**Deployment Document**

**Group –05**

**Learning Management System for Pre-Onboarding Freshers**

**Rishik Velpuri - 7388**

**Shaik Siddik - 7396**

**Steps to deploy the project:**

**1) Connect to the Instance:**

* We need to create an instance with t2.medium with the OS image – Ubuntu.
* While creating the instance, create a new key pair to login into the instance and save the key pair .pem file in the folder created in the computer.
* Connect to that instance and copy the SSH key from the SSH client.
* Open Git Bash in the folder where the .pem file is saved and paste the SSH key in the Git Bash to connect to the instance.

**2) Become the Root User:**

* sudo –i**.**

**3) Installation of the Docker:**

* Add Docker's official GPG key:

**sudo apt-get update**

**sudo apt-get install ca-certificates curl**

**sudo install -m 0755 -d /etc/apt/keyrings**

**sudo curl-fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc**

**sudo chmod a+r /etc/apt/keyrings/docker.asc**

* Add the repository to Apt sources:

**echo \**

**"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]**  **https://download.docker.com/linux/ubuntu \**

**$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \**

**sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

**sudo apt-get update**

**sudo apt-get install docker-ce docker-ce-cli containerd.io**

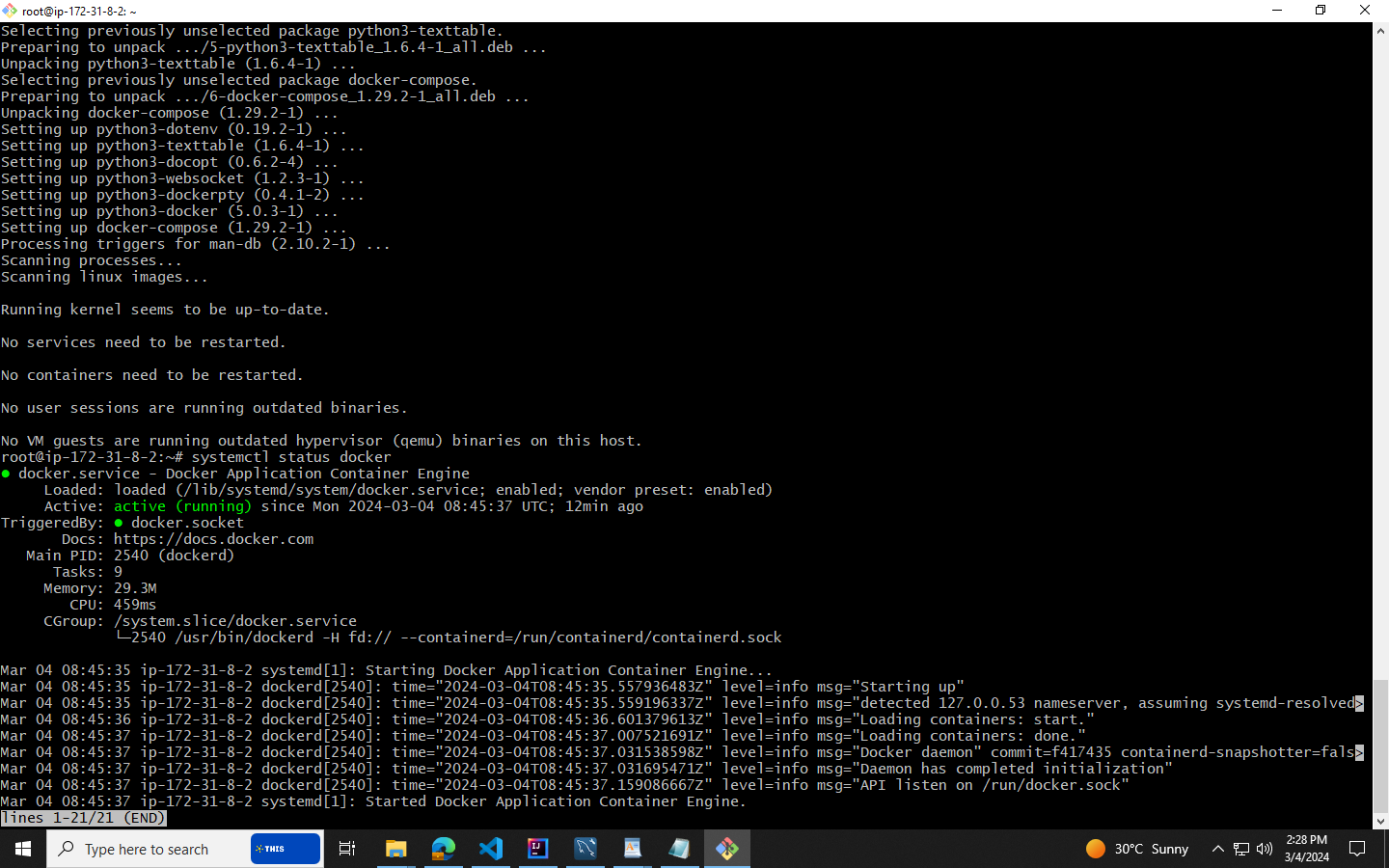
4) **Install Docker Compose:**

* apt install docker-compose

5) **Verify the docker status:**

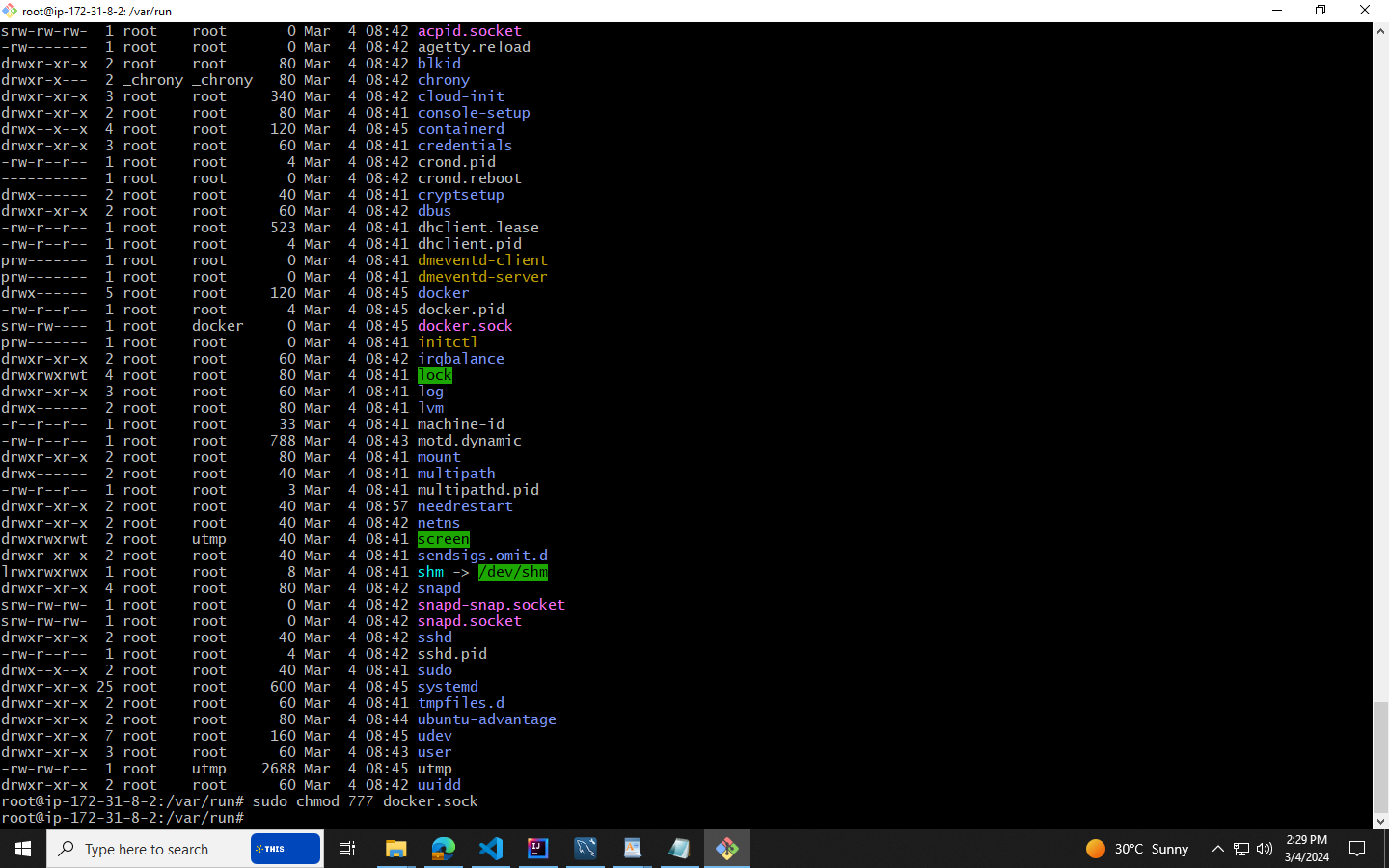
* Check the running status of the docker:

