

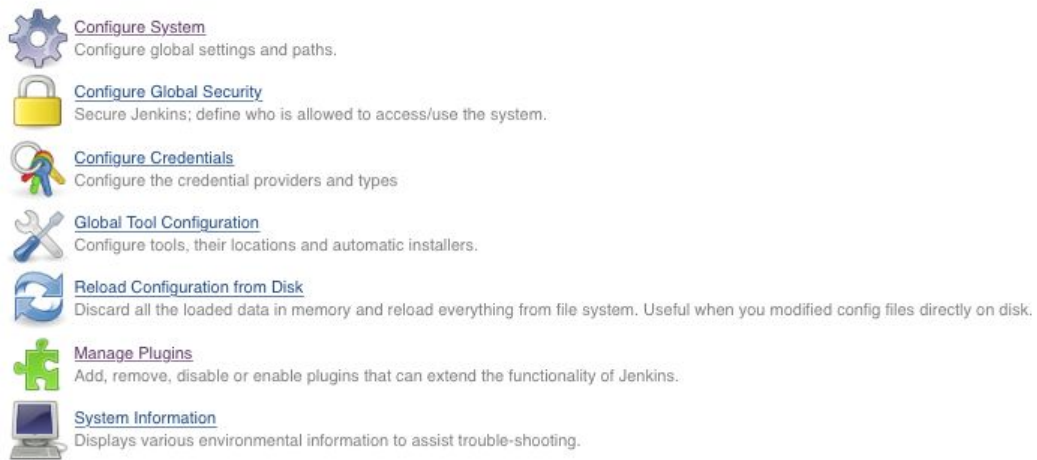
## Chapter 2

### Lab 2.4

**Objective:** Create a pipeline that runs multiple jobs, and display the process with the `pipeline-stage-view` plugin

1. To ensure the correct plugin is installed to visualize the pipeline stages, navigate to “Manage Jenkins” on the left hand menu of the console home page, and then select “Manage Plugins”.

#### Manage Jenkins



On the “Installed” tab, search for “Pipeline: Stage View Plugin” and ensure it is installed. If not, install it from the “Available” tab.

Updates Available Installed Advanced					
Enabled	Name ↓	Version	Previously installed version	Pinned	Uninstall
<input checked="" type="checkbox"/>	<a href="#">bouncycastle API Plugin</a> Provides an stable API to Bouncy Castle related tasks. Plugins using Bouncy Castle should depend on this plugin and not directly on Bouncy Castle	<a href="#">2.16.0</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">JavaScript GUI Lib: Handlebars bundle plugin</a> JavaScript GUI Lib: Handlebars bundle plugin.	<a href="#">1.1.1</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) plugin</a> JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) plugin.	<a href="#">1.2.1</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">JavaScript GUI Lib: Moment.js bundle plugin</a> JavaScript GUI Lib: Moment.js bundle plugin.	<a href="#">1.1.1</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Pipeline</a> A suite of plugins that lets you orchestrate automation, simple or complex. See <a href="#">Pipeline as Code with Jenkins</a> for more details.	<a href="#">2.4</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Pipeline: Job</a> Defines a new job type for pipelines and provides their generic user interface.	<a href="#">2.9</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Pipeline: REST API Plugin</a> Provides a REST API to access pipeline and pipeline run data.	<a href="#">2.4</a>			<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Pipeline: Stage View Plugin</a> Pipeline Stage View Plugin.	<a href="#">2.4</a>			<a href="#">Uninstall</a>

- In order to have multiple stages for the pipeline to trigger, a new project will be created. Select the “New Item” option from the left hand menu of the console home page.

Next, name this new item “secondary job”, and select “Freestyle project” as the project type.

Enter an item name
= 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 slaves. 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.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If you want to create a new item from other existing, you can use this option:

Copy from

Add an “Execute shell” build step, and enter the following commands into the “Execute” field:

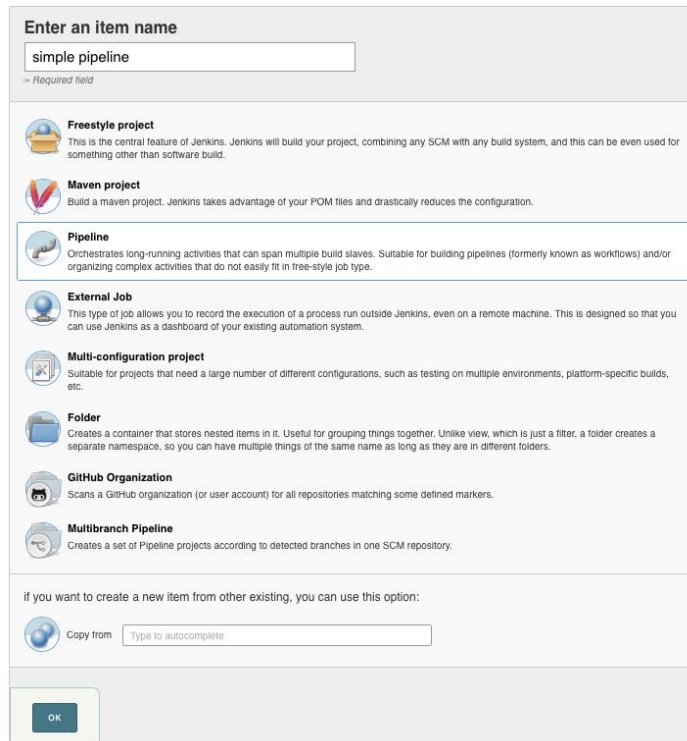
```
$ #!/bin/bash
$ echo The secondary job was triggered!
```



Save the project and return to the console home screen.

