Module 6: Automated and Continuous Deployment

Demo Document - 2

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Demo: Deployment of Java app with Jenkins pipeline

Problem statement: Execute commands to do deployment of JAVA application using Jenkins pipeline project

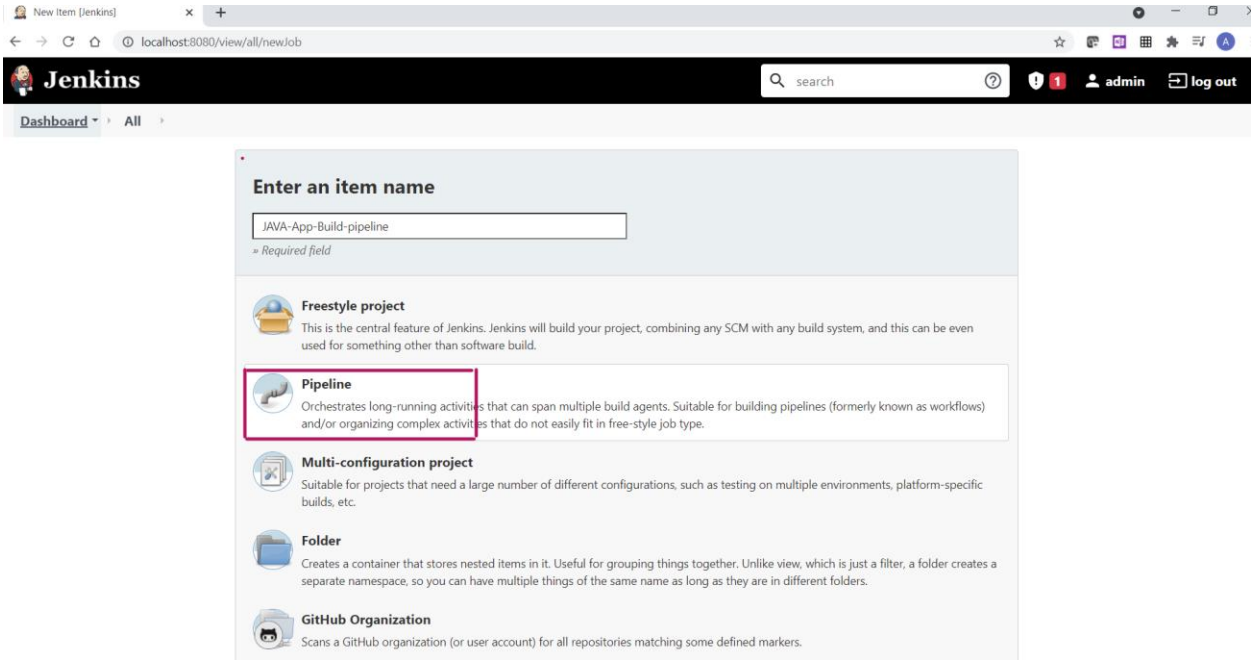
Solution:

Step 1: To run Jenkins, follow relevant instructions provided in Module 7/Lab 1. Let's assume Jenkins server is started.

Now we will create a Jenkins pipeline project.

