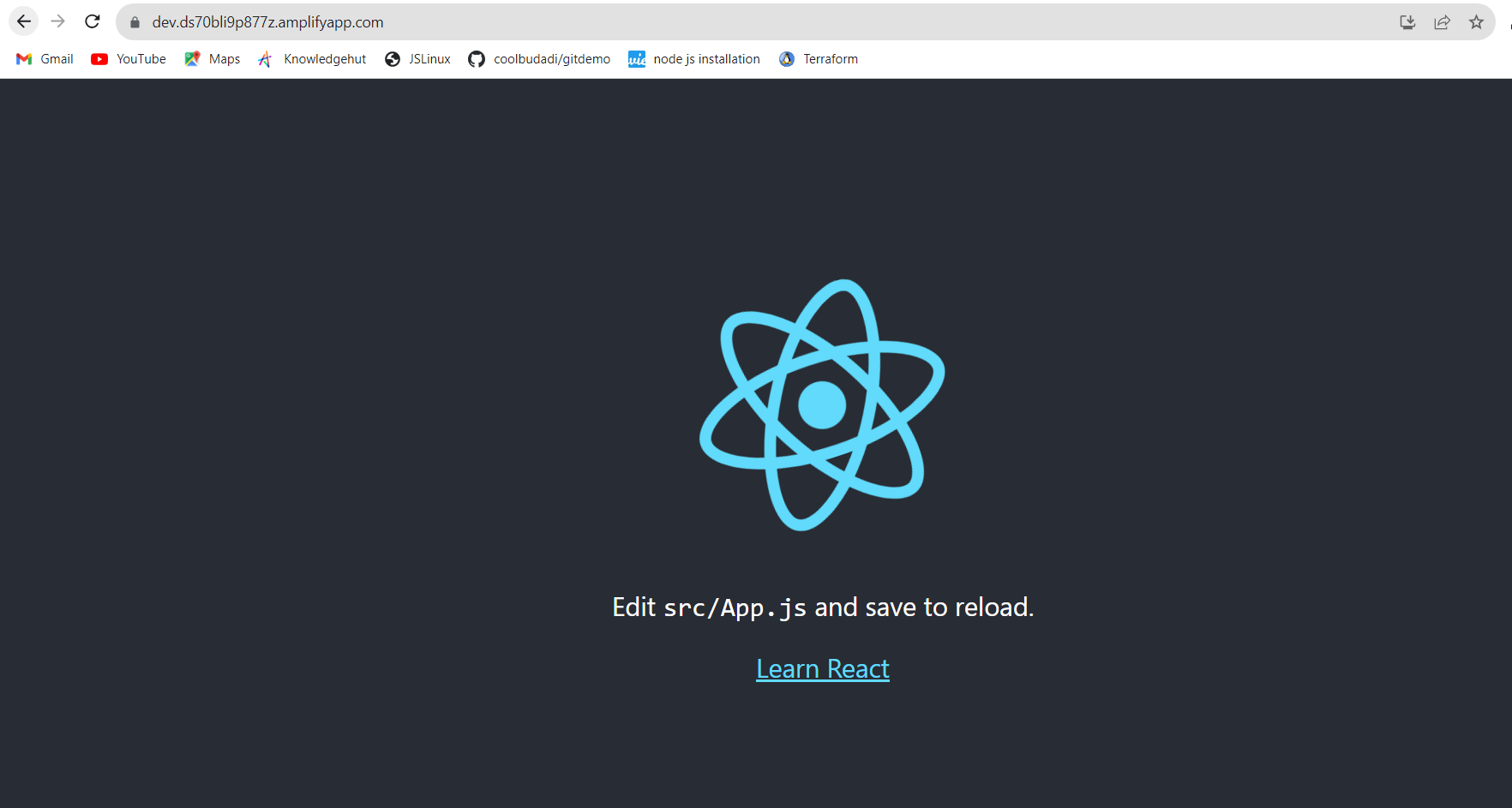


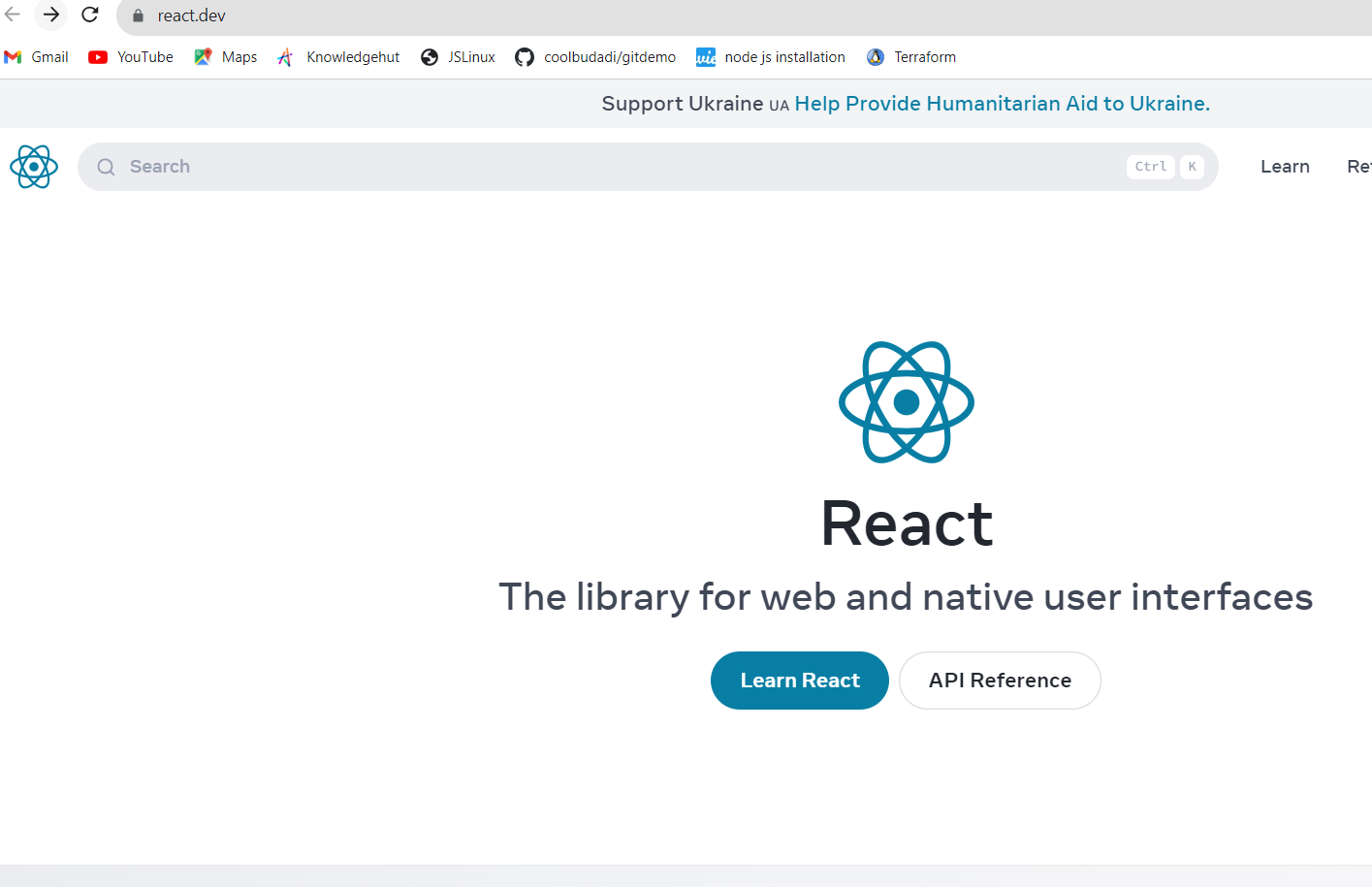


After deployment completed got url for accessing app on to the browser as below:

