Practice for Lesson 4: Integrating Jenkins Pipelines with Jobs

Practices for Lesson 4

Overview

In these practices, you will trigger the Jenkins Jobs from one job to another automatically in the Jenkins Dashboard, create the pipeline using Groovy script in Jenkins instance and learn to integrate the GitHub source code to the Jenkins pipeline, and further delegate a pipeline Job using Agent in Jenkins Instance.

Practice 4-2: Create Pipeline Using Groovy Script in Jenkins

Overview

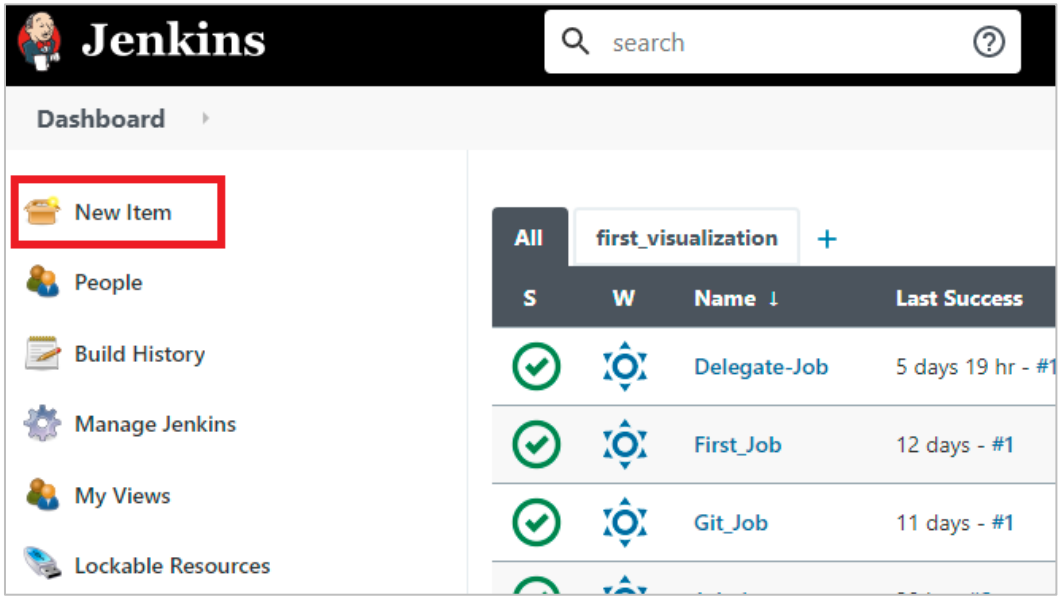
In this practice, you will learn how to create the pipeline using Groovy scripting in Jenkins instance using a sample example.

