**Create Full Continuous Delivery lifecycle including automated tomcat deployment using Groovy pipeline code**.

**Jenkins Pipeline as Code**

**Process:**

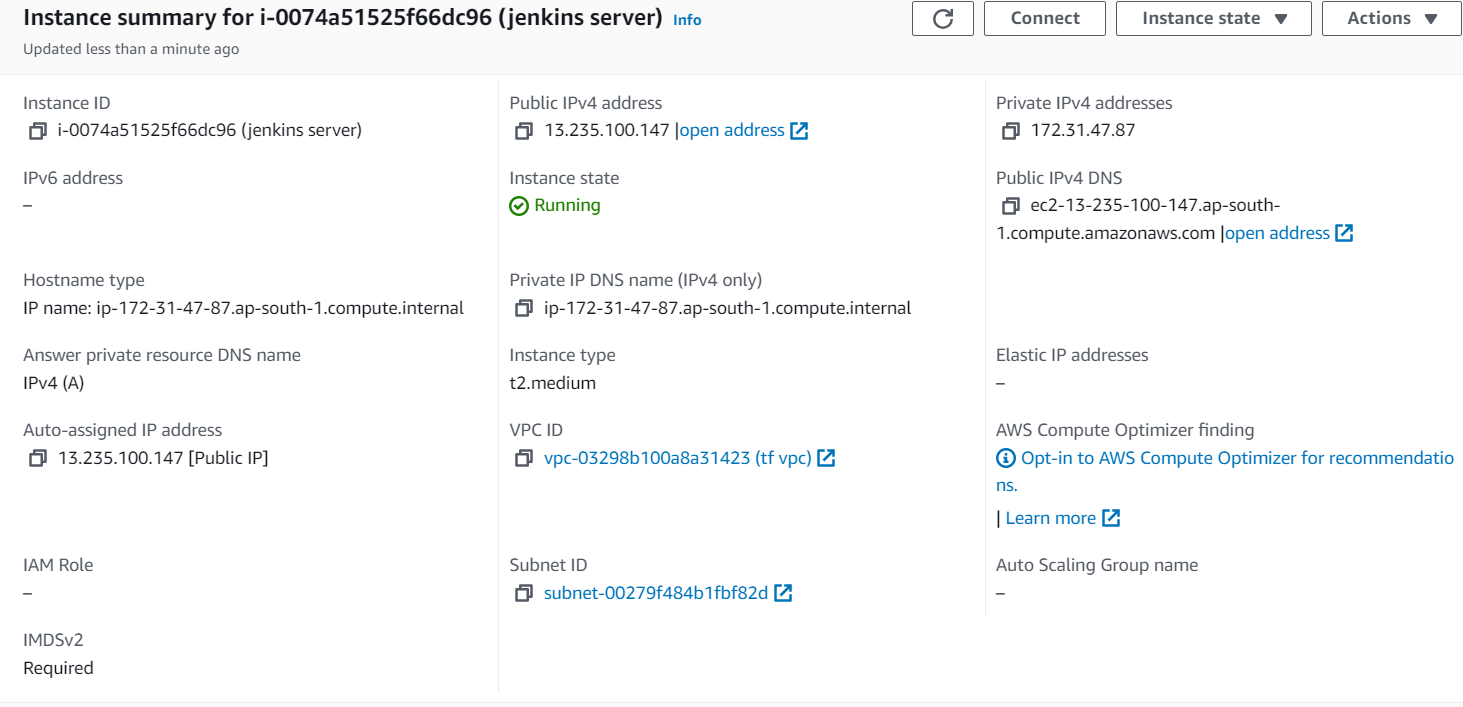
**Set Up AWS Resources**

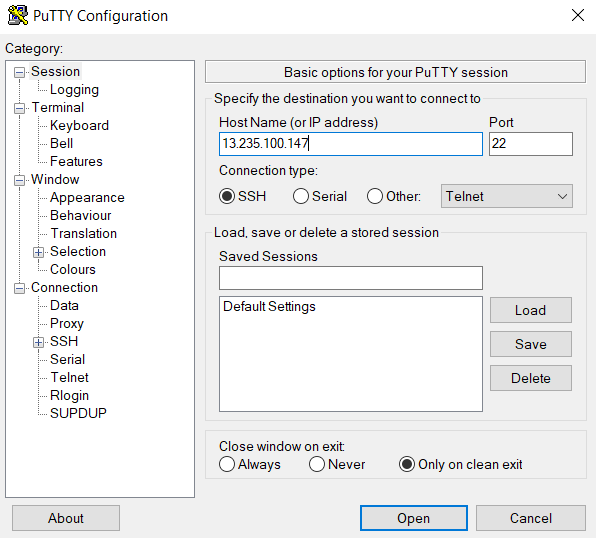
Logged in to AWS Management Console.  
• Launched EC2 Instance: (virtual machine - Ubuntu).

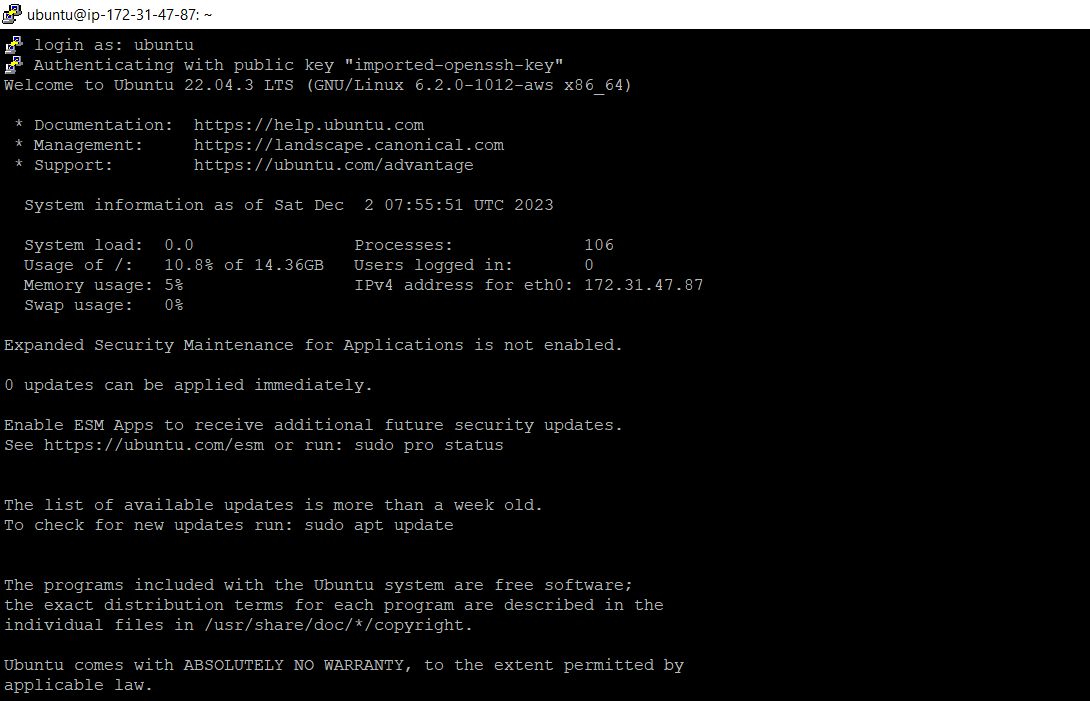
And security groups to allowed SSH (port 22) traffic with instance type - T2.medium and 15Gb disk size and created PPK file.

**Connect to Your EC2 Instance**

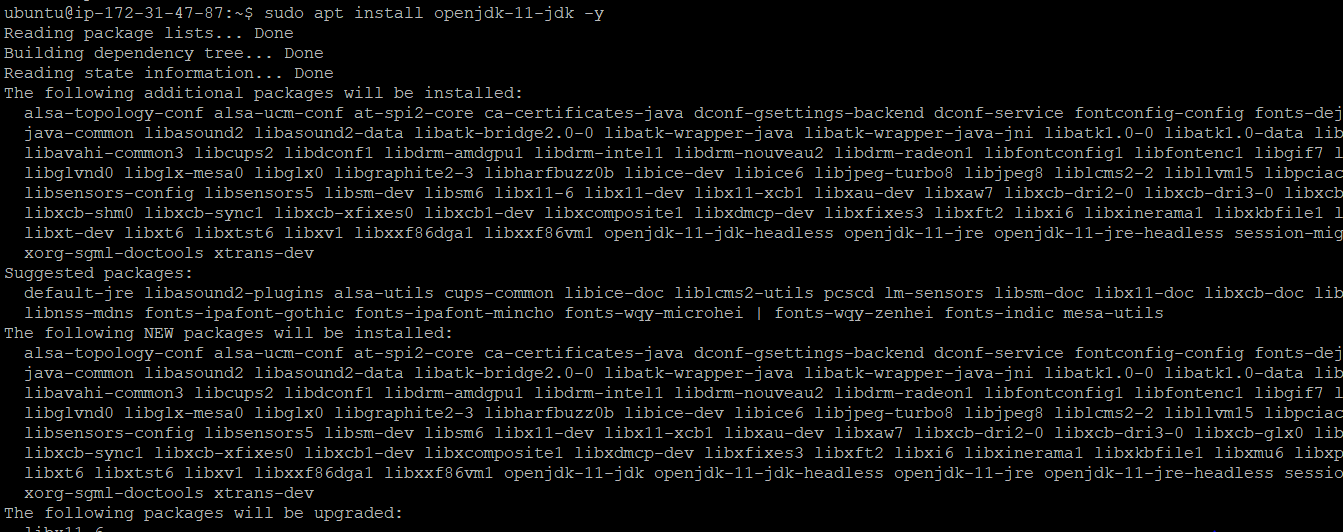
•    Used SSH to connect to EC2 instance using the PPK file via Putty as below.

