Practice for Lesson 6: Blue Ocean (Advanced Jenkins)

Practices for Lesson 6

Overview

In these practices, you will learn how to install the Blue Ocean plugin in Jenkins and further create a Pipeline in Jenkins using Blue Ocean.

.

Practice 6-1: Create Pipeline using Blue Ocean

Overview

In this practice, you will learn how to create Pipeline in Jenkins using BlueOcean.

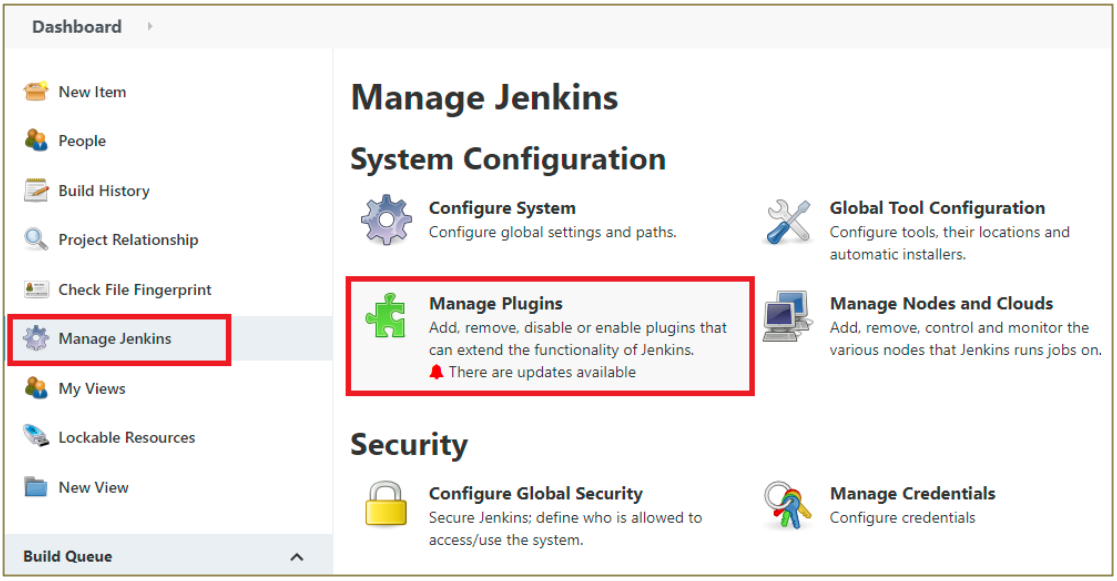
Assumptions

You should have completed the Practice of Lesson 5.

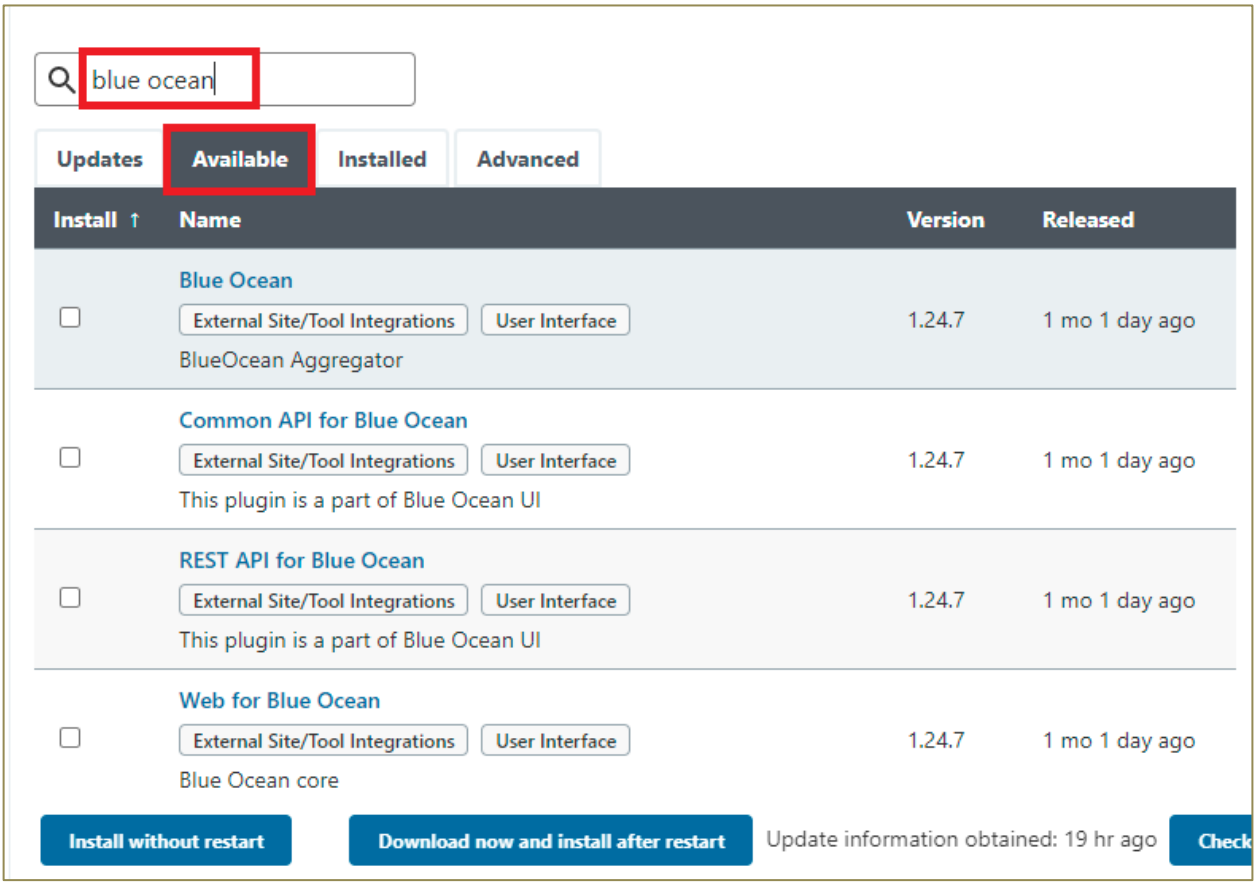
Tasks

1. Sign in to **Jenkins Instance Dashboard**.
   a. In a browser on your local machine, enter the **Public IP** address of the EC2 instance followed by the IP address to sign in to the Jenkins Dashboard (for example, **<Public-IP>:8080**).

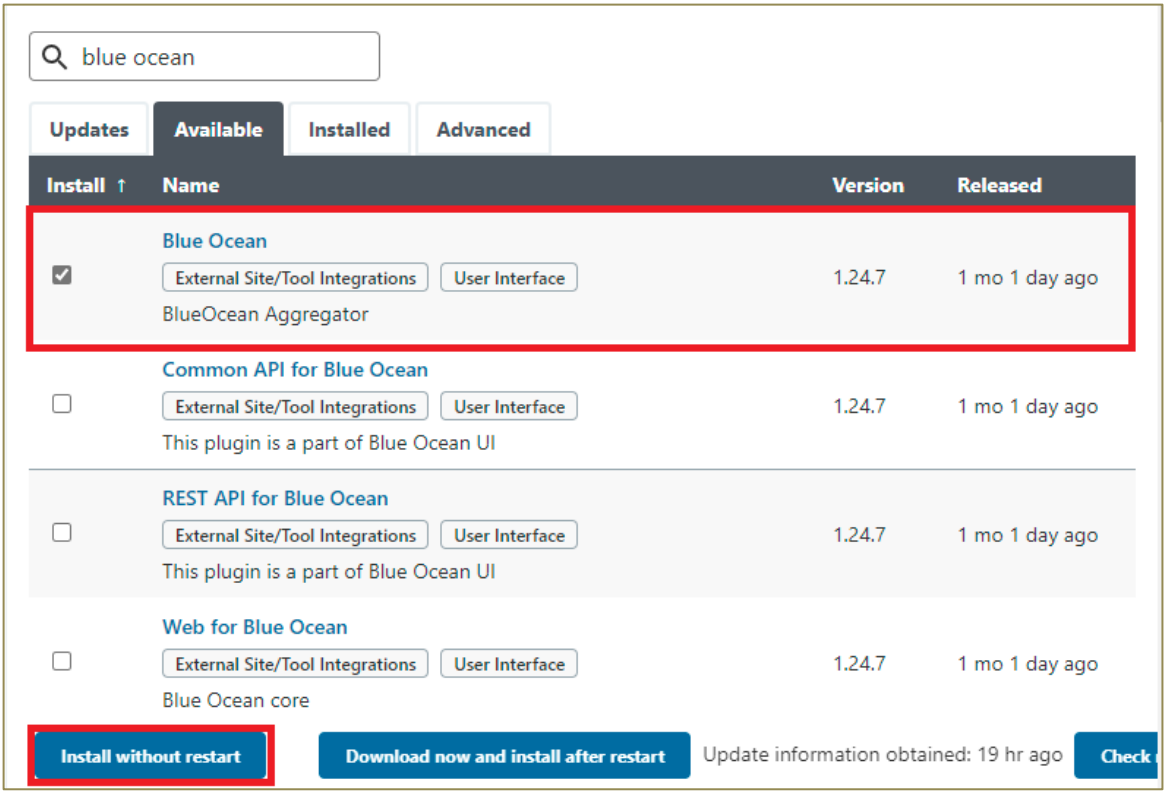
   b. Enter the user name and password provided.
   c. You will have access to the Jenkins Dashboard.
2. Install Blue Ocean on the **Linux instance**.
   a. Navigate to Dashboard, select **Manage Jenkins** and click **Manage Plugins** as shown below.



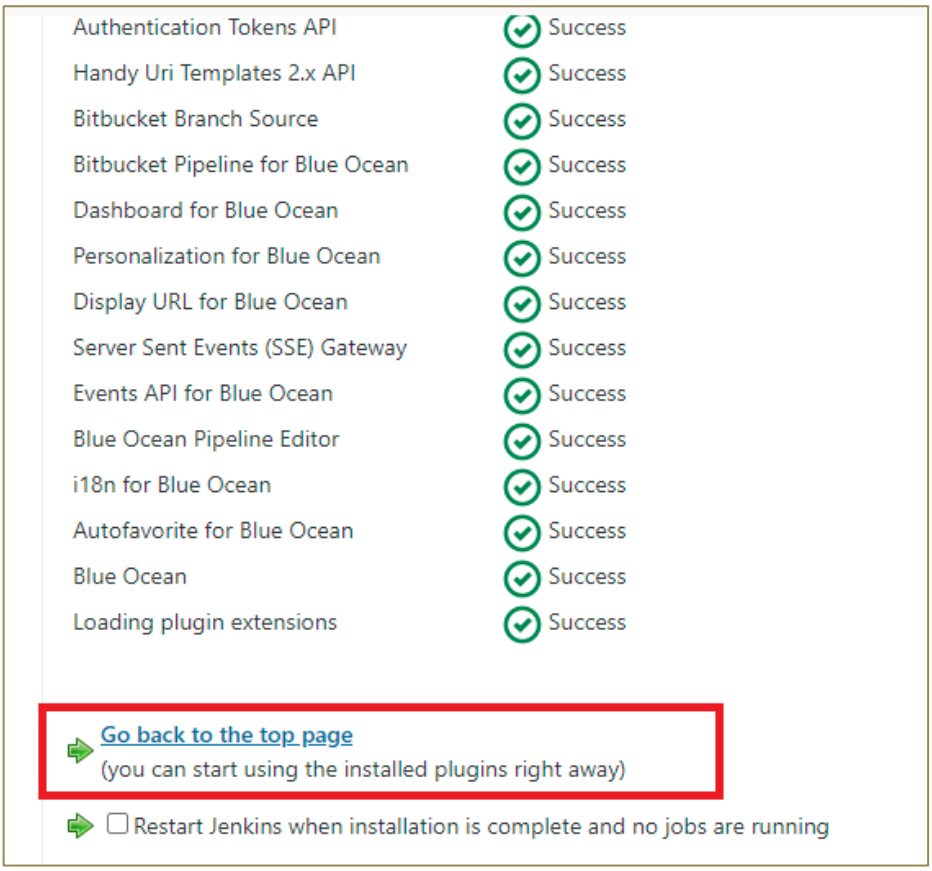
b. Select **Available** and search for **Blue Ocean**.



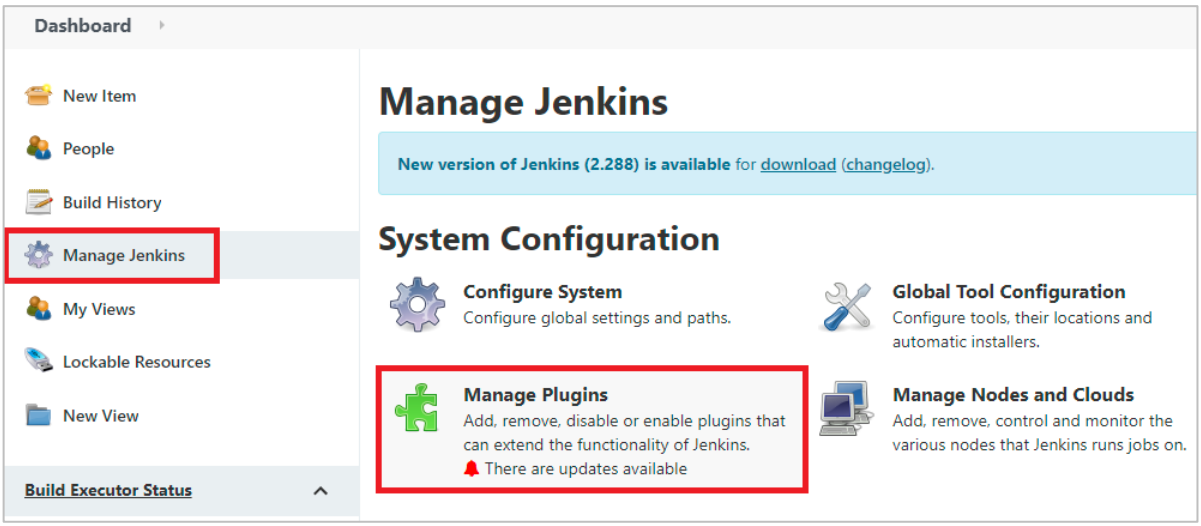
c. Select the check box of the **Blue Ocean** plugin and click **Install without restart**.