Url= <https://dev.ds70bli9p877z.amplifyapp.com>

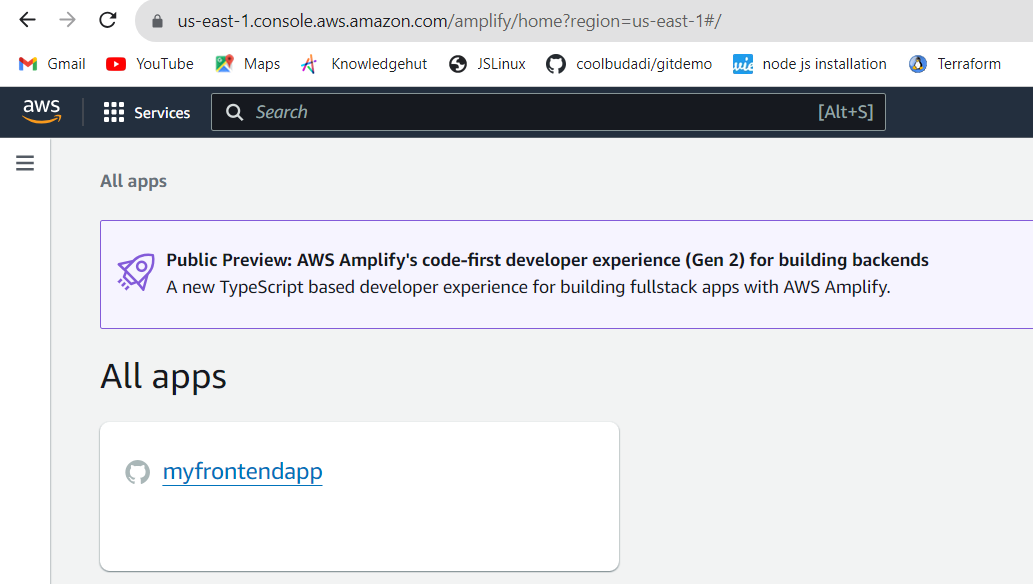
After hit up above url on the browser we got below result as below:

