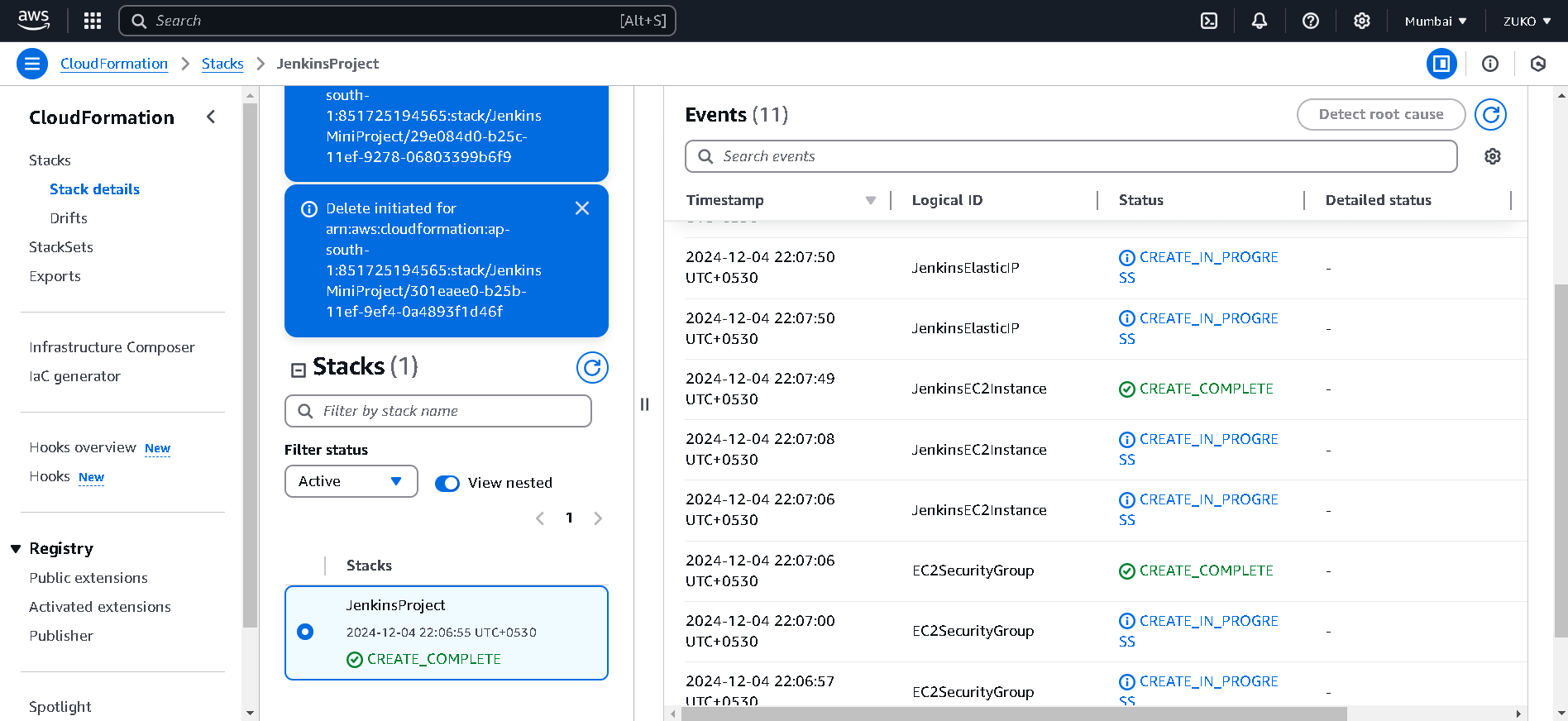
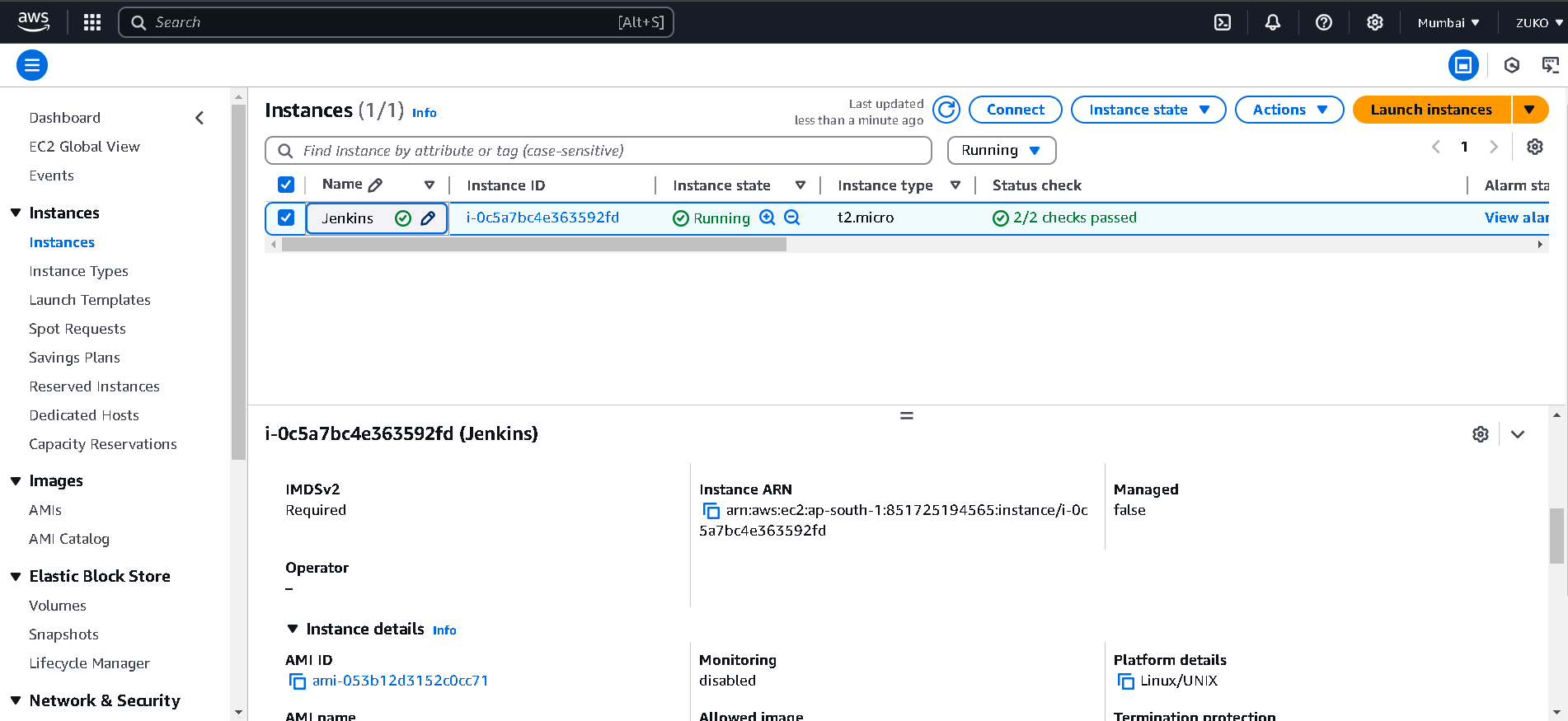
TASK12- AWS CloudFormation-based Jenkins CI/CD Pipeline Setup and Git Integration