Assumptions

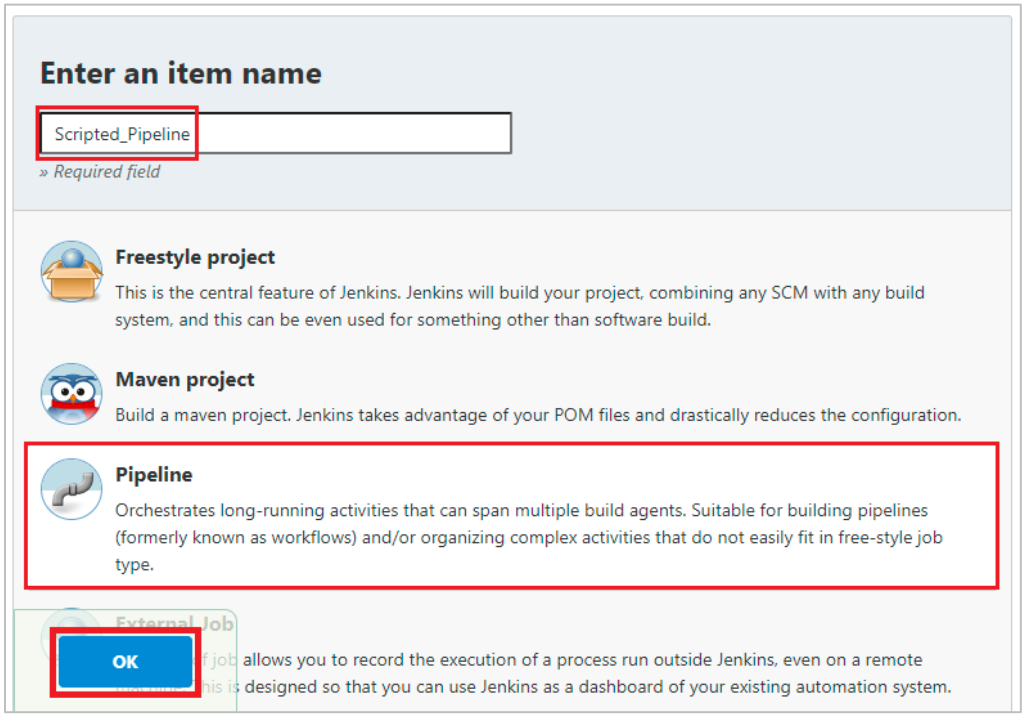
You should have completed the Practice of Lesson 4-1.

Tasks

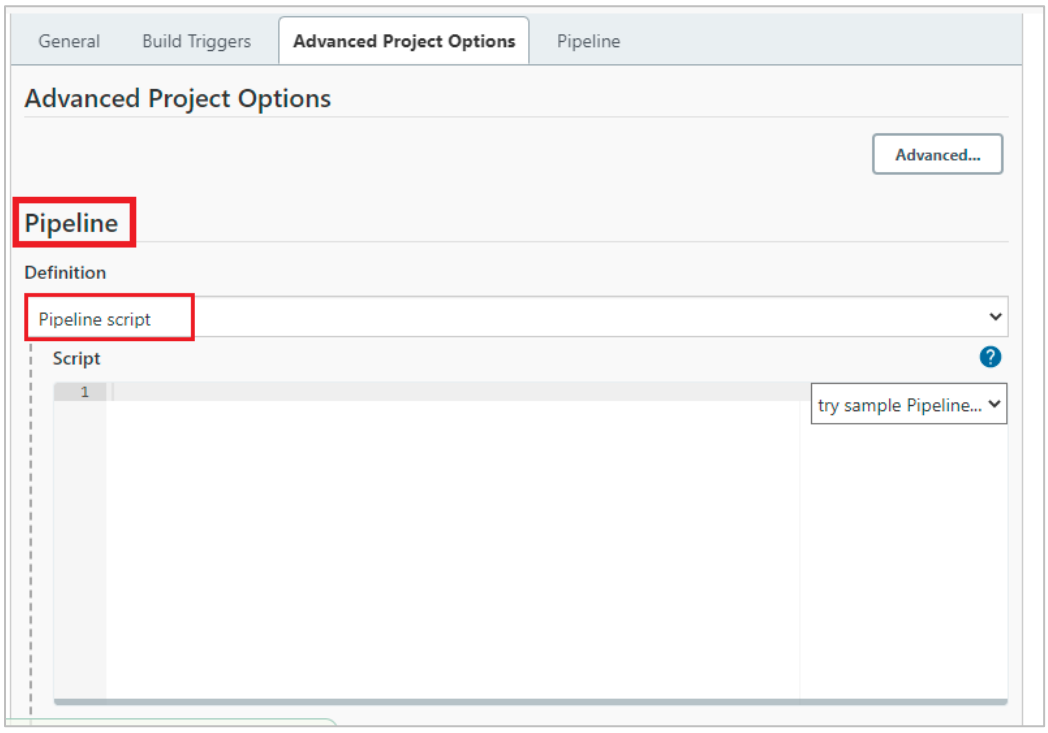
1. Create a scripted pipeline using the Groovy scripting in Jenkins instance.
   a. In the Jenkins Dashboard, navigate to main menu and select **New Item** to create a Groovy Scripted Pipeline as shown below.



   b. Provide the **name** for the Pipeline, select **Pipeline** and click **OK** as shown below.



