**Graded Assignment on Container Orchestration**

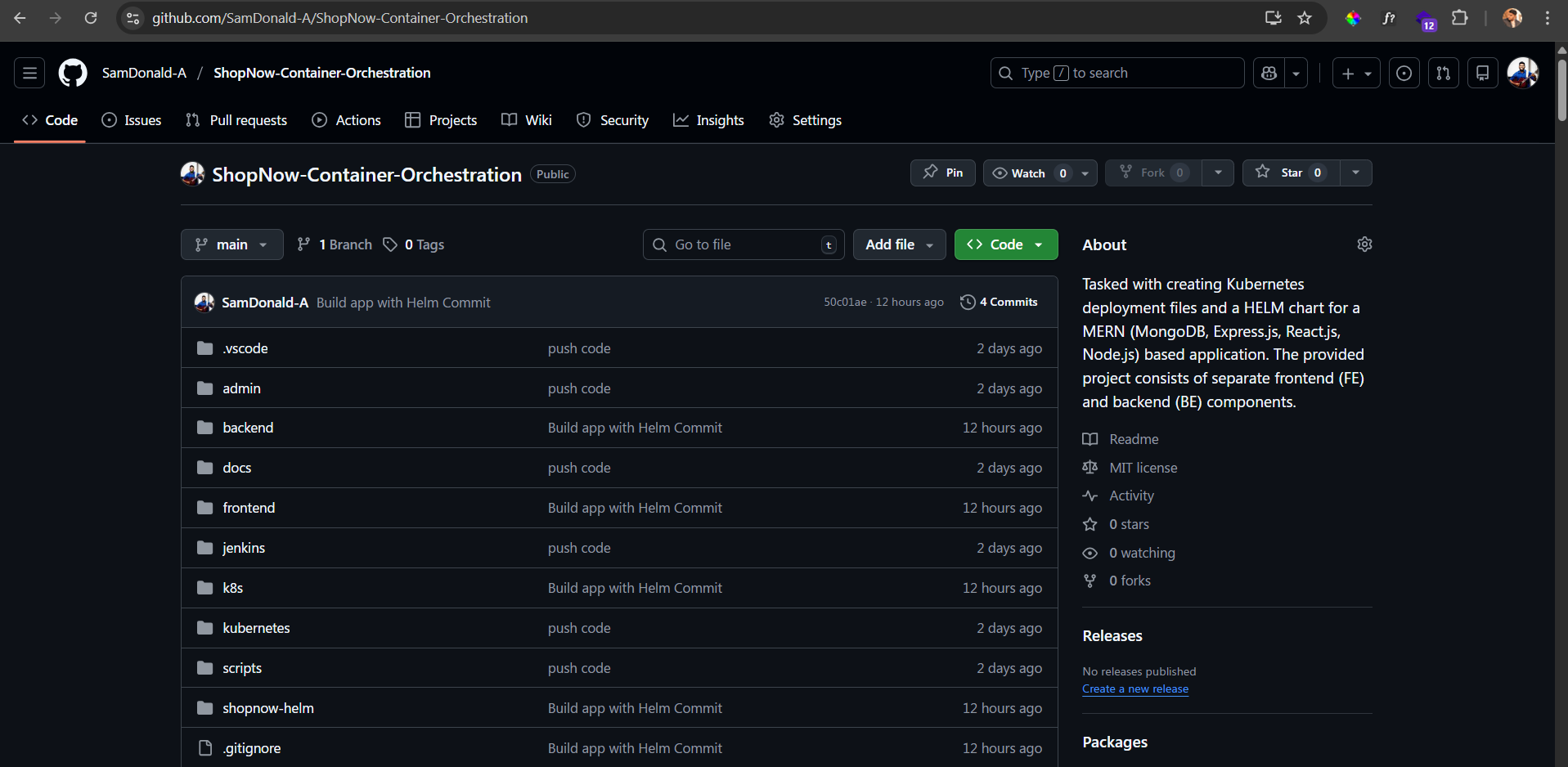
**Objective:**  creating Kubernetes deployment files and a HELM chart for a MERN (MongoDB, Express.js, React.js, Node.js) based application.

GitHub Link: <https://github.com/SamDonald-A/ShopNow-Container-Orchestration>

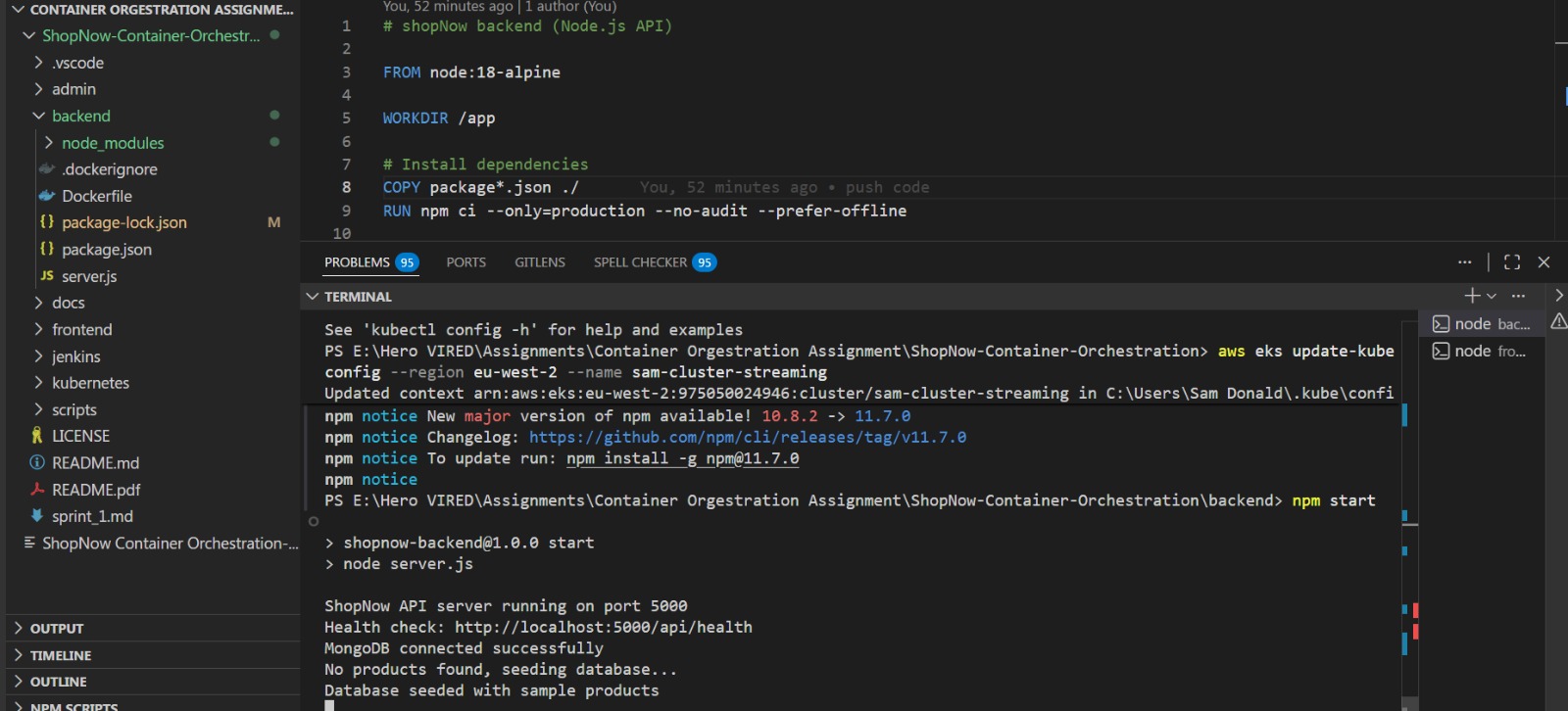
Read file: <https://github.com/SamDonald-A/ShopNow-Container-Orchestration/blob/main/README.md>

**Step 1: Git and local code setup**

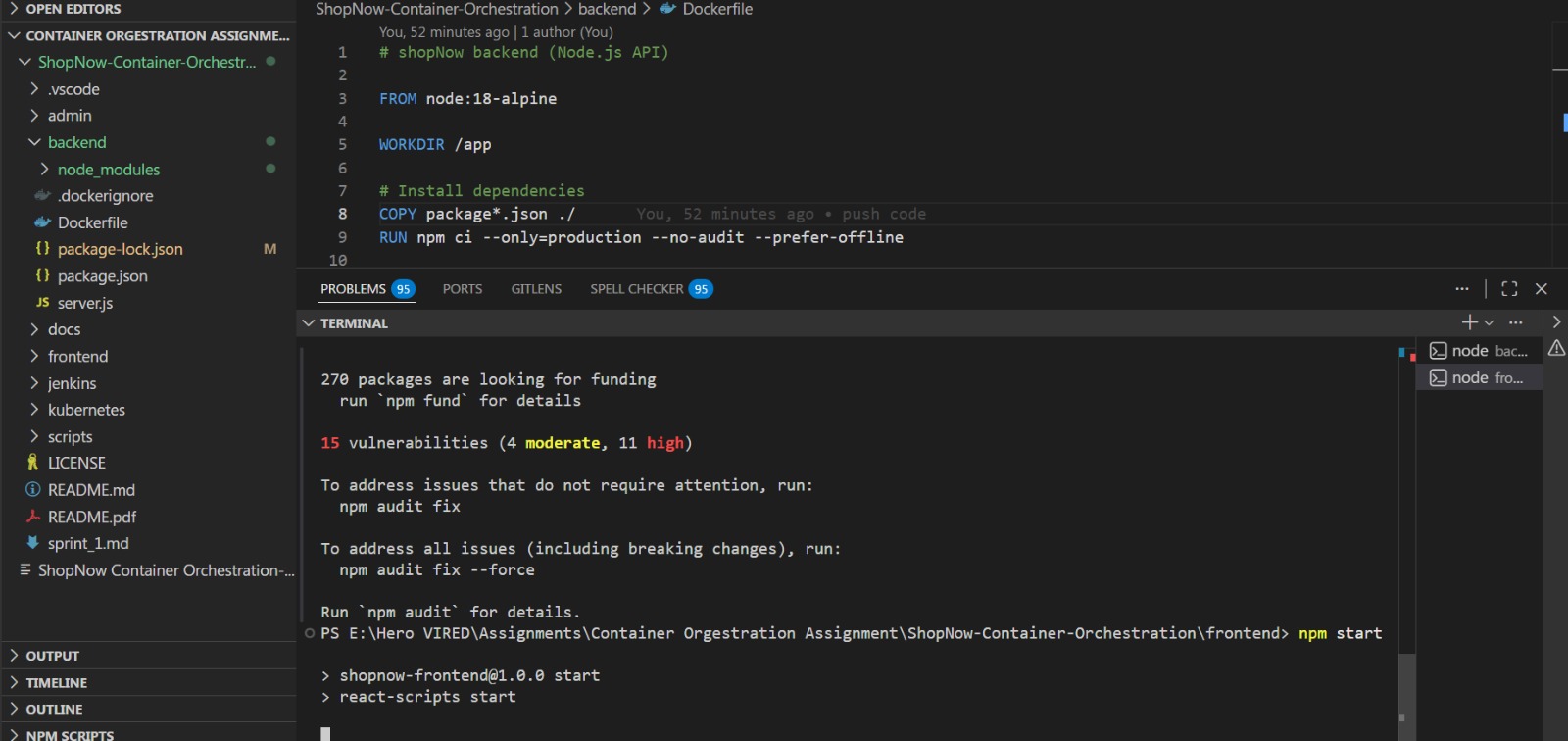
* **Fork or Clone and Push it to your repo**

****

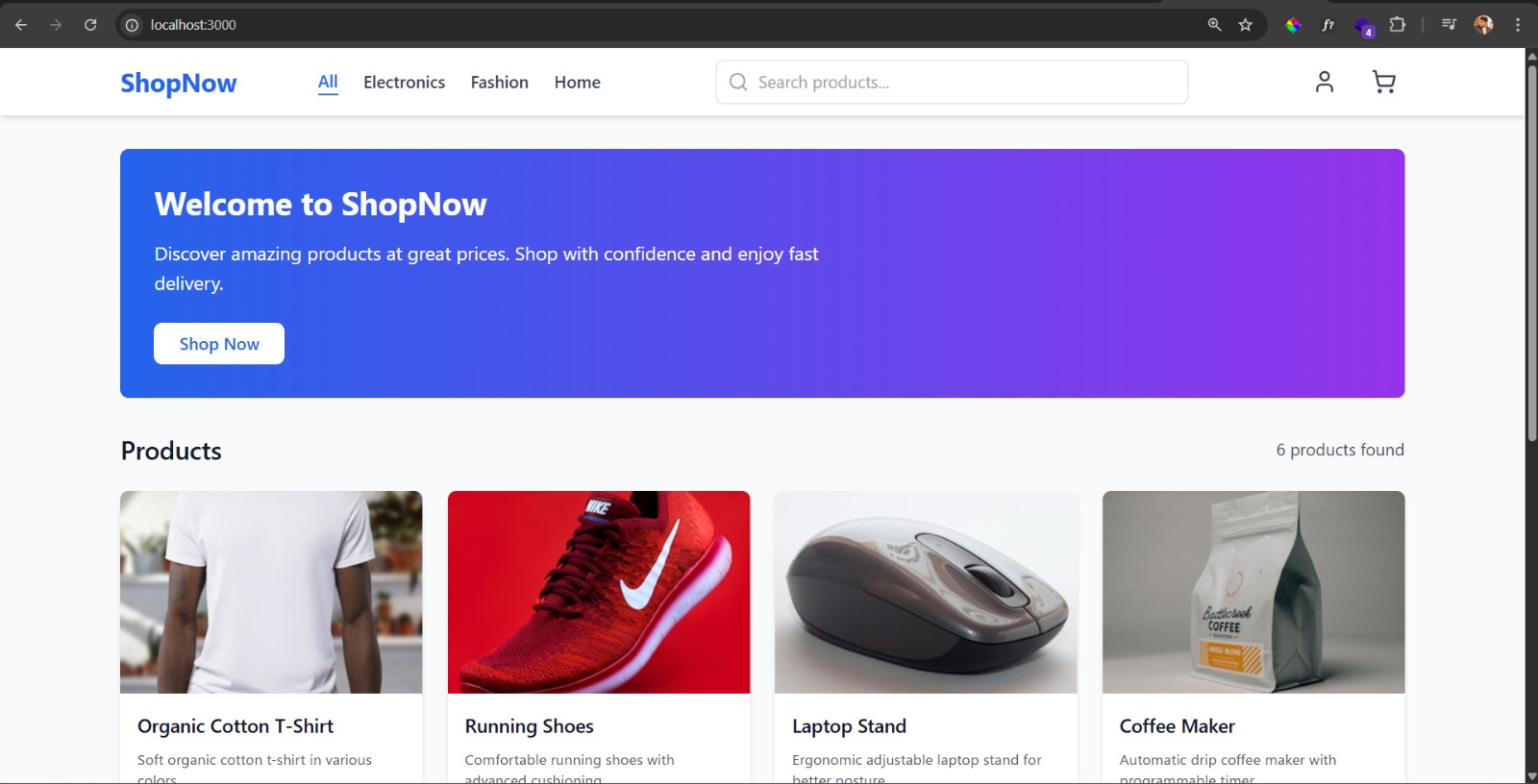
* **Go to backend folder and do npm install then start the backend server to locally run the app for testing**