**systemctl status docker**



* Move into the **cd /var/run .**Use the **ls –la** command and check for docker.sock file
* Give the permission to the “docker.sock” file

**sudo chmod 777 docker.sock**



6) **Installation of Jenkins and MySQL :**

* Install the Jenkins application in the instance through

**sudo apt update**

**sudo apt install openjdk-17-jre**

**curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \**

**/usr/share/keyrings/jenkins-keyring.asc > /dev/null**

**echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \**

**https://pkg.jenkins.io/debian binary/ | sudo tee \**

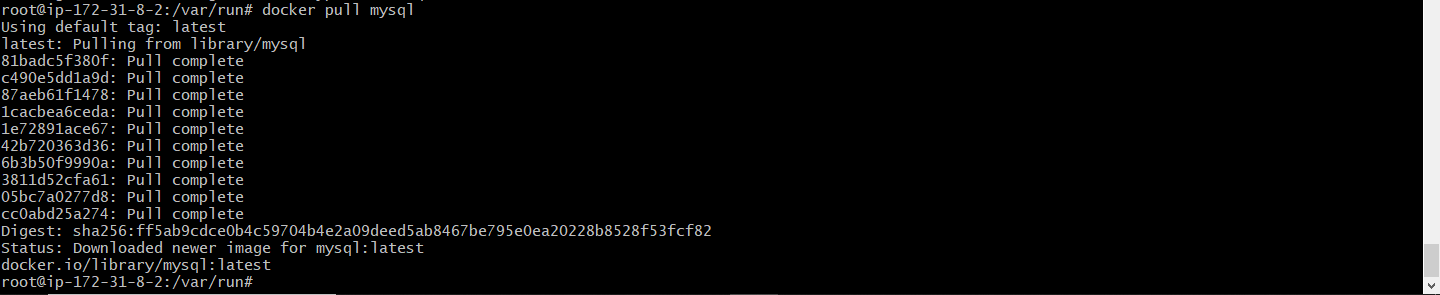
**/etc/apt/sources.list.d/jenkins.list > /dev/null**

**sudo apt-get update**

**sudo apt-get install jenkins**

* Install the MySQL

**docker pull mysql**

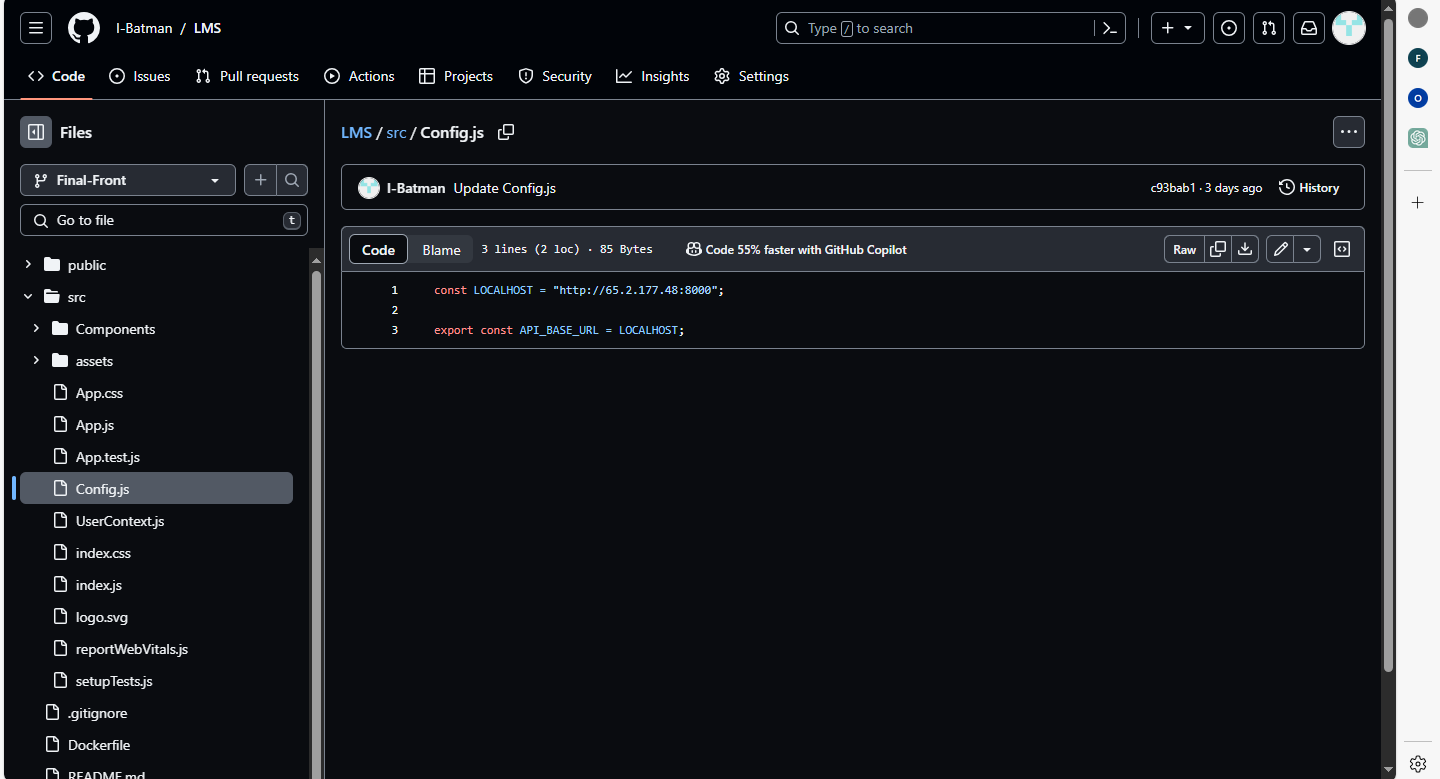


7) Docker Hub Login:

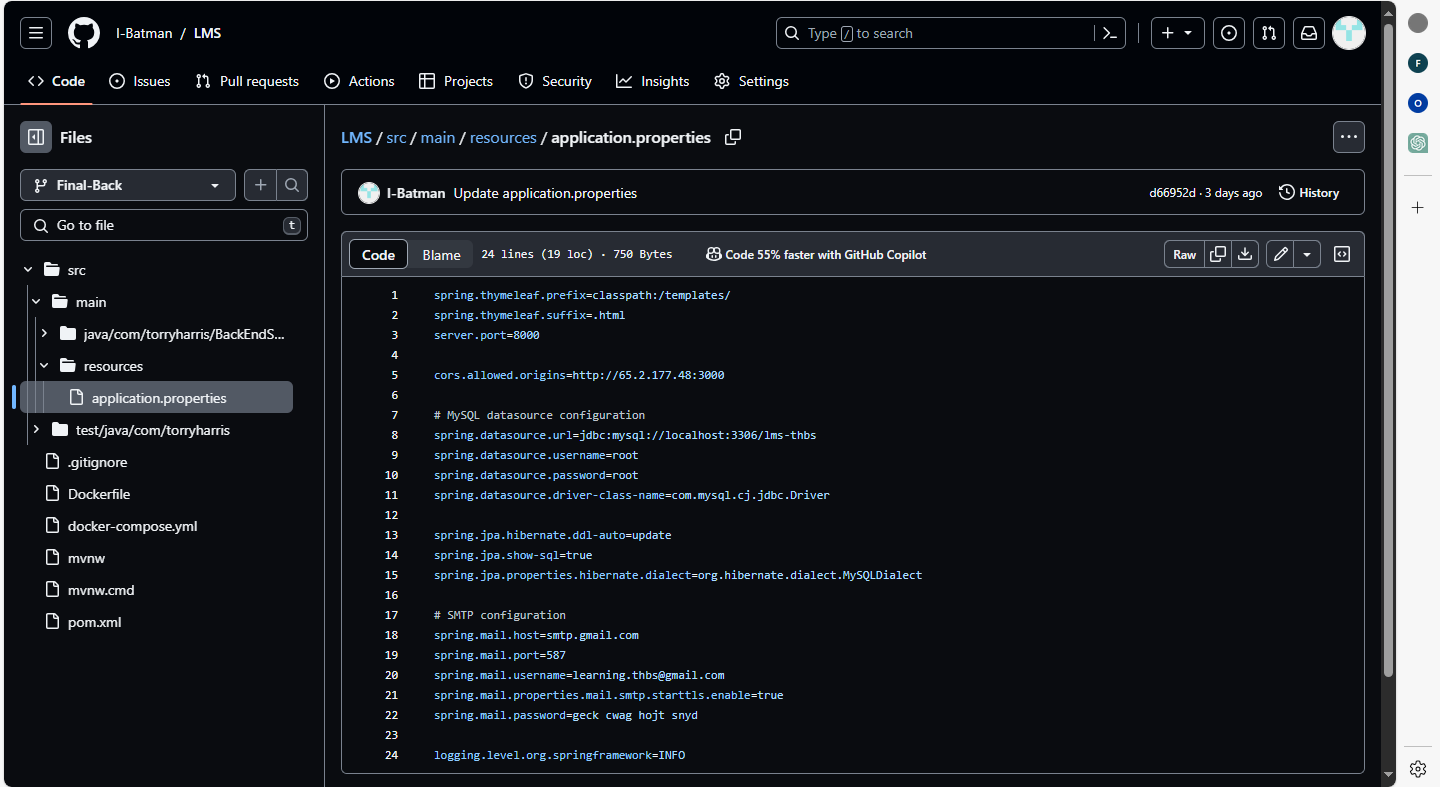
* Create an account with email. Note down the username and password of that account.
* Login into the Docker Hub:

**docker login**

8) Open the Git Repository and replace the Ip address in the both Frontend (Final-Front->src->Config.js) and Backend (Final-Back->src->main->resources->application properties) properties with the current Ip address of the instance.