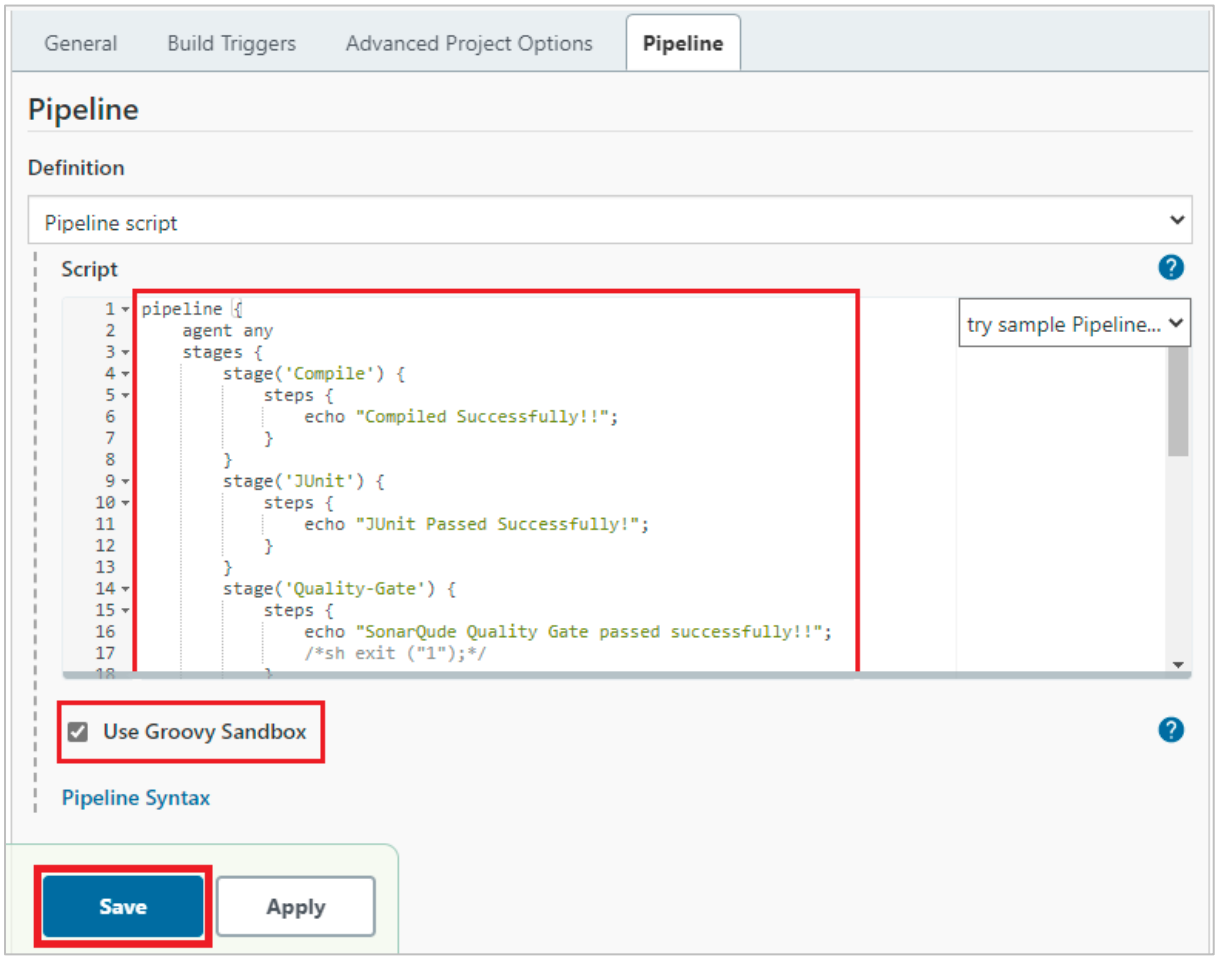
c. Scroll down to **Pipeline**, select **Pipeline script** under **Definition** as shown below.