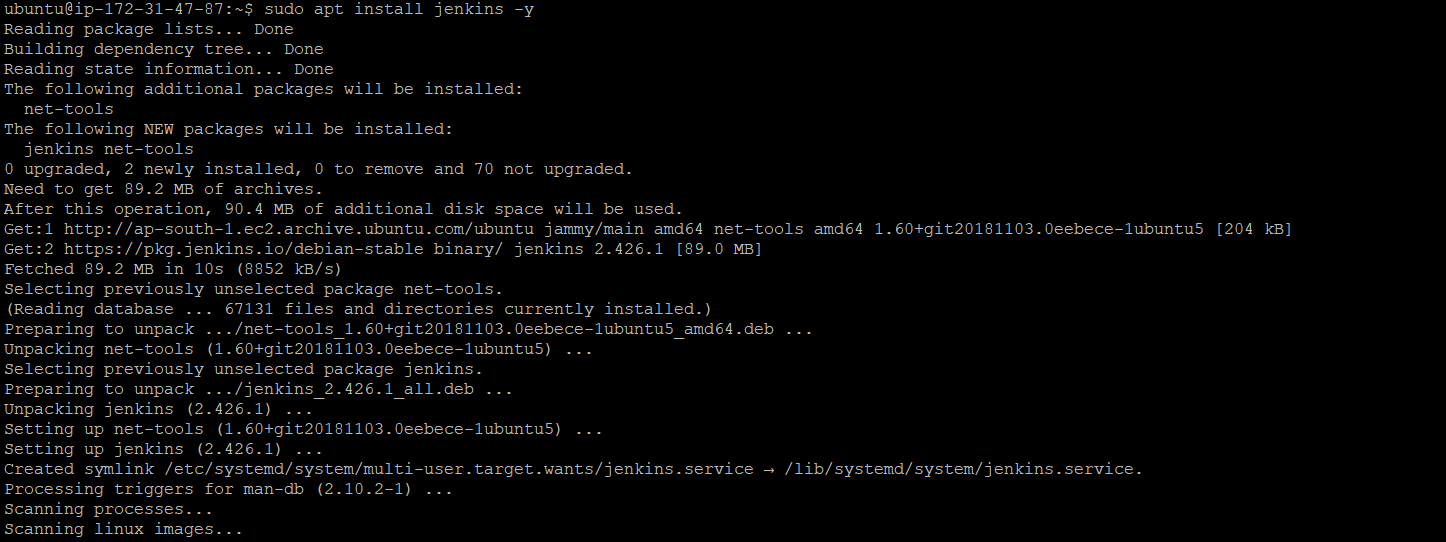




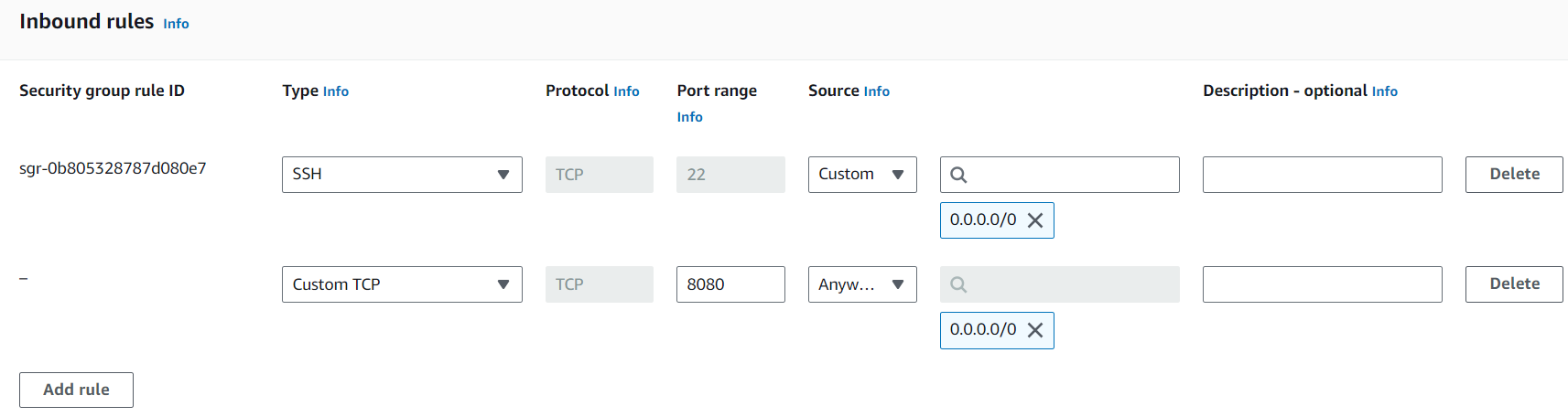


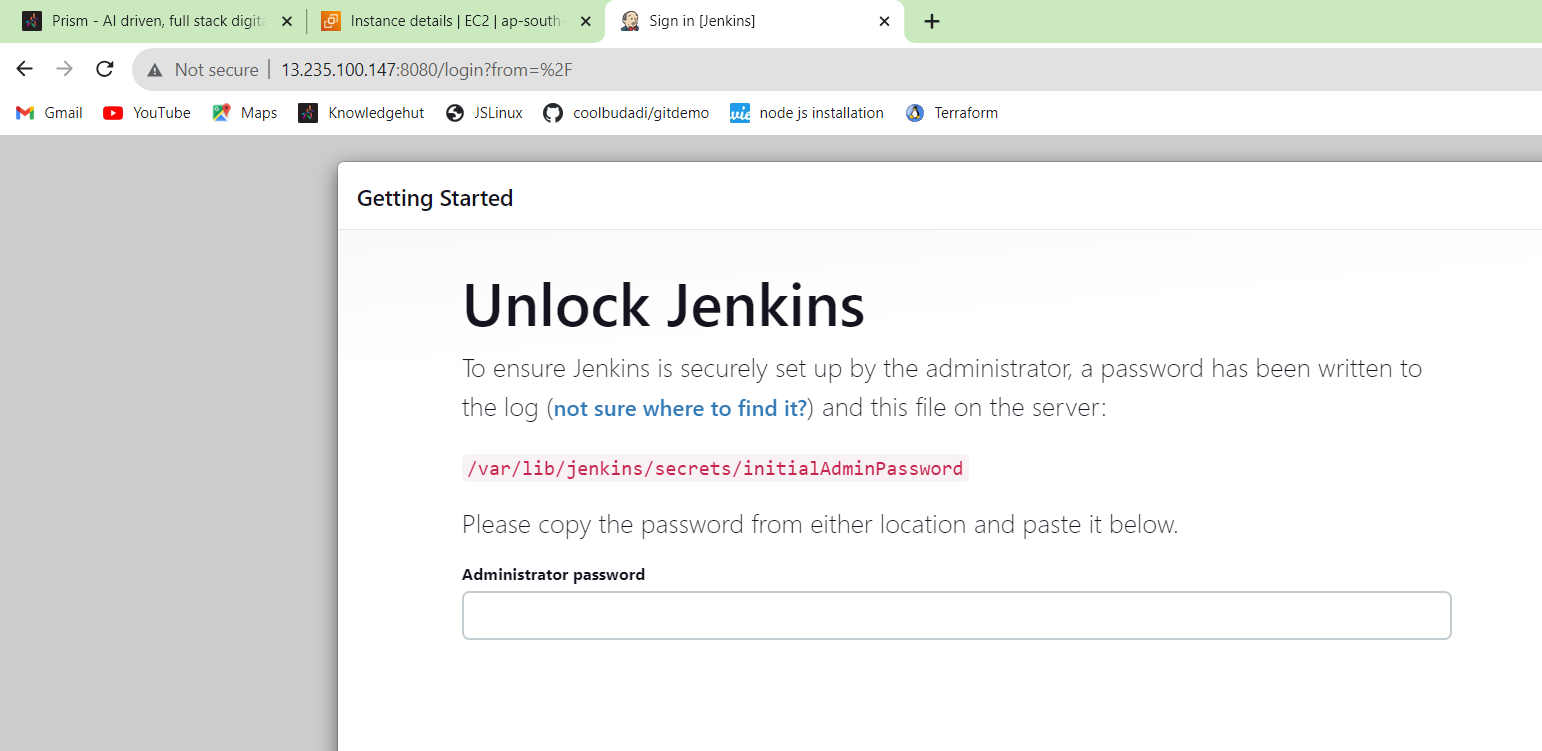
**Installed java and Jenkins on to the machine.**

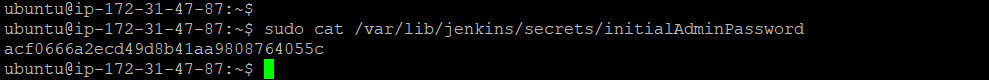




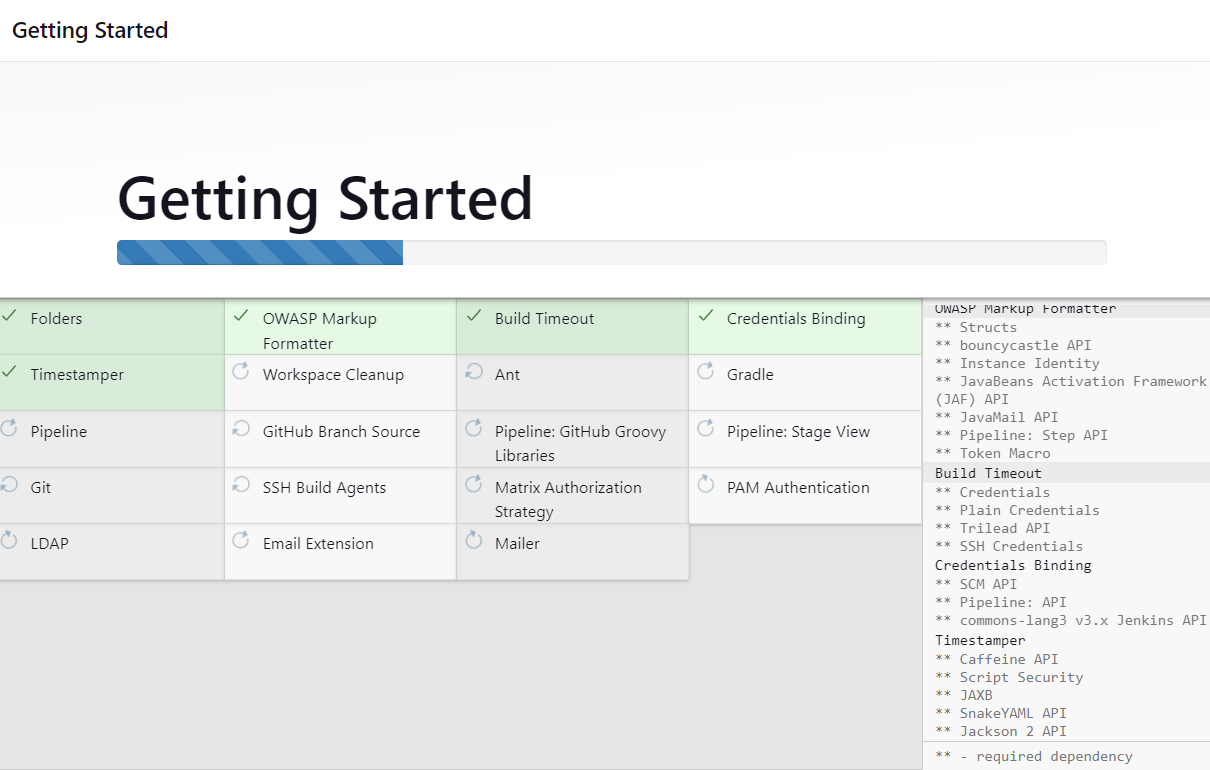
**Added 8080 port entry to the machine for accessing Jenkins**