****

* **Do the same for frontend as well**

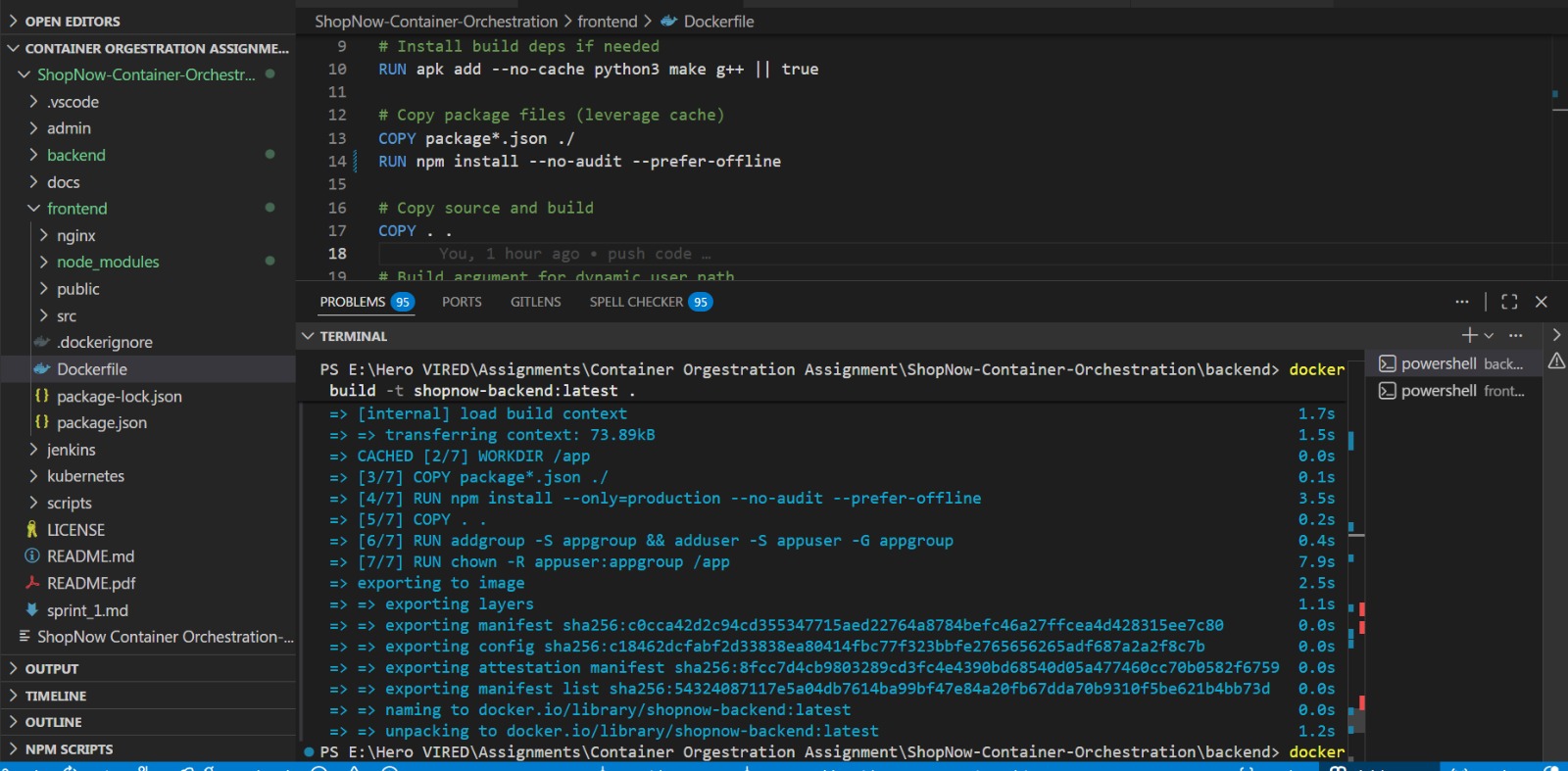
****

* **App is running locally without Docker and Kubernetes**

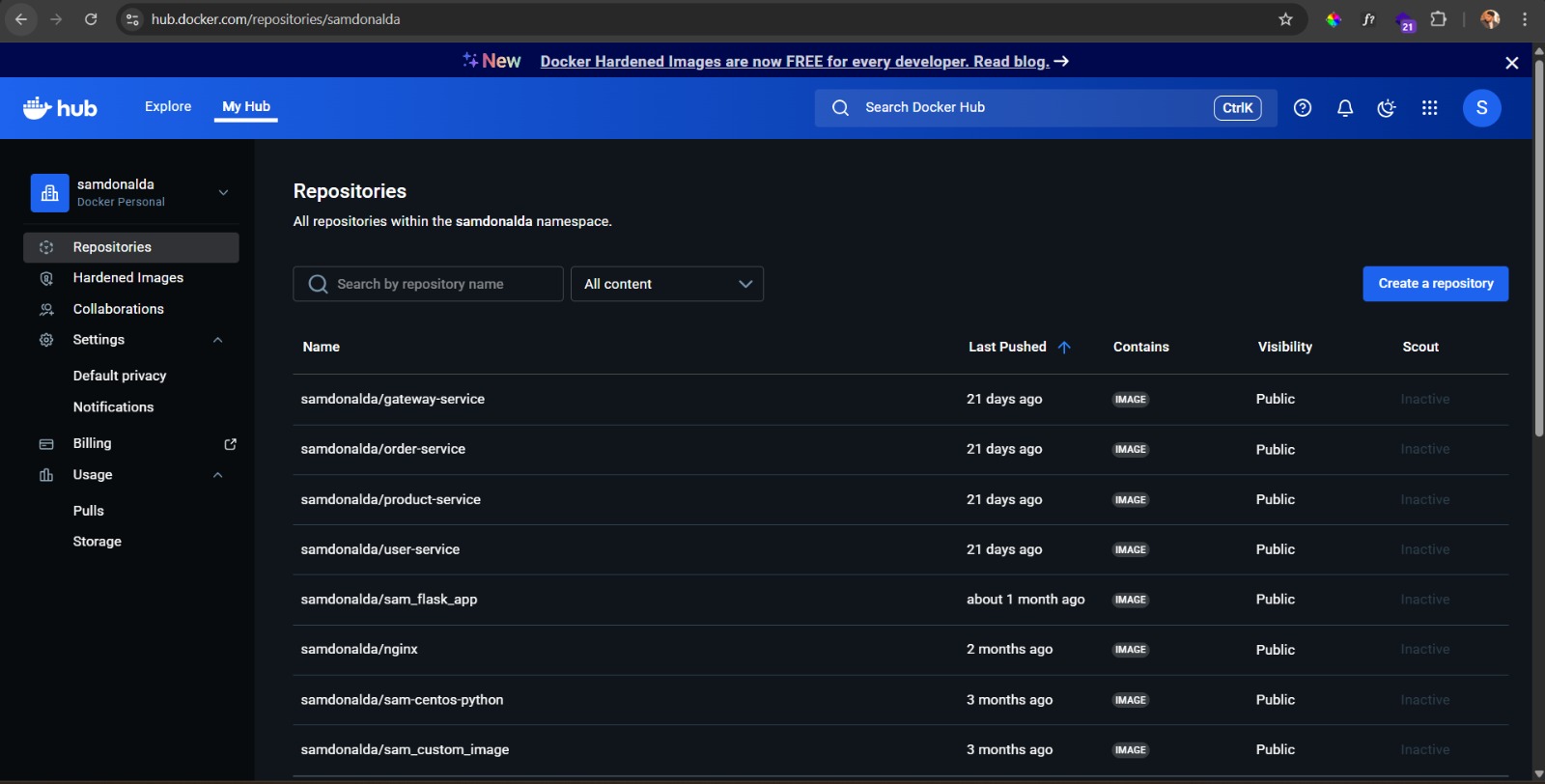
****

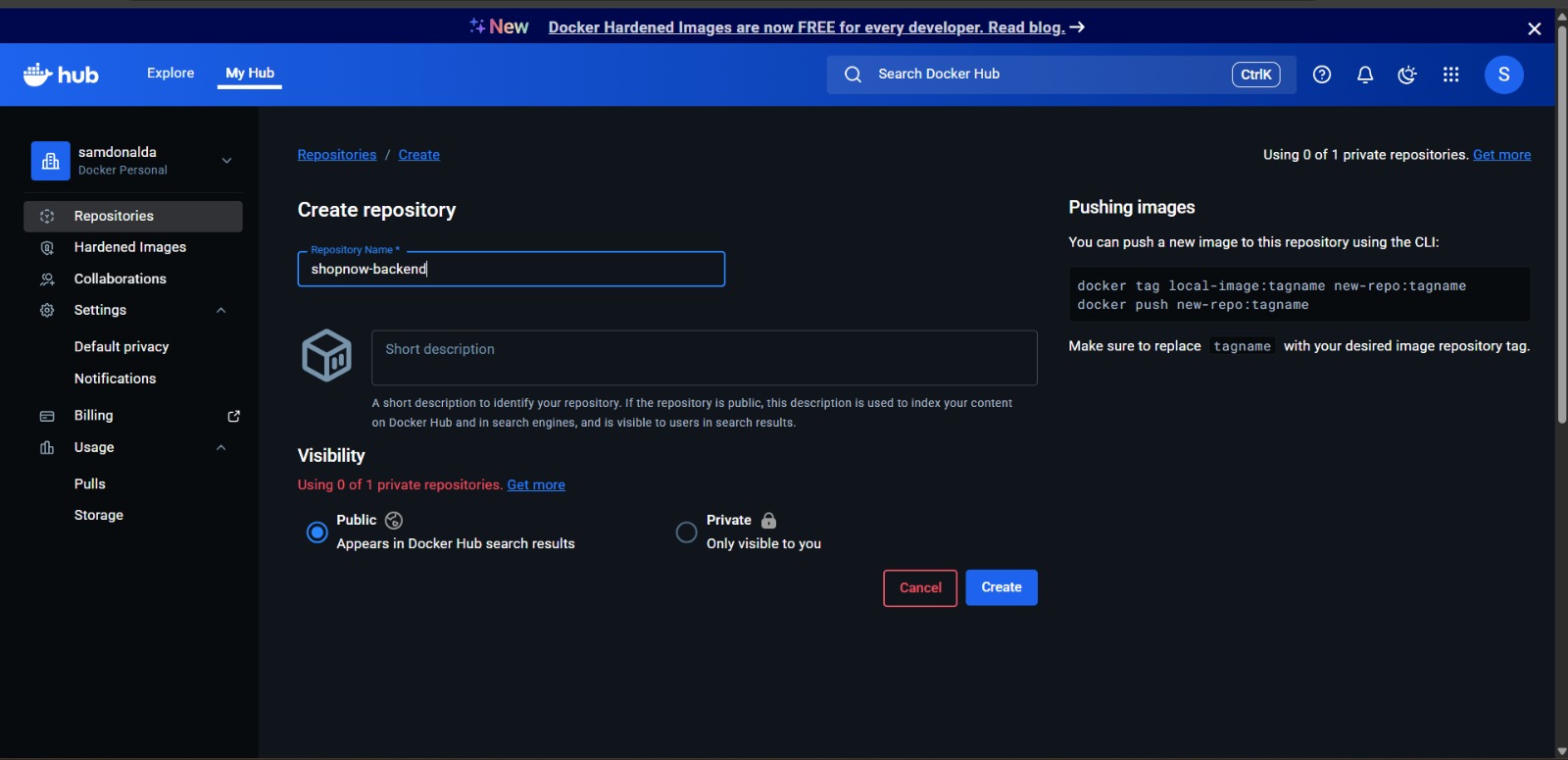
**Step 2: Containerization - Docker setup**

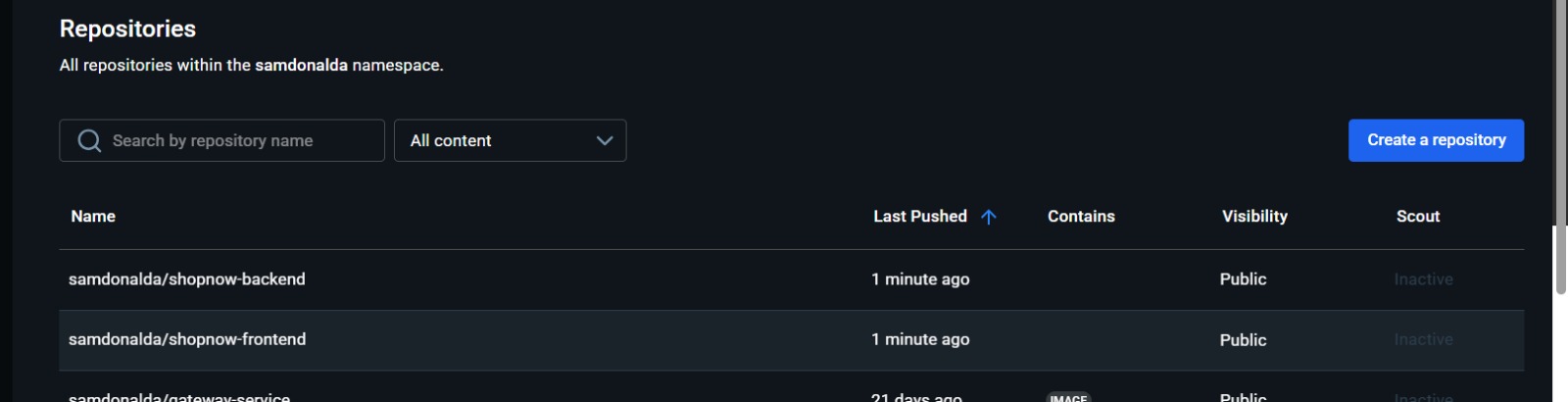
* **Since we have Dockerfile Build it**

****

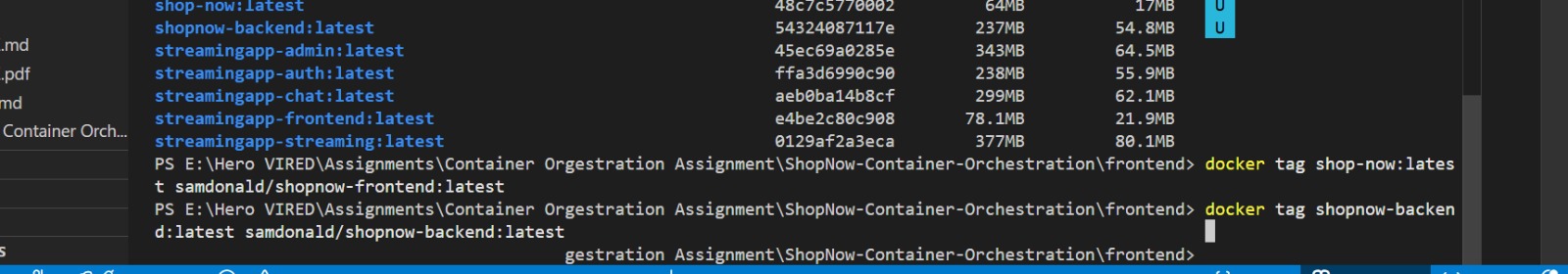
* **Need to create repository for frontend and backend in DockerHub for pushing images**

