Practice for Lesson 5: Understanding of Parallel Jenkins Jobs and Jenkins Slave on AWS

Practices for Lesson 5

Overview

In these practices, you will learn how to Build and Deploy an Application to Webserver using Jenkins Pipeline. Further create a parallel Agent Pipeline Job on Jenkins.

.

Practice 5-3: Create a DevOps Pipeline Job on Jenkins

Overview

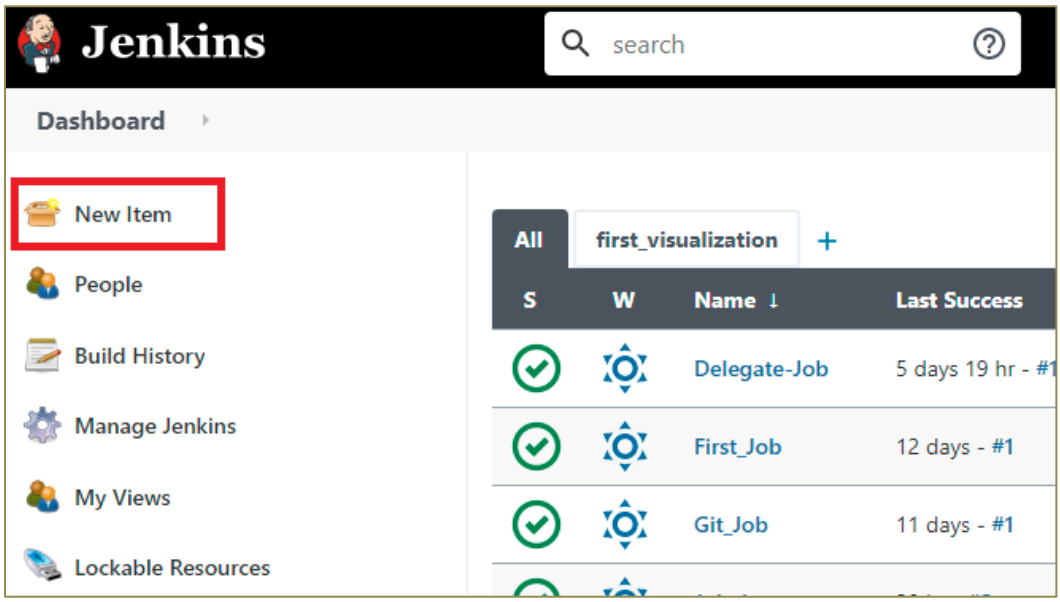
In this practice, you will learn how to create the DevOps Pipeline Job on Jenkins instance using as sample example.

Assumptions

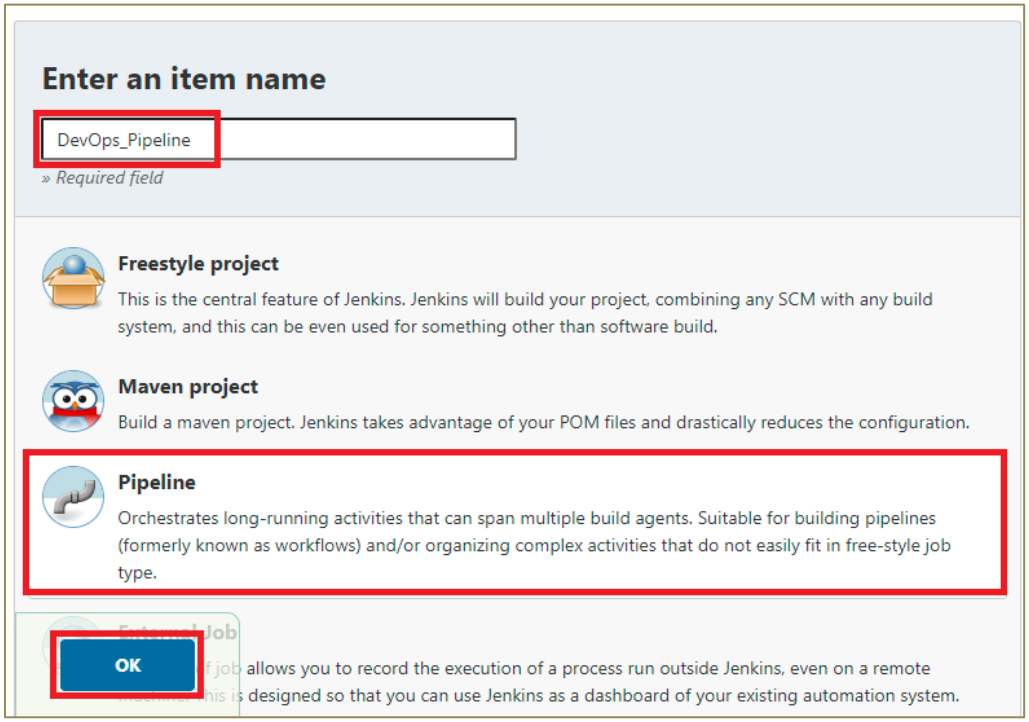
You should have completed the Practice of Lesson 5-1.

