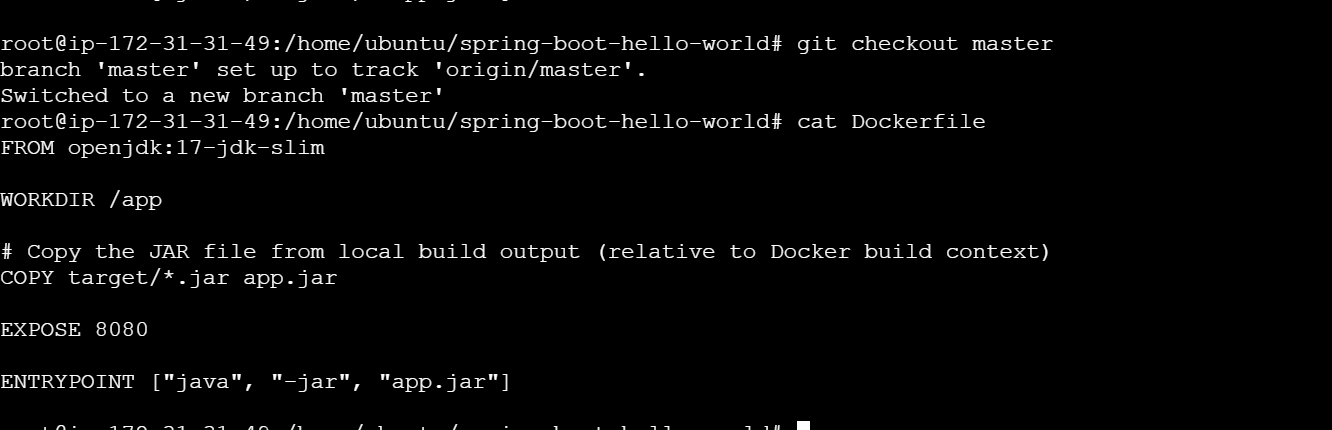
**Devops exam**

**Repo url:** https://github.com/Abhiramikannan/spring-boot-hello-world.git

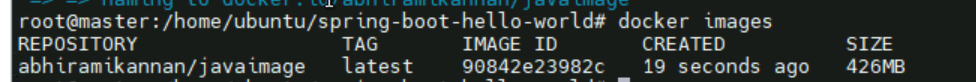
* **Created a Dockerfile**



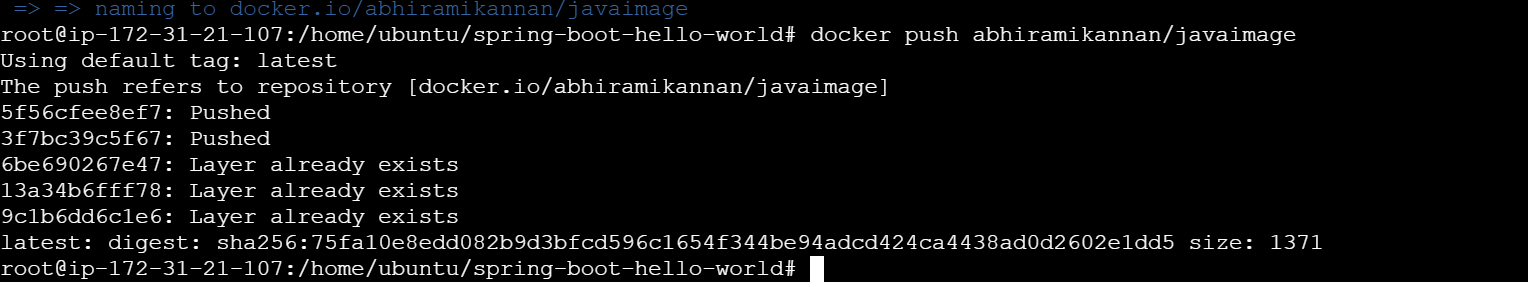
* **builded image**

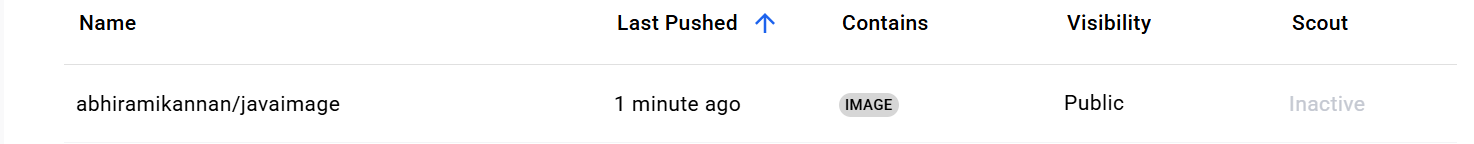
docker build -t abhiramikannan/javaimage .

