

**Practice for Lesson 3: Assign
Roles to User and Build
Master-slave configuration in
Jenkins Instance**

Practices for Lesson 3

Overview

In these practices, you will assign Role based authorization plugins to users created in Jenkins instance. Further, you will learn how to build a master-slave configuration on a Jenkins instances and followed by Backing up the configuration files of Jenkins in the AWS EC2 instance.

Practice 3-2: Distributed Build Master-Slave Configuration

Overview

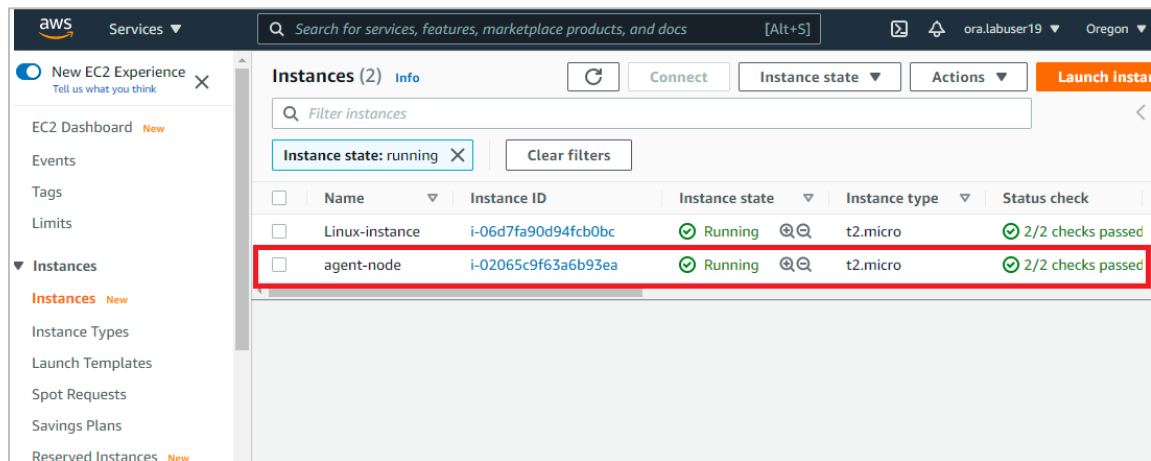
In this practice, you will learn how to build a master-slave configuration on a Jenkins instances and build a Job on Jenkins AWS agent instance.

Assumptions

You should have completed the Practice of Lesson 3-1.

Tasks


1. Create a new instance named agent-node in the AWS EC2 instance.
 - a. Follow the steps in practice 1-1 to create the instance in AWS EC2 console.
 - b. As shown below, create the agent-node for the master-slave configuration on Jenkins.



2. Enable Java Network Launch Protocol (JNLP) port in Jenkins instance.
 - a. Navigate to **Manage Jenkins** and select **Configure Global Security** under **Security** as shown below.

System Configuration


Configure System
 Configure global settings and paths.


Global Tool Configuration
 Configure tools, their locations and automatic installers.



Manage Plugins
 Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
 🚨 There are updates available


Manage Nodes and Clouds
 Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security


Configure Global Security
 Secure Jenkins; define who is allowed to access/use the system.


Manage Credentials
 Configure credentials


Configure Credential Providers
 Configure the credential providers and types


Manage and Assign Roles
 Handle permissions by creating roles and assigning them to users/groups


Manage Users
 Create/delete/modify users that can log in to this Jenkins

- b. Under **Agents**, select **Random** to open the ports of JNLP and click **Save** as shown below.

Markup Formatter

Markup Formatter

Plain text

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

Agents

TCP port for inbound agents

☐ Fixed :
☒ Random
 ☐ Disable

Agent protocols...

CSRF Protection

Crumb Issuer

Default Crumb Issuer


☐ Enable proxy compatibility


3. Create an agent node (Slave node) in the Jenkins instance.
- a. Navigate to **Manage Jenkins** and select **Manage Nodes and Clouds** under **Security Configuration** as shown below.


Manage Jenkins


New version of Jenkins (2.287) is available for [download](#) ([changelog](#)).

