

Project 01

Project Overview

Your organization is implementing continuous integration (CI) practices to streamline the software development lifecycle. As part of this initiative, you will create a Jenkins declarative pipeline for building a simple Maven project hosted on GitHub. This project aims to automate the build process, ensure code quality, and facilitate continuous delivery (CD).

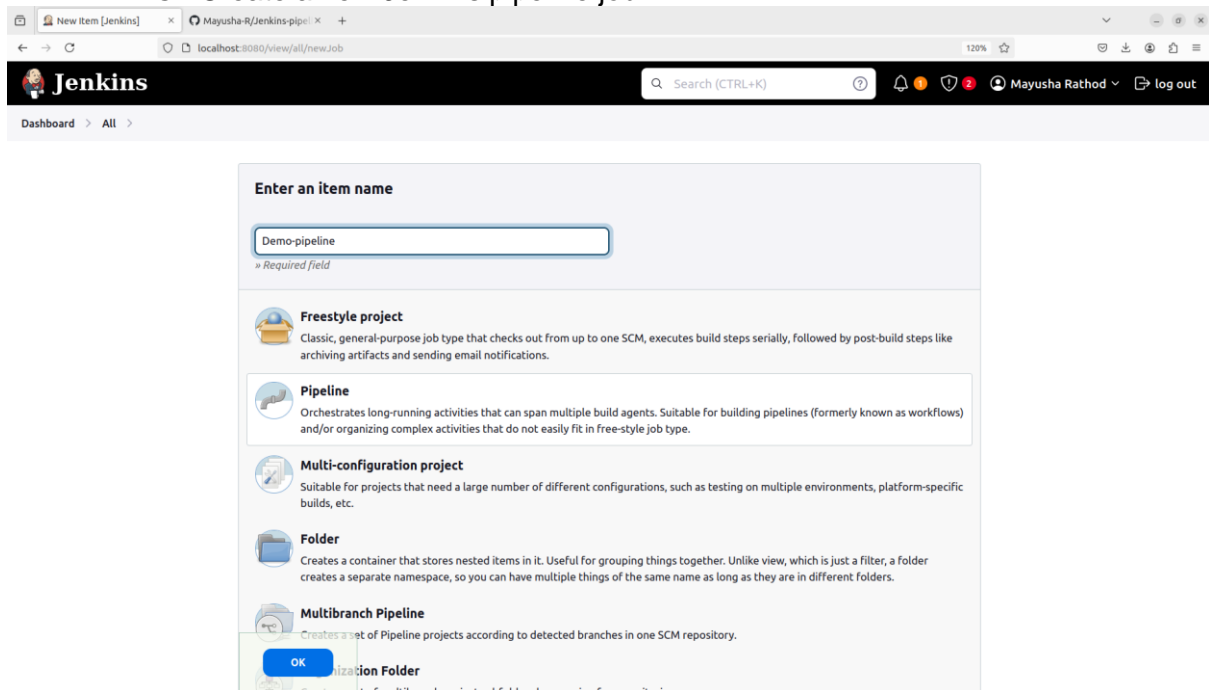
Objectives

- Create a Jenkins pipeline script using declarative syntax.
- Clone a Maven project from a specified GitHub repository.
- Execute the build process and run unit tests.
- Archive build artifacts.
- Provide clear feedback on build status through Jenkins' UI and console output.

Instructions

1. Setup Jenkins Job

- Create a new Jenkins pipeline job.



- Configure the job to pull the Jenkinsfile from the GitHub repository.

The screenshot shows the Jenkins Pipeline Configuration page. The left sidebar has a 'Configure' section with 'General', 'Advanced Project Options', and 'Pipeline' (selected). The main area is titled 'Pipeline' and contains the following fields:

- Definition:** Pipeline script from SCM (dropdown)
- SCM:** Git (dropdown)
- Repositories:** A dashed box containing:
 - Repository URL:** https://github.com/Mayusha-R/Jenkins-pipeline-demo.git
 - Credentials:** none (dropdown)
 - + Add:** button
 - Advanced:** dropdown
 - Add Repository:** button
- Branches to build:** (empty dashed box)
- Buttons:** Save, Apply

