

JENKINS

CI/CD (Continuous Integration & Development)

Continuous Integration is a development practice that requires developers to commit the code into a repository at regular intervals so that code will be available to other developer as earliest. This will remove the last minute occurrence of issues in the build lifecycle. Continuous integration requires the developers to have frequent builds. The common practice is that whenever a code commit occurs, a build should be triggered.

CI/CD SCOPE

Required Tools for CI/CD: Jenkins

CI/CD Scope Diagram

What is Jenkins?

Jenkins is a popular tool for performing continuous integration of software projects. It is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software.

Why Jenkins?

Jenkins will be installed on a server where the central build will take place. The following flowchart demonstrates a very simple workflow of how Jenkins works.

Jenkins setup and Installation

Installation

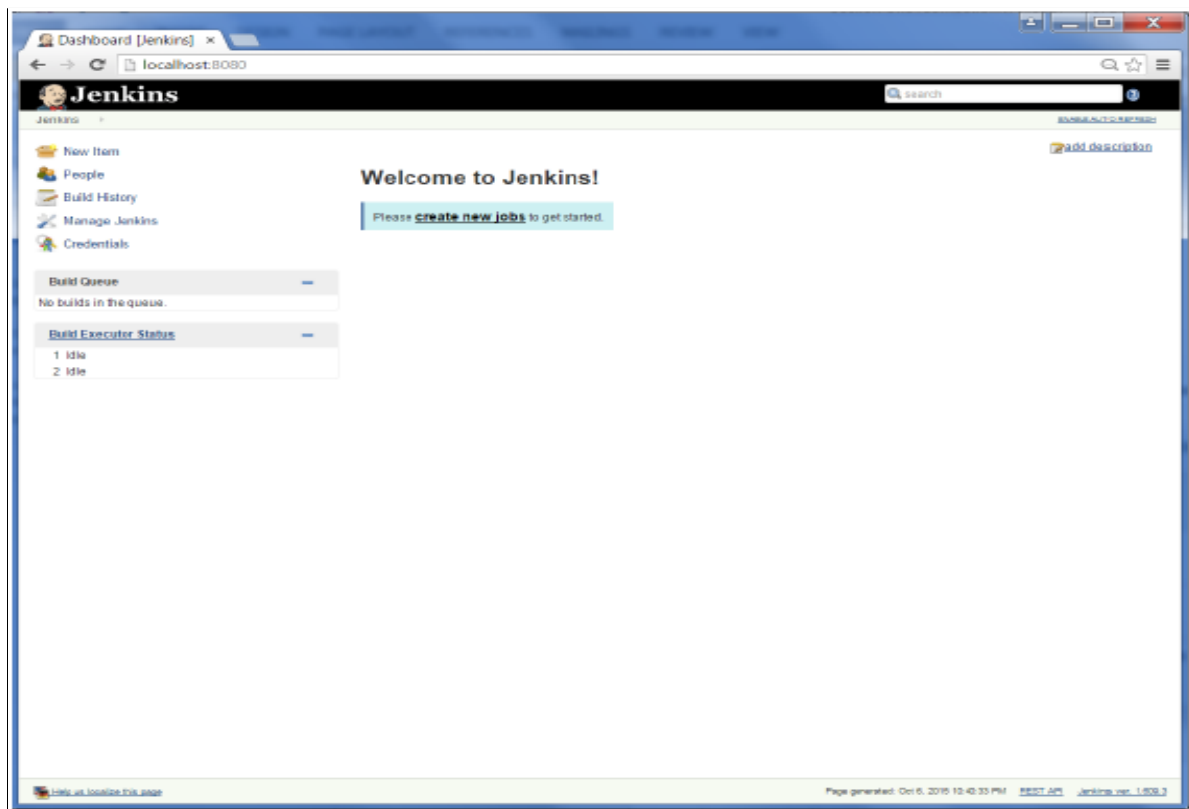
1. Go to the official website of Jenkins to download the latest version of Jenkins. The download is available for different operating systems.