System Configuration

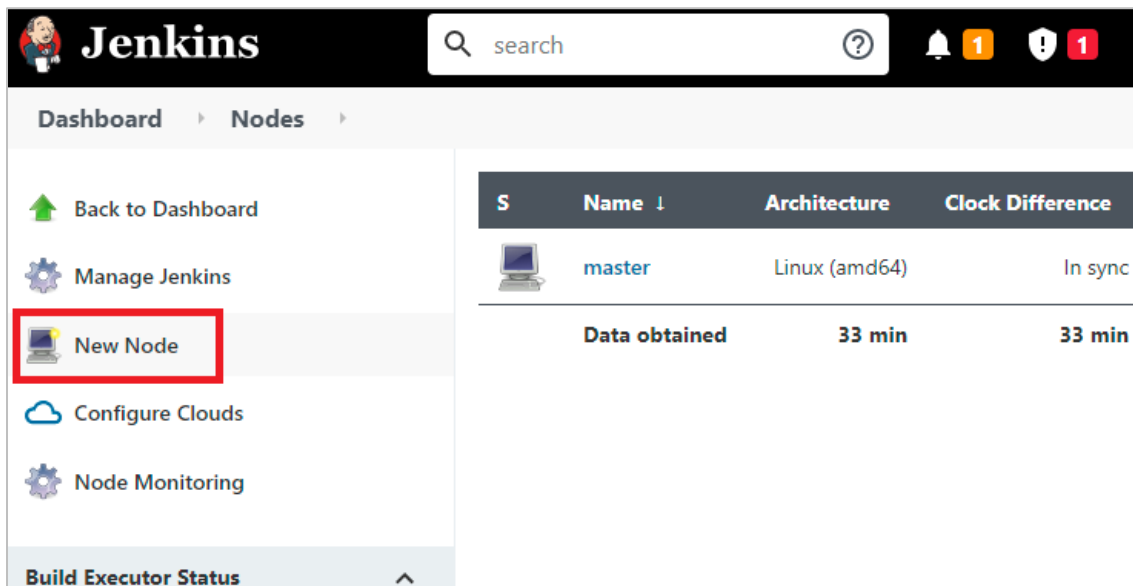

Configure System
 Configure global settings and paths.


Global Tool Configuration
 Configure tools, their locations and automatic installers.



Manage Plugins
 Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
 🚨 There are updates available


Manage Nodes and Clouds
 Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

- b. Verify that the Jenkins instances consists of only master node. Select **New Node** to add a new node to the Jenkins instance.



The screenshot shows the Jenkins dashboard with the 'Nodes' tab selected. On the left sidebar, the 'New Node' button is highlighted with a red box. The main content area displays a table with the following data:

S	Name ↓	Architecture	Clock Difference
	master	Linux (amd64)	In sync
	Data obtained	33 min	33 min

- c. Provide the name for the node, select **Permanent Agent** and click **OK** as shown below.

Dashboard > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Executor Status ^

master

1 Idle

Node name

AWS_instance_Node

☒ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

OK

- d. Provide a path in **Remote root directory** and **AWS_instance_Node** as label in **Labels** section as shown below.

Name

AWS_instance_Node

Description

of executors

1

Remote root directory

/home/ec2-user/jenkins

Labels

AWS_instance_Node

Usage

- e. Scroll down to **Usage** and select **Only build jobs with label expressions matching this node**, and for **Launch method** select **Launch agent by connecting it to the master** as shown below.

Labels ?

AWS_instance_Node

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agent by connecting it to the master

☐ Disable WorkDir ?

Custom WorkDir path ?

Internal data directory ?

remoting

☐ Fail if workspace is missing ?

- f. Scroll down to end of the page and click **Save**.

Availability ?

Keep this agent online as much as possible

Node Properties

☐ Disable deferred wipeout on this node ?

☐ Environment variables

☐ Tool Locations

Save

- g. Verify the agent node is created as shown below with the red cross mark (x), it indicates that the agent node is not connected to the master node.