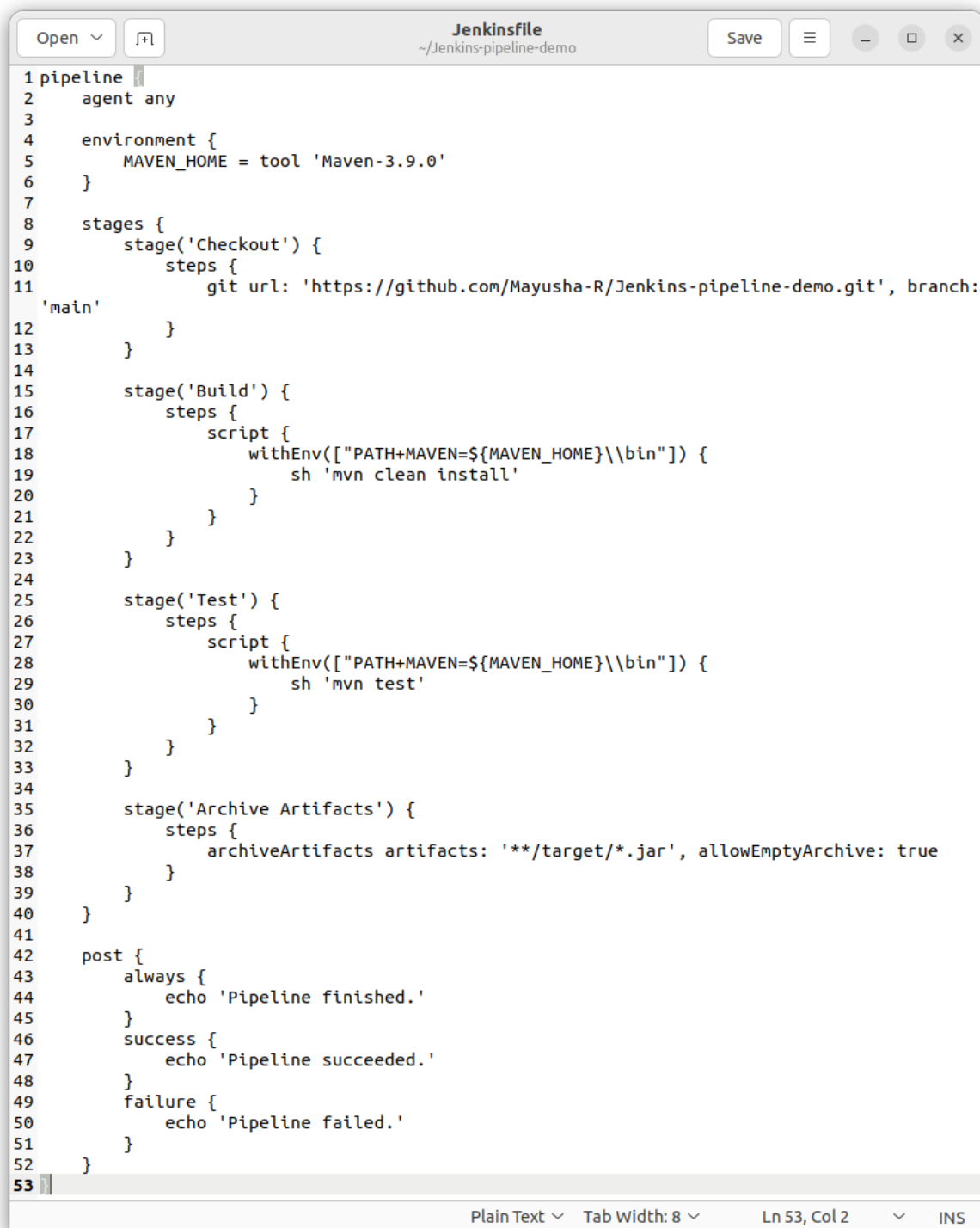
The screenshot shows the Jenkins Pipeline Configuration page, continuing from the previous one. The main area contains the following fields:

- Add Repository:** button
- Branches to build:** A dashed box containing:
 - Branch Specifier (blank for 'any'):** */main
 - Add Branch:** button
- Repository browser:** (Auto) (dropdown)
- Additional Behaviours:** Add (dropdown)
- Script Path:** Jenkinsfile
- Lightweight checkout:** ☒ (checkbox)
- Buttons:** Save, Apply

2. Create Jenkinsfile

- Write a declarative pipeline script (**Jenkinsfile**) that includes the following stages:
 - **Clone Repository:** Clone the Maven project from the GitHub repository.
 - **Build:** Execute the Maven build process (**mvn clean install**).
 - **Test:** Run unit tests as part of the Maven build.
 - **Archive Artifacts:** Archive the build artifacts for future use.