URL: <https://jenkins.io/download/>

2. Once the download is complete, install the software into your system.

3. Once the software is installed and it is been properly running one can access the Jenkins from the given link <http://localhost:8080>

This will navigate you to the home page of Jenkins.



In order to create a new job click on the create new job link.

1. Enter the name for the job.
2. Select the Freestyle Project.
3. Click on the OK button.

localhost:8080/view/all/newJob

Jenkins







4

search admin log out

Enter an item name

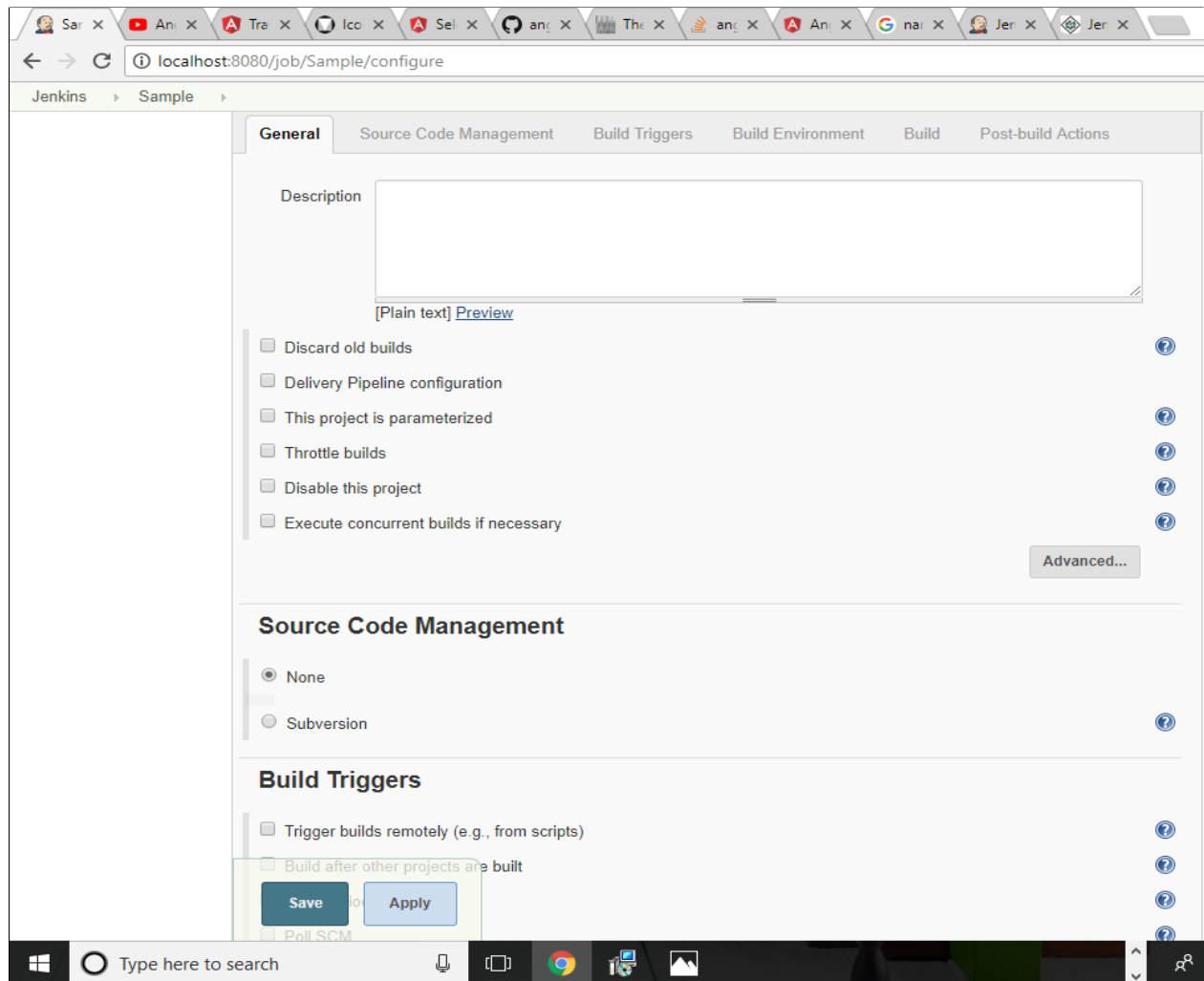
Sample

» Required field

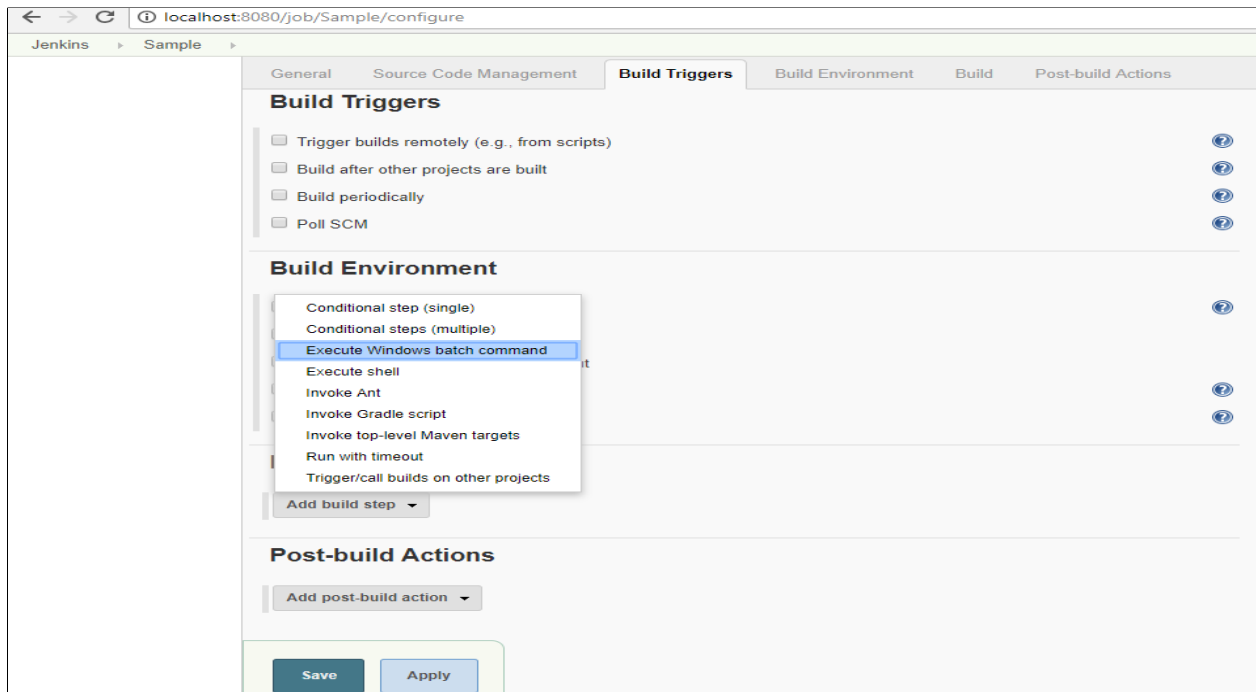
-  **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
-  **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically 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 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, folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