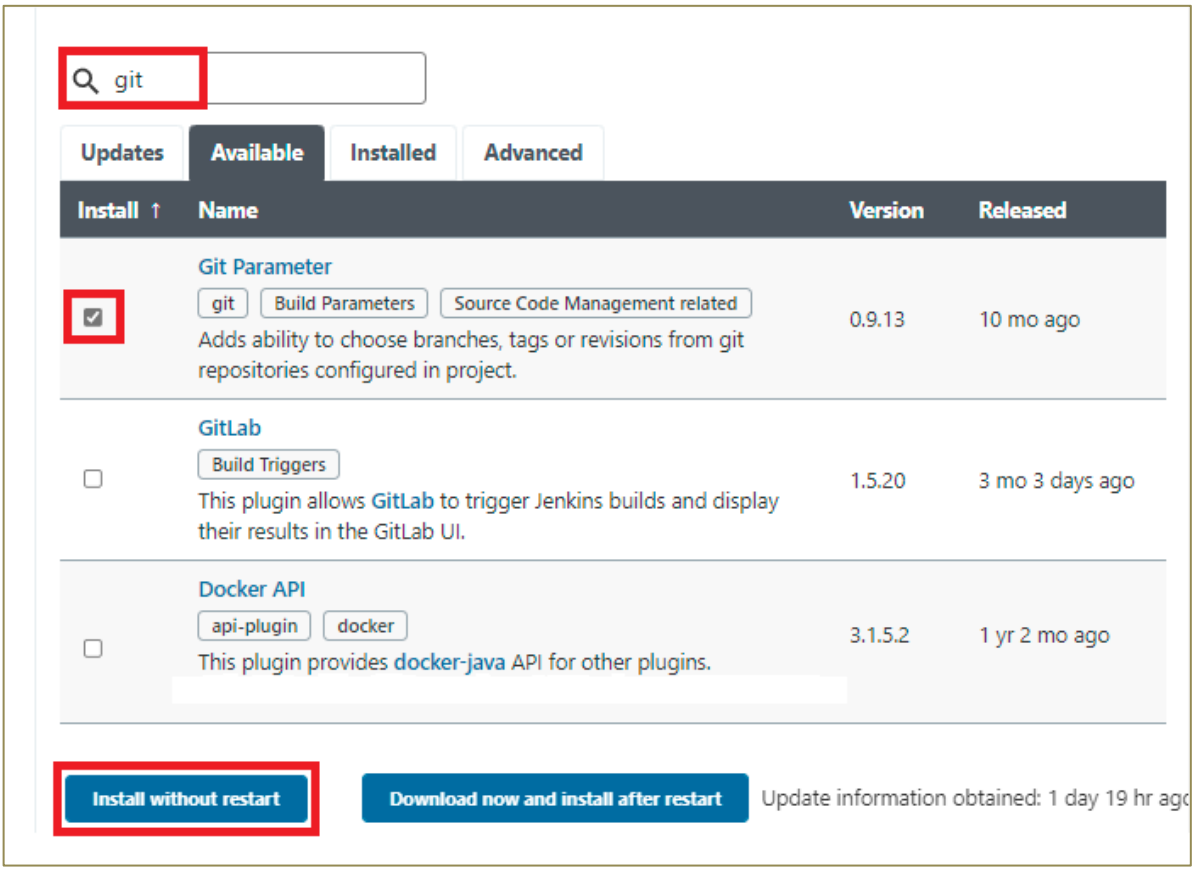
d. The installation process will proceed and Success message will be displayed as shown below.



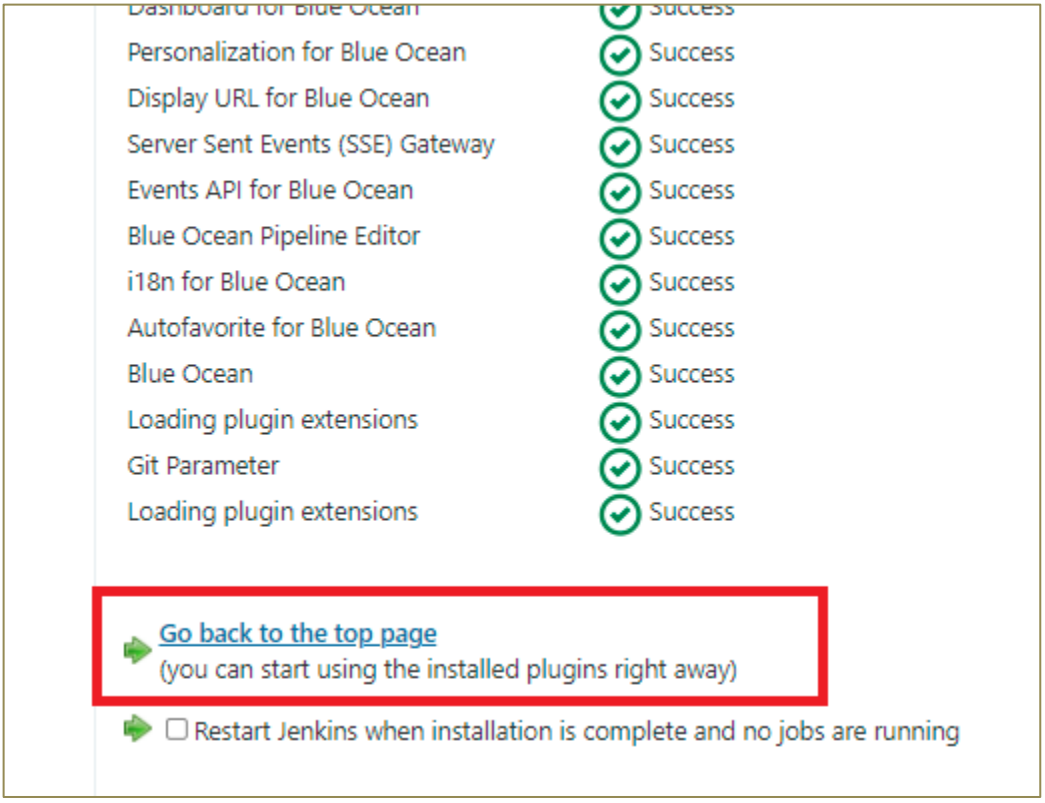
e. Similarly, install **Git Parameter** plugin in the Jenkins. Navigate to **Main menu** and select **Manage Plugins** as shown below.



f. Select **Available**, search for **Git** and select the check box of **Git** Parameter. Click **Install without restart**.

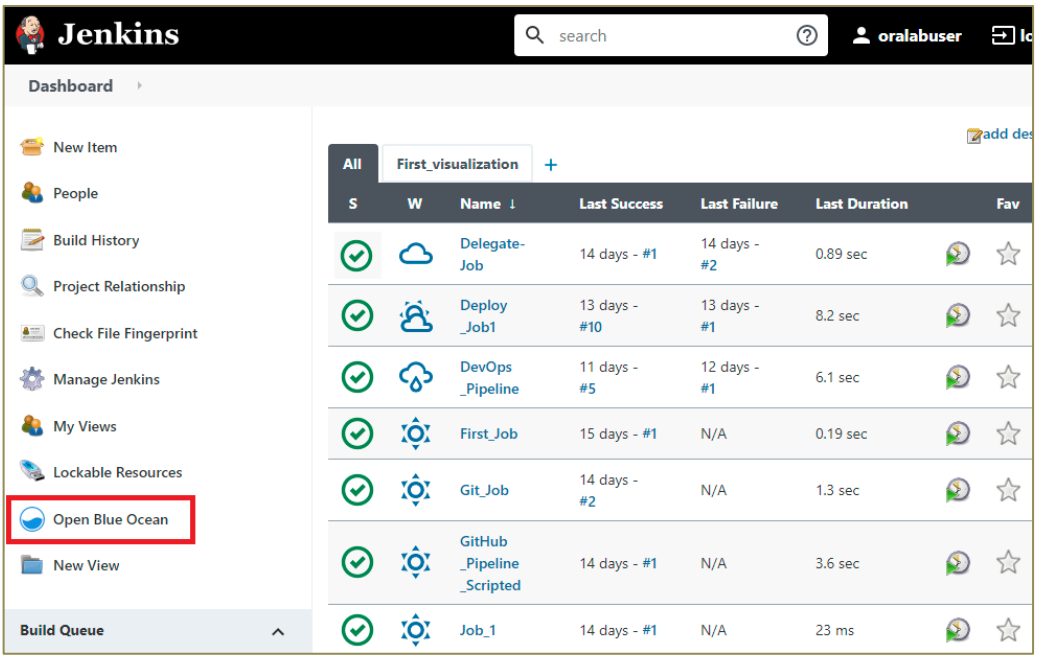


g. As shown below the installation of the plugin is successful, click **Go back to the top page**.

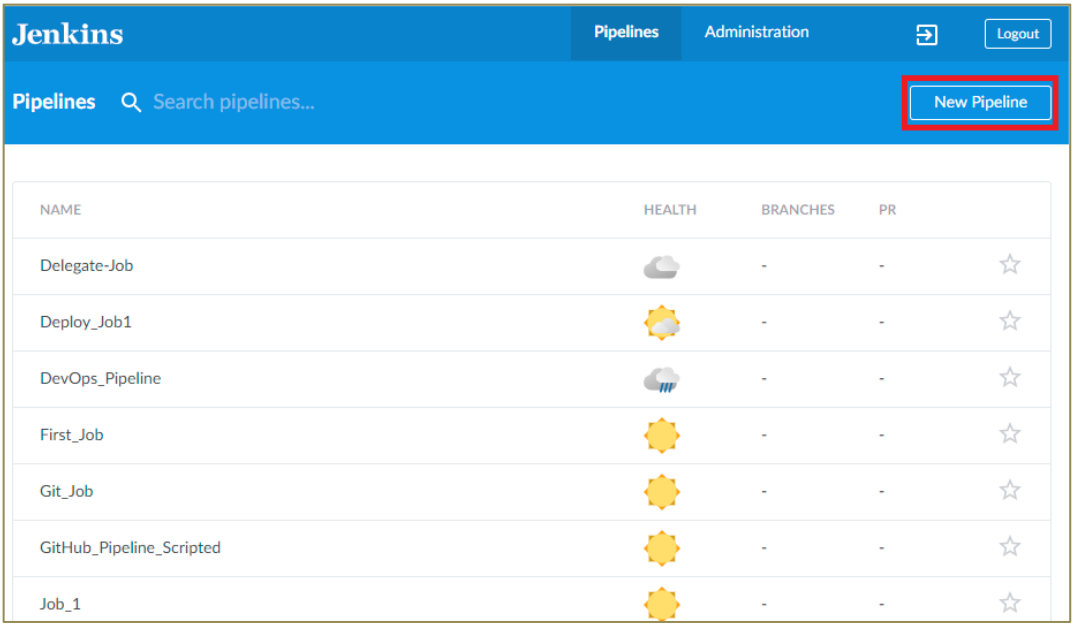


3. Create Jenkins Pipeline using **Blue Ocean**.

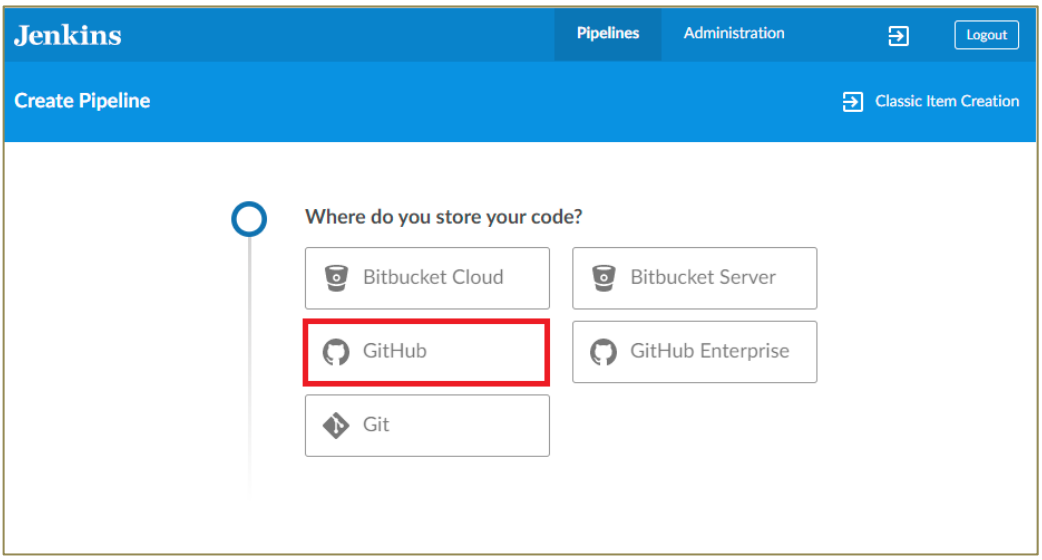
   a. In the Dashboard click **Open Blue Ocean** to create the pipeline as shown below.



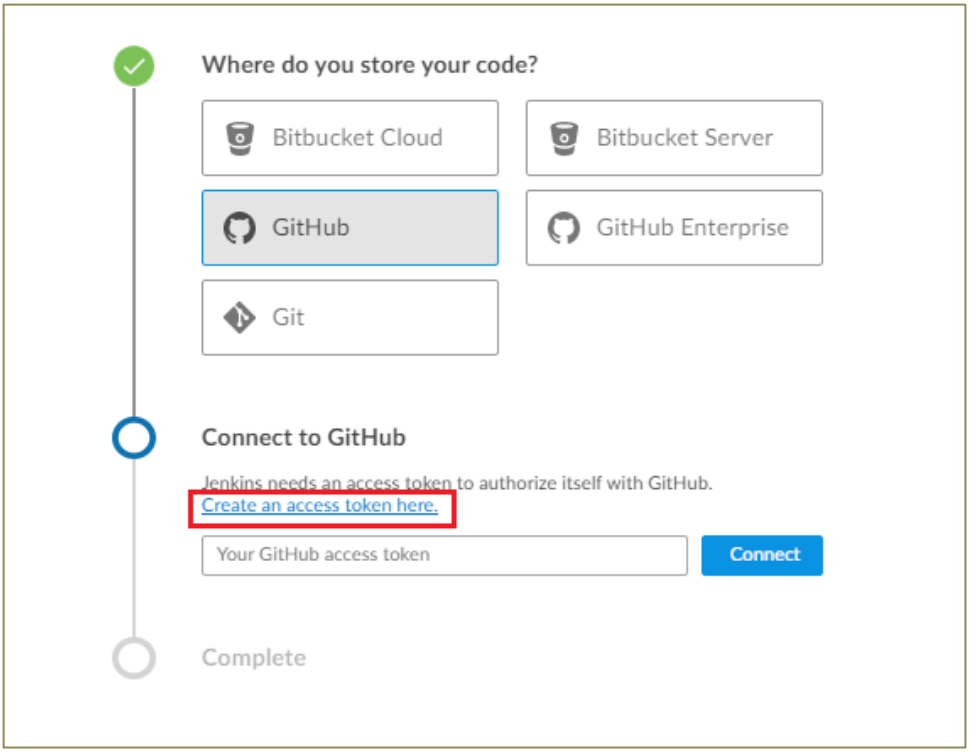
   b. View the Pipelines page and click on **New Pipeline** to create a pipeline in Jenkins**.**



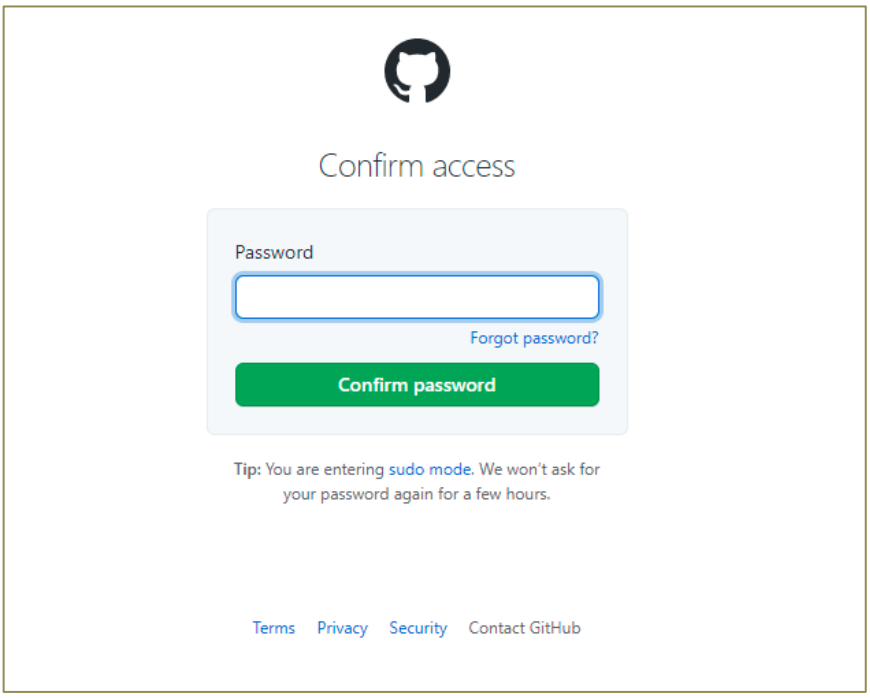
c. For "**Where do you store your code**", select **GitHub** as shown below.