Dashboard > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds



Node Monitoring

Build Queue


S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free
	AWS_instance_Node		N/A	N/A	
	master	Linux (amd64)	In sync	5.83 GB	
Data obtained		2 min 20 sec	2 min 20 sec	2 min 20 sec	

Refresh status

4. Provision the Agent node created in the Jenkins instance.
 - a. Click on the new agent node created as shown below.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free
	AWS_instance_Node		N/A	N/A	
	master	Linux (amd64)	In sync	5.83 GB	
Data obtained		2 min 20 sec	2 min 20 sec	2 min 20 sec	
Refresh status					

- b. To connect to the agent node, download the **agent.jar** file and **slave-agent.jnlp** file.




Agent

AWS_instance_Node

Mark this node temporarily offline

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-agent.jnlp -secret
@secret-file -workDir "/home/ec2-user/jenkins"
```

- c. The files are downloaded successfully to the local system. These files have to be placed in the AWS agent-node instance.

Jenkins search ? 1 1 oralabuser log out

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

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-agent.jnlp -secret @secret-file -workDir "/home/ec2-user/jenkins"
```

Projects tied to AWS_instance_Node

None

agent.jar	...	jenkins-agent.jnlp	...
Open file		Open file	

Show all

- d. Open **WinSCP** to transfer the files to the **agent-node** instance.

Note: Download and install WinSCP if it is not installed in your local system. It is an open source FTP tool.

Login

New Site

Session

File protocol: SFTP

Host name: Port number: 22

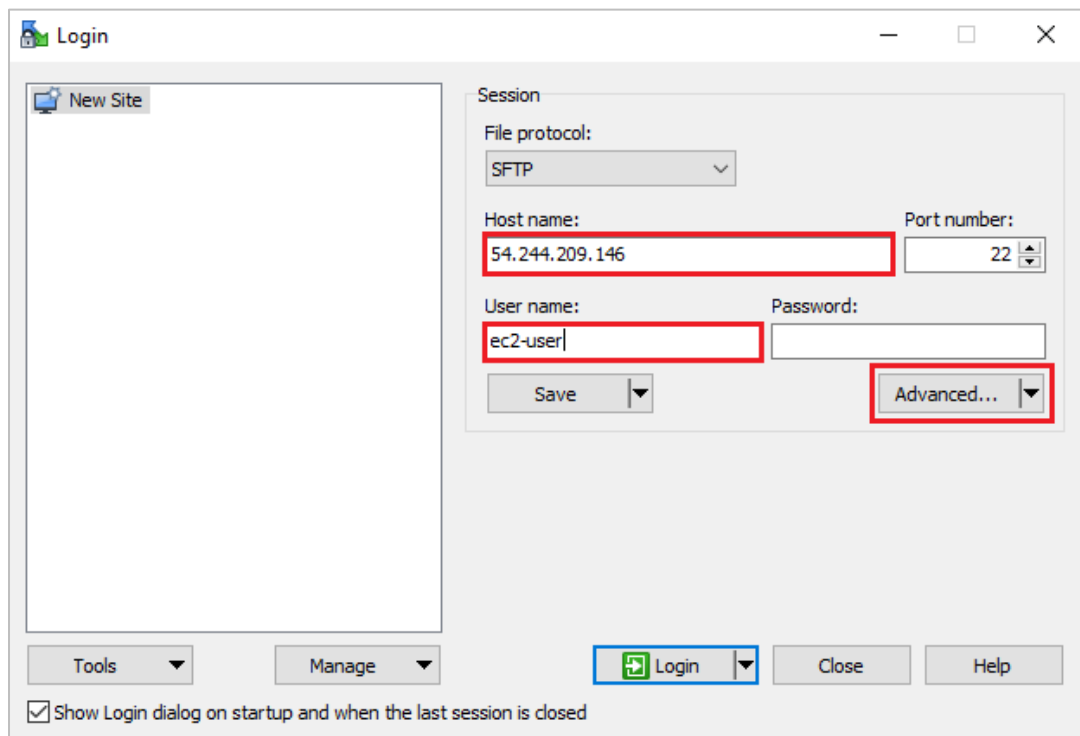
User name: Password:

Save Advanced...