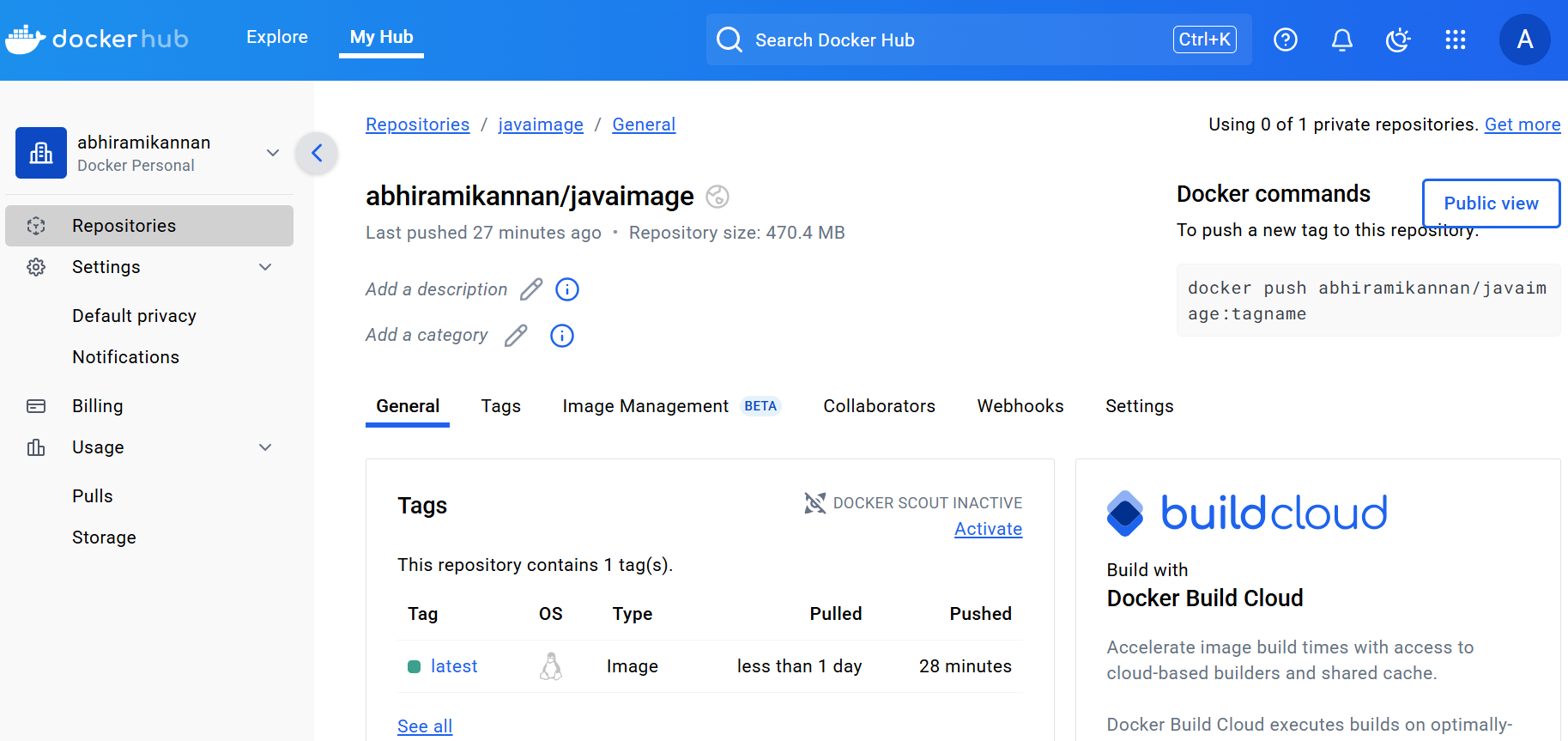




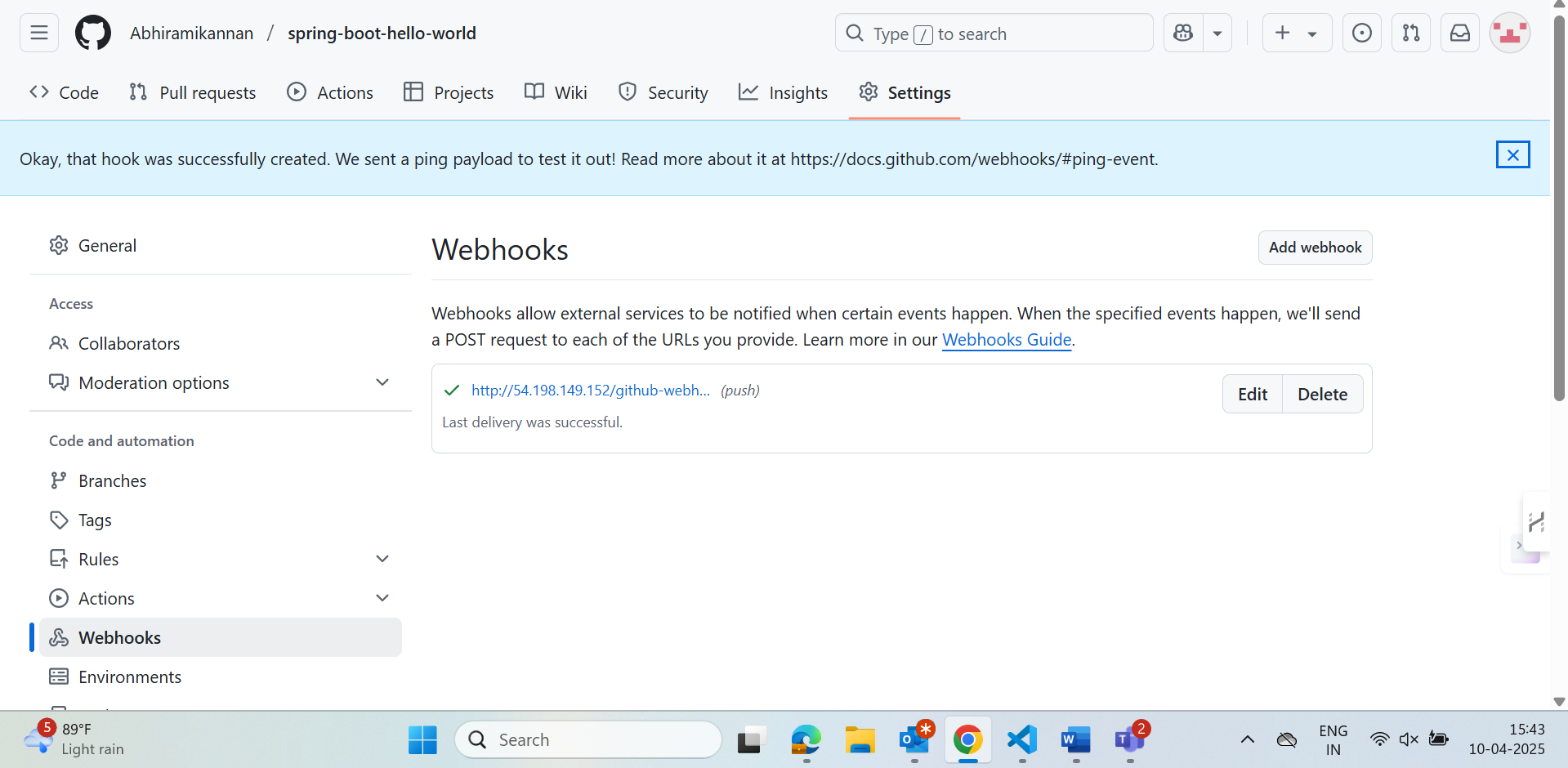
* **pushed into dockerhub**



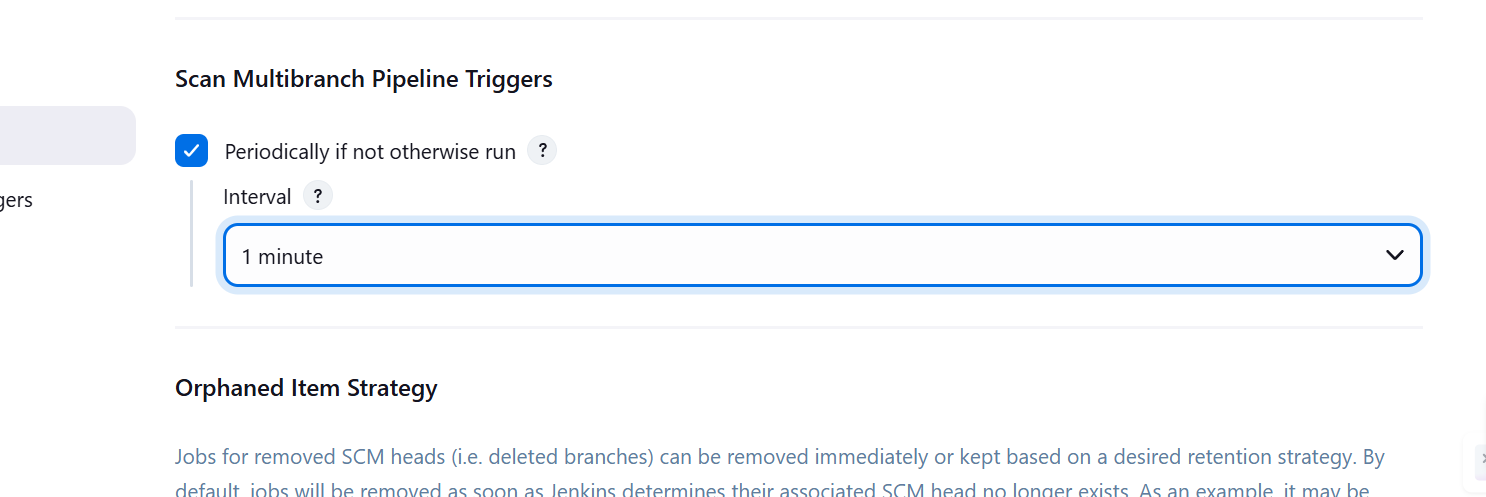




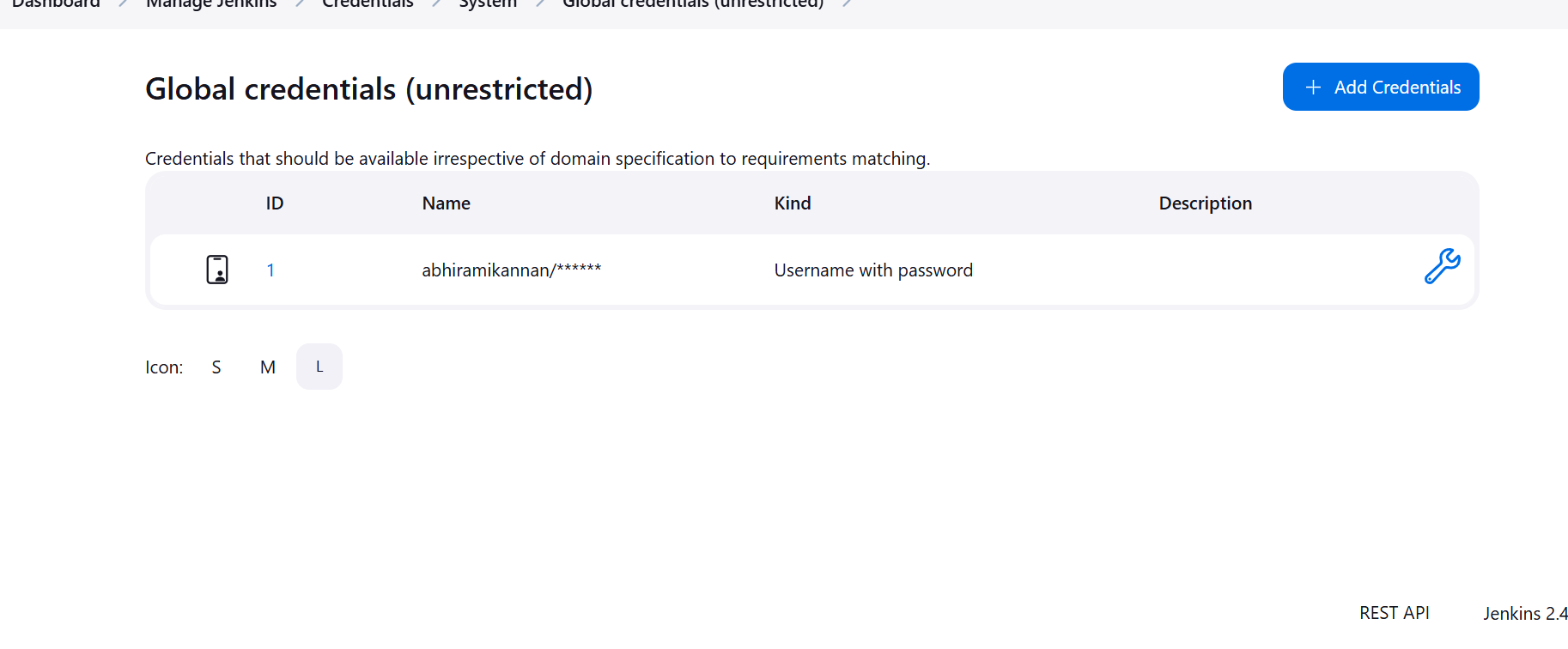