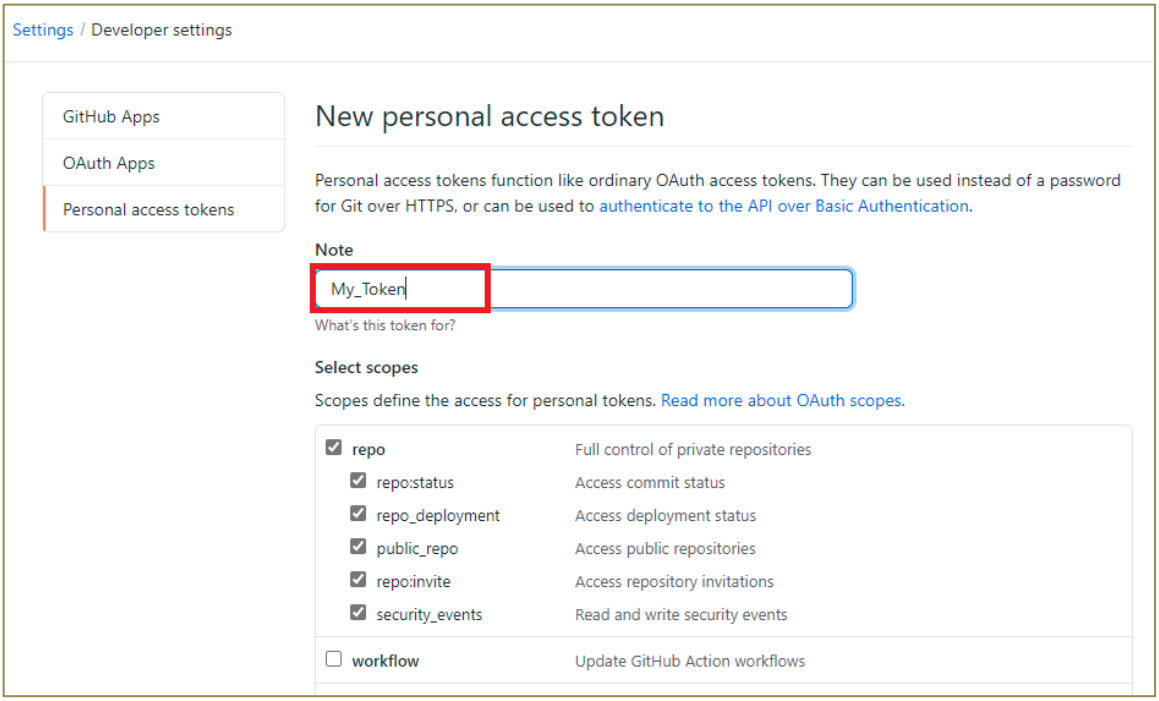
d. To Connect to GitHub, an access token is required. Click on **Create an access token here**.



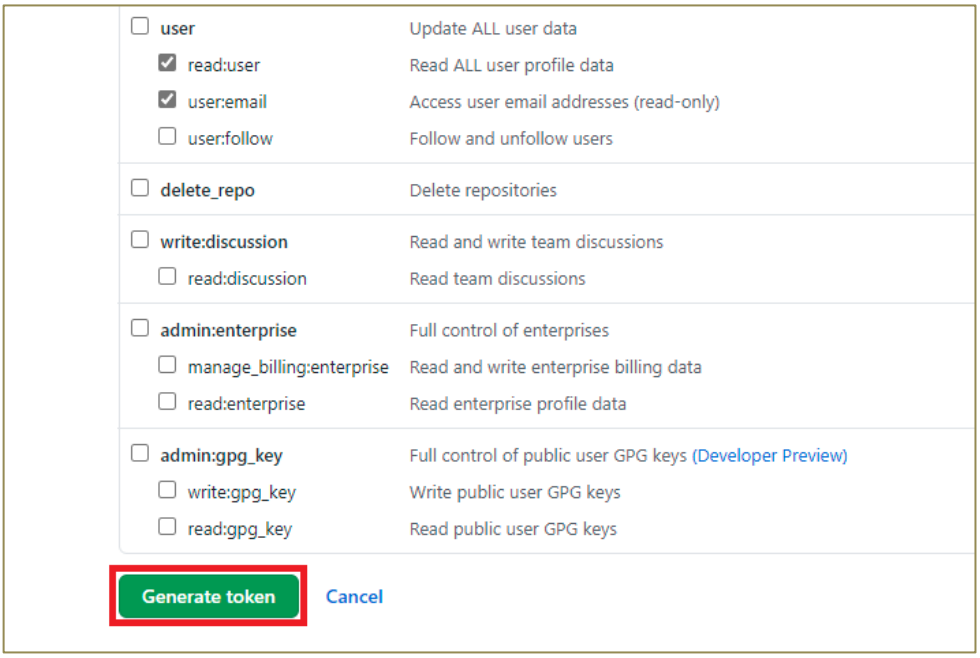
e. Login to the respective GitHub repository with the credentials as shown below.



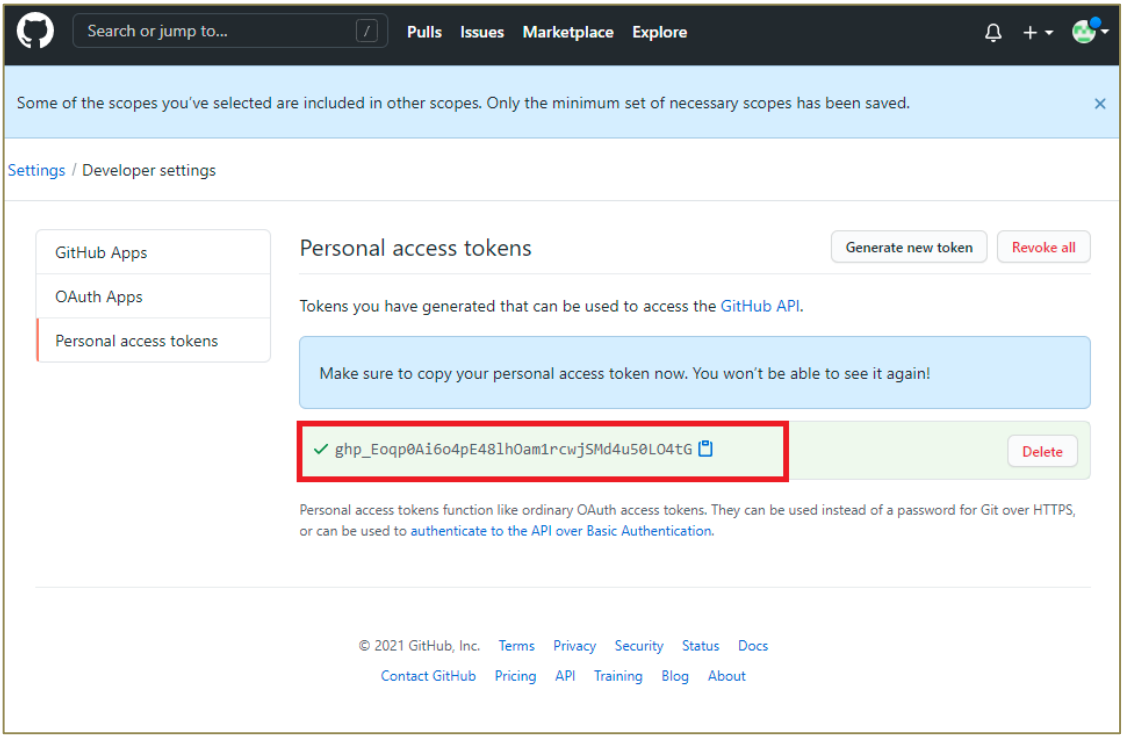
f. Provide the Name for the **Token** as shown below.



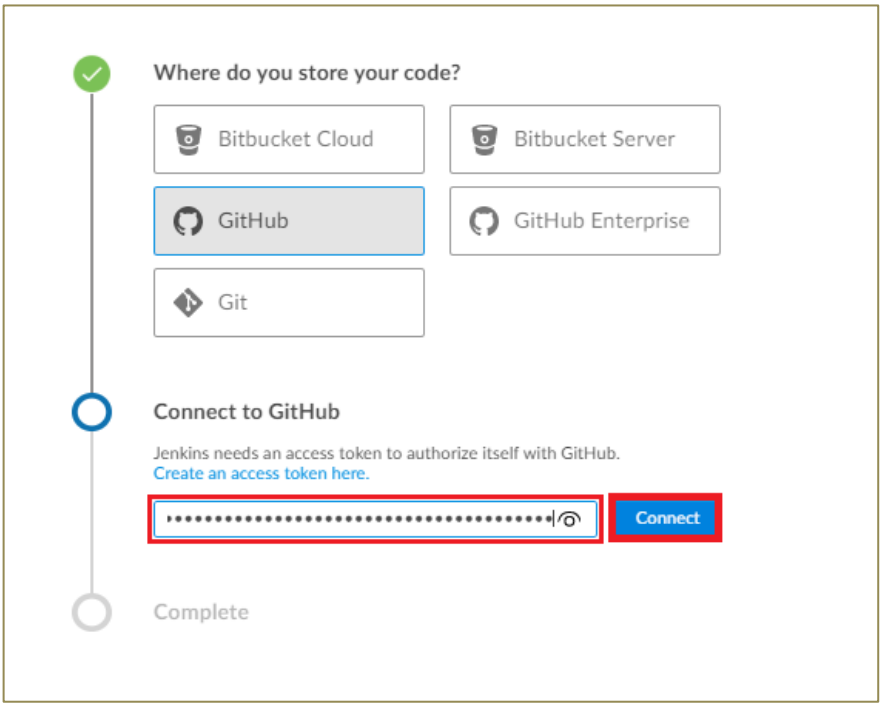
g. Accept the default values and click **Generate token** as shown below.



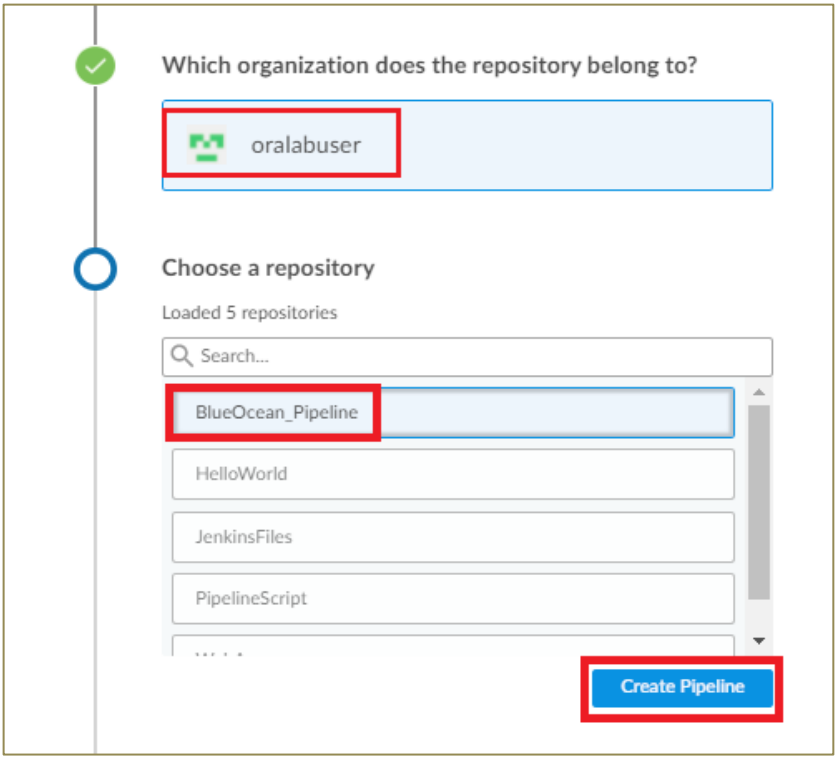
h. The Token is generated successfully, copy the **Token**.