Tasks

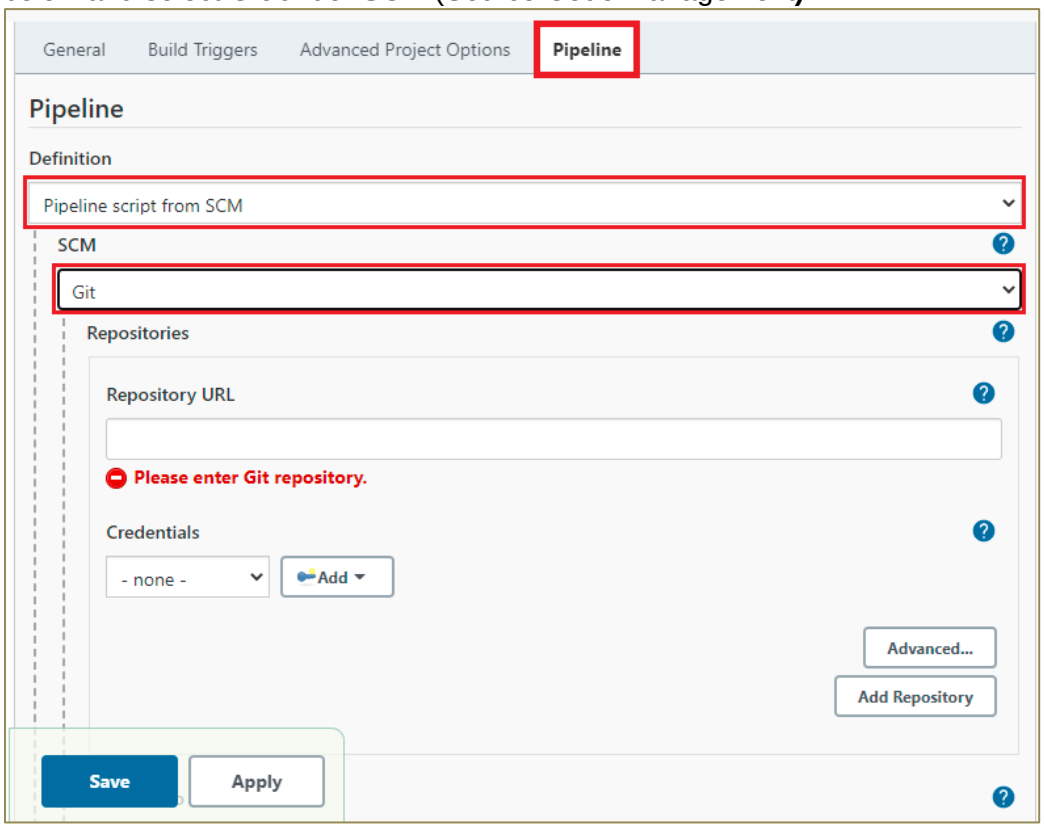
1. Create a DevOps Pipeline Job on Jenkins.
   a. In the Jenkins Dashboard, navigate to main menu and select **New Item** to create a Parallel Agent Pipeline as shown below.