* **done webhook** in github by creating another instance because my instance terminated

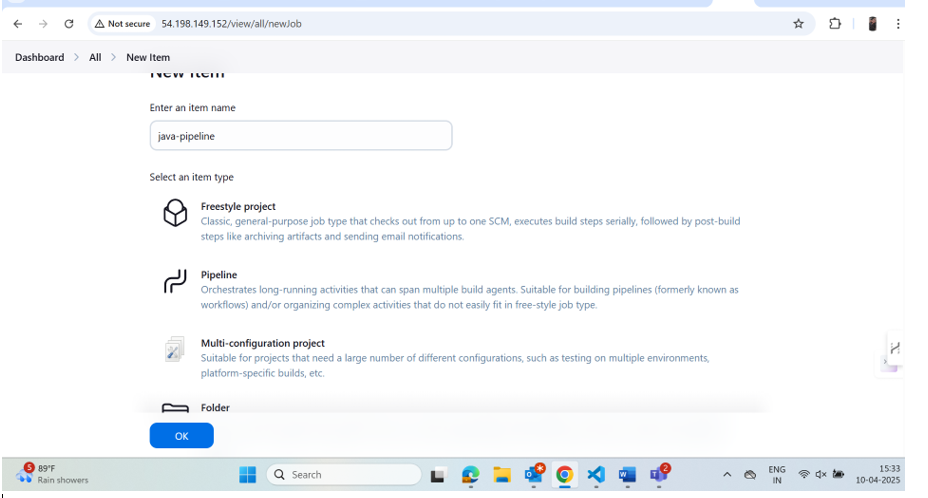
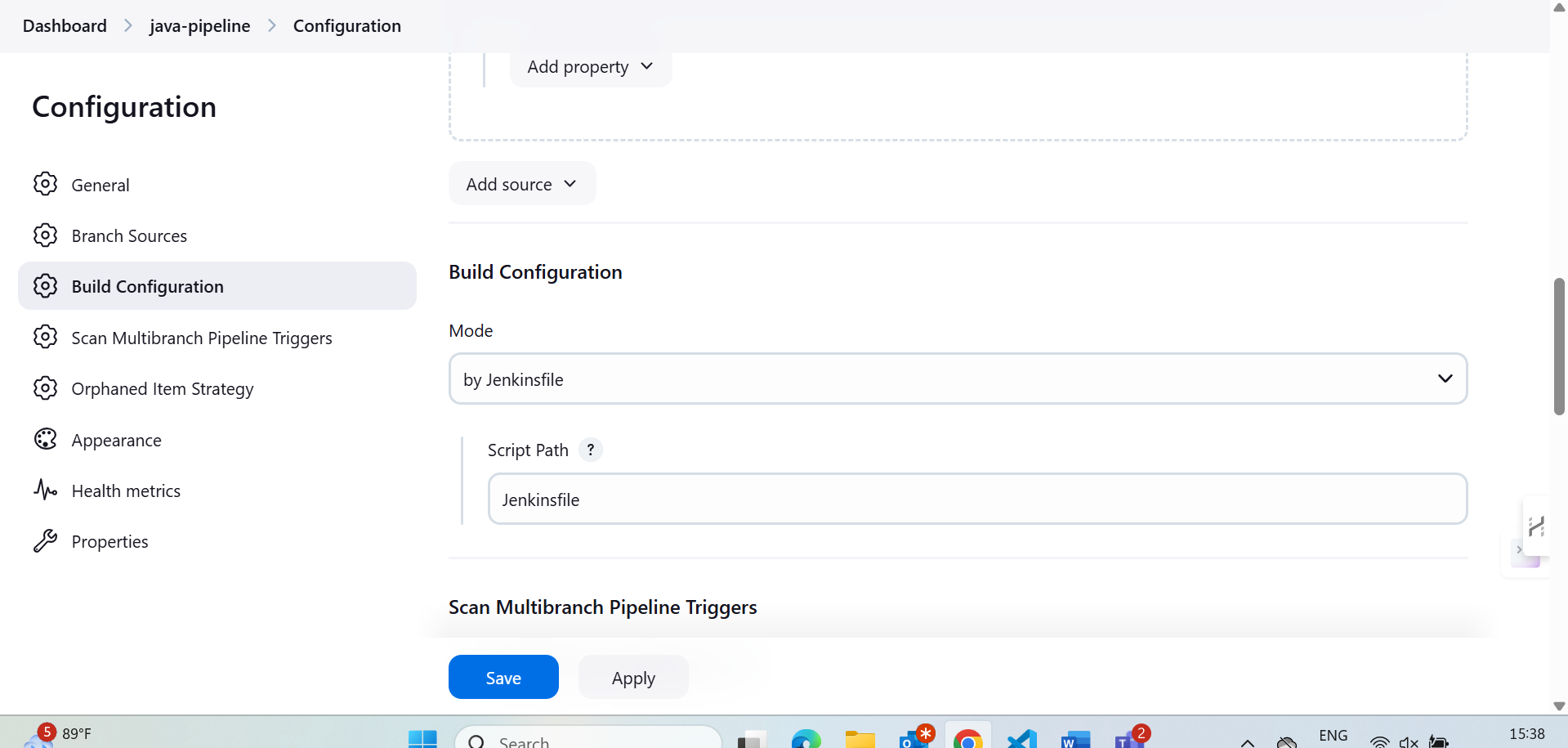