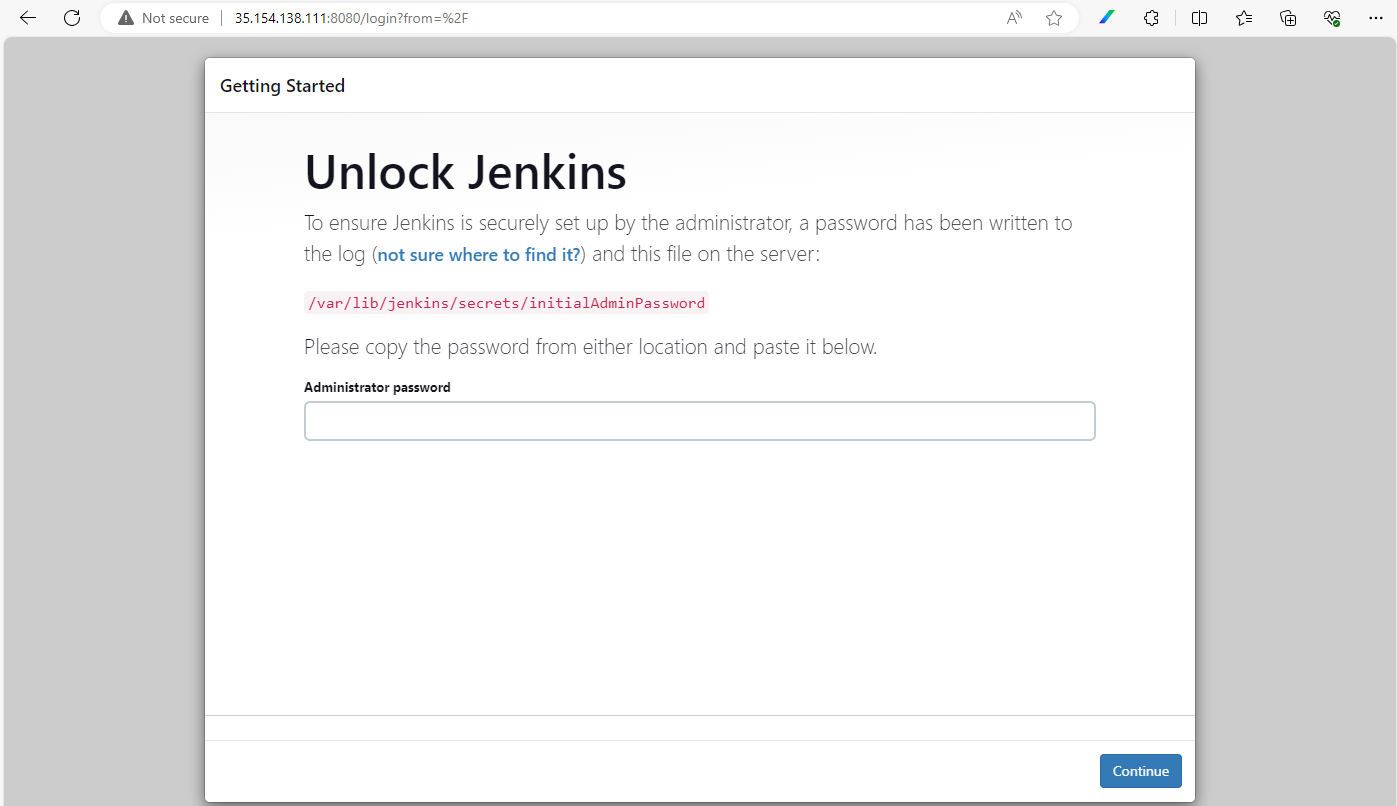
BackEnd Git Branch:



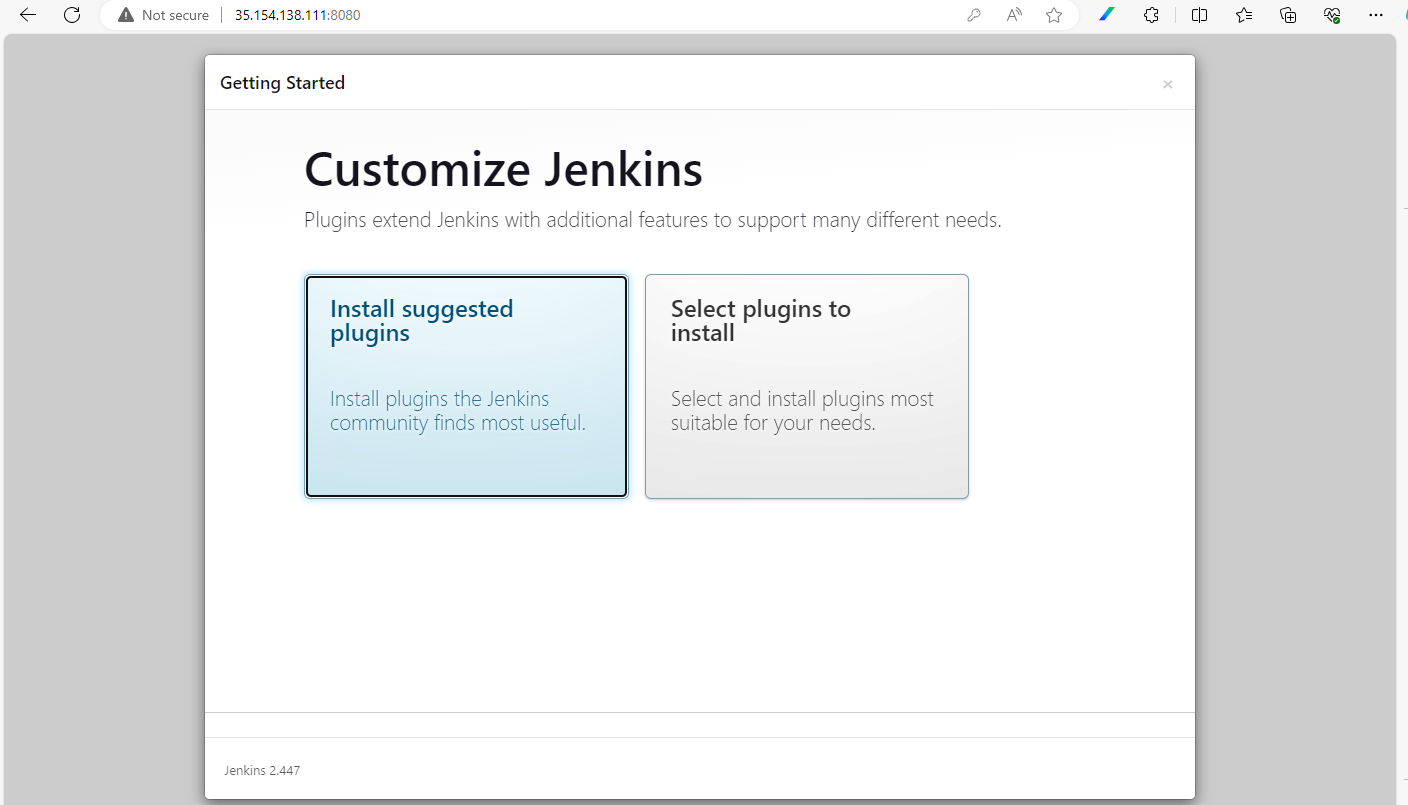
9)Jenkins Configuration:

* Open the Jenkins with the current Ip address with port number ‘8080’
* After opening the Jenkins portal, paste the path which is displaying in the instance and get the password.