"Using AWS CloudFormation, I created an Ubuntu EC2 instance, installed Jenkins along with the necessary prerequisites, and attached an Elastic IP to it. A webhook was created to extract code from GitHub to the Jenkins pipeline. The pipeline script, written in Groovy, was saved in a Jenkinsfile and pushed to GitHub. When a change is committed in the repository, the webhook triggers the build, and an email notification is sent with the build output."

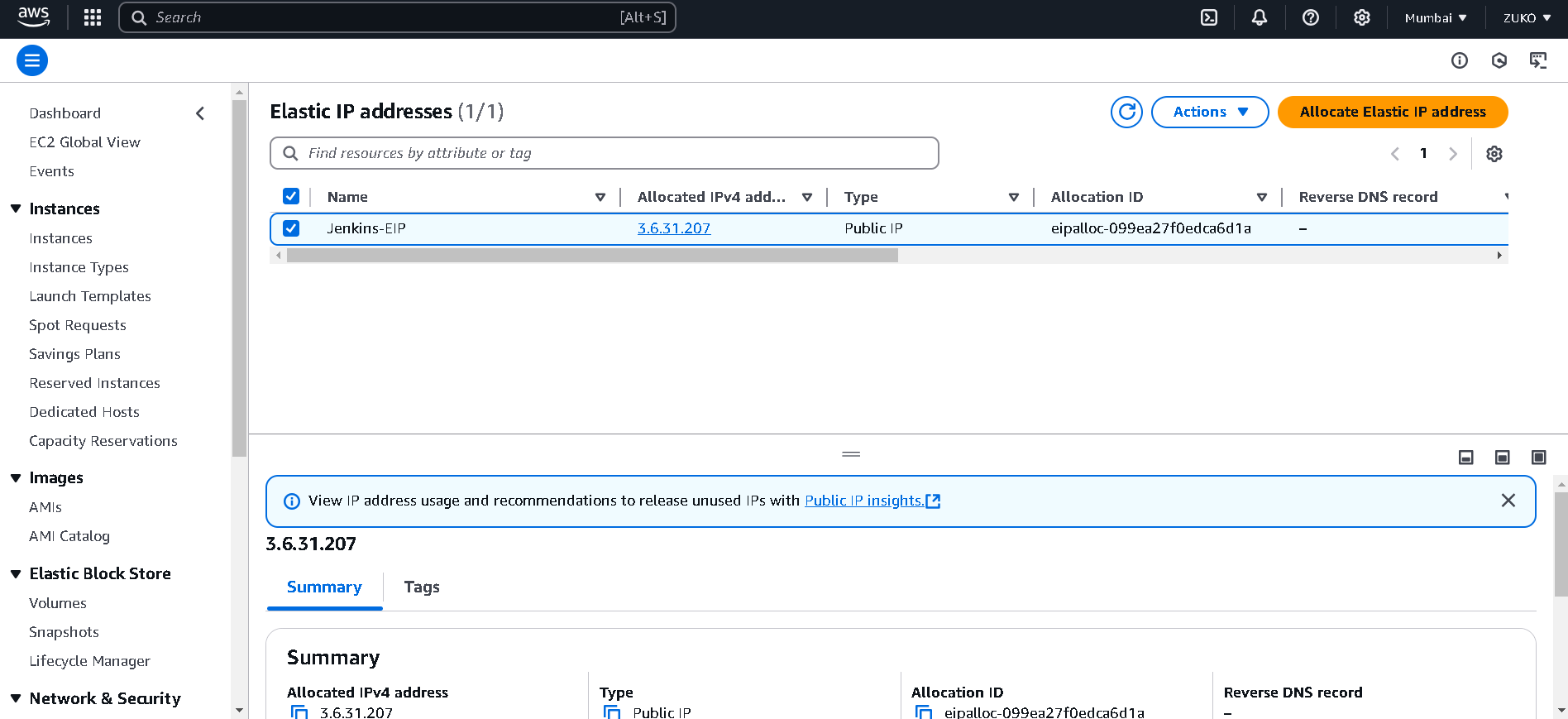
**CREATED STACKS**



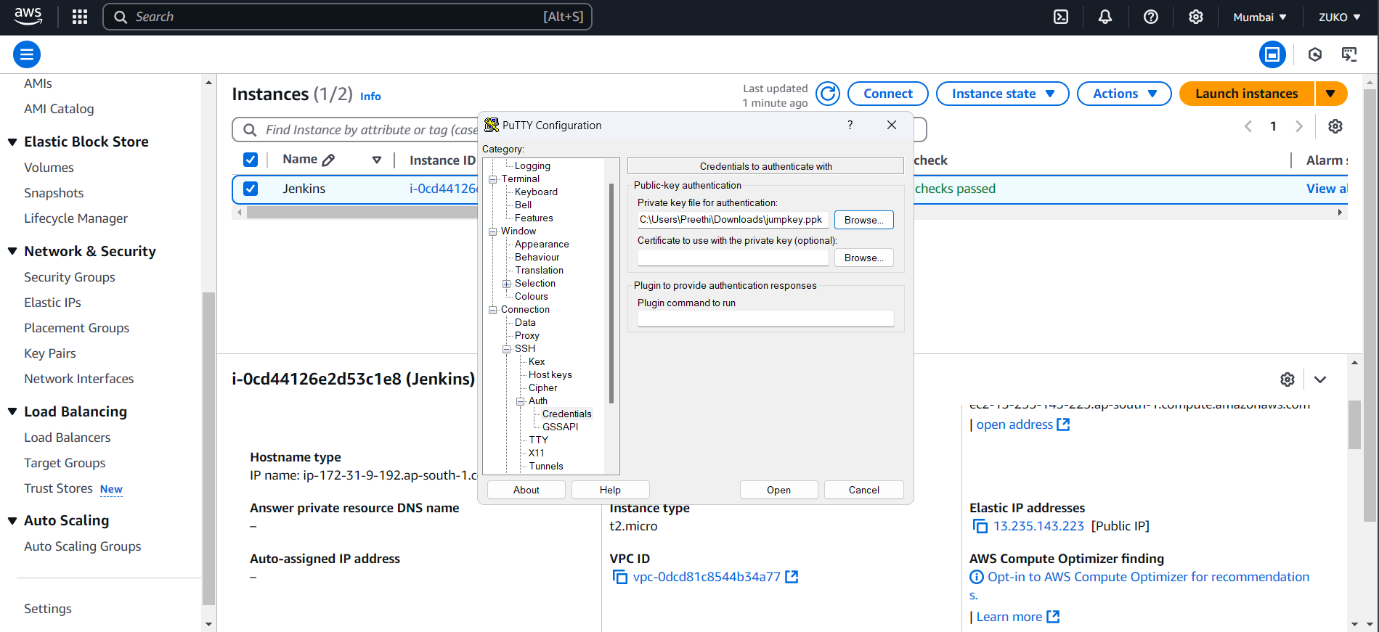