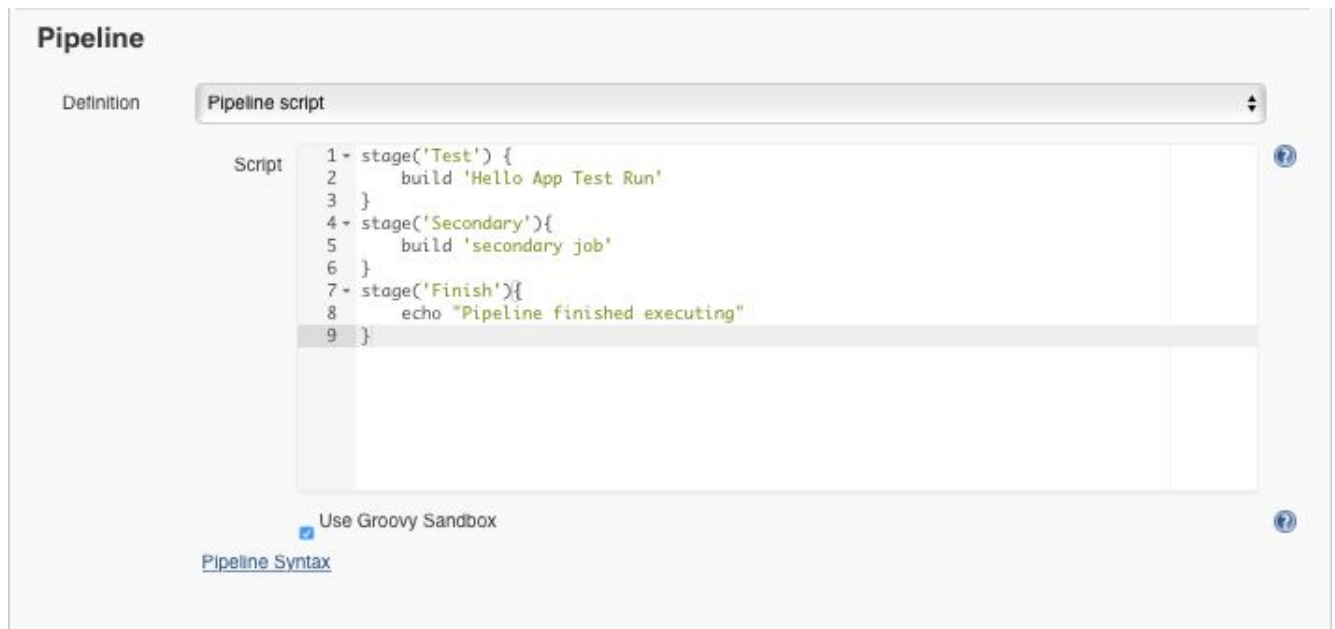
3. To build a pipeline with code from a **Jenkinsfile**, create another new project.

Name the new project “simple pipeline”, and select “Pipeline” as the project type.



In the “Pipeline” section add the following:

```
$ stage('Test') {  
    build 'Hello-App-Test-Run'  
}  
$ stage('Secondary'){  
    build 'secondary-job'  
}  
$ stage('Finish'){  
    echo "Pipeline finished executing"  
}
```



Save the job and return to the Jenkins console home page.

4. Click on the “simple pipeline” job view.

Trigger the “simple pipeline” job by clicking the “Build Now” option in the left hand menu.

-  [Back to Dashboard](#)
-  [Status](#)
-  [Changes](#)
-  [Build Now](#)
-  [Delete Pipeline](#)
-  [Configure](#)
-  [Move](#)
-  [Full Stage View](#)
-  [Pipeline Syntax](#)

## Pipeline simple pipeline



[Recent Changes](#)

### Stage View

No data available. This Pipeline has not yet run.

The pipeline job will trigger each build in the pipeline section of the “simple pipeline” job. It is possible to trigger jobs that execute specific actions, or have the pipeline job execute actions itself, as demonstrated with both the **secondary** job, as well as the **simple pipeline echo** commands.

-  [Back to Dashboard](#)
-  [Status](#)
-  [Changes](#)
-  [Build Now](#)
-  [Delete Pipeline](#)
-  [Configure](#)
-  [Move](#)
-  [Full Stage View](#)
-  [Pipeline Syntax](#)

## Pipeline simple pipeline



[Recent Changes](#)

### Stage View

Average stage times:

Test	Secondary	Finish
13s	5s	49ms
13s	5s	49ms

Build History

find

X

#1

Dec 19, 2016 11:41 PM

RSS for all

RSS for failures

### Permalinks

- [Last build \(#1\), 2 min 31 sec ago](#)
- [Last stable build \(#1\), 2 min 31 sec ago](#)
- [Last successful build \(#1\), 2 min 31 sec ago](#)
- [Last completed build \(#1\), 2 min 31 sec ago](#)

## Jenkins Pipeline Example:

[Jenkins Pipeline Example](#) will help you to learn more about simple Jenkins Pipelines. Please link to this exercise before you move to the next section.

You can also read how to [setup and visualize code delivery pipelines](#) using Blue Ocean UI.