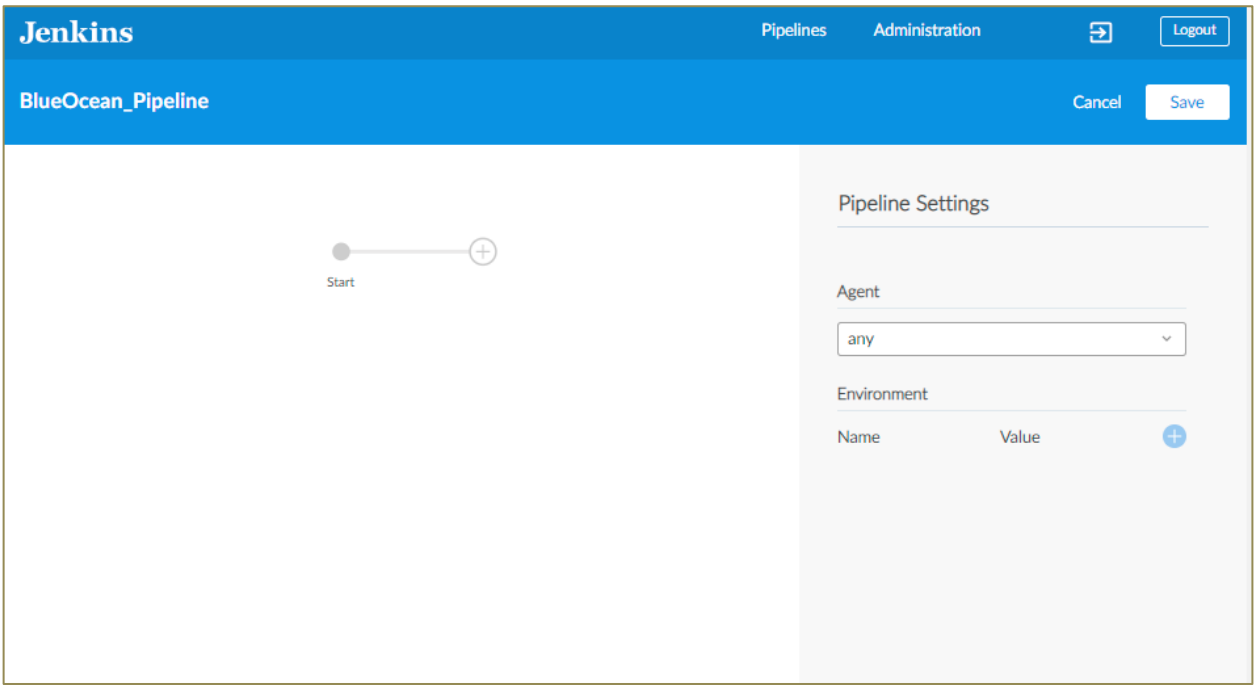
i. Paste the **Token** as shown below and click **Connect**.



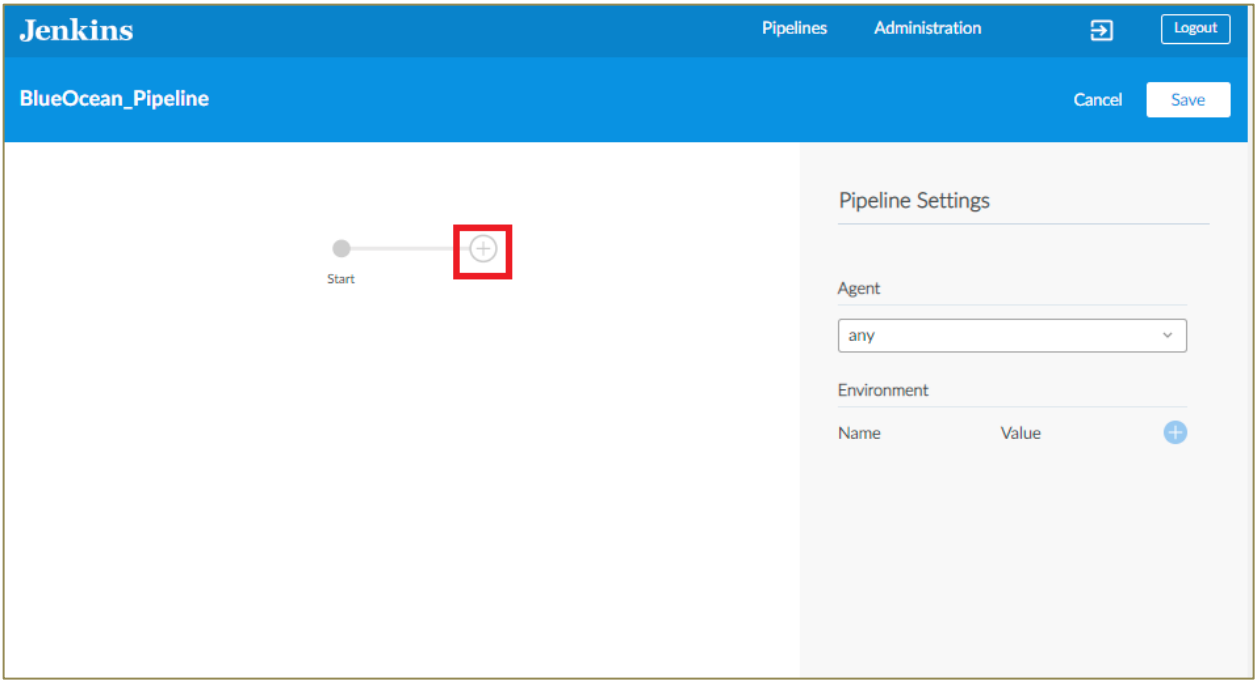
j. The **GitHub** account is linked to the **Jenkins**. Choose the repository from the GitHub account and click Create **Pipeline**.



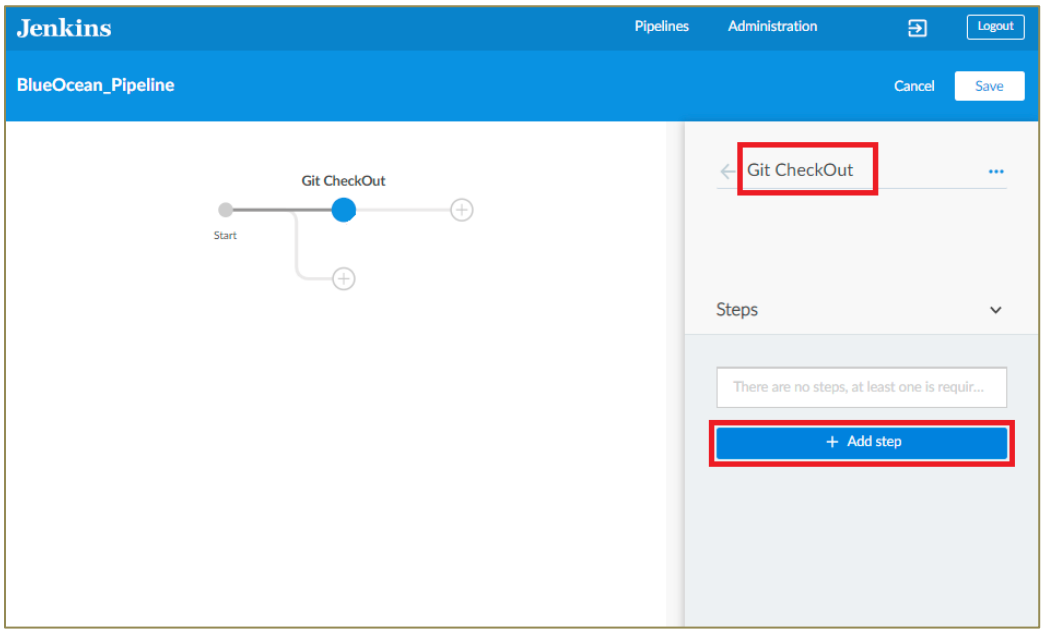
k. The Jenkins try to search for the **Jenkinsfile** in the GitHub repository, if not found it creates a new **Jenkinsfile**. As shown below is the Jenkins Pipeline UI.



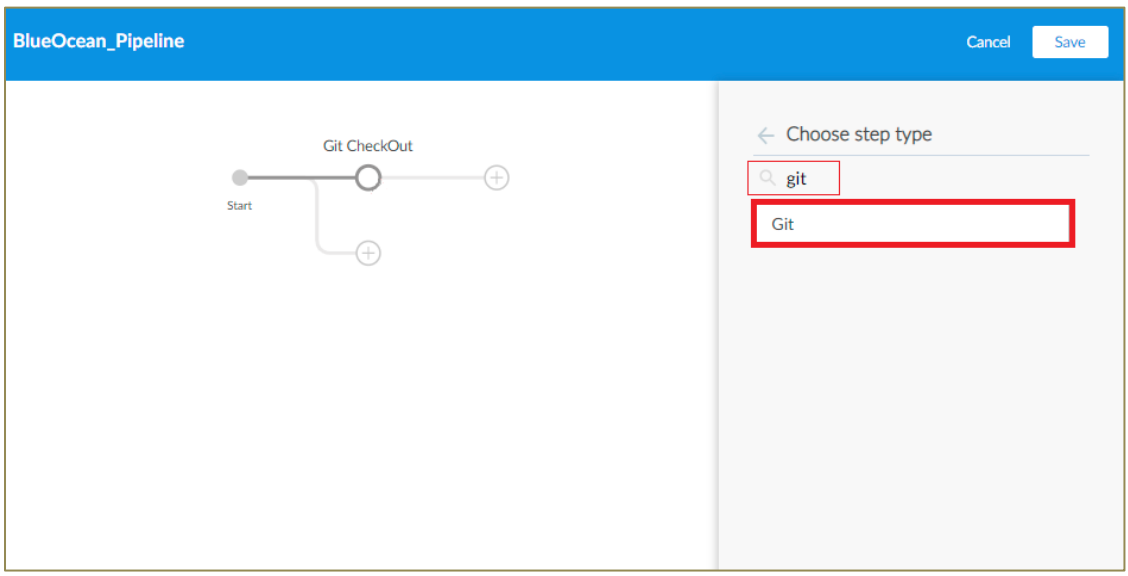
l. Select the + symbol as shown below to add the pipeline step.



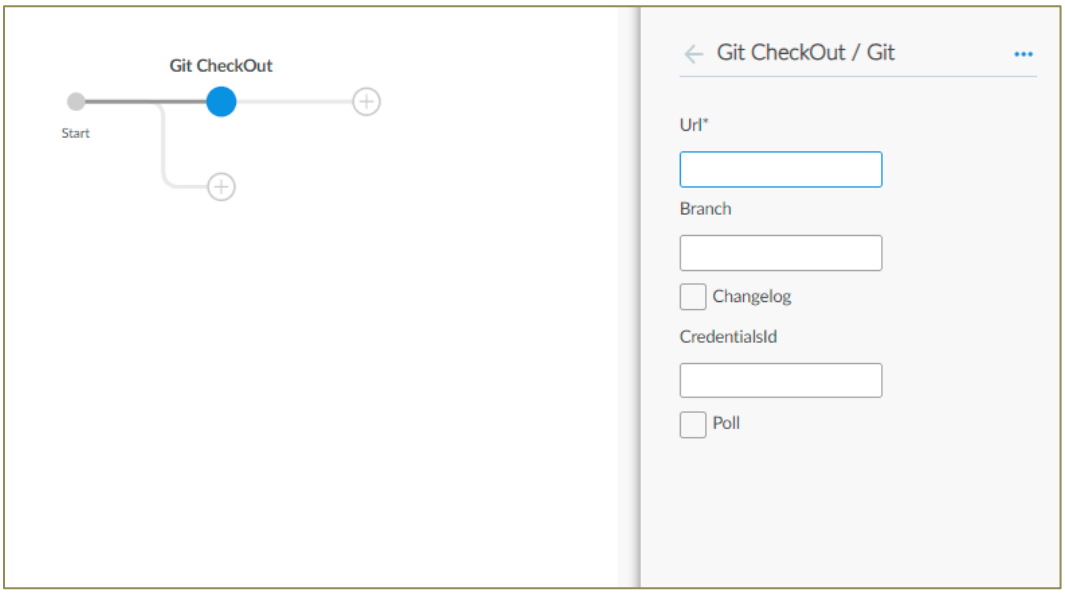
i) Provide the name for the Stage as shown below and click on **Add Step** to add steps to the Stage.



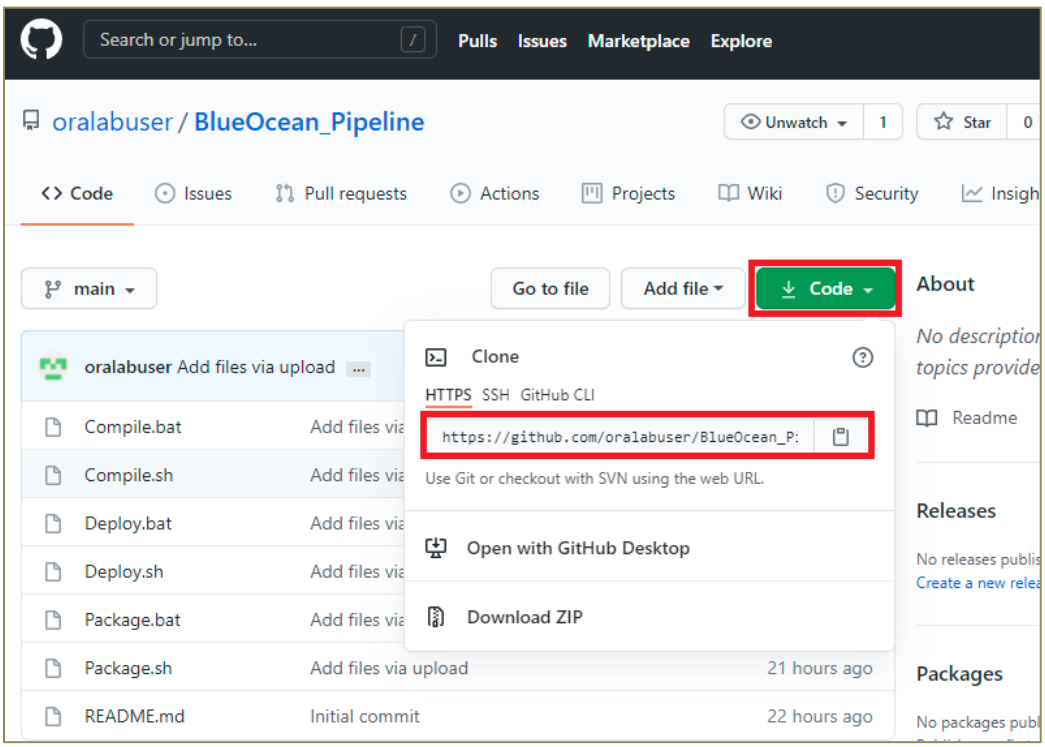
ii) Search for **Git** as shown below and select the **Git**.



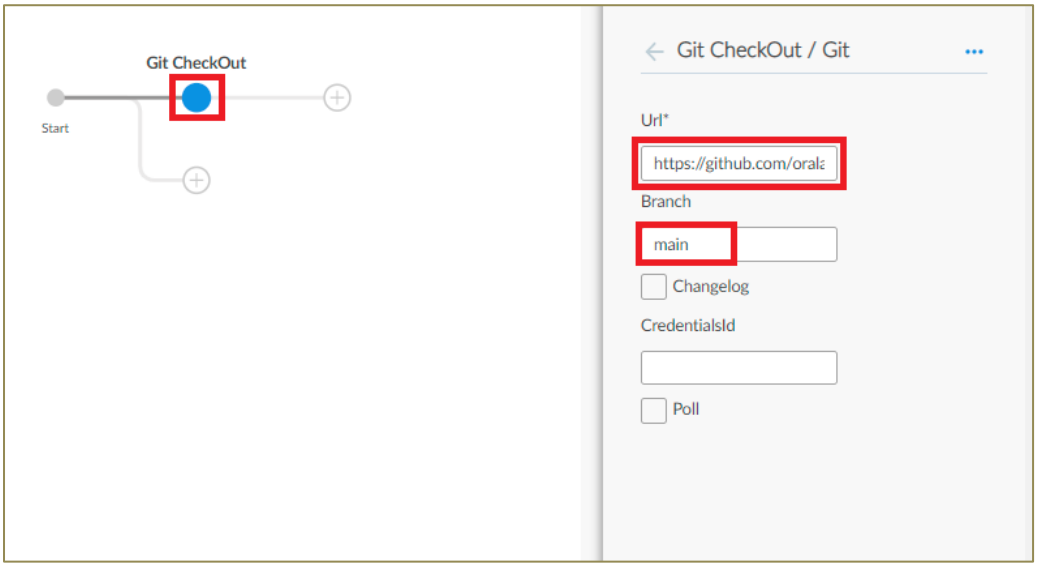
iii) Provide the **URL** and the Branch name for the respective GitHub repository.