**IAAC SUCCESFULL INSTANCE CREATED**



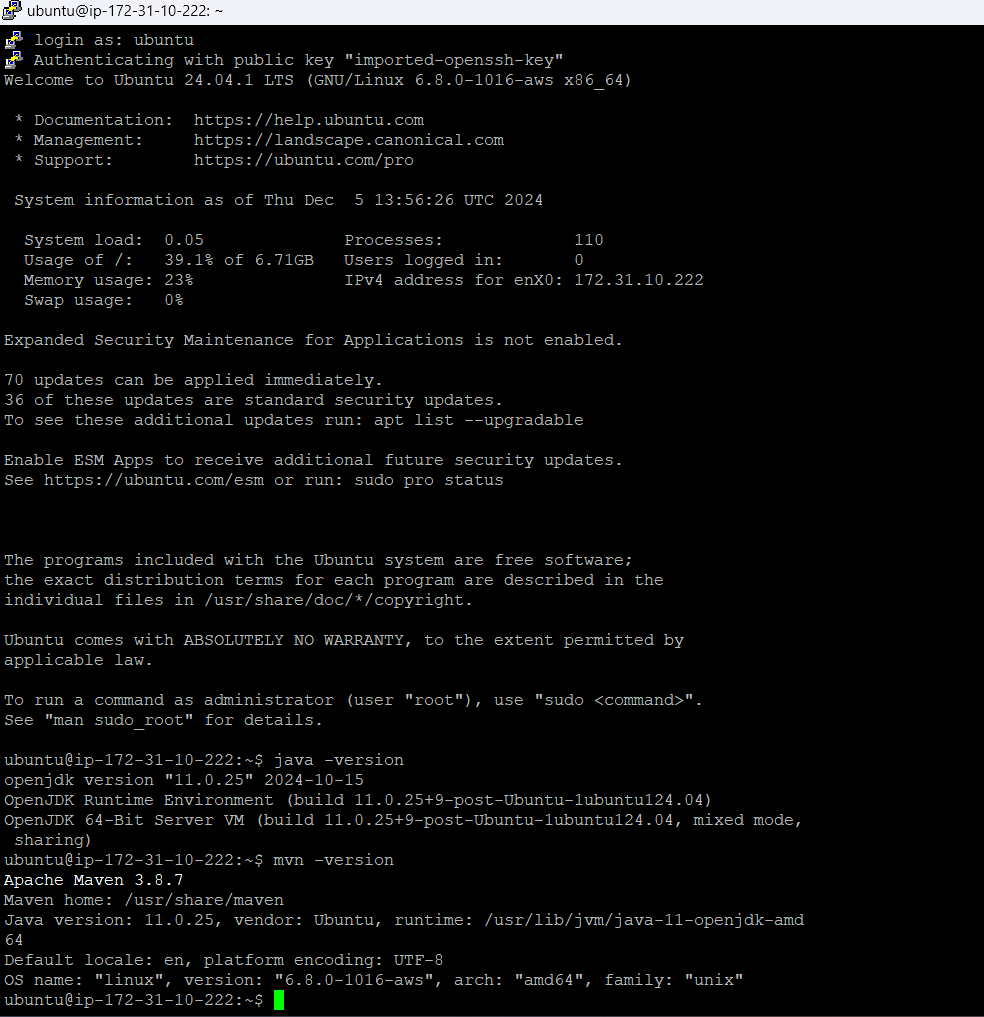
**CREATED EIP via IAAC**



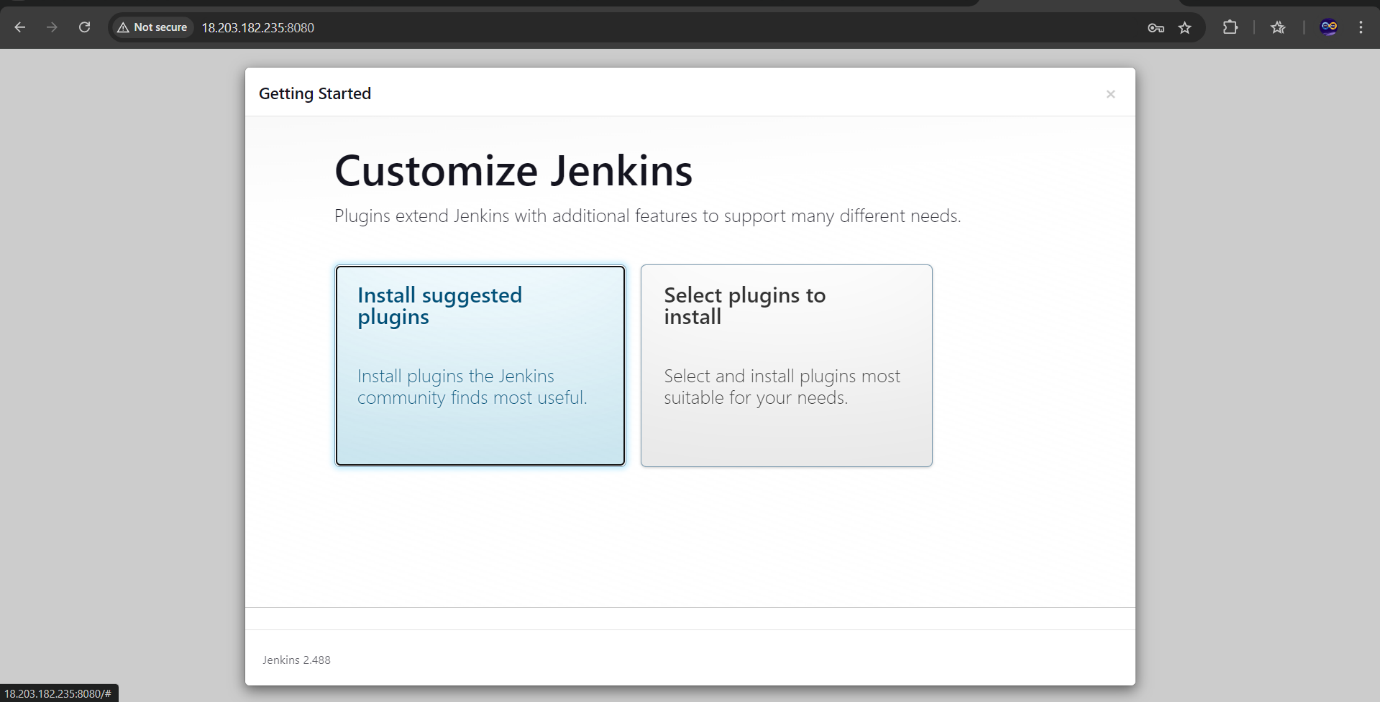
**USING PUTTY CONNECT TO SERVER**



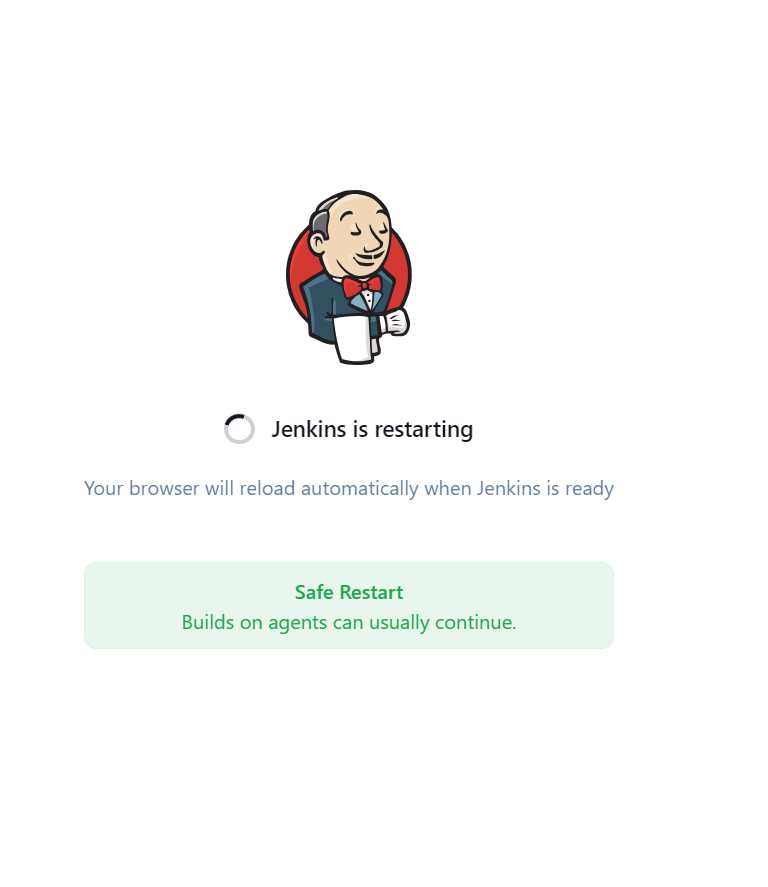
**CHECKING JAVA, MAVEN, JENKINS**.