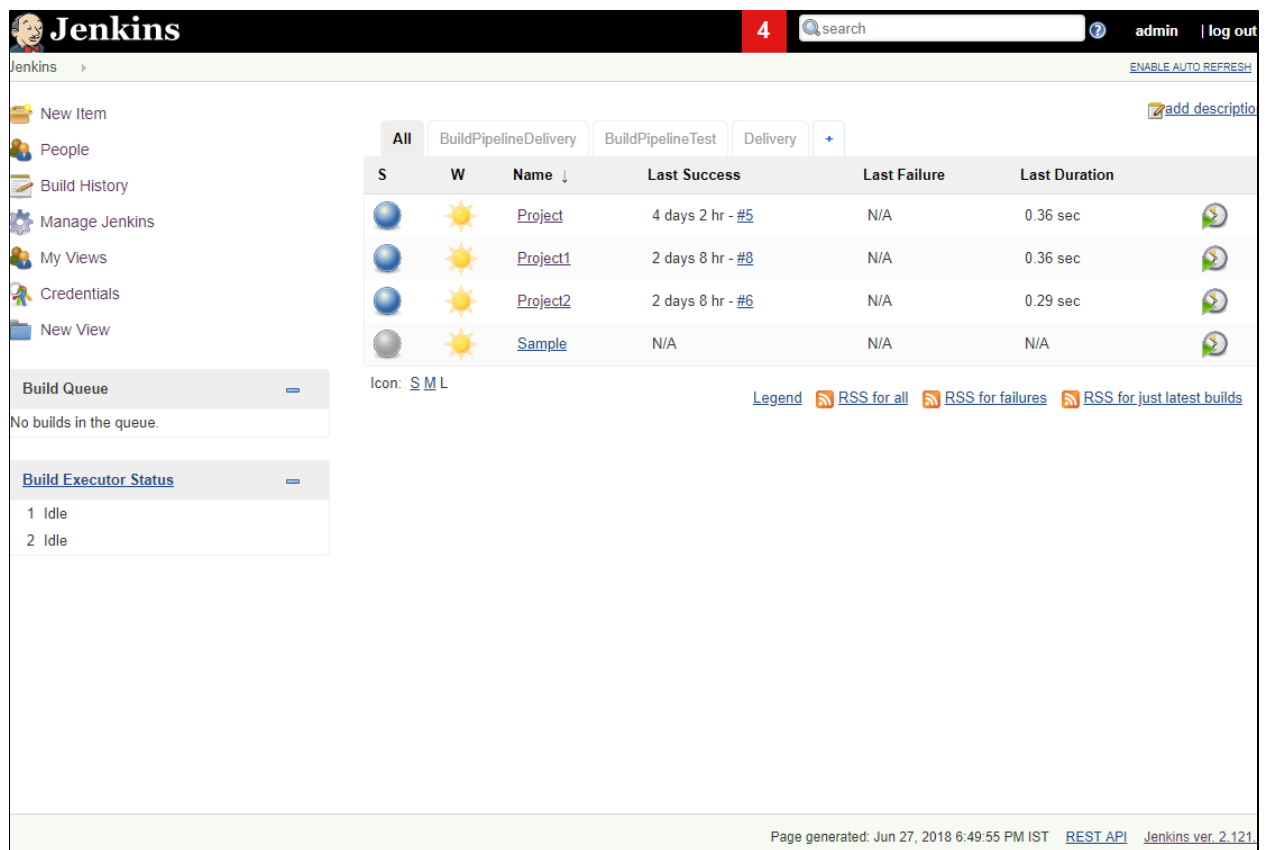
Once the job is been successfully created you will be navigated to the given page:



1. In the description tab give a description of your project.
2. Go to the Build Tab wherein you can see different Add Build Step options.
3. Click on the Apply button and save.



One can see that a job is been created.



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a red tab with the number '4', a search bar, and user information 'admin | log out'. The left sidebar contains navigation links: 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Credentials', and 'New View'. The main content area displays a table of jobs under the 'All' filter. The table has columns for status (S), weather icon (W), name, last success, last failure, and last duration. Four jobs are listed: 'Project', 'Project1', 'Project2', and 'Sample'. Below the table, there is a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing '1 Idle' and '2 Idle'. At the bottom, there is a footer with the text 'Page generated: Jun 27, 2018 6:49:55 PM IST' and links for 'REST API' and 'Jenkins ver. 2.121'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Project	4 days 2 hr - #5	N/A	0.36 sec
		Project1	2 days 8 hr - #8	N/A	0.36 sec
		Project2	2 days 8 hr - #6	N/A	0.29 sec
		Sample	N/A	N/A	N/A

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)


Page generated: Jun 27, 2018 6:49:55 PM IST [REST API](#) [Jenkins ver. 2.121](#)

Click on the job and you can see different options in order to build a job click on Build Now option.