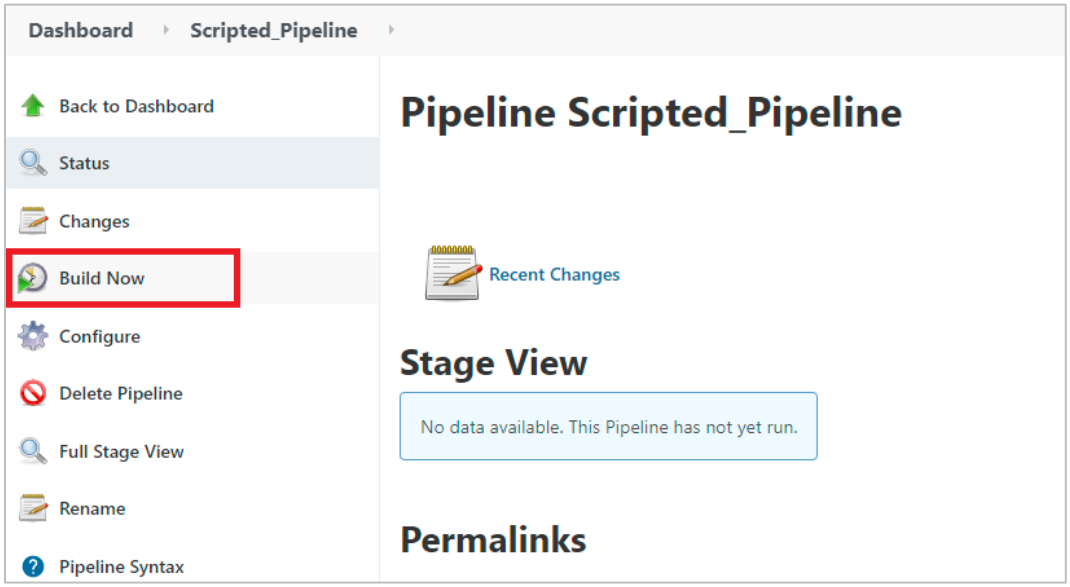
d. Copy the Groovy script provided below and paste it in the **Script** block as shown below. Select the checkbox of **Use Groovy Sandbox** and click **Save**.

```
pipeline {
      agent any
      stages {
            stage('Compile') {
                  steps {
                        echo "Compiled Successfully!!";
                  }
            }
            stage('JUnit') {
                  steps {
                        echo "JUnit Passed Successfully!";
                  }
            }
            stage('Quality-Gate') {
                  steps {
                        echo "SonarQude Quality Gate passed
successfully!!";
                        /*sh exit ("1");*/
                  }
            }
            stage('Deploy') {
                  steps {
                        echo "Pass!";
                  }
            }
      }
      post {
            always {
                  echo 'This will always run'
            }
            success {
                  echo 'This will run only if successful'
            }
            failure {
                  echo 'This will run only if failed'
            }
            unstable {
                  echo 'This will run only if the run was marked as
unstable'
            }
            changed {
```