****

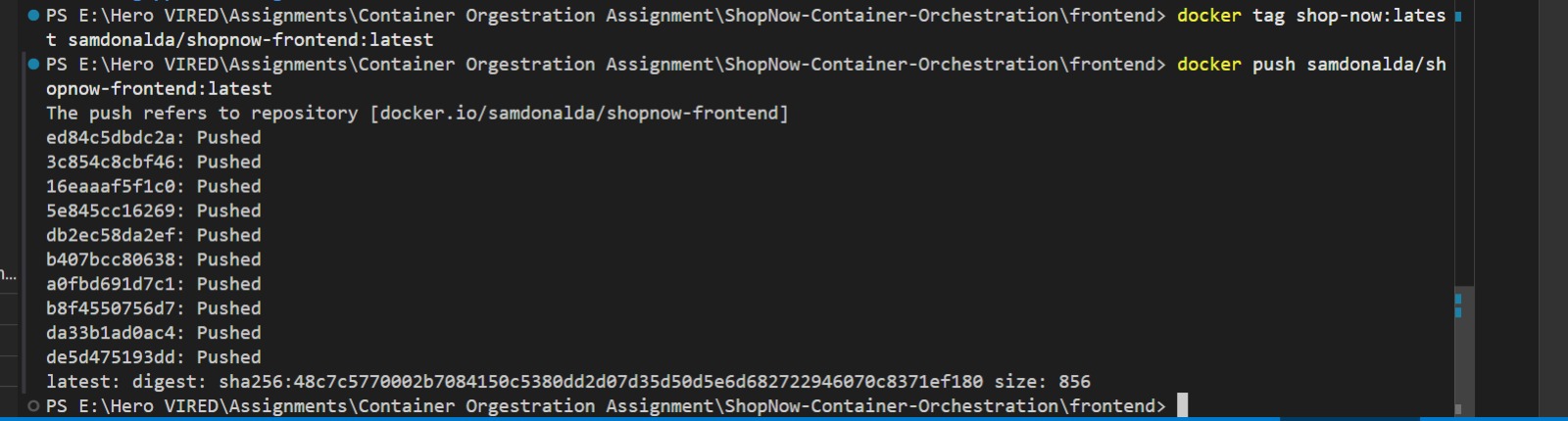
****

****

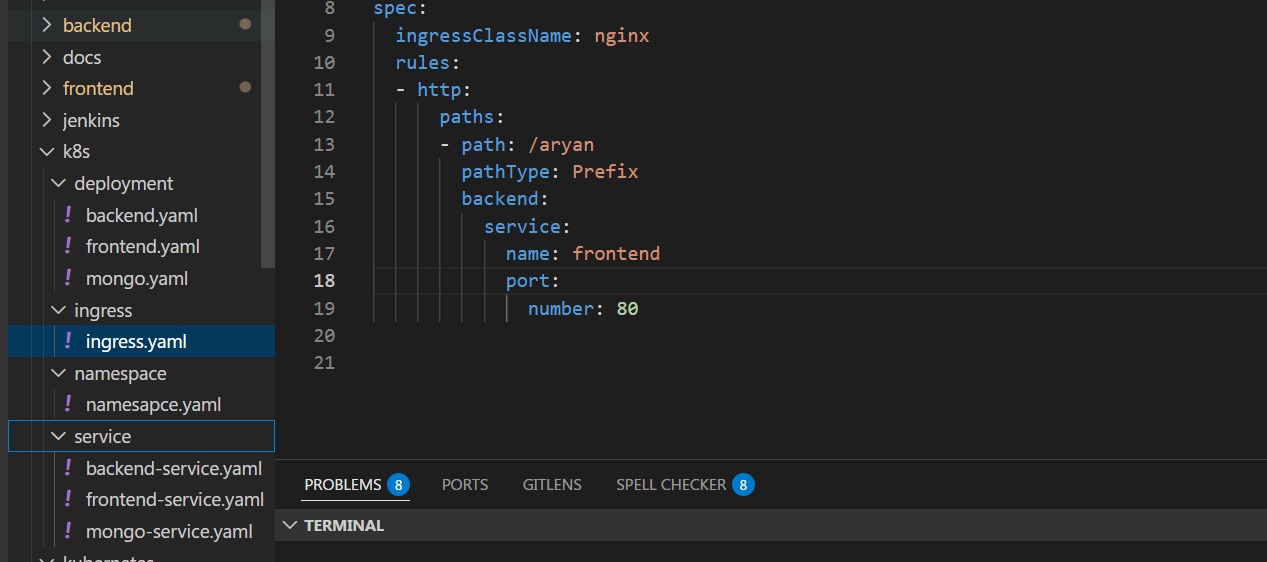
* **Tag each images from local**

****

* **Push it to DockerHub**

****

**Step 3: Container orchestration - Create Kubernetes manifest files**

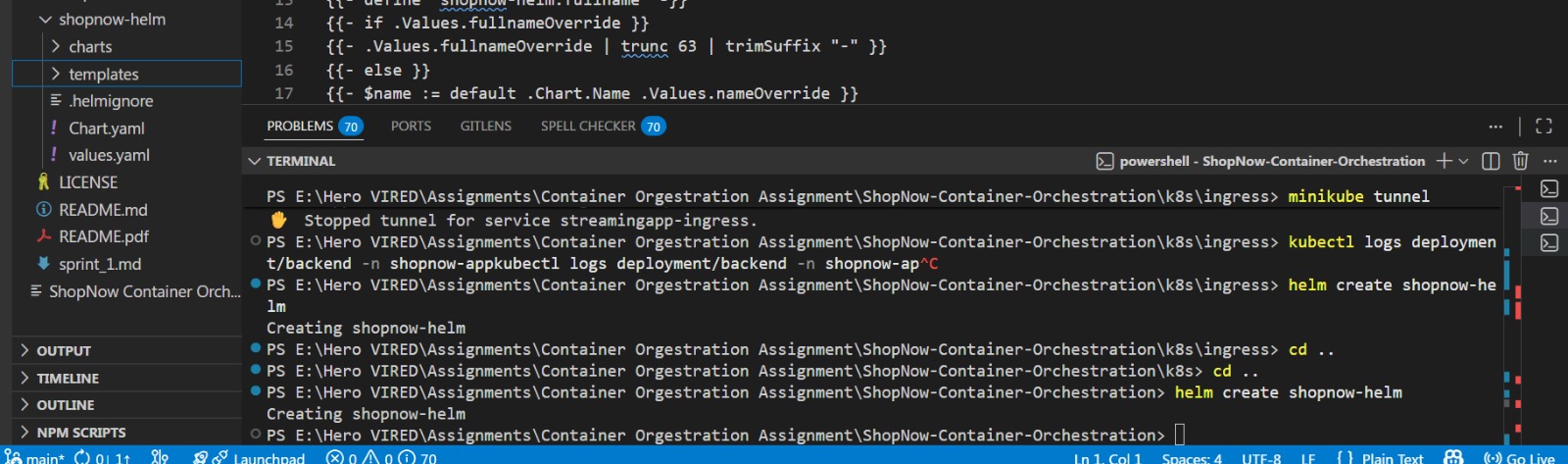
****

* **Now app is running in Minikube from Kubernetes – We can Access it via minikube tunnel**

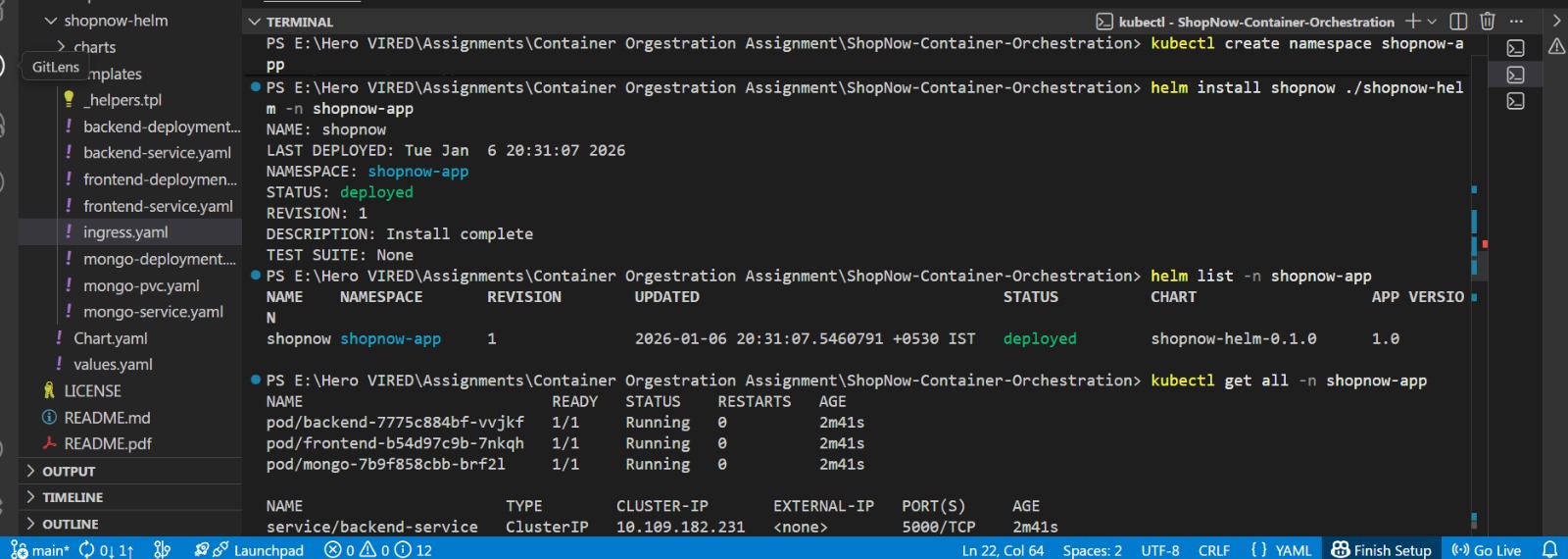
****

**Step 4: Create Helm and EKS**

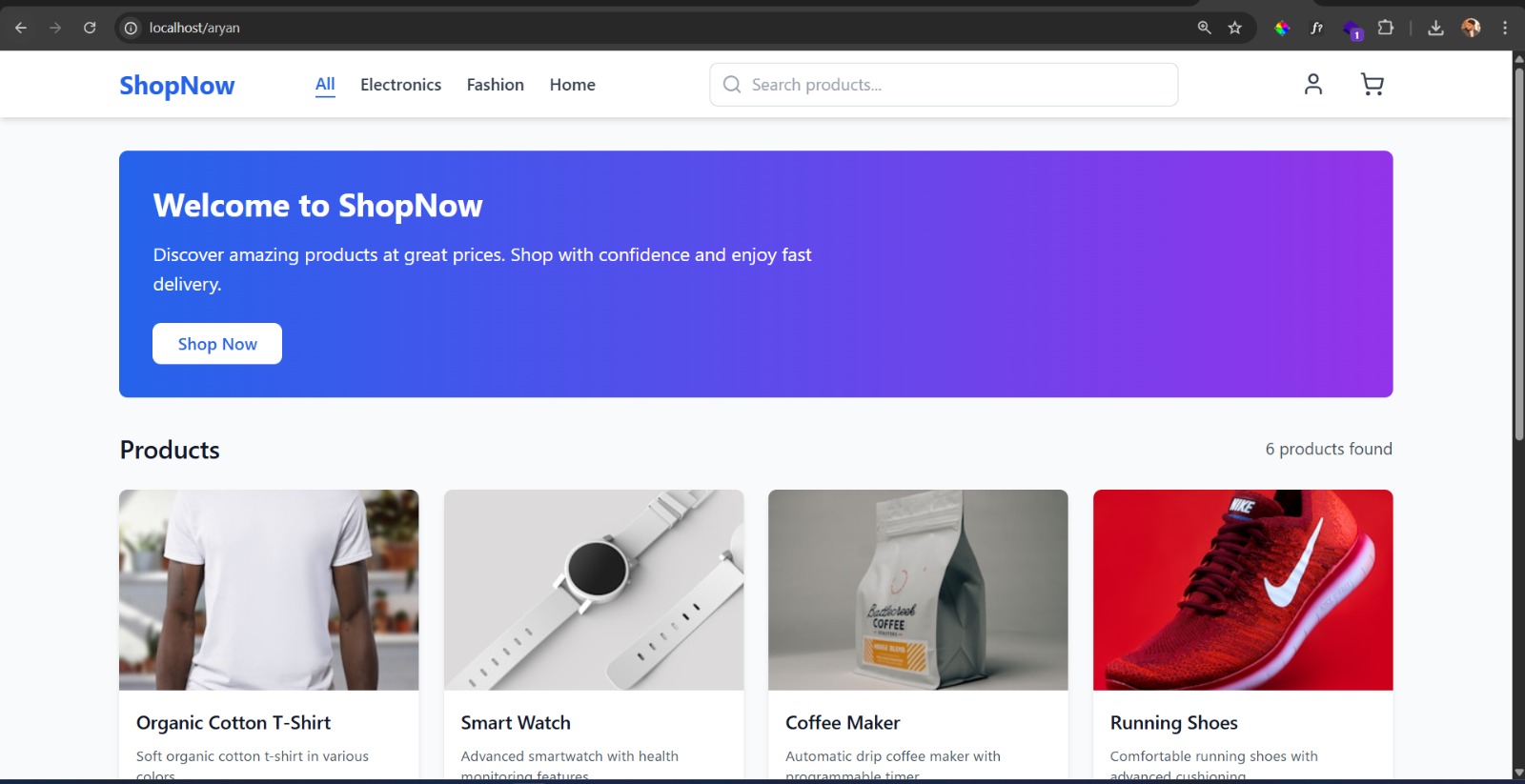
* **Create Helm folder inside your project root**

****

* **Install and deploy helm to create resources – Containers and services are running now**
* **Make sure your delete all the resources that are created before for testing k8s manifest files because helm will create everything fresh again.**