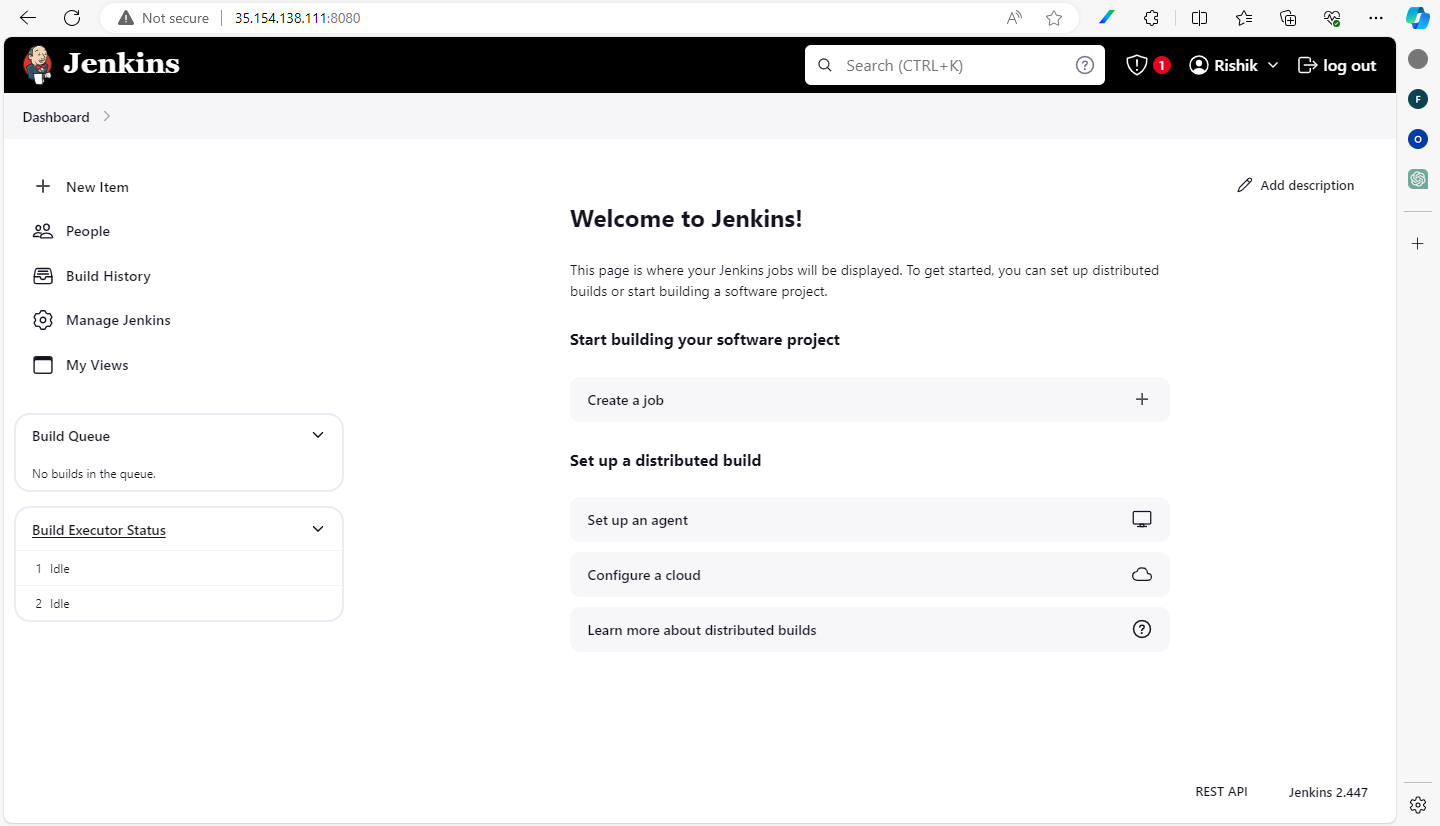
**sudo cat /var/lib/jenkins/secrets/initialAdminPassword**



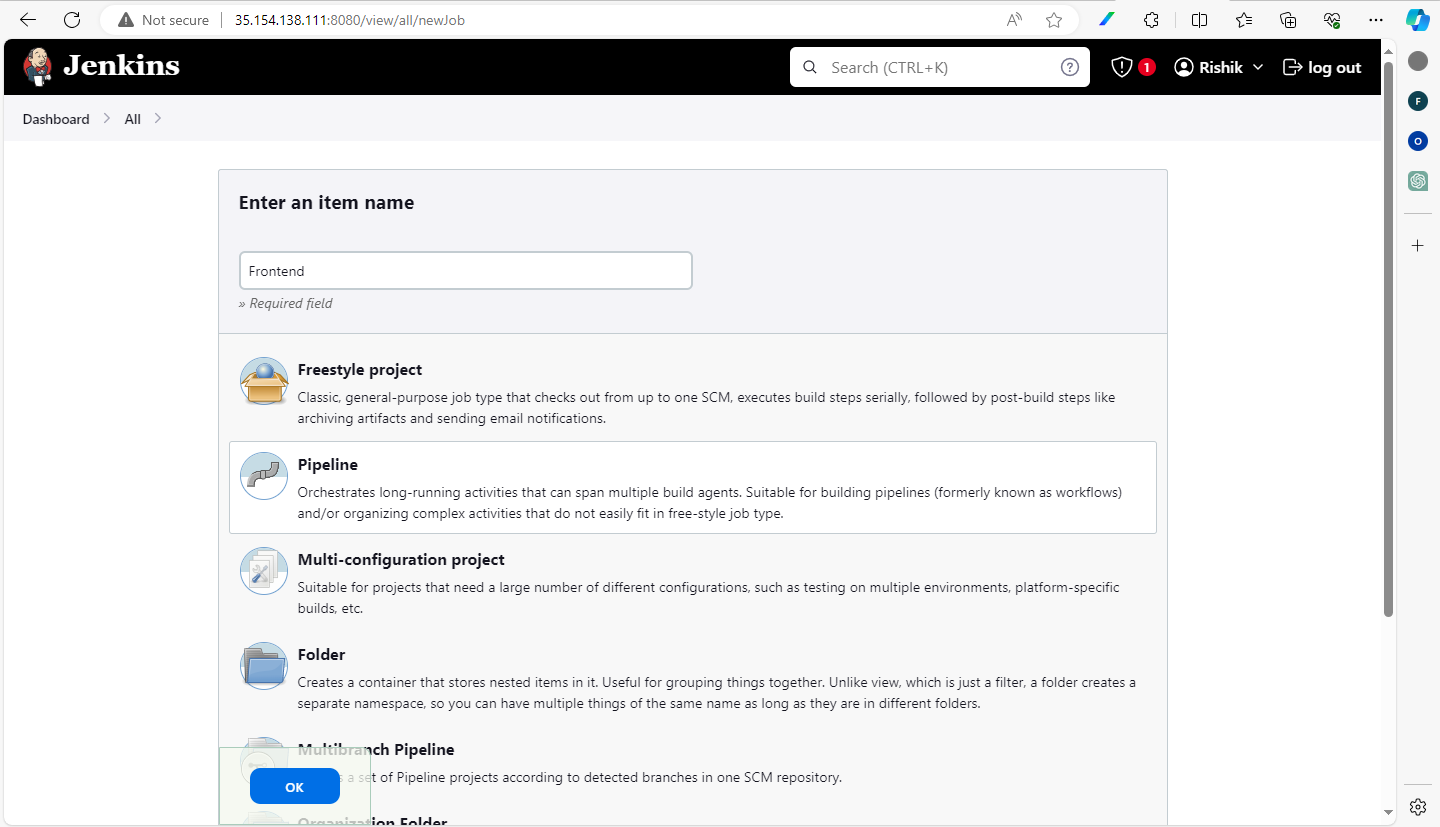
* Install the “Suggested Plugins” and create an account username password mail and click on next.



