

**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 4-1: Triggering Jenkins Jobs Automatically

Overview

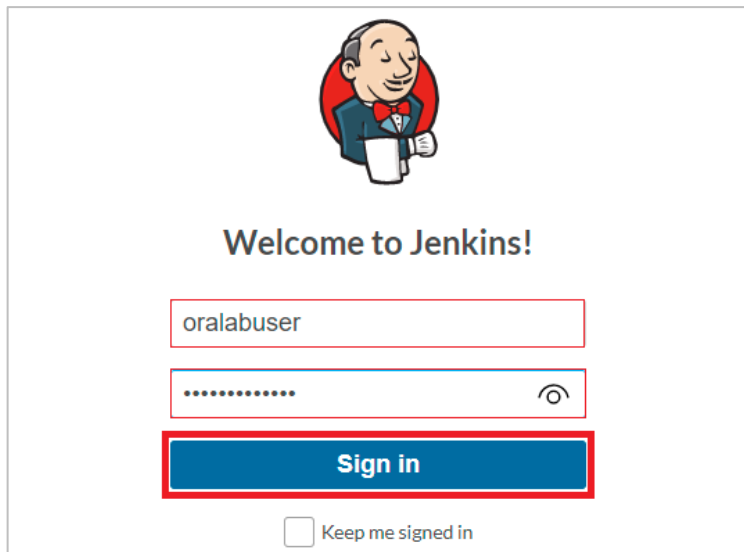
In this practice, you will learn how to trigger the Jenkins Jobs from one job to another automatically in the Jenkins Dashboard.

Assumptions

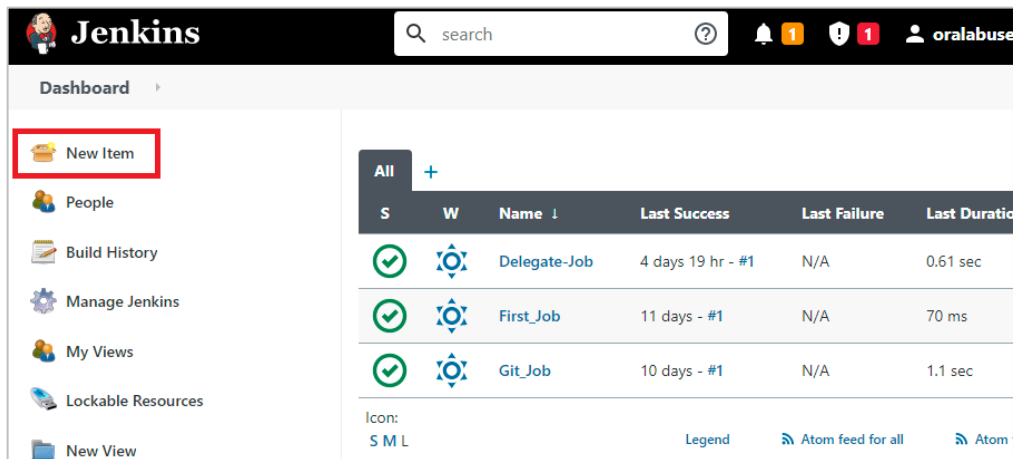
You should have completed the Practice of Lesson 3.

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: **34.208.77.168:8080**).

A screenshot of the Jenkins login page. At the top center is the Jenkins logo, a cartoon character with a red bow tie. Below the logo, the text "Welcome to Jenkins!" is displayed. Underneath, there is a text input field containing the username "oralabuser". Below that is a password input field with masked characters "....." and an eye icon to toggle visibility. A blue "Sign in" button is highlighted with a red border. At the bottom, there is a checkbox labeled "Keep me signed in".

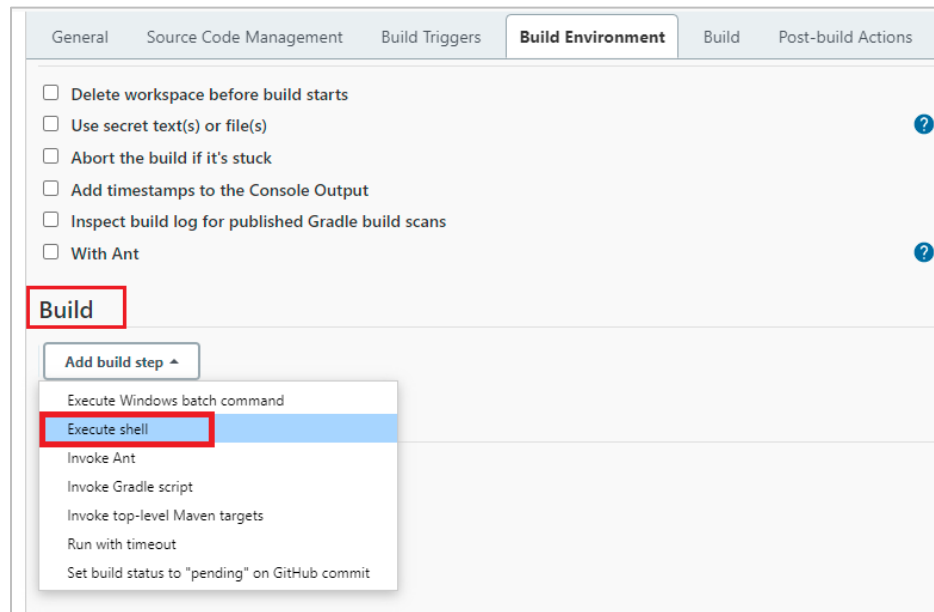
- b. Enter the user name and password provided.
 - c. You will have access to the Jenkins Dashboard.
2. Create Jobs on Jenkins AWS Instance to be triggered automatically.
 - a. In the Jenkins Instance Dashboard, navigate to **Main menu** and select **New Item** to create a Job as shown below.