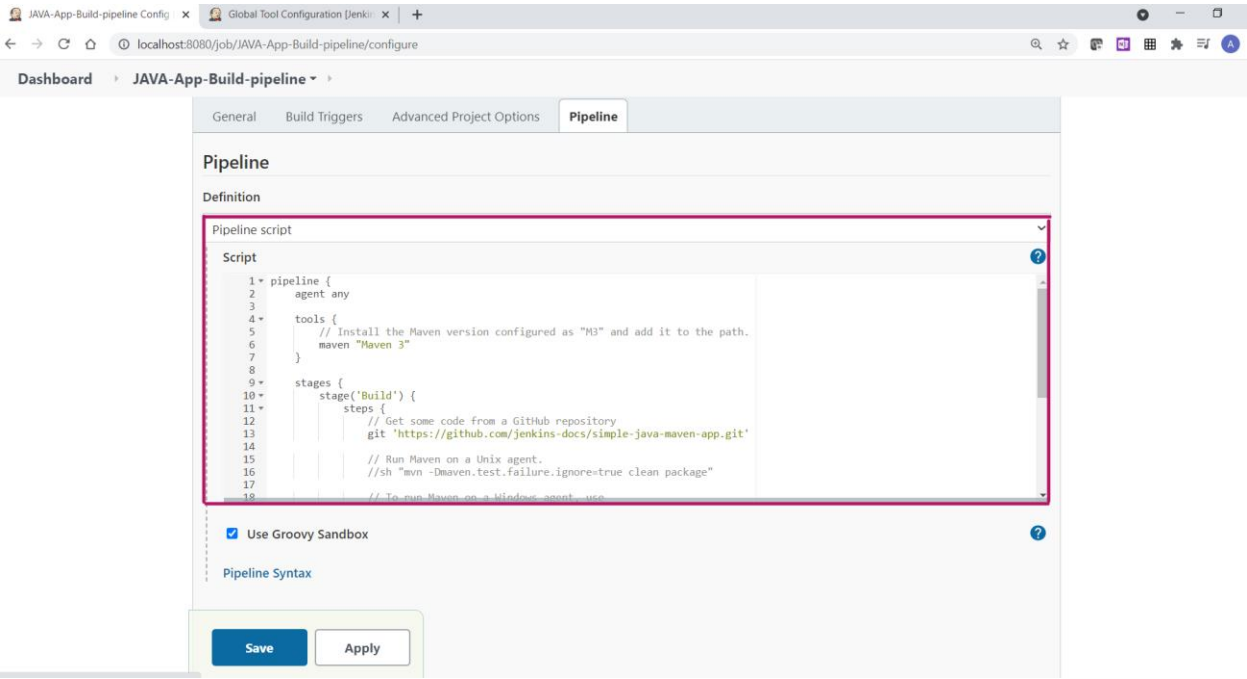
URL to access: http://localhost:8080/view/all/newJob

Please create a pipeline project named as 'JAVA-App-Build-pipeline'.



Step 2: Now we can configure the pipeline project.

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline/configure



Please add below pipeline script in the Script section

Pipeline Script to be added:

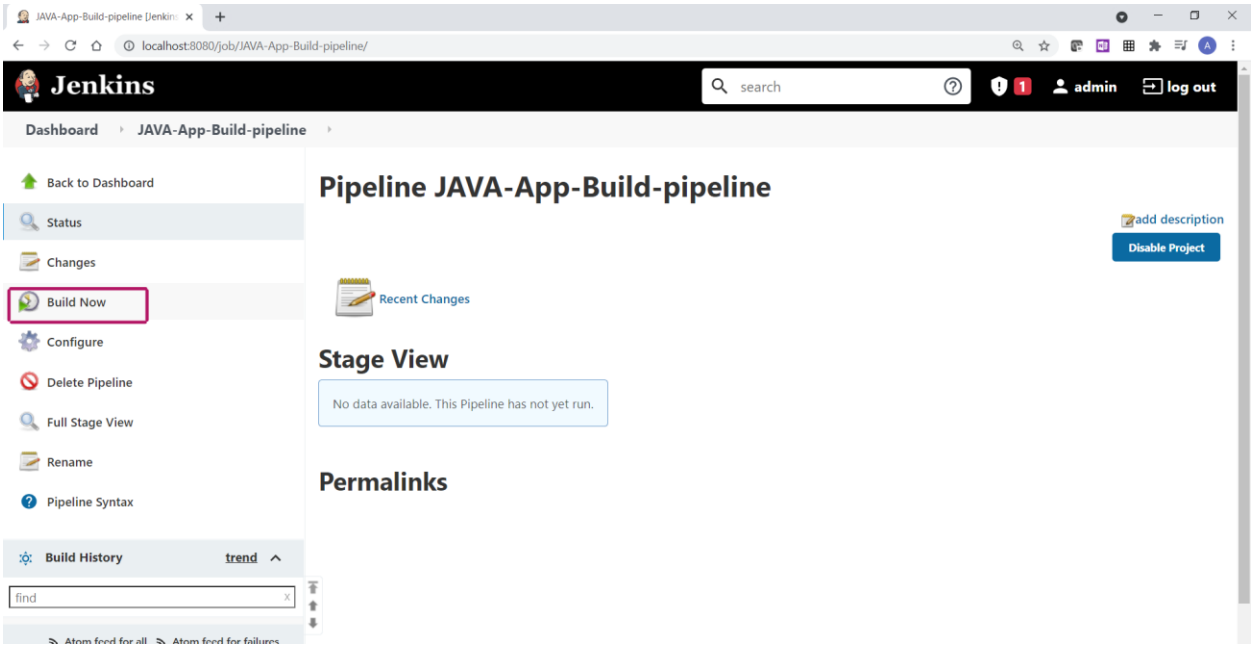
```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "Maven 3"
    }

    stages {
        stage('Build') {
```

```
        steps {
            // Get some code from a GitHub repository
            git 'https://github.com/jenkins-docs/simple-java-maven-app.git'

            // Run Maven on a Unix agent.
            //sh "mvn -Dmaven.test.failure.ignore=true clean package"

            // To run Maven on a Windows agent, use
            bat "mvn -Dmaven.test.failure.ignore=true clean package"
        }

        post {
            // If Maven was able to run the tests, even if some of the test
            // failed, record the test results and archive the jar file.
            success {
                junit '**/target/surefire-reports/TEST-*.xml'
                archiveArtifacts 'target/*.jar'
            }
        }
    }
  }
}
```