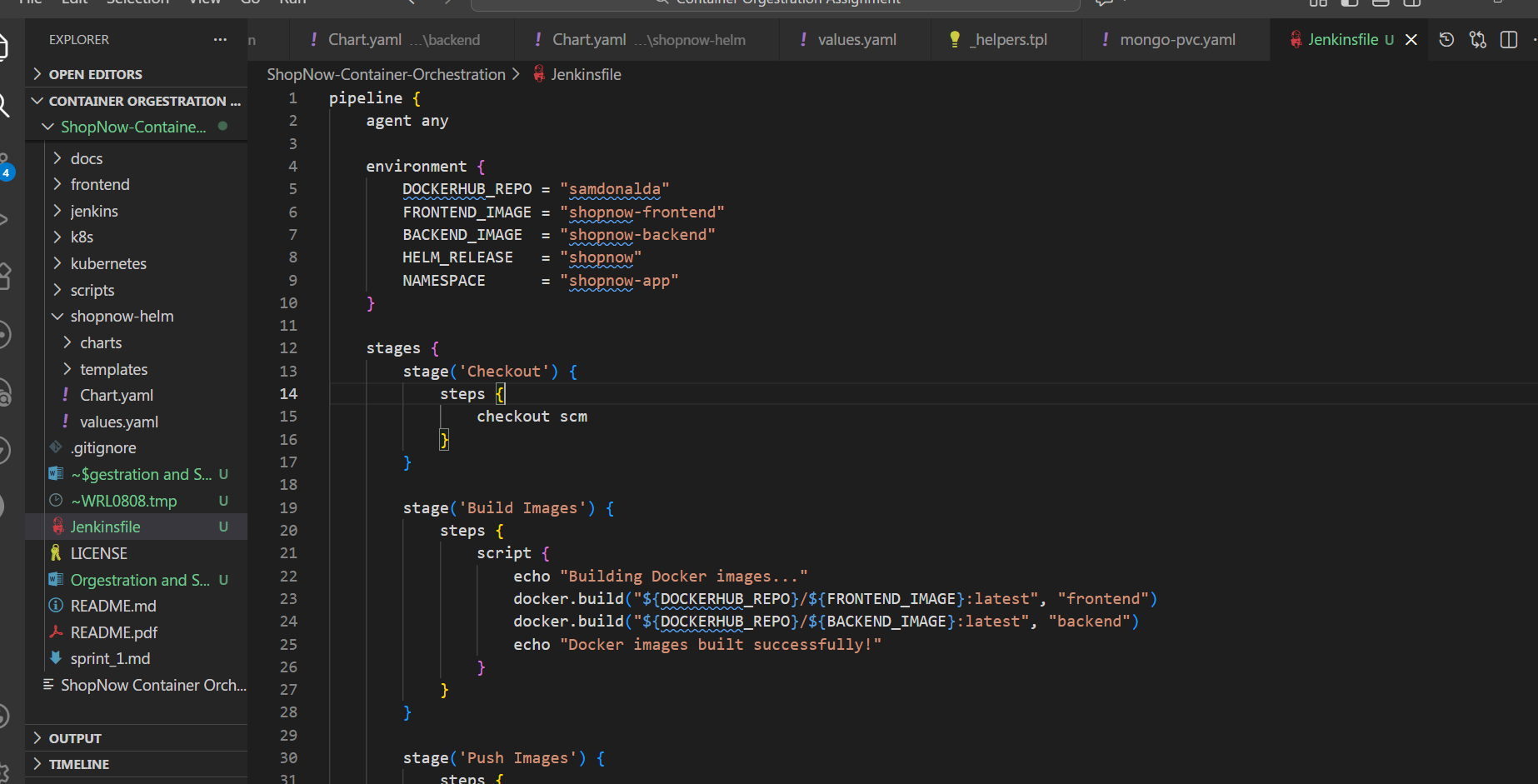
****

* **Resources are created and app deployed using helm chart**

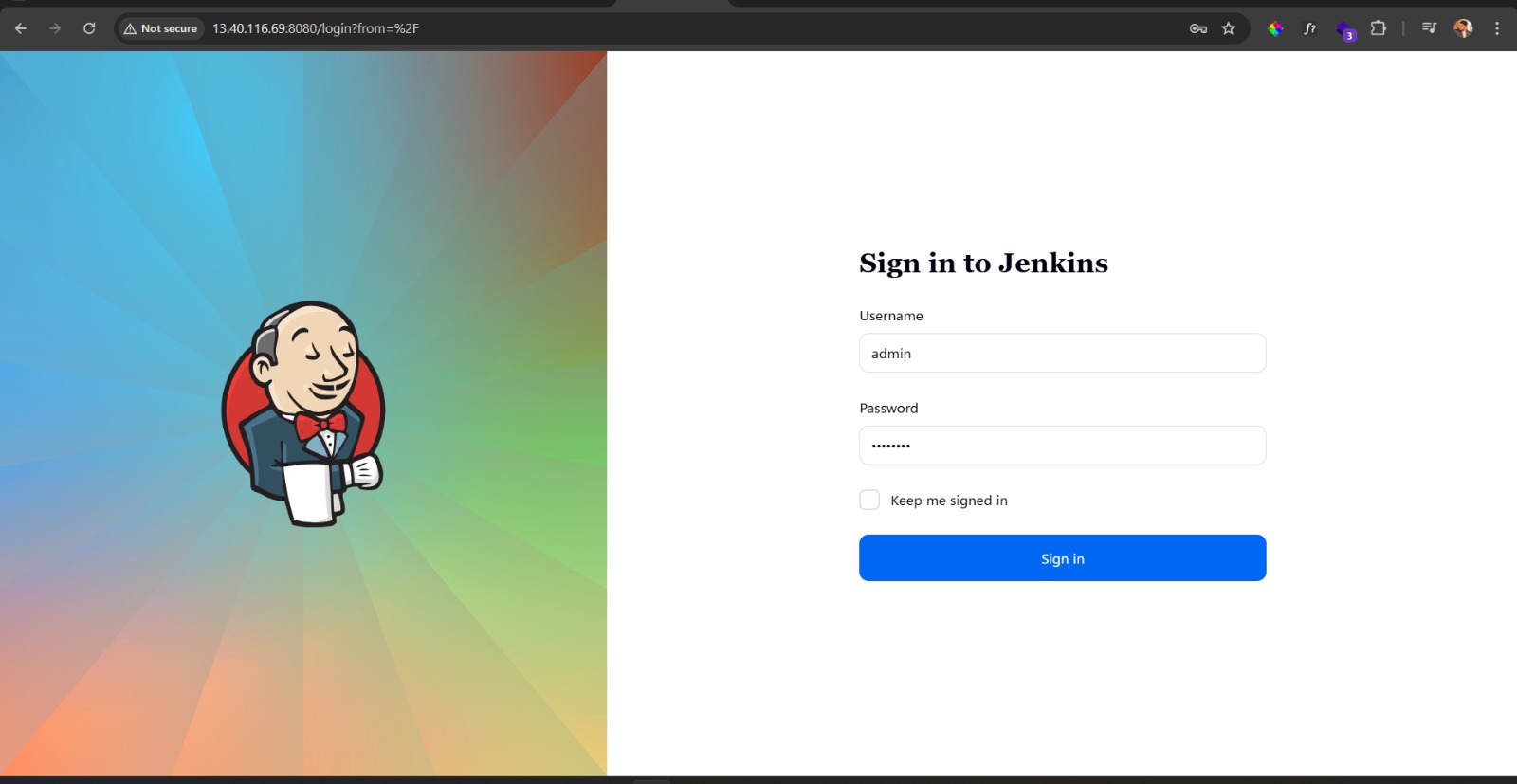
****

**Step 5: Jenkinsfile and Jenkin setup**

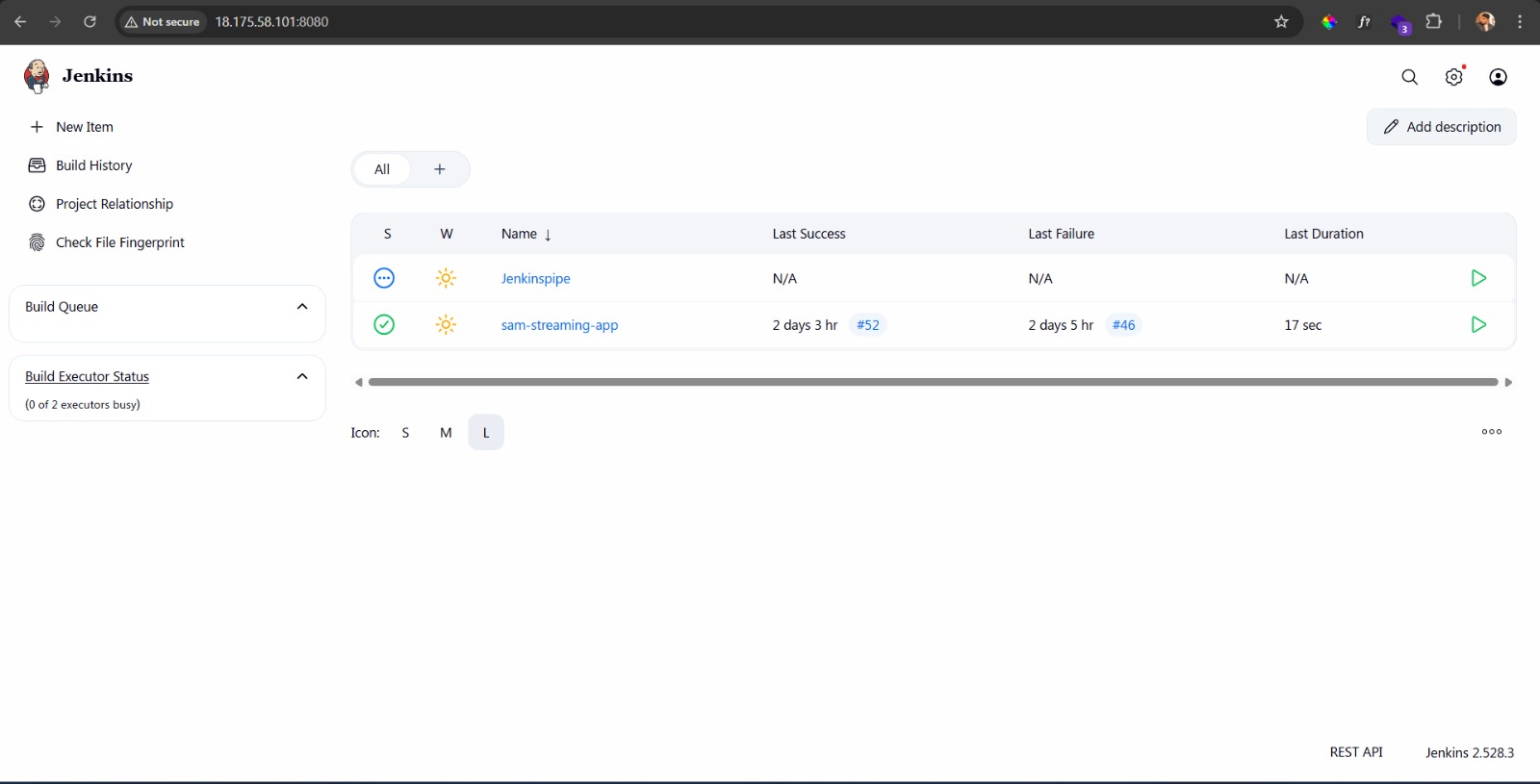
* **Create Jenkins file**

****

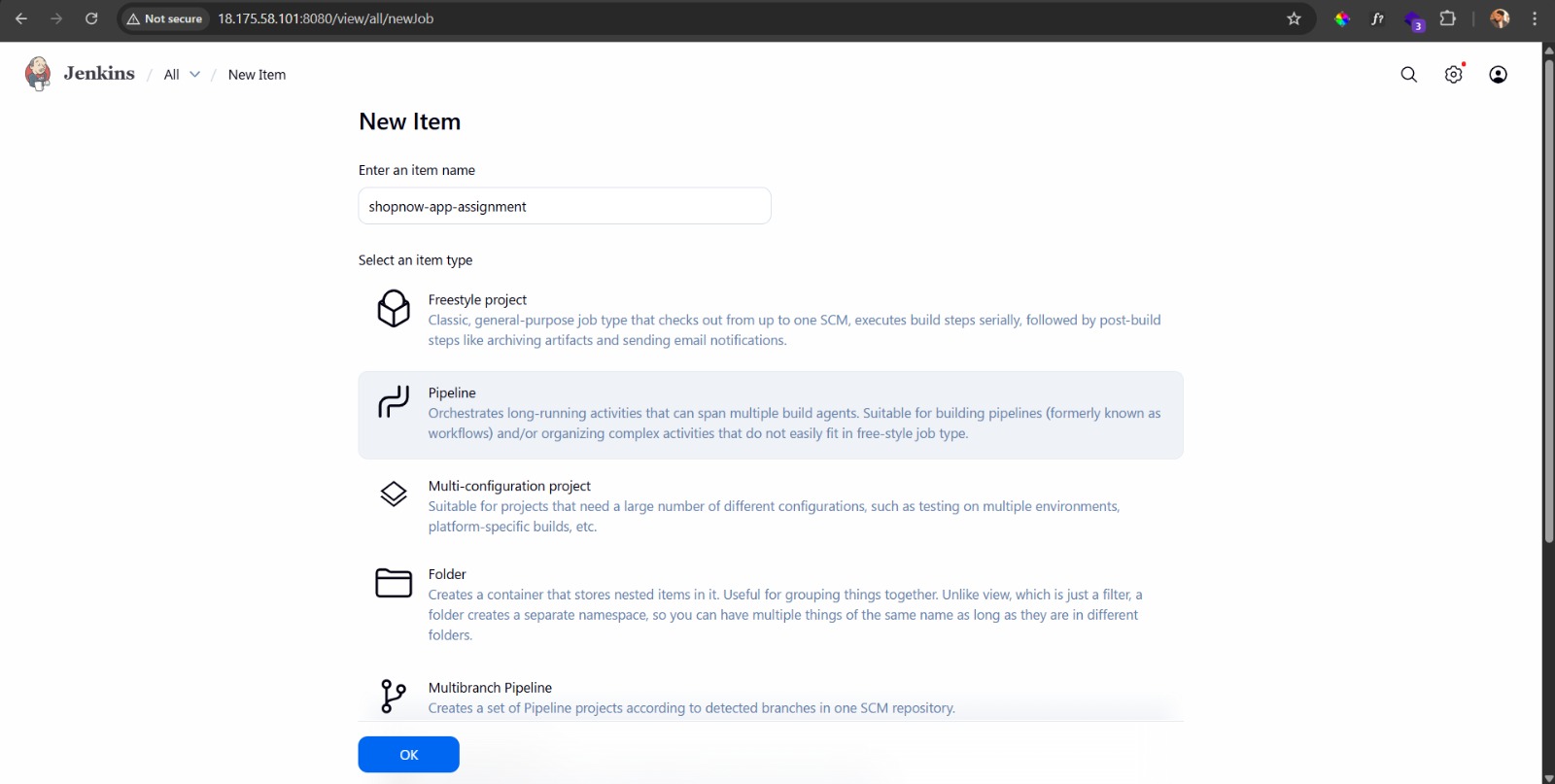
* **Create Jenkins server in the EC2 and open it in the browser via IP address and login**

****

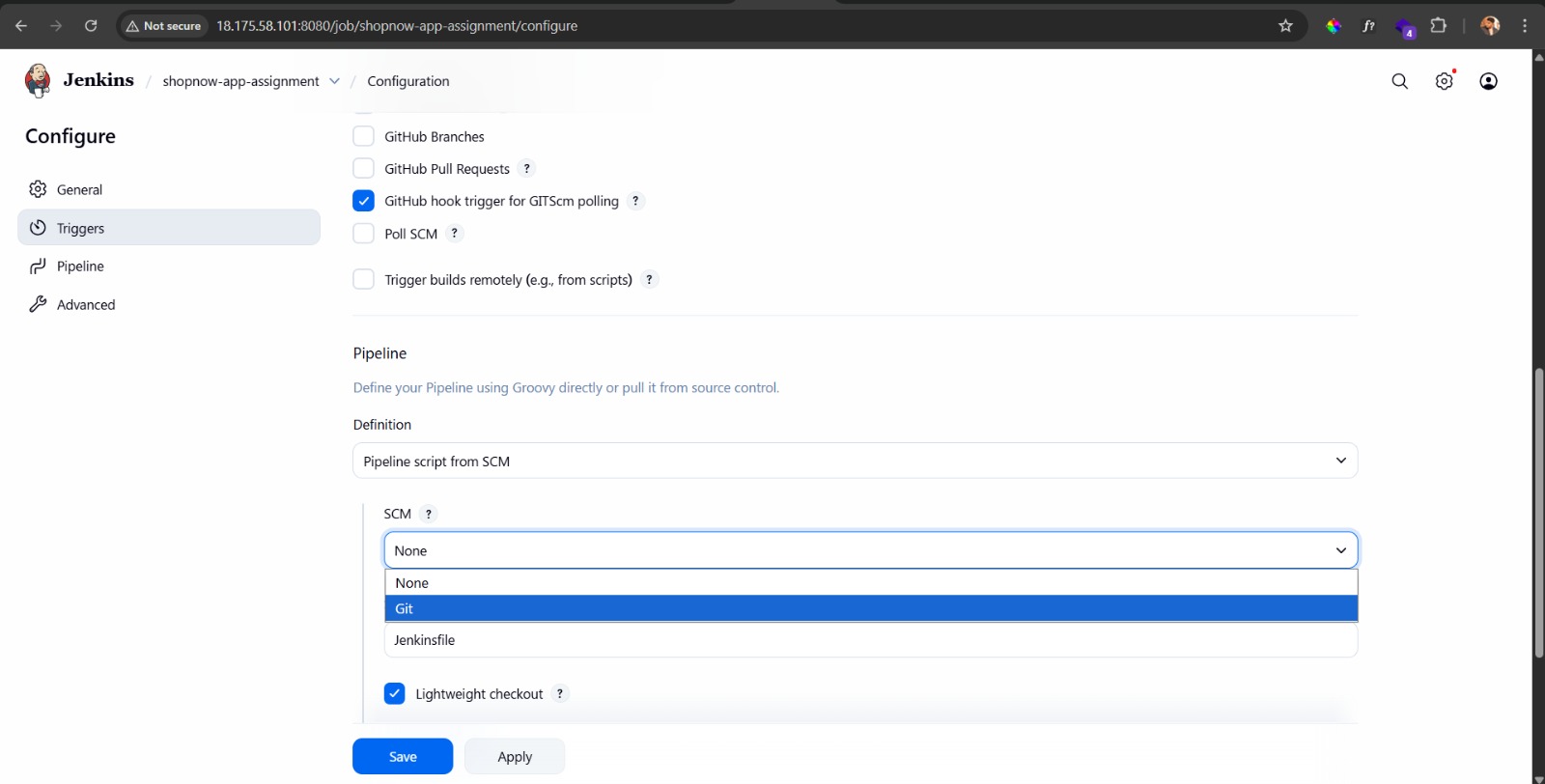
* **Click create New Item on the top left**

