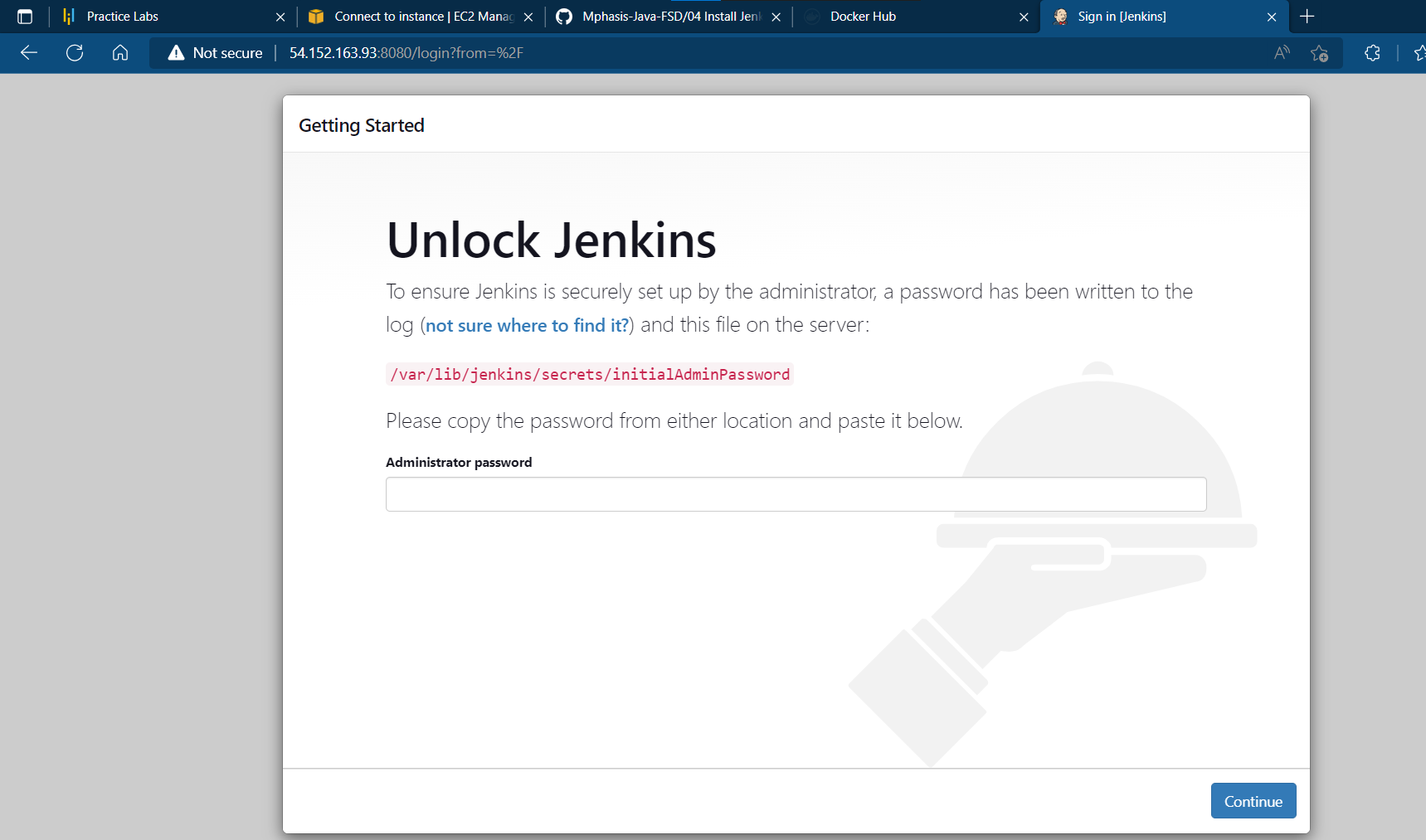
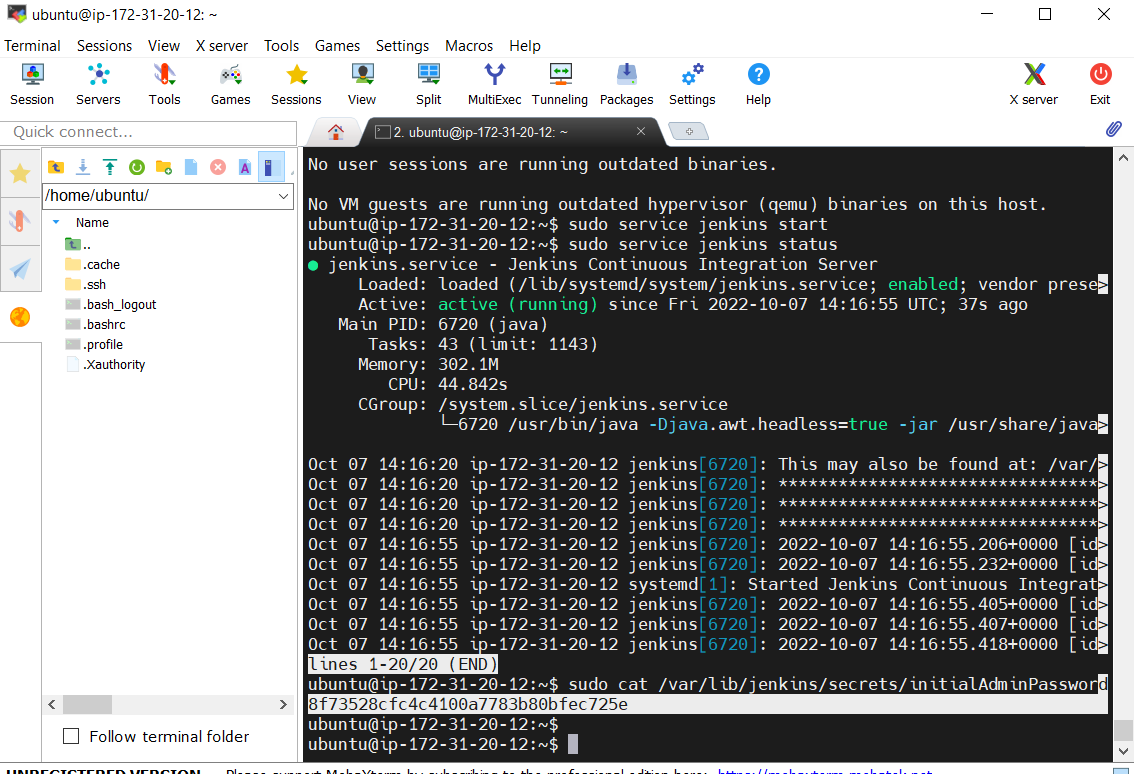
**Screenshots:**

