

**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 4-4: Delegate a Pipeline Job using Agent in Jenkins Instance

### Overview

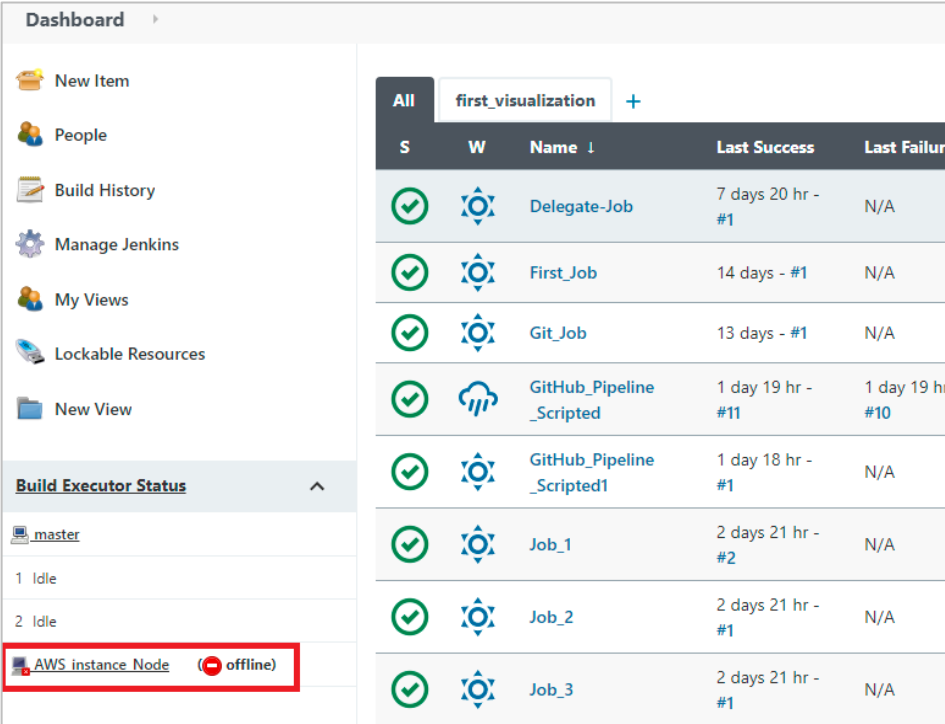
In this practice, you will learn how to delegate a pipeline Job using Agent in Jenkins Instance.

### Assumptions

You should have completed the Practice of Lesson 4-3.

### Tasks

1. Activate the Agent node configured in practice 3-3 in Jenkins instance.
  - a. In the Jenkins Dashboard page navigate to **Build Executor Status** and verify the Agent node status. Click on the agent node as shown below.



The screenshot shows the Jenkins Dashboard with the 'Build Executor Status' tab selected. On the left sidebar, 'Build Executor Status' is highlighted. Below it, the 'master' branch is shown with two idle executors. The 'AWS instance Node' is listed below, marked as 'offline' with a red icon. The main table displays the status of various jobs and their executors.

Dashboard				
All first_visualization +				
S	W	Name ↓	Last Success	Last Failure
✓	⚙️	Delegate-Job	7 days 20 hr - #1	N/A
✓	⚙️	First_Job	14 days - #1	N/A
✓	⚙️	Git_Job	13 days - #1	N/A
✓	🔗	GitHub_Pipeline_Scripted	1 day 19 hr - #11	1 day 19 hr - #10
✓	⚙️	GitHub_Pipeline_Scripted1	1 day 18 hr - #1	N/A
✓	⚙️	Job_1	2 days 21 hr - #2	N/A
✓	⚙️	Job_2	2 days 21 hr - #1	N/A
✓	⚙️	Job_3	2 days 21 hr - #1	N/A

- b. To activate the **Agent node**, copy the link from **Run from agent command line** as shown below.