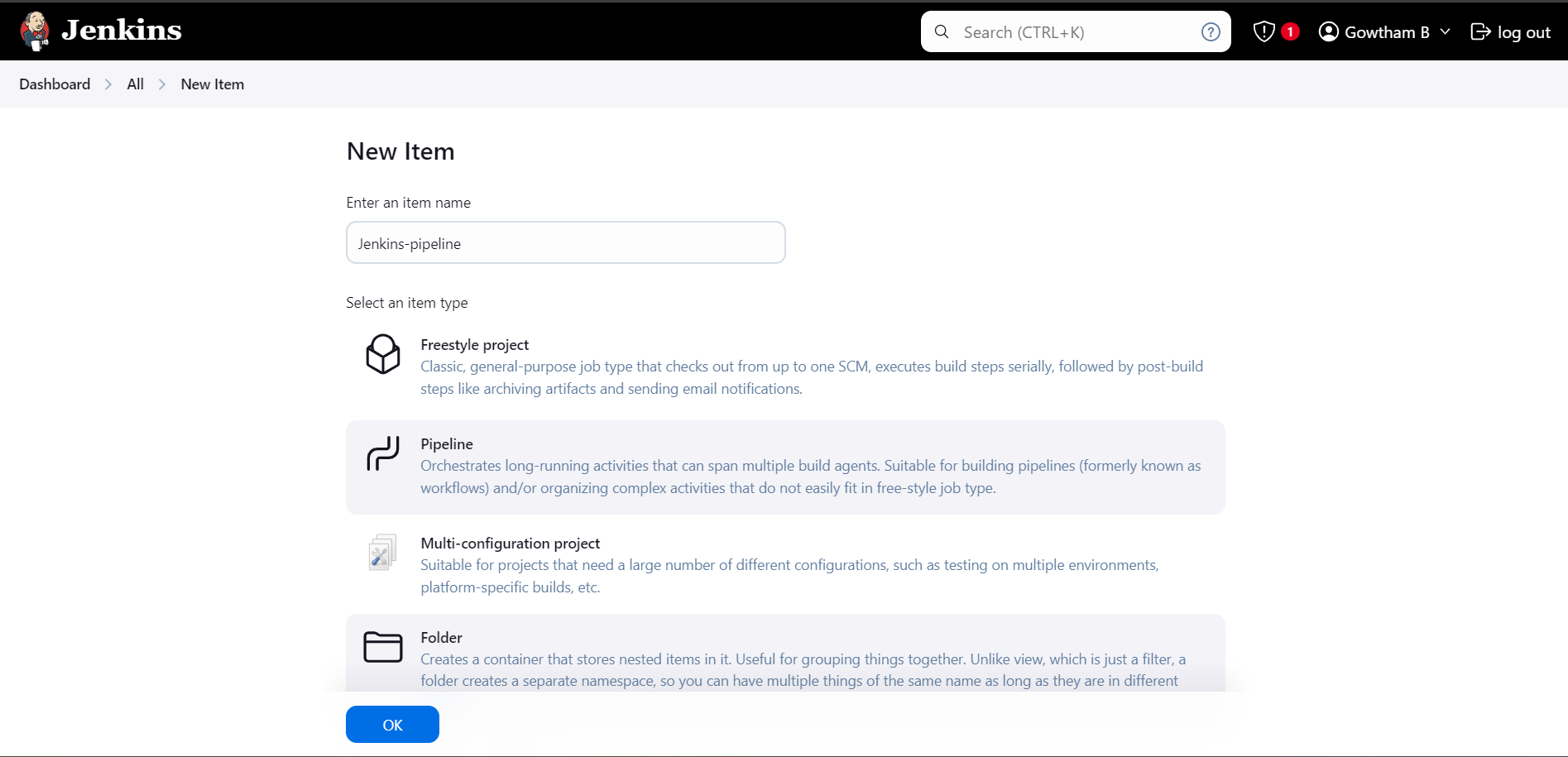
**INSTALLING PLUGINS**



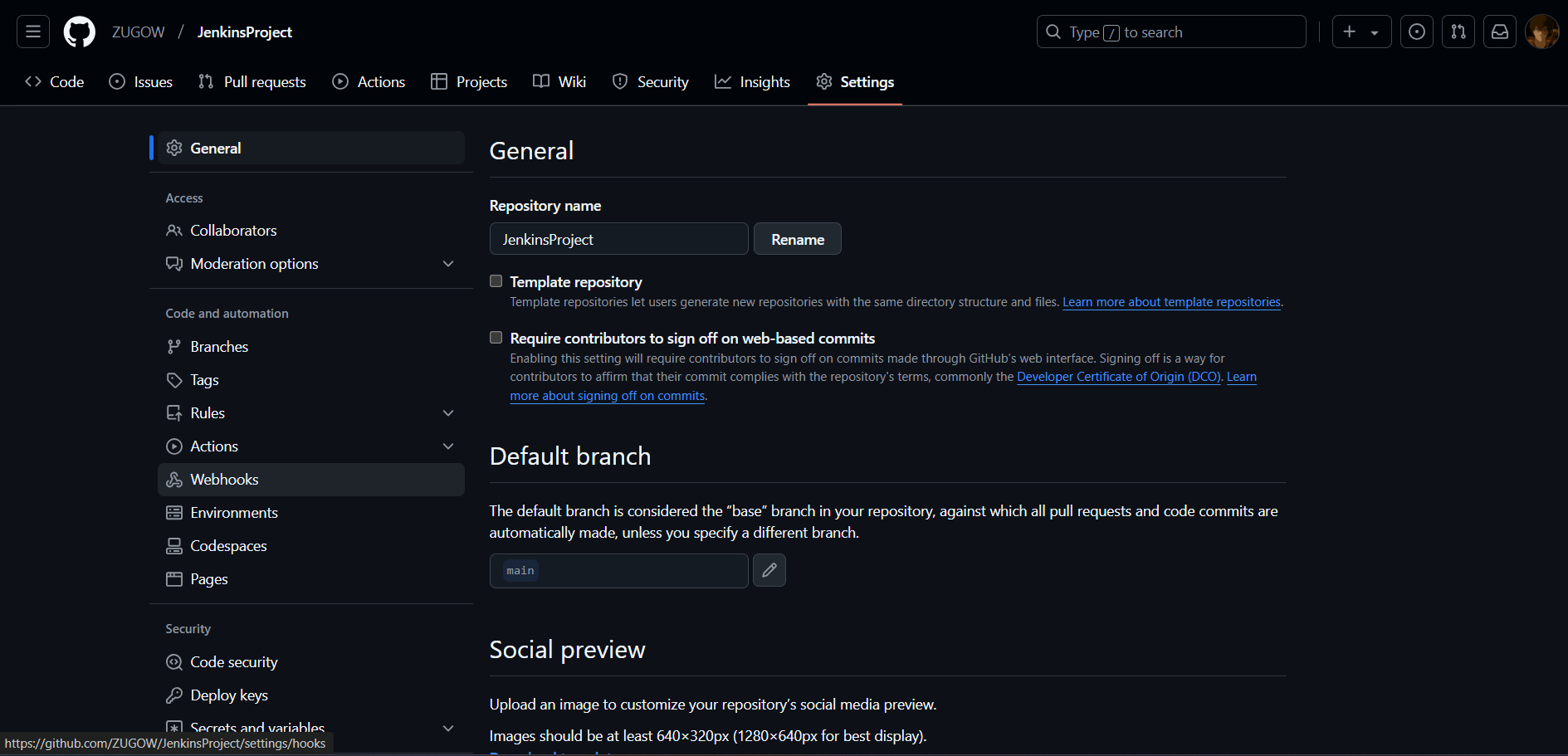
**JENKINS IS RESTARTING**



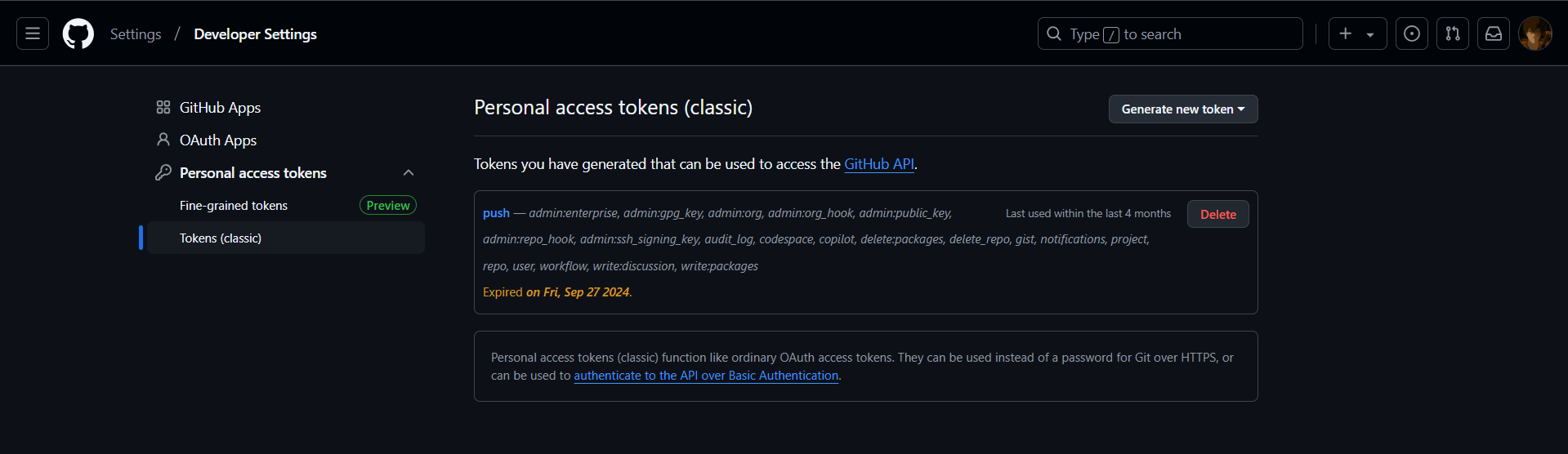
**CREATE A PIPELINE**



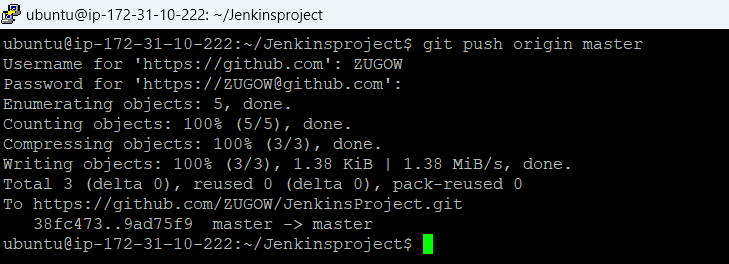
**CREATE A REPO AND CREATE WEBHOOK**

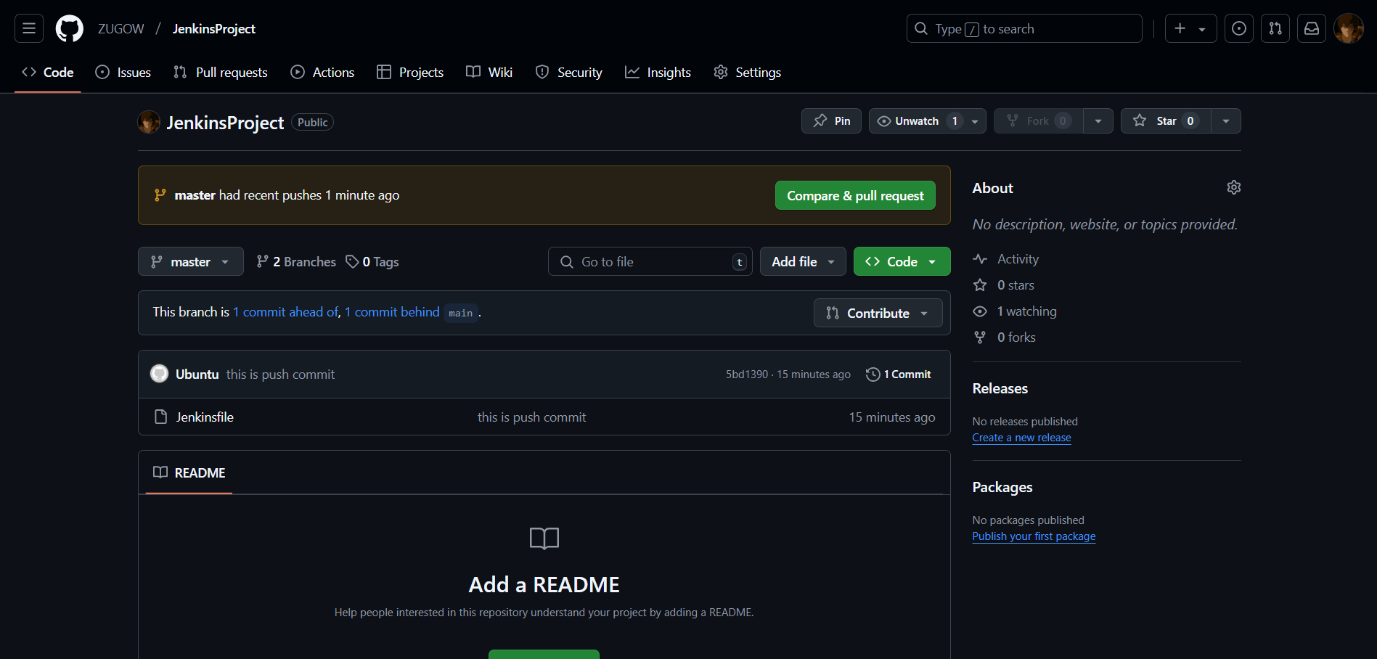


**CREATING PERSONAL ACCESS TOKENS FOR AUTHENTICATION**

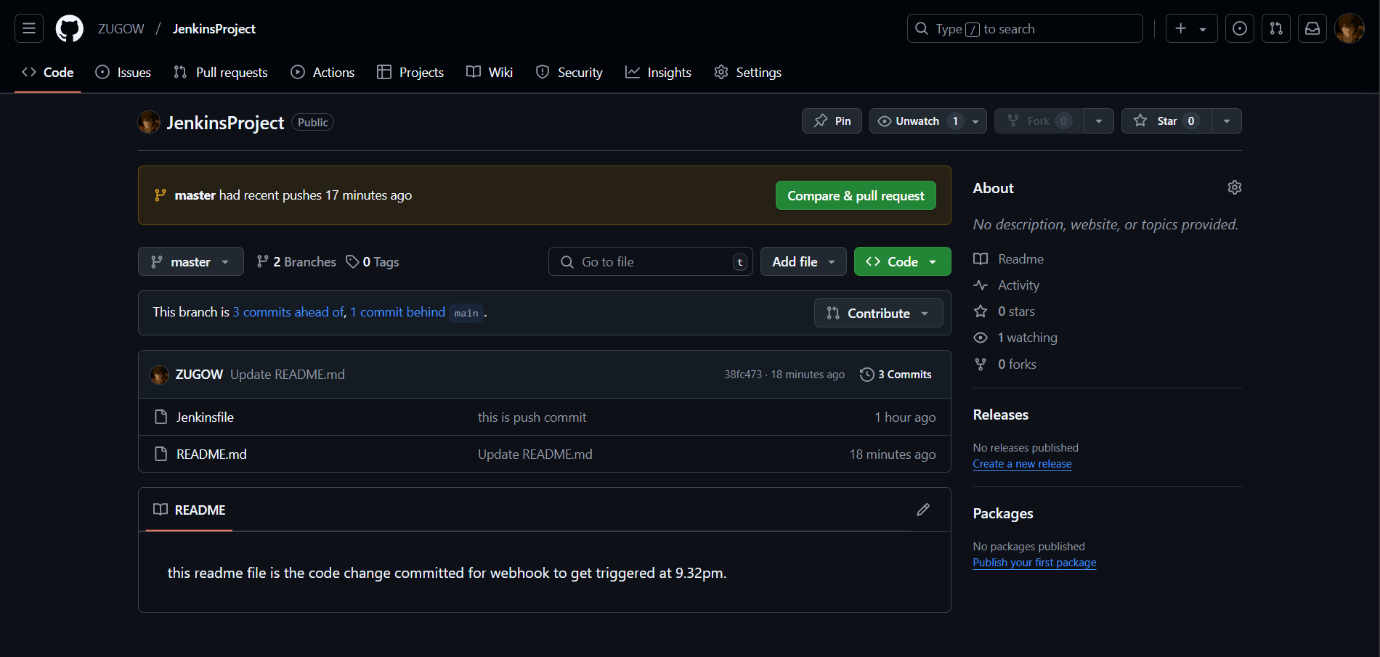


**PUSHED Jenkinsfile FROM LOCAL TO REMOTE (GITHUB)**

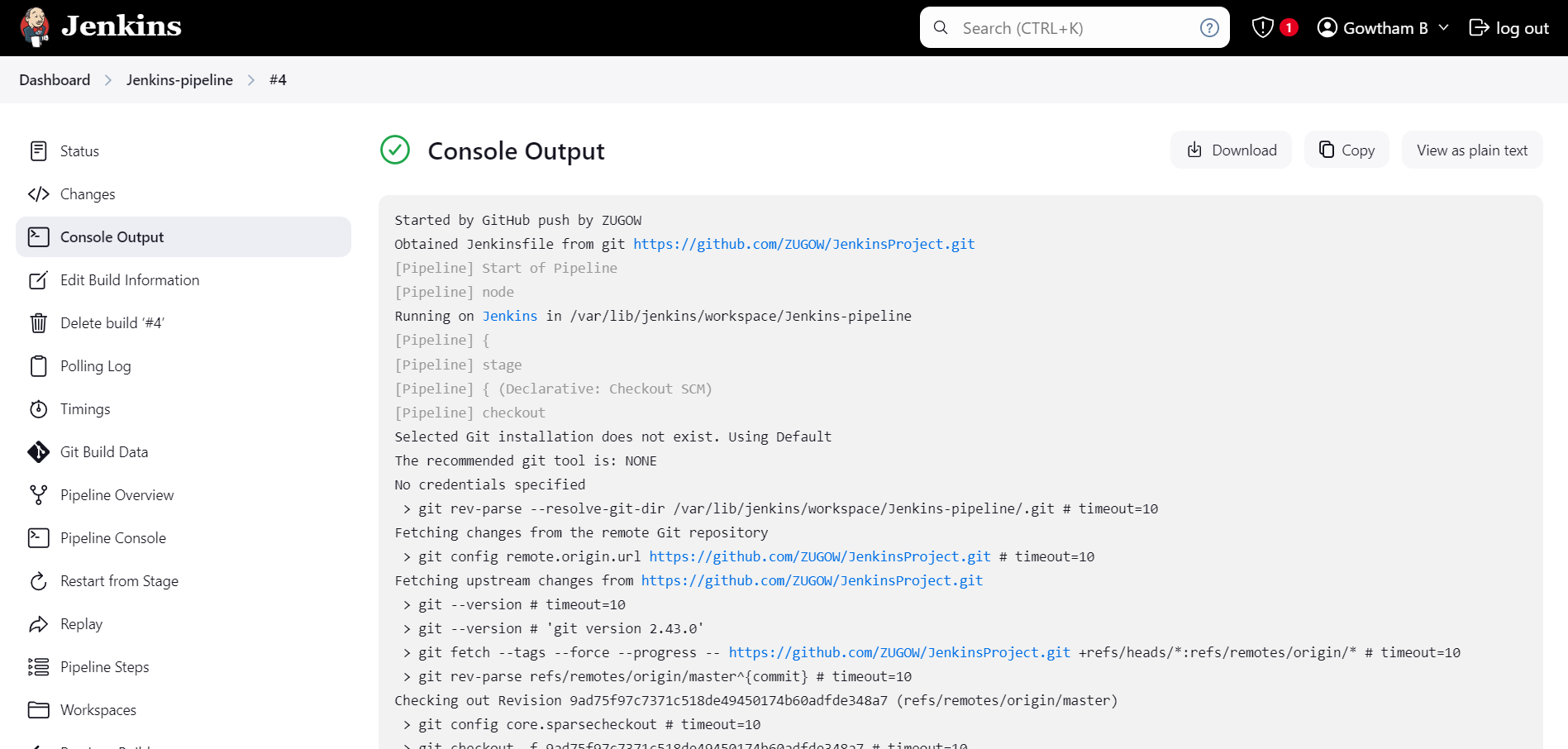
****



**CODE CHANGED AND COMMITTED**



**BUILD SUCCESS**

****

**PIPELINE SCRIPT**

Started by GitHub push by ZUGOW

Obtained Jenkinsfile from git <https://github.com/ZUGOW/JenkinsProject.git>

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](http://18.203.182.235:8080/computer/(built-in)/) in /var/lib/jenkins/workspace/Jenkins-pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

No credentials specified

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Jenkins-pipeline/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/ZUGOW/JenkinsProject.git> # timeout=10

Fetching upstream changes from <https://github.com/ZUGOW/JenkinsProject.git>

> git --version # timeout=10

> git --version # 'git version 2.43.0'

> git fetch --tags --force --progress -- <https://github.com/ZUGOW/JenkinsProject.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/master^{commit} # timeout=10

Checking out Revision 9ad75f97c7371c518de49450174b60adfde348a7 (refs/remotes/origin/master)

> git config core.sparsecheckout # timeout=10

> git checkout -f 9ad75f97c7371c518de49450174b60adfde348a7 # timeout=10