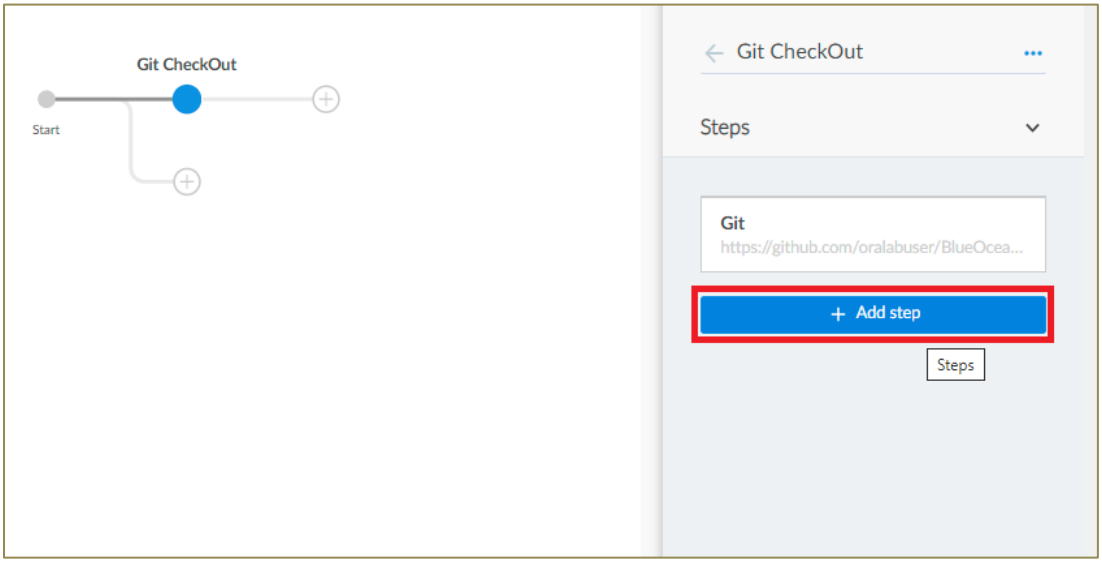
iv) Open the GitHub repository, click on **Code** and copy the **HTTPS** path as shown below.



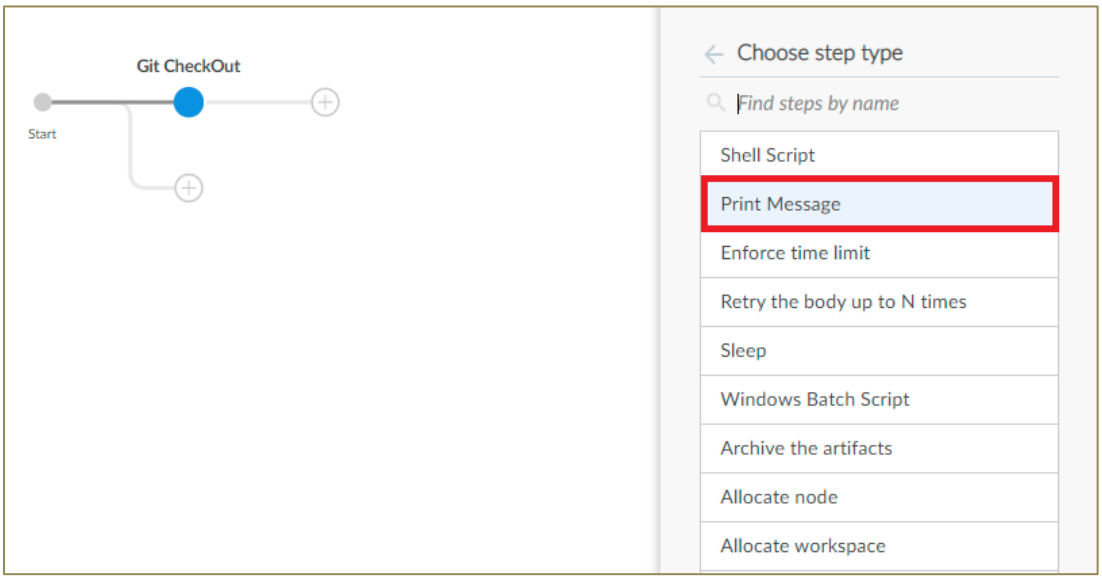
v) Navigate to Jenkins pipeline, paste the URL and provide **Branch** as **main** as shown below.



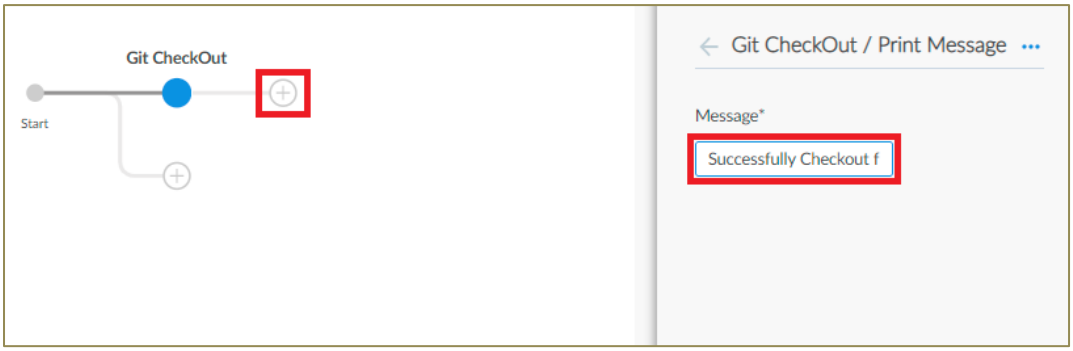
vi) Click **Add Step** to further add new steps to the Stage as shown below.



vii) Select **Print Message** to display the message while execution.

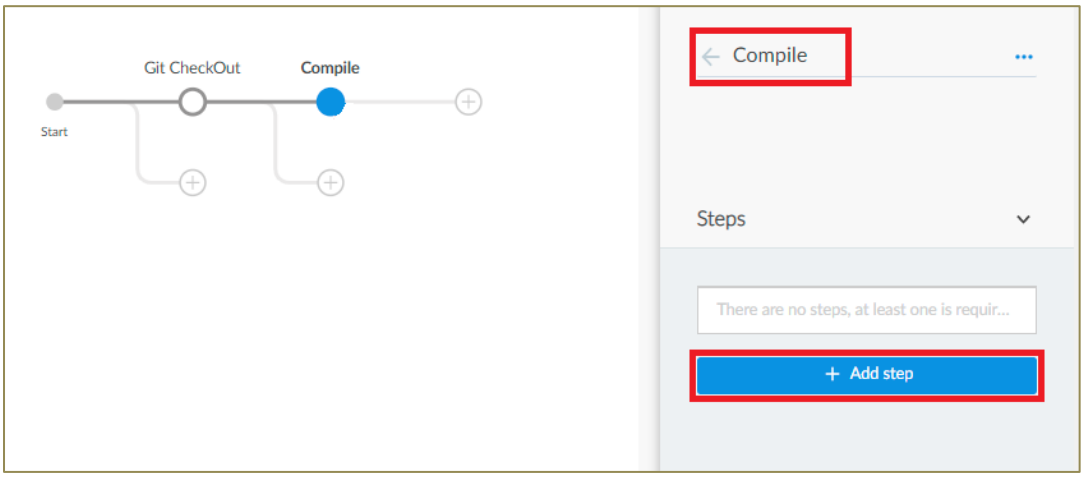


viii) Type **Successfully Checkout from GitHub!!** in the **Message** box as shown below.

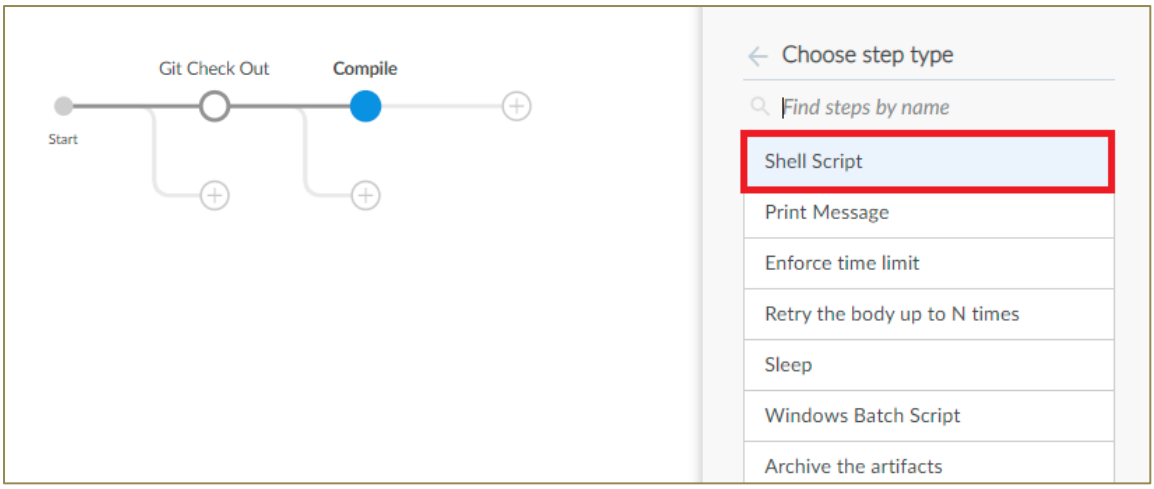


m. Add next Stage **Compile** in the Jenkins pipeline.

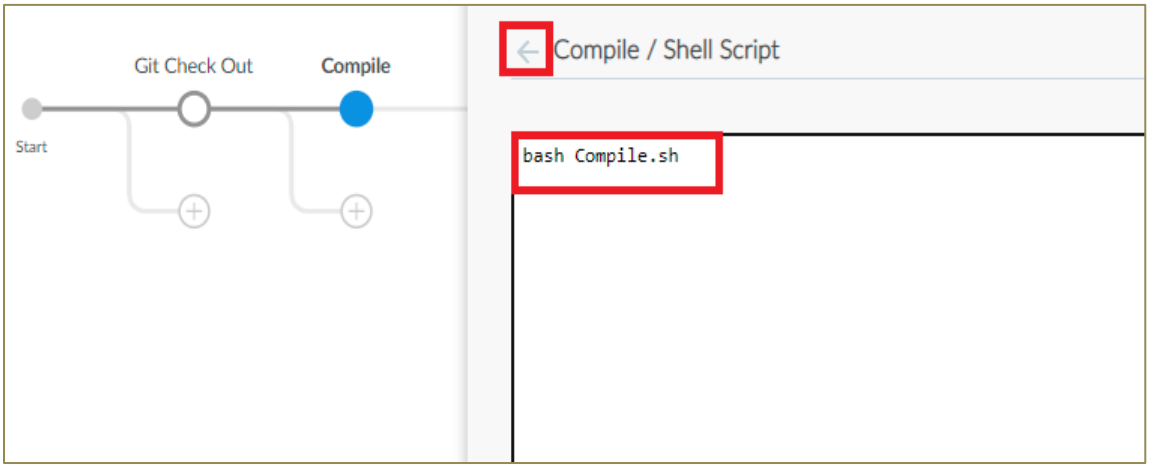
i) Provide the name for the Stage as shown below and click on **Add Step** to add steps to the Stage.



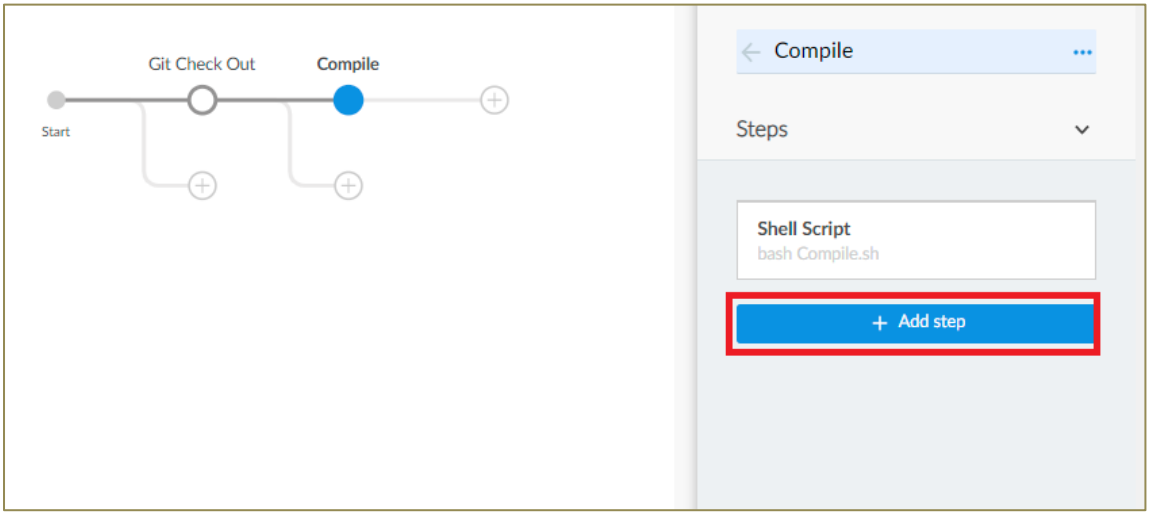
ii) Choose the step type as **Shell Script** as shown below.