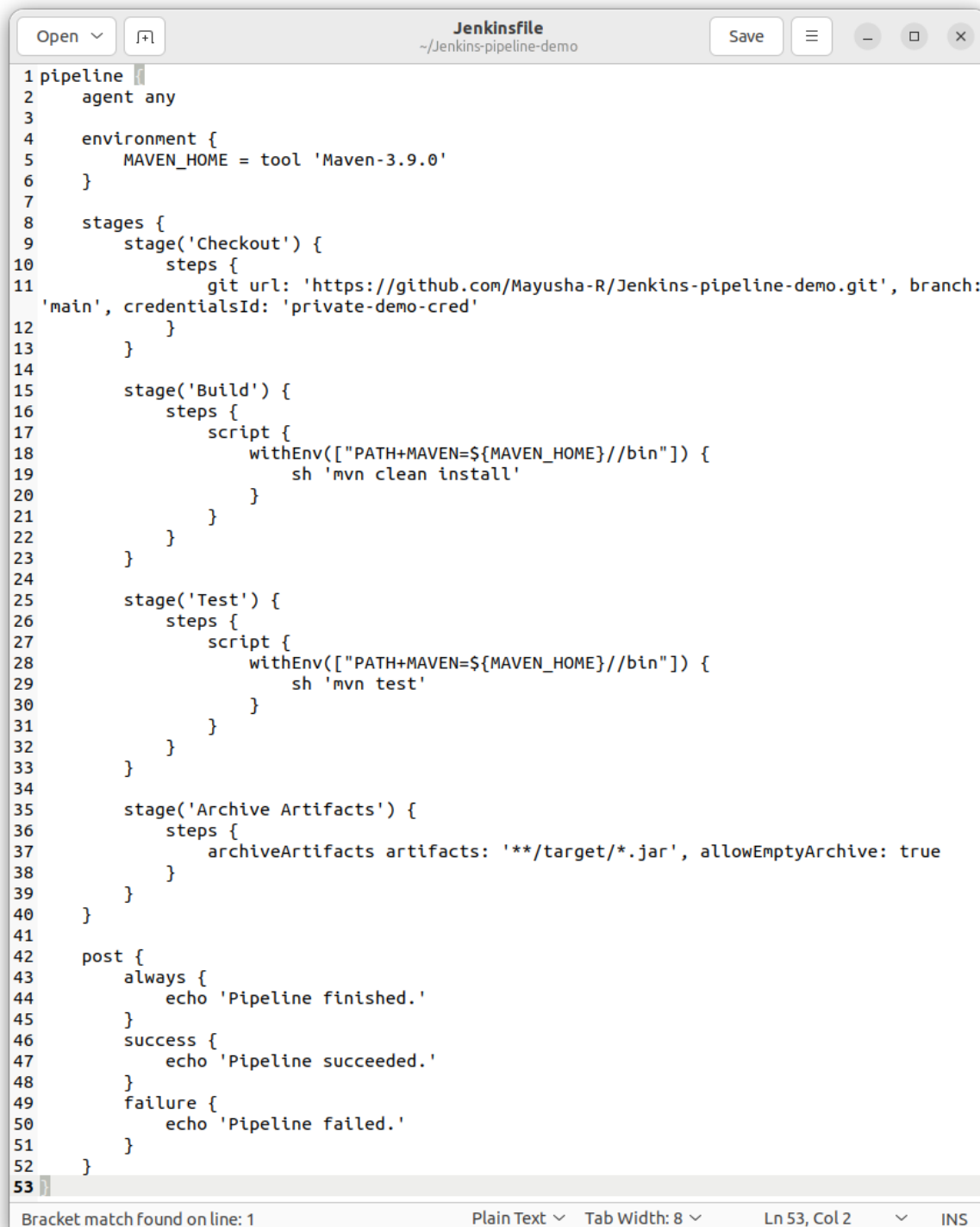
Public Repo:



The image shows a code editor window titled "Jenkinsfile" with the path "~/Jenkins-pipeline-demo". The editor contains a Jenkins pipeline script. The script starts with a pipeline block containing an agent 'any', an environment block setting MAVEN_HOME to 'Maven-3.9.0', and a stages block. The stages block includes 'Checkout', 'Build', 'Test', and 'Archive Artifacts'. The 'Checkout' stage checks out the repository. The 'Build' stage runs 'mvn clean install'. The 'Test' stage runs 'mvn test'. The 'Archive Artifacts' stage archives the target directory. A post block contains always, success, and failure actions with echo commands. The editor interface includes an 'Open' button, a 'Save' button, and a status bar at the bottom showing 'Plain Text', 'Tab Width: 8', 'Ln 53, Col 2', and 'INS'.

```
1 pipeline {
2   agent any
3
4   environment {
5     MAVEN_HOME = tool 'Maven-3.9.0'
6   }
7
8   stages {
9     stage('Checkout') {
10      steps {
11        git url: 'https://github.com/Mayusha-R/Jenkins-pipeline-demo.git', branch:
12        'main'
13      }
14    }
15    stage('Build') {
16      steps {
17        script {
18          withEnv(["PATH+MAVEN=${MAVEN_HOME}\\bin"]) {
19            sh 'mvn clean install'
20          }
21        }
22      }
23    }
24    stage('Test') {
25      steps {
26        script {
27          withEnv(["PATH+MAVEN=${MAVEN_HOME}\\bin"]) {
28            sh 'mvn test'
29          }
30        }
31      }
32    }
33    stage('Archive Artifacts') {
34      steps {
35        archiveArtifacts artifacts: '**/target/*.jar', allowEmptyArchive: true
36      }
37    }
38  }
39
40  post {
41    always {
42      echo 'Pipeline finished.'
43    }
44    success {
45      echo 'Pipeline succeeded.'
46    }
47    failure {
48      echo 'Pipeline failed.'
49    }
50  }
51 }
52
53
```

Private Repo :



The screenshot shows a text editor window titled "Jenkinsfile" with the path "~/Jenkins-pipeline-demo". The editor contains a Jenkins pipeline script. The script defines a pipeline with an agent, environment, stages (Checkout, Build, Test, Archive Artifacts), and a post section with always, success, and failure blocks. The script is as follows:

```
1 pipeline {
2   agent any
3
4   environment {
5     MAVEN_HOME = tool 'Maven-3.9.0'
6   }
7
8   stages {
9     stage('Checkout') {
10      steps {
11        git url: 'https://github.com/Mayusha-R/Jenkins-pipeline-demo.git', branch:
12        'main', credentialsId: 'private-demo-cred'
13      }
14    }
15
16    stage('Build') {
17      steps {
18        script {
19          withEnv(["PATH+MAVEN=${MAVEN_HOME}//bin"]) {
20            sh 'mvn clean install'
21          }
22        }
23      }
24    }
25
26    stage('Test') {
27      steps {
28        script {
29          withEnv(["PATH+MAVEN=${MAVEN_HOME}//bin"]) {
30            sh 'mvn test'
31          }
32        }
33      }
34    }
35
36    stage('Archive Artifacts') {
37      steps {
38        archiveArtifacts artifacts: '**/target/*.jar', allowEmptyArchive: true
39      }
40    }
41
42    post {
43      always {
44        echo 'Pipeline finished.'
45      }
46      success {
47        echo 'Pipeline succeeded.'
48      }
49      failure {
50        echo 'Pipeline failed.'
51      }
52    }
53 }
```

At the bottom of the editor, a status bar shows "Bracket match found on line: 1", "Plain Text", "Tab Width: 8", "Ln 53, Col 2", and "INS".