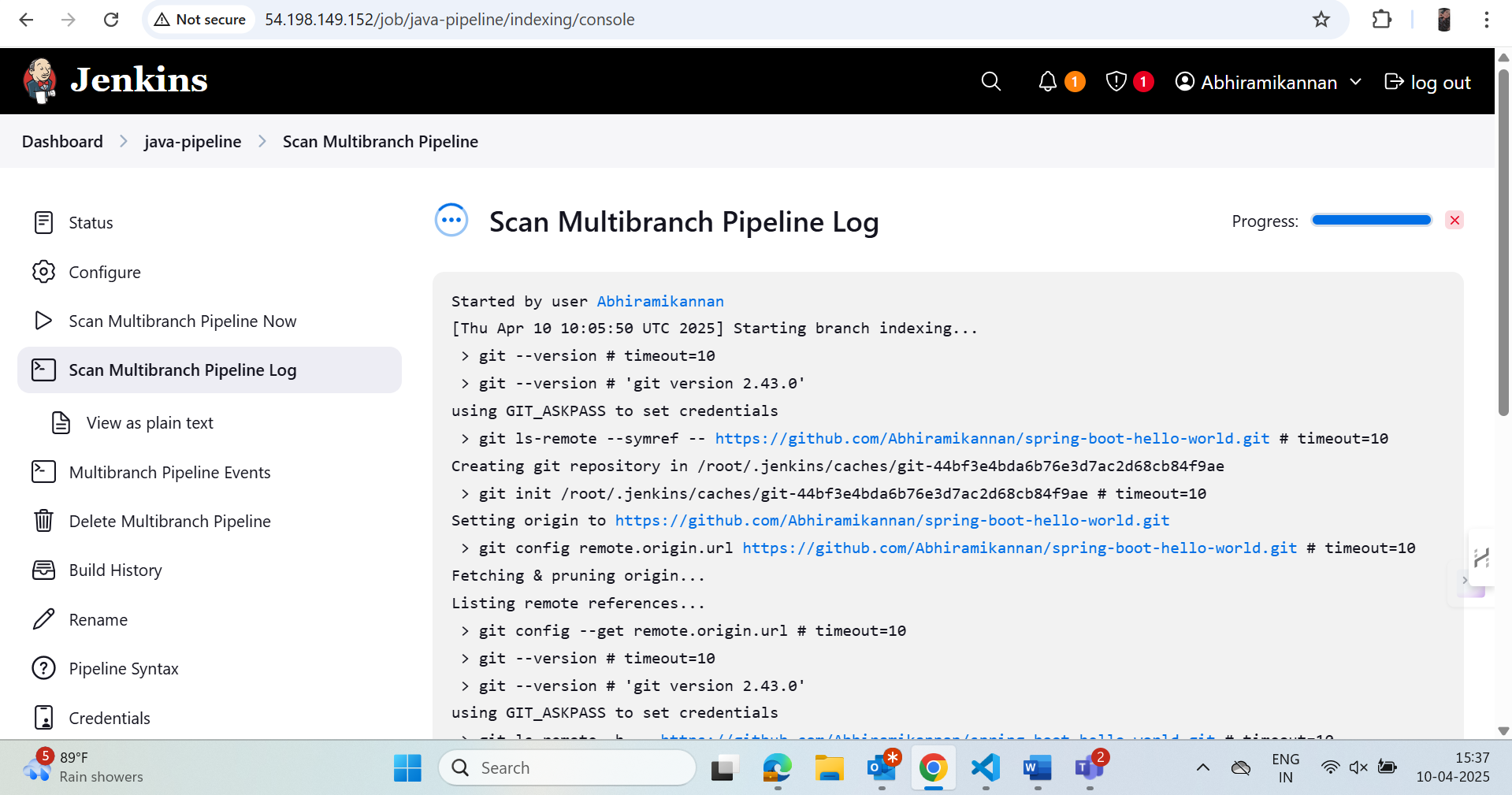
* **Jenkins multibranch-pipeline**

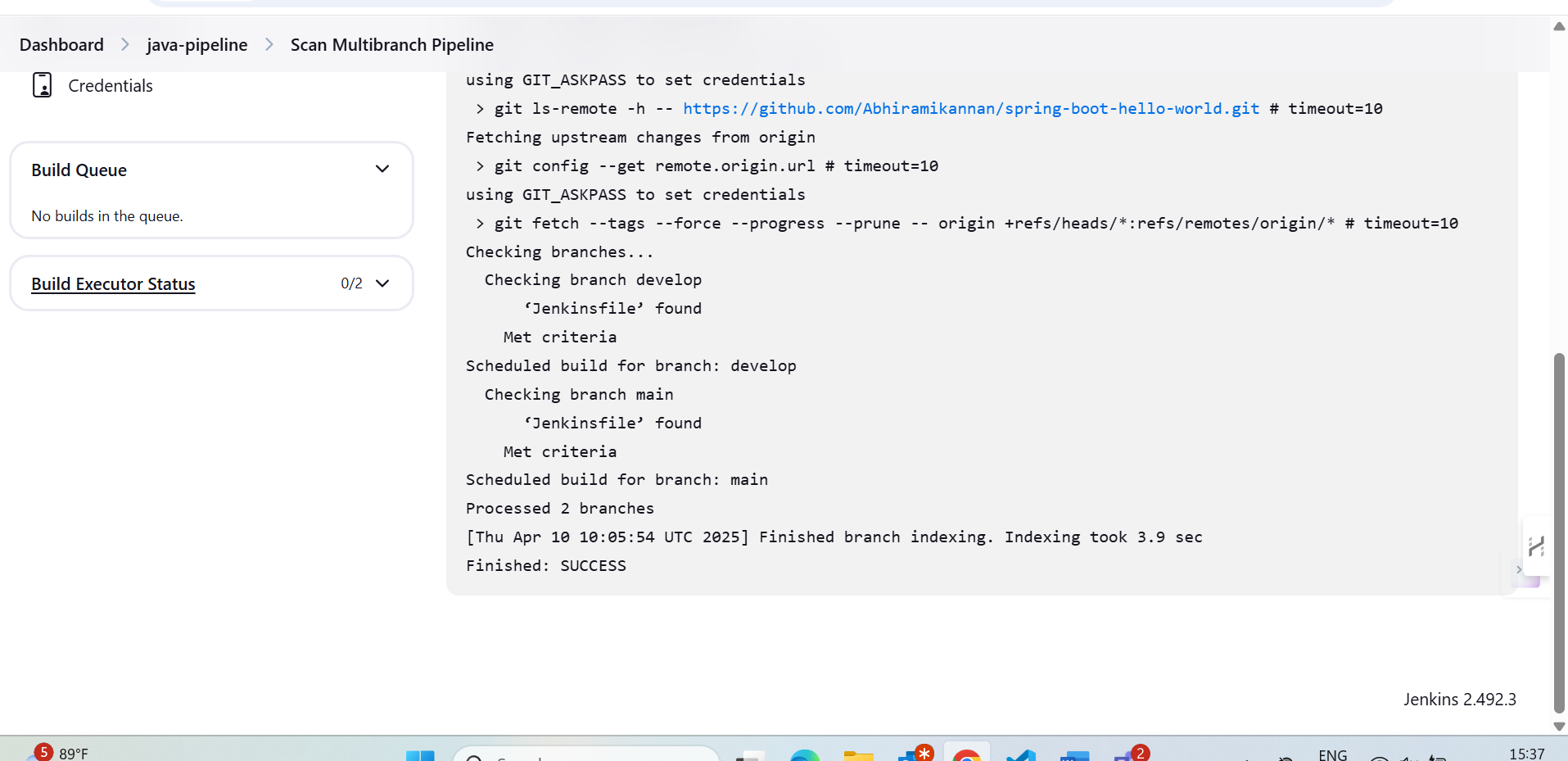


* **credentials added in Jenkins**

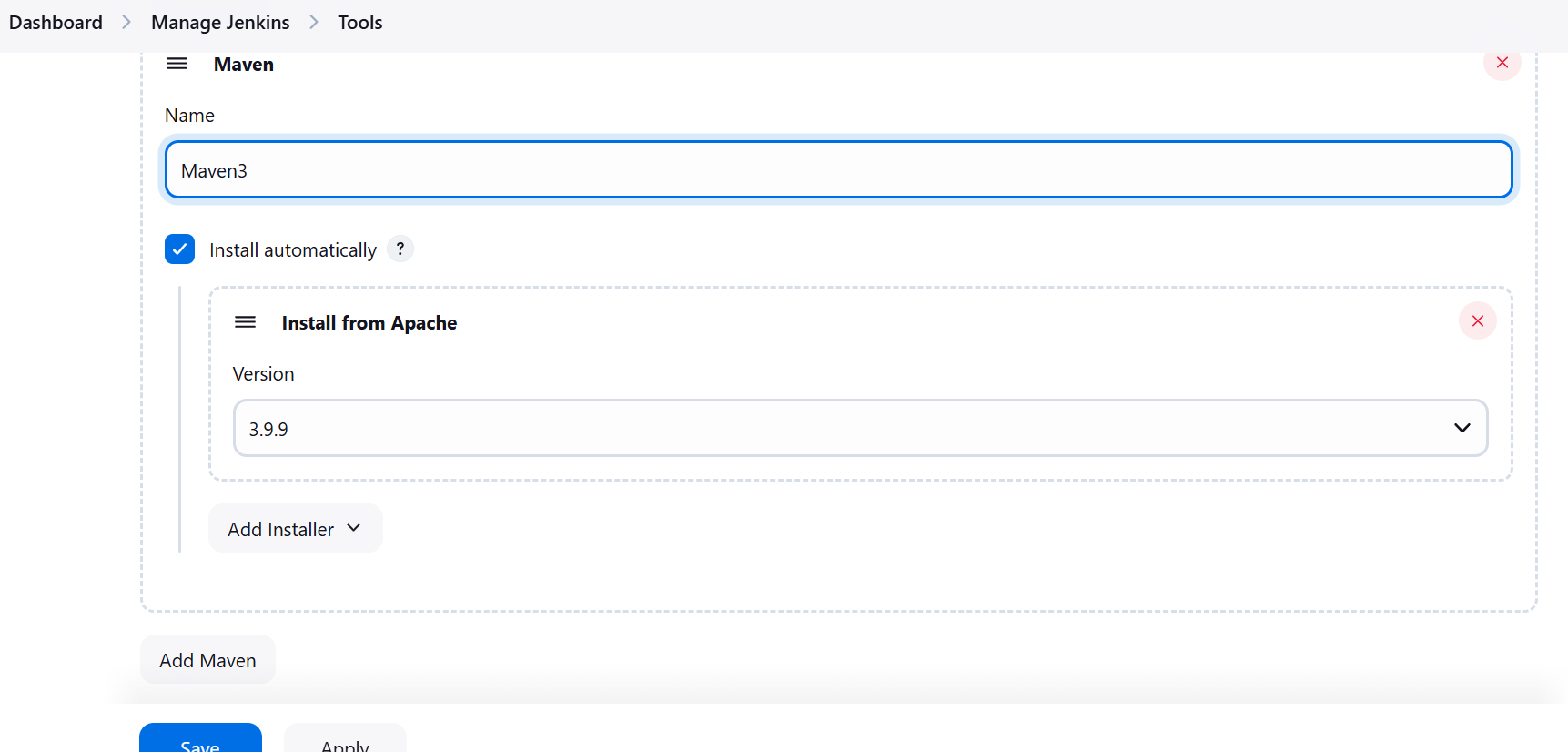


* **Created a multibranch pipeline**
* 
* 

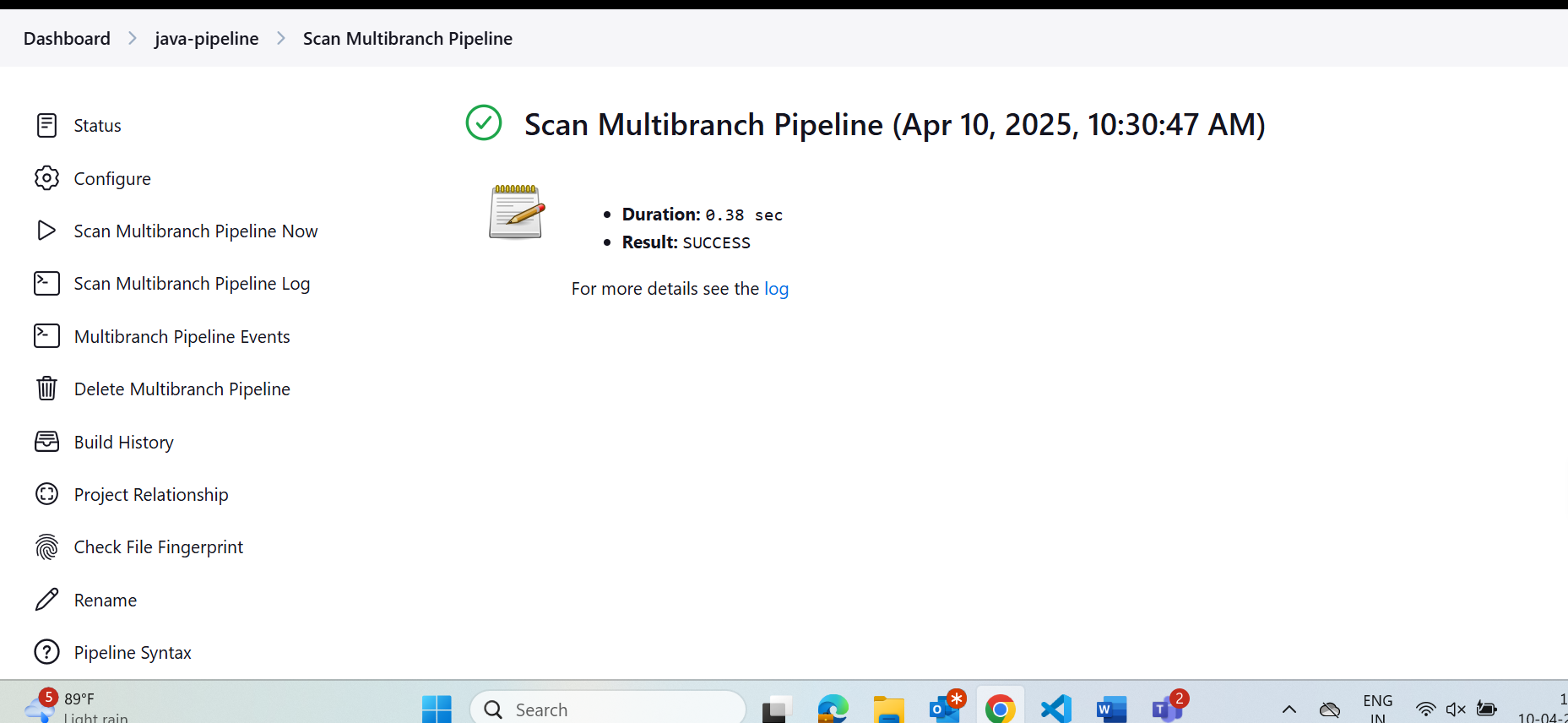




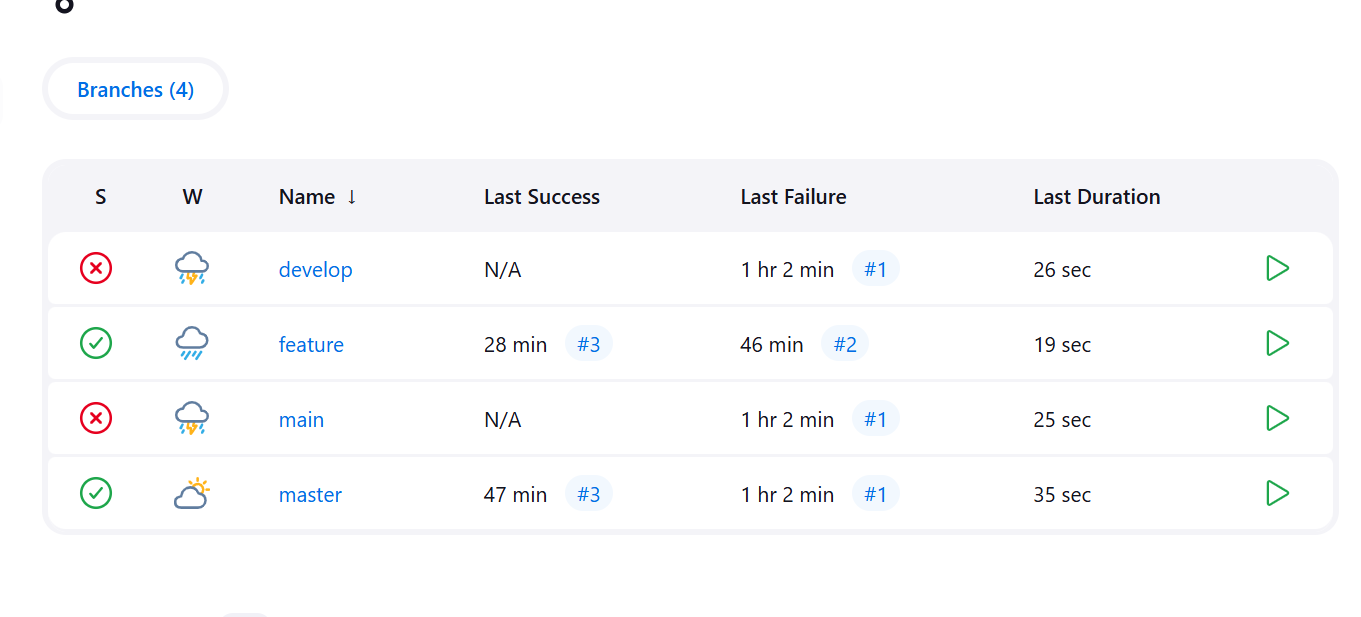
Installed maven



Result:



* **Jenkins part success**- tried by creating master branch and feature branch separately
  + - I created a new branches feature and master because of some configuration issues in Jenkins file.



* Master:[https://github.com/Abhiramikannan/spring-boot-hello-world/tree/master#](https://github.com/Abhiramikannan/spring-boot-hello-world/tree/master)