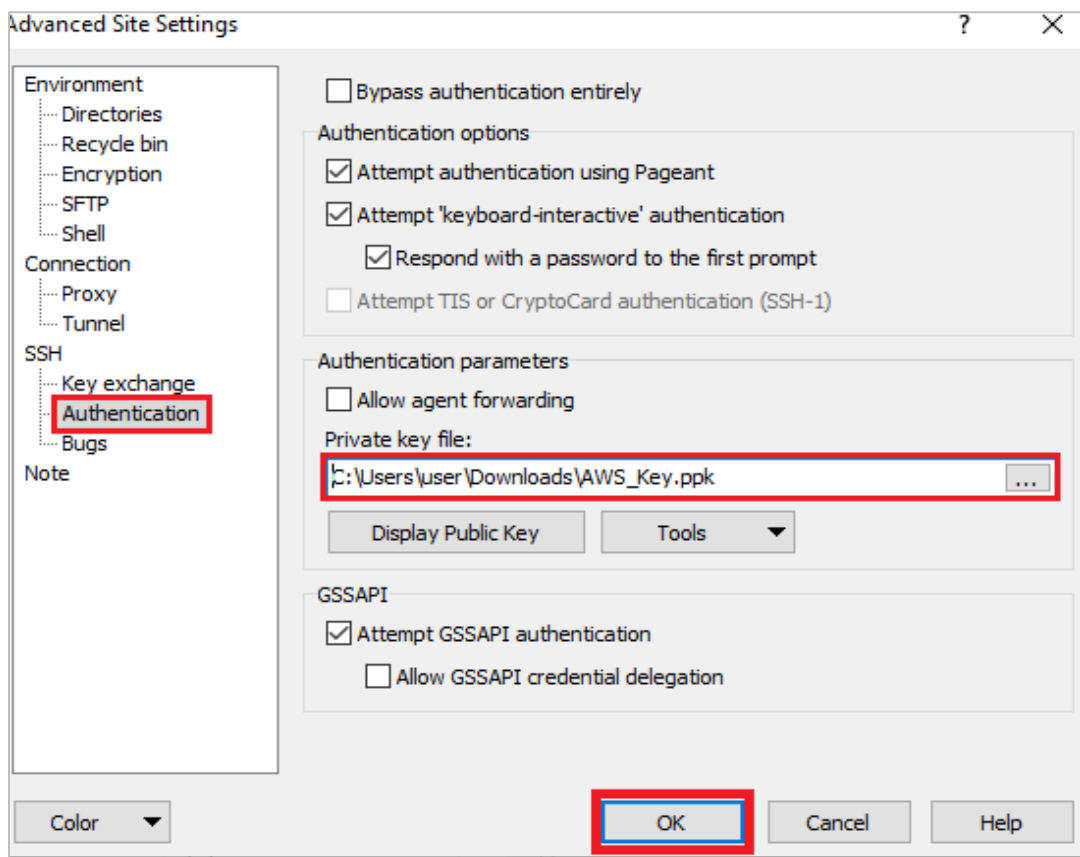
Tools Manage Login Close Help

☒ Show Login dialog on startup and when the last session is closed

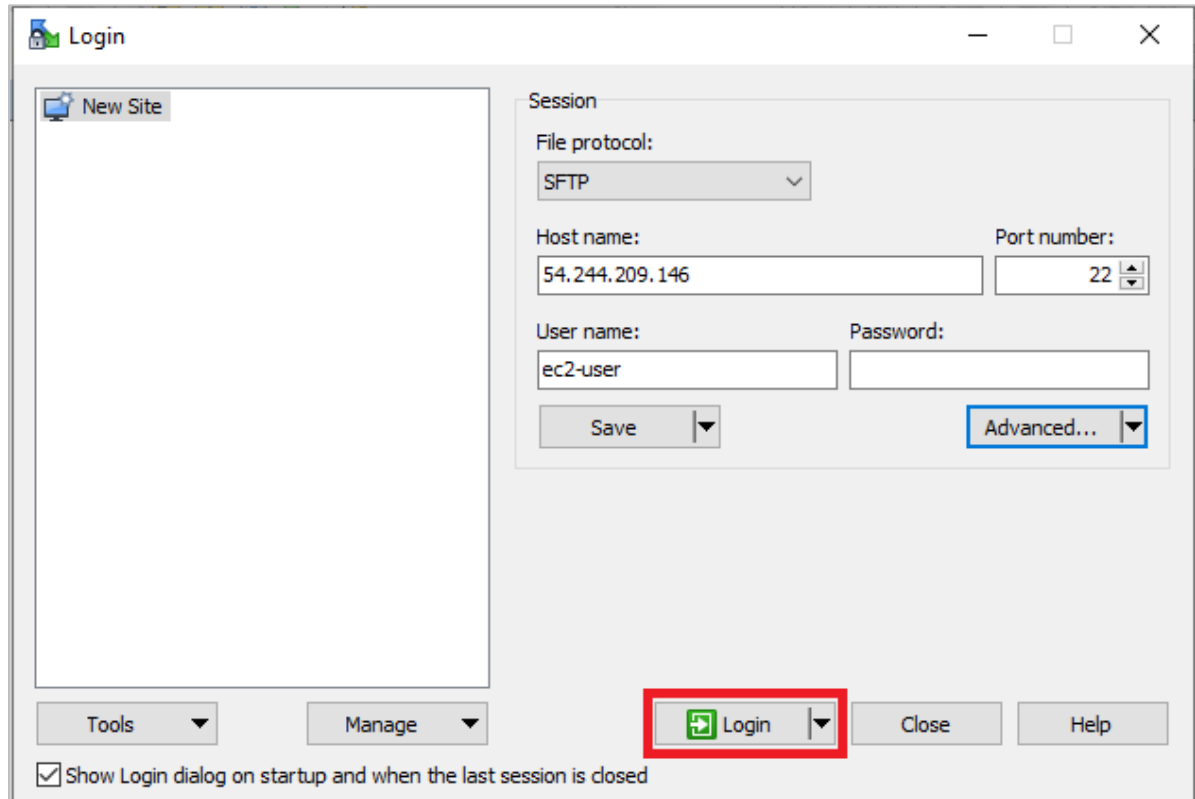
- e. Provide the **Public IP address** of the **agent-node** instance for the **Host name**, **ec2-user** for **User name** and