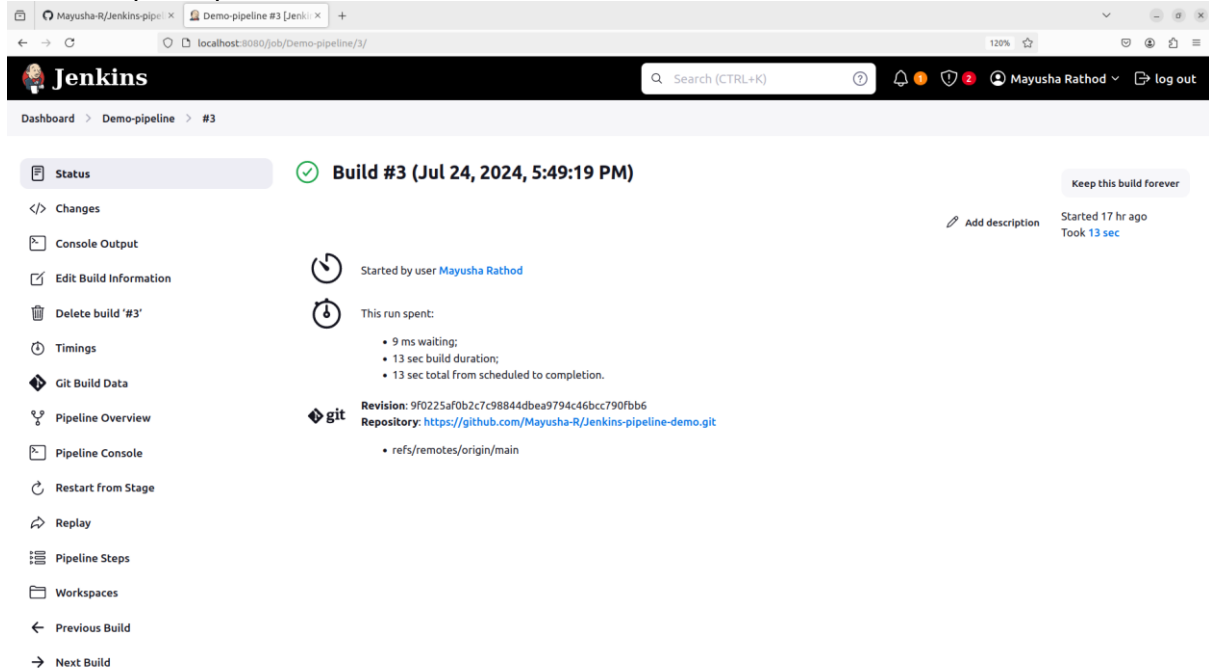
3. Configure Pipeline Parameters

- Allow the pipeline to accept parameters such as Maven goals and options for flexibility.
- Ensure the pipeline can be easily modified for different build configurations.

4. Run the Pipeline

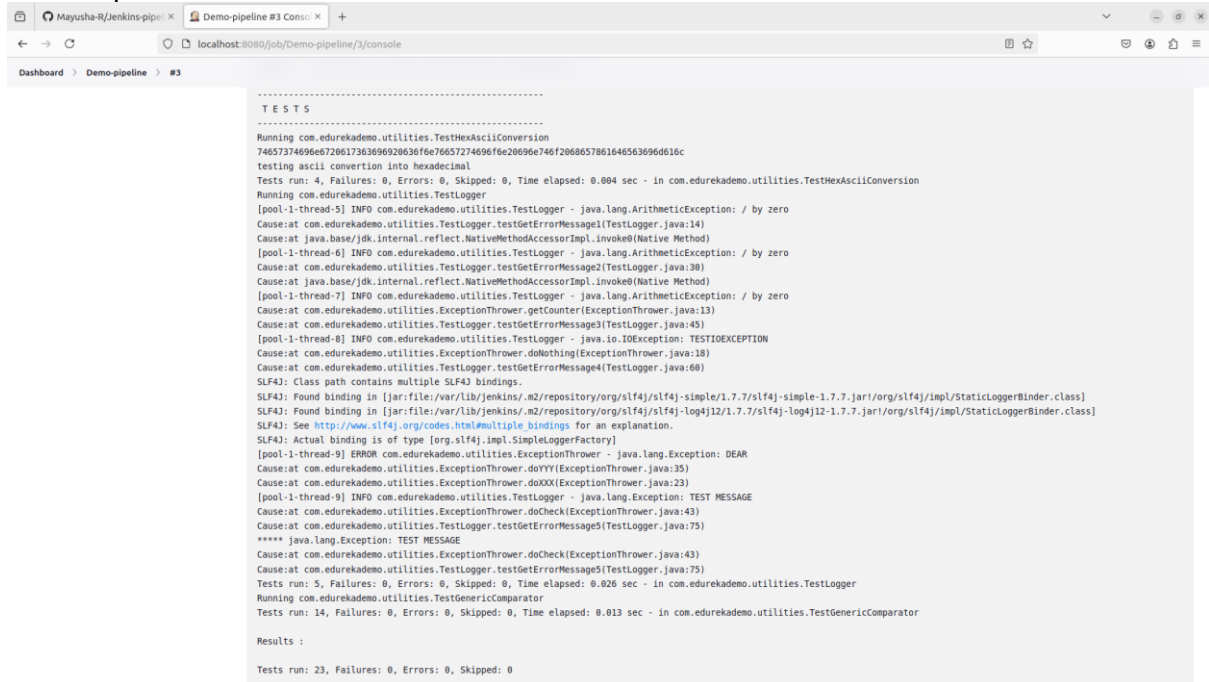
- Trigger the Jenkins pipeline job manually or set up a webhook for automatic triggering on GitHub repository changes.
- Monitor the build process through Jenkins' UI and console output.