* Feature: <https://github.com/Abhiramikannan/spring-boot-hello-world/tree/feature>

**Jenkinsfile in masterbranch:**

pipeline {

agent any

environment {

IMAGE\_NAME = "abhiramikannan/javaimage"

TAG = "latest"

config\_docker\_credentials\_id = "1"

}

stages {

stage('Git Cloning') {

steps {

git branch: 'main', url: 'https://github.com/Abhiramikannan/spring-boot-hello-world.git'

}

}

stage('Build JAR File') {

steps {

sh '''

mvn dependency:go-offline

mvn clean package -DskipTests

'''

}

}

stage('Copy JAR File to Docker Context') {

steps {

sh '''

rm -f app.jar

cp /root/.jenkins/workspace/multi-branch\_master/target/\*.jar app.jar

'''

}

}

stage('Login & Push Docker Image') {

steps {

withCredentials([usernamePassword(

credentialsId: "${config\_docker\_credentials\_id}",

usernameVariable: 'DOCKER\_USER',

passwordVariable: 'DOCKER\_PASS'

)]) {

sh '''

echo $DOCKER\_PASS | docker login -u $DOCKER\_USER --password-stdin

docker build -t ${IMAGE\_NAME}:${TAG} .

docker push ${IMAGE\_NAME}:${TAG}

'''

}

}

}

stage('Deploy to Kubernetes') {

steps {

sh '''

kubectl delete deployment hello-world-deployment || true

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

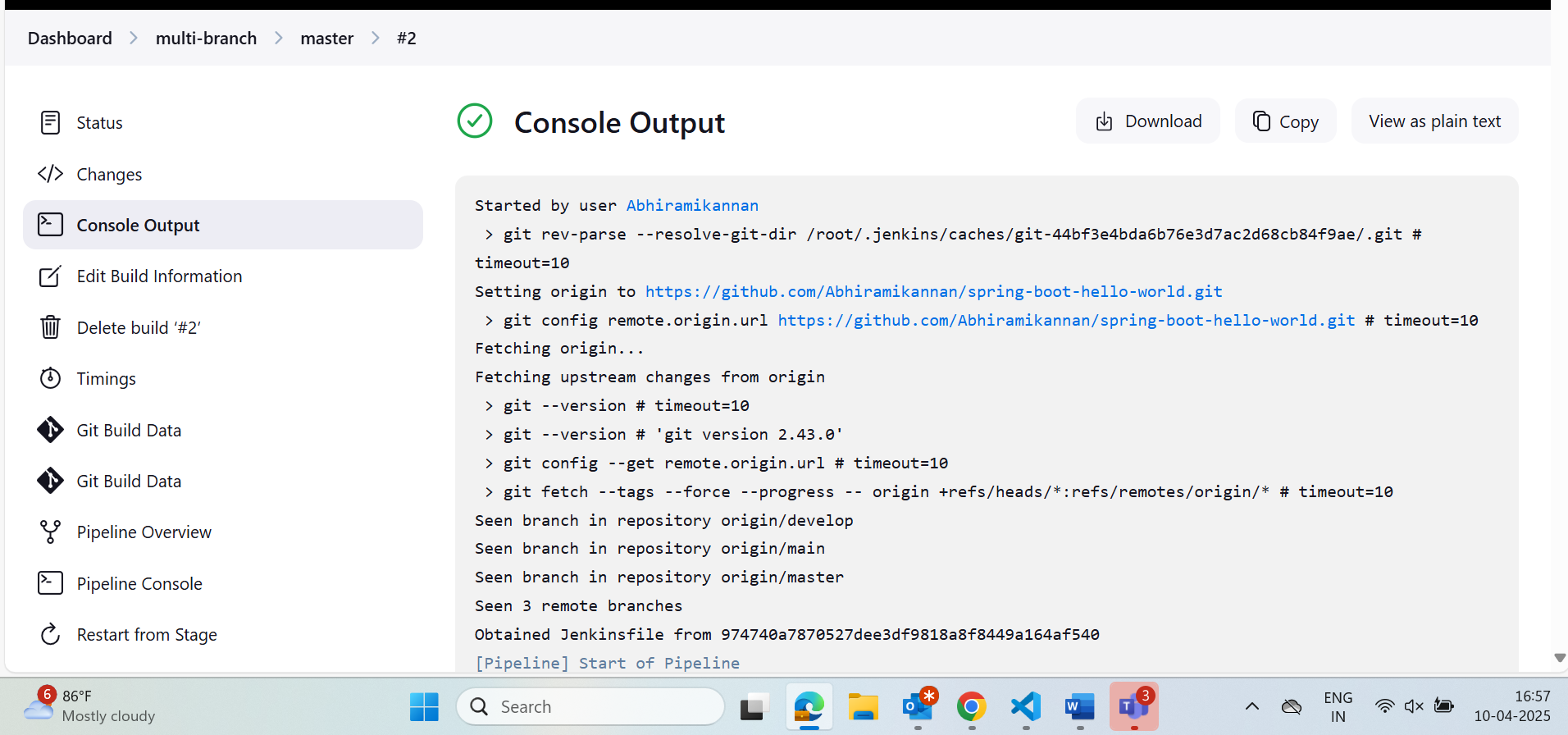
'''