**After creating an instance and installing Jenkins, docker, maven start the Jenkins.**

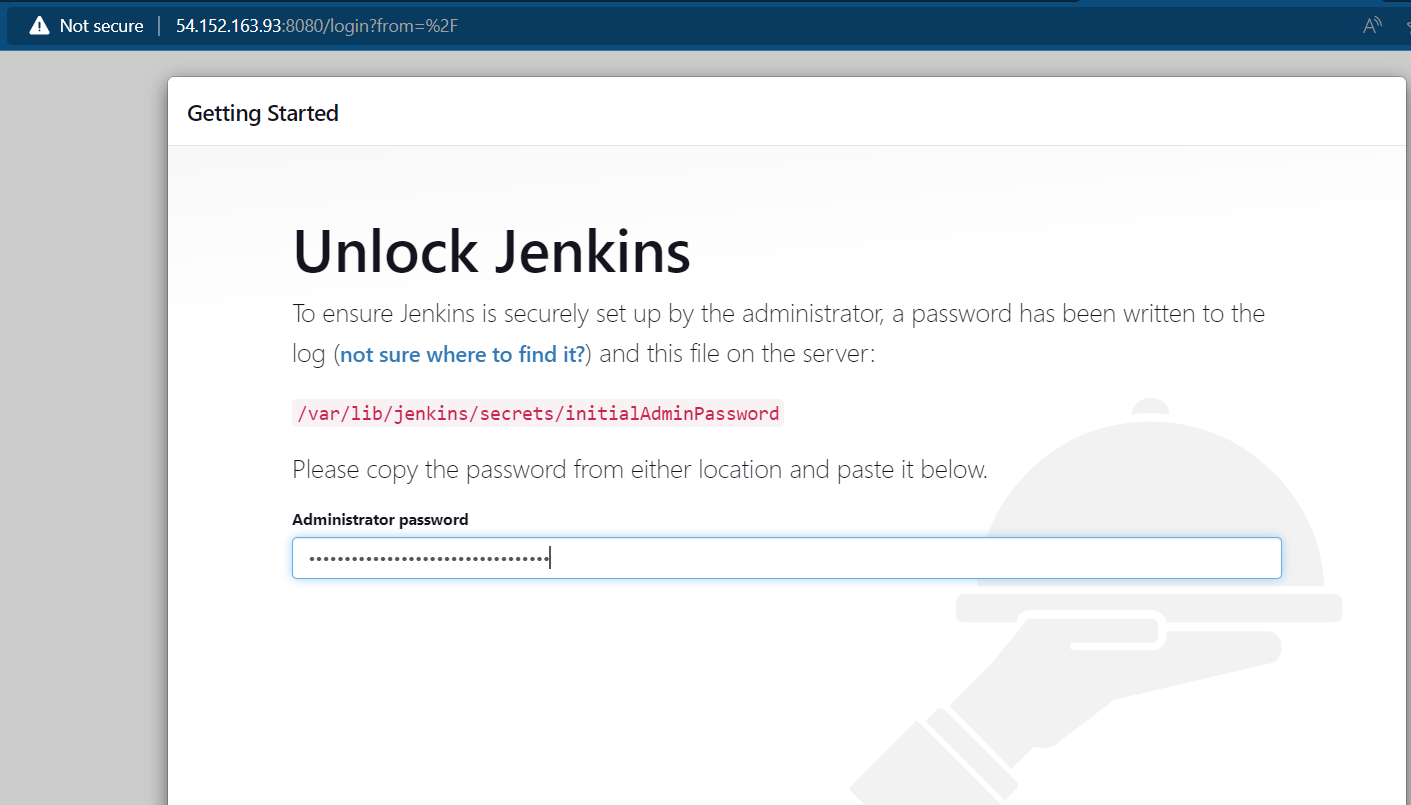
**When Jenkins starts:**

****

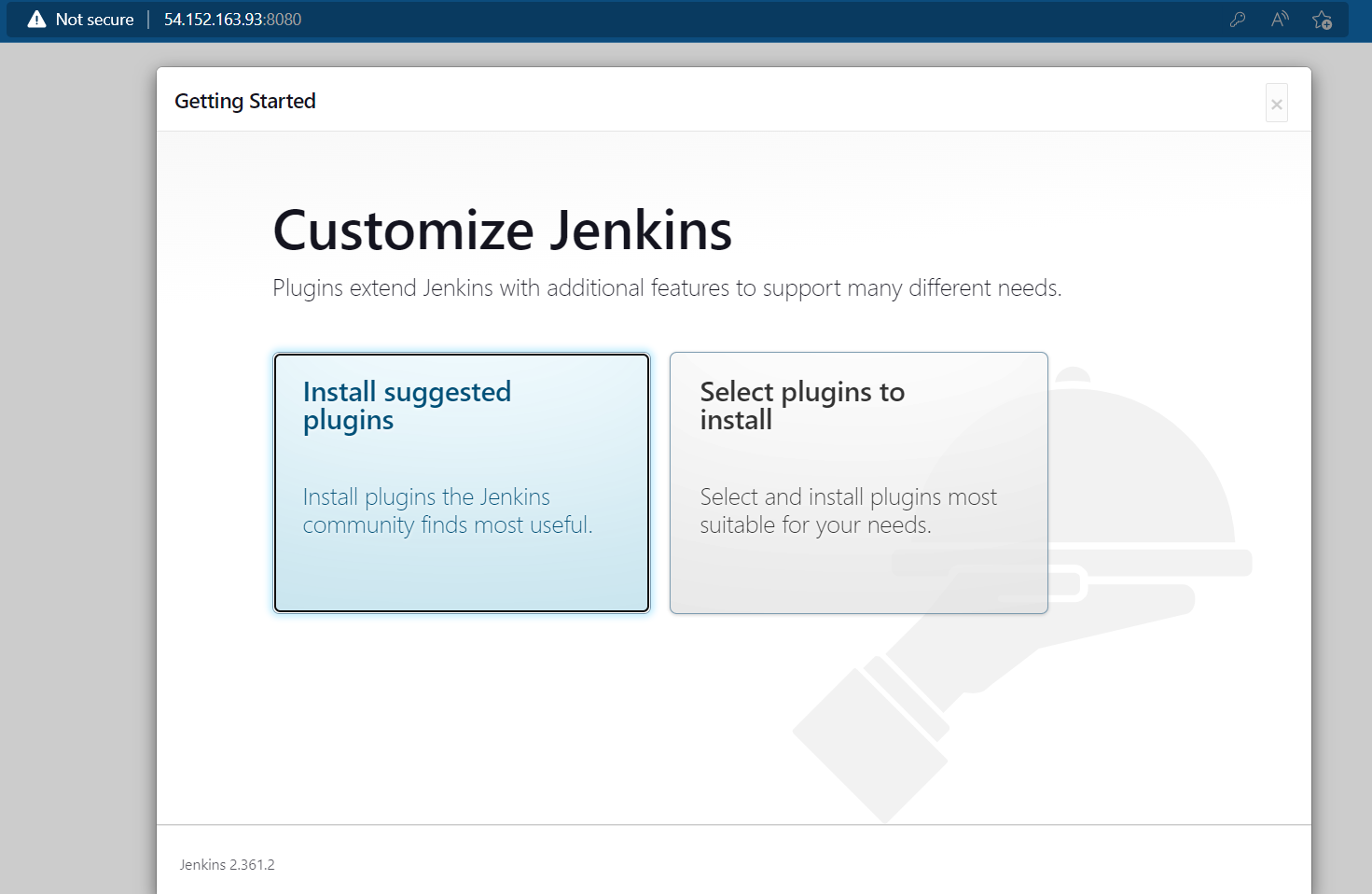
**After starting Jenkins password is generated:**