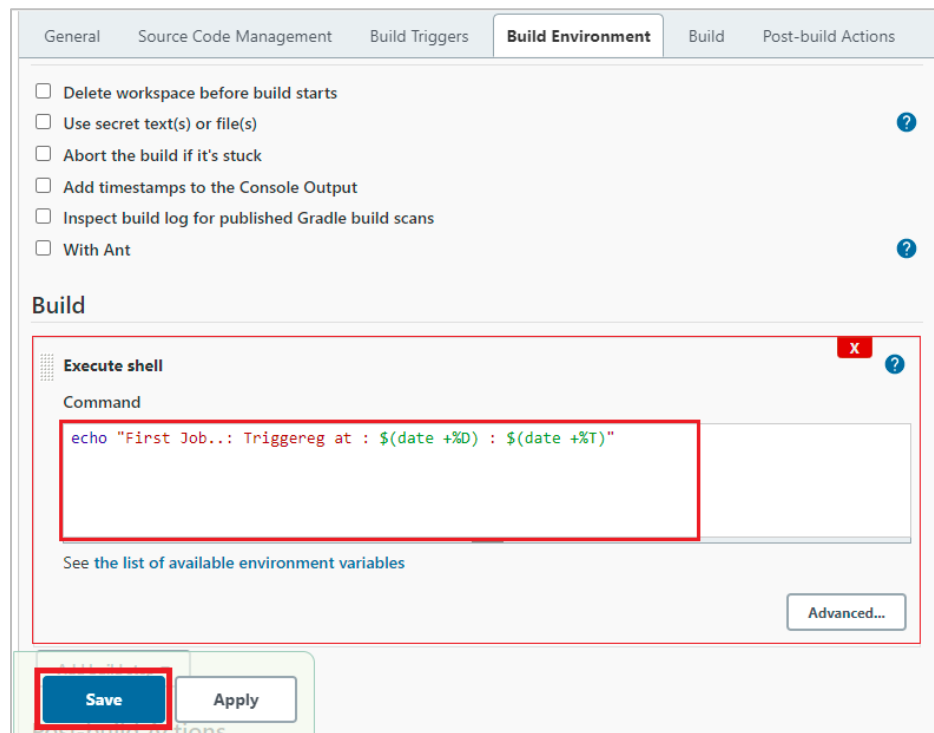
- b. Create **Job_1** as the initial Job to be trigger in the Jenkins.
 - i. Provide the name for the first Job, select **Freestyle project** and click **OK**.

The screenshot shows the 'Enter an item name' form in Jenkins. The 'Job_1' text is entered in the name field and highlighted with a red box. Below the name field, there are three options: 'Freestyle project', 'Pipeline', and 'External Job'. The 'Freestyle project' option is selected and highlighted with a red box. At the bottom of the form, there is an 'OK' button, which is also highlighted with a red box.

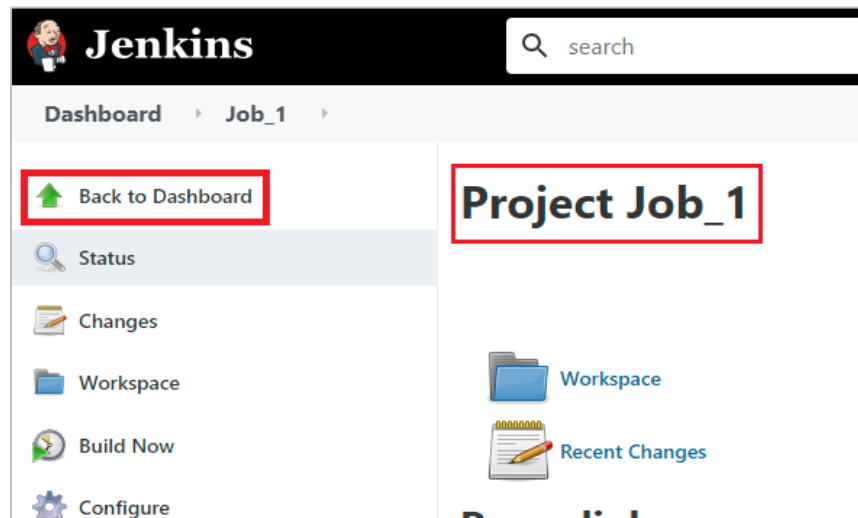
- ii. Scroll down to **Build**, click **Add build step** and select **Execute shell** as shown below.