Declarative pipeline is recently added to Jenkins pipeline project. It provides an opinionated syntax for creating the script. This approach also follow same rules as groovy's syntax.

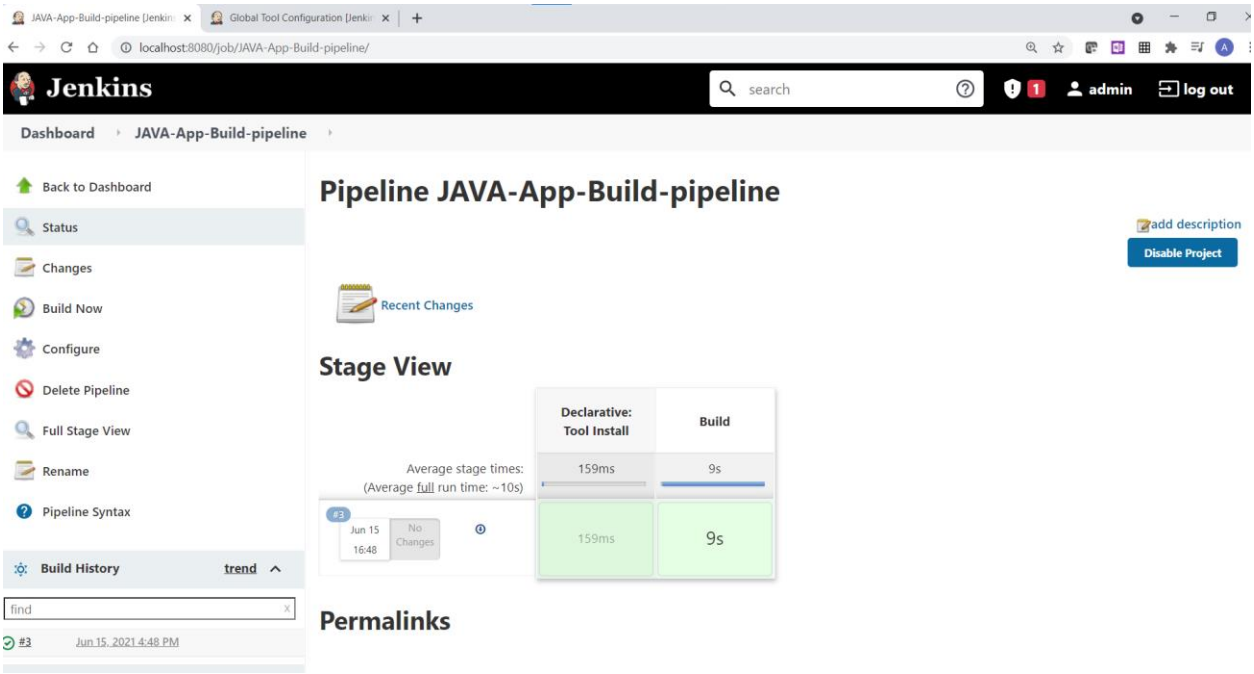
Step 3: You can now build the project. This is a similar option as provided in freestyle project.