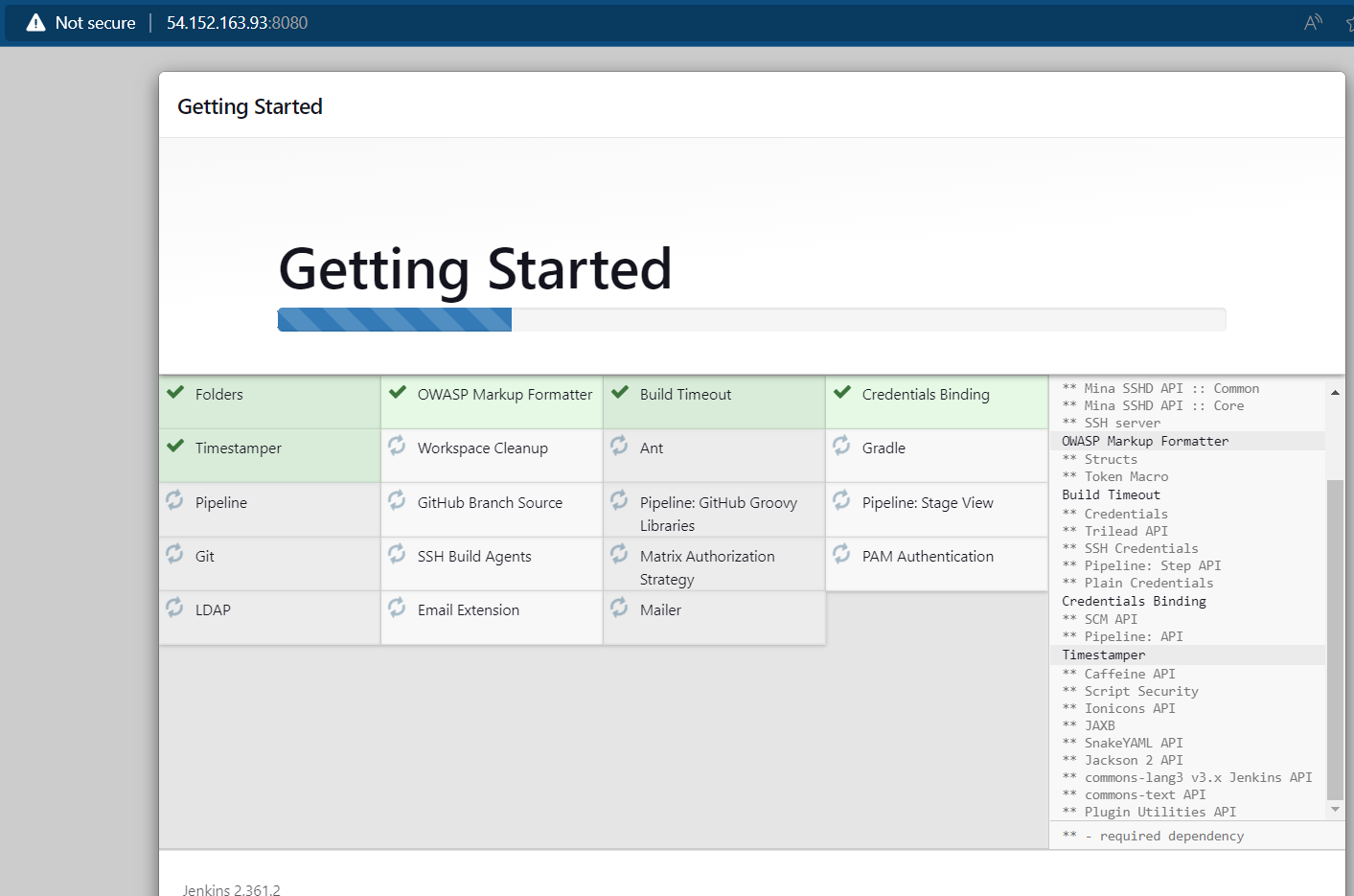
****

**Then it is pasted in the administrative password:**

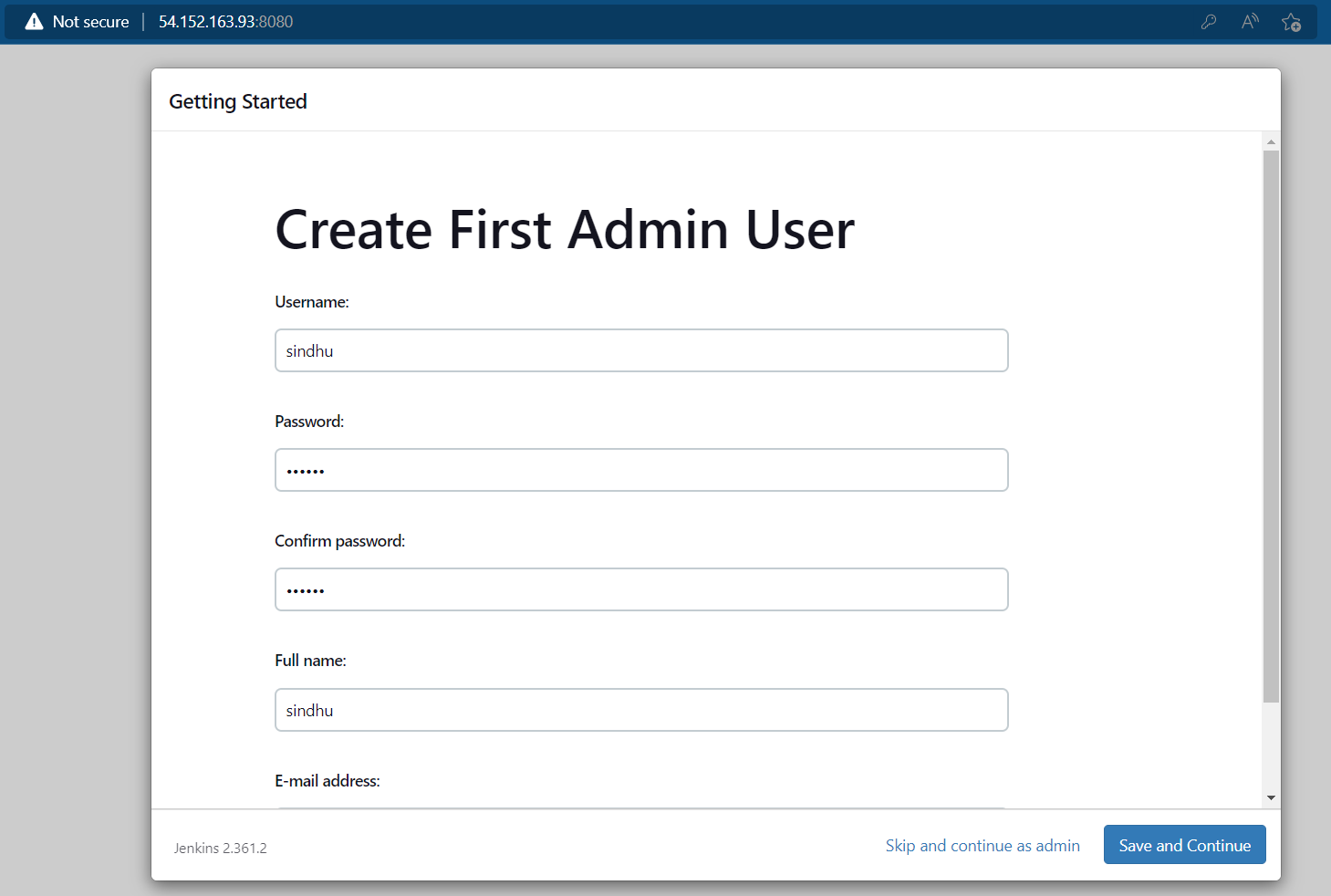
****

**After clicking on continue it goes to:**

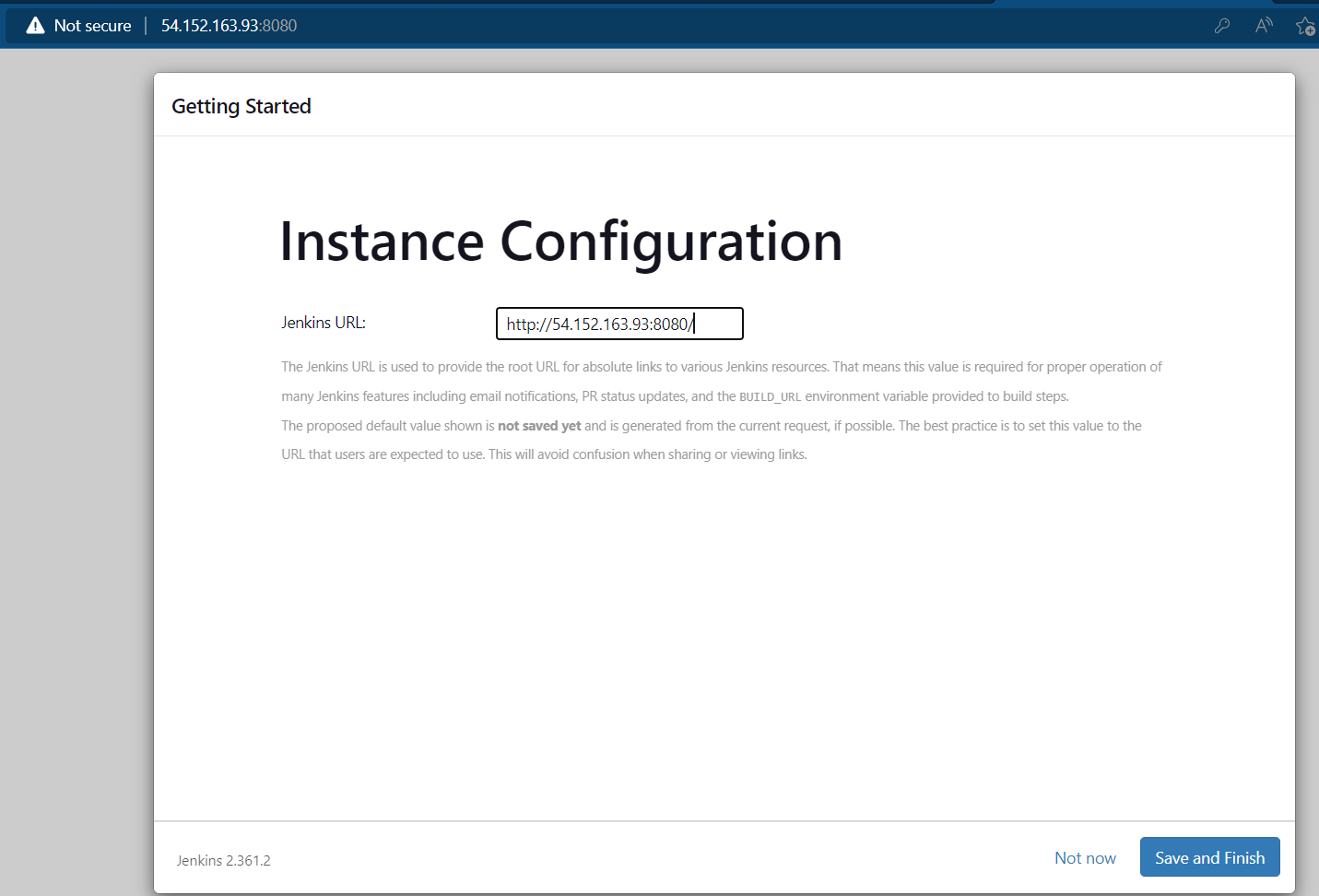
****

****

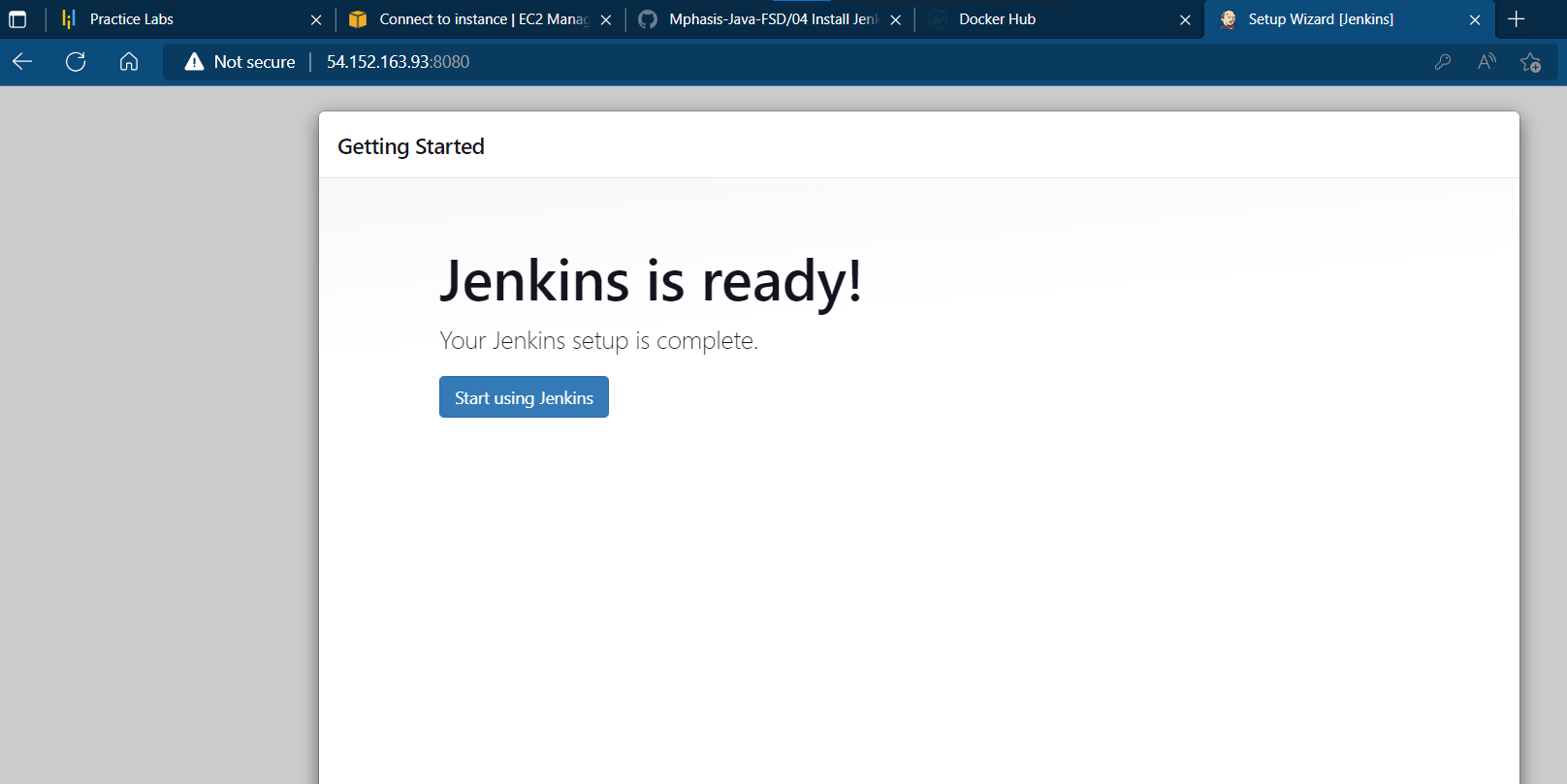
**Then admin user details need to be given:**