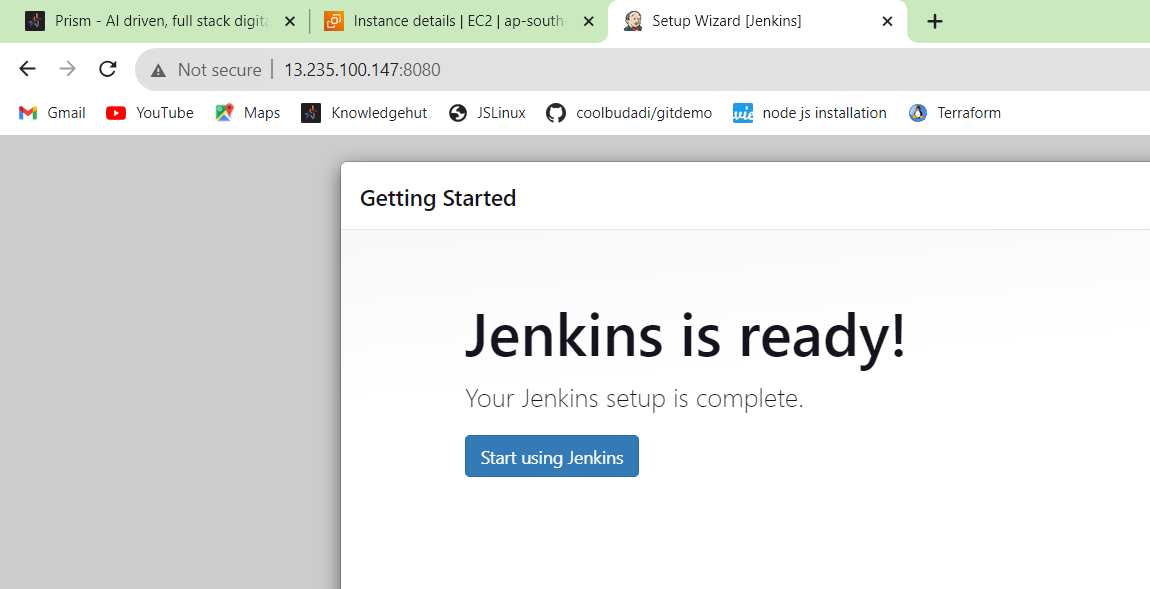




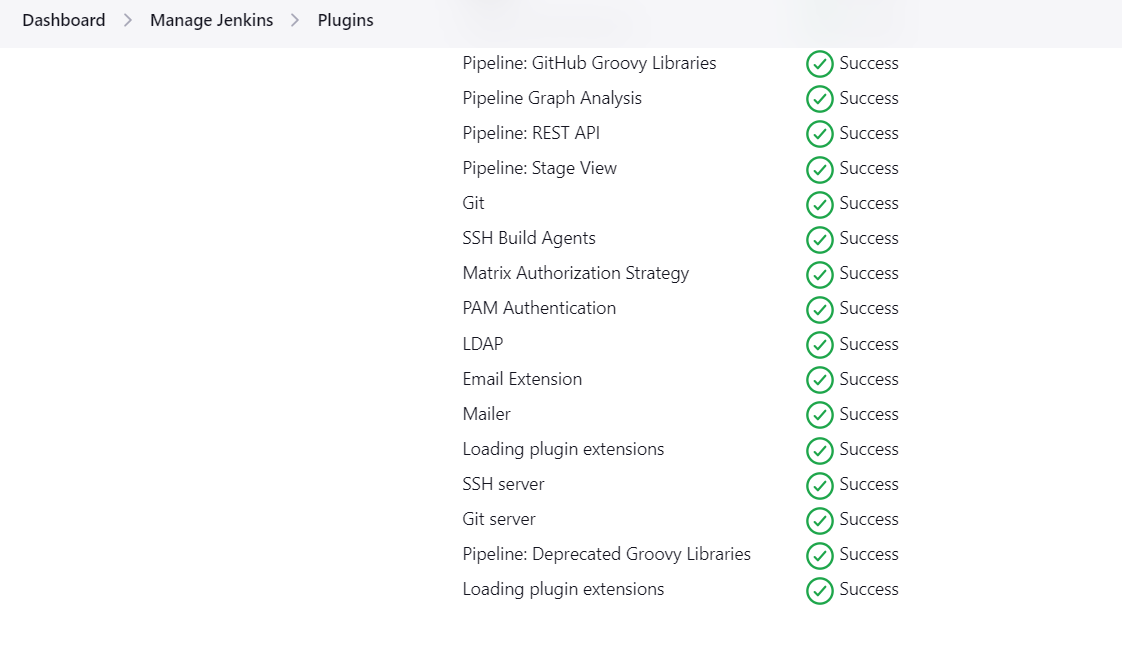
**Installing suggested plugins :**



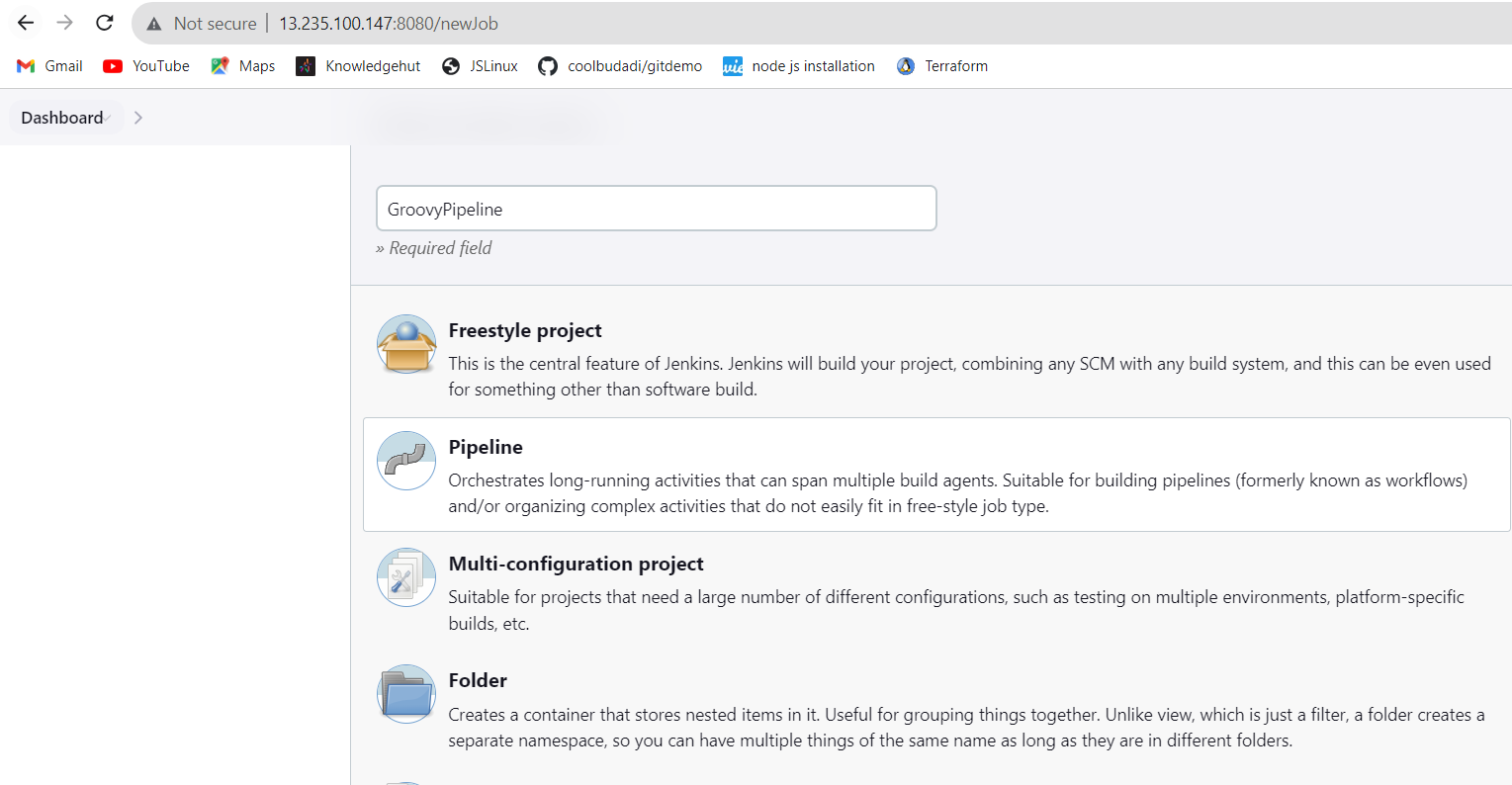
**And Jenkins is ready:**



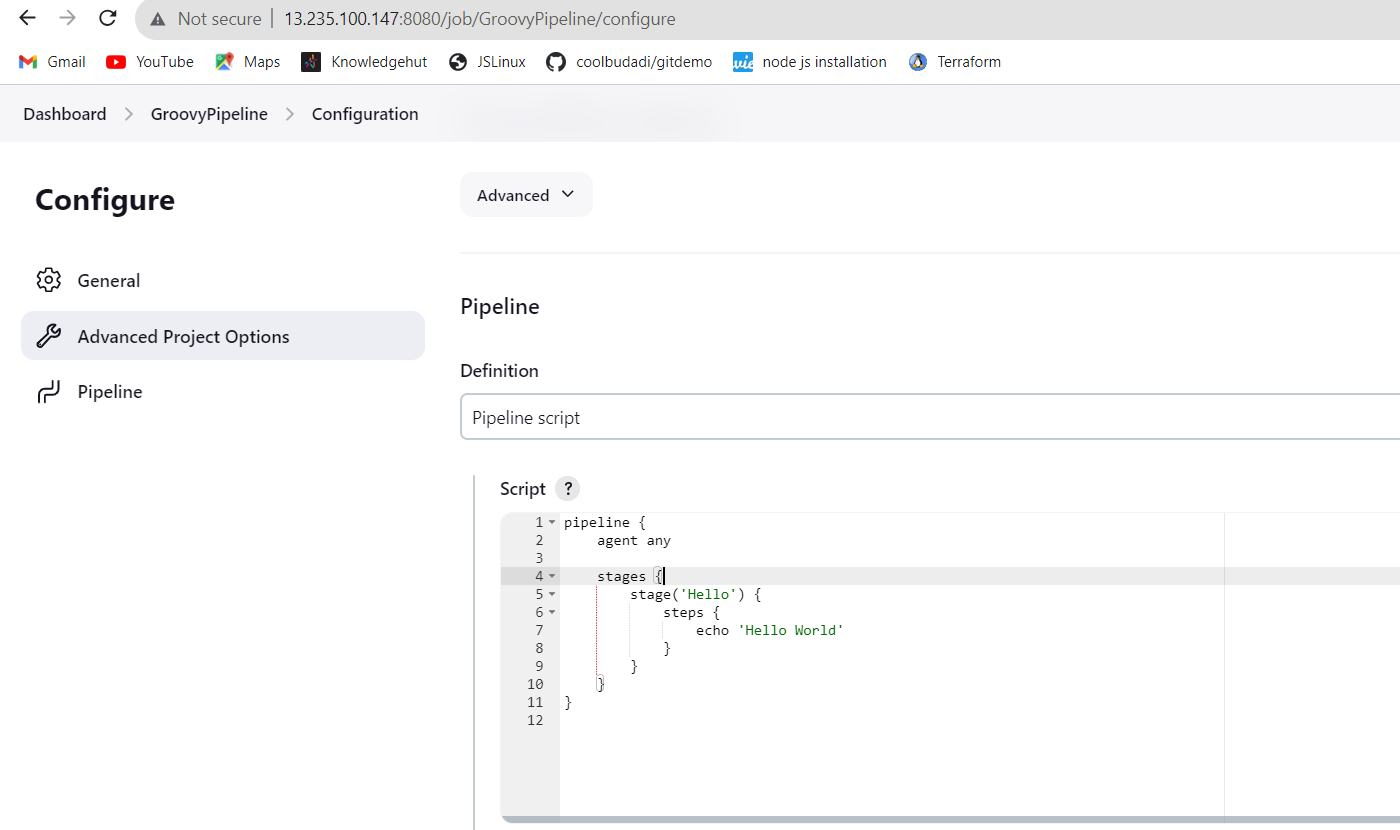
**Installed “pipeline” plugin (manage jenkins>plugin manager)**