Dashboard > Nodes > AWS\_instance\_Node

Back to List

Status

Delete Agent

Configure

Build History

Load Statistics

Log

**Build Executor Status**

## Agent AWS\_instance\_Node

Mark this node temporarily offline

**Connection was broken**

```
java.nio.channels.ClosedChannelException
    at org.jenkinsci.remoting.protocol.NetworkLayer.onRecvClosed(NetworkLayer.java:154)
    at org.jenkinsci.remoting.protocol.impl.NIONetworkLayer.ready(NIONetworkLayer.java:142)
    at org.jenkinsci.remoting.protocol.IOHub$OnReady.run(IOHub.java:792)
    at
jenkins.util.ContextResettingExecutorService$1.run(ContextResettingExecutorService.java:28)
    at
jenkins.security.ImpersonatingExecutorService$1.run(ImpersonatingExecutorService.java:68)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
```

Connect agent to Jenkins one of these ways:

- Launch Launch agent from browser
- Run from agent command line:

```
java -jar agent.jar -jnlpUrl http://34.208.77.168:8080/computer/AWS_instance_Node/jenkins-agent.jnlp -secret f293bf0b6b737573ae5ccc822c24779c22054fdc6a3424a7b232f60e6f328625 -workDir "/home/ec2-user/jenkins"
```

Run from agent command line, with the secret stored in a file:

```
echo f293bf0b6b737573ae5ccc822c24779c22054fdc6a3424a7b232f60e6f328625 > secret-file
java -jar agent.jar -jnlpUrl http://34.208.77.168:8080/computer/AWS_instance_Node/jenkins-
```

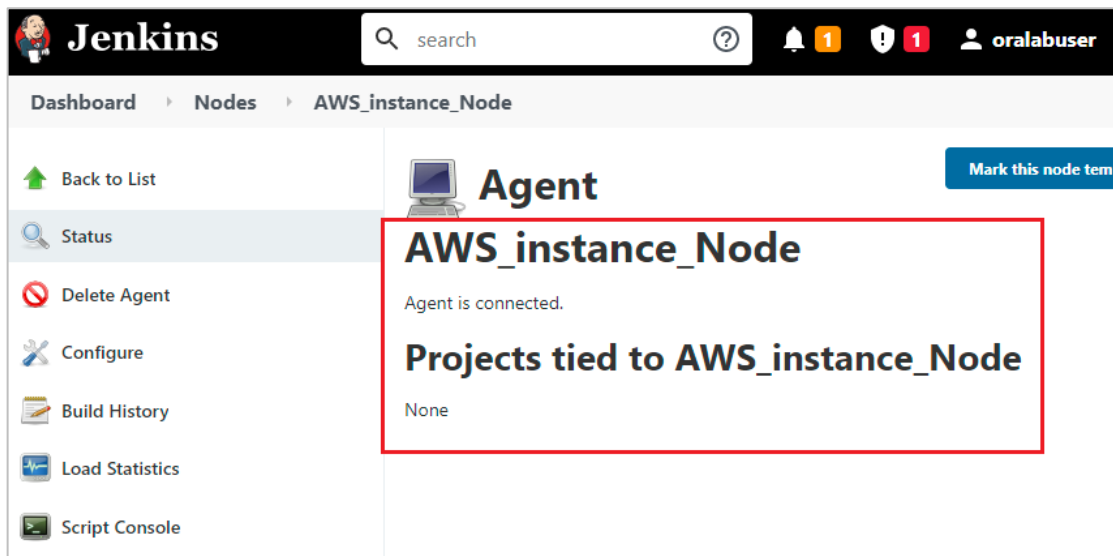
- c. Open Putty terminal connected to the **agent-node** and paste the command to execute as shown below.

```
[ec2-user@ip-172-31-26-239 ~]$ java -jar agent.jar -jnlpUrl http://34.208.77.168:8080/computer/AWS_instance_Node/jenkins-agent.jnlp -secret f293bf0b6b737573ae5ccc822c24779c22054fdc6a3424a7b232f60e6f328625 -workDir "/home/ec2-user/jenkins"
Apr 15, 2021 10:29:23 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/jenkins/remoting as a remoting work directory
Apr 15, 2021 10:29:23 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/jenkins/remoting
Apr 15, 2021 10:29:23 AM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: AWS_instance_Node
Apr 15, 2021 10:29:24 AM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Apr 15, 2021 10:29:24 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.7
```