click **Advance** to provide the **Password** as shown below.



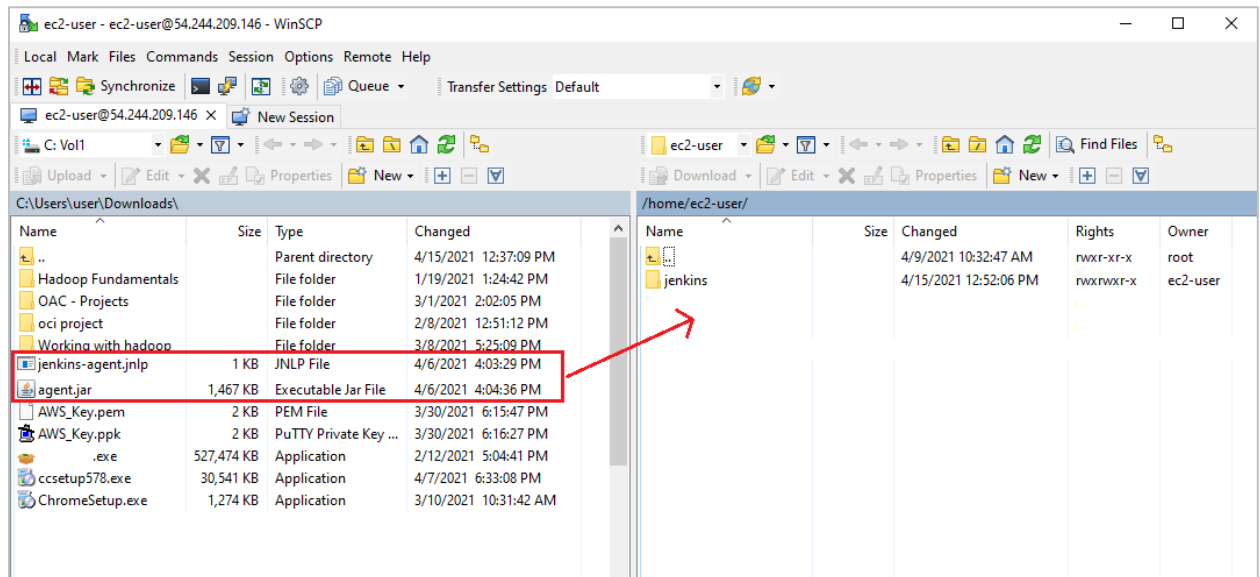
- f. In **Advanced Site Settings** navigate to **SSH** and select **Authentication**. Browse the path for the **Private key file** for the instance connection and click **OK** as shown below.