**After installing pipeline plugin Created a new job type pipeline and name “GroovyPipeline**”

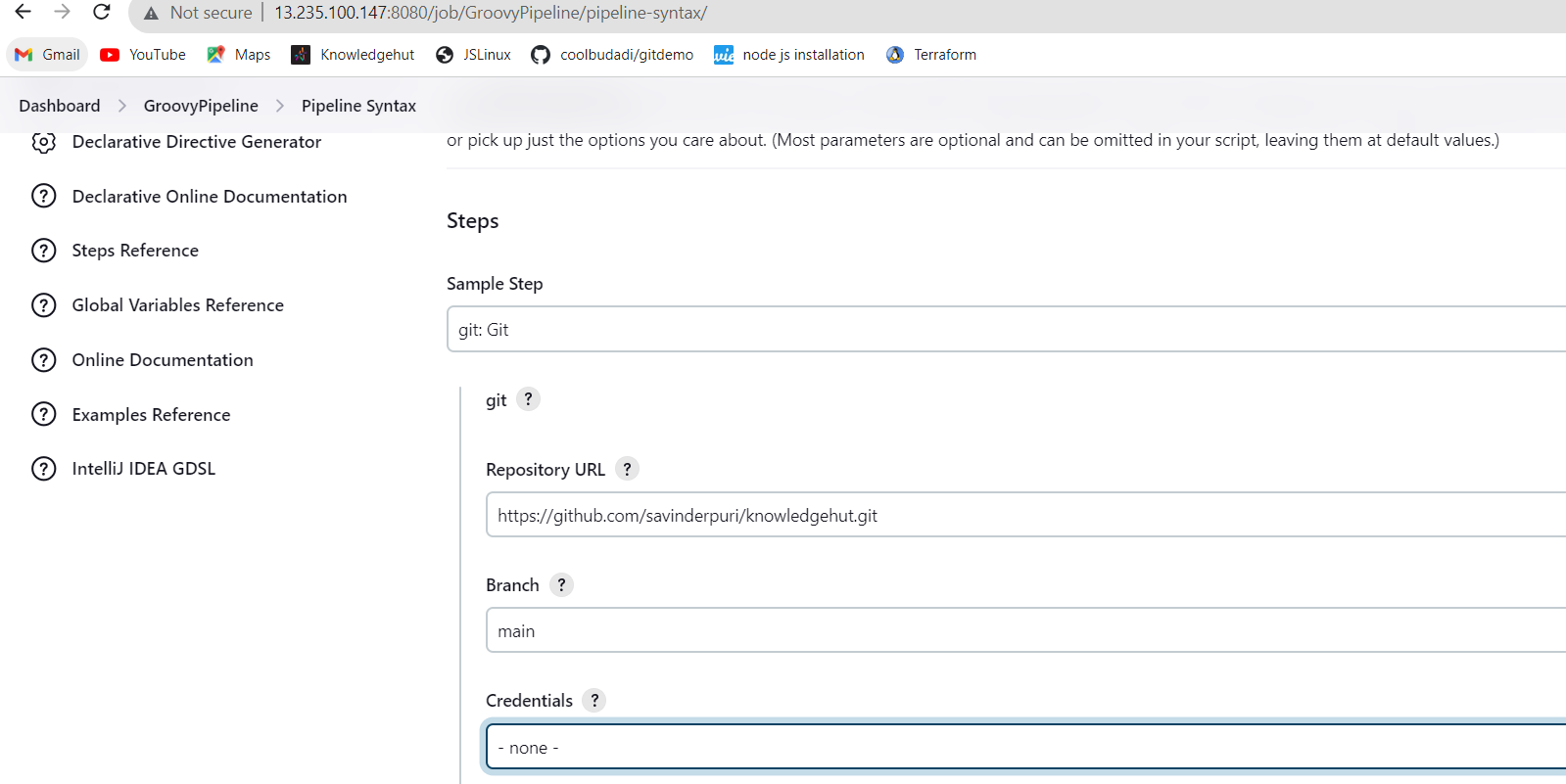


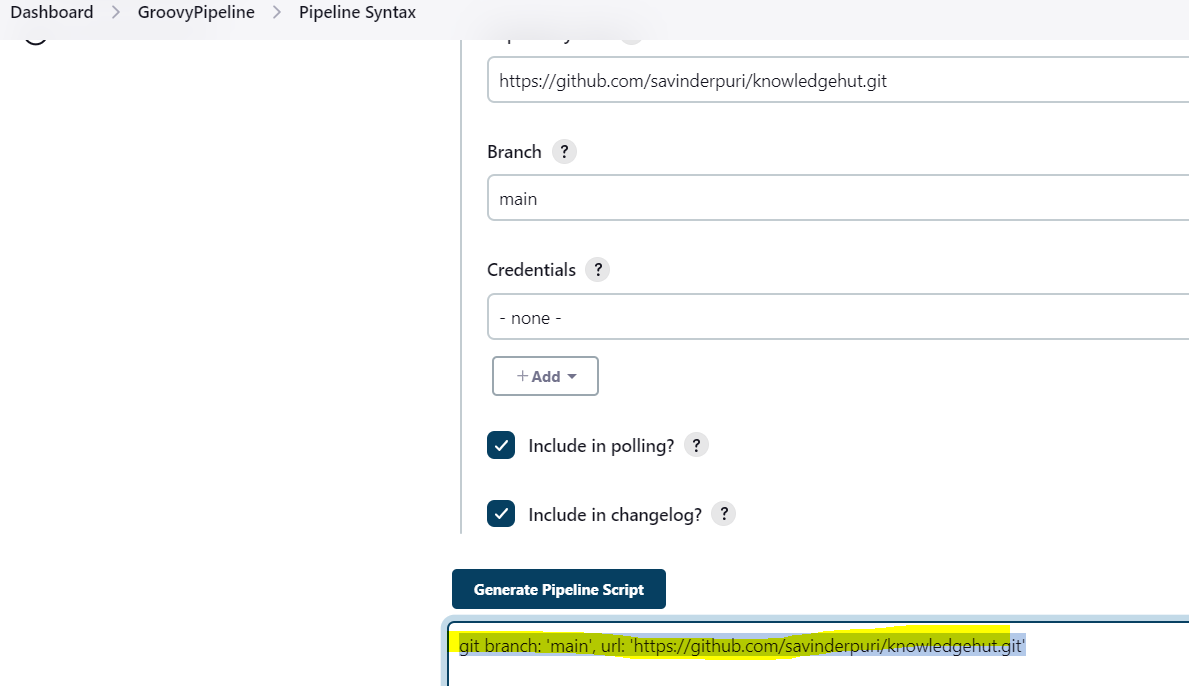
**In the “script” field of Pipeline section selected the “Hello World” pipeline as below**:



**Then generated Groovy code for pipeline. TO do that - duplicate your browser page and Add following to your browser page URL to get to the pipeline syntax page: /pipeline-syntax/**

**On this page selected Sample Step. For example we need to download the git repository**[**https://github.com/savinderpuri/knowledgehut.git**](https://github.com/savinderpuri/knowledgehut.git)**from the branch main without credentials**