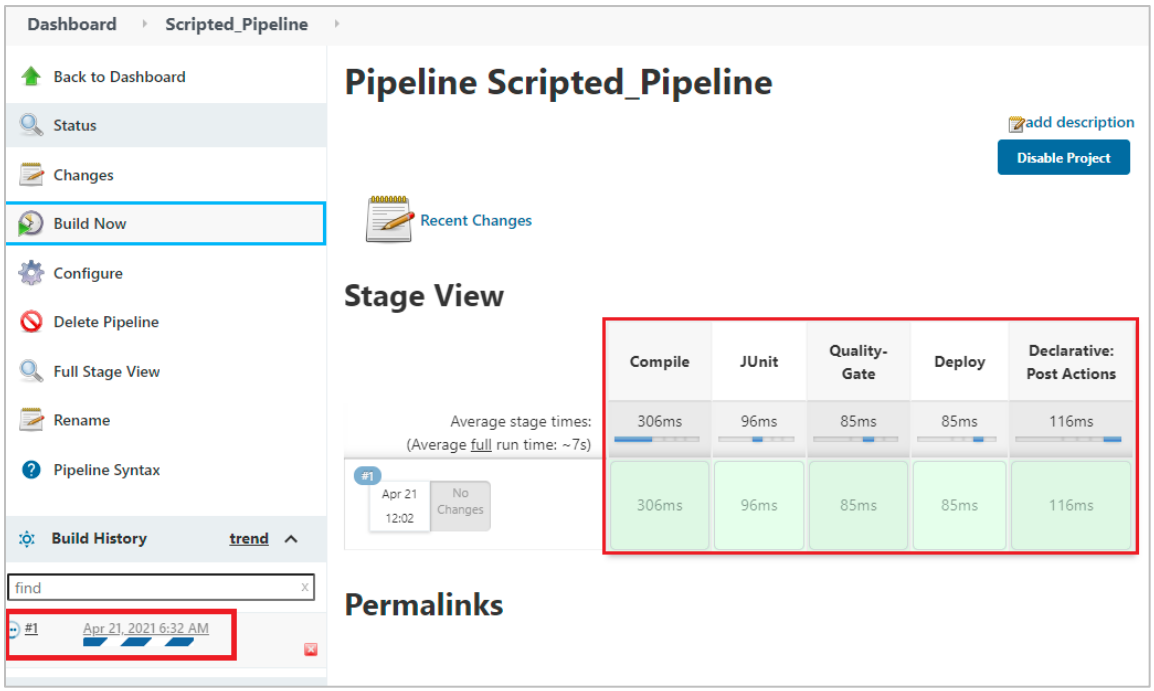
```
                    echo 'This will run only if the state of the pipeline
has changed'
                    echo 'For example, if the Pipeline was previously
failing but is now successful'
                }
        }
}
```



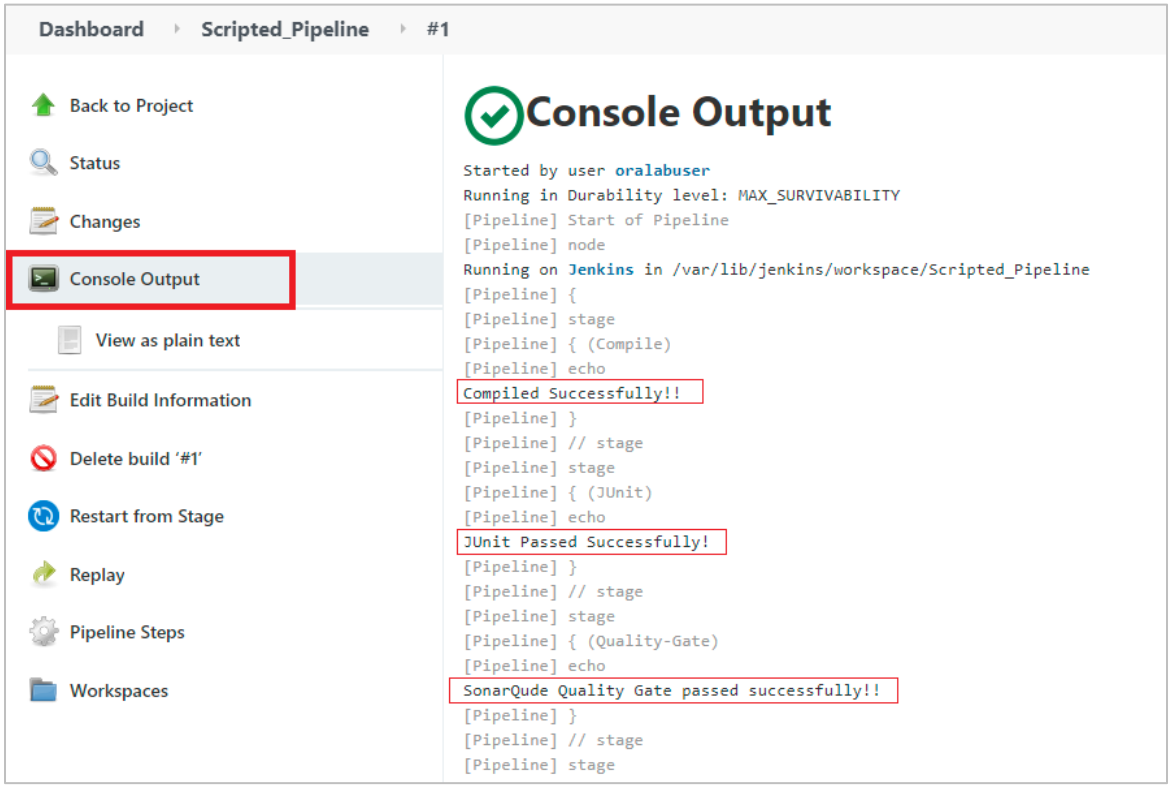
e. Scripted pipeline is created successfully. Click **Build Now** to execute the pipeline as shown below.