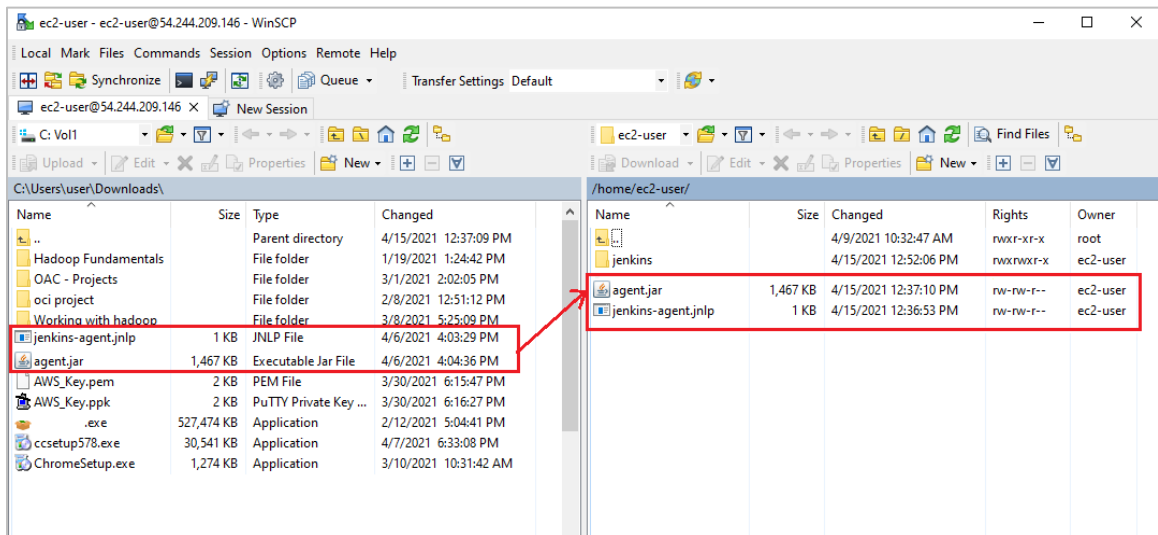
- g. Click **Login** to connect to the **agent-node** instance to transfer the files.



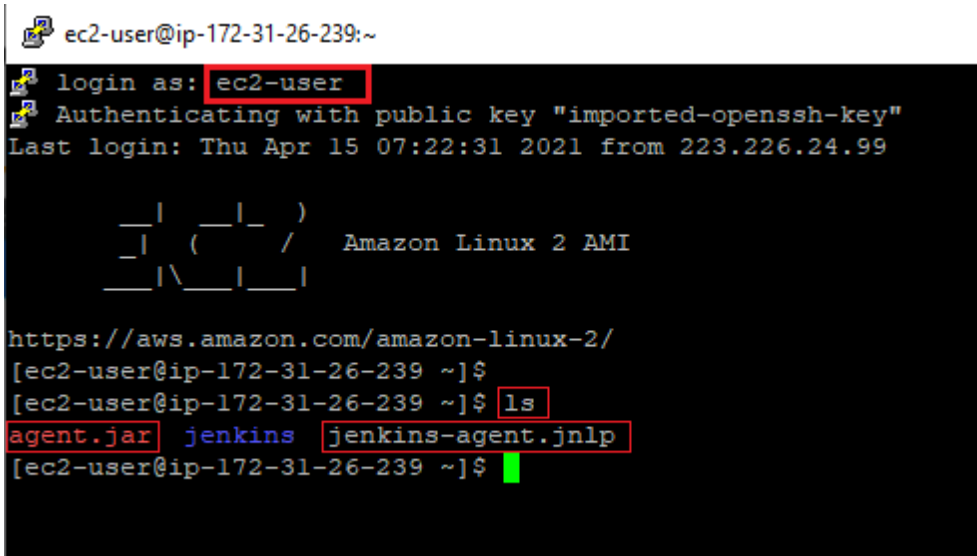
- h. As shown below, select the files from left box, drag and drop it to the right box.