* Now, click on the new item “Front-End”

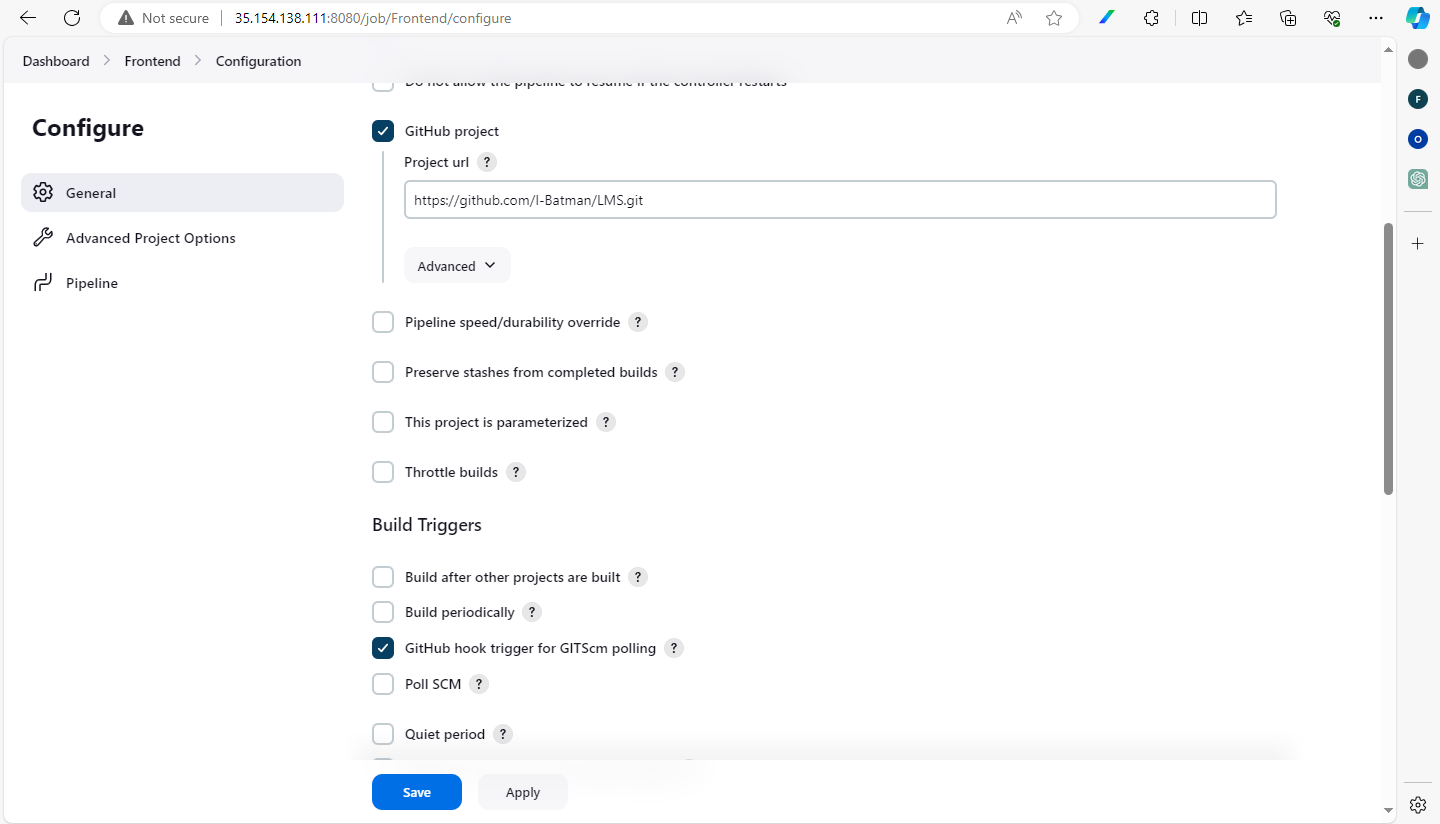


* Create a new pipeline.

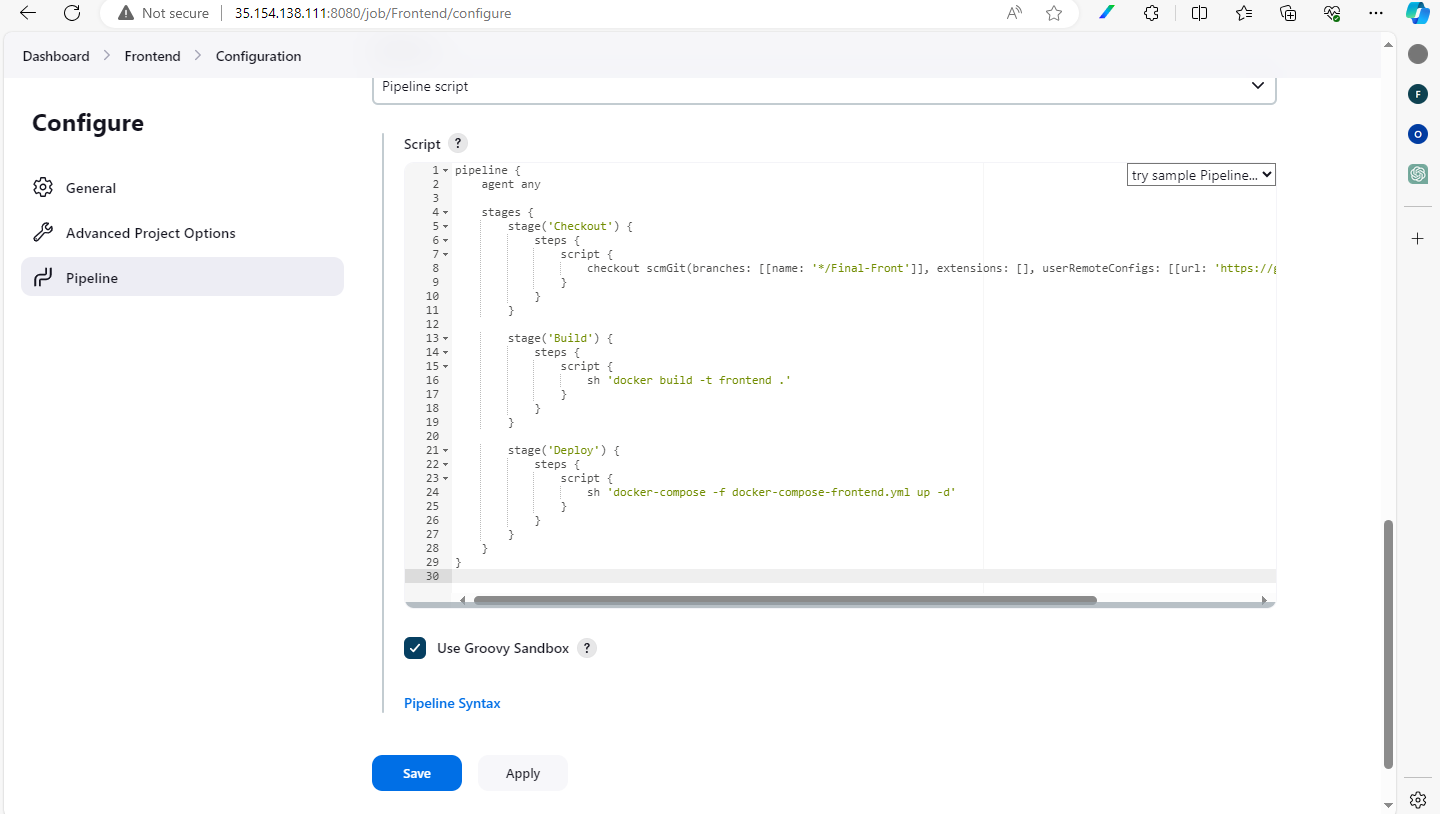


* Select GitHub project below in the configure and paste the GitHub link and GitHub hook trigger for GITScm polling should be enabled

<https://github.com/I-Batman/LMS.git>



* Provide the pipeline script for the project in the pipeline script region.



* Frontend Pipeline Script :

**pipeline {**

**agent any**

**stages {**

**stage('Checkout') {**

**steps {**

**script {**

**checkout scmGit(branches: [[name: '\*/Final-Front']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/I-Batman/LMS.git']])**