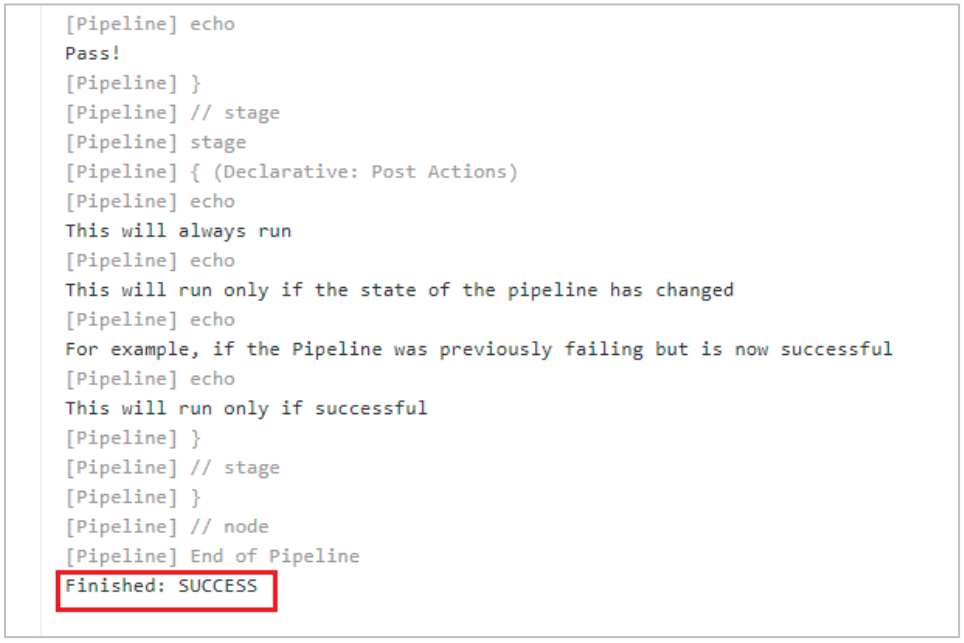
f. As shown below, **view** the **stages** of **pipeline** getting executed and click on the link under **Build History** as shown below.



g. Navigate to **Console Output** to view the execution output of the scripted pipeline.



h. As shown below, the scripted pipeline sample is executed successfully.



2. Keep the Jenkins Dashboard and the AWS Management Console open for the next practice.