- i. The files have been transferred to the connected **agent-node** AWS instance as shown below.



- j. Connect to the **agent-node** instance from Putty and list the files in it as shown below.



5. Launch agent.jar in the agent-node instance to establish the connection to the master node.
 - a. In Jenkins, click on the agent node and copy the code as shown below.




Agent

AWS_instance_Node

Mark this node temporarily offline

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-agent.jnlp -secret
@secret-file -workDir "/home/ec2-user/jenkins"
```

Projects tied to AWS_instance_Node

None

- b. 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
```

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

Note: As long as this command prompt is up and running, the agent would be connected.

```
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
```

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

The screenshot shows the Jenkins dashboard with the breadcrumb 'Dashboard > Nodes > AWS_instance_Node'. On the left sidebar, there are links: 'Back to List', 'Status', 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', 'Script Console', 'Log', and 'System Information'. The main content area is titled 'Agent AWS_instance_Node' with a sub-header 'Agent is connected.' and 'Projects tied to AWS_instance_Node: None'. A red box highlights the 'Agent' title, 'AWS_instance_Node', and the project list.

6. Create a new job to be executed on the agent node in Jenkins.
- a. In Jenkins Dashboard, navigate to main menu and select **New item** as shown below.

The screenshot shows the Jenkins dashboard with the breadcrumb 'Dashboard >'. In the left sidebar, the 'New Item' button is highlighted with a red box. The main content area displays a table of existing jobs. The table has columns: 'S' (Status), 'W' (Icon), 'Name', 'Last Success', and 'Last Failure'. There are two rows of jobs: 'First_Job' and 'Git_Job'. Below the table, there is a legend and links for 'Atom feed for all' and 'Atom feed for failure'.


S	W	Name	Last Success	Last Failure
✓	⚙️	First_Job	6 days 23 hr - #1	N/A
✓	⚙️	Git_Job	6 days 1 hr - #1	N/A

- b. Enter the name for the job, select **Freestyle project** and click **OK**.

Enter an item name


Delegate-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.



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.



Multi-configuration project

Projects that need a large number of different configurations, such as testing on multiple platforms, platform-specific builds, etc.

OK

- c. Select the check box for **Restrict where this project can be run** and provide the label of the agent node as shown below.

Dashboard > Delegate-Job >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

[Plain text] [Preview](#)

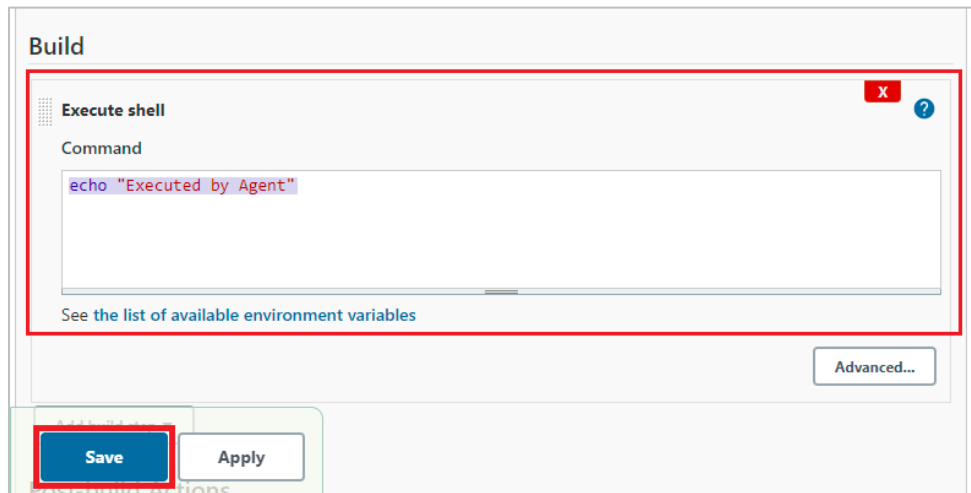
- ☐ Discard old builds
- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☒ Restrict where this project can be run