Commit message: "Added 10-stage Jenkins declarative pipeline"

> git rev-list --no-walk 38fc4738b39bbbcb826f42e1b5e141b197967bfb # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Stage 1)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 1: Jenkins is awesome!

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 2)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 2: Jenkins automates builds.

[Pipeline] echo

Stage 2: Jenkins makes CI/CD easy.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 3)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 3: Jenkins integrates with GitHub.

[Pipeline] echo

Stage 3: Jenkins allows plugin installations.

[Pipeline] echo

Stage 3: Jenkins can handle complex pipelines.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 4)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 4: Jenkins pipelines are reusable.

[Pipeline] echo

Stage 4: Jenkins supports multiple languages.

[Pipeline] echo

Stage 4: Jenkins can be extended via APIs.

[Pipeline] echo

Stage 4: Jenkins is open-source.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 5)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 5: Jenkins has a large user community.

[Pipeline] echo

Stage 5: Jenkins supports Docker integration.

[Pipeline] echo

Stage 5: Jenkins can run tests in parallel.

[Pipeline] echo

Stage 5: Jenkins integrates with AWS.

[Pipeline] echo

Stage 5: Jenkins is great for DevOps.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 6)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 6: Jenkins can use AWS CloudFormation for setup.

[Pipeline] echo

Stage 6: Jenkins supports various SCM tools.

[Pipeline] echo

Stage 6: Jenkins can trigger builds via webhooks.

[Pipeline] echo

Stage 6: Jenkins enables distributed builds.

[Pipeline] echo

Stage 6: Jenkins supports continuous integration.

[Pipeline] echo

Stage 6: Jenkins has a wide range of plugins.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 7)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 7: Jenkins can handle complex workflows.

[Pipeline] echo

Stage 7: Jenkins offers easy-to-use GUI for job configuration.

[Pipeline] echo

Stage 7: Jenkins allows automated deployments.

[Pipeline] echo

Stage 7: Jenkins supports multi-branch pipelines.

[Pipeline] echo

Stage 7: Jenkins can automate testing workflows.

[Pipeline] echo

Stage 7: Jenkins supports integration with Slack.

[Pipeline] echo

Stage 7: Jenkins helps to monitor build results.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 8)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 8: Jenkins integrates with Docker.

[Pipeline] echo

Stage 8: Jenkins can use Kubernetes for scaling.

[Pipeline] echo

Stage 8: Jenkins enables easy integration with version control systems.

[Pipeline] echo

Stage 8: Jenkins provides build notifications.

[Pipeline] echo

Stage 8: Jenkins can be automated using scripts.

[Pipeline] echo

Stage 8: Jenkins enables versioned artifacts.

[Pipeline] echo

Stage 8: Jenkins provides real-time build logs.

[Pipeline] echo

Stage 8: Jenkins supports both freestyle and declarative pipelines.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 9)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 9: Jenkins can trigger deployment after build completion.