****

**After save and continue Jenkins url is created:**

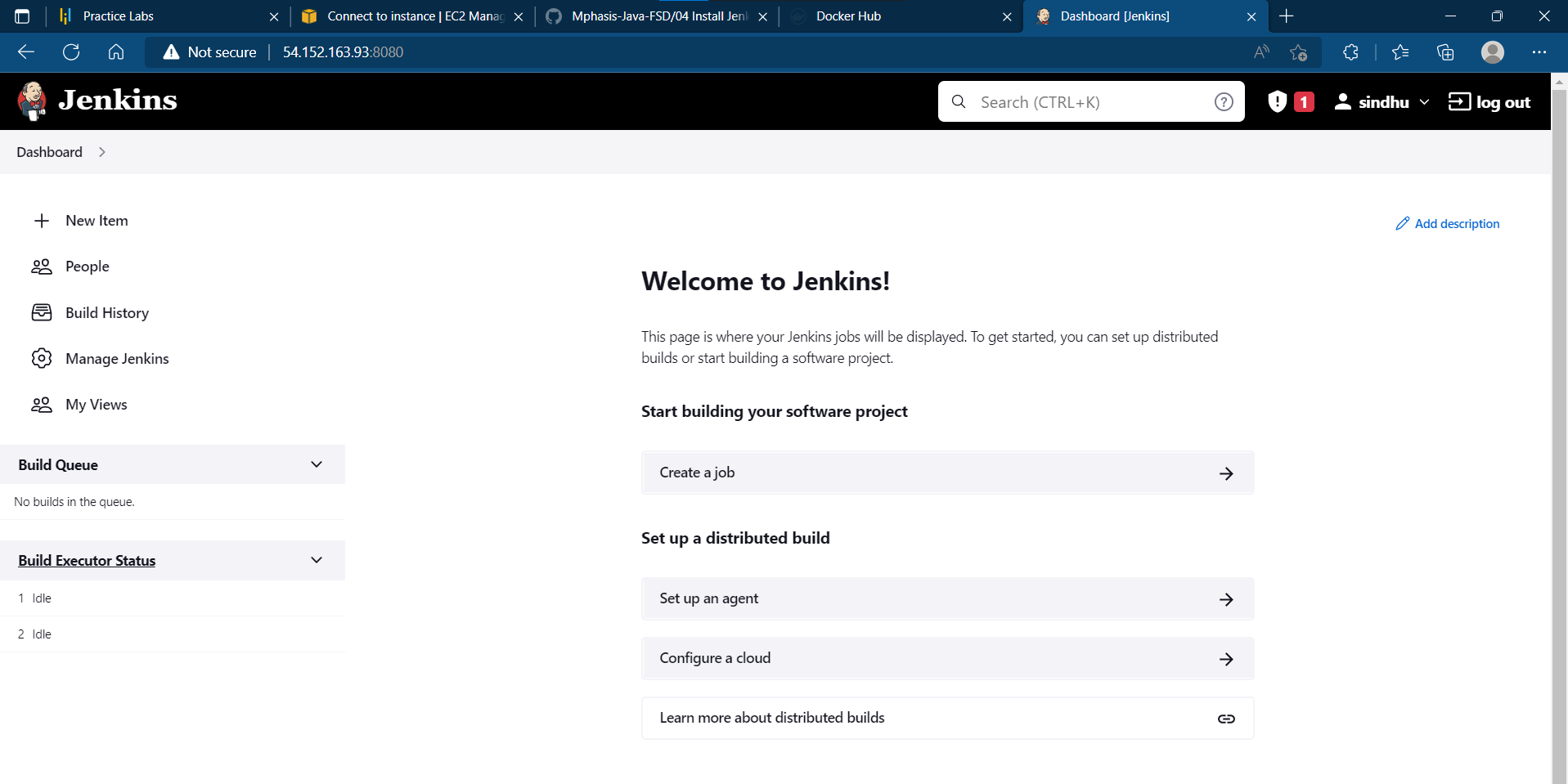
****

**Now Jenkins is ready:**

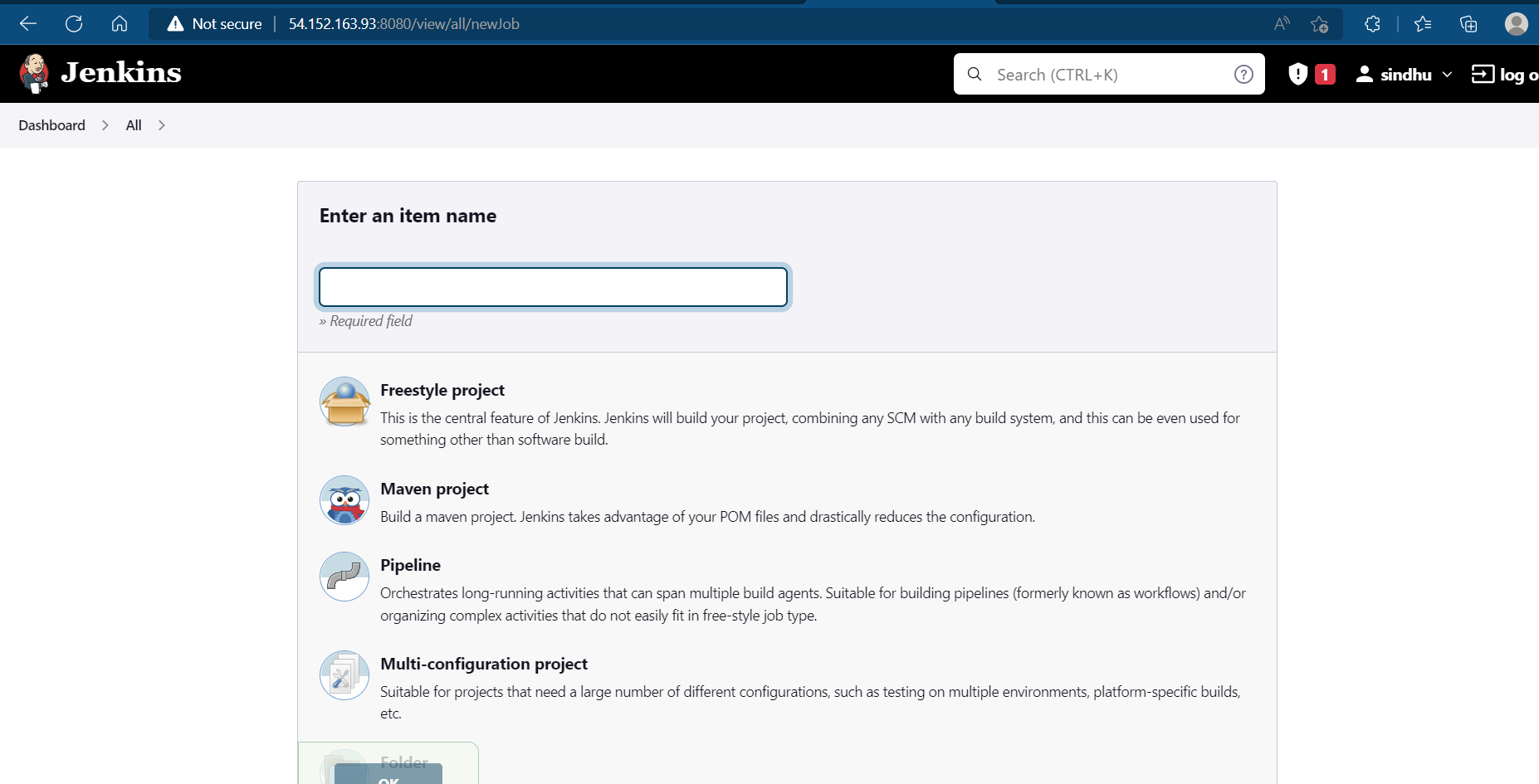
****

**Now Jenkins is opened then click on manage Jenkins >manage plugins > search for maven integration and install it.**

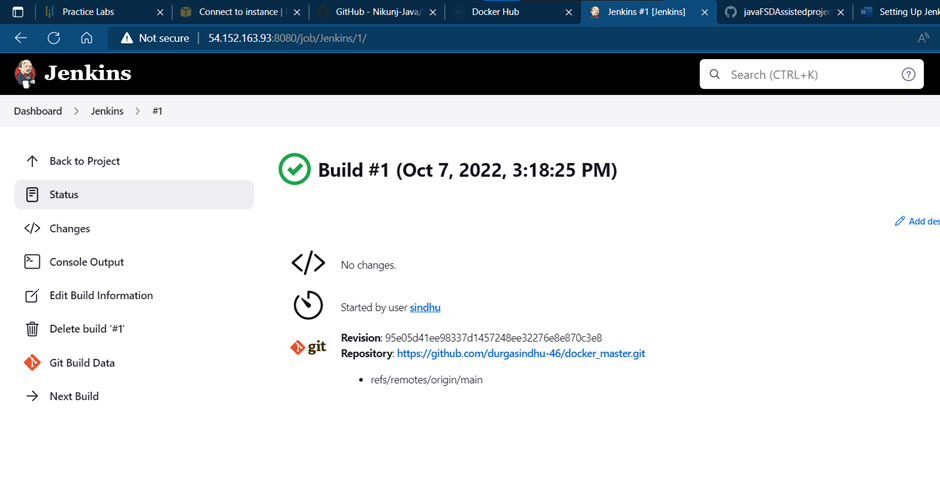
**Now create a new job:**

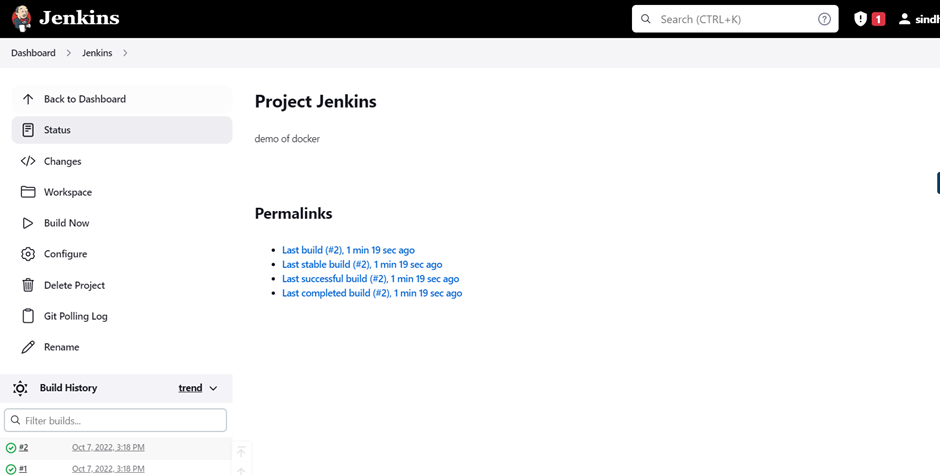
****

**After clicking on create new job enter the name of item:**

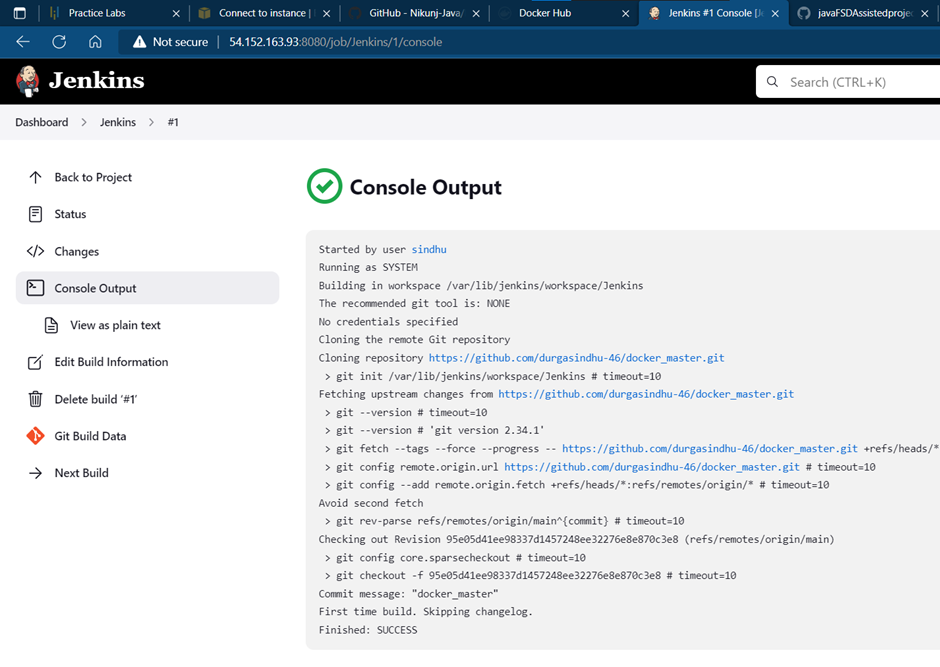
****

**After creating a new project it is build and any changes in the project will make in Jenkins also.**

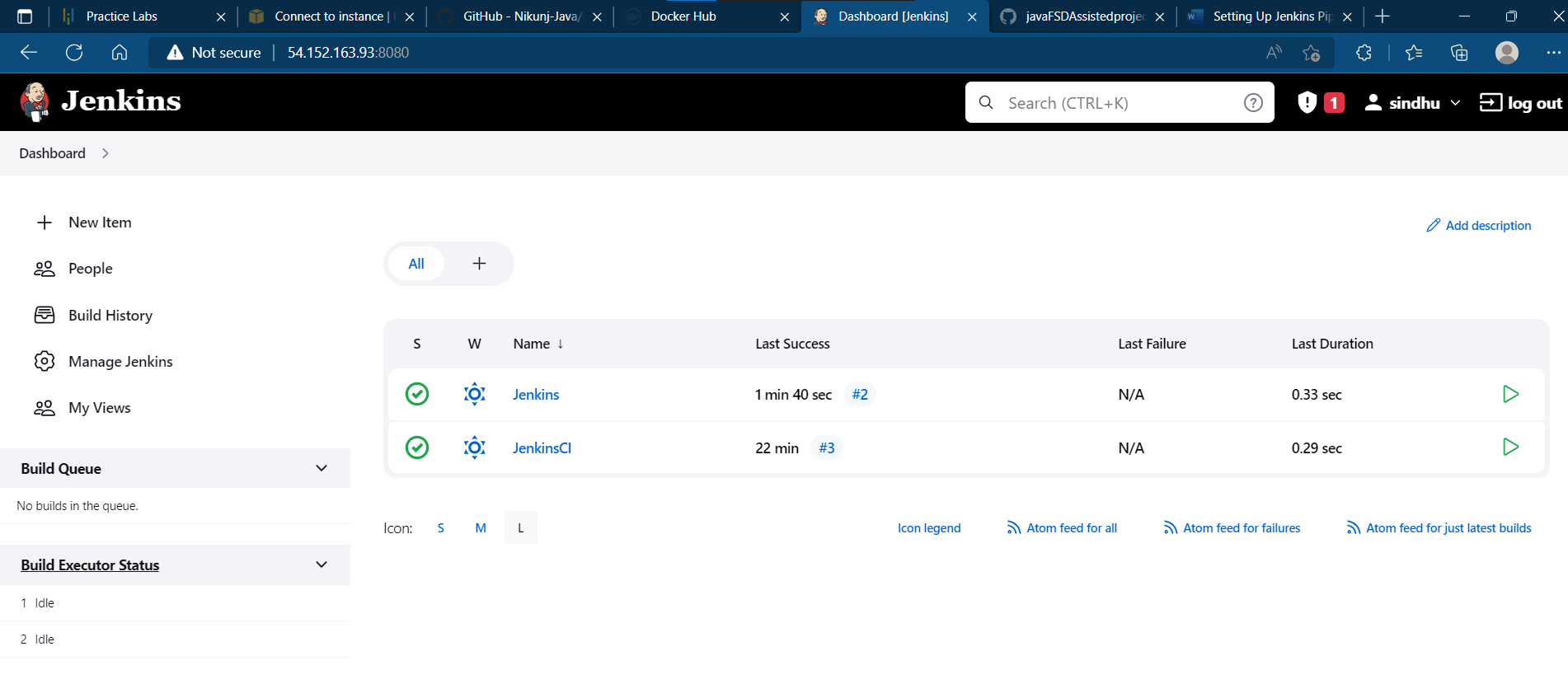
****

****

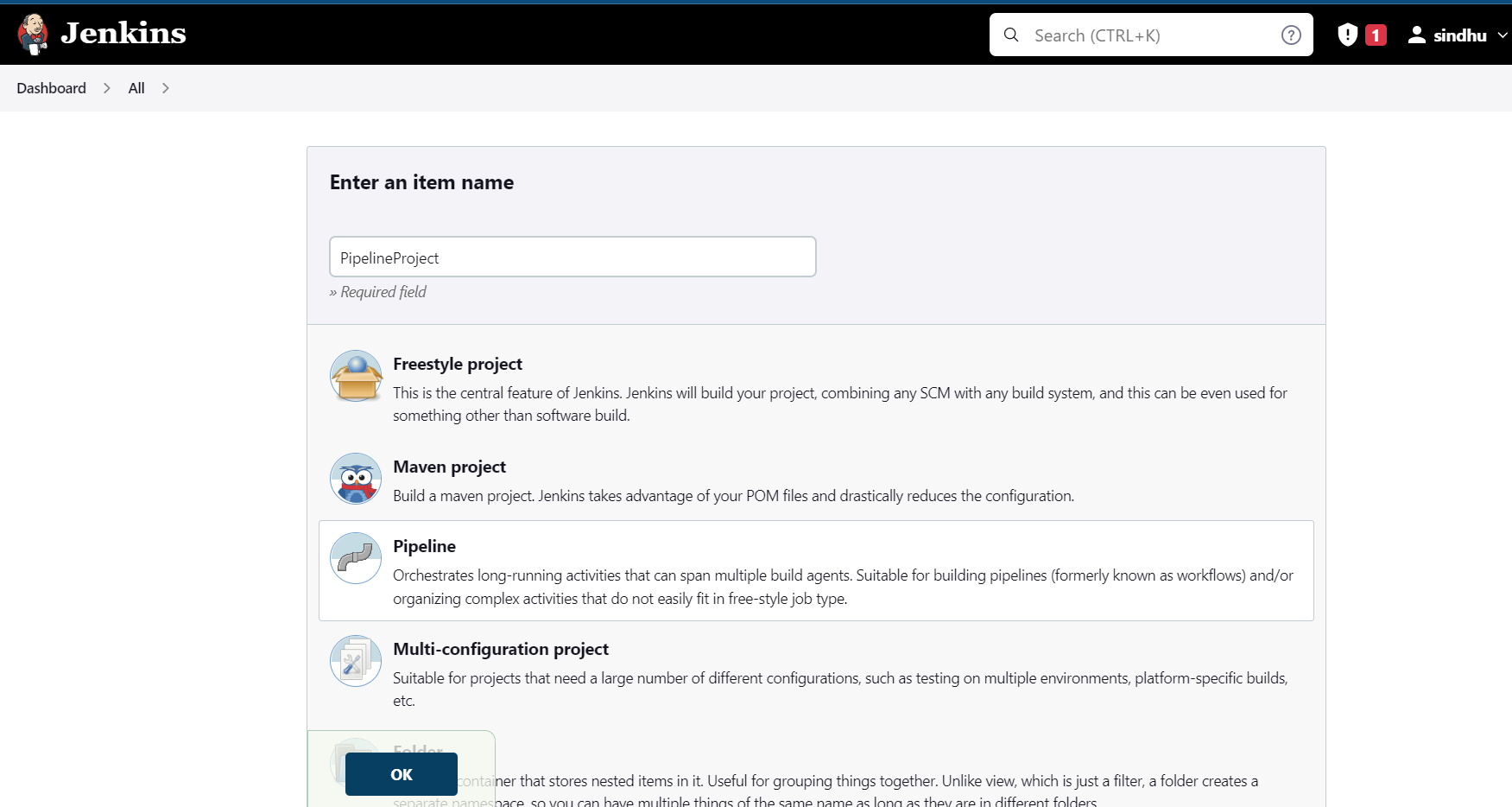
**Console output:**

****

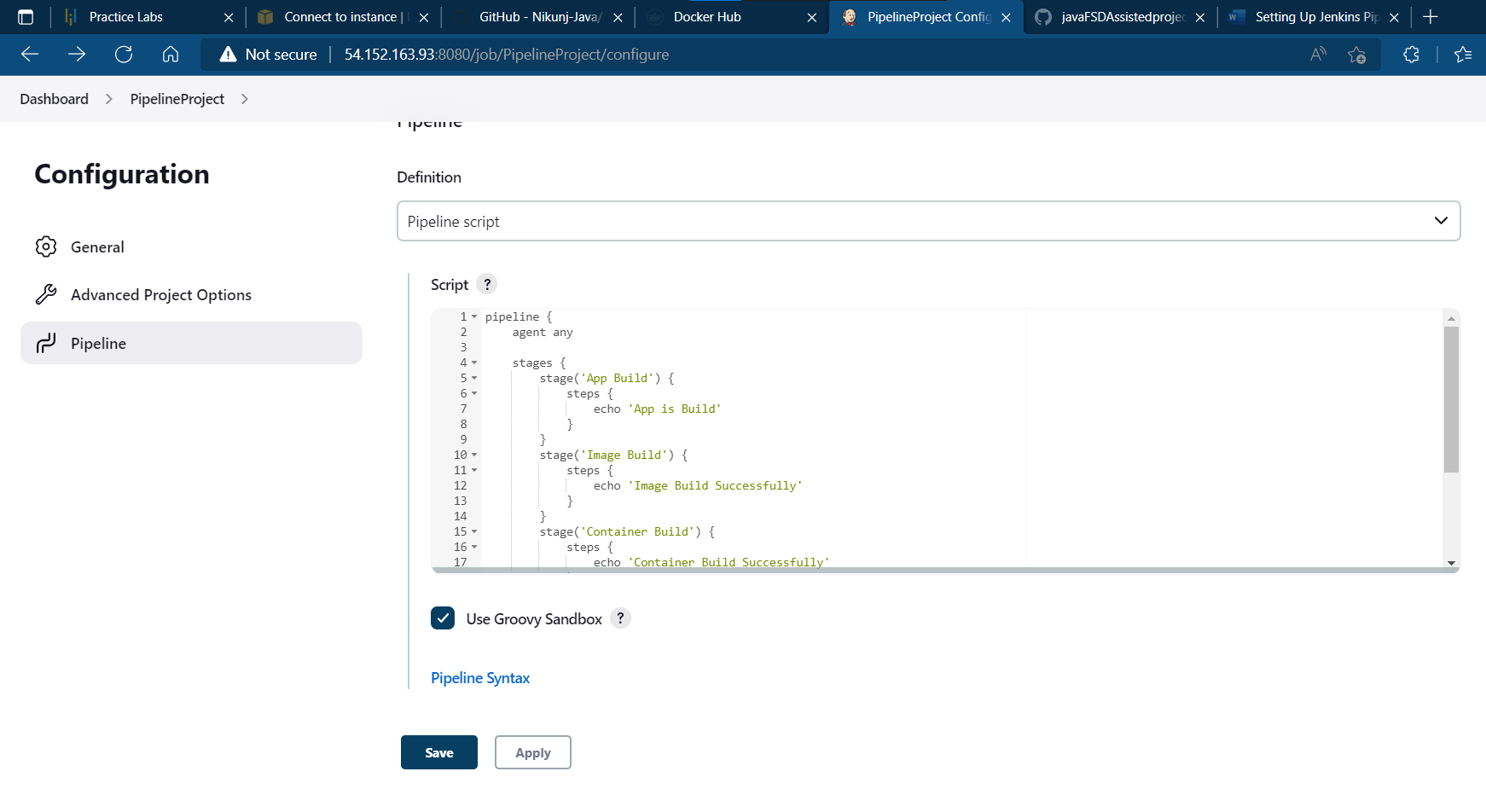
**Build got successful:**

****

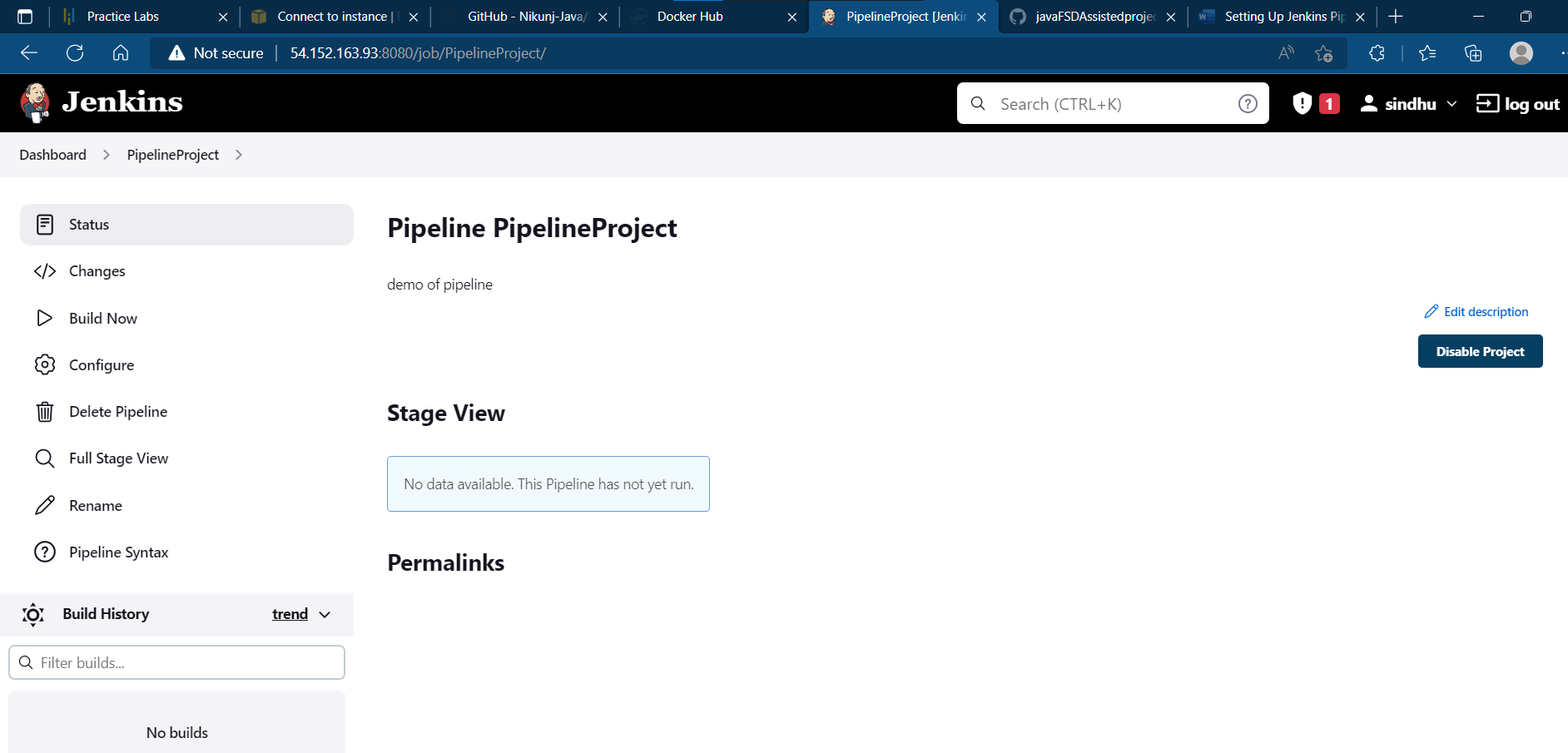
**Pipeline:**

****

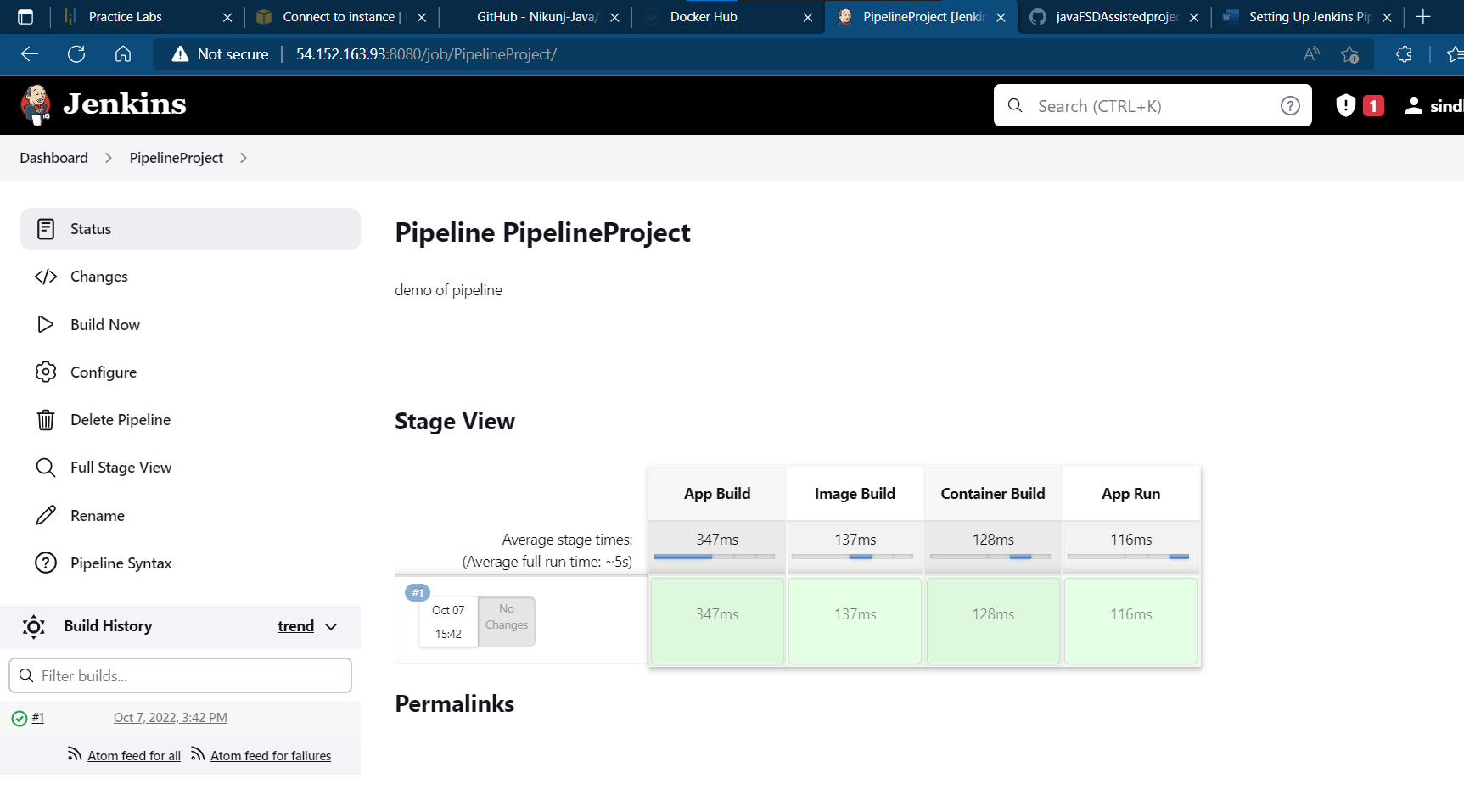
**Pipeline script:**

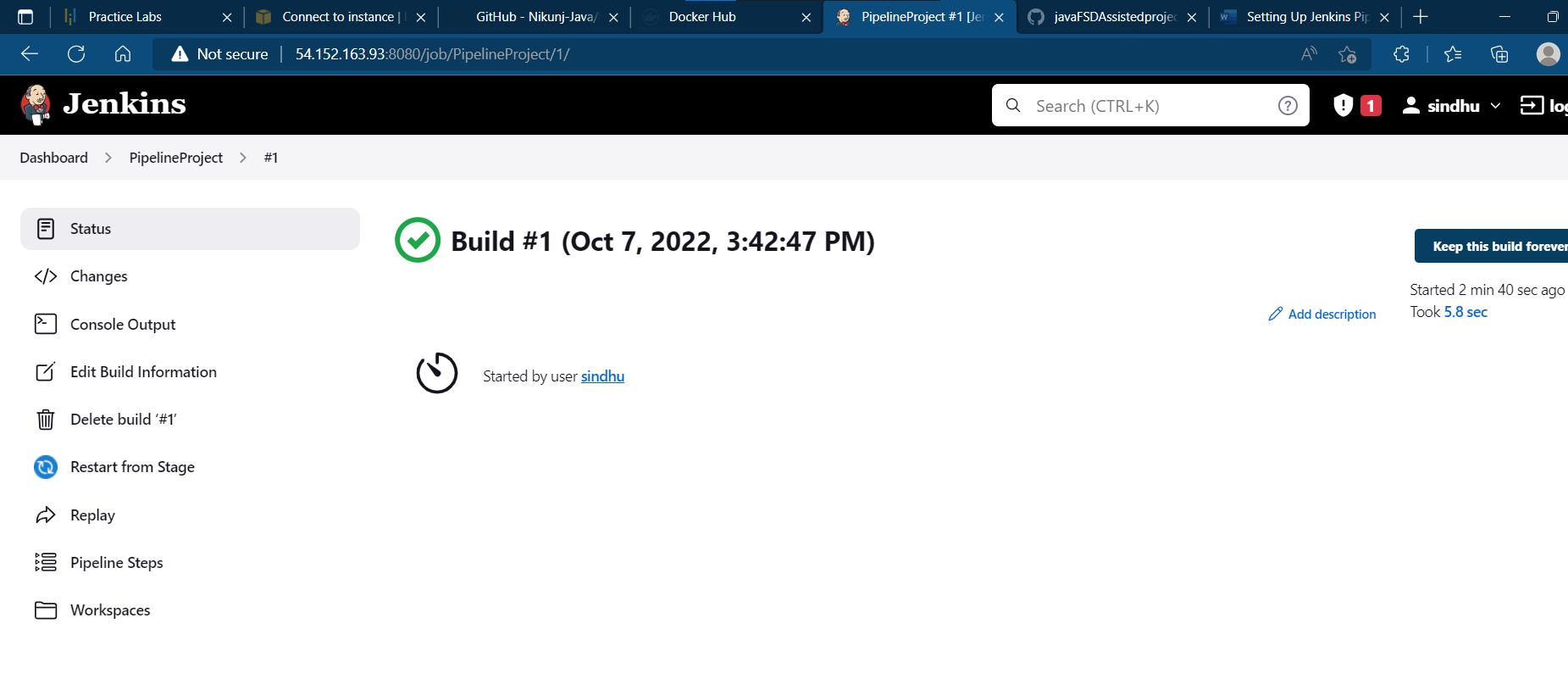
****

**Pipeline project creation:**

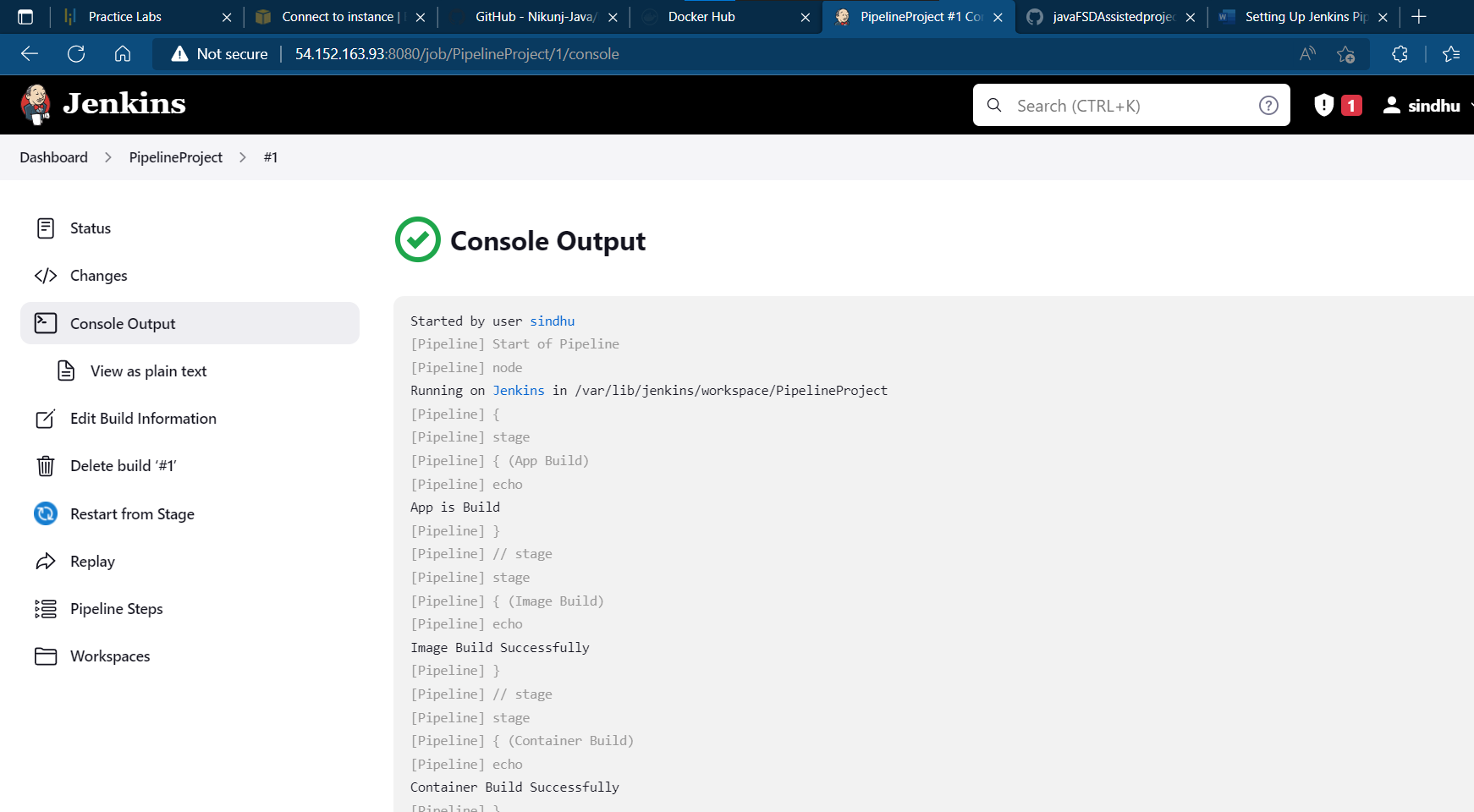