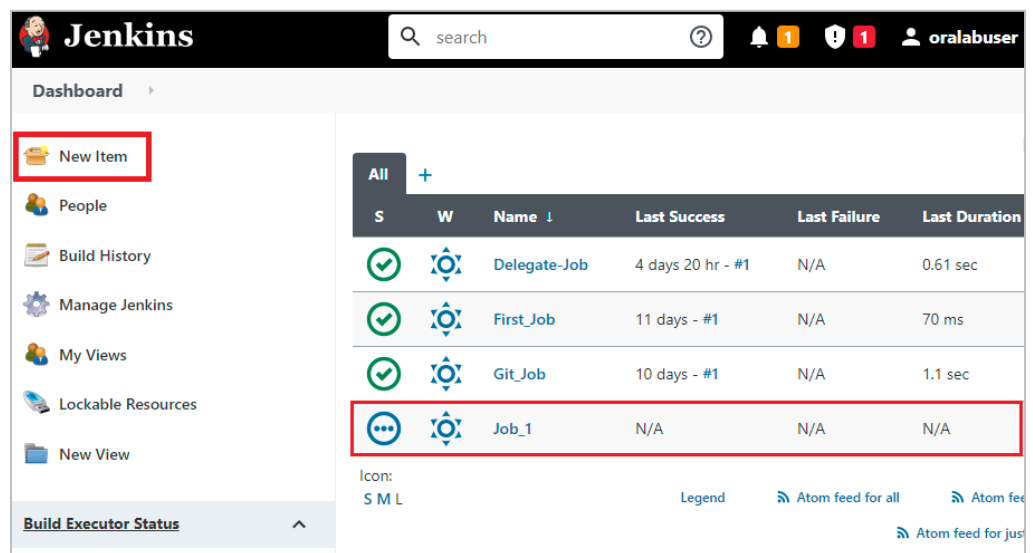
- iii. Type the **echo** command as shown below in the **Command** box and click **Save**.



- iv. As shown below, the Job_1 is created successfully. Click **Back to Dashboard** to return to main page.



- c. Create **Job_2** as second Job to be trigger in Jenkins pipeline.
 - i. Navigate to main menu, click **New Item** to create second job as shown below.




- ii. Provide the name for the second Job, select **Freestyle project** and click **OK**.


Enter an item name

Job_2


» 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.



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



Configuration project
Projects that need a large number of different configurations, such as testing on multiple platforms, operating systems, etc.

OK

- iii. Scroll down to **Build**, click **Add build step** and select **Execute shell**. Type the **echo** command as shown below and click **Save**.

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

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ Inspect build log for published Gradle build scans

☐ With Ant

Build

Execute shell

Command

```
echo "Second Job... Triggered at : $(date +%D) : $(date +%T)"
```

[See the list of available environment variables](#)

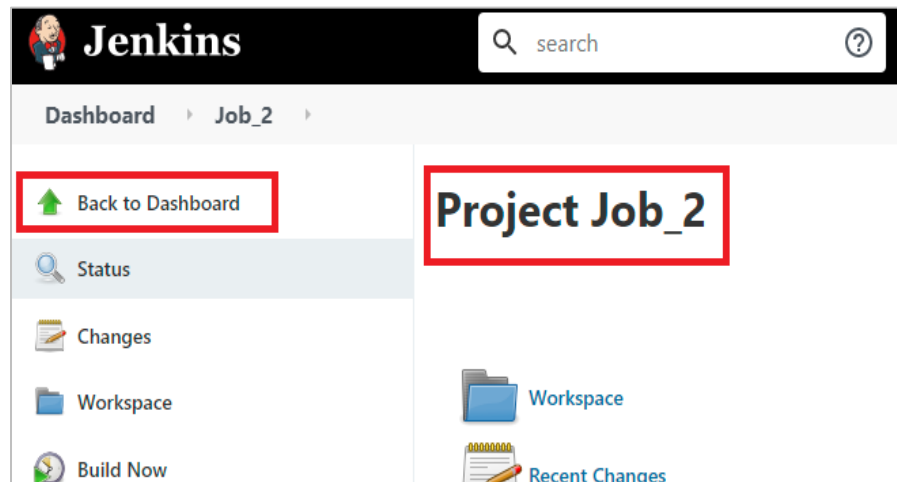
Advanced...

Add build step ▾

Post-build Actions

Save Apply

- iv. As shown below, the **Job_2** is created successfully. Click **Back to Dashboard**.



- d. Create **Job_3** to be triggered automatically in the Jenkins pipeline.
- i. Navigate to main menu, click **New Item** to create third job. Provide the name for the third Job, select **Freestyle project** and click **OK**.

- ii. Scroll down to **Build**, click **Add build step** and select **Execute shell**. Type the **echo** command as shown below and click **Save**.

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

☐ Add timestamps to the Console Output
☐ Inspect build log for published Gradle build scans
☐ With Ant

Build

Execute shell

Command

```
echo "Third Job... Triggered at : ${date +%D} : ${date +%T}"
```

See [the list of available environment variables](#)

Advanced...