[Pipeline] echo

Stage 9: Jenkins enables rollbacks in deployment.

[Pipeline] echo

Stage 9: Jenkins provides efficient debugging.

[Pipeline] echo

Stage 9: Jenkins allows integration with testing tools.

[Pipeline] echo

Stage 9: Jenkins facilitates automation across environments.

[Pipeline] echo

Stage 9: Jenkins supports master-slave architecture.

[Pipeline] echo

Stage 9: Jenkins can be customized using Groovy scripting.

[Pipeline] echo

Stage 9: Jenkins enables continuous delivery.

[Pipeline] echo

Stage 9: Jenkins provides plugin support for CI/CD integrations.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Stage 10)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

Stage 10: Jenkins offers scalability.

[Pipeline] echo

Stage 10: Jenkins has an intuitive web interface.

[Pipeline] echo

Stage 10: Jenkins can handle pipeline as code.

[Pipeline] echo

Stage 10: Jenkins allows you to manage job configurations easily.

[Pipeline] echo

Stage 10: Jenkins integrates with JIRA for issue tracking.

[Pipeline] echo

Stage 10: Jenkins supports multiple execution environments.

[Pipeline] echo

Stage 10: Jenkins provides integration with Cloud services.

[Pipeline] echo

Stage 10: Jenkins allows you to manage credentials securely.

[Pipeline] echo

Stage 10: Jenkins enables reporting and test result visualizations.

[Pipeline] echo

Stage 10: Jenkins supports various SCM systems including Git, SVN, etc.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS

**EMAIL TRIGGERED**