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline/



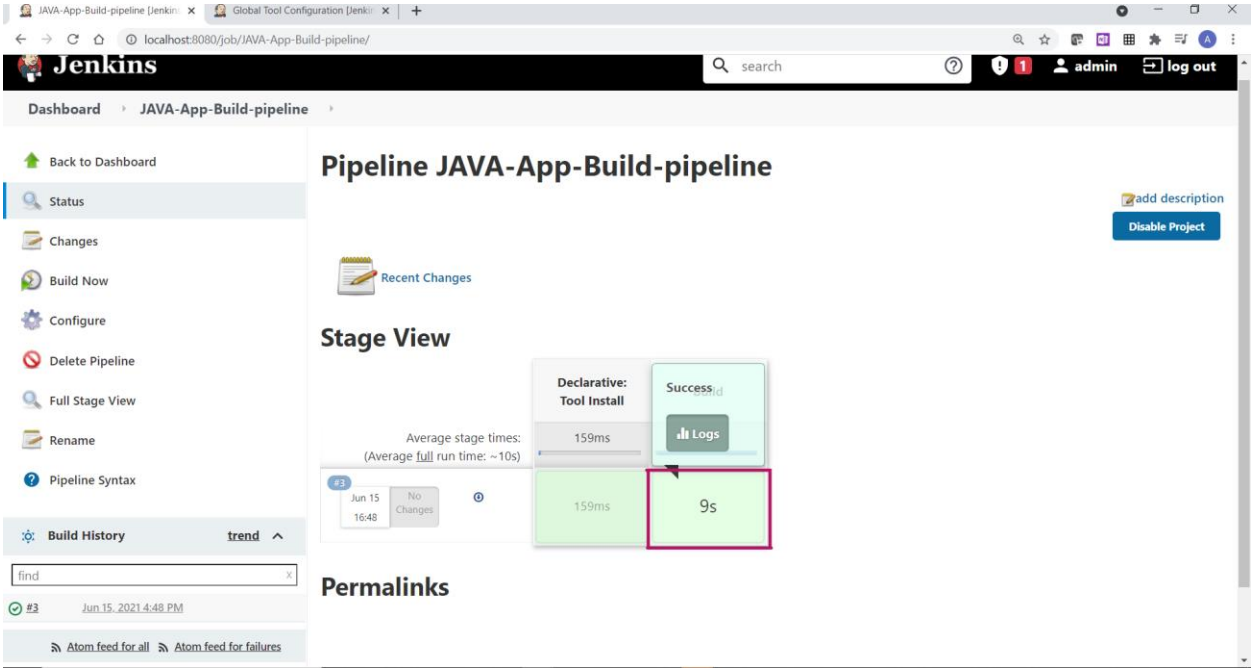
Step 4: You can also view whether build is successful or failed

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline/



Step 5: View logs generated for the build.