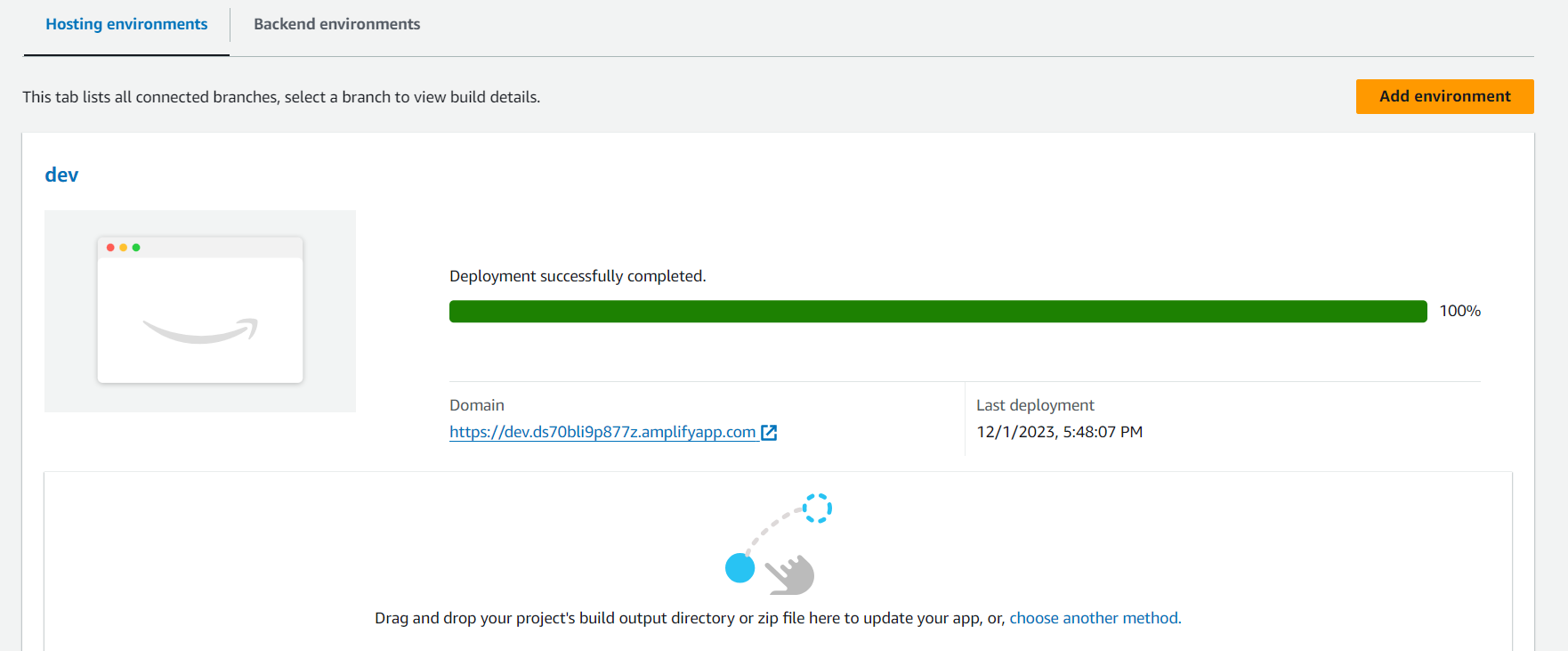


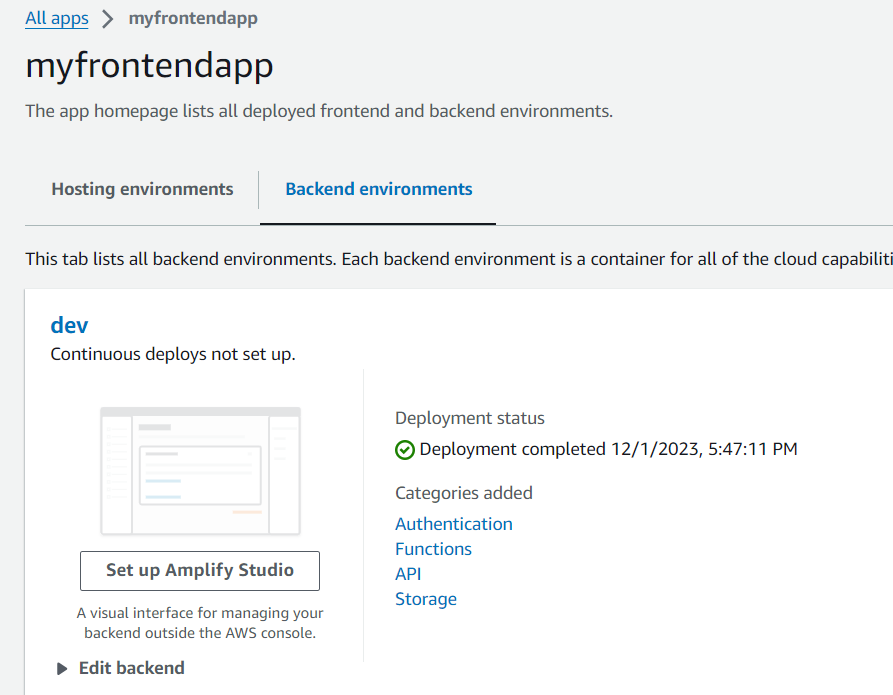


<https://react.dev/>

**Successful Results from AWS console as below :**







**Middleware development and Elastic Beanstalk deployment.**

**Middleware Development**• Created a new directory for middleware project:



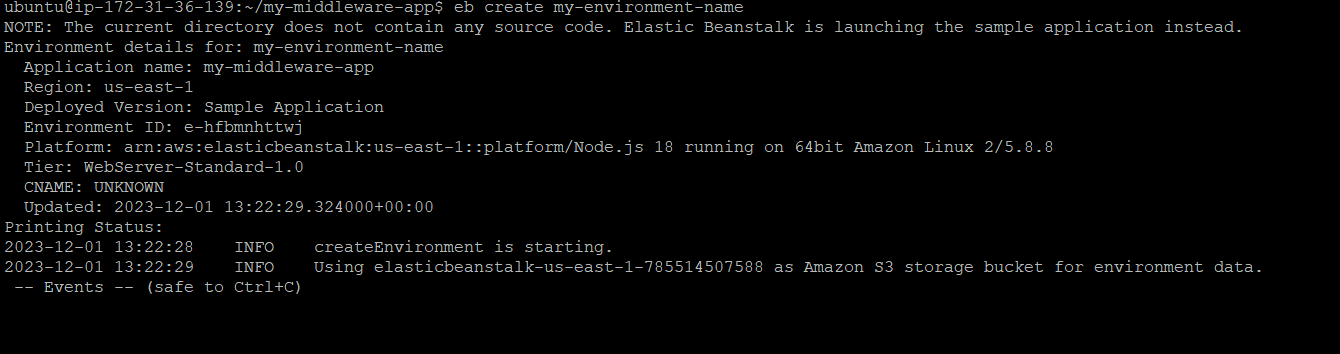
**Set Up AWS Elastic Beanstalk:**

Installed the AWS Elastic Beanstalk CLI (EB CLI) globally:

Initialized an Elastic Beanstalk application.

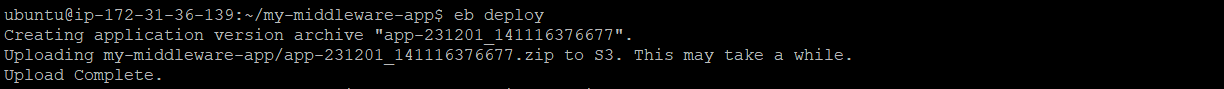
eb init -p “node-js” “my-middlware-app”

**Created an Elastic Beanstalk Node.js environment as below:**



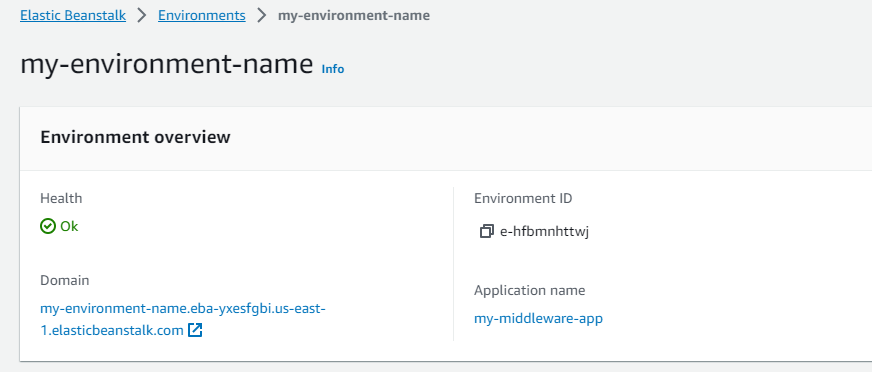
Deployed middleware application to Elastic Beanstalk by using below command:

* eb deploy

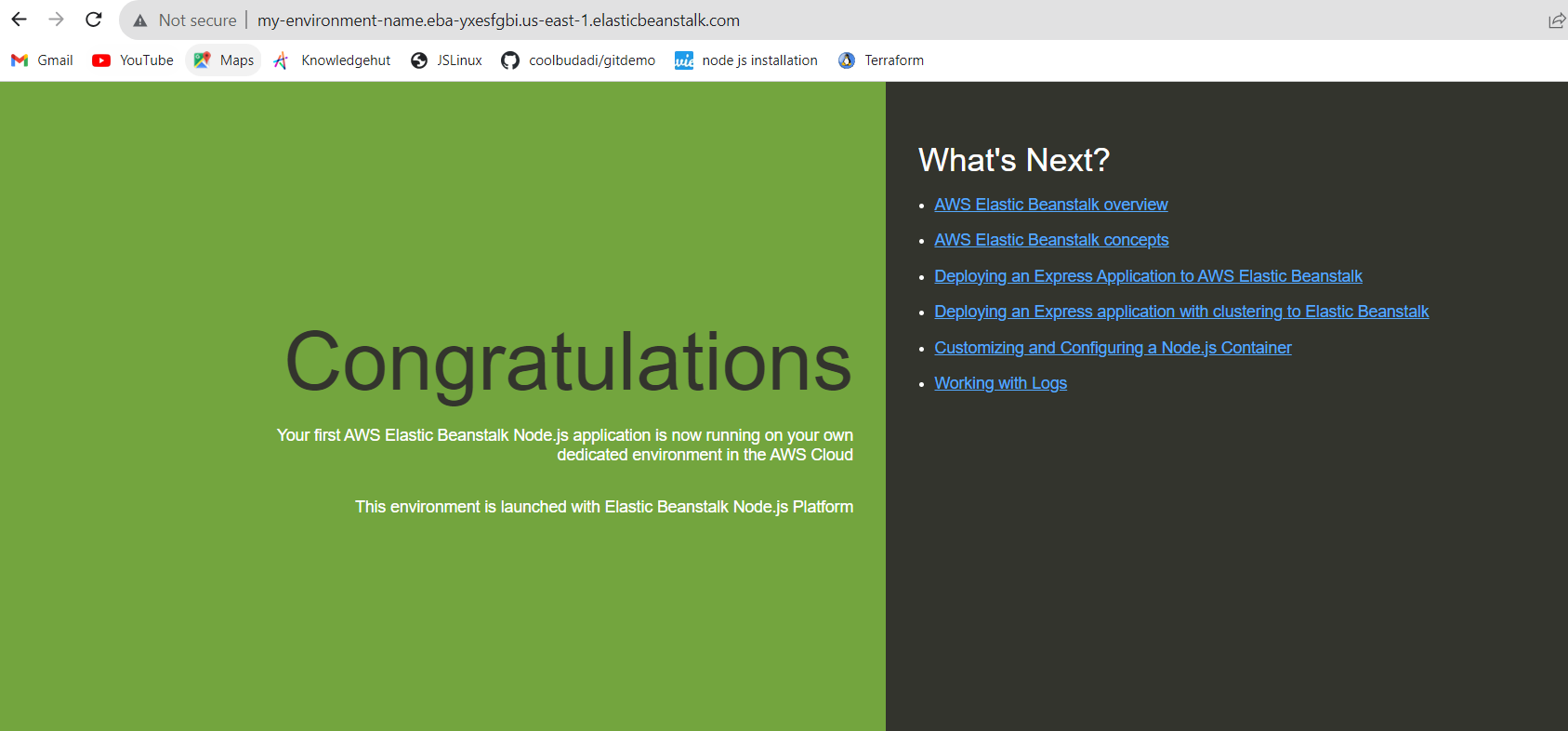


**As we can below the staus is Ok**

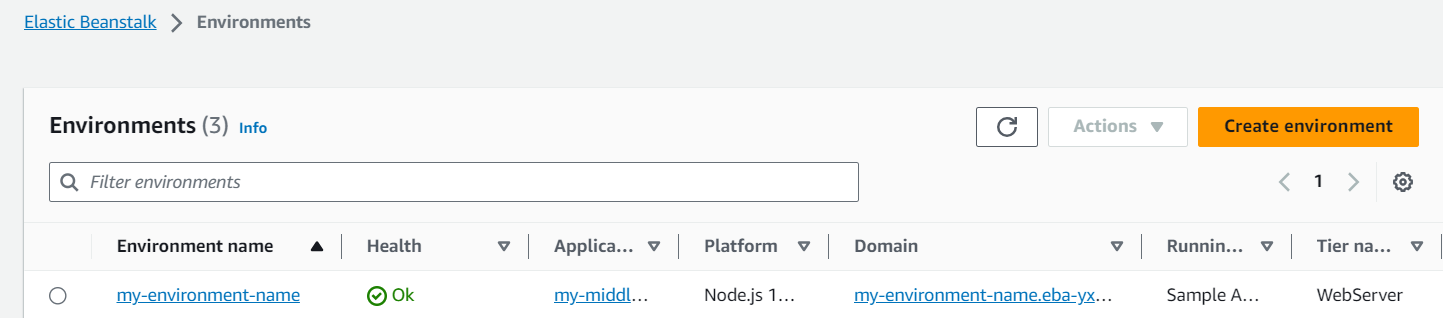
Url : [my-environment-name.eba-yxesfgbi.us-east-1.elasticbeanstalk.com](http://my-environment-name.eba-yxesfgbi.us-east-1.elasticbeanstalk.com/)



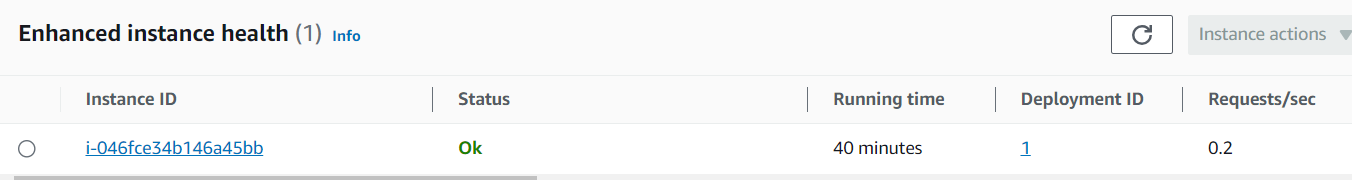
**By hit url we got below result as below:**