**}**

**}**

**}**

**stage('Build') {**

**steps {**

**script {**

**sh 'docker build -t frontend .'**

**}**

**}**

**}**

**stage('Deploy') {**

**steps {**

**script {**

**sh 'docker-compose -f docker-compose-frontend.yml up -d'**

**}**

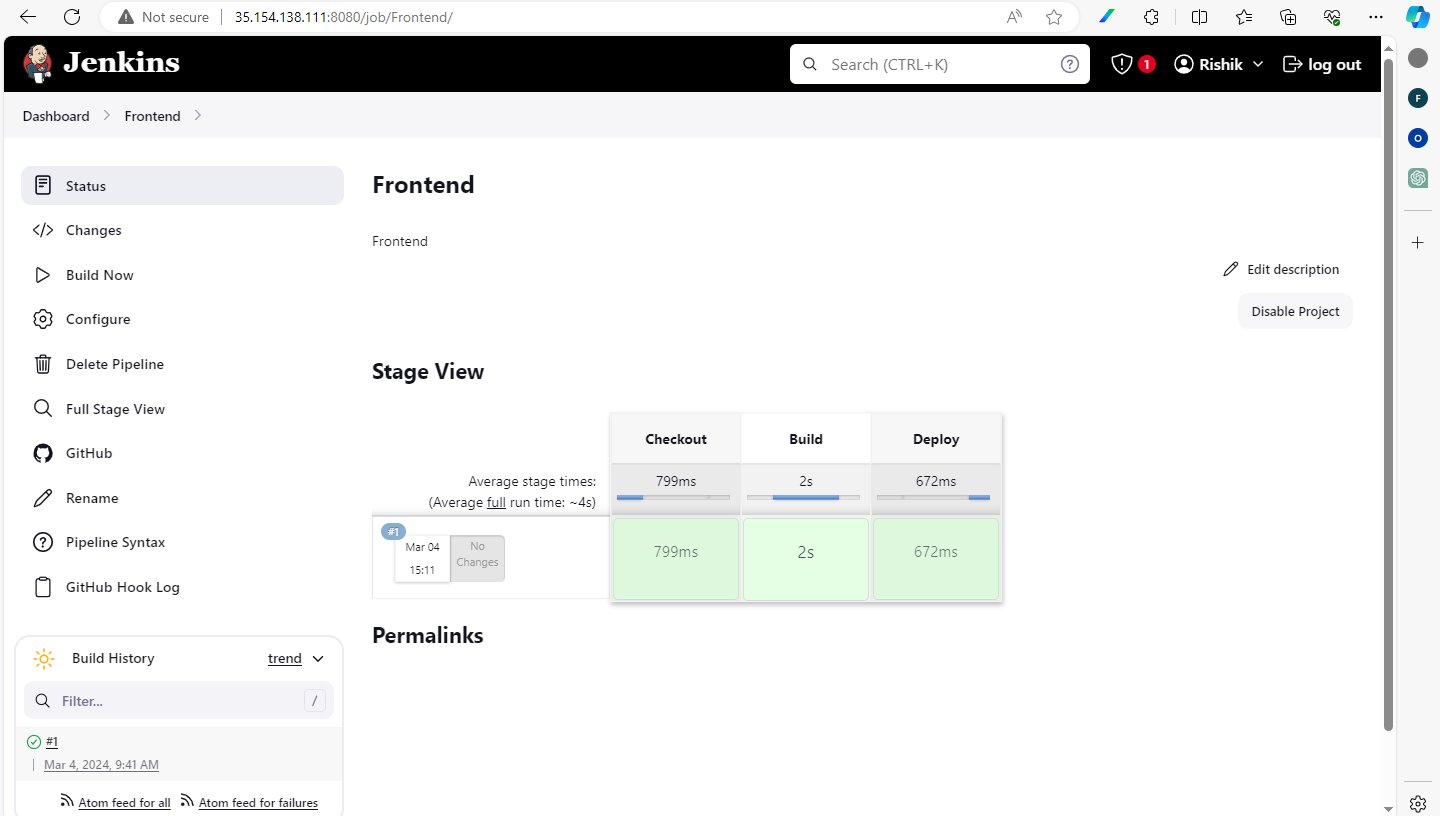
**}**

**}**

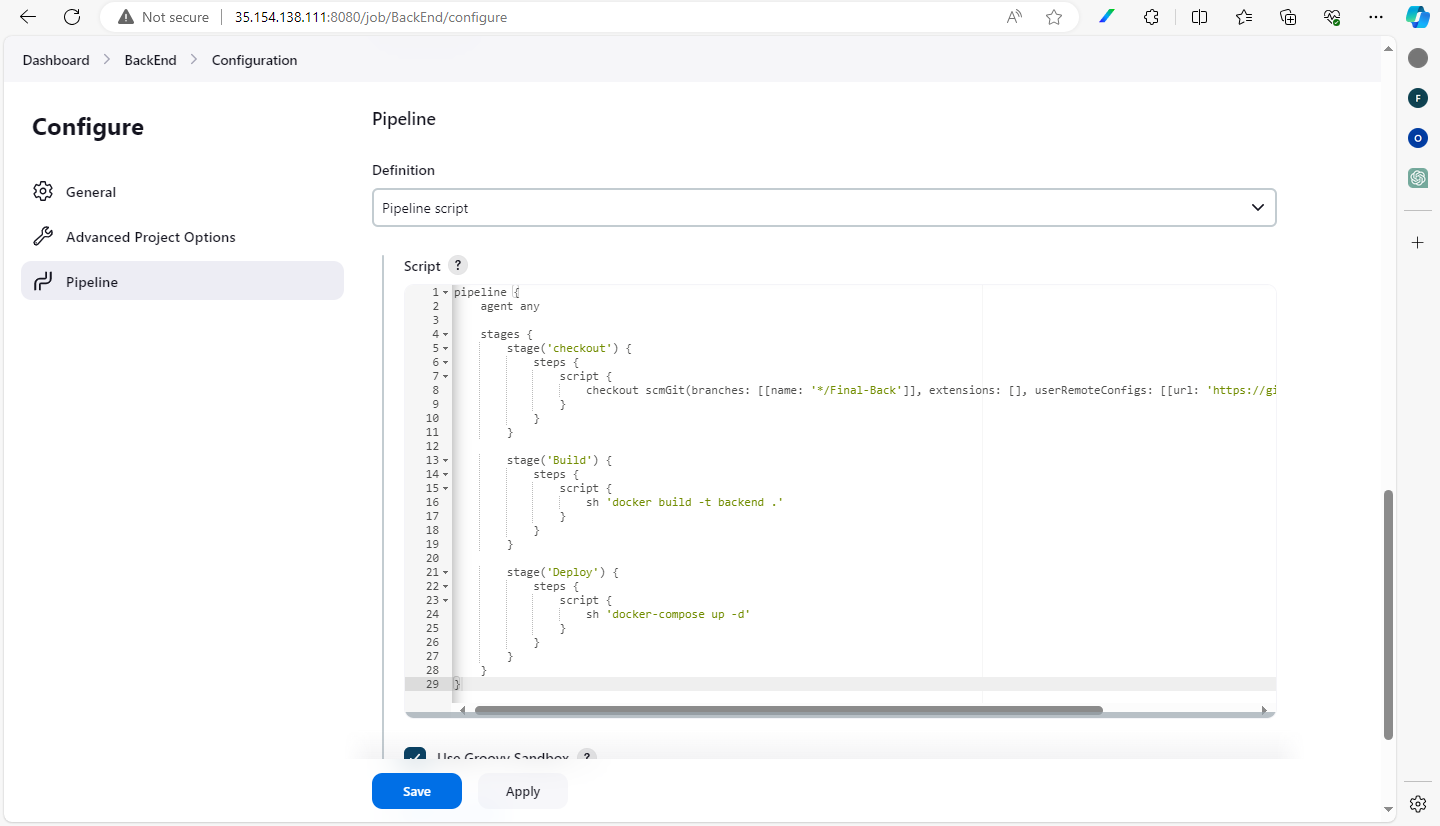
**}**

**}**

* Save the pipeline and build it.



* Again create a new item with pipeline named as “Back-End” and follow the same steps.



* Back-End Pipeline Script:

**pipeline {**

**agent any**

**stages {**

**stage('checkout') {**

**steps {**

**script {**

**checkout scmGit(branches: [[name: '\*/Final-Back']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/I-Batman/LMS.git']])**