****

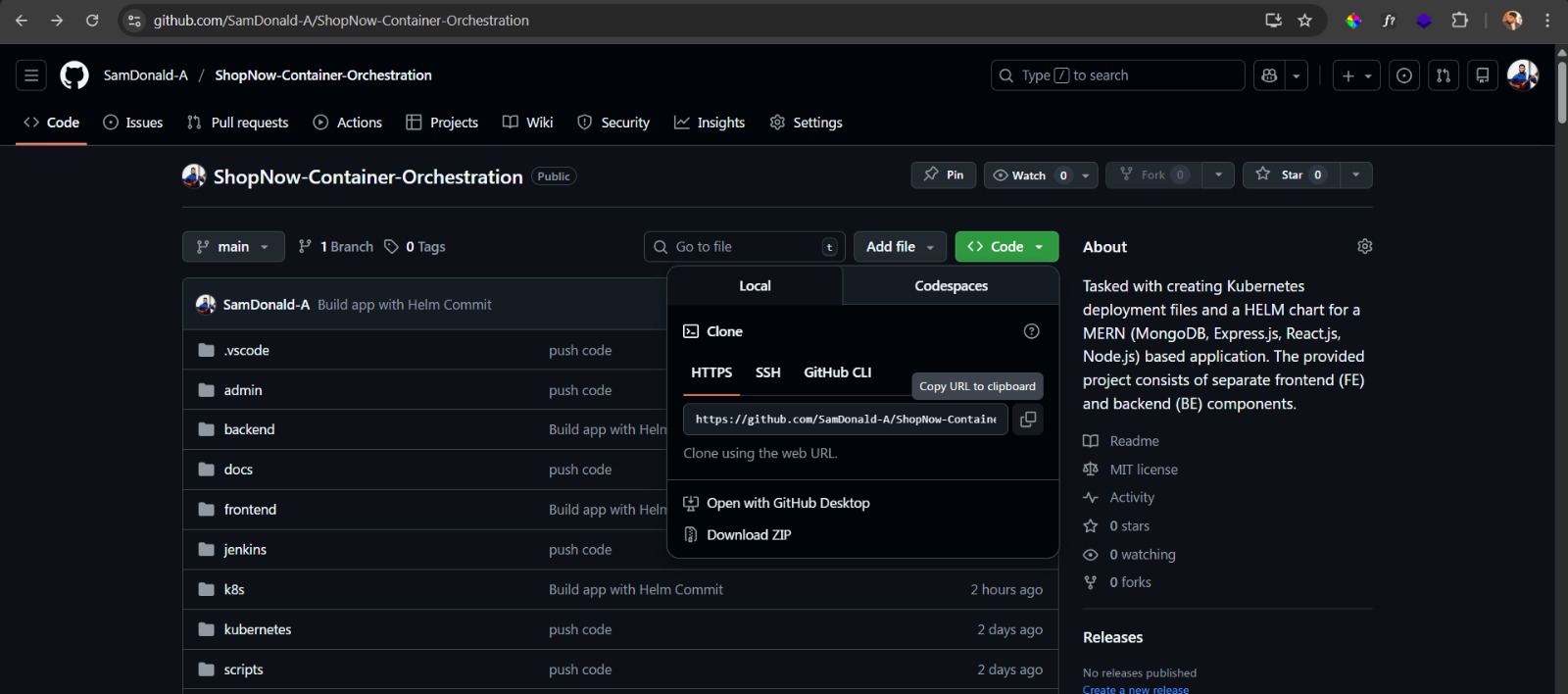
* **Give name & Select Pipeline**

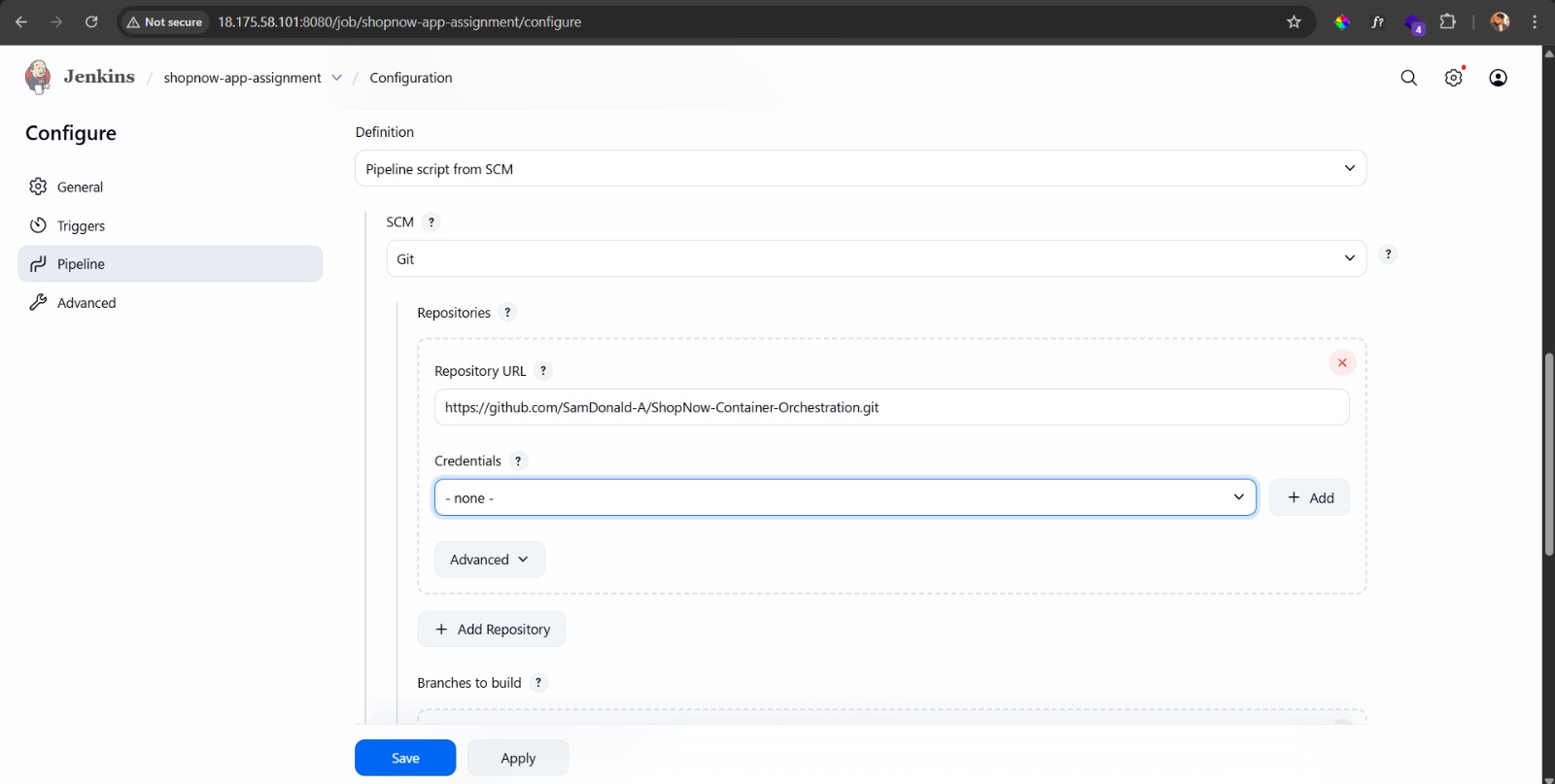
****

* **Selcte GitHub Hook trigger for GITScm polling (We need to setup webhook in the git repo)**

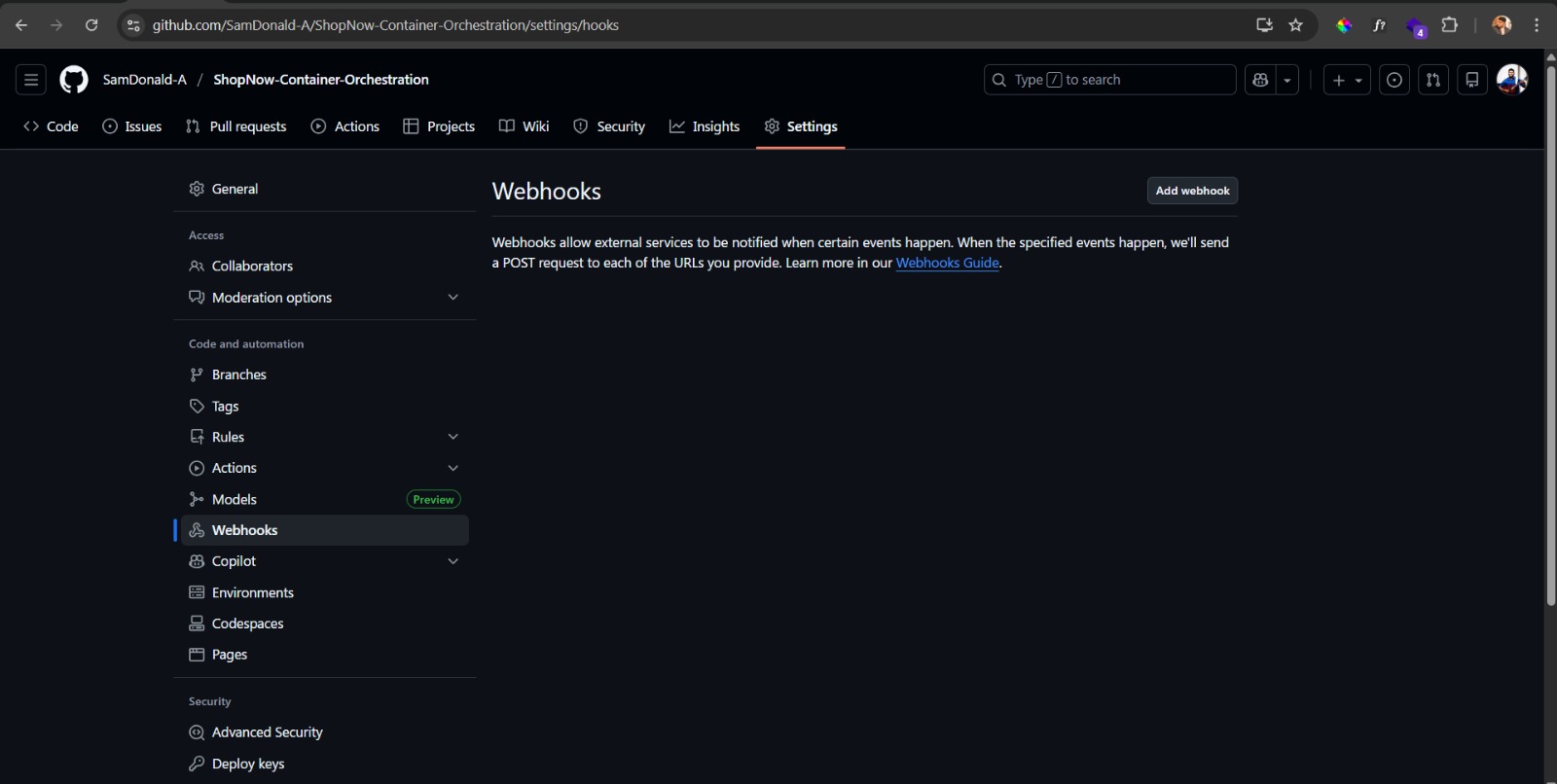
****

* **Provide that Git repo link here and select your branch**

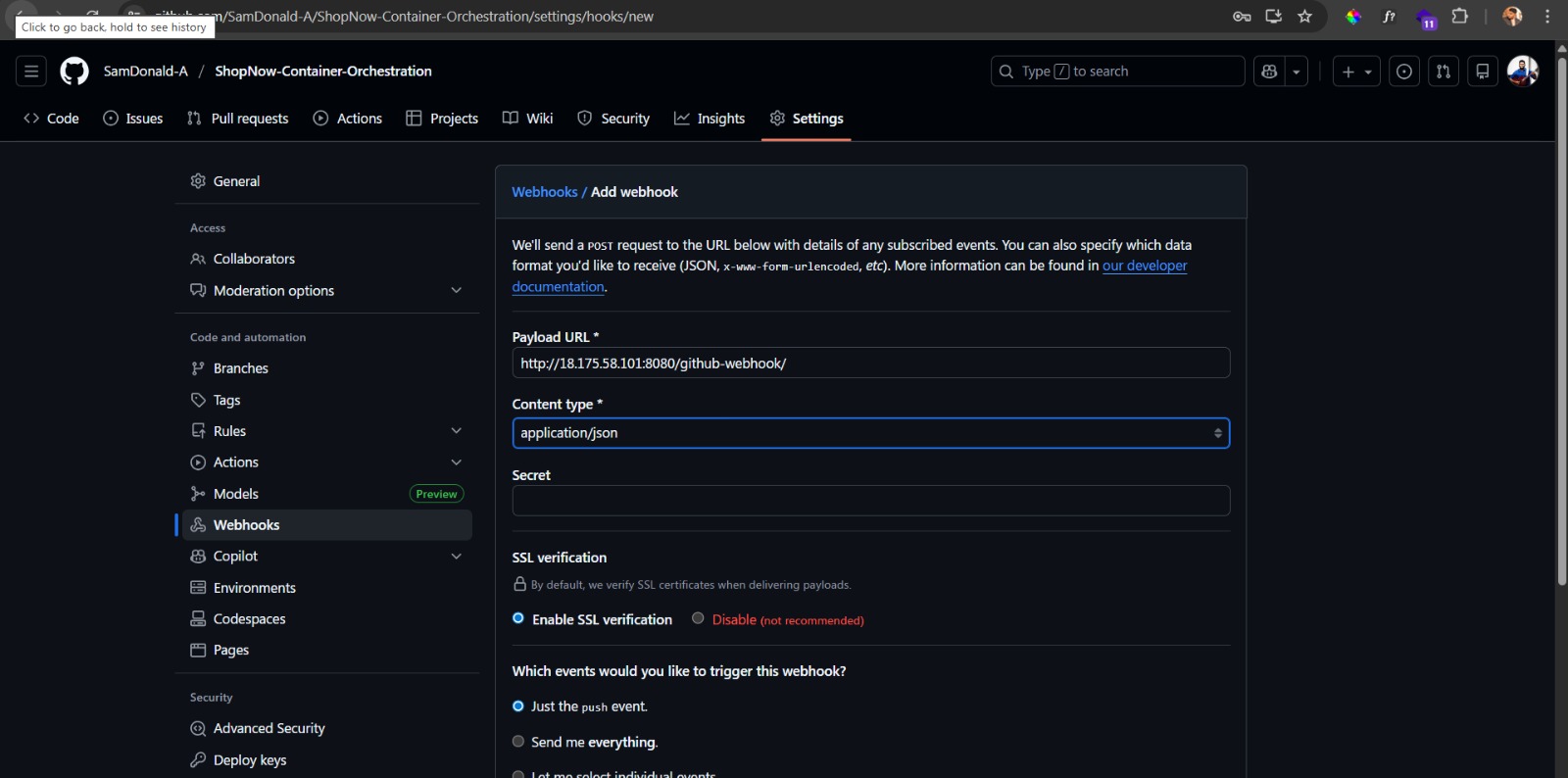
****

****

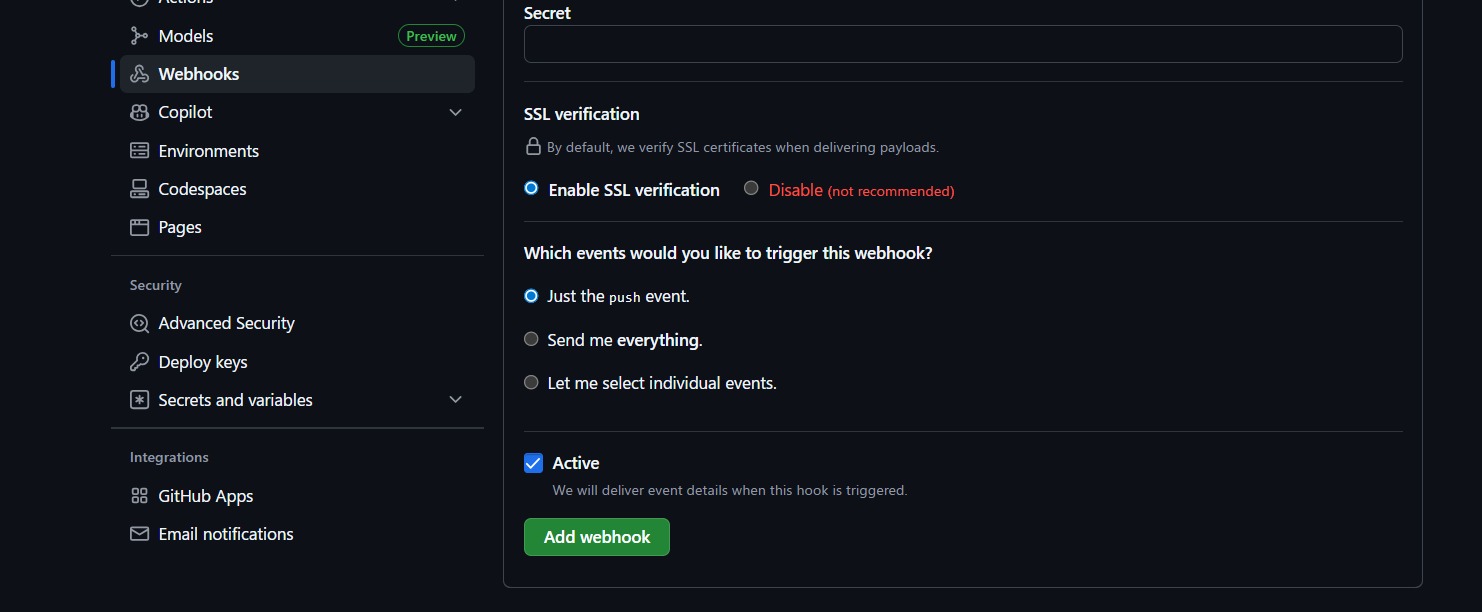
* **Setup Webhook in Github Repository – Goto settings of your repo and click webhook**