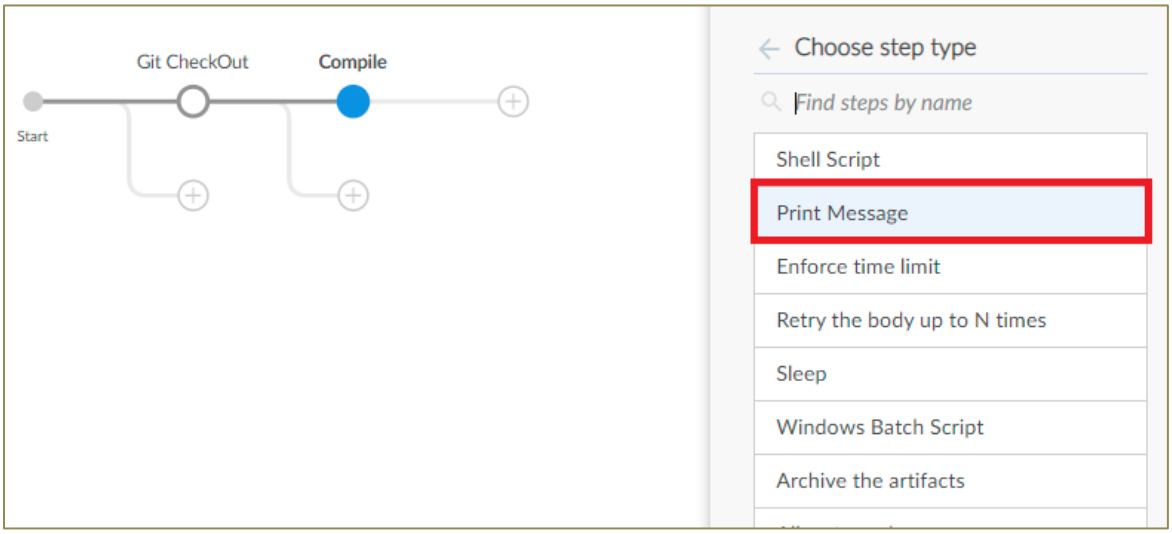
iii) Type the command in the **Shell Script** to execute the file.



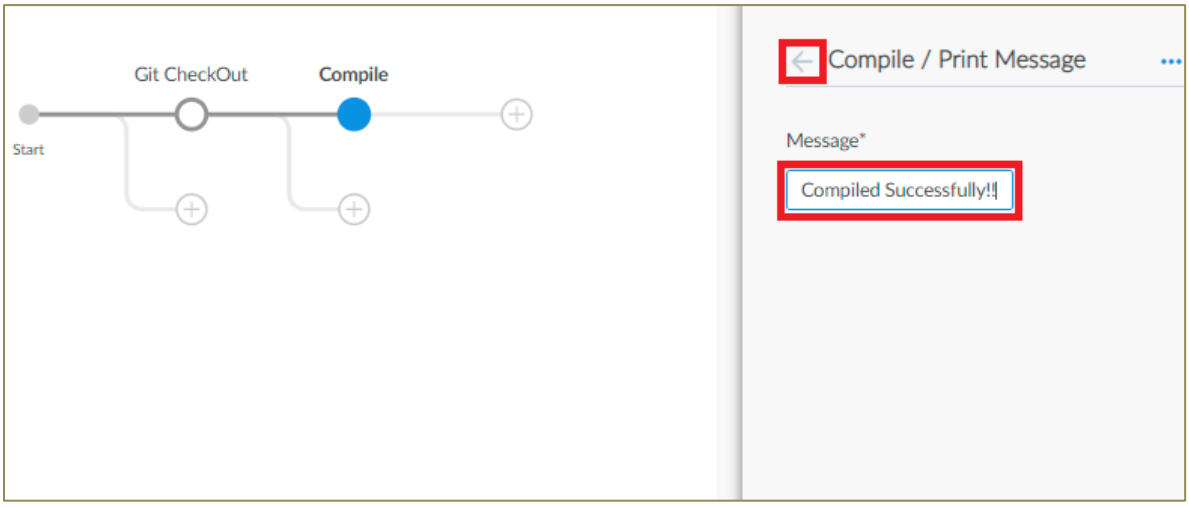
iv) Click **Add step** to further add the step to the Stage as shown below.



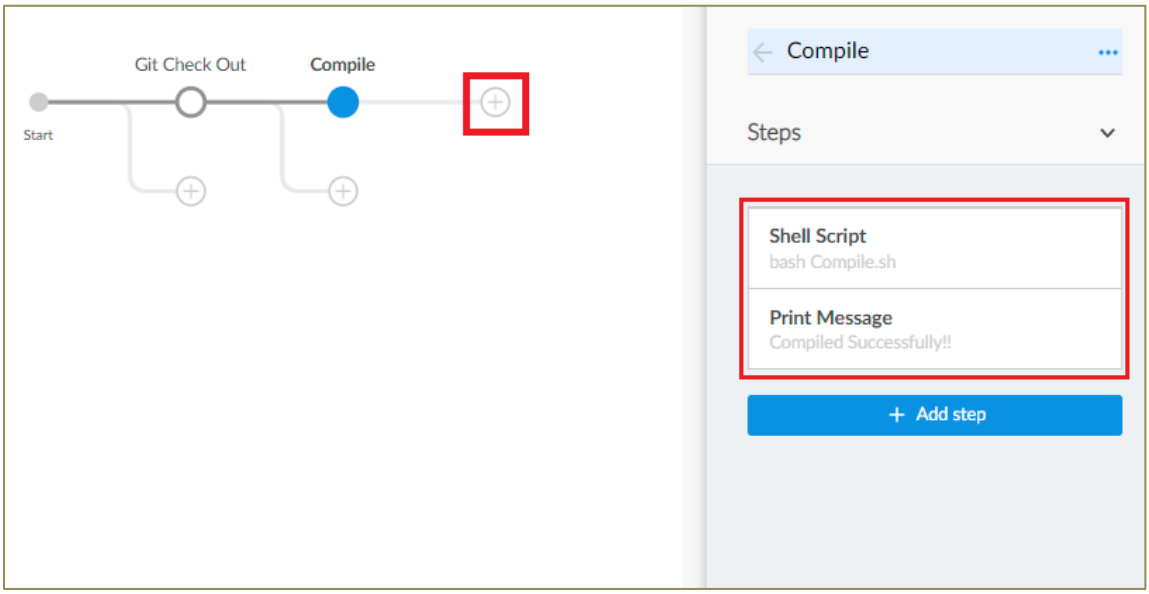
v) Select **Print Message** to display the message while execution.



vi) Type **Compiled Successfully!!** in the **Message** box as shown below.

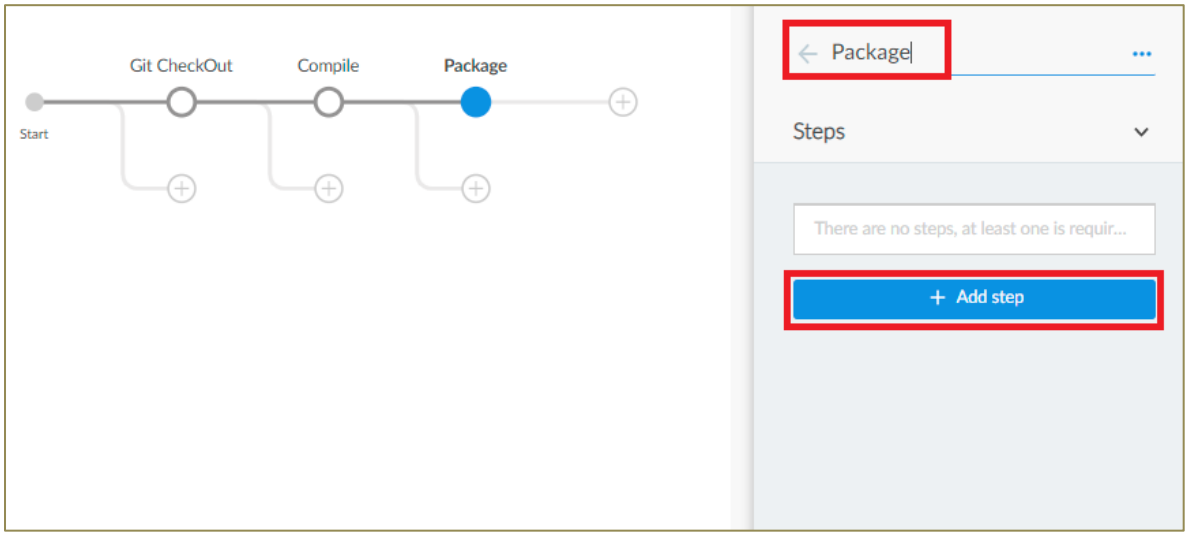


i) Verify the Steps added to the **Compile** stage and click on the + symbol to add the new stage.

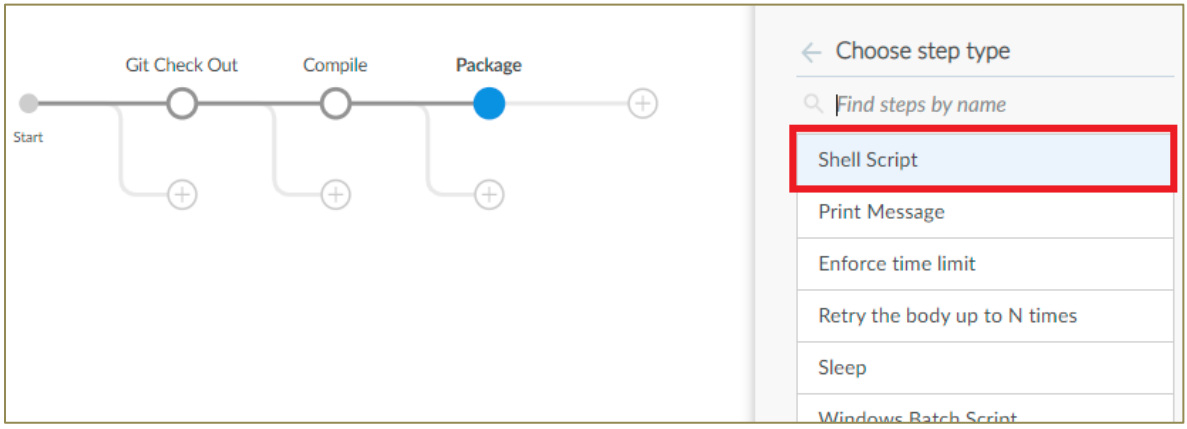


n. Add next Stage **Package** in the Jenkins pipeline.

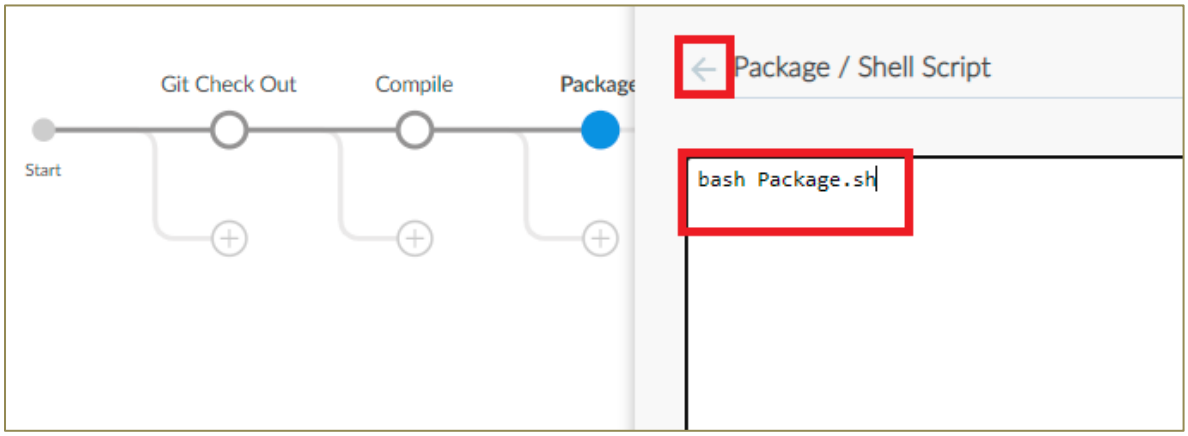
i) Provide the name for the Stage as shown below and click on **Add Step** to add steps to the Stage.



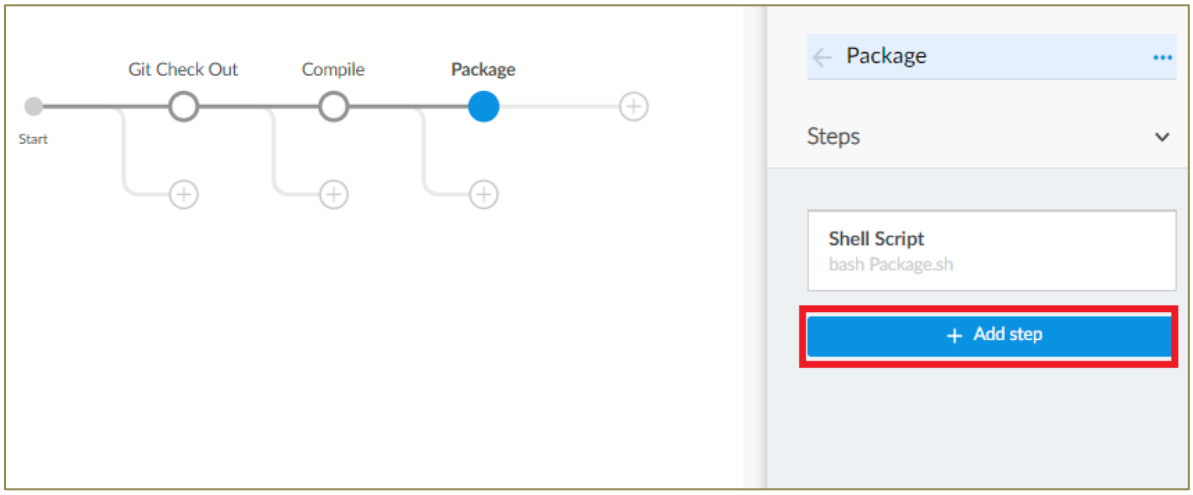
ii) Choose the step type as **Shell Script** as shown below.



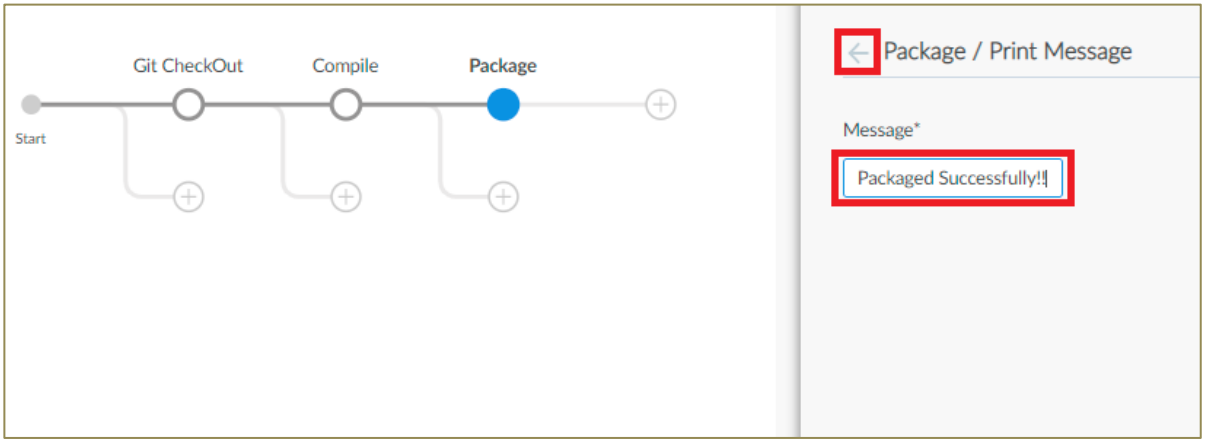
iii) Type the command in the **Shell Script** to execute the file.