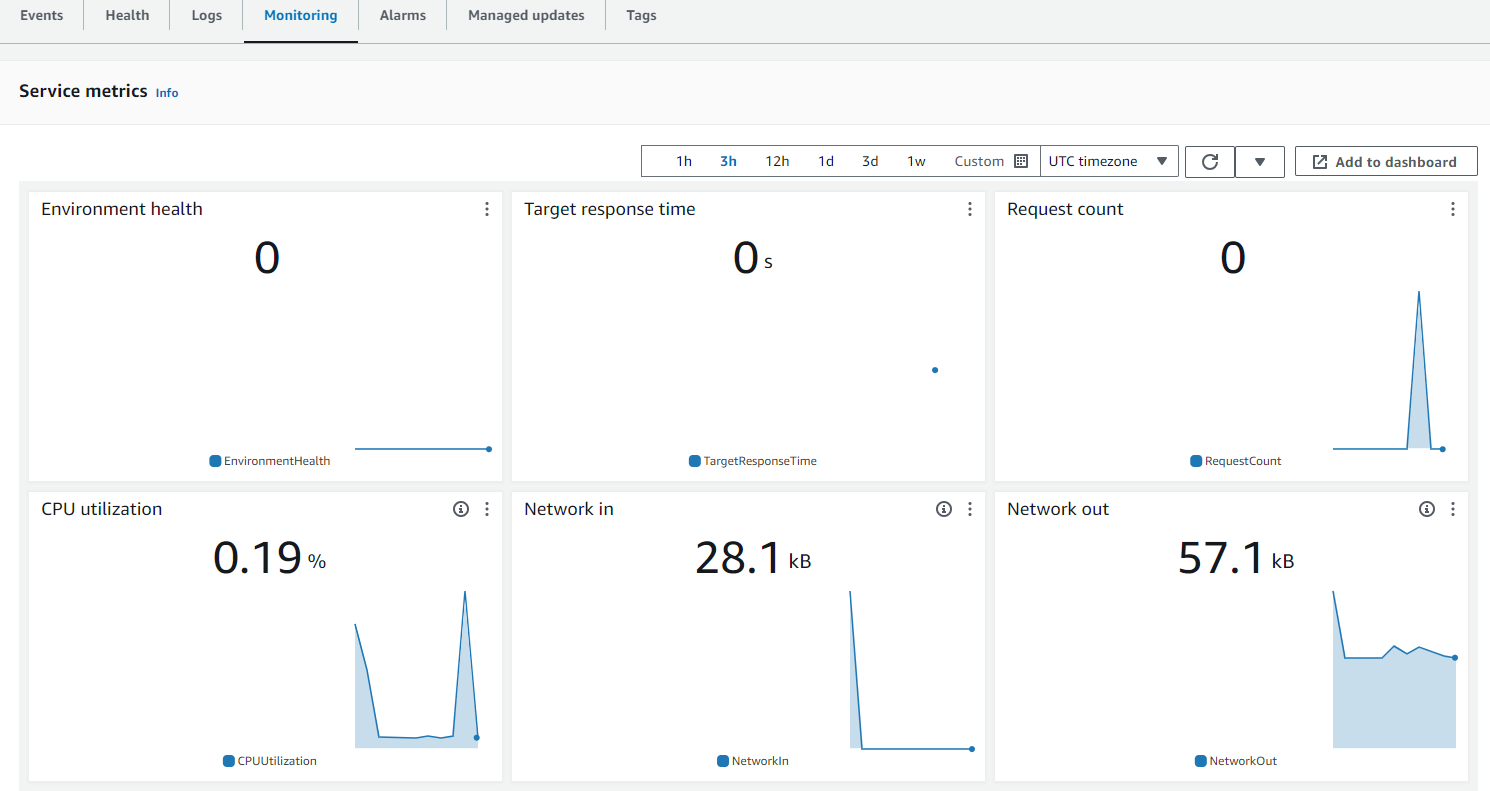
**Successful Results from AWS console as below :**



Instance details:



Monitoring:

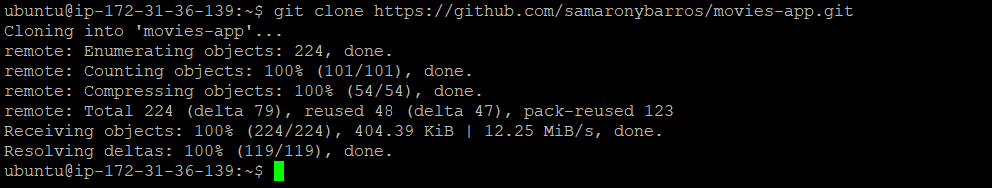


**AWS Elasticbeanstalk Url :**

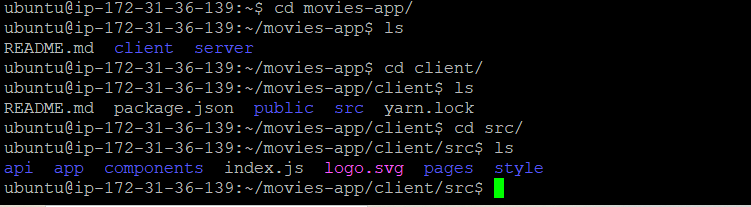
<https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environments>

**Backend Containerization and EKS Deployment**

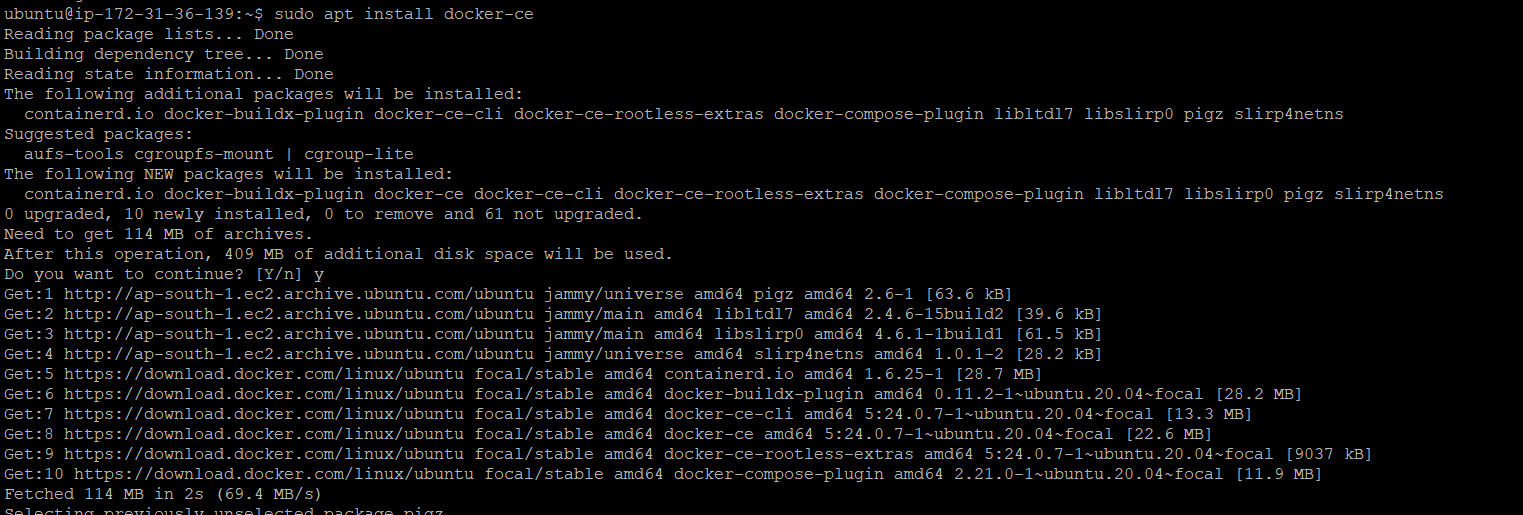
**Cloned the app code on to the machine as below:**