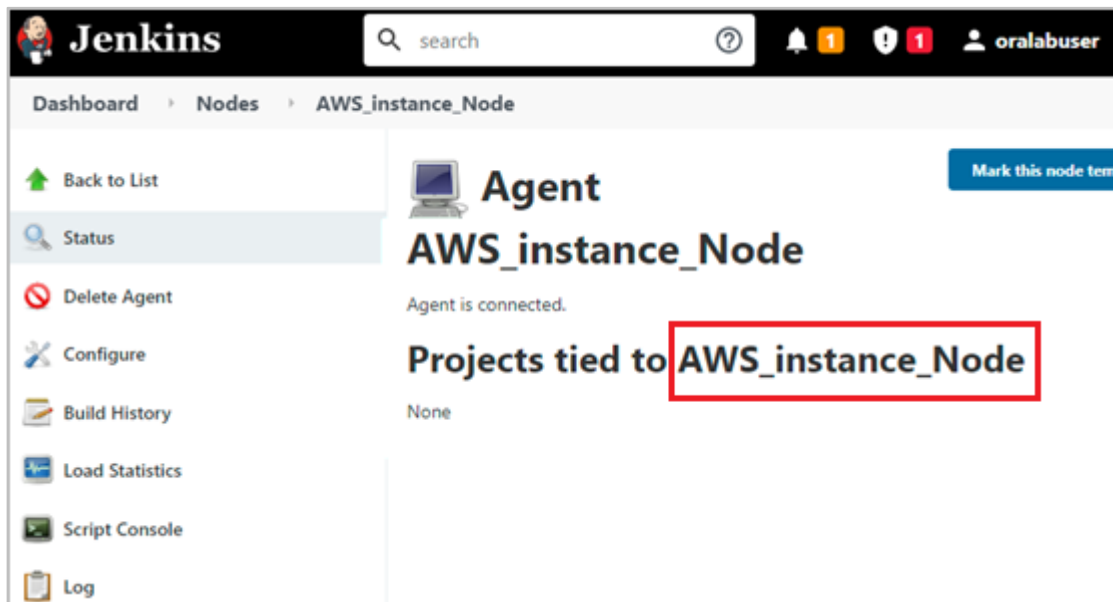
- d. On successful execution, the connection will be established by displaying **Connected** as shown below.

```
Apr 15, 2021 10:29:24 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Apr 15, 2021 10:29:24 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 34.208.77.168:46111
Apr 15, 2021 10:29:24 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Apr 15, 2021 10:29:24 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 22:59:fd:4e:67:d9:7d:87:06:49:16:00:6c:13:c8:2d
Apr 15, 2021 10:29:25 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

- e. Open Jenkins dashboard and refresh the agent node page to verify the connection to master node as shown below.



- f. Copy the label of the agent node to execute the pipeline job on agent node as shown below.



2. Create a new pipeline job to be executed on the agent node in Jenkins
  - a. In Jenkins Dashboard, navigate to main menu, select **New item** and provide the name for the job, select **Pipeline** and click **OK**

Dashboard > All >

### Enter an item name

AgentScripted\_Job

» Required field

**Freestyle project**  
 This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
 Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**  
 Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**External Job**  
 This type of job allows you to record the execution of a process run outside Jenkins, even on a remote system. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

**OK**

- b. Copy the Groovy script provided below and paste it in the **Script** block. As shown below provide the respective label of the Agent node in Jenkins.

**Note:** It is the same code used in practice 4-3.