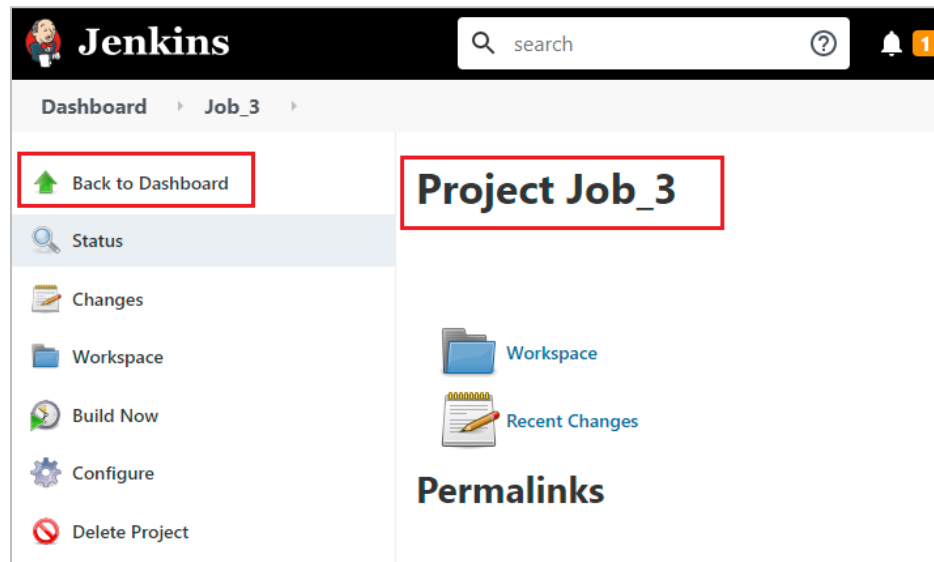
Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

- iii. As shown below, the **Job_3** is created successfully. Click **Back to Dashboard**.



3. Configure the Jenkins Jobs to Trigger automatically in the Jenkins instance.
 - a. Click on the arrow next to the **Job_1** as shown below and select **Configure** to configure first job.

The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with navigation links: New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. Below this is the 'Build Executor Status' section showing two idle executors. The main area displays a table of jobs:

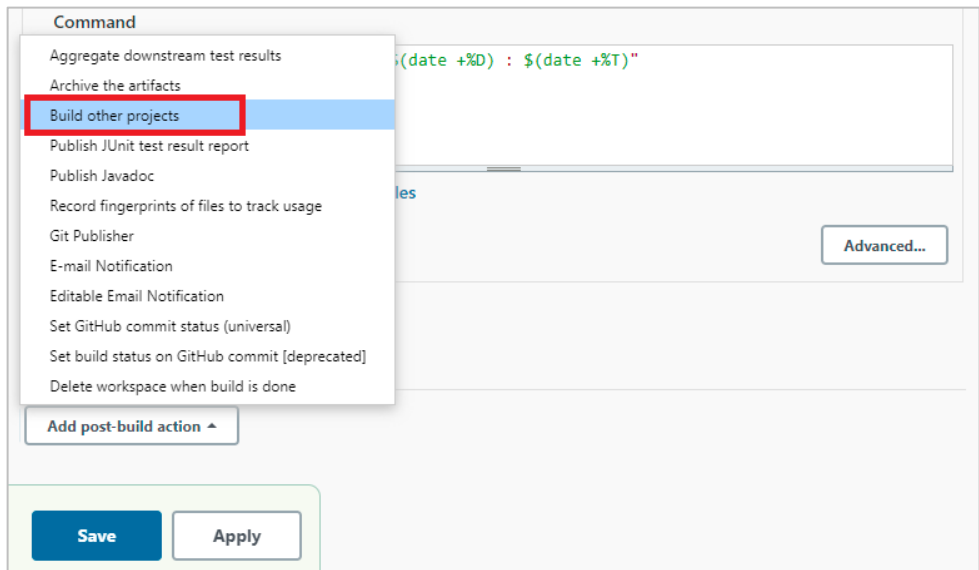
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	⚙️	Delegate-Job	4 days 20 hr - #1	N/A	0.61 sec
✓	⚙️	First_Job	11 days - #1	N/A	70 ms
✓	⚙️	Git_Job	10 days - #1	N/A	1.1 sec
⋮	⚙️	Job_1	N/A	N/A	N/A
⋮	⚙️		N/A	N/A	N/A
⋮	⚙️		N/A	N/A	N/A

A context menu is open for 'Job_1', showing options: Changes, Workspace, Build Now, **Configure** (highlighted), Delete Project, and Rename.

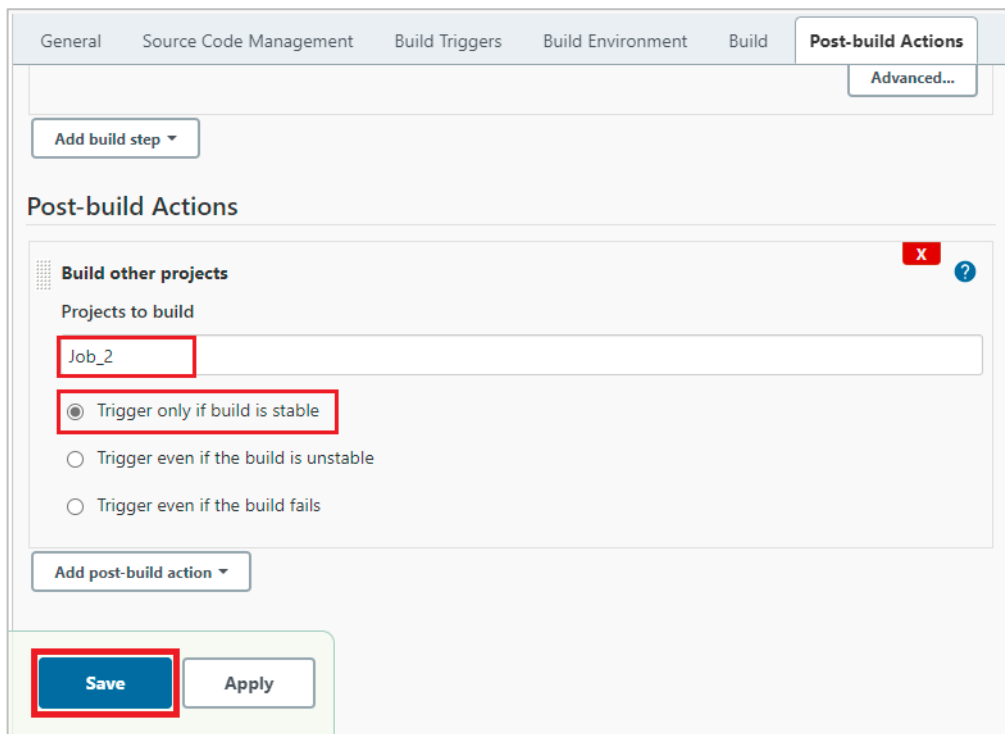
- b. Scroll down to **Post-Build Actions** and click on **Add post-build actions** to view the list of actions available as shown below.