}

}

}

}

* 
* **Jenkins in featurebranch**

**Jenkinsfile:**

pipeline {

agent any

stages {

stage('Git Cloning') {

steps {

// Clone code from the GitHub repository

git branch: 'feature', url: 'https://github.com/Abhiramikannan/spring-boot-hello-world.git'

}

}

stage('Build JAR File') {

steps {

// Download dependencies

sh 'mvn dependency:go-offline'

// Build the project, skipping tests

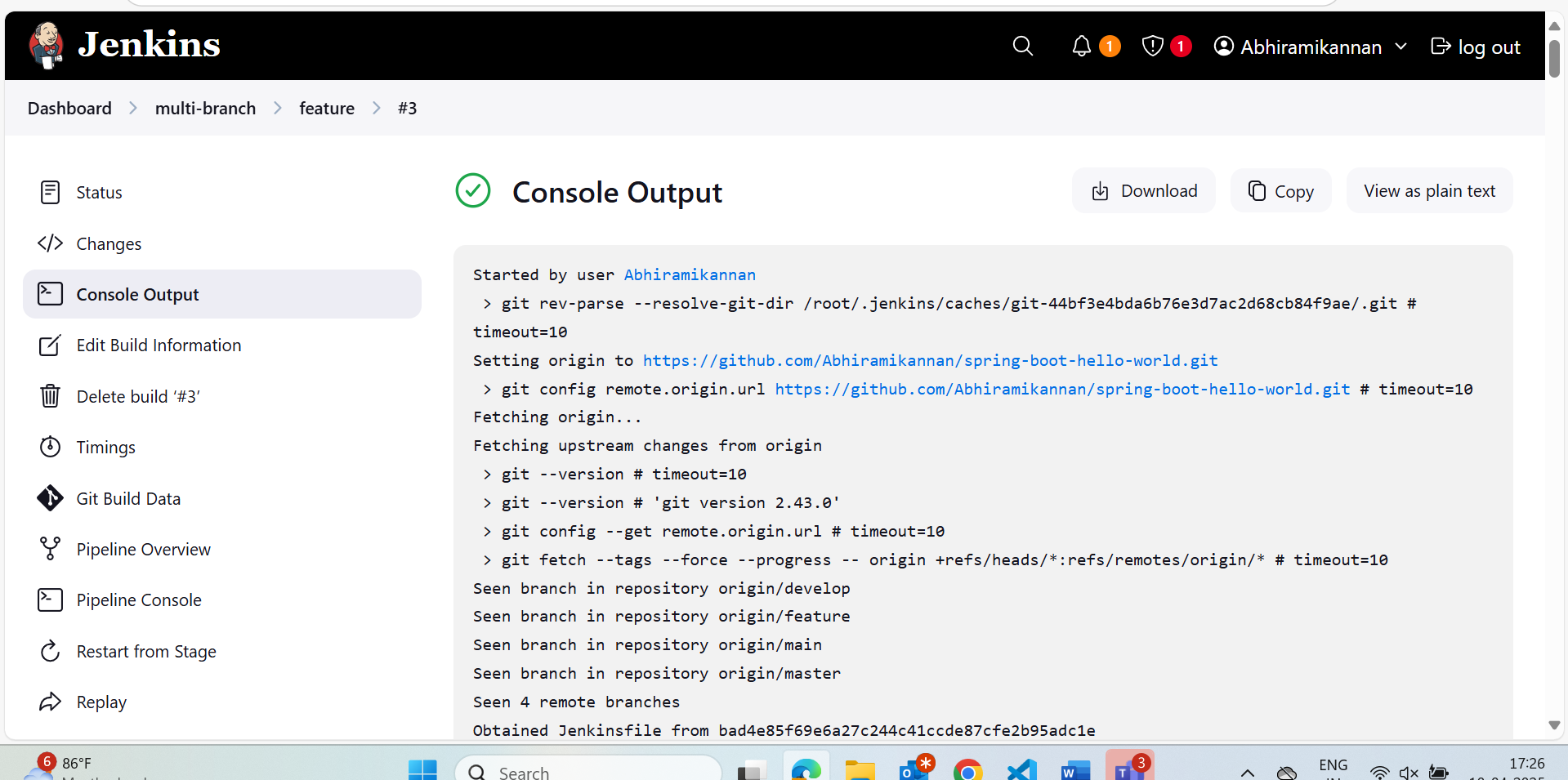
sh 'mvn clean package -DskipTests'