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

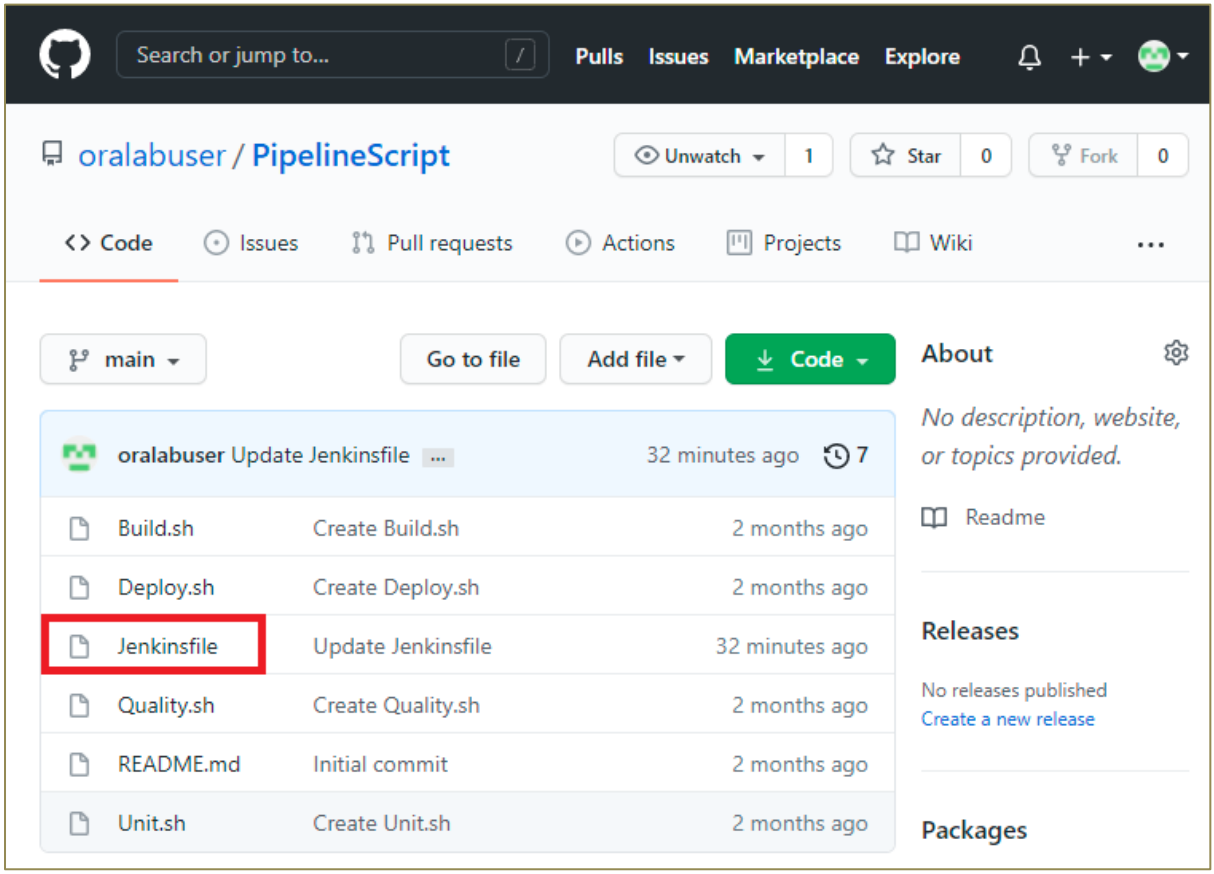


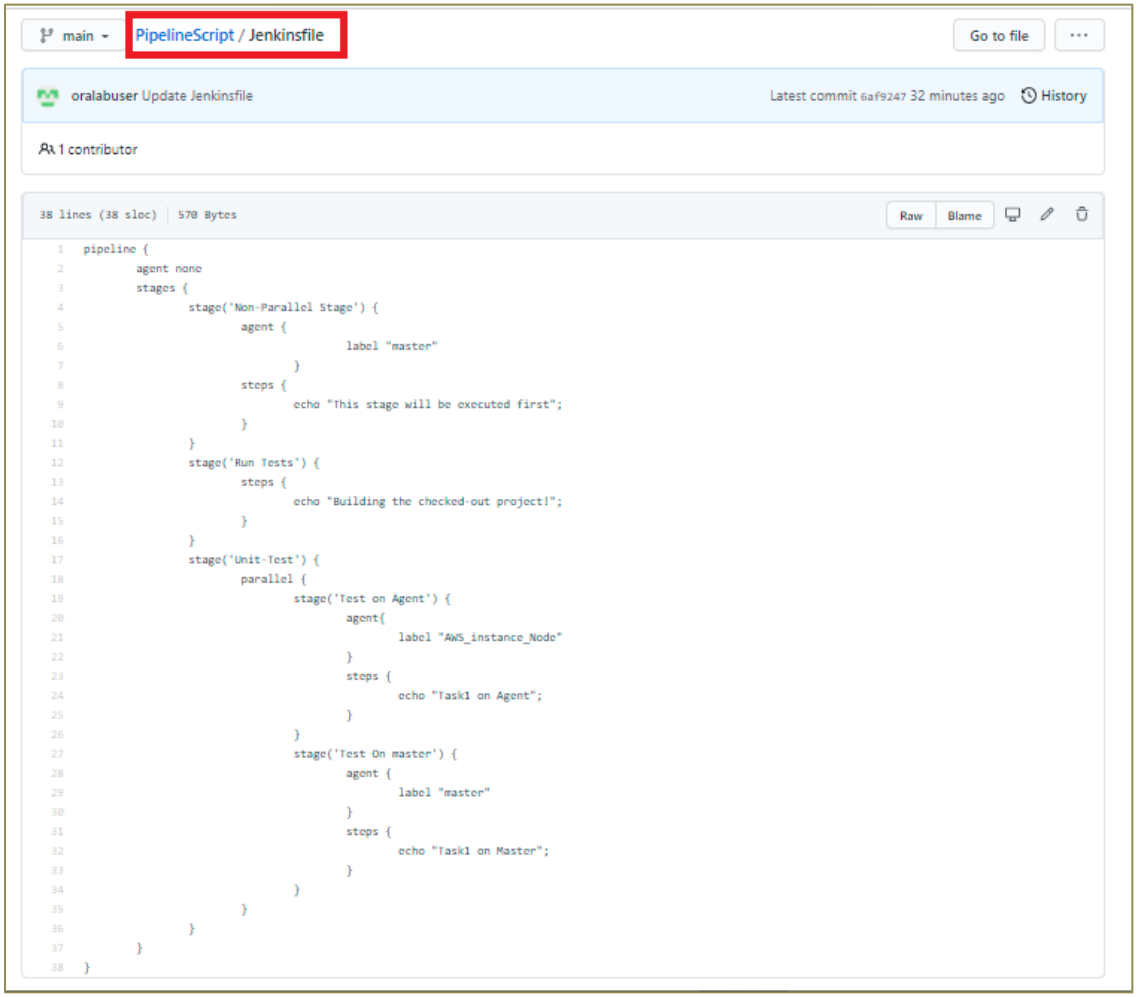
c. Navigate to **Pipeline**, select **Pipeline script from SCM** under **Definition** as shown below and select **Git** under **SCM** (Source Code Management**)**.