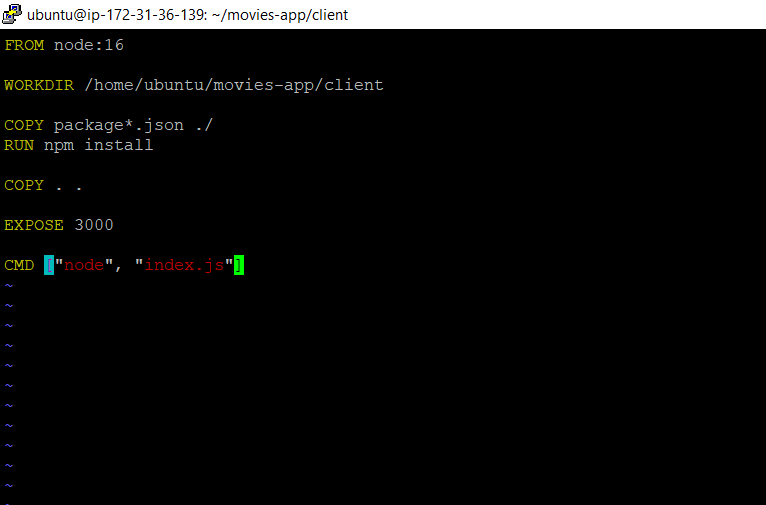
**Changed directory to the application code as below :**



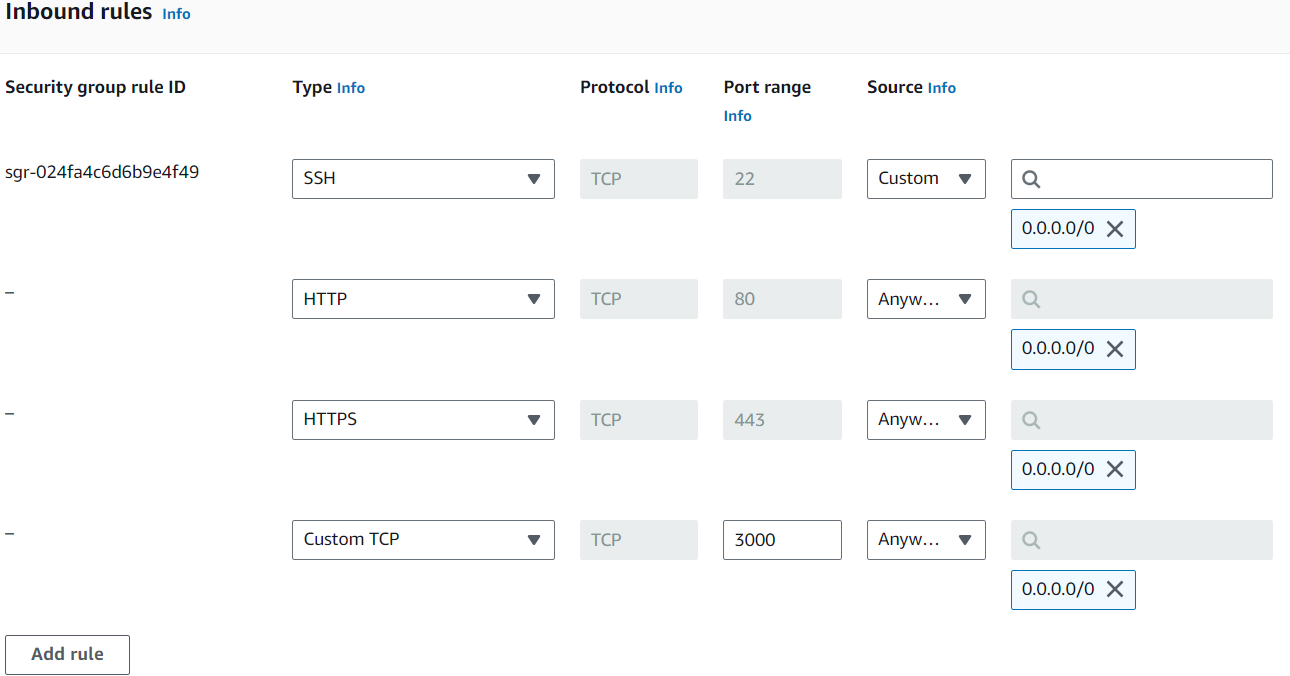
**Installed Docker on to the machine :**



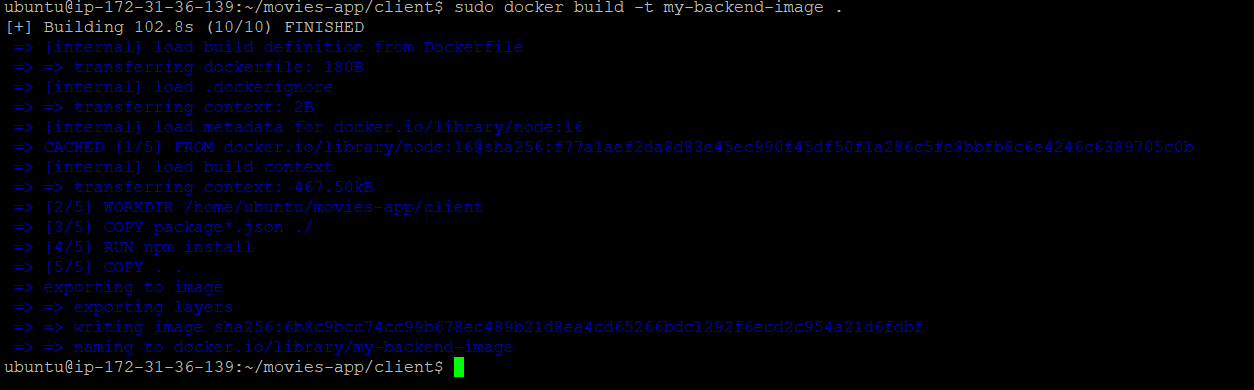
Created a `Dockerfile` in backend project directory as below:

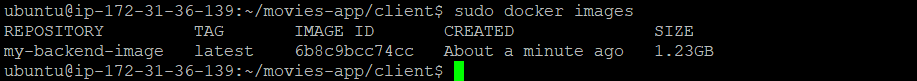


Added 3000 port entry for accessing node js as below :

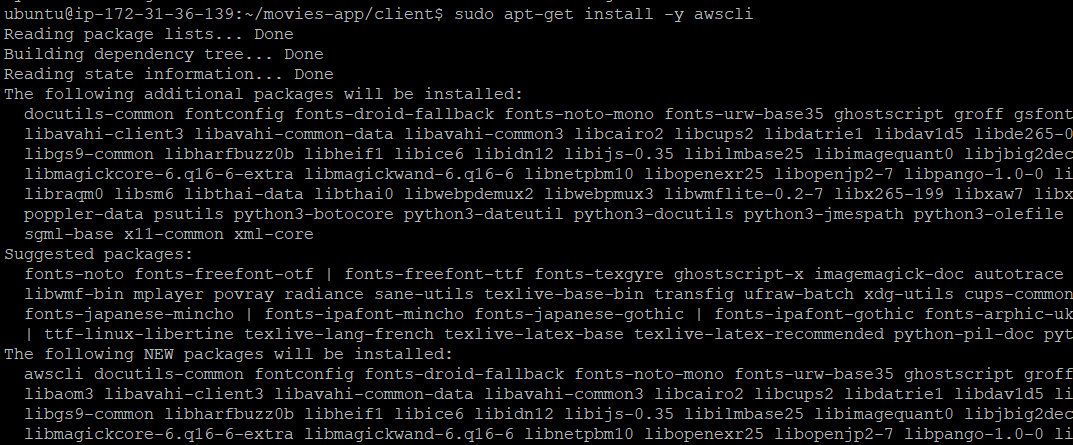


**Built up and tagged the Docker image as below :**

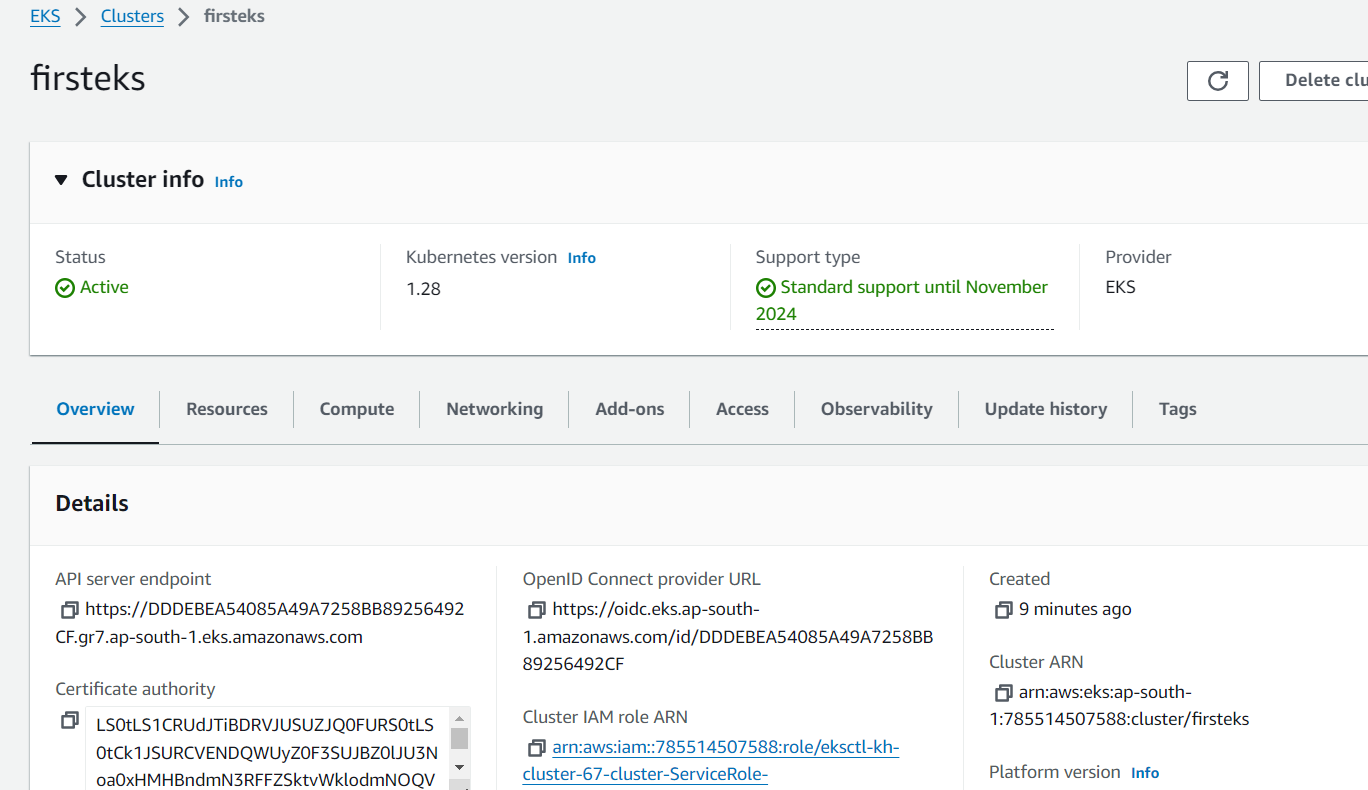




**Installed AWS cli :**

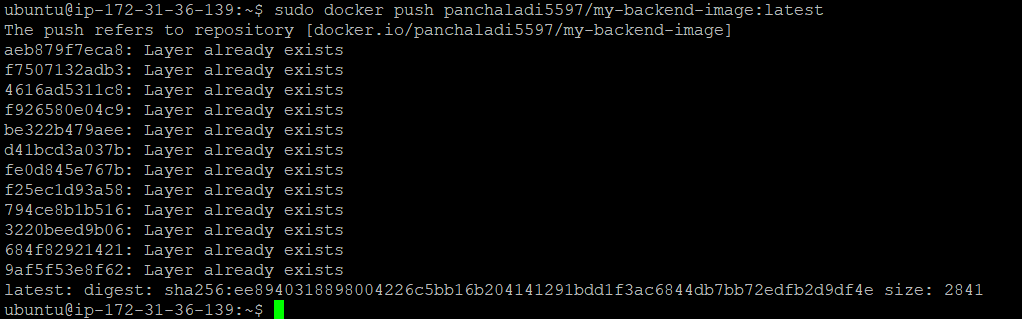