**}**

**}**

**}**

**stage('Build') {**

**steps {**

**script {**

**sh 'docker build -t backend .'**

**}**

**}**

**}**

**stage('Deploy') {**

**steps {**

**script {**

**sh 'docker-compose up -d'**

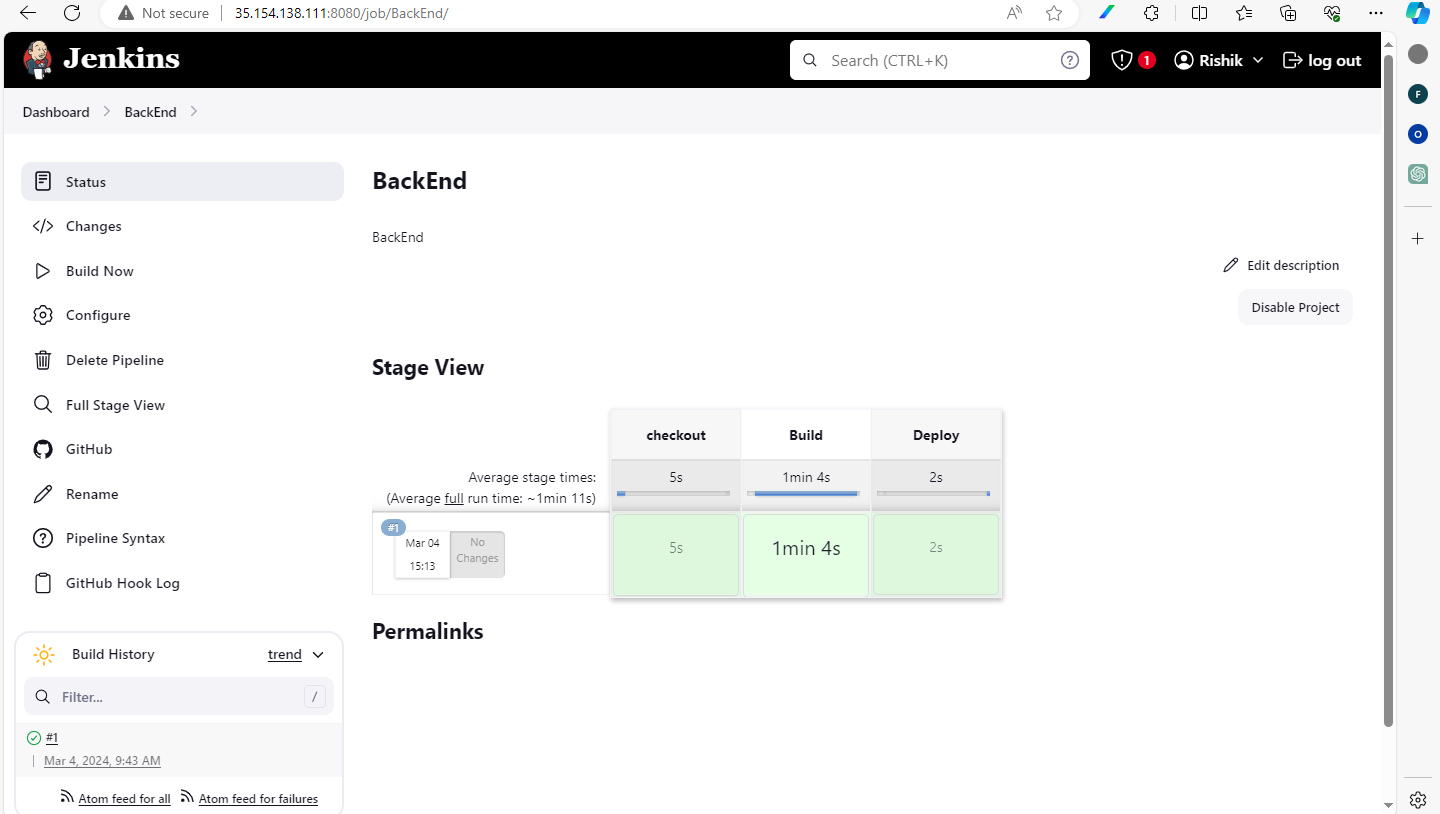
**}**

**}**

**}**

**}**

**}**

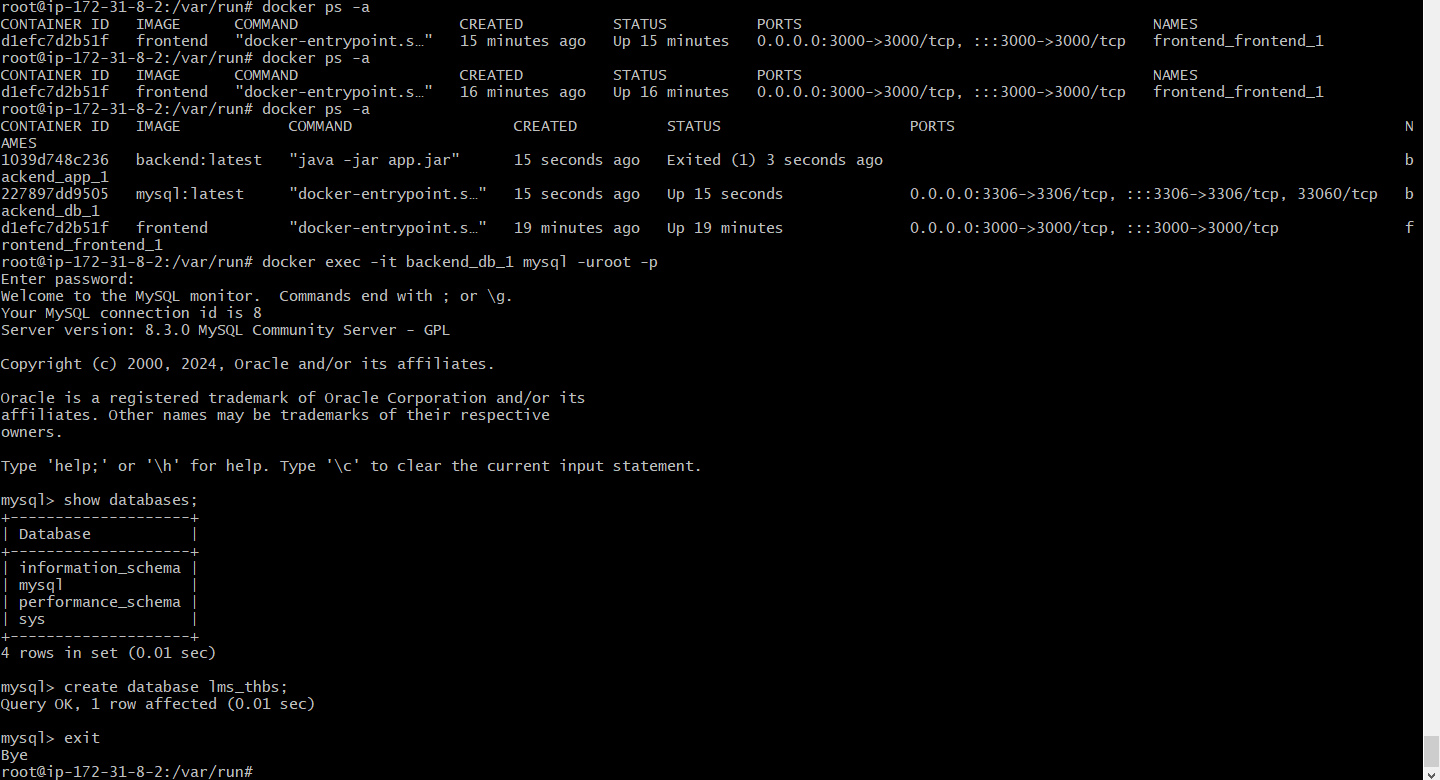


10) Creation of the Database in the MySQL

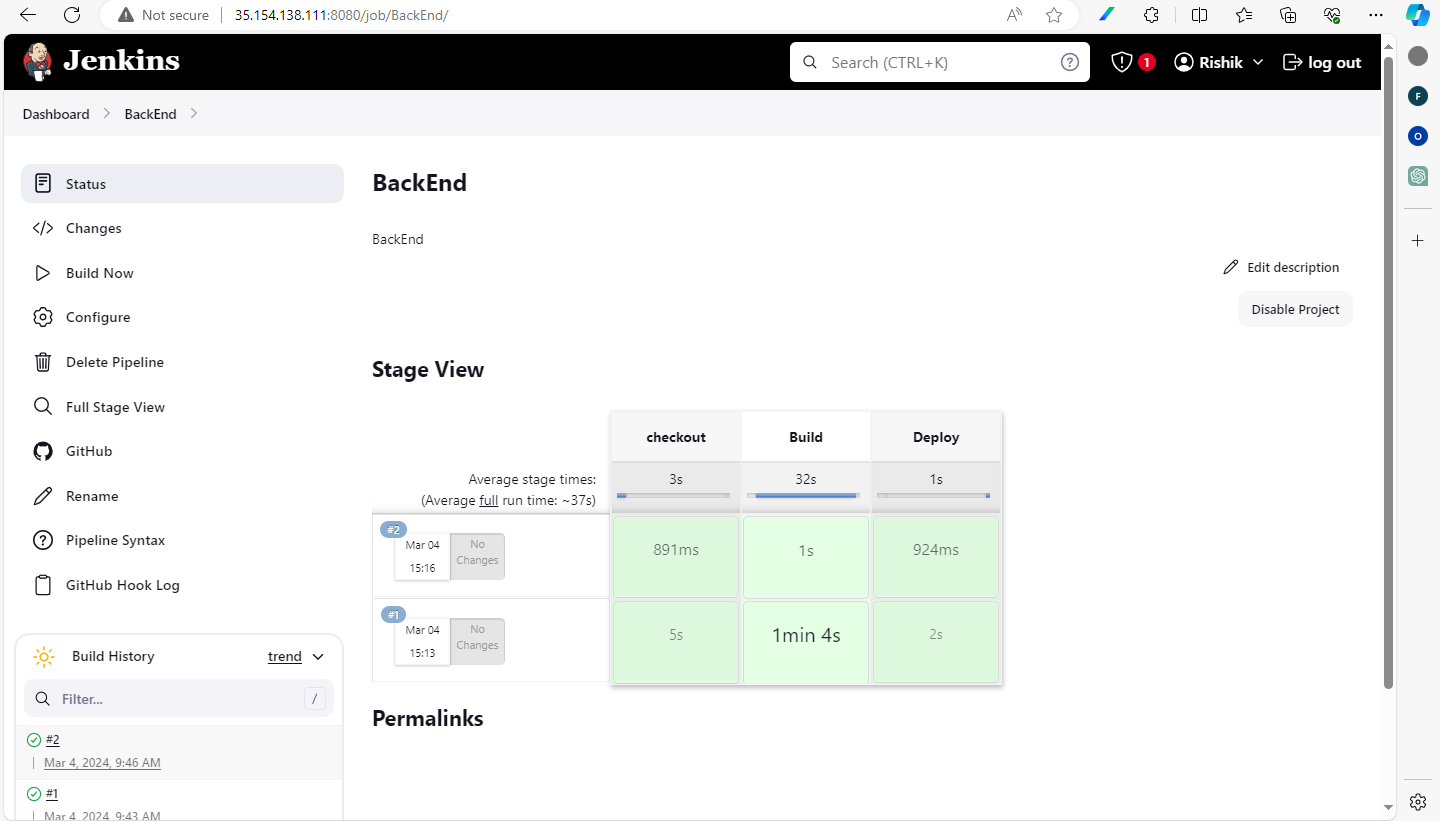
* Backend will fail since the MySQL db is not created. So

**docker exec -it "container\_name" mysql -uroot –p**

* Enter the password “root” and create the database “lms\_thbs” and use it.