****

**In build:**

****

****

**Console output:**

****

**Output is:**

**Started by user** [**sindhu**](http://54.152.163.93:8080/user/sindhu)

**[Pipeline] Start of Pipeline**

**[Pipeline] node**

**Running on** [**Jenkins**](http://54.152.163.93:8080/computer/(built-in)/) **in /var/lib/jenkins/workspace/PipelineProject**

**[Pipeline] {**

**[Pipeline] stage**

**[Pipeline] { (App Build)**

**[Pipeline] echo**

**App is Build**

**[Pipeline] }**

**[Pipeline] // stage**

**[Pipeline] stage**

**[Pipeline] { (Image Build)**

**[Pipeline] echo**

**Image Build Successfully**

**[Pipeline] }**

**[Pipeline] // stage**

**[Pipeline] stage**

**[Pipeline] { (Container Build)**

**[Pipeline] echo**

**Container Build Successfully**

**[Pipeline] }**

**[Pipeline] // stage**

**[Pipeline] stage**

**[Pipeline] { (App Run)**

**[Pipeline] echo**

**App is Running**

**[Pipeline] }**

**[Pipeline] // stage**

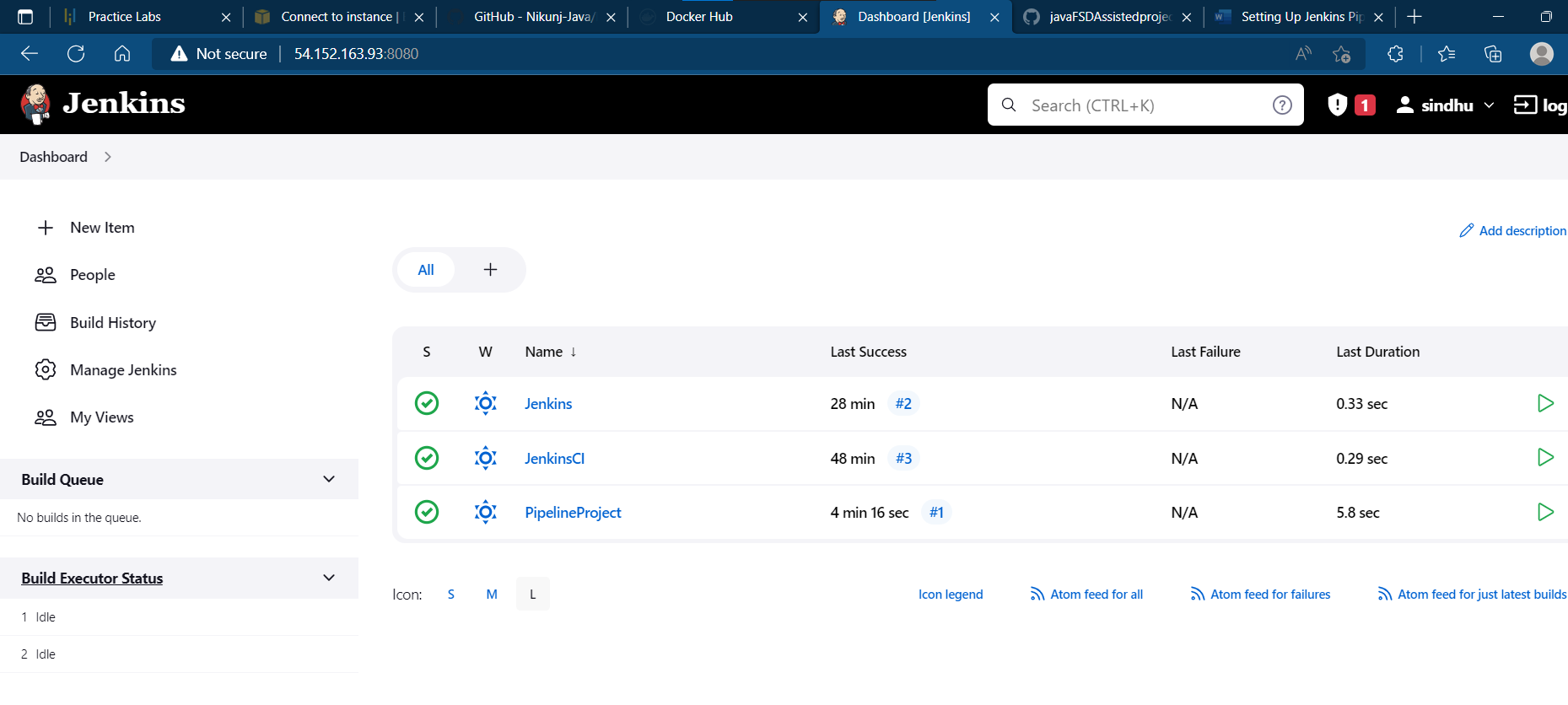
**[Pipeline] }**

**[Pipeline] // node**

**[Pipeline] End of Pipeline**

**Finished: SUCCESS**

**Dashboard:**

****

**Code:**

**Pipeline code:**

pipeline {

agent any

stages {

stage('App Build') {

steps {

echo 'App is Build'

}

}

stage('Image build') {

steps {

echo 'Image build successfully'

}

}

stage('Container Build') {

steps {

echo 'Container Build successfully'

}

}

stage('App Run') {

steps {

echo 'App is running'