The screenshot shows the Jenkins Job Configuration page for 'Job_1'. The 'Post-build Actions' tab is selected. The 'Command' section contains the text: `echo "First Job...: Triggereg at : $(date +%D) : $(date +%T)"`. Below this is a link to 'See the list of available environment variables' and an 'Advanced...' button. The 'Add build step' button is visible. In the 'Post-build Actions' section, the 'Add post-build action' button is highlighted. At the bottom, there are 'Save' and 'Apply' buttons.

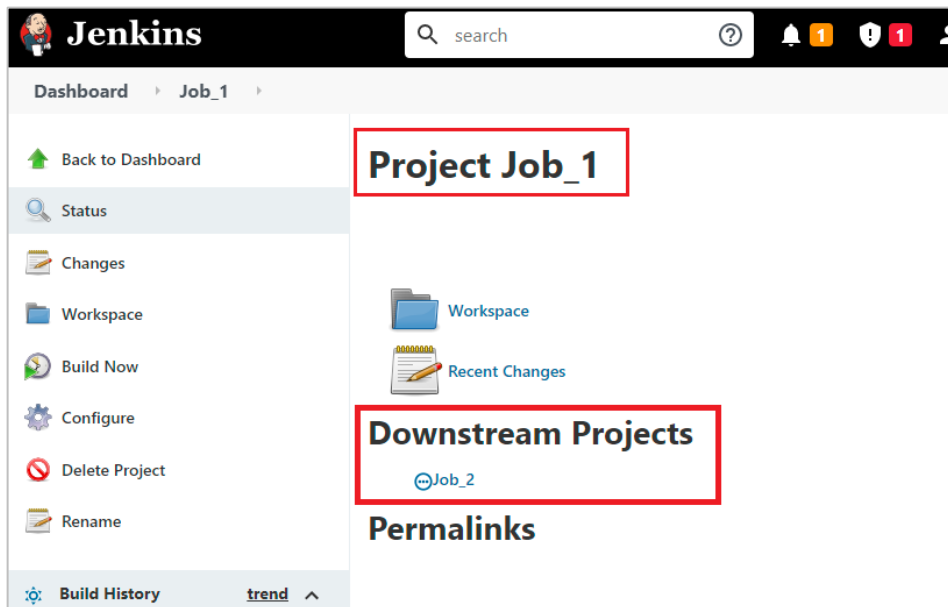
- c. Select **Build other projects** from the list of post-build actions as shown below.



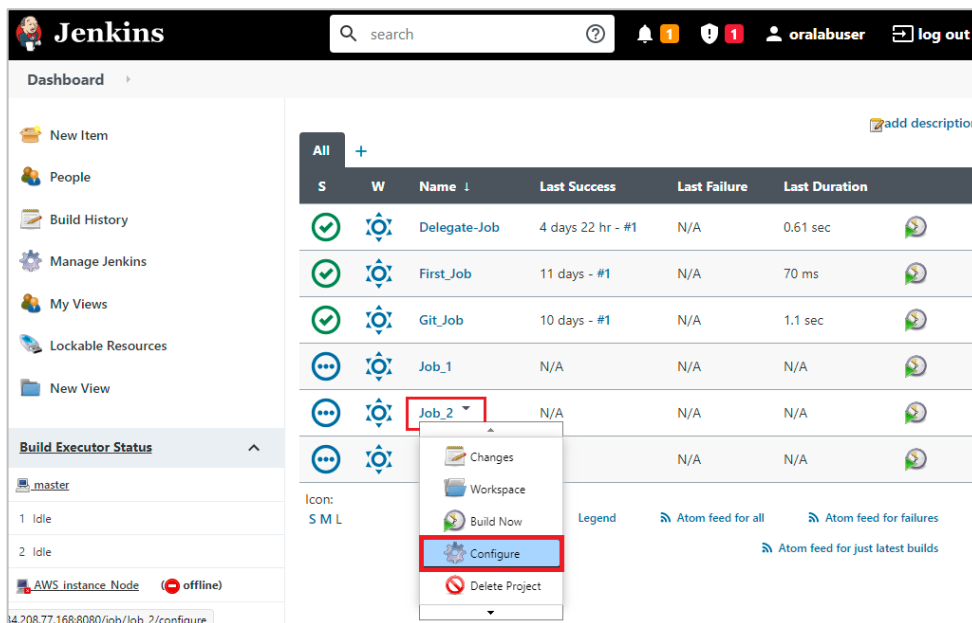
- d. In **Build other projects**, provide name of the second job in the **Projects to build**, select **Trigger only if build is stable** and click **Save**.