Public Repo output:



The screenshot shows the Jenkins web interface for a build named 'Demo-pipeline #3'. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Mayusha Rathod'. The main content area is titled 'Build #3 (Jul 24, 2024, 5:49:19 PM)' and features a green success icon. On the left, a sidebar lists various actions: Status, Changes, Console Output, Edit Build Information, Delete build '#3', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, Previous Build, and Next Build. The main panel displays build details: 'Started by user Mayusha Rathod', 'This run spent: 9 ms waiting, 13 sec build duration, 13 sec total from scheduled to completion.', and 'Revision: 9f0225af0b2c7c98844dbee9794c46bcc790fbb6' from the repository 'https://github.com/Mayusha-R/Jenkins-pipeline-demo.git'. A 'Keep this build forever' button is visible in the top right.

Test output:



The screenshot displays the 'Console Output' for the Jenkins build. The output shows the execution of tests for 'com.edurekademo.utilities'. It begins with 'T E S T S' and 'Running com.edurekademo.utilities.TestHexAsciiConversion'. The tests run successfully, with 23 tests passed, 0 failures, 0 errors, and 0 skipped. The output includes detailed logs for each test, such as 'testing ascii conversion into hexadecimal', 'SLF4J: Found binding in [jar:file:/var/lib/jenkins/m2/repository/org/slf4j/slf4j-simple/1.7.7/slf4j-simple-1.7.7.jar/org/slf4j/impl/StaticLoggerBinder.class]', and 'SLF4J: Actual binding is of type [org.slf4j.impl.SimpleLoggerFactory]'. The final summary line reads: 'Tests run: 23, Failures: 0, Errors: 0, Skipped: 0'.

Pipeline overview:

Jenkins-pipeline-demo/

Graph [Demo-pipeline #3]

DevOps-EF-July-24 - Google

Day 13 - Google Docs

localhost:8080/me/my-views/view/all/job/Demo-pipeline/3/pipeline-graph/

Jenkins

Search (CTRL+K)

Mayusha Rathod

log out

Dashboard > Mayusha Rathod > My Views > All > Demo-pipeline > #3 > Pipeline Overview

< Build #3 >

Rebuild Console Configure

Pipeline

Start Checkout SCM Checkout Build Test Archive Artifacts Post Actions End

Details

Manually run by Mayusha Rathod

Started 20 hr ago

Queued 4 ms

Took 13 sec

Jenkins 2.452.3

Jenkins-pipeline-demo/

Build log [#3] [Jenkins]

DevOps-EF-July-24 - Google

Day 13 - Google Docs

localhost:8080/me/my-views/view/all/job/Demo-pipeline/3/pipeline-console/

Jenkins

Search (CTRL+K)

Mayusha Rathod

log out

Dashboard > Mayusha Rathod > My Views > All > Demo-pipeline > #3 > Pipeline Console

< Build #3 >

Rebuild Overview Configure

Success 20 hr ago in 13 sec

Checkout SCM

Checkout

Build

Test

Archive Artifacts

Post Actions

Stage "Post Actions"

Started 20 hr ago

Queued 0 ms

Took 0.11 sec

Success

View as plain text

Pipeline finished.