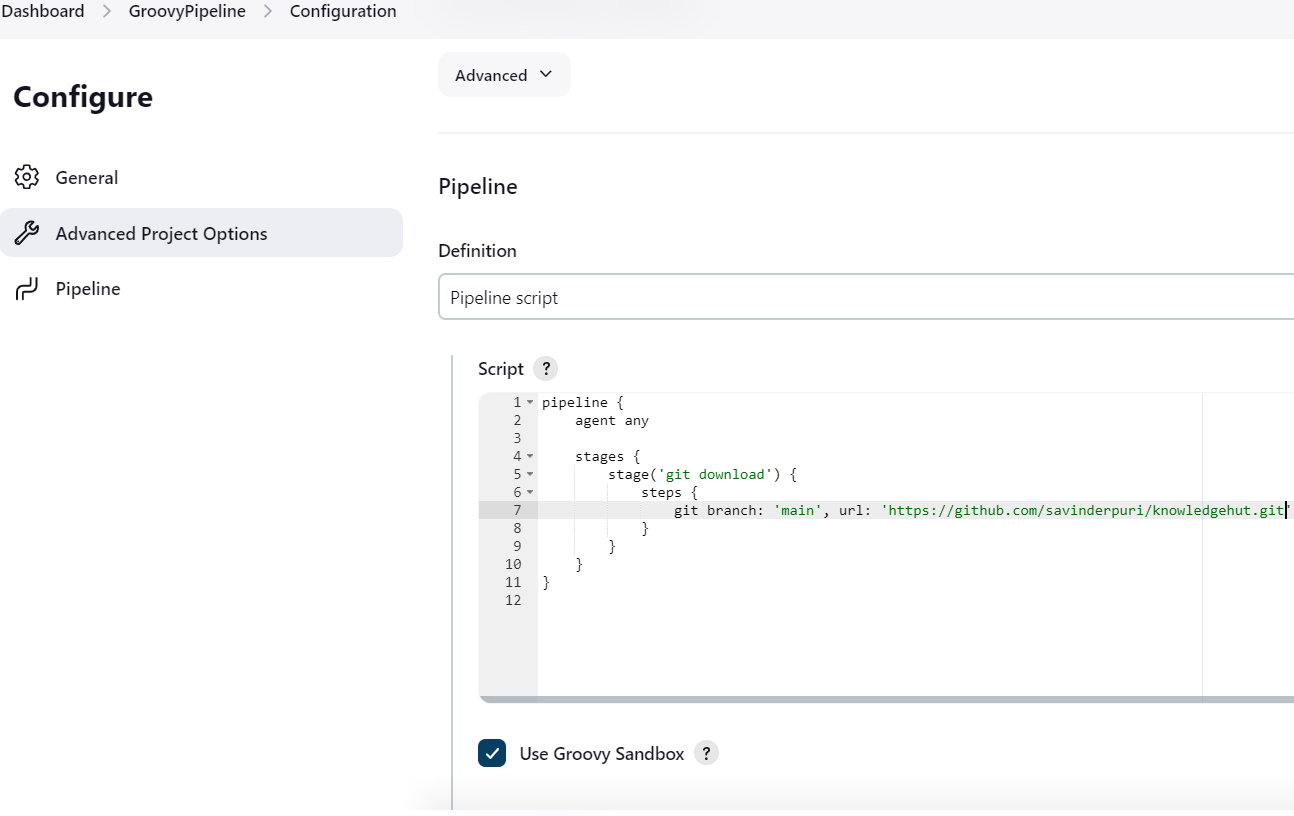


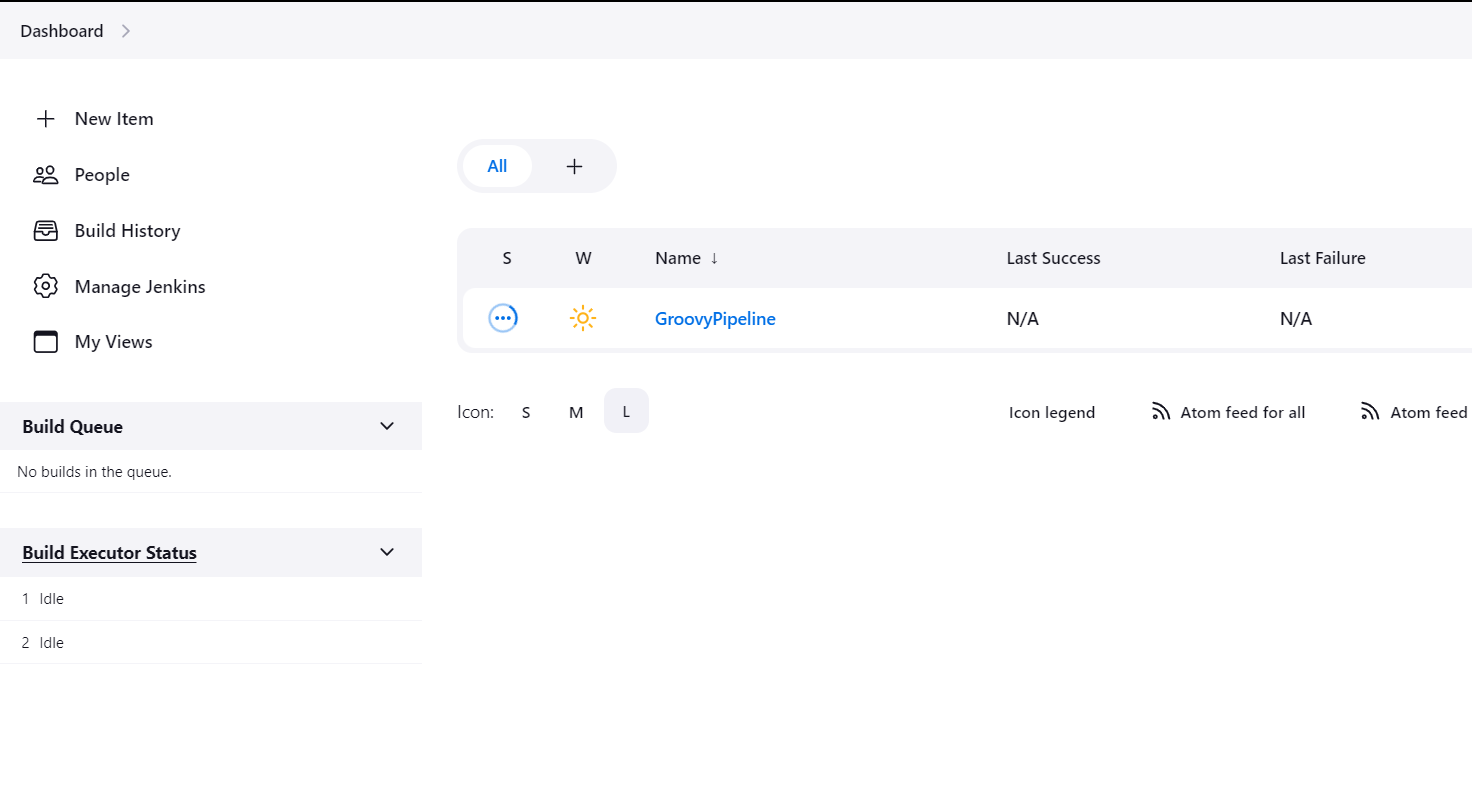


**Generated Groovy code copied and add to pipeline script as below:**

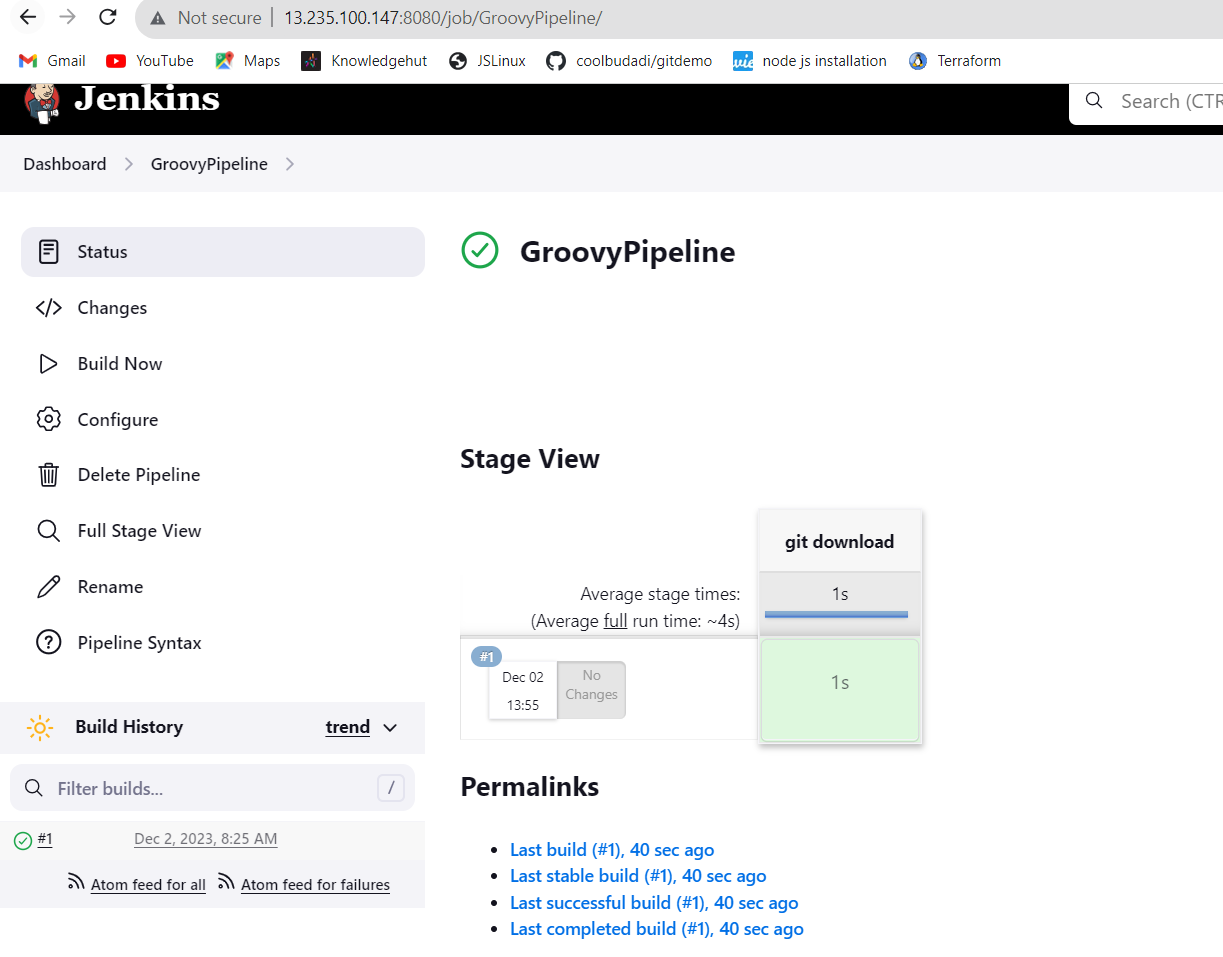
**git branch: 'main', url: 'https://github.com/savinderpuri/knowledgehut.git'**

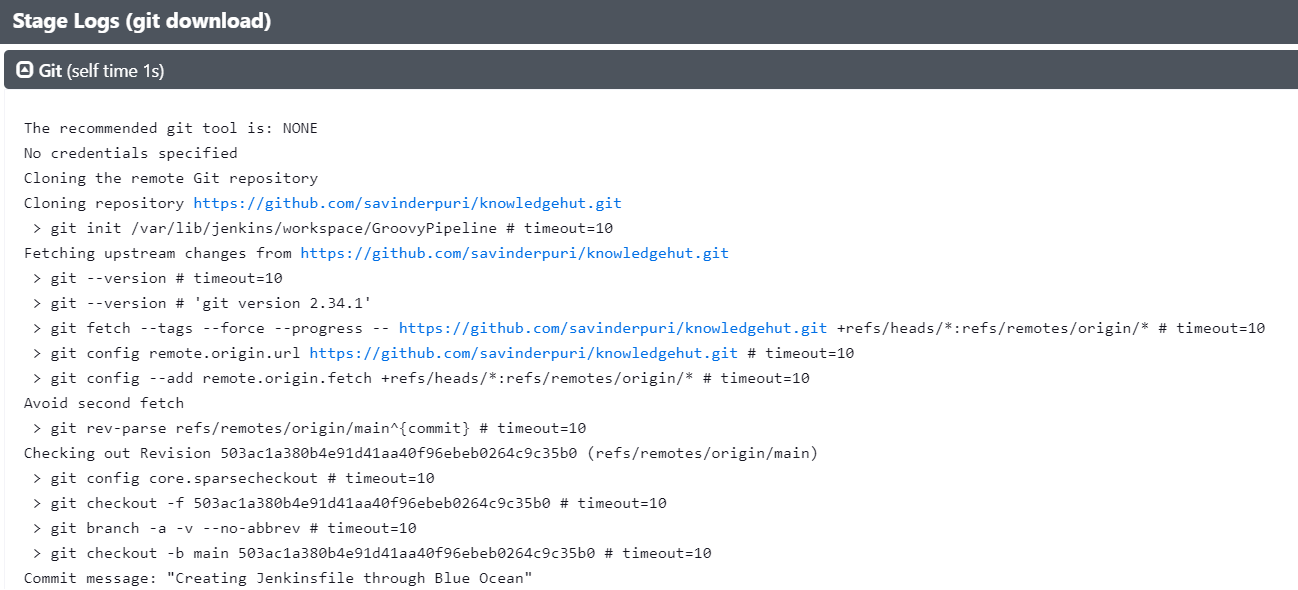
**Added git download as a separate stage in pipeline and named as a “Git download”**





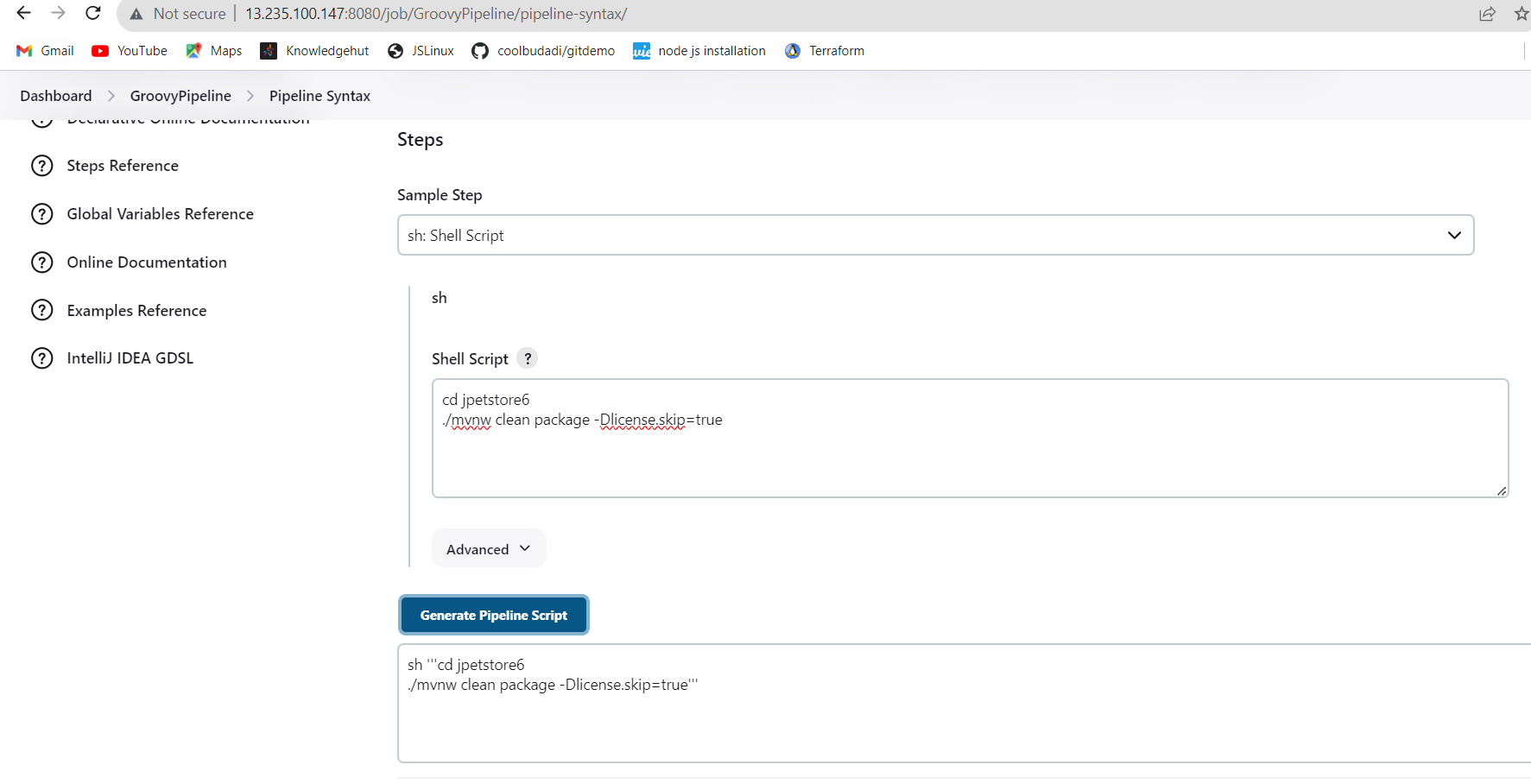
**After built up we got this result as below:**