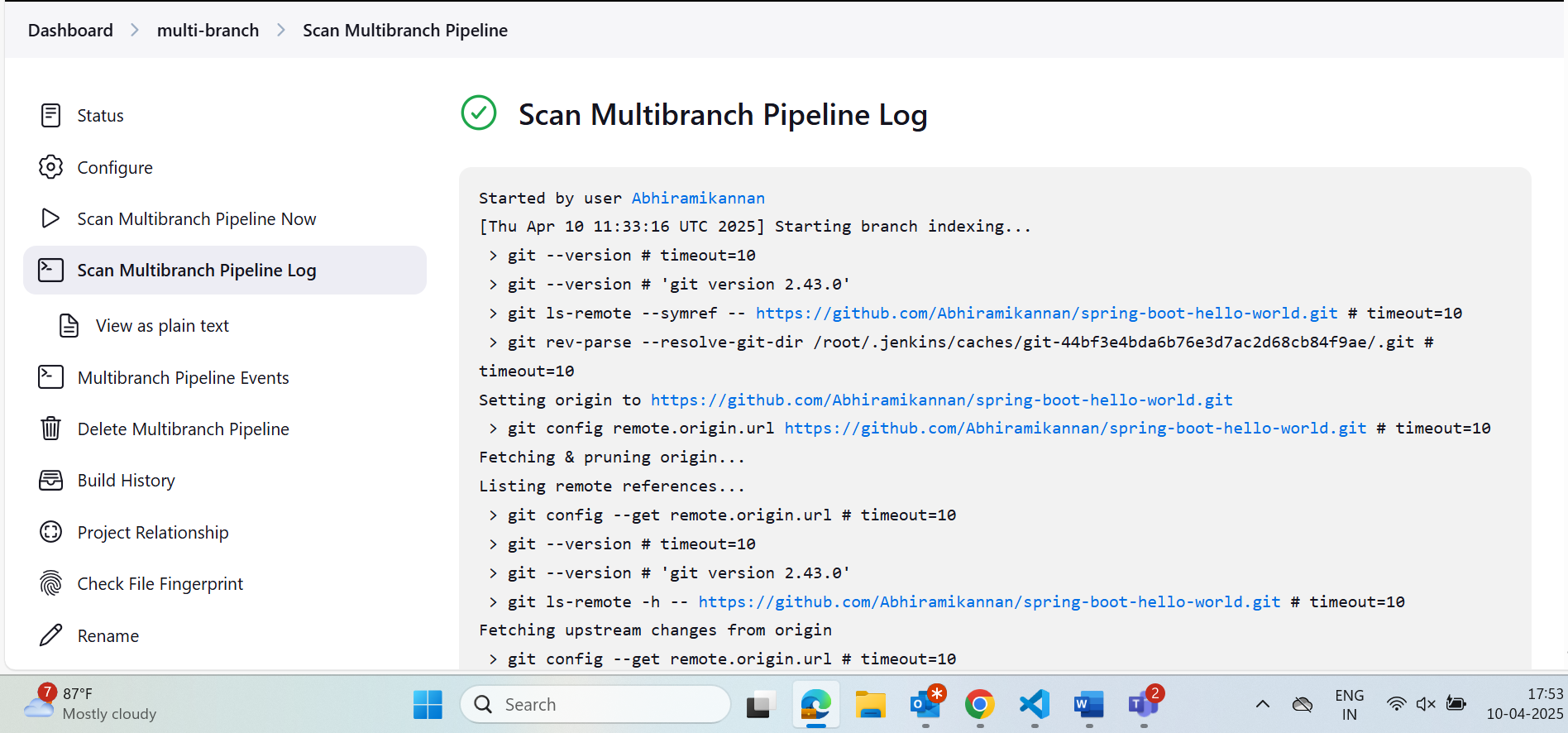
}

}

}

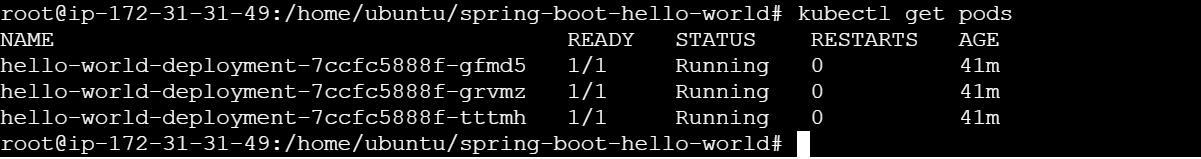
}

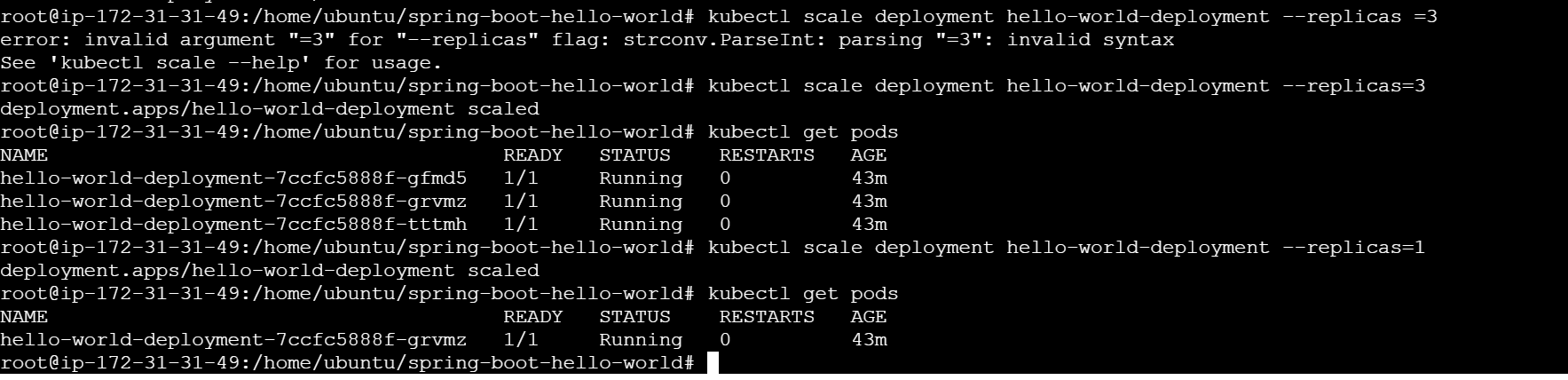


* **Multipipeline logs**
* 
* **Kubernetes yaml files**

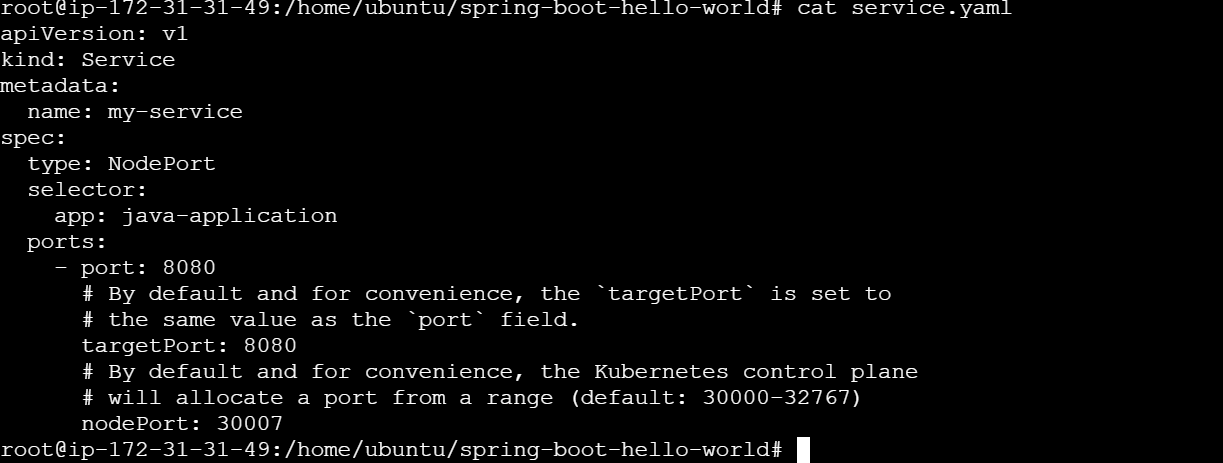
1. Deployment.yaml



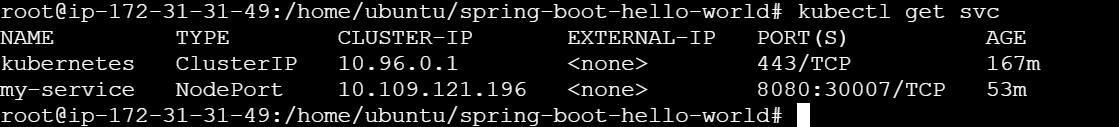




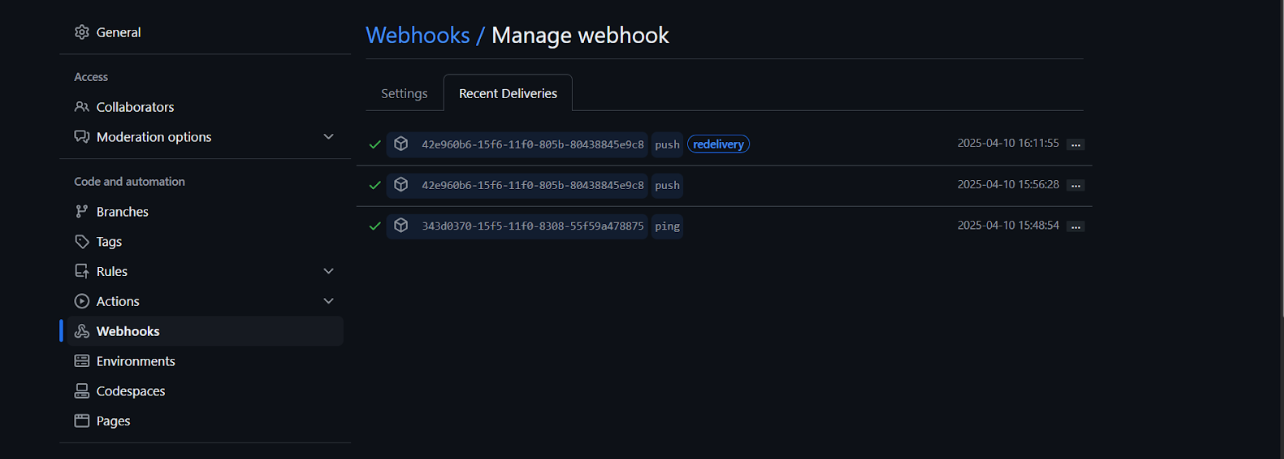
1. **Service.yaml** using nodeport



Node port access of app, exposed this using service



**Webooks:**



* **Multipipeline configurations:**