****

* **Click add New then add Payload and select Json**

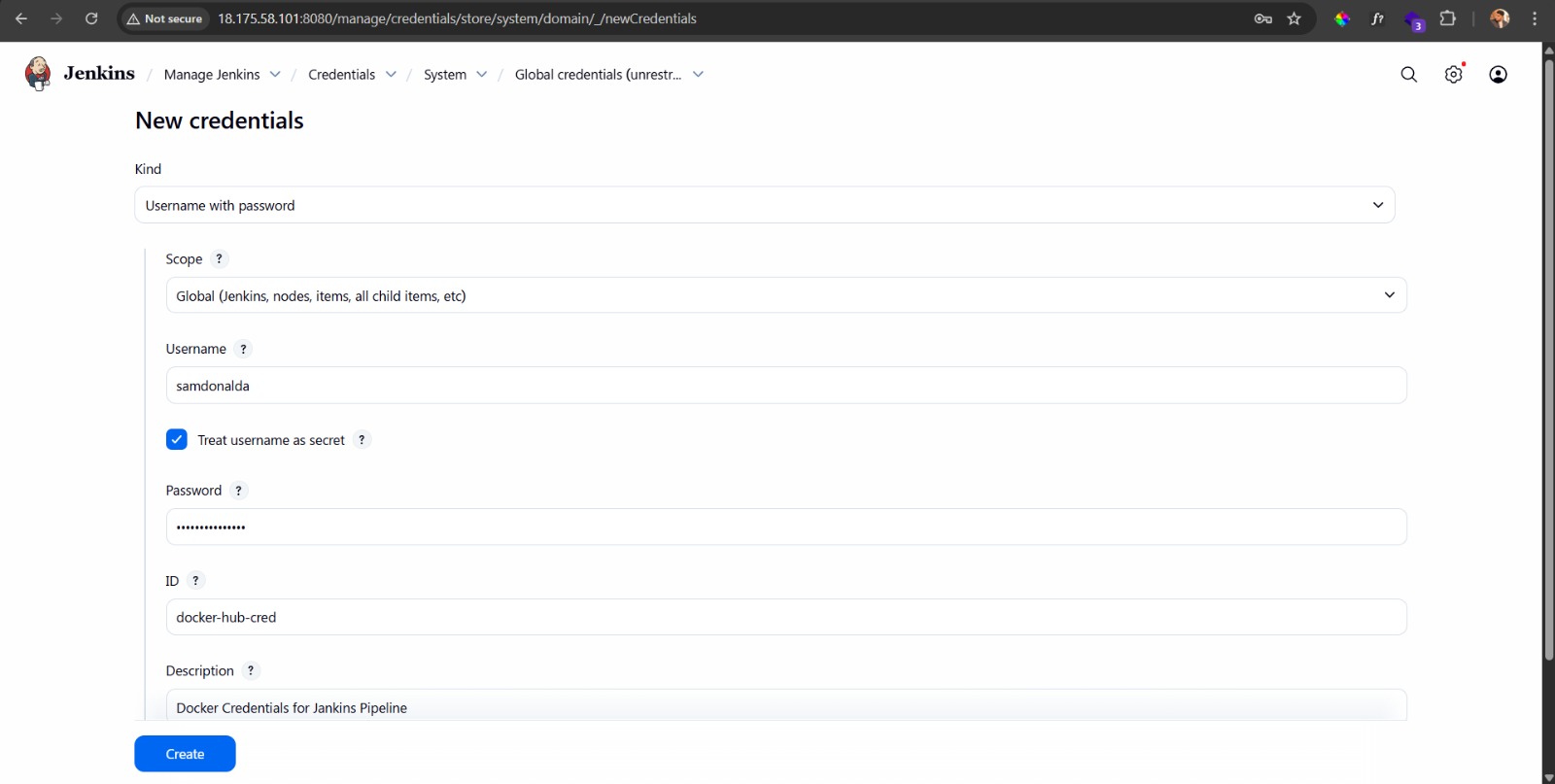
****

* **Click Add webhook**

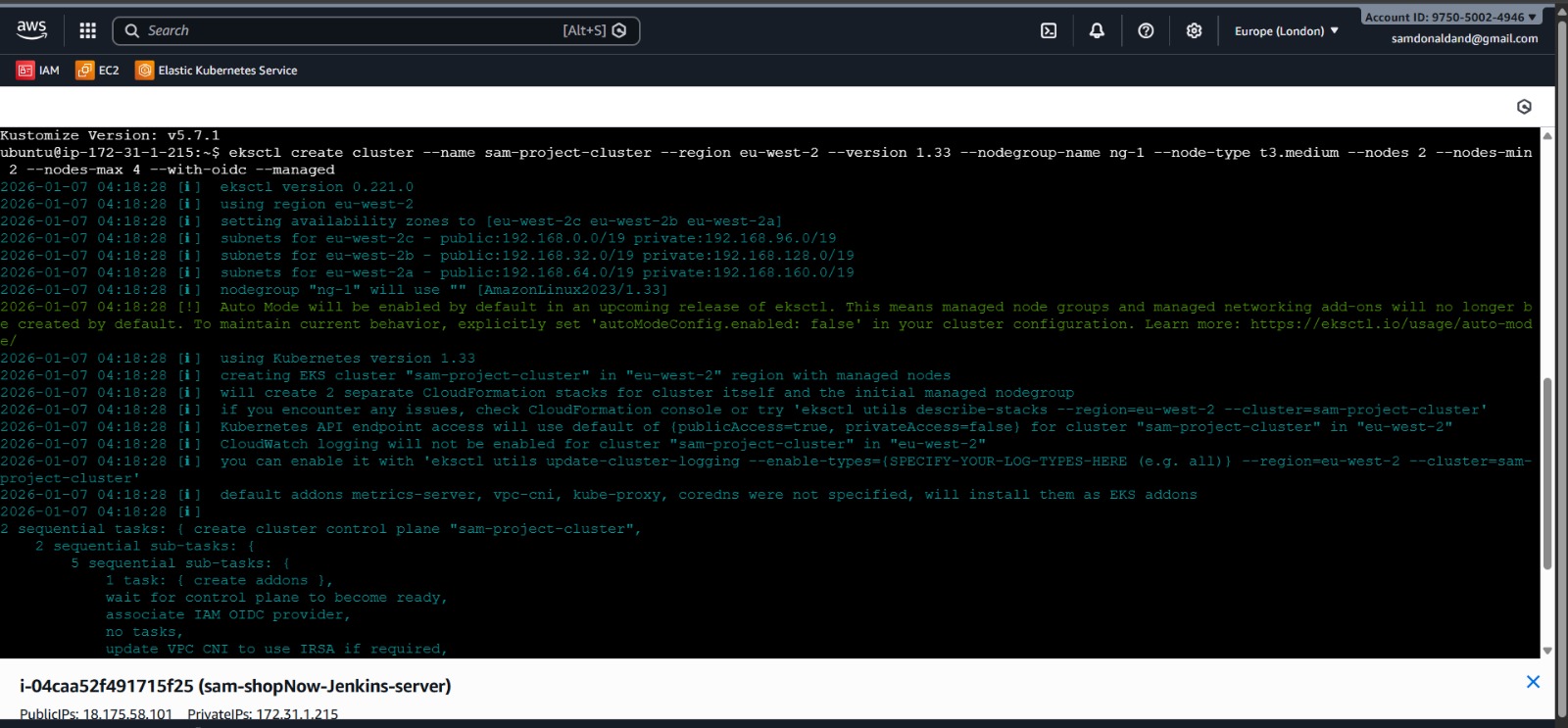
****

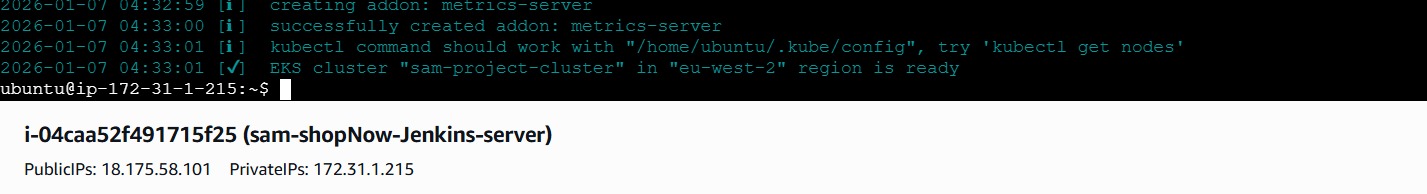
**Now we can push the and the Jenkin will be triggered**

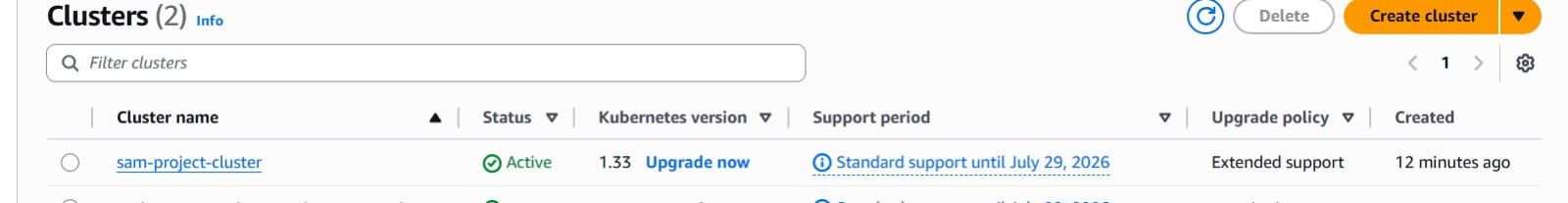
* **Now setup the Docker Credentials in the Jenkins Global secret**

****

* **Create EKS Cluster**

****

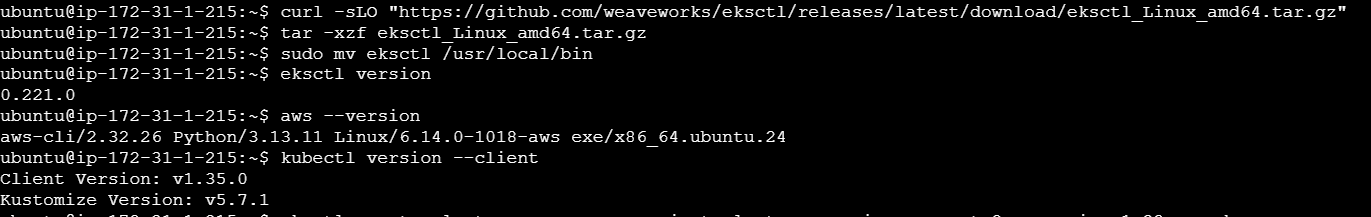
****

****

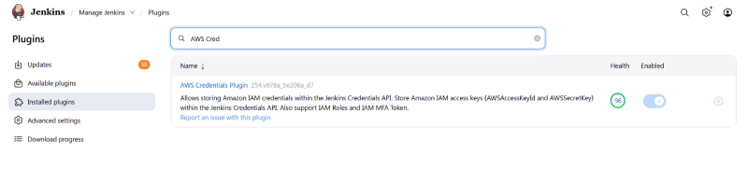
**Step 6: Check Jenkins Host server requirements for EKS to run the app**

***Make sure all the services are installed***

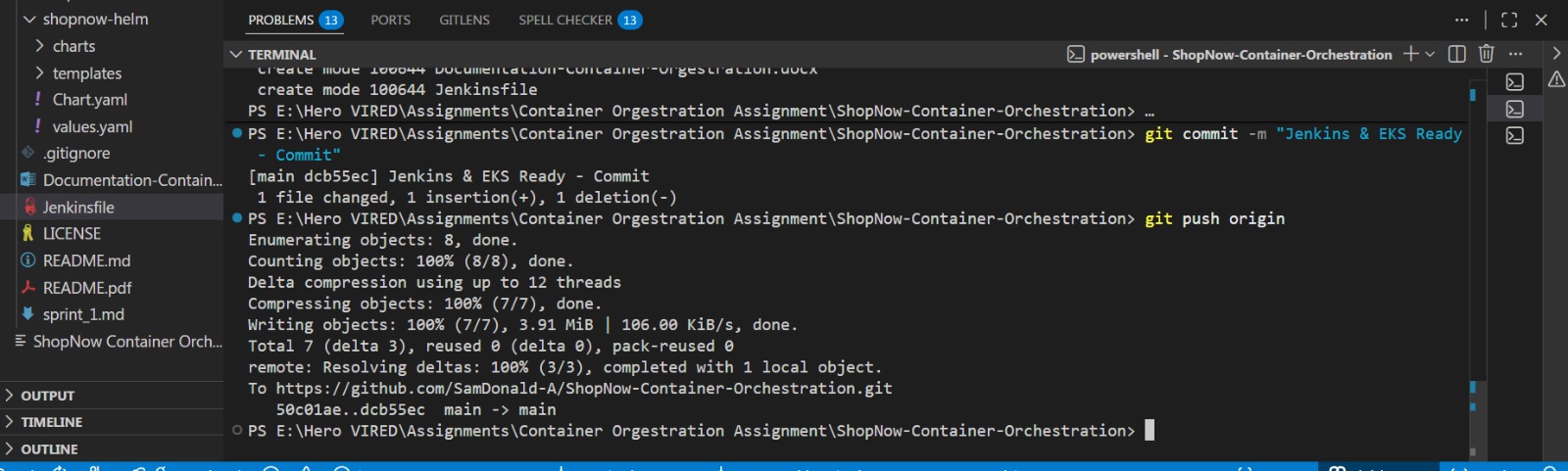
* **Docker**
* **Helm**
* **Aws CLI**
* **kubectl**
* **At least t3.midium in EC2 for running the pipeline**
* **At least 20gb storage for the npm and other installation process on the machine**
* **EKS IAM Role permissions**
* **Jenkins**
* **Jenkins credentials**
* **Jenkins plugins**
* **eksctl**