```

pipeline {
    agent {
        label 'AWS instance Node'
    }
    stages {
        stage('Git-Checkout') {
            steps {
                echo "Checking out from Git Repo";
                git branch: 'main', url:
                'https://github.com/oralabuser/PipelineScript.git'
            }
        }
        stage('Build') {
            steps {
                echo "Building the checked-out project!";
                sh 'bash Build.sh'
            }
        }
    }
}
  
```

```

        stage('Unit-Test') {
            steps {
                echo "Running JUnit Tests";
                sh 'bash Unit.sh'
            }
        }
        stage('Quality-Gate') {
            steps {
                echo "Verifying Quality Gates";
                sh 'bash Quality.sh'
            }
        }
        stage('Deploy') {
            steps {
                echo "Deploying to Stage Environment for more
tests!";
                sh 'bash Deploy.sh'
            }
        }
    }
    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'
        }
    }
}

```

- c. Click **Save** as shown below.

General Build Triggers Advanced Project Options **Pipeline**

**Pipeline**

Definition

Pipeline script

Script

```
1 pipeline {
2   agent {
3     label 'AWS_instance_Node'
4   }
5   stages {
6     stage('Git-Checkout') {
7       steps {
8         echo "Checking out from Git Repo";
9         git branch: 'main', url: 'https://github.com/oralabuser/PipelineScript.git'
10      }
11    }
12    stage('Build') {
13      steps {
14        echo "Building the checked-out project!";
15        sh 'bash Build.sh'
16      }
17    }
18    stage('Unit-Test') {
```

☒ Use Groovy Sandbox

Pipeline Syntax

**Save** Apply

- d. Select **Build Now** from the menu to execute the Pipeline agent scripted job as shown below.

Dashboard > AgentScripted\_Job >

Back to Dashboard

Status

Changes

**Build Now**

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

## Pipeline AgentScripted\_Job

Recent Changes

### Stage View

No data available. This Pipeline has not yet run.

### Permalinks



- e. As shown below **view** the **stages** of **pipeline** getting executed on the agent node and click on the link under **Build History** as shown below.

The screenshot shows the Jenkins Pipeline AgentScripted\_Job dashboard. On the left is a sidebar with navigation links: Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. Below these is the Build History section with a search bar and a list of builds. Build #1 is highlighted with a red box, showing a timestamp of Apr 23, 2021 10:44 AM. The main content area is titled 'Pipeline AgentScripted\_Job' and includes a 'Recent Changes' section. Below this is the 'Stage View' table, which is also highlighted with a red box. The table shows the average stage times and the execution times for the first build (#1).

	Git-Checkout	Build	Unit-Test	Quality-Gate	Deploy	Declarative: Post Actions
Average stage times: (Average full run time: ~3s)	4s	571ms	192ms	193ms	202ms	67ms
#1 Apr 23 16:14 No Changes	591ms	1s	348ms	357ms	375ms	68ms

- f. Click on **Console Output** to view the execution of the scripted pipeline on agent node integrating with the GitHub.

The screenshot shows the Jenkins Pipeline AgentScripted\_Job #2 console output. On the left is a sidebar with navigation links: Back to Project, Status, Changes, Console Output (highlighted with a red box), View as plain text, Edit Build Information, Delete build '#2', Git Build Data, Restart from Stage, Replay, and Pipeline Steps. The main content area is titled 'Console Output' and shows the execution log. A red box highlights the line 'Running on AWS\_instance\_Node in /home/ec2-user/jenkins/workspace/AgentScripted\_Job'.

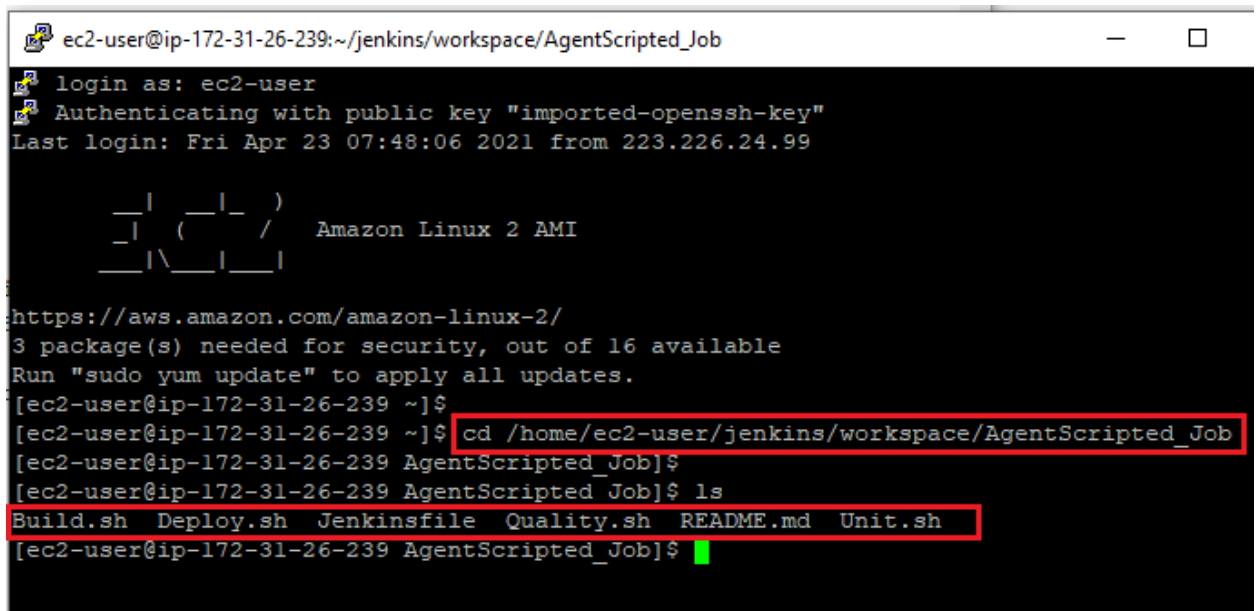
```