Print Message

44 ms

Pipeline succeeded.

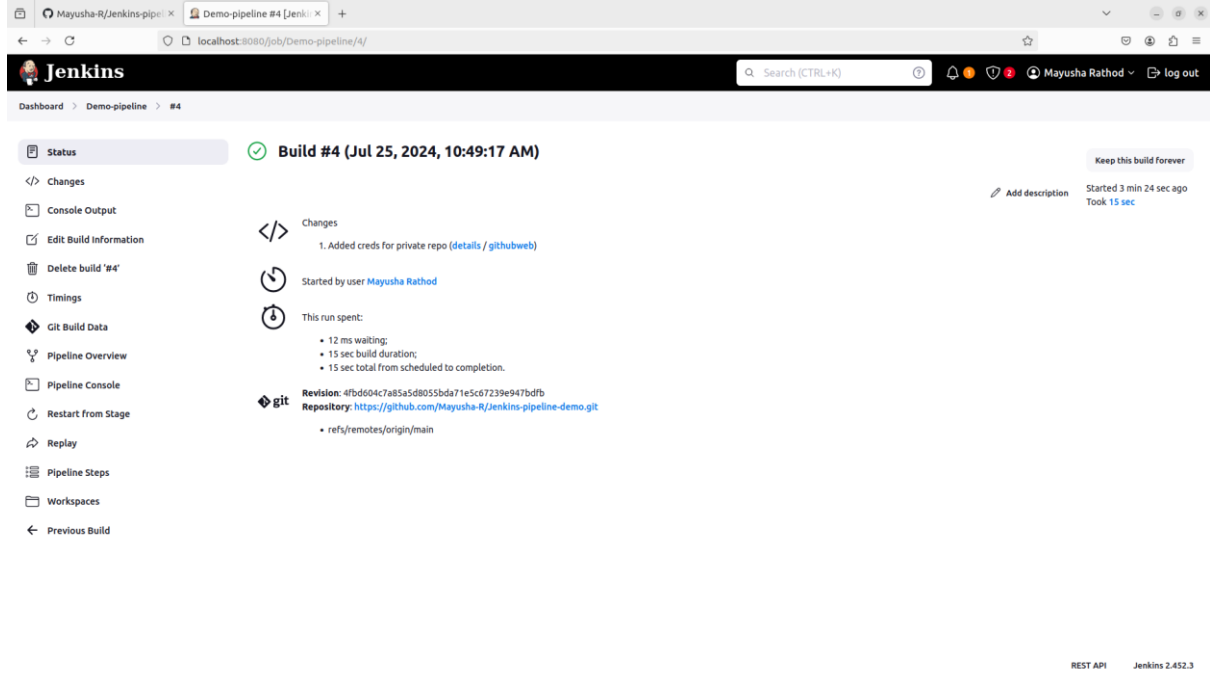
Print Message

30 ms

Pipeline succeeded.