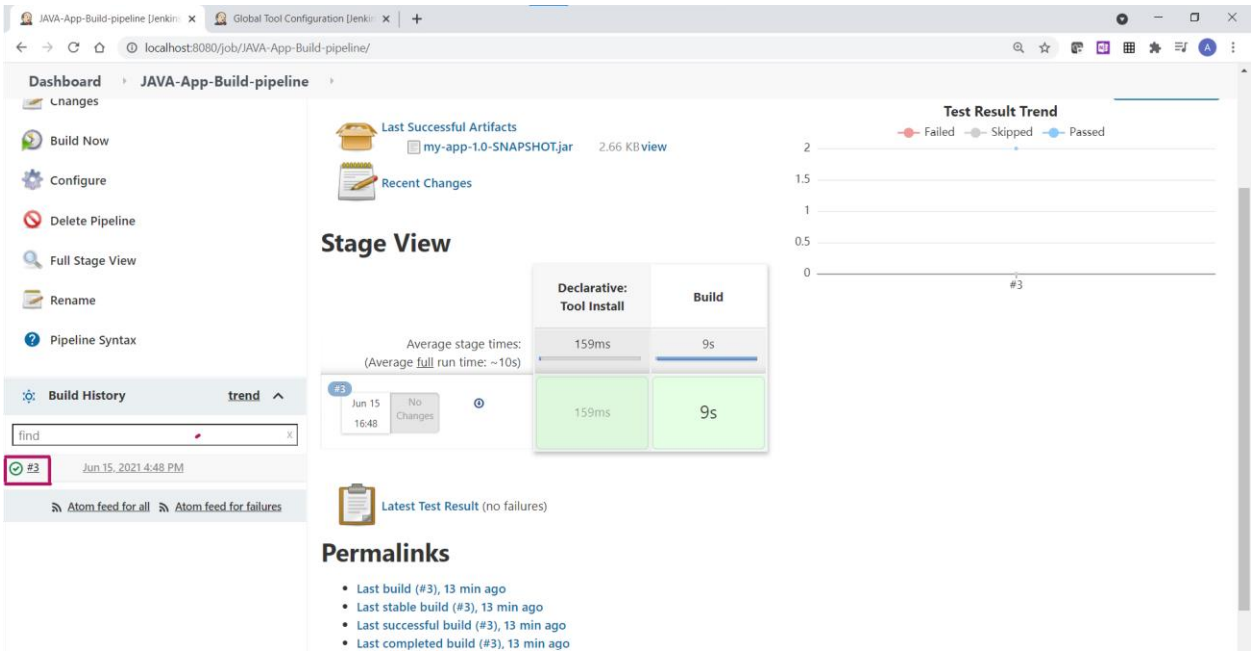
URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline/



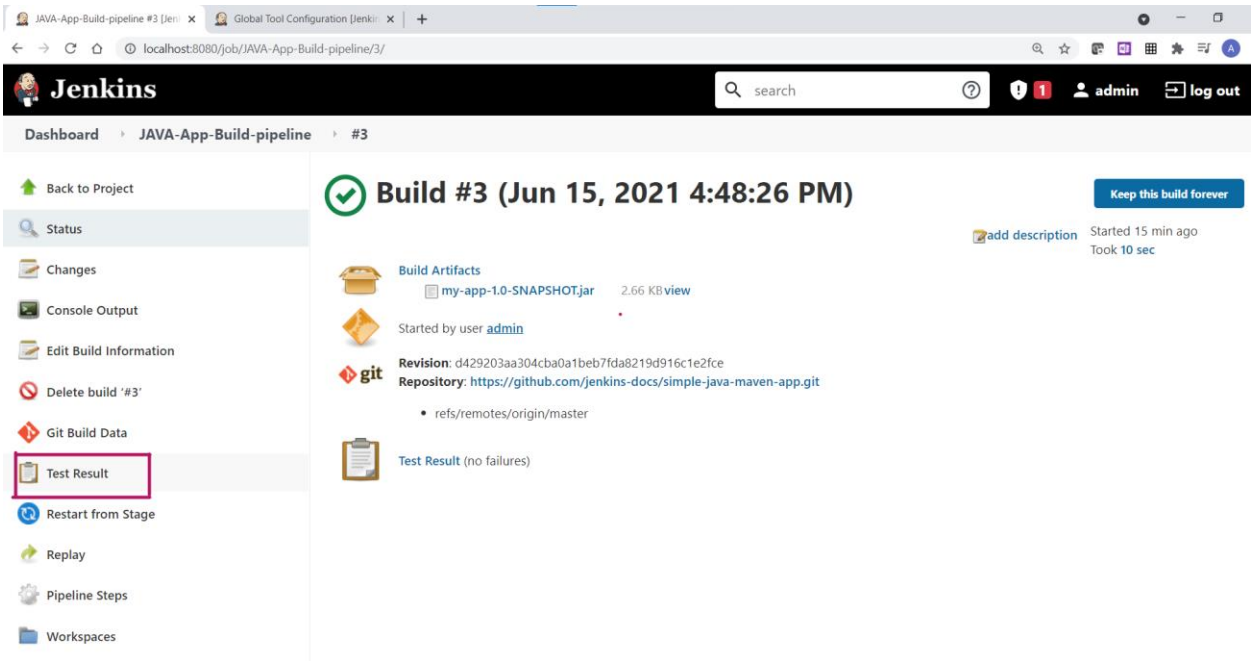
This build will have access to the artifact generated, test reports and useful information about the build execution in the form of logs.