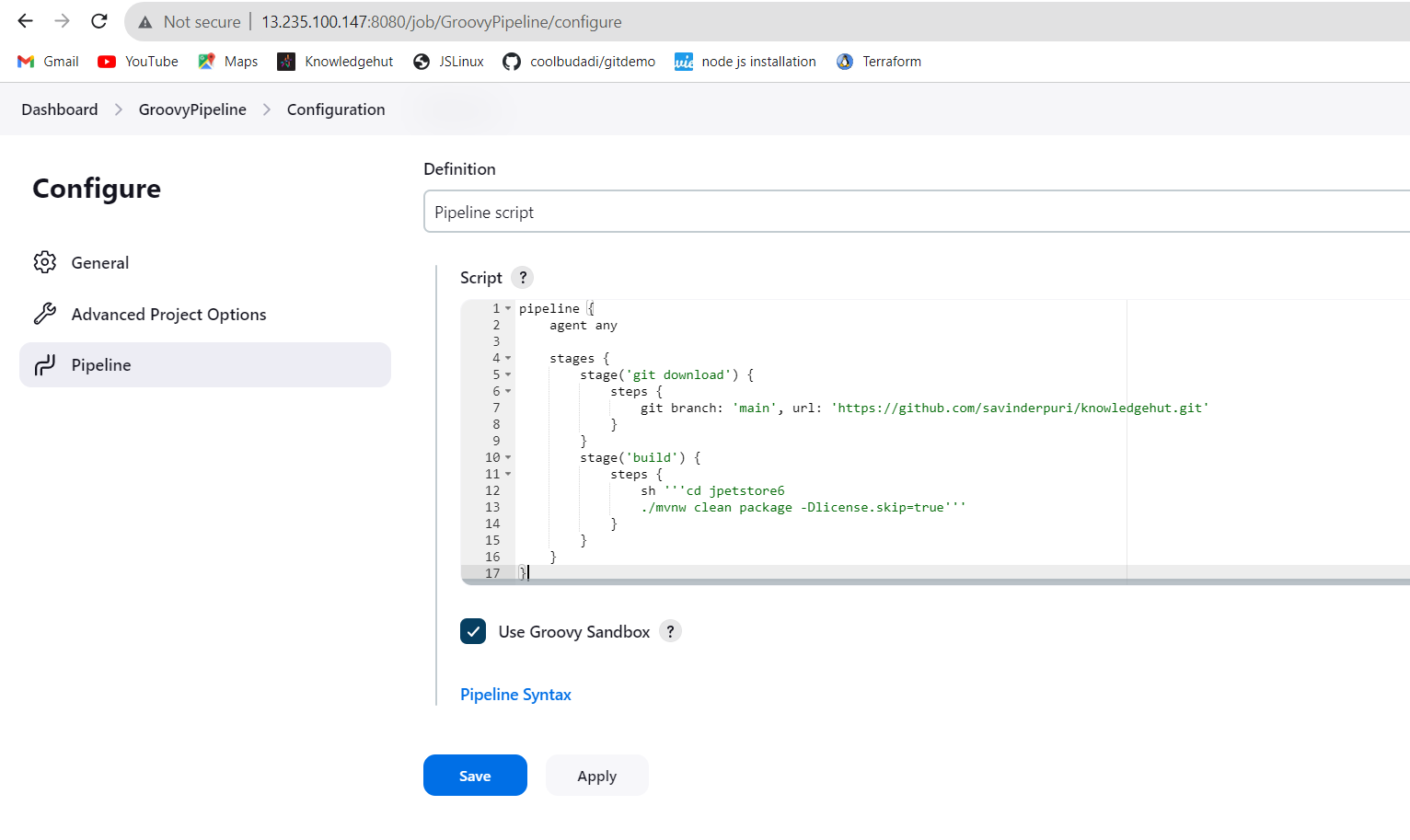


**Next generated shell script that will build the jpetstore application:**

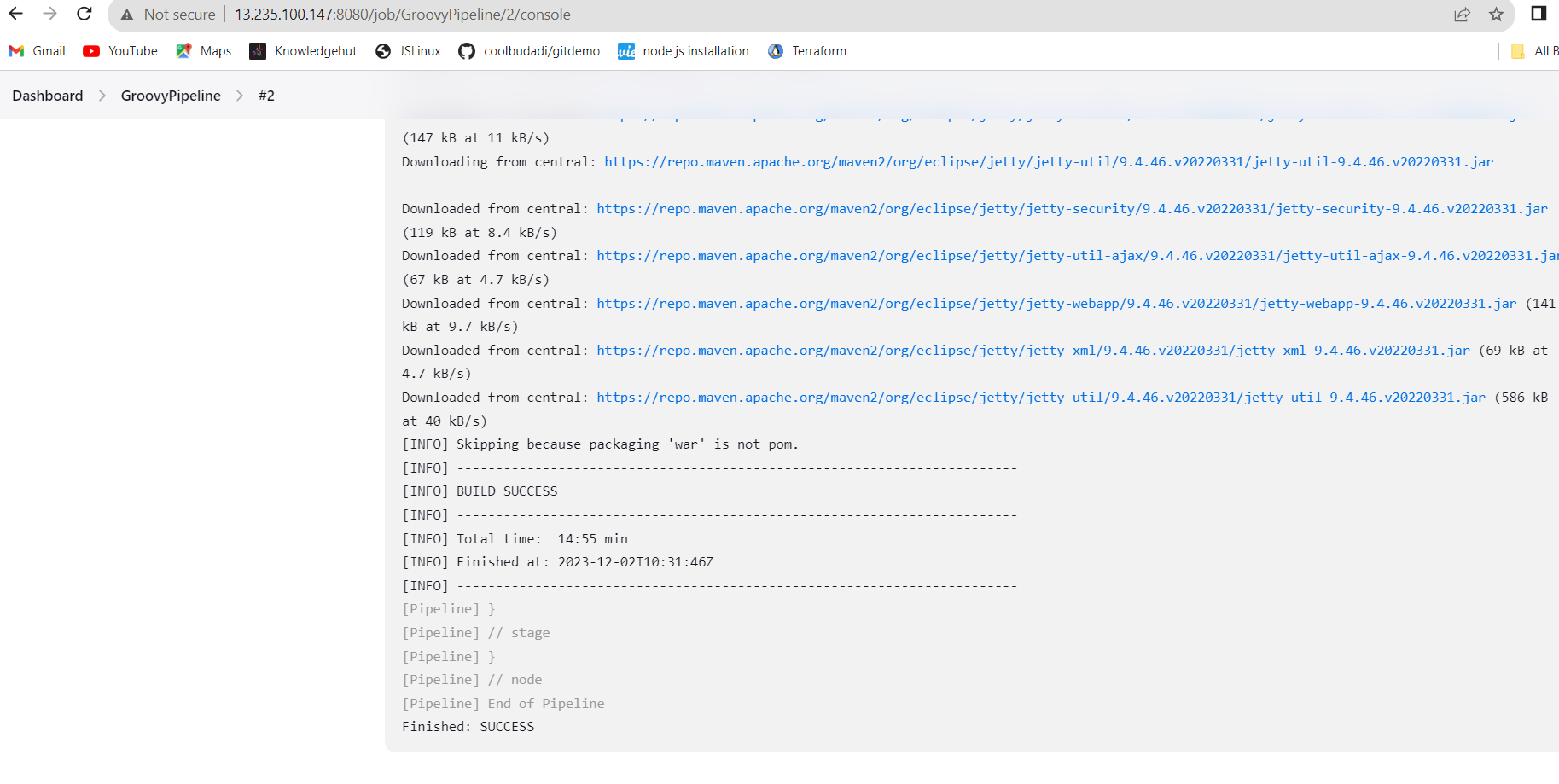
* **cd jpetstore6**
* **./mvnw clean package -Dlicense.skip=true**



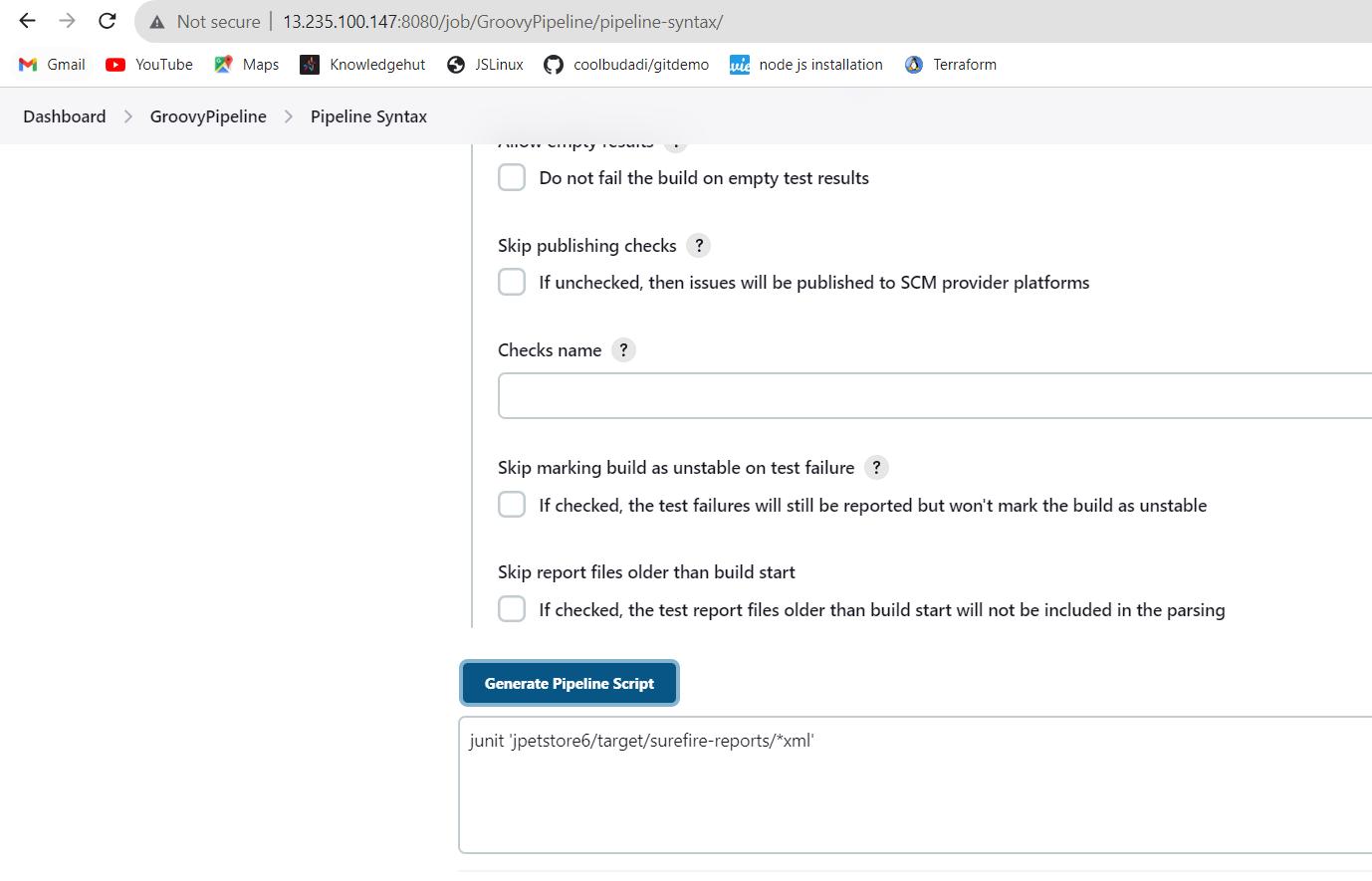
**Genrated script added to the pipeline script as below:**