Started by user orlabuser
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on AWS_instance_Node in /home/ec2-user/jenkins/workspace/AgentScripted_Job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git-Checkout)
[Pipeline] echo
Checking out from Git Repo
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision 88e4246d7b255c6b4b262a0b4b2a4f9199d9469b (refs/remotes/origin/main)
Commit message: "Create Unit.sh"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building the checked-out project!
  
```

- g. As shown below the scripted pipeline is executed successfully on agent node.

```
+ bash Deploy.sh
Deploying Build : 04/23/21 : 10:44:57
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
This will always run
[Pipeline] echo
This will run only if the state of the pipeline has changed
[Pipeline] echo
For example, if the Pipeline was previously failing but is now successful
[Pipeline] echo
This will run only if successful
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

- h. Connect to **agent node** instance from Putty to verify the path of the workspace consisting of the GitHub repository files as shown below.



```
ec2-user@ip-172-31-26-239:~/jenkins/workspace/AgentScripted_Job
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Apr 23 07:48:06 2021 from 223.226.24.99

 _ _ | _ _ | _ )
 _ | ( _ _ /   Amazon Linux 2 AMI
 _ | \ _ | _ _ |

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-239 ~]$
[ec2-user@ip-172-31-26-239 ~]$ cd /home/ec2-user/jenkins/workspace/AgentScripted_Job
[ec2-user@ip-172-31-26-239 AgentScripted_Job]$
[ec2-user@ip-172-31-26-239 AgentScripted_Job]$ ls
Build.sh  Deploy.sh  Jenkinsfile  Quality.sh  README.md  Unit.sh
[ec2-user@ip-172-31-26-239 AgentScripted_Job]$
```

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