Step 6: Now your build is successful. Let's access Test reports, workspaces and artifacts for a specific build

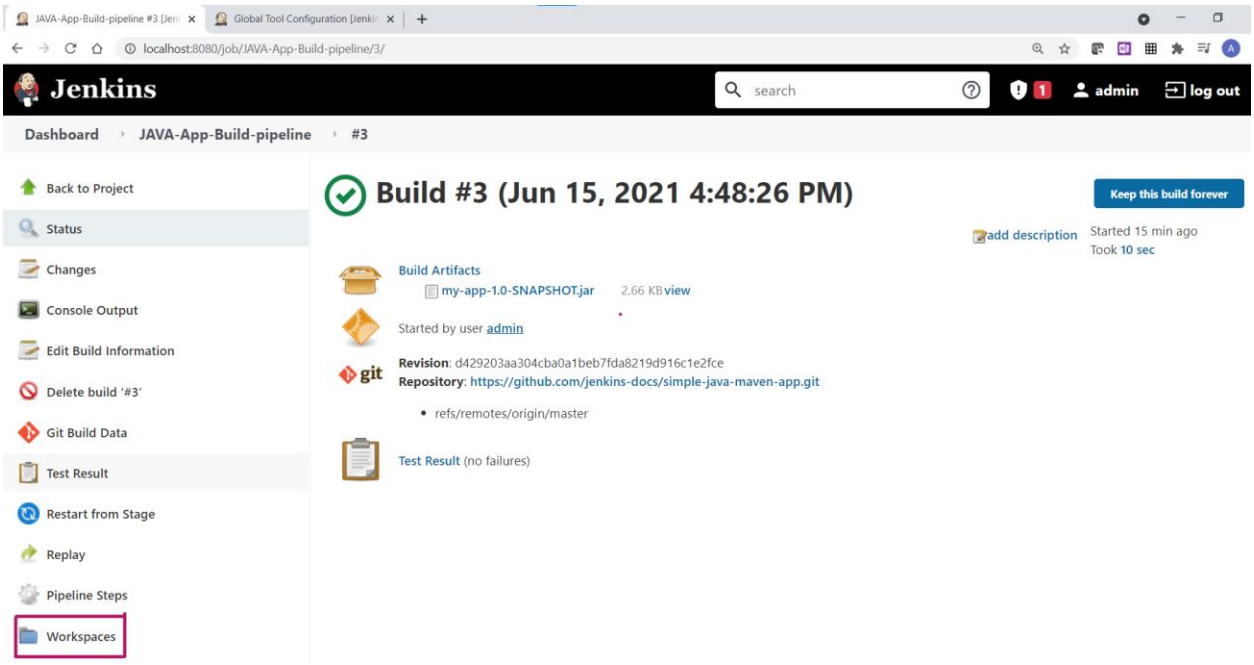
URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline/1/ [1 is the build number]