}

}

}

}

**Setting Up Jenkins Pipeline to Deploy Docker Swarm.**

DESCRIPTION

**Project objective:**

You have to develop an environment for Docker networking.

**Background of the problem statement:**

As you have worked on Docker containers previously, your manager has asked you to perform container scheduling over multiple hosts using Docker CLI and connect multiple hosts with Docker containers.

**You must use the following:**

● Jenkins: To create a pipeline to deploy Docker Swarm  
● Docker Swarm: To implement container networking  
● Git: To connect and push files from the local system to GitHub   
● GitHub: To store the Angular application

**Following requirements should be met:**

● A few of the source code should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.  
● Submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.  
● The step-by-step process involved in completing this task should be documented.

**Solution:**

**Steps to run this application**:

**Step:1 Create Aws Ubuntu Instance With Port No:8080**

1. Prepare AWS Instance (Ubuntu Server 22.04 LTS (HVM), SSD Volume Type)

2. Security: Add port no: 80 with Custom TCP Rule and ssh with 22 port number.

3. Then create an instance and connect it.

4. Download .pem Key and connect using Moba x-term open moba x-term

5. Then in moba x-term give the path and paste the .pem file.

6. copy the ssh example command into moda x-term.

**Step:2 Install Docker**

Link to install: <https://docs.docker.com/engine/install/ubuntu/>

Give below commands in moba x-term:

>sudo apt-get update

>sudo apt-get install ca-certificates curl gnupg lsb-release

>sudo mkdir -p /etc/apt/keyrings

>curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

>echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) 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 docker-compose-plugin

**To verify installation:**

> sudo docker –v

**Step:3 Install Jdk**

> sudo apt-get update

> sudo apt install default-jdk -y

**Step:4 Install Maven**

> sudo apt-get update

> sudo apt install maven -y

**Step:5 Install Jenkins**

goto> google> how to download jenkins in ubuntu

link: https://www.jenkins.io/doc/book/installing/linux/

>curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.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-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

>sudo apt-get update

>sudo apt-get install Jenkins

**Start Jenkins:**

> sudo service jenkins start

> sudo service jenkins status

CONNECT: goto> AWS>EC2>Copy Public IP:8080 on browser hit enter

**To generate secret password**

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

---To deal with permission denied error----

> sudo chmod 777 /var/run/docker.sock

copy the secret password to jenkins and start installtion

**For continuous integration (ci)**

1. click on install suggested plugins

2. provide your credentials

3. welcome to jenkins

4. manage plugin>maven integration> click on install without restart

5. goto>dashboard>create>new job>select freestyle project> give name>click ok

6. give description

7. In source code management->git->

Url: https://github.com/durgasindhu-46/docker\_master

Branches to build: \*/master or \*/main

8. Build Trigger click pollscm and give H/2 \* \* \* \*

9. Click on apply and save

10. Build the project

**For creating a pipeline:**

1. click on install suggested plugins

2. provide your credentials

3. welcome to jenkins

4. manage plugin->maven integration-> click on install without restart

5. goto>dashboard>create>new job>select pipeline> give name>click ok

6. give description

7. write pipeline script

8. Click on apply and save

9. Build the project