****

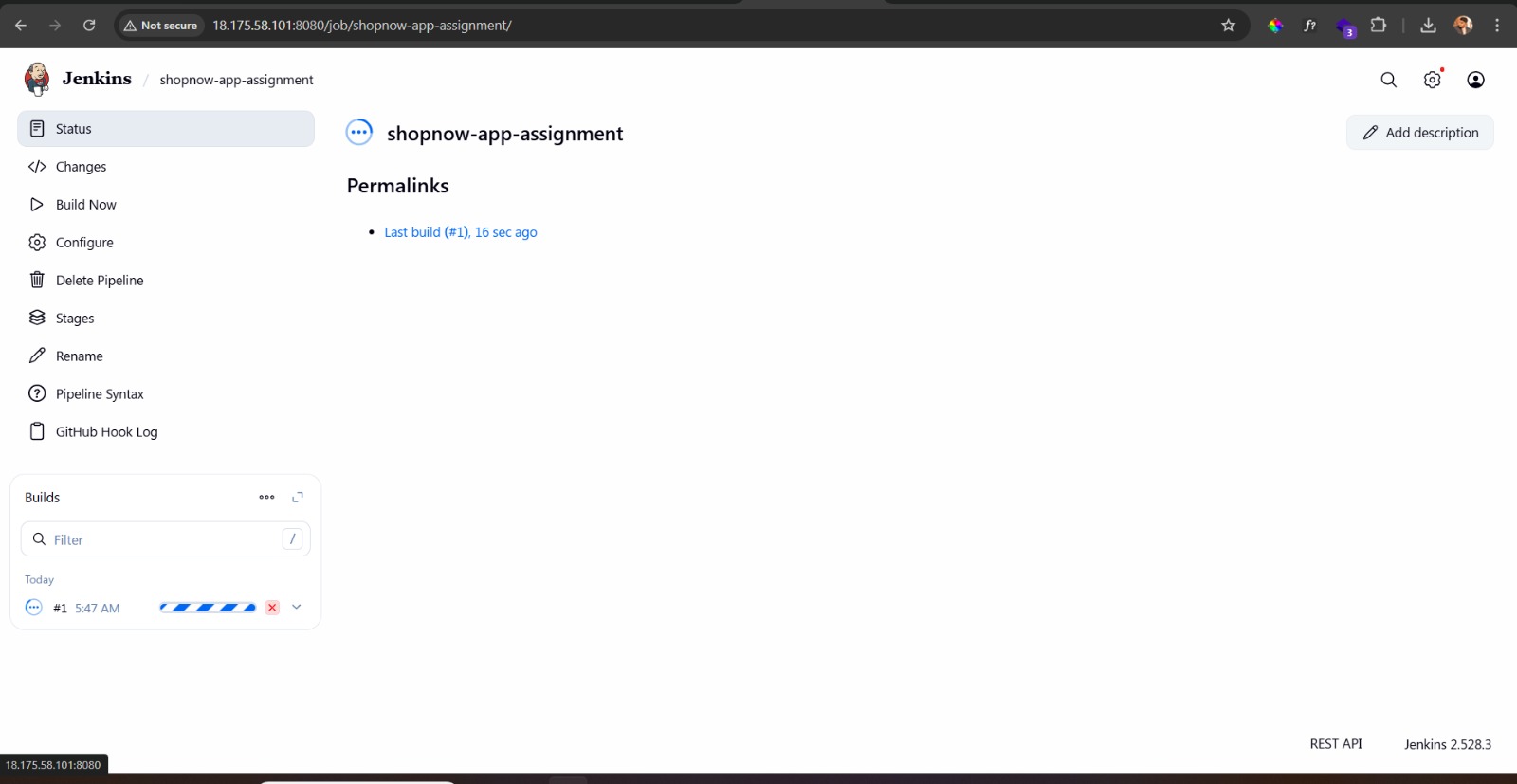
* **Also make sure all required plugins installed in Jenkins**

****

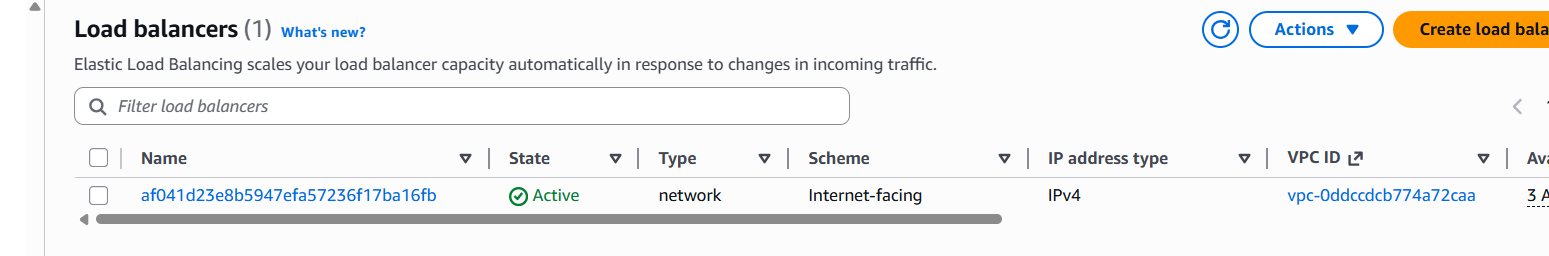
* **Then Change the Jenkins pipeline flow according to your requirements and push the code to the repository**

****

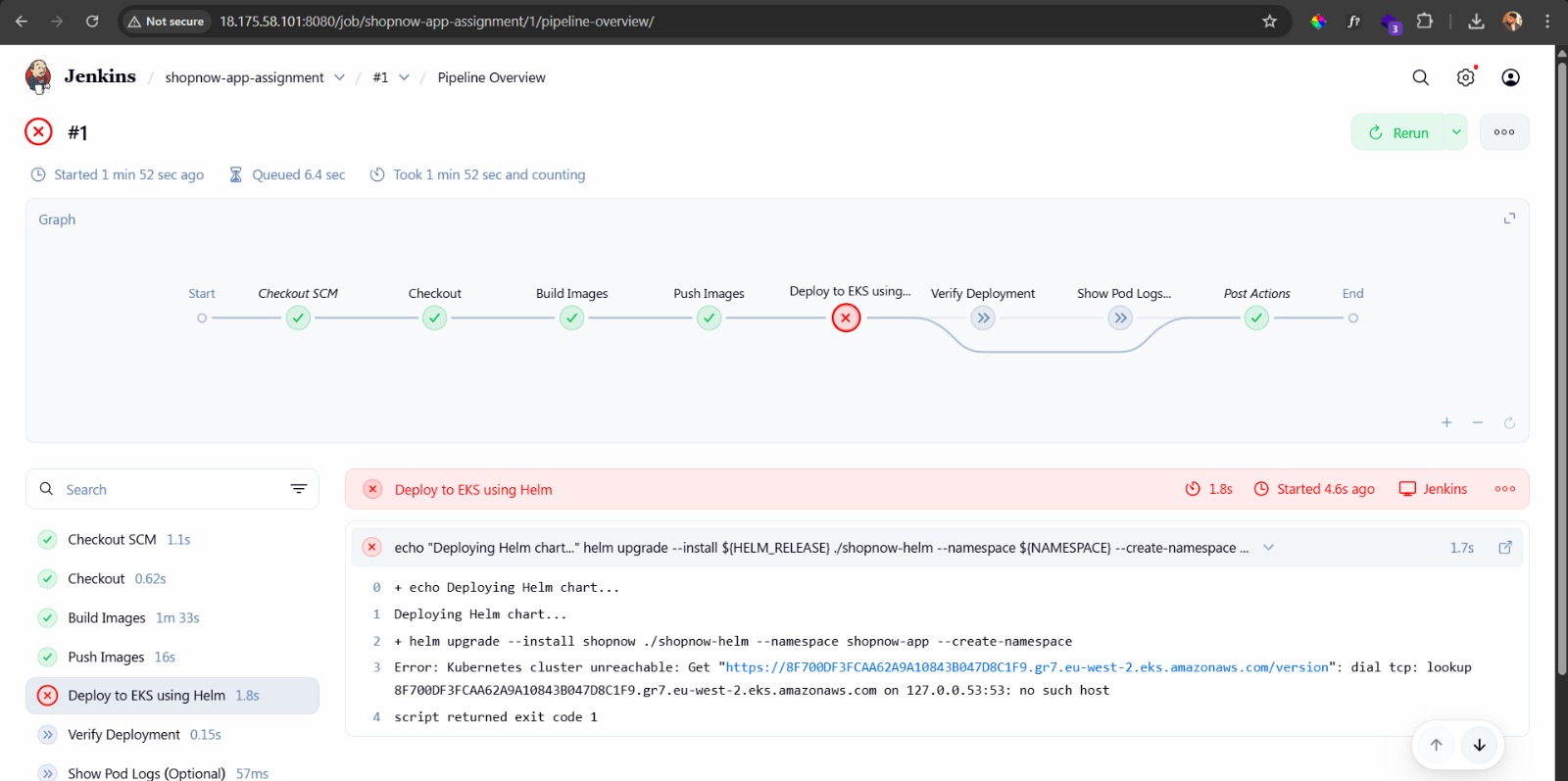
* **Pipeline triggered**

****

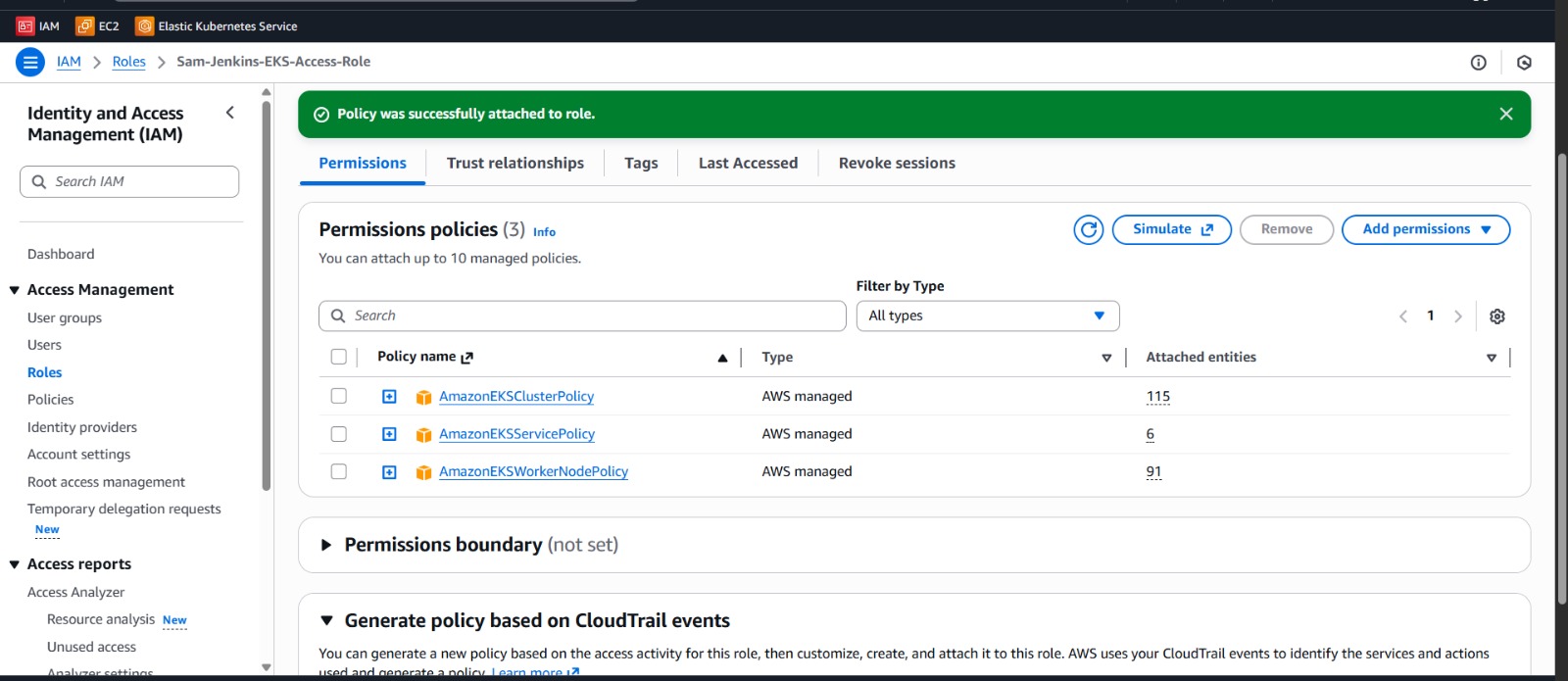
* **Loadbalancer also created**

****

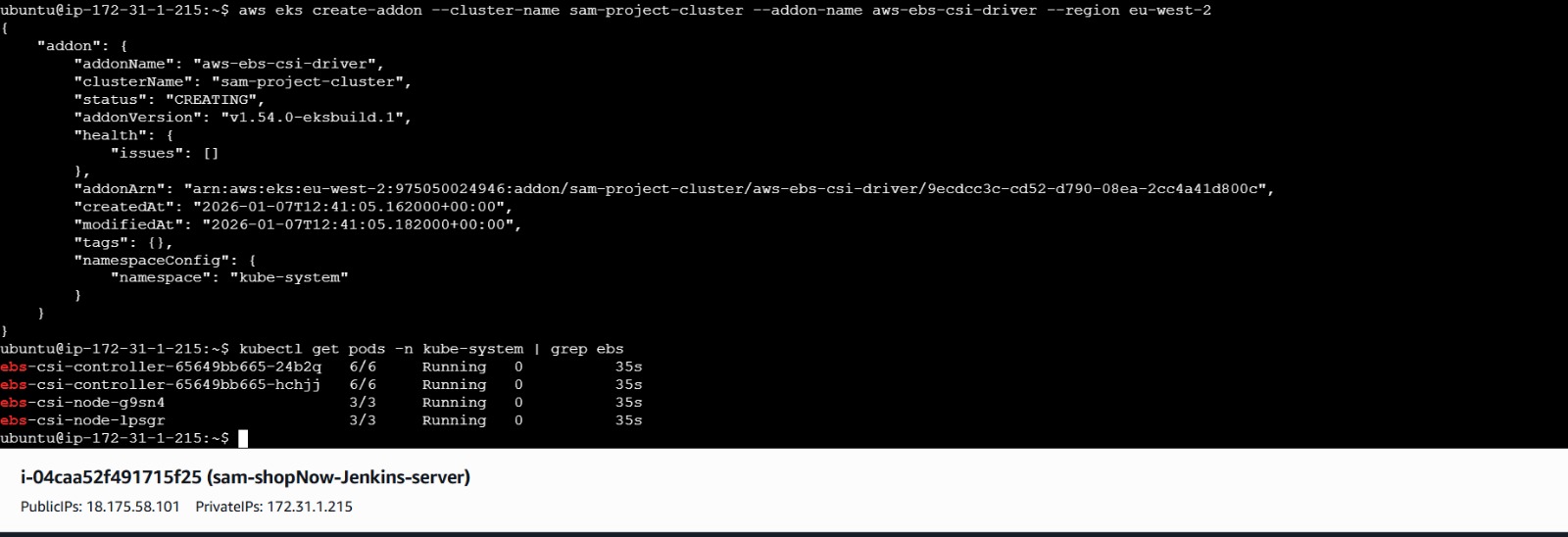
* **Study the log on the pipeline failure and Fix**