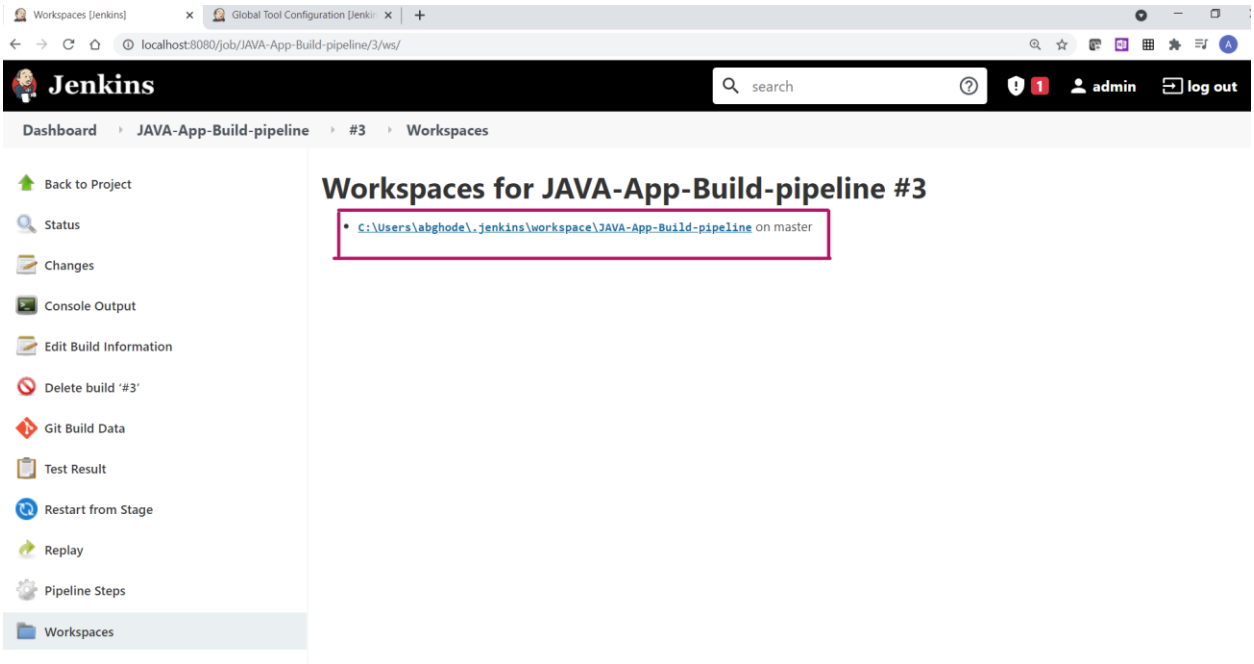


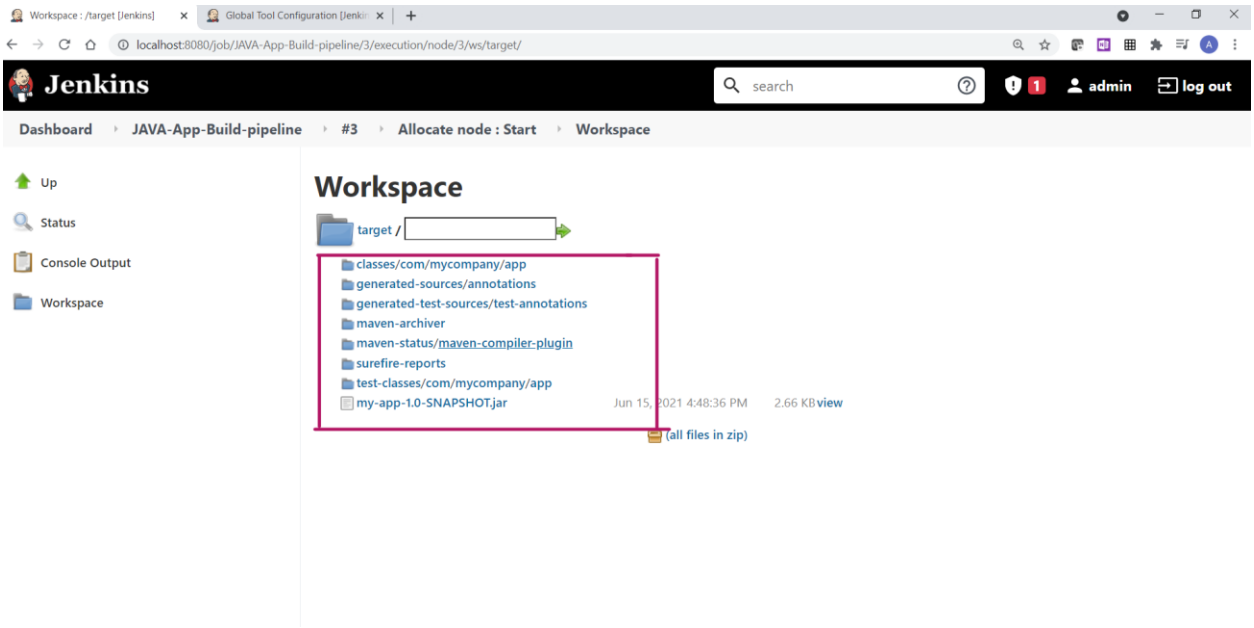
Click on 'Test Result' to observe how many test cases passed.