Label Expression

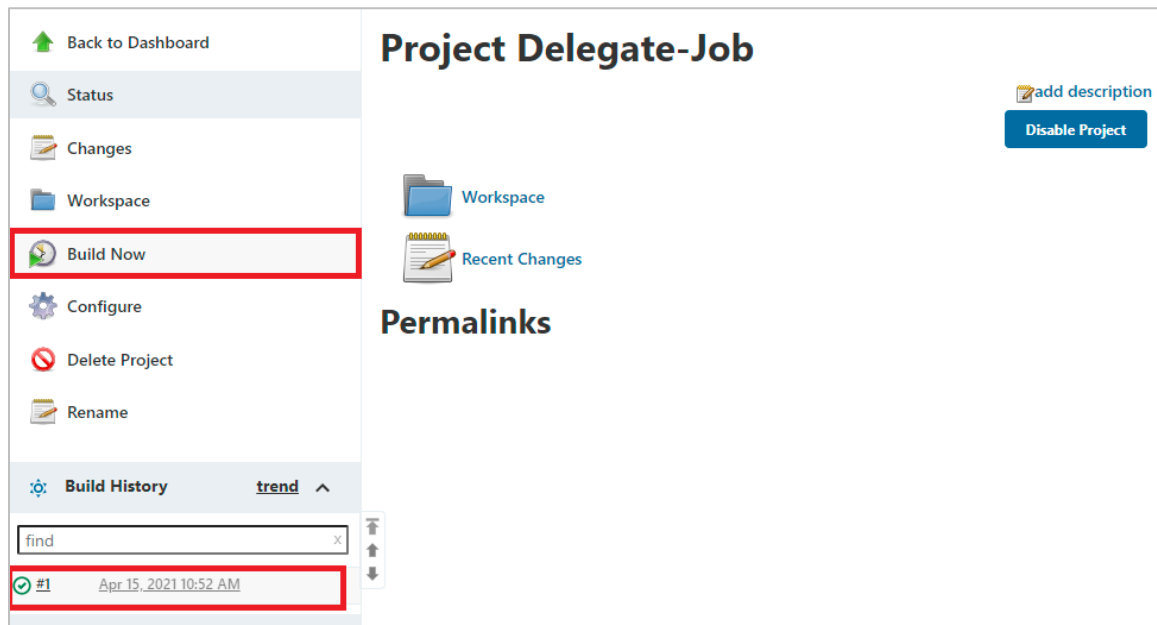
AWS_instance_Node

Advanced...

- d. Scroll down to **Build** and select **Execute shell**. Type the **echo** command as shown below in the command prompt and click **Save**.



- e. Click **Build Now** to build the job in Jenkins and click on the build job link provided in the bottom to verify the execution as shown below.



- f. Select Console Output to view the execution. Note the path of the workspace highlighted.

The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a search bar. Below the navigation bar, the breadcrumb trail is 'Dashboard > Delegate-Job > #1'. On the left sidebar, the 'Console Output' link is highlighted with a red box. The main content area displays the 'Console Output' for build #1 of 'Delegate-Job'. It shows a green checkmark icon and the text: 'Started by user oralabuser', 'Running as SYSTEM', 'Building remotely on AWS_instance_Node in workspace /home/ec2-user/jenkins/workspace/Delegate-Job', and a terminal snippet: '[Delegate-Job] \$ /bin/sh -xe /tmp/jenkins6965217127956023176.sh + echo \'Executed by Agent\''. Below this, it says 'Executed by Agent' and 'Finished: SUCCESS'. Other sidebar links include 'Back to Project', 'Status', 'Changes', 'View as plain text', 'Edit Build Information', and 'Delete build \'#1\''.

- g. Open Putty new terminal and navigate to the workspace path to view the Job name in the **agent-node** instance.

The screenshot shows a terminal window with the prompt 'ec2-user@ip-172-31-26-239:~'. It displays the login process for 'ec2-user' on an Amazon Linux 2 AMI. The terminal shows the command 'cd /home/ec2-user/jenkins/workspace/' being executed, which is highlighted with a red box. The output of the command is 'Delegate-Job'. The terminal prompt is now '[ec2-user@ip-172-31-26-239 workspace]\$'.

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