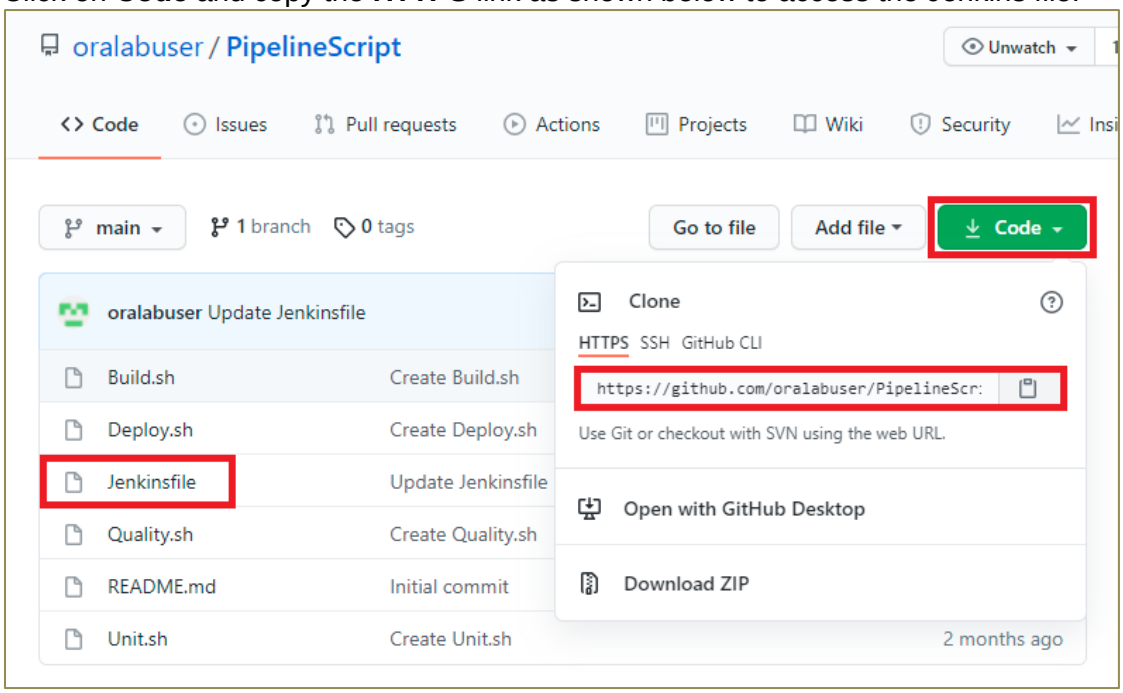
d. The GitHub link is provided below which is in the public domain, consisting of the Jenkins file.

Link: [oralabuser/PipelineScript (github.com)]