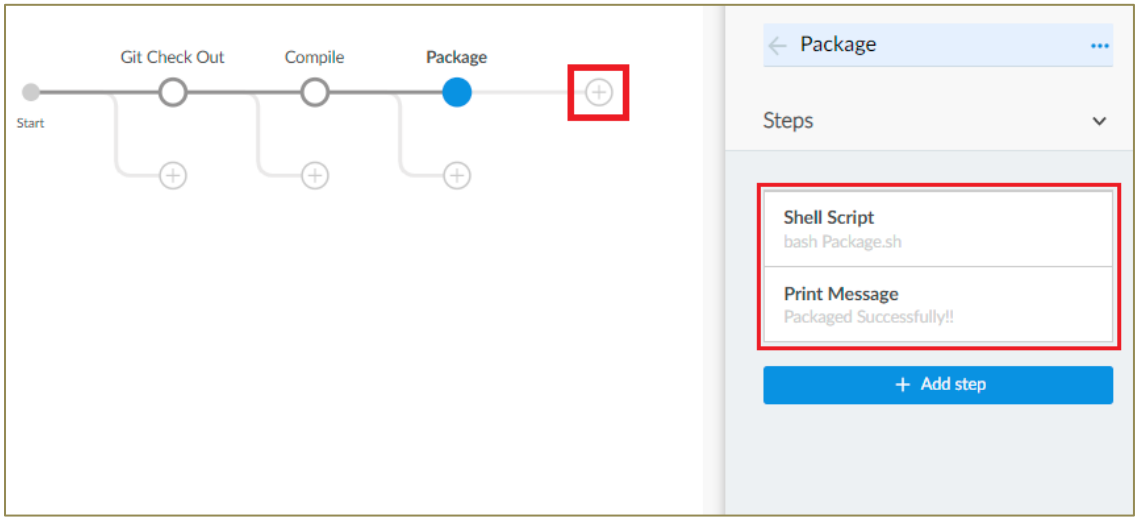
iv) Click **Add step** to add the further step to the Stage as shown below.



v) Select **Print Message** and type **Packaged Successfully!!** in the **Message** box as shown below.

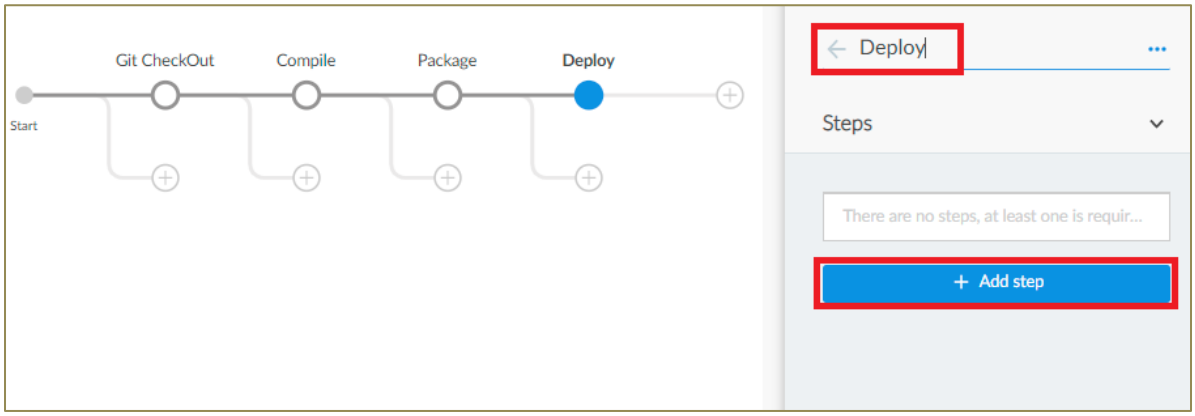


vi) Verify the Steps added to the **Package** stage and click on the + symbol to add the new stage.

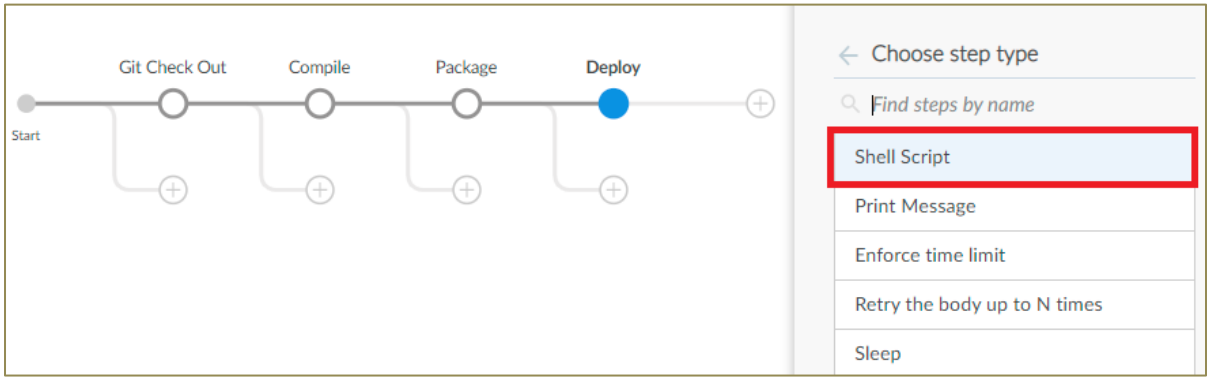


o. Add next Stage **Deploy** in the Jenkins pipeline.

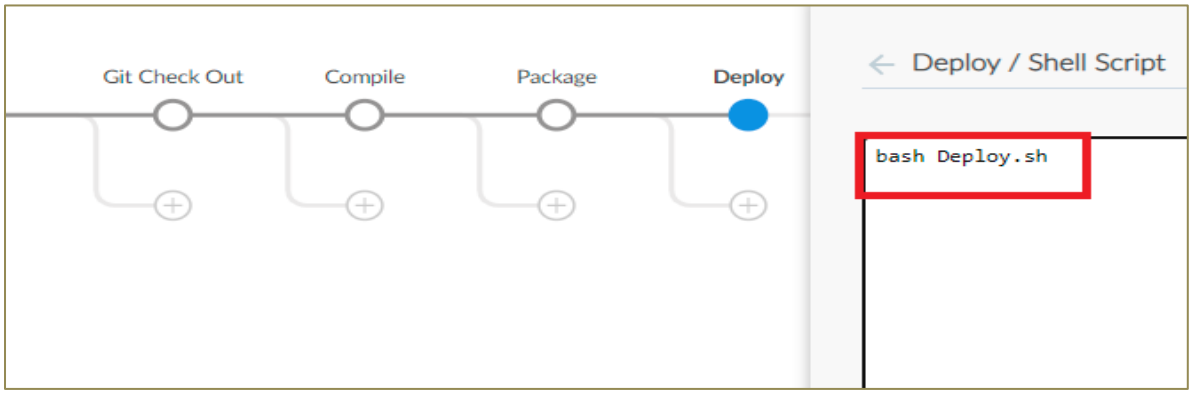
i) Provide the name for the Stage as shown below and click on **Add Step** to add steps to the Stage.



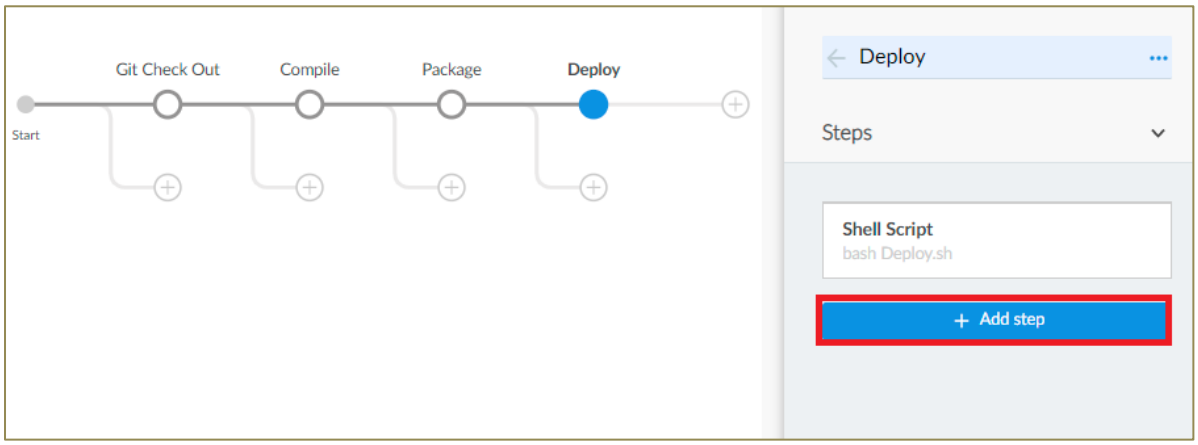
ii) Choose the step type as **Shell Script** as shown below.



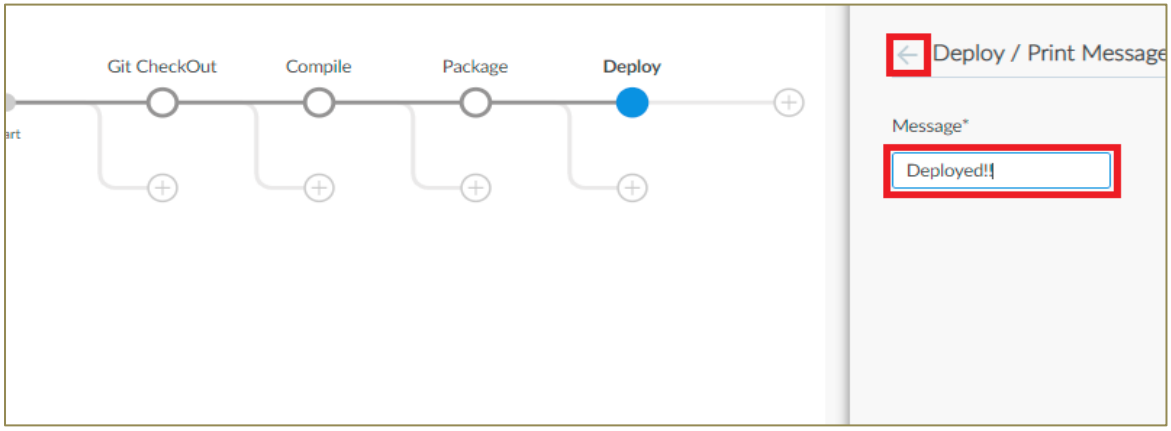
iii) Type the command in the Shell Script to execute the file.



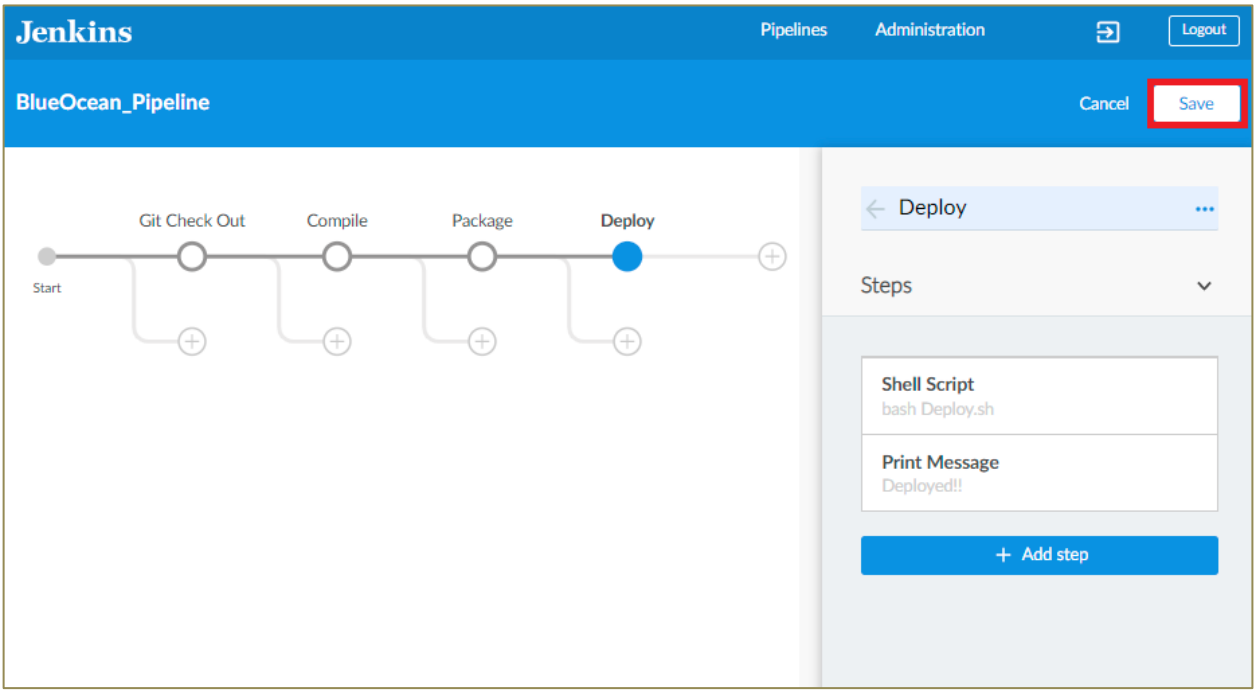
iv) Click **Add step** to add the further step to the Stage as shown below.