11) Re-run the Jenkins Build for Back-End.

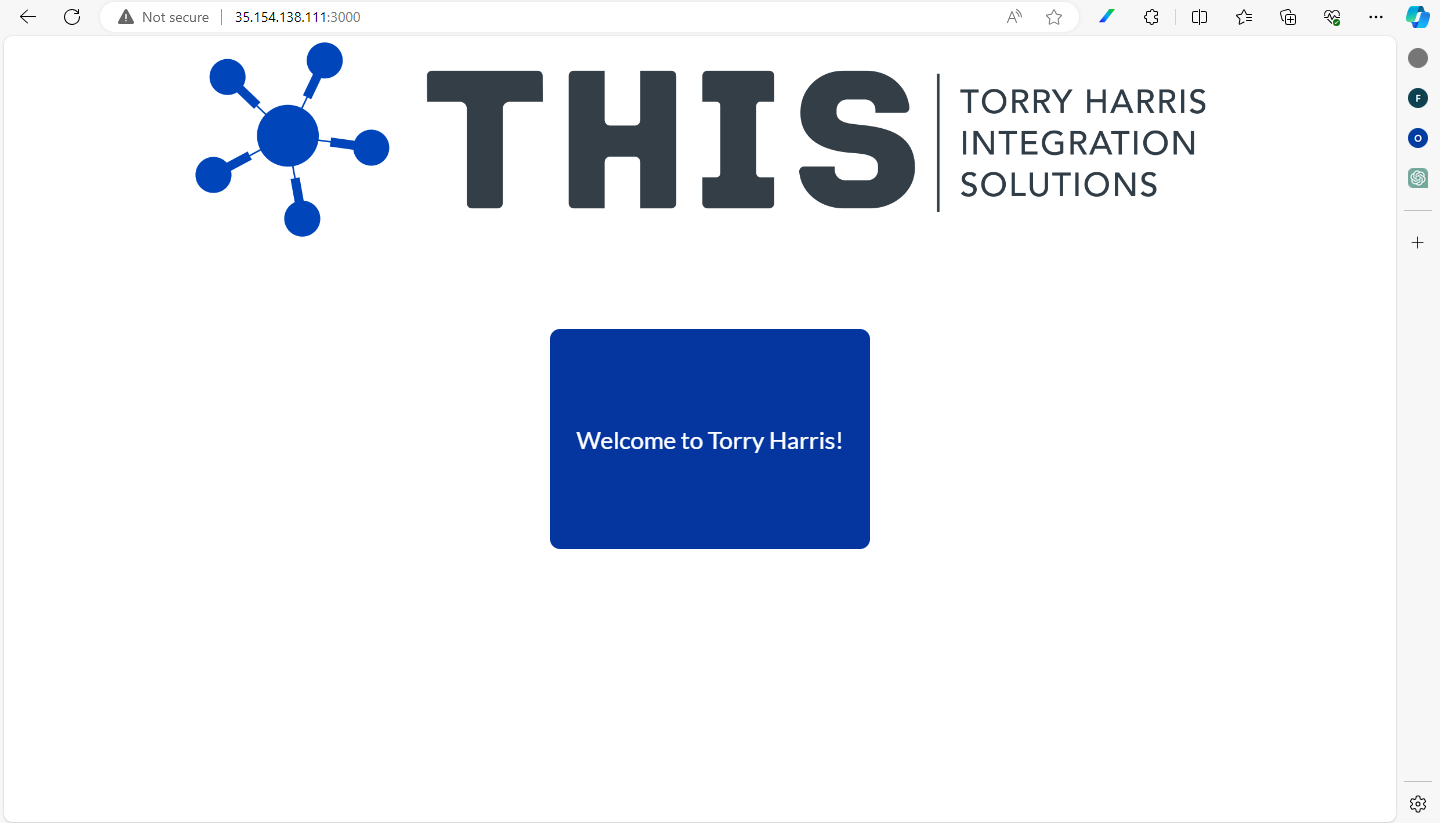


12) To access the both Application and Back-End:

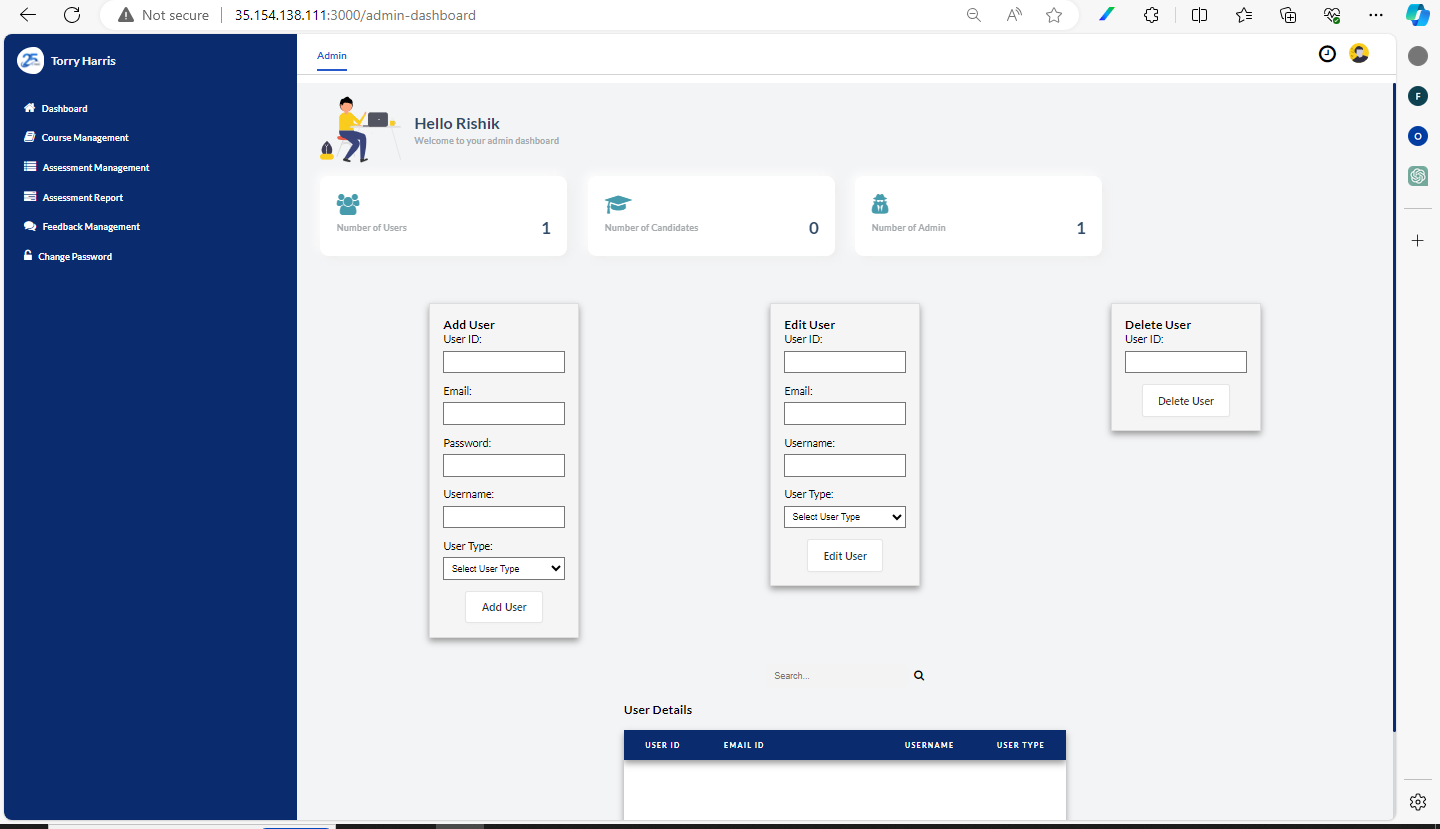
* Application – ip-address:3000
* Back-End – ip-address:8000/admin/dashboard

13) Final Result:

**Home Page:**



**Admin Dashboard:**



**Candidate Dashboard:**