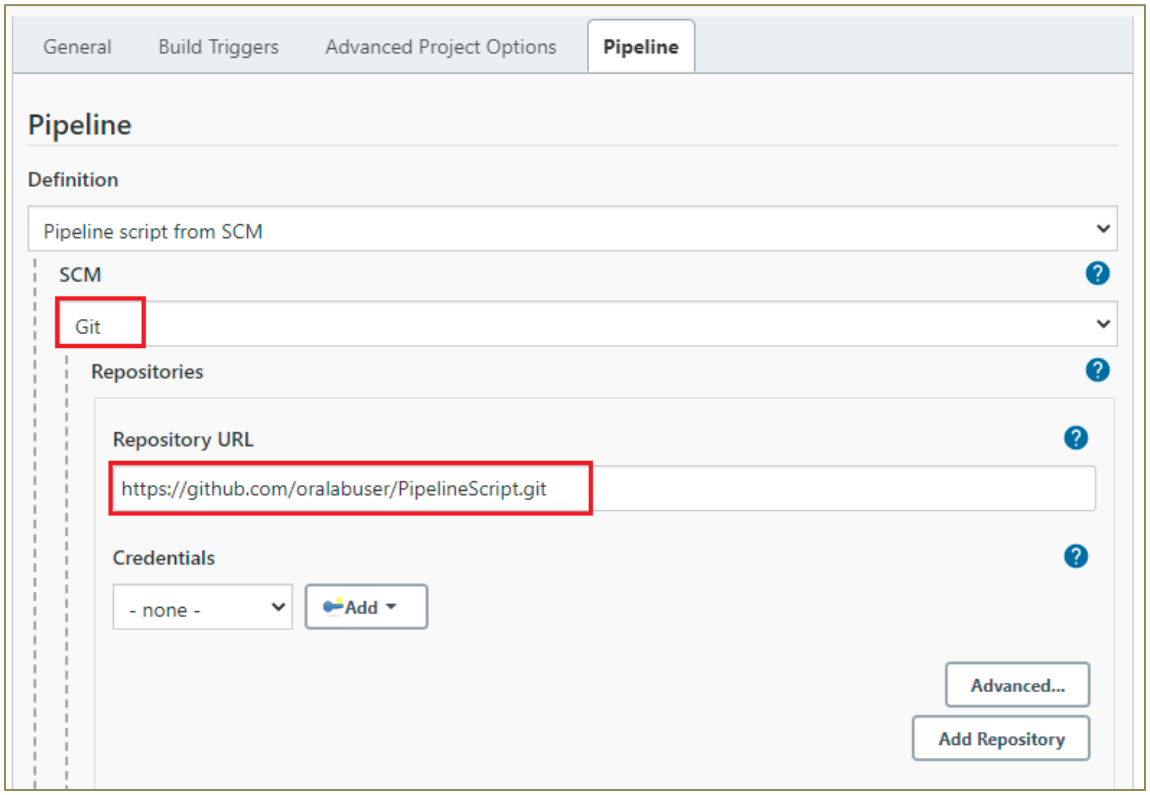




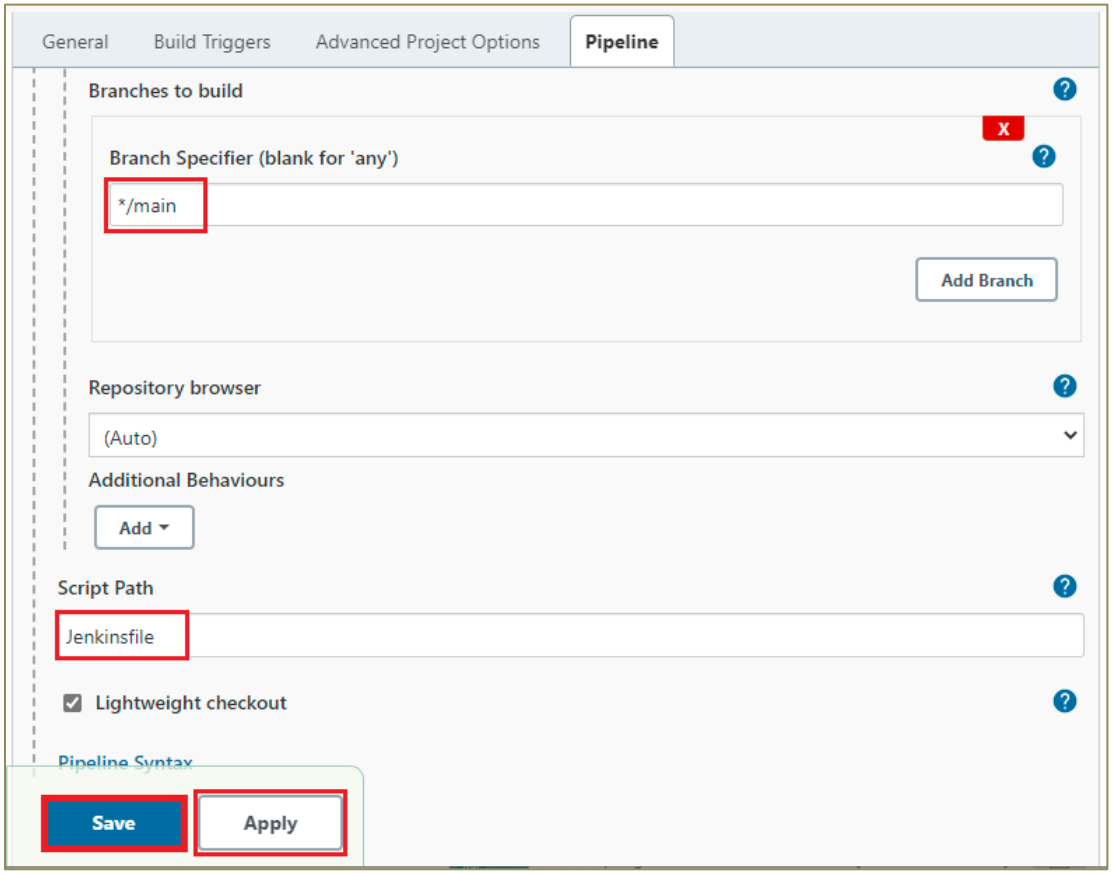
e. Click on **Code** and copy the **HTTPS** link as shown below to access the Jenkins file.



f. Paste the HTTPS GitHub link copied in the **Repository URL** as shown below.