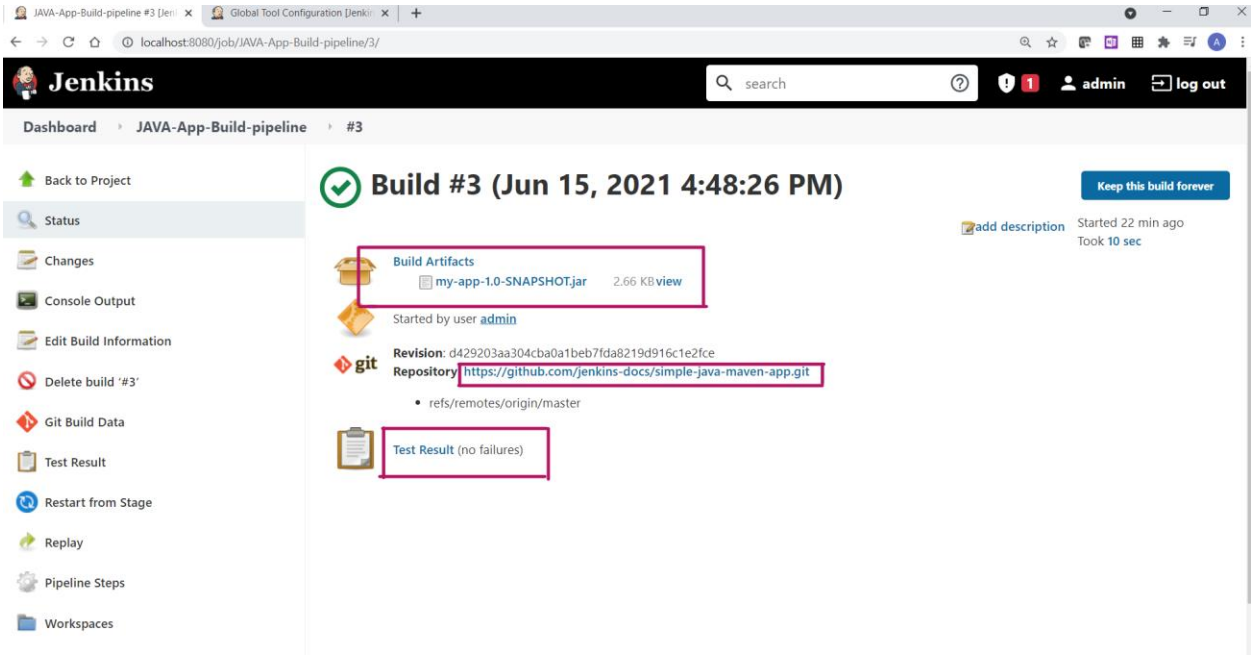


Click on ‘Workspaces’ to access the project contents available on Jenkins server. You would be able to access artifacts generated by the successful build.







Step 7: You can also have an access to build artifacts, git repository and test results on a single view.

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline/3/