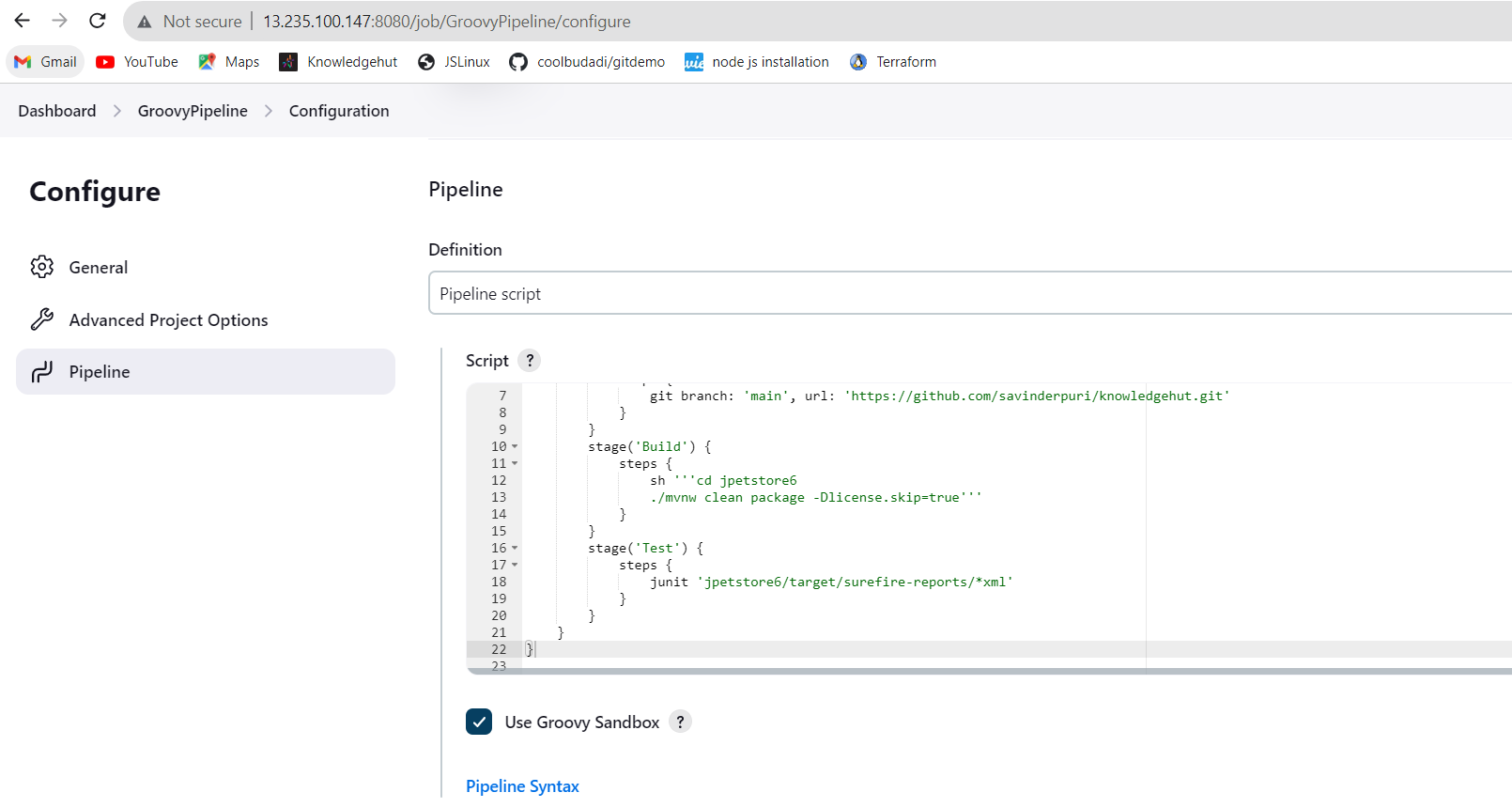
**After built up we got this result as below:**



**Then added JUnit tests to the groovy pipeline. Generated junit: Archive JUnit-formatted test results step like shown on the screenshot below. specified Test report XML’s location to jpetstore6/target/surefire-reports/\*xml**