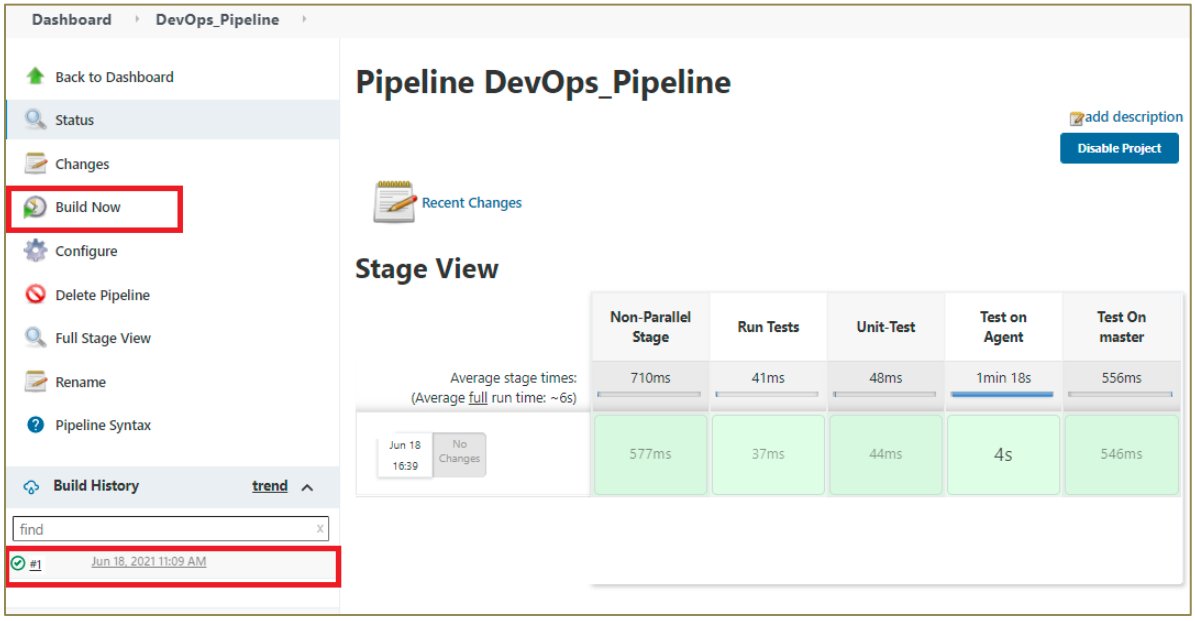
g. Provide the **Branch** as **main** and in **Script Path** provide the File name to access the code as shown below. Click **Apply** and **Save**.