The screenshot shows the Jenkins web interface in a browser window. The address bar indicates the URL is localhost:8080. The Jenkins logo and name are at the top left. A search bar and user information (admin | log out) are at the top right. On the left sidebar, there are links for New Item, People, Build History, Manage Jenkins, My Views, Credentials, and New View. Below these are sections for Build Queue (No builds in the queue) and Build Executor Status (2 Idle). The main content area displays a table of jobs. The table has columns for S, W, Name, Last Success, Last Failure, and Last Duration. The jobs listed are Project, Project1, Project2, and Sample. A context menu is open over the 'Sample' job, showing options: Changes, Workspace, Build Now (highlighted), Delete Project, Configure, and Rename. At the bottom, the page generation timestamp is Jun 27, 2018 6:49:55 PM IST, and the Jenkins version is 2.121.1.

S	W	Name	Last Success	Last Failure	Last Duration
		Project	4 days 2 hr - #5	N/A	0.36 sec
		Project1	2 days 8 hr - #8	N/A	0.36 sec
		Project2	2 days 8 hr - #6	N/A	0.29 sec
		Sample	N/A	N/A	N/A

One can see that the job is been build.

**Jenkins**

4

admin | log out

Jenkins >

New Item

People

Build History

Manage Jenkins

My Views

Credentials


New View

Build Queue













No builds in the queue.

Build Executor Status

1 Idle

2  [Sample](#) #1


AllBuildPipelineDeliveryBuildPipelineTestDelivery+

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		Project	4 days 2 hr - #5	N/A	0.36 sec	
		Project1	2 days 8 hr - #8	N/A	0.36 sec	
		Project2	2 days 8 hr - #6	N/A	0.29 sec	
		Sample	N/A	N/A	N/A	

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

ENABLE AUTO REFRESH

 [add description](#)

Page generated: Jun 27, 2018 6:49:55 PM IST RFE API Jenkins ver. 2.121

To see whether the job is created successfully, go to the console output.

localhost:8080/job/Project%20Build/

Jenkins

2

search

admin | log

Jenkins > Project Build >

ENABLE AUTO REFRESH

Project Project Build

[add description](#)

[Disable Project](#)

[Workspace](#)

[Recent Changes](#)

Downstream Projects

[Project Test](#)

Permalinks

- [Last build \(#2\), 28 min ago](#)
- [Last stable build \(#2\), 28 min ago](#)
- [Last successful build \(#2\), 28 min ago](#)
- [Last completed build \(#2\), 28 min ago](#)

Build History

trend

find

#2 Jun 28, 2018 5:53 PM

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Delete Build](#)

localhost:8080/job/Project Build/2/console

Page generated: Jun 28, 2018 6:21:59 PM IST [REST API](#) Jenkins ver. 2.10.1

localhost:8080/job/Project%20Build/2/console

Jenkins

2

search

admin | log

Jenkins > Project Build > #2

Console Output

Started by user [admin](#)

Building in workspace C:\Program Files (x86)\Jenkins\workspace\Project Build

[Project Build] \$ cmd /c call C:\WINDOWS\TEMP\jenkins1976469084283504789.bat

C:\Program Files (x86)\Jenkins\workspace\Project Build>echo build success

build success

C:\Program Files (x86)\Jenkins\workspace\Project Build>exit 0

Triggering a new build of [Project Test](#)

Finished: [SUCCESS](#)

localhost:8080/job/Project Build/2/console

Page generated: Jun 28, 2018 6:23:00 PM IST [REST API](#) Jenkins ver. 2.10.1

Pipelines

What is Jenkins Pipeline?

Jenkins Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

A continuous delivery (CD) pipeline is an automated expression of your process for getting software from version control right through to your users and customers. Every change to your software (committed in source control) goes through a complex process on its way to being released. This process involves building the software in a reliable and repeatable manner, as well as progressing the built software (called a "build") through multiple stages of testing and deployment.