- e. Verify the post-build action **Downstream Projects** is successfully updated to the **Job_1** as shown below.



- f. Similarly, update the post-build action to Job_2. Click **Job_2** as shown below and select **Configure** to configure second job.



- g. Scroll down to **Post-Build** Actions and select **Build other projects**. Provide name of the third job in the **Projects to build**, select **Trigger only if build is stable** and click **Save**.

Post-build Actions

Build other projects

Projects to build

Job_3,

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

Add post-build action ▾

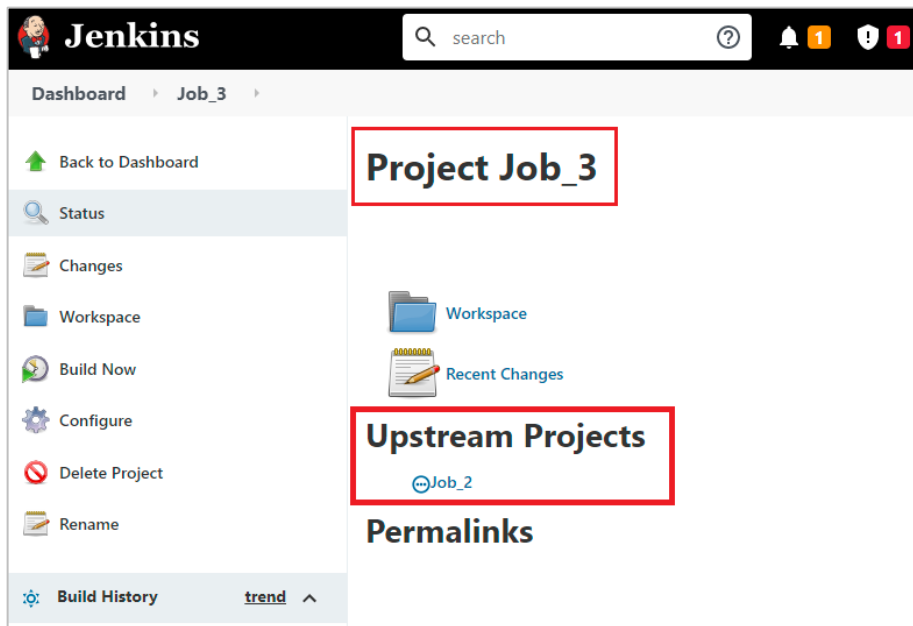
Save

Apply

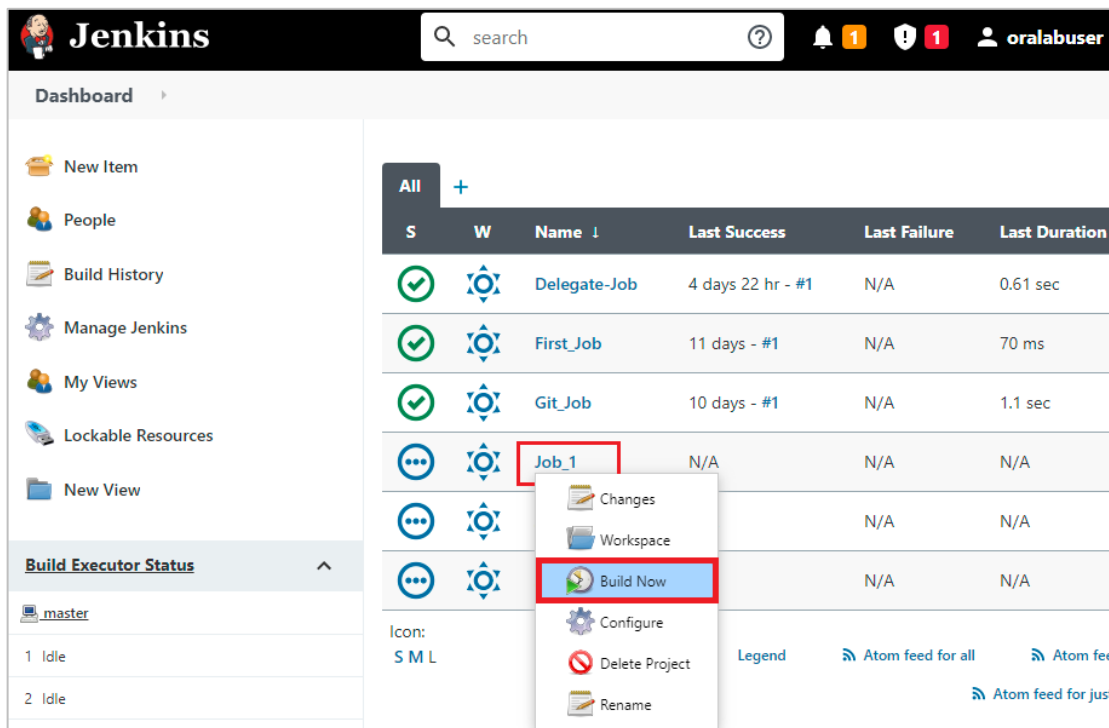
- h. Verify the post-build action **Upstream Projects** and **Downstream Projects** that are successfully updated to the **Job_2** as shown below.