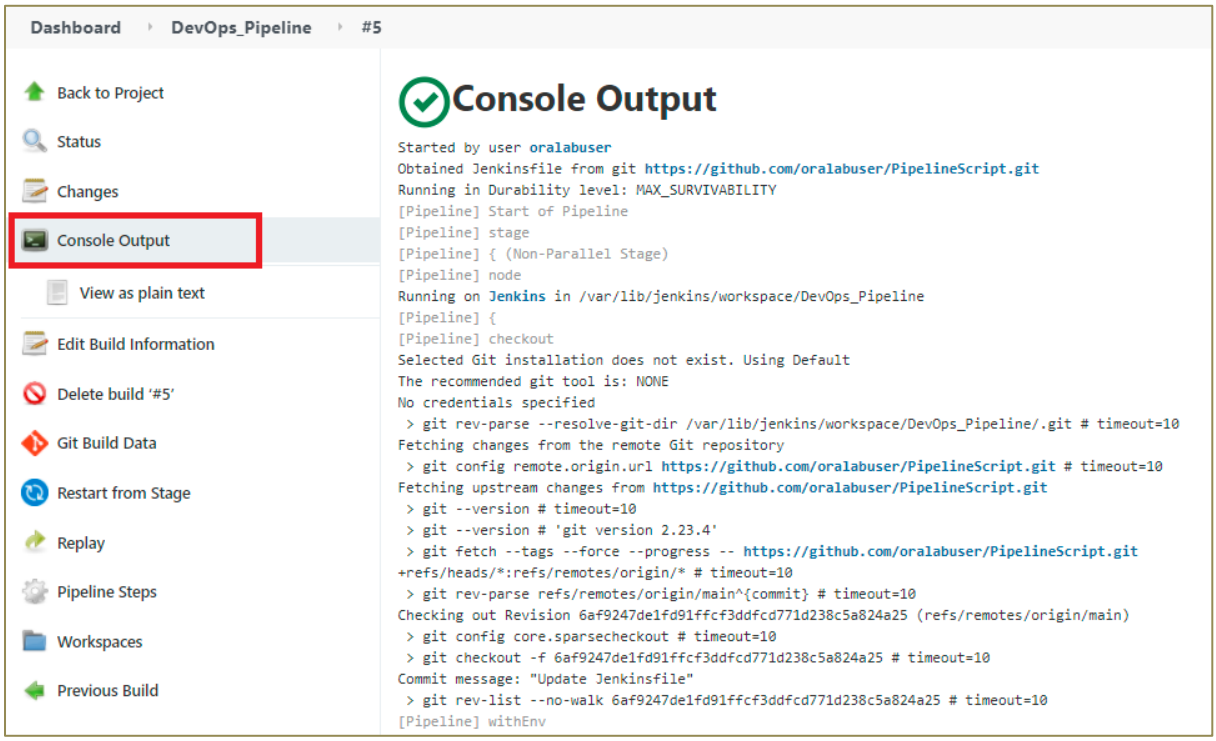
h. Connect to **Agent node** server and install **Git** in it as shown below.

```
[ec2-user@ip-172-31-36-198 ~]$ sudo yum install -y git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                               | 3.7 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package git.x86_64 0:2.23.4-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.23.4-1.amzn2.0.1 for package: git-2.23.
-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.23.4-1.amzn2.0.1 for package: git-2
23.4-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.23.4-1.amzn2.0.1 for package: git-2.23.
-1.amzn2.0.1.x86_64
```

i. Select **Build Now** from the menu to execute the Pipeline and **view** the **stages** of **DevOps_pipeline** getting executed and click on the link under **Build History** as shown below.



j. Click on **Console Output** to view the execution of the **DevOps_Pipeline** Job as shown below.



```
                 [Pipeline] {
                 [Pipeline] echo
[Test On master] Task1 on Master
                 [Pipeline] }
                 [Pipeline] // withEnv
                 [Pipeline] }
                 [Pipeline] // node
                 [Pipeline] }
                 [Pipeline] // stage
                 [Pipeline] }
 [Test on Agent] Avoid second fetch
                 Checking out Revision 6af9247de1fd91ffcf3ddfcd771d238c5a824a25 (refs/remotes/origin/main)
                 Cloning repository https://github.com/oralabuser/PipelineScript.git
                  > git init /home/ec2-user/jenkins/workspace/DevOps_Pipeline # timeout=10
                 Fetching upstream changes from https://github.com/oralabuser/PipelineScript.git
                  > git --version # timeout=10
                  > git --version # 'git version 2.23.4'
                  > git fetch --tags --force --progress -- https://github.com/oralabuser/PipelineScript.git +refs/heads/*:refs/remotes/origin/* #
                 timeout=10
                  > git config remote.origin.url https://github.com/oralabuser/PipelineScript.git # timeout=10
                  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
                  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
                  > git config core.sparsecheckout # timeout=10
                  > git checkout -f 6af9247de1fd91ffcf3ddfcd771d238c5a824a25 # timeout=10
                 Commit message: "Update Jenkinsfile"
                 [Pipeline] withEnv
                 [Pipeline] {
                 [Pipeline] echo
 [Test on Agent] Task1 on Agent
                 [Pipeline] }
                 [Pipeline] // withEnv
                 [Pipeline] }
                 [Pipeline] // node
                 [Pipeline] }
                 [Pipeline] // stage
                 [Pipeline] }
                 [Pipeline] // parallel
                 [Pipeline] }
                 [Pipeline] // stage
                 [Pipeline] End of Pipeline
                 Finished: SUCCESS
```

2. Close the terminal, Logout from the AWS Management console and Jenkins Dashboard.