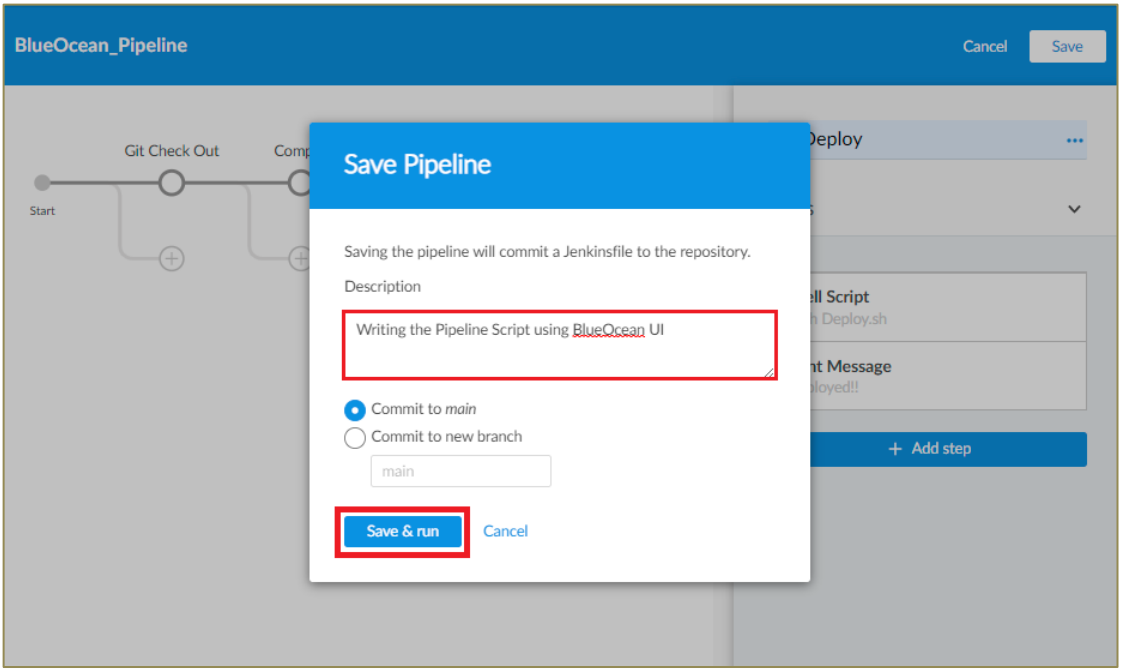
v) Select **Print Message** and type **Deployed!!** in the **Message** box as shown below.



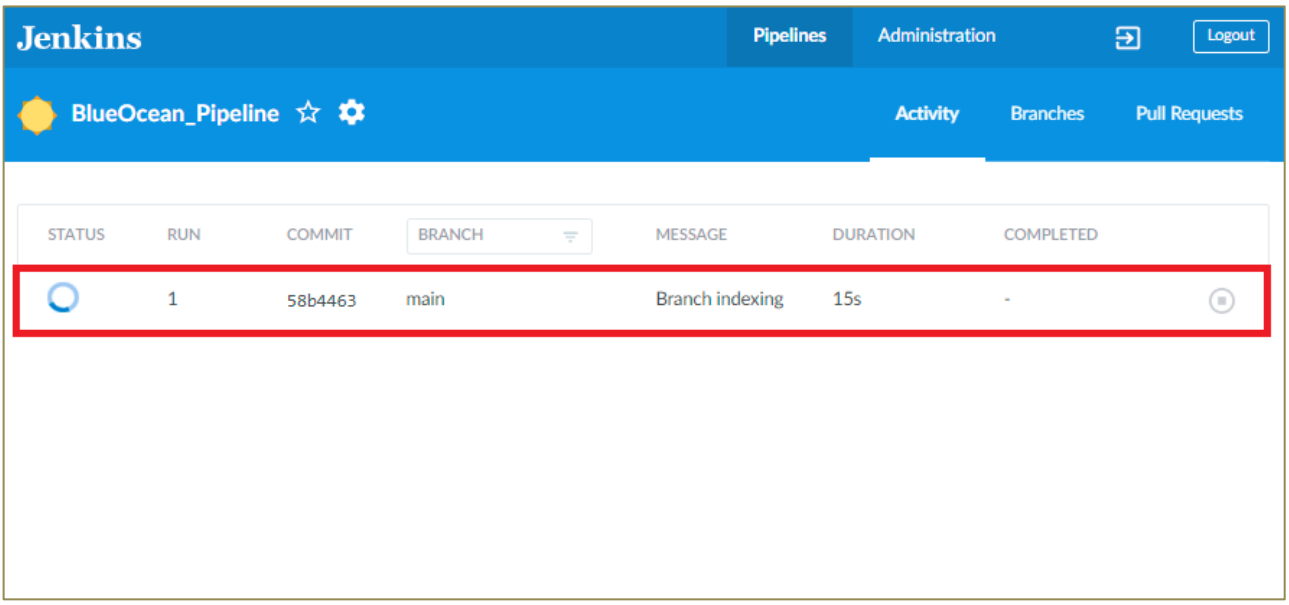
p. All the stages for the **Pipeline** is added successfully. Click **Save** as shown below.



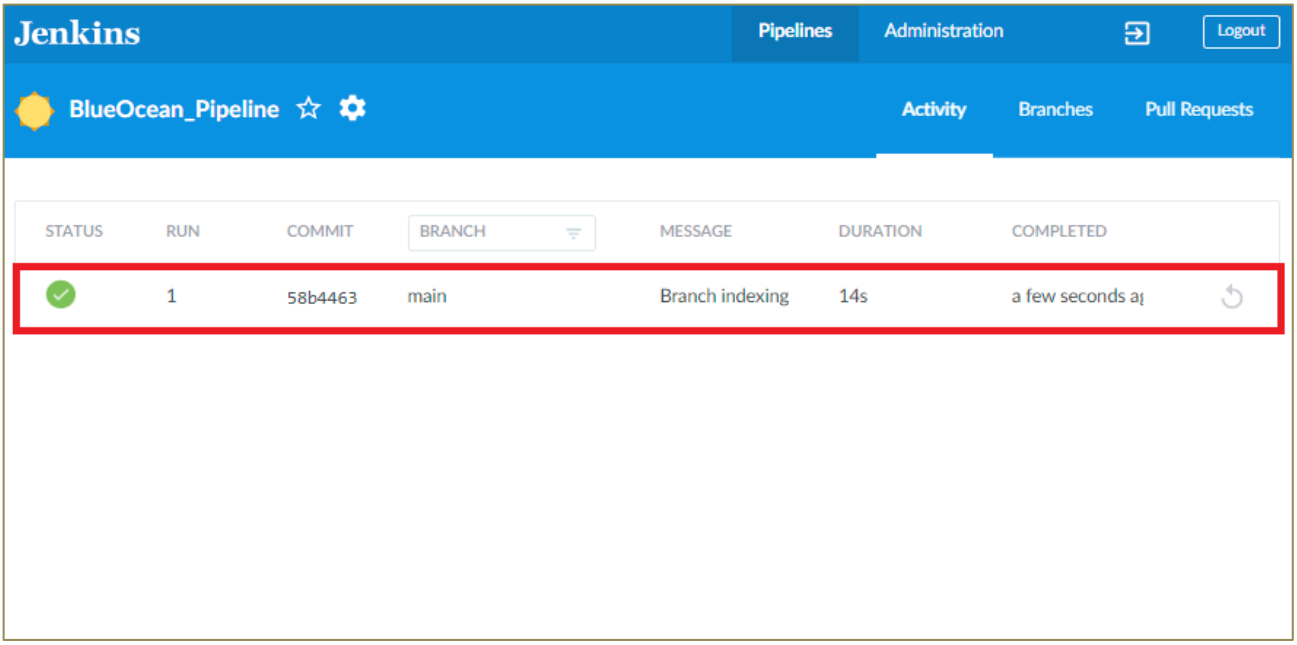
q. In the Save Pipeline dialog box type the description "**Writing the Pipeline Script using BlueOcean UI**" and click **Save & run** as shown below.



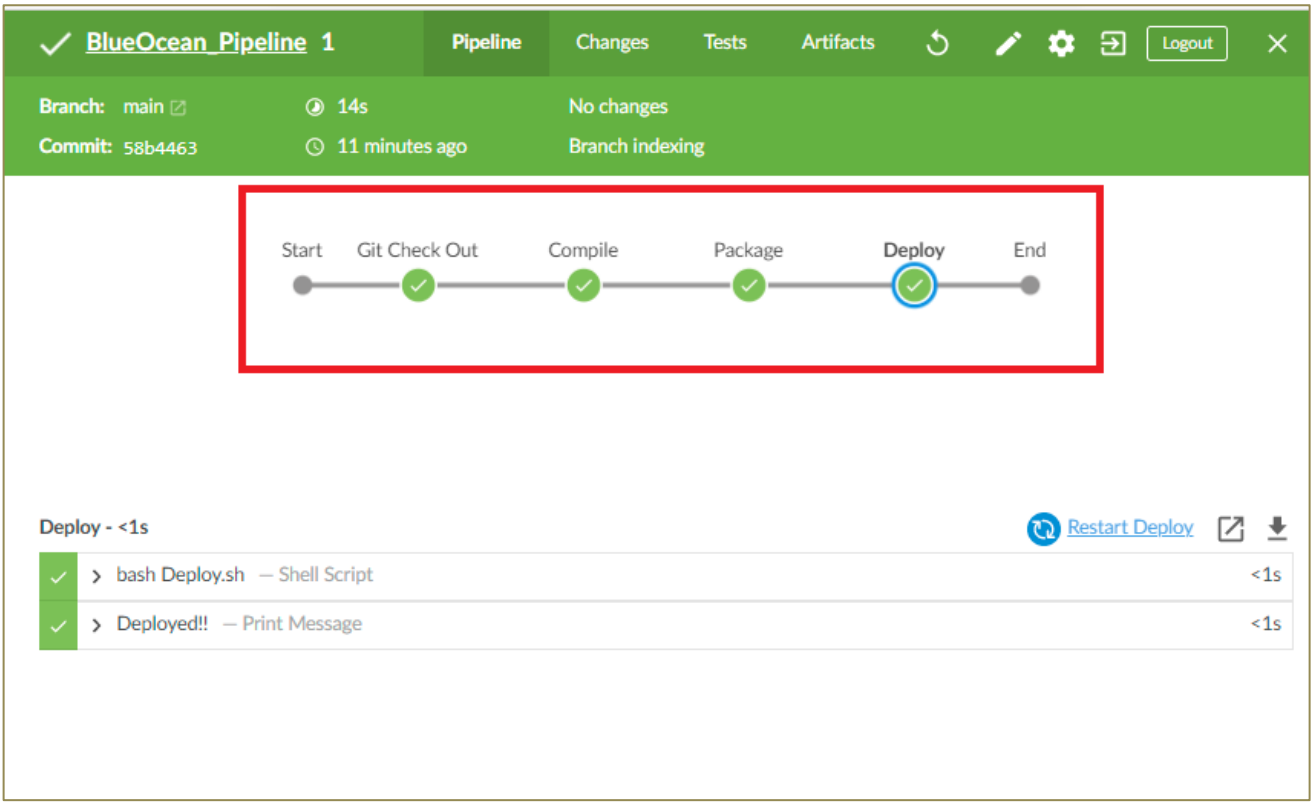
r. Verify that the **Pipeline** starts creating as shown below.



s. Verify that the **Pipeline** is created successfully as shown below and click on the pipeline.

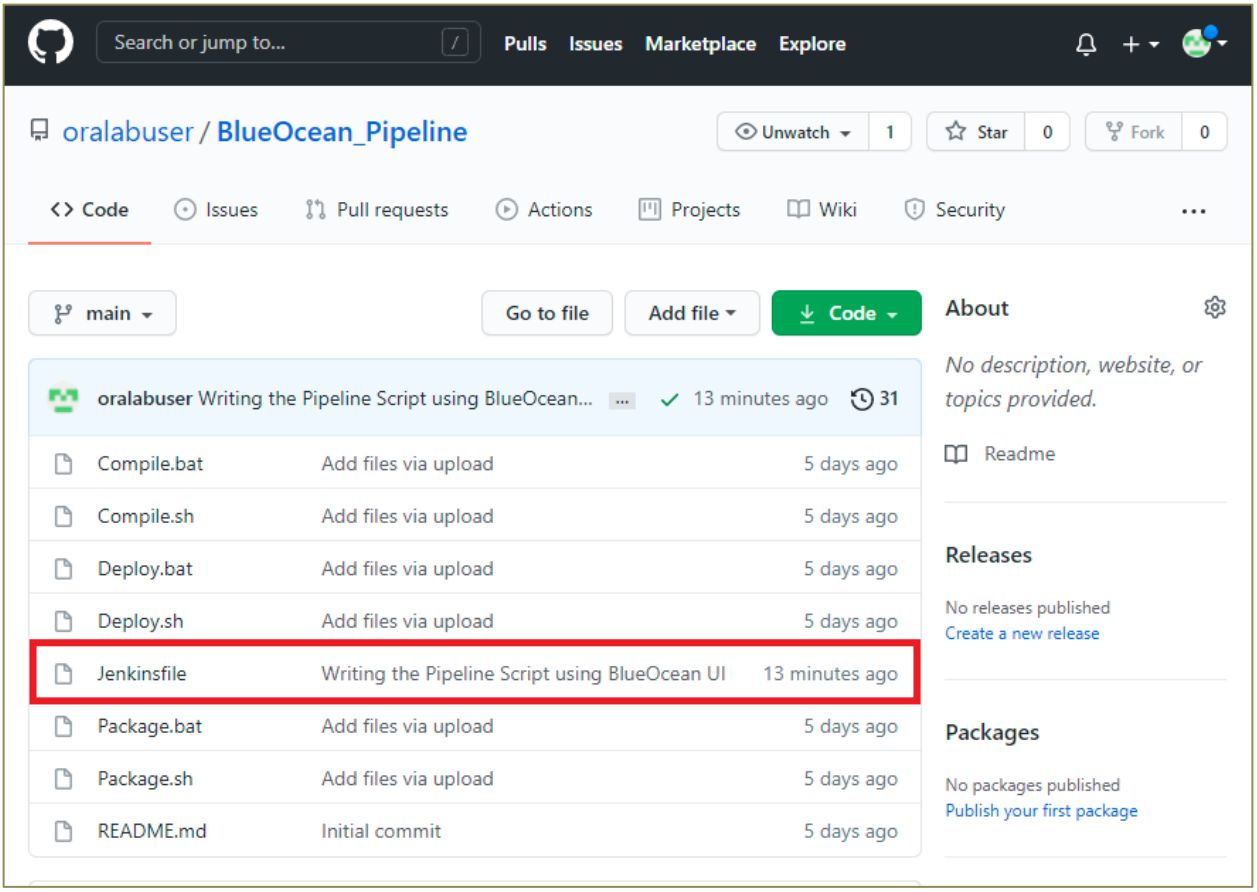


t. View the **Pipeline** stages which are created and verify the out messages from the stages as shown below.



u. Open Jenkins repository and verify the **Jenkinsfile** being created as shown below.

Note: Refresh the page.



v. Click on the **Jenkinsfile** and view the Groovy script for the Pipeline creation as shown below.

main BlueOcean_Pipeline / Jenkinsfile Go to file ···

oralabuser Writing the Pipeline Script using BlueOcean UI Latest commit 82f8e76 3 days ago History

1 contributor

33 lines (29 sloc) 570 Bytes Raw Blame

```
pipeline {
  agent any
  stages {
    stage('Git Check Out') {
      steps {
        git(url: 'https://github.com/oralabuser/BlueOcean_Pipeline.git', branch: 'main')
        echo 'Successfully Checkout!!'
      }
    }

    stage('Compile') {
      steps {
        sh 'bash Compile.sh'
        echo 'Compiled Successfully!!'
      }
    }

    stage('Package') {
      steps {
        sh 'bash Package.sh'
        echo 'Packaged Successfully!!'
      }
    }

    stage('Deploy') {
```

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