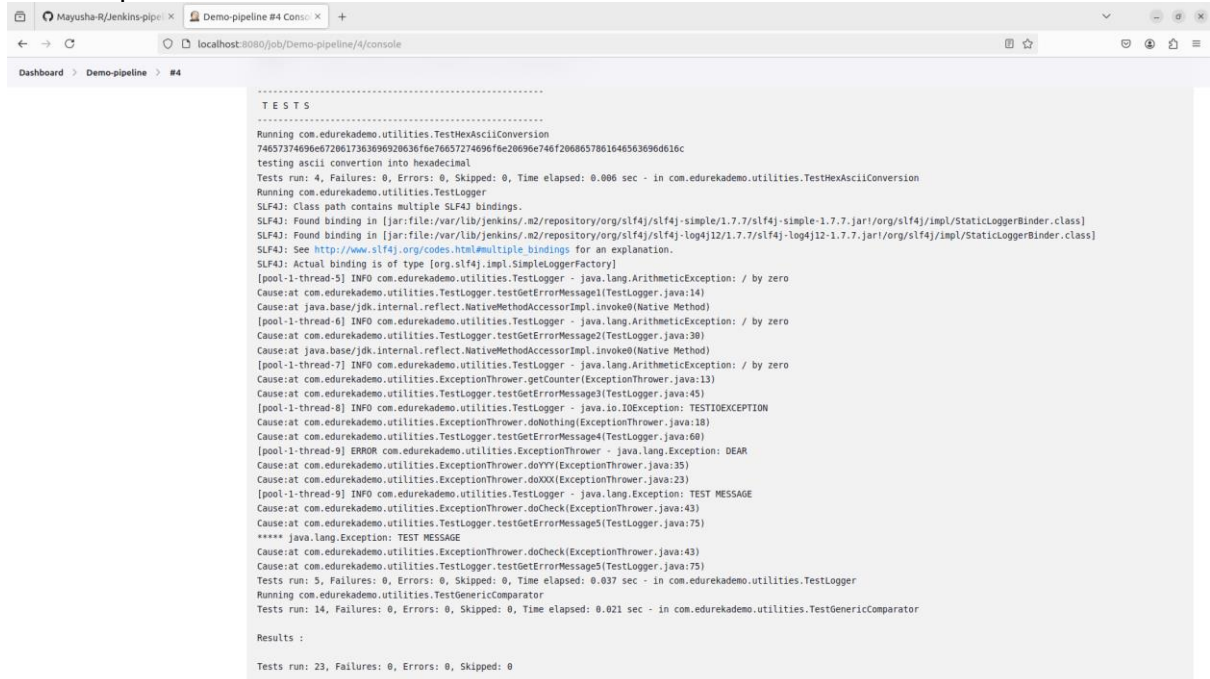
Jenkins 2.452.3

Private Repo output:



The screenshot shows the Jenkins web interface for a build named 'Build #4' (dated Jul 25, 2024, 10:49:17 AM). The interface includes a left sidebar with navigation links: Status, Changes, Console Output, Edit Build Information, Delete build '#4', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area displays build details: a green status icon, a link to 'Add description', and a 'Keep this build forever' button. It lists changes (1. Added creds for private repo), start time (Started by user Mayusha Rathod), and run times (12 ms waiting, 15 sec build duration, 15 sec total from scheduled to completion). It also shows the revision (4fbd604c7a85a5d8055bda71e5c67239e947bdfb) and repository (https://github.com/Mayusha-R/Jenkins-pipeline-demo.git). The bottom right corner indicates 'REST API' and 'Jenkins 2.452.3'.

Test Output:



The screenshot shows the Jenkins console output for Build #4. The output is a text log of test results. It starts with 'T E S T S' and 'Running com.edurekademo.utilities.TestHexAsciiConversion'. The log shows the execution of a test suite, including the output of 'Testing ascii conversion into hexadecimal'. It reports 'Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec'. The log then shows the execution of 'Running com.edurekademo.utilities.TestLogger', which outputs 'SLF4J: Class path contains multiple SLF4J bindings.' and 'SLF4J: Found binding in [jar:file:/var/lib/jenkins/.m2/repository/org/slf4j/slf4j-simple/1.7.7/slf4j-simple-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]'. It then shows the execution of 'Running com.edurekademo.utilities.TestGenericComparator', which reports 'Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.021 sec'. The log ends with 'Results : Tests run: 23, Failures: 0, Errors: 0, Skipped: 0'.

< Build #4

Rebuild

Console

Configure

Pipeline

Start

Checkout SCM

Checkout

Build

Test

Archive Artifacts

Post Actions

End

Details

Manually run by Mayusha Rathod

Started 3 hr 4 min ago

Queued 6 ms

Took 15 sec

Jenkins 2.452.3

< Build #4

Rebuild

Overview

Configure

...

Success 3 hr 5 min ago in 15 sec

- Checkout SCM
- Checkout
- Build
- Test
- Archive Artifacts
- Post Actions

Stage "Post Actions"

Started 3 hr 5 min ago

Queued 0 ms

Took 72 ms

Success

View as plain text

Pipeline finished.

Print Message

19 ms

Copy

Close

Pipeline succeeded.

Print Message

30 ms

Copy

Close

Pipeline succeeded.

Jenkins 2.452.3