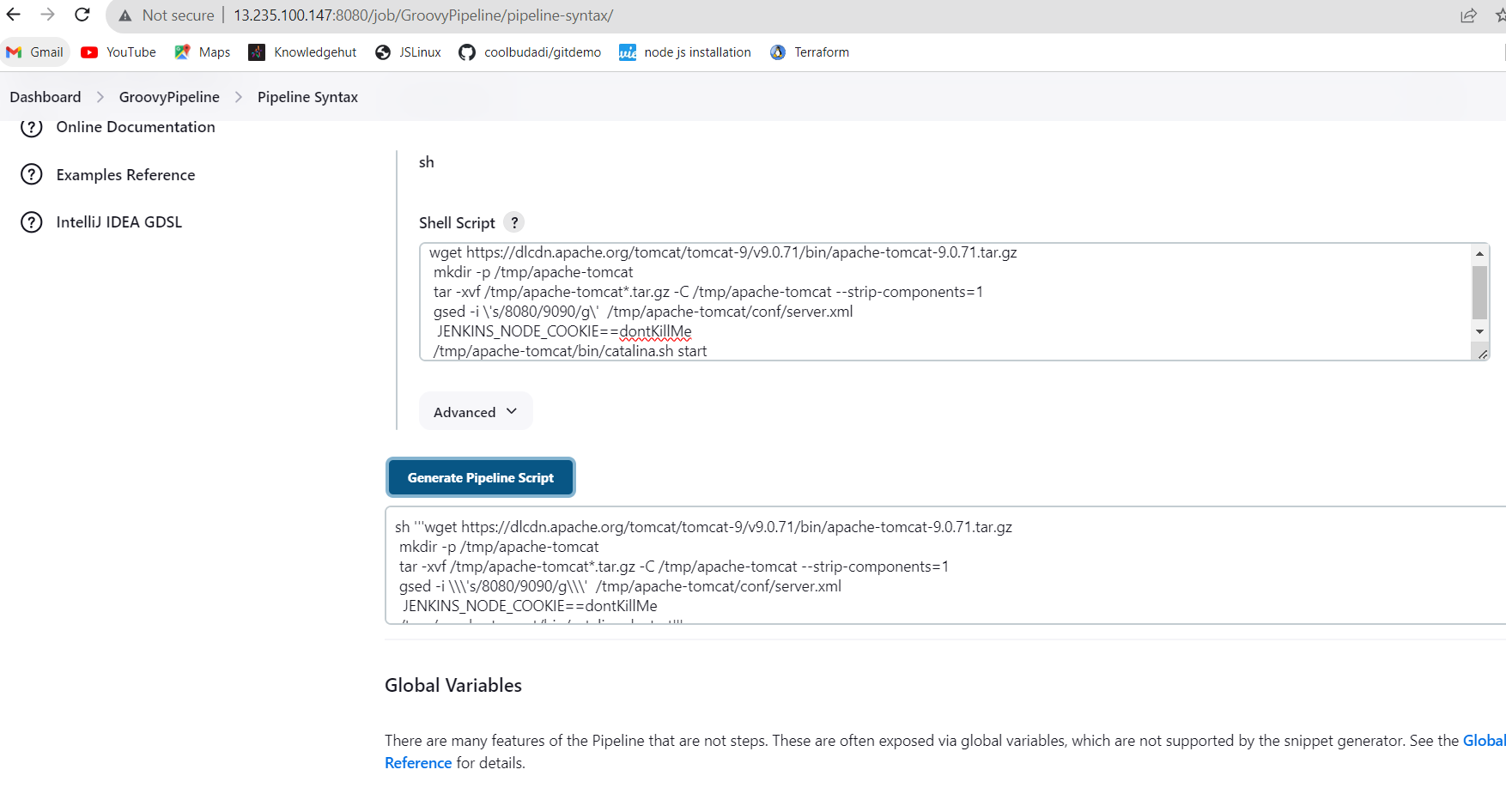


**Added generated code to the pipeline as a separate “Test” stage.**

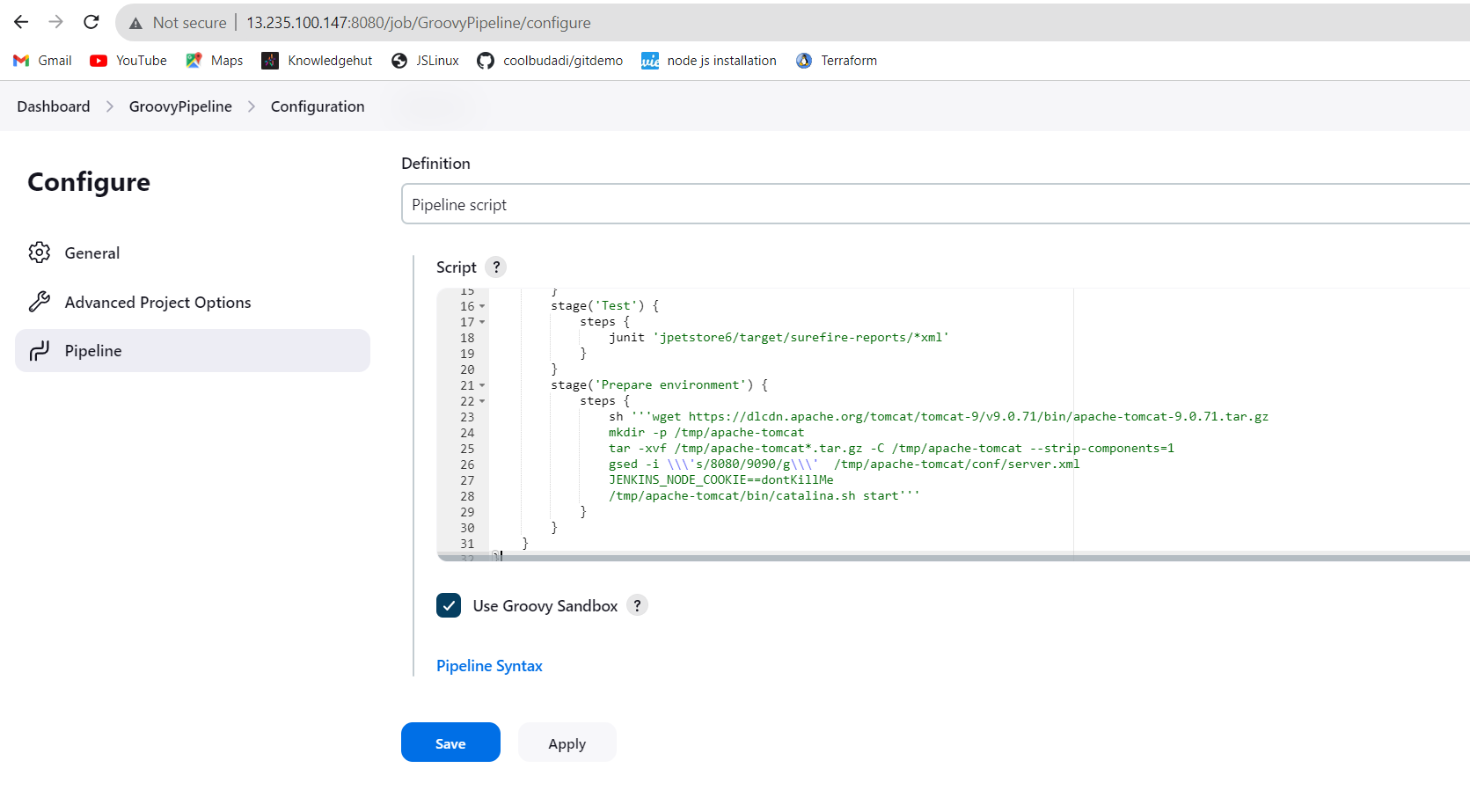


**After this bulid our Continuous Integration and Continuous Testing part is done**

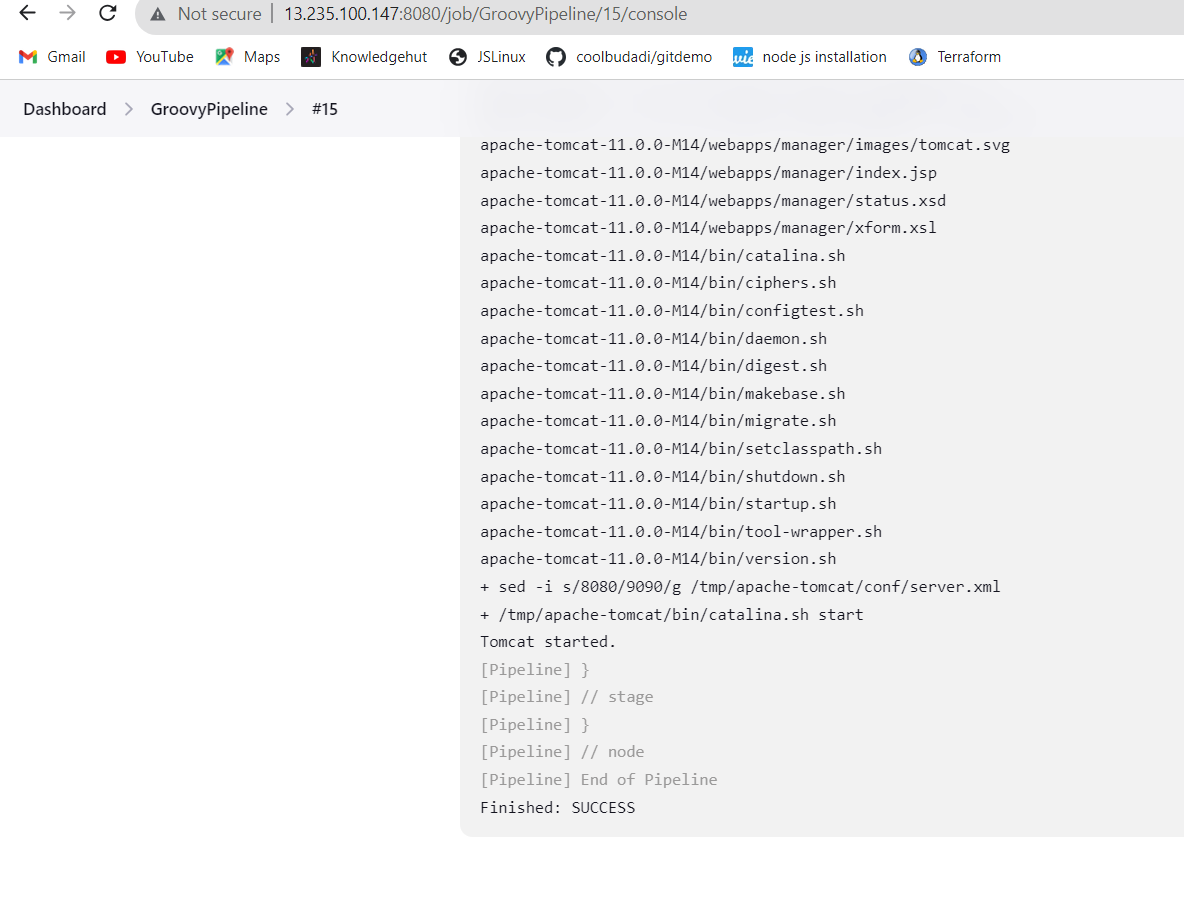
**Now prepared the tomcat environment using shell script as below:**



**Now Added stage “Prepare environment” to pipeline which will make sure tomcat is up and running.**

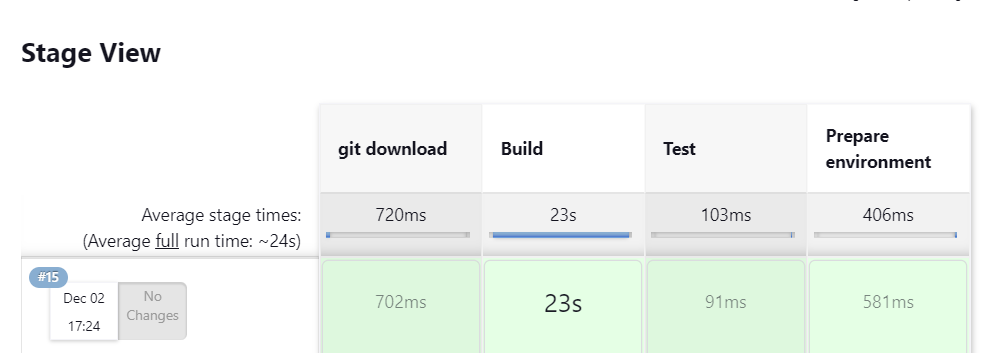


And Tomcat server started as below :

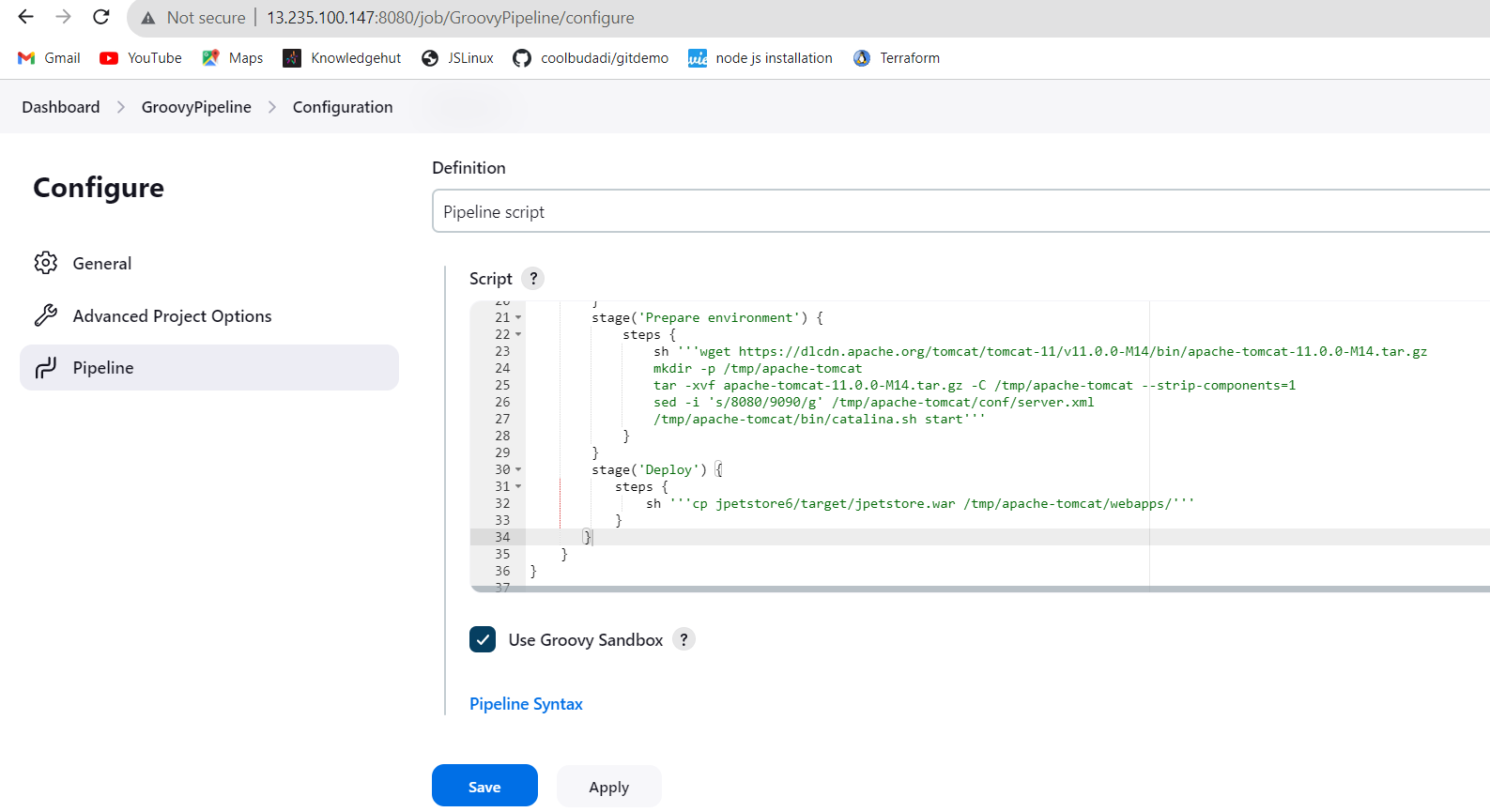


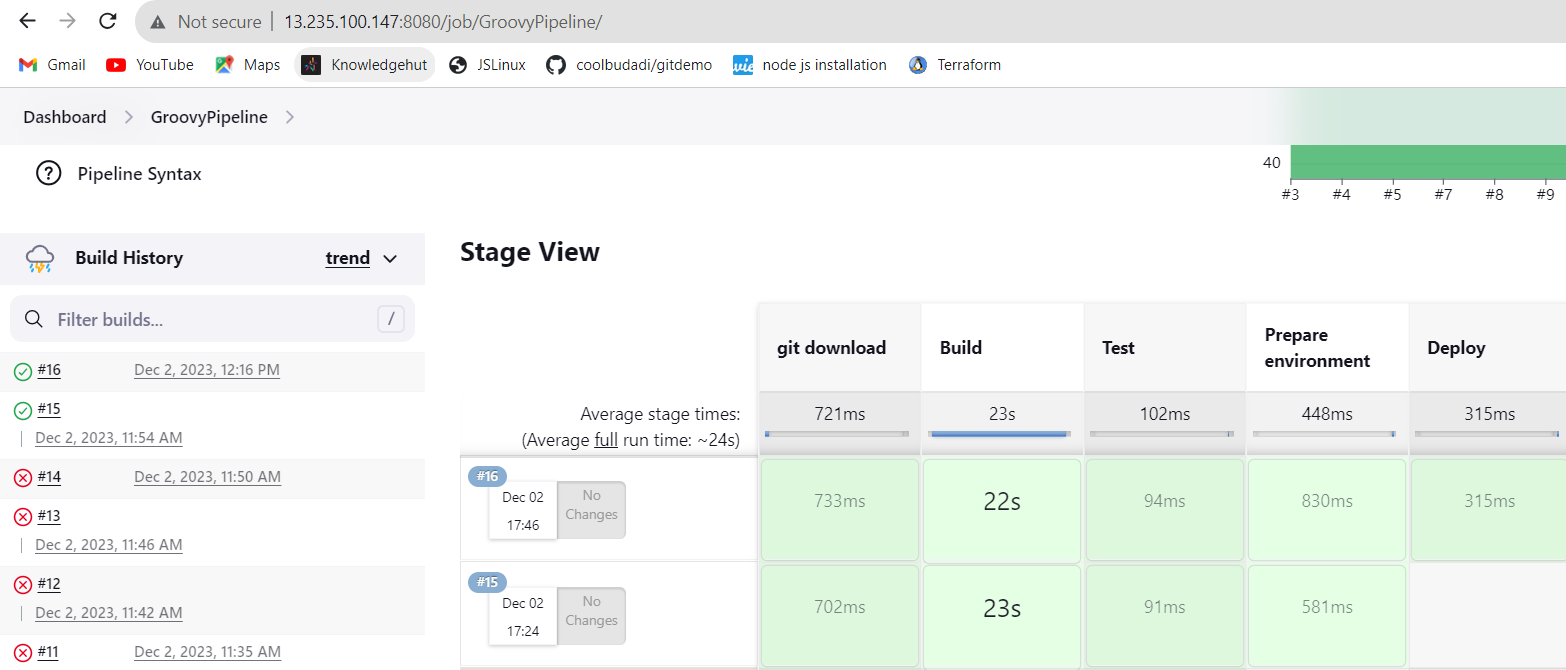
Below is the successful status:

4 stages are successfully build and executed :



Now added “Deploy” stage to the pipeline which will copy jpetstore.war file to the tomcat folder:





Finally application deployed successfully as below :