The screenshot shows the Jenkins interface for Job_2. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area is titled 'Project Job_2' and contains sections for 'Workspace', 'Recent Changes', 'Upstream Projects' (listing Job_1), 'Downstream Projects' (listing Job_3), and 'Permalinks'. A search bar is visible at the top right of the main content area.

- i. Refresh the Jenkins Dashboard page, navigate to **Job_3** and view the update of **Upstream projects** as shown below.



4. Trigger the first job in the pipeline of Jenkins Jobs.
 - a. Click on the **Job_1** and select **Build Now** as shown below to execute the first job. On success, it triggers the second job.



- b. In the left bottom corner verify the **Build Queue**, which consists of the **Job_2** in the queue to execute as shown below.

Dashboard

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Build Queue (1)

Job_2

Build Executor Status

All	S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	⚙️		Delegate-Job	4 days 22 hr - #1	N/A	0.61 sec
✓	⚙️		First_Job	11 days - #1	N/A	70 ms
✓	⚙️		Git_Job	10 days - #1	N/A	1.1 sec
✓	⚙️		Job_1	3.8 sec - #2	N/A	28 ms
⋮	⚙️		Job_2	N/A	N/A	N/A
⋮	⚙️		Job_3	N/A	N/A	N/A

Icon: S M L Legend Atom feed for all Atom feed for failures Atom feed for just latest builds

- c. Click on the **Job_1** to view the **Build History**. Click on **Build History** link to view the console output.

Jenkins

search

Dashboard > **Job_1**

Back to Dashboard

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend

find

#1 Apr 20, 2021 9:52 AM

Project Job_1

Workspace

Recent Changes

Downstream Projects

Job_2

Permalinks

- Last build (#2), 24 sec ago
- Last stable build (#2), 24 sec ago
- Last successful build (#2), 24 sec ago
- Last completed build (#2), 24 sec ago

- d. Select **Console Output** to view the output of **Job_1** as shown below, on successful execution of the **Job_1** it triggers **Job_2**. Click on **Job_2** to view the execution of job.

Dashboard > Job_1 > #2

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#2'

Console Output

Started by user [oralabuser](#)
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/Job_1
[Job_1] \$ /bin/sh -xe /tmp/jenkins4216578693080654623.sh
++ date +%D
++ date +%T
+ echo 'First Job... Triggereg at : 04/20/21 : 09:52:20'
First Job... Triggereg at : 04/20/21 : 09:52:20
Triggering a new build of [Job_2](#)
Finished: SUCCESS

- e. Navigates to Job_2 page as shown below, click on the **Build History** link to view the console output.

Dashboard > Job_2 >

Back to Dashboard

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend ^

find

#1 Apr 20, 2021 9:52 AM

Project Job_2

Workspace

Recent Changes

Upstream Projects

Job_1

Downstream Projects

Job_3

Permalinks

- Last build (#1), 1 min 9 sec ago
- Last stable build (#1), 1 min 9 sec ago
- Last successful build (#1), 1 min 9 sec ago

- f. Select **Console Output** to view the output of **Job_2** as shown below. On successful execution of the **Job_2**, it triggers **Job_3**. Click on **Job_3** to view the execution of job.

Dashboard ▸ Job_2 ▸ #1

Back to Project
 Status
 Changes
Console Output
 View as plain text
 Edit Build Information
 Delete build '#1'

Console Output

Started by upstream project "Job_1" build number 2 originally caused by:
 Started by user [oralabuser](#)
 Running as SYSTEM
 Building on master in workspace /var/lib/jenkins/workspace/Job_2
 [Job_2] \$ /bin/sh -xe /tmp/jenkins4208060672679394564.sh
 ++ date +%D
 ++ date +%T
 + echo 'Second Job... Triggered at : 04/20/21 : 09:52:27'
 Second Job... Triggered at : 04/20/21 : 09:52:27
 Triggering a new build of [Job_3](#)
 Finished: SUCCESS

- g. Navigates to **Job_3** page as shown below, click on the **Build History** link to view the console output.

Dashboard ▸ Job_3 ▸

Back to Dashboard
 Status
 Changes
 Workspace
 Build Now
 Configure
 Delete Project
 Rename

Build History trend ^

 #1 Apr 20, 2021 9:52 AM

Project Job_3

Workspace
 Recent Changes

Upstream Projects

Job_2

Permalinks

- Last build (#1), 1 min 52 sec ago
- Last stable build (#1), 1 min 52 sec ago
- Last successful build (#1), 1 min 52 sec ago
- Last completed build (#1), 1 min 52 sec ago

- h. Select **Console Output** to view the output of **Job_3** as shown below.

Dashboard
Job_3
#1

Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'

Console Output

Started by upstream project "Job_2" build number 1 originally caused by:
Started by upstream project "Job_1" build number 2 originally caused by:
Started by user oralabuser
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/Job_3
[Job_3] \$ /bin/sh -xe /tmp/jenkins7740543922560853039.sh
++ date +%D
++ date +%T
+ echo 'Third Job..: Triggered at : 04/20/21 : 09:52:37'
Third Job..: Triggered at : 04/20/21 : 09:52:37
Finished: SUCCESS

5. Visualize the holistic picture of the Jobs triggered in sequence.
 - a. In the Jenkins Instance Dashboard, navigate to **Main menu** and select **Manage Plugins** as shown below.