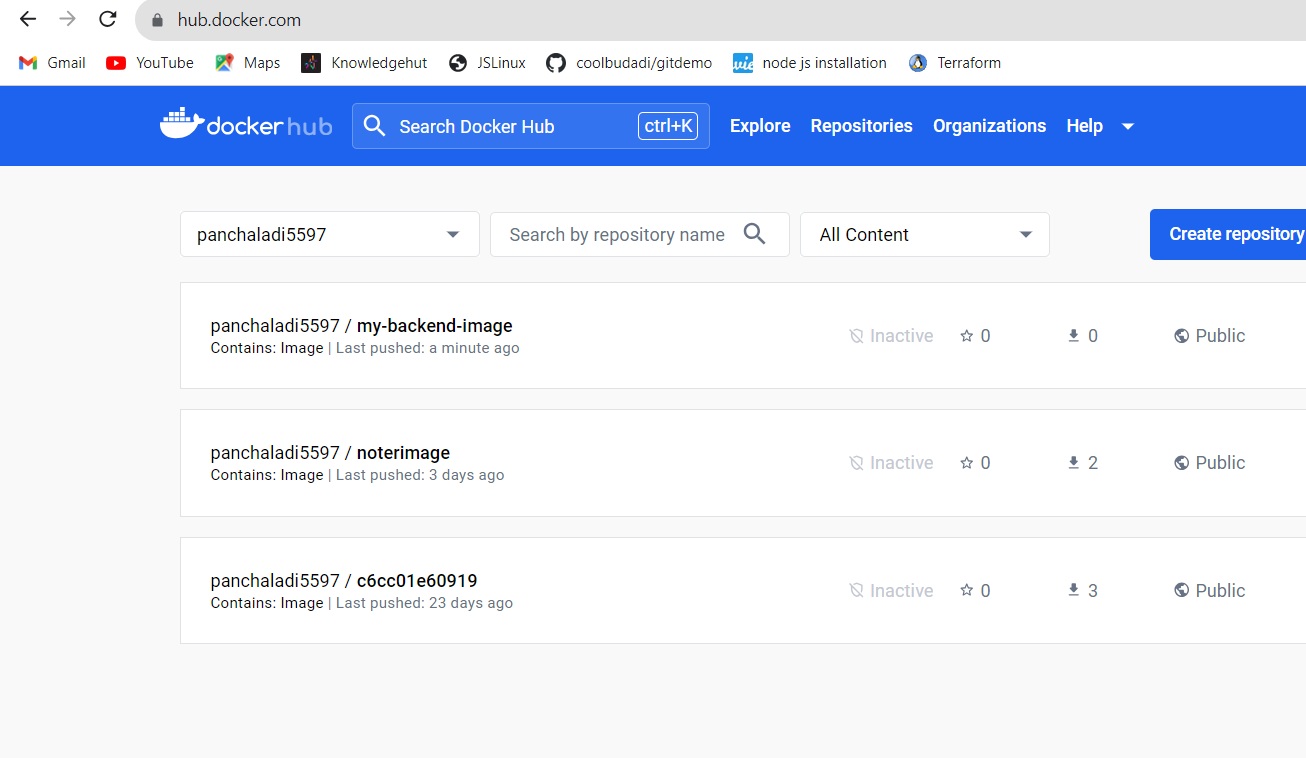
**Created EKS cluster :**



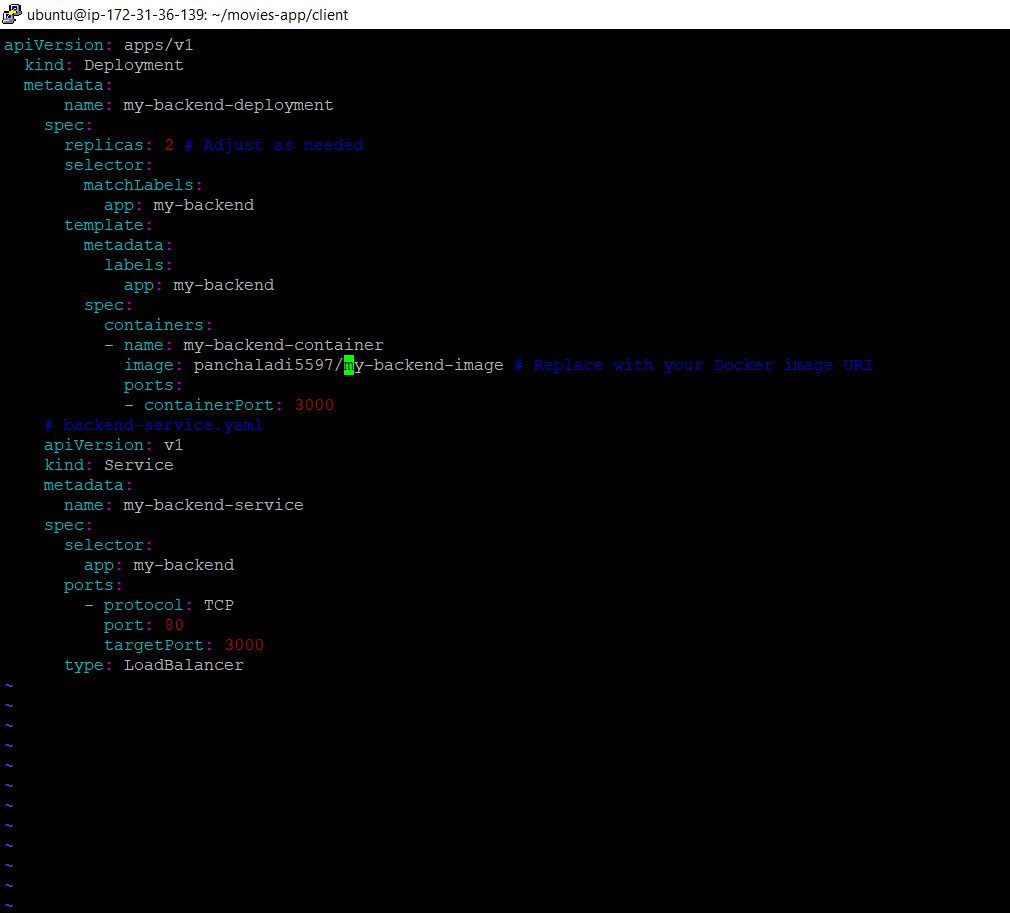
Installed Kubectl On to the machine as below as access cluster :







Created a yml file (my-backend-deployment.yml)



•    Applied the Kubernetes configuration to deployed backend and got successful result.