****

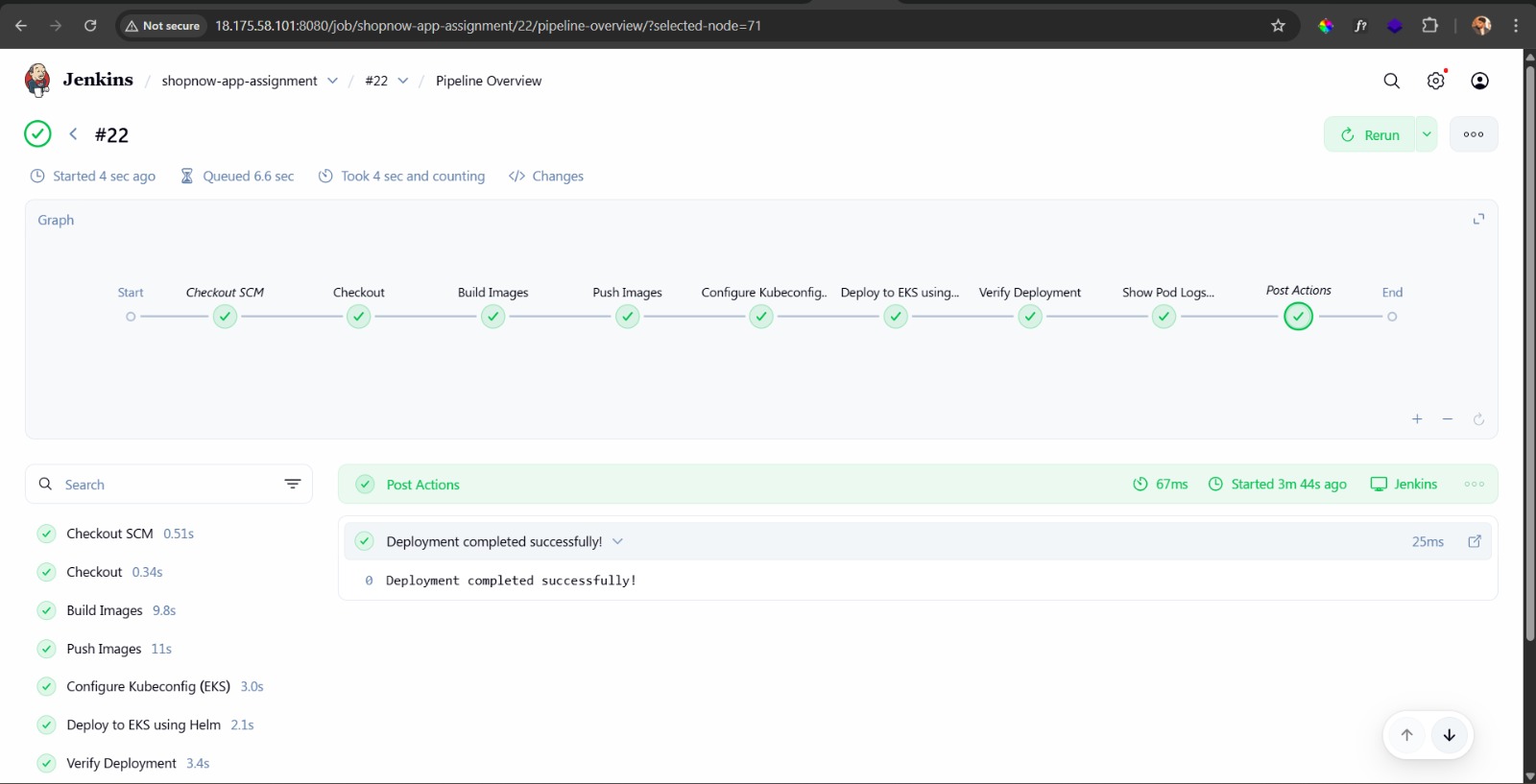
* **Make sure you have this role in your EC2 IAM of Jenkins Server**

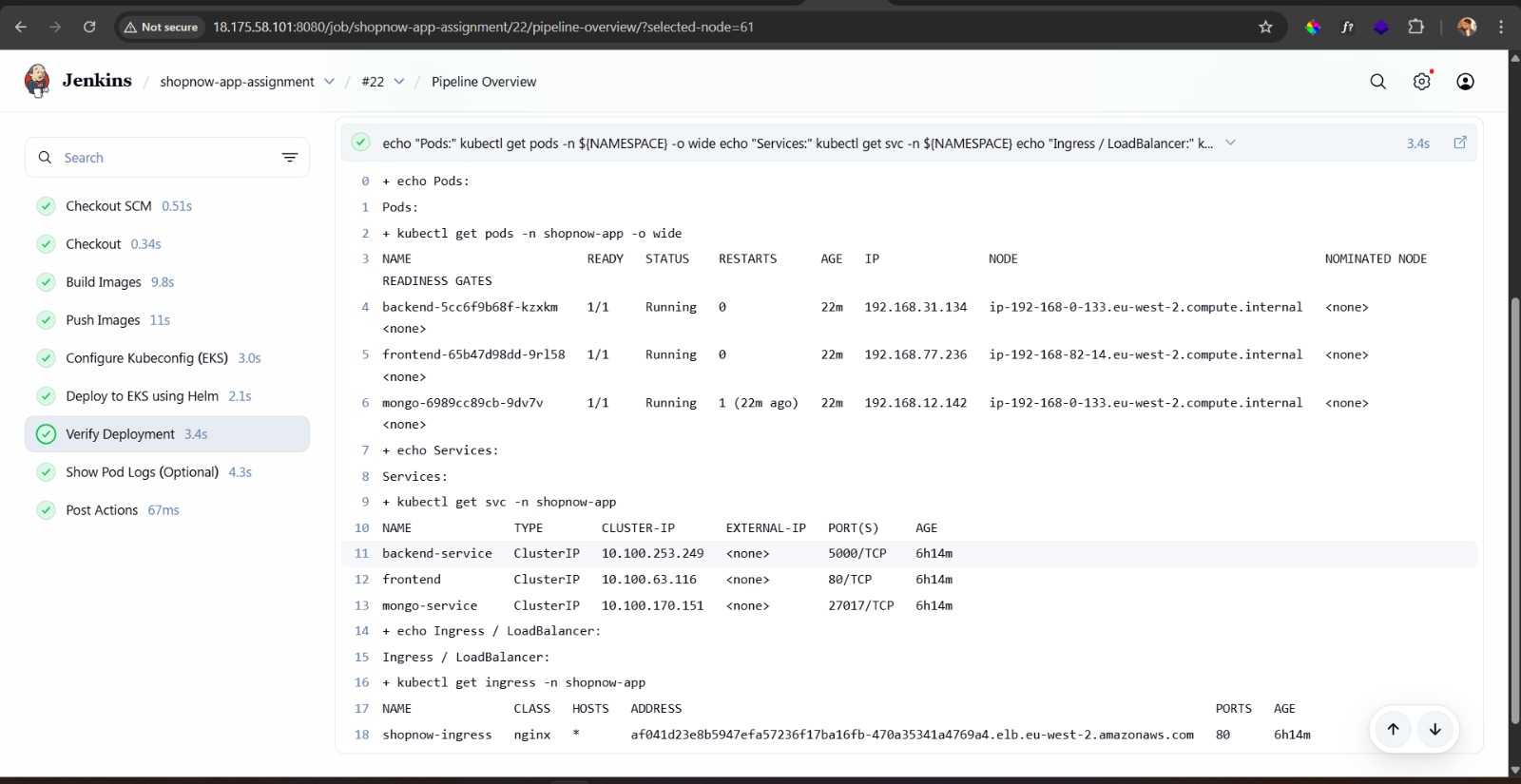
****

* **Create and Attach EBS setup in your EC2 Jenkins Host since we need PVC for MongoDB (EKS always Need EBS if PVC required – In AWS PVC means EBS)**
* **And make sure your node group also has the IAM Role policy to access EBS, So that It can create and attach EBS-CSI Volume on the deployment**

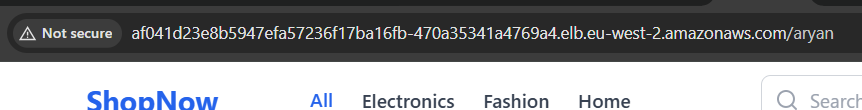
****

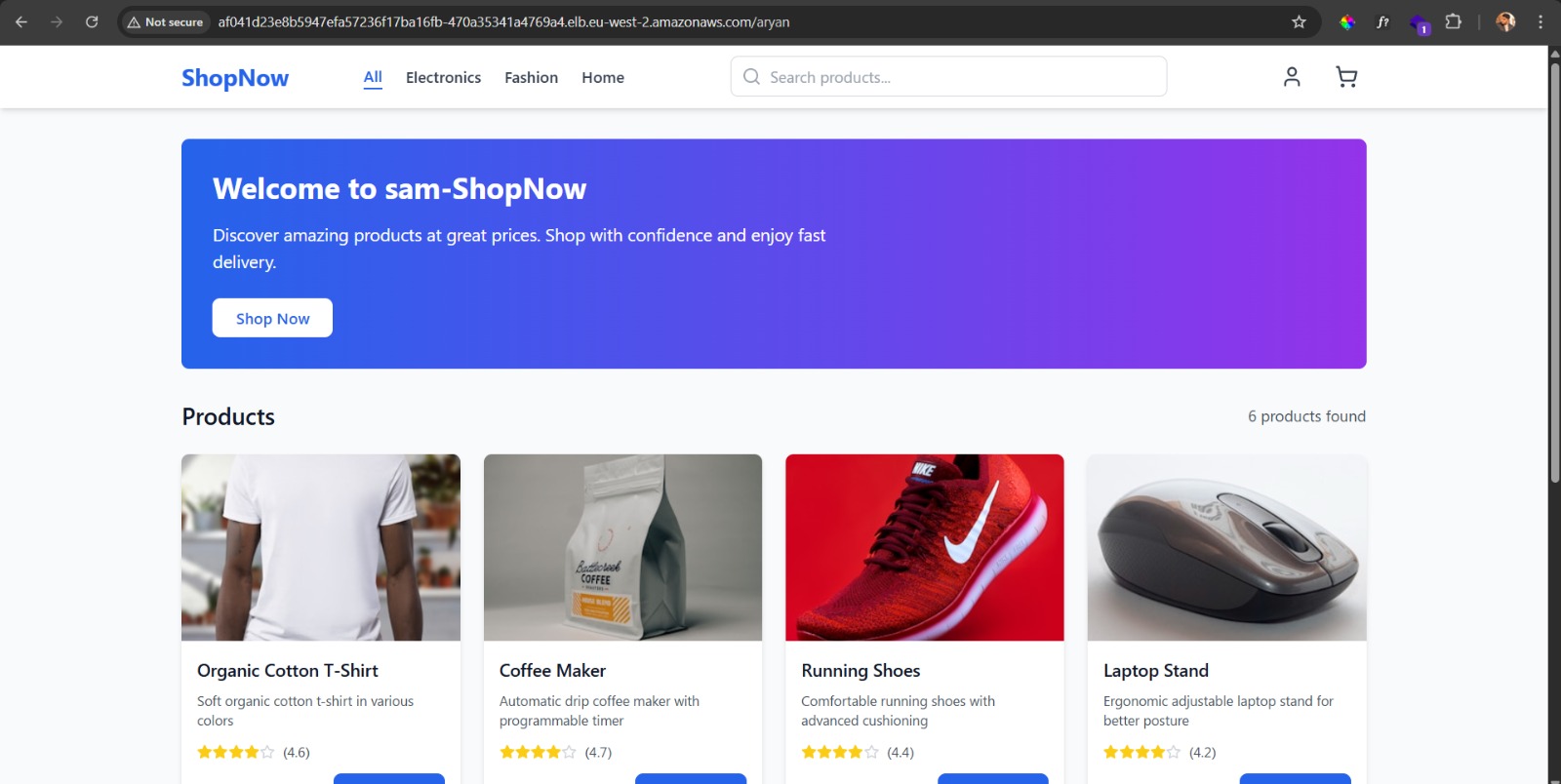
* **After debugging, push the code and check the pipeline then Check All Pods and services are running on the success**

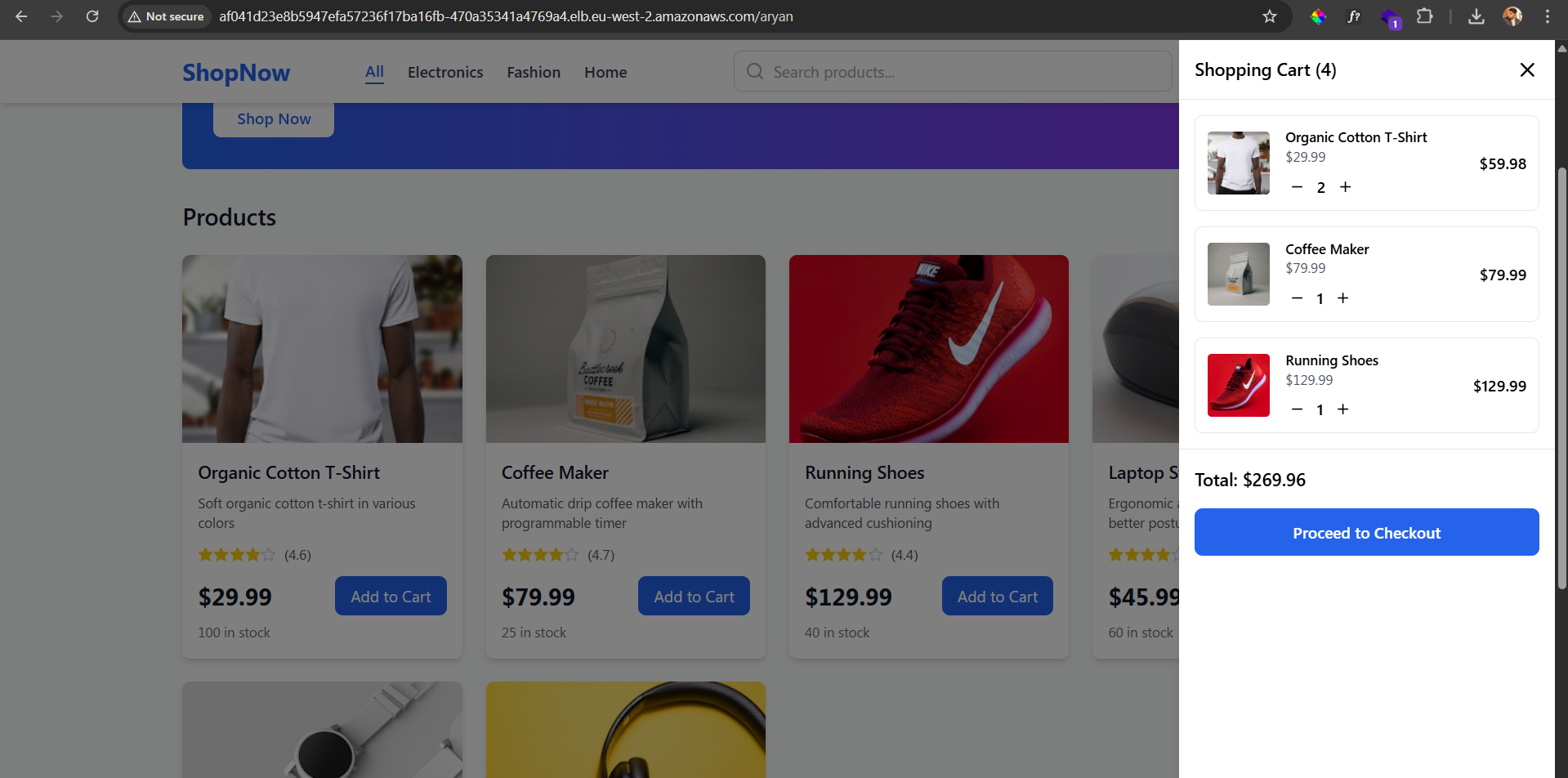
****

****

* **Check the Load balancer on the browser – /aryan after ELB is able to fetch products on the Landing page**

****

****

****

**App Successfully Deployed using Helm, Jenkins and AWS EKS Cluster**

Documentation by: Sam Donald A  
Email: [samdonaldand@gmail.com](mailto:samdonaldand@gmail.com)

GitHub: <https://github.com/SamDonald-A>

Website: [www.samdonald.in](https://www.samdonald.in/)