Dashboard

New Item
People
Build History
Manage Jenkins
My Views
Lockable Resources
New View
Build Executor Status

Manage Jenkins

New version of Jenkins (2.288) 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. Select **Available**, search for **Delivery Pipeline** and select the check box of Delivery Pipeline. Click **Install without restart**.

Copyright © 2021, Proton Expert Systems & Solutions. All rights reserved.

18

Practice for Lesson 4: Integrating Jenkins Pipelines with Jobs

Search:

[Updates](#)
[Available](#)
[Installed](#)
[Advanced](#)

Install ↑	Name	Version	Released
<input checked="" type="checkbox"/>	Delivery Pipeline User Interface This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)	1.4.2	1 yr 1 mo ago

[Install without restart](#)
[Download now and install after restart](#)
 Update information obtained: 23 hr ago

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

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Role-based Authorization Strategy	✓ Success
Loading plugin extensions	✓ Success
WMI Windows Agents	✓ Success
Javadoc	✓ Success
External Monitor Job Type	✓ Success
Backup	✓ Success
Loading plugin extensions	✓ Success
Run Condition	✓ Success
Maven Integration	✓ Success
Conditional BuildStep	✓ Success
Parameterized Trigger	✓ Success
jQuery	✓ Success
Delivery Pipeline	✓ Success
Loading plugin extensions	✓ Success

[Go back to the top page](#)
 (you can start using the installed plugins right away)

☐ Restart Jenkins when installation is complete and no jobs are running

- d. Click on **+** as shown below, to create the new view.

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

All

+

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	⚙	Delegate-Job	5 days 0 hr - #1	N/A	0.61 sec
✓	⚙	First_Job	11 days - #1	N/A	70 ms
✓	⚙	Git_Job	11 days - #1	N/A	1.1 sec
✓	⚙	Job_1	1 hr 59 min - #2	N/A	28 ms
✓	⚙	Job_2	1 hr 59 min - #1	N/A	15 ms

e. Provide the name for the view, select **Delivery Pipeline View** and click **OK**.

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Executor Status

master

1 Idle

View name

first_visualization

☒ **Delivery Pipeline View**
 Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or more delivery pipeline instances, based on traditional Jenkins jobs with upstream/downstream dependencies.

☐ **Delivery Pipeline View for Jenkins Pipelines**
 Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or more delivery pipeline instances, based on Jenkins pipelines (created using the Pipeline or Workflow plugin).

☐ **List View**
 Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

☐ **My View**
 This view automatically displays all the jobs that the current user has an access to.

OK

f. Scroll down to **Pipelines**, click **Add** under **Components** as shown below.

URL for custom CSS file ?

URL for custom CSS file (fullscreen) ?

☐ Enable paging ?

Pipelines

Components

Add

Regular Expression

Add

OK **Apply**

g. Provide the name for the **Component** and the **initial Job** name and click **OK**.

☐ Enable paging ?

Pipelines

Components

Component

Name ?

sample_Job

Initial Job ?

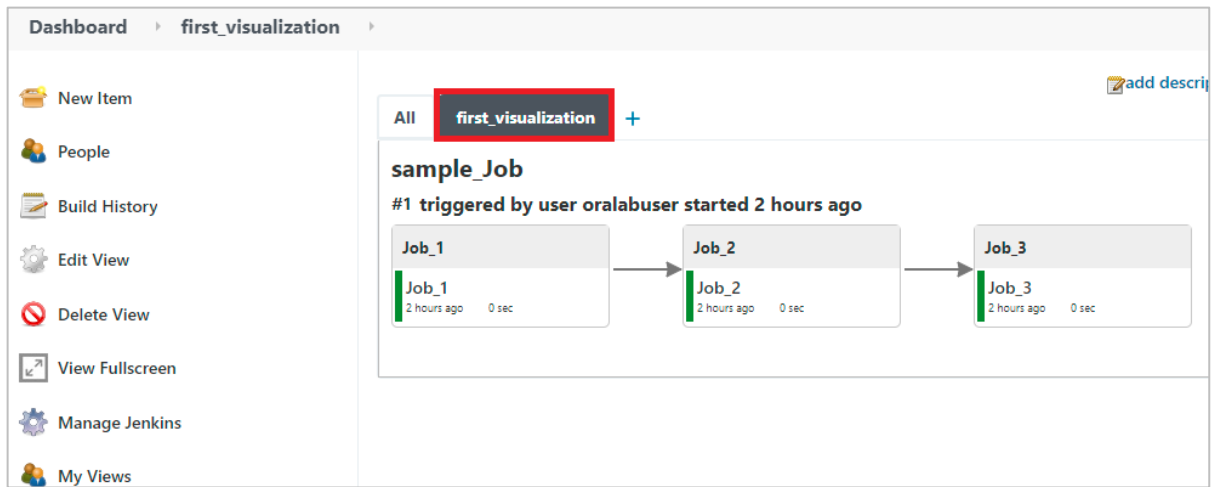
Job_1

Final Job (optional) ?

☐ Show upstream

OK **Apply** **Delete**

h. Verify the visualization as shown below with the view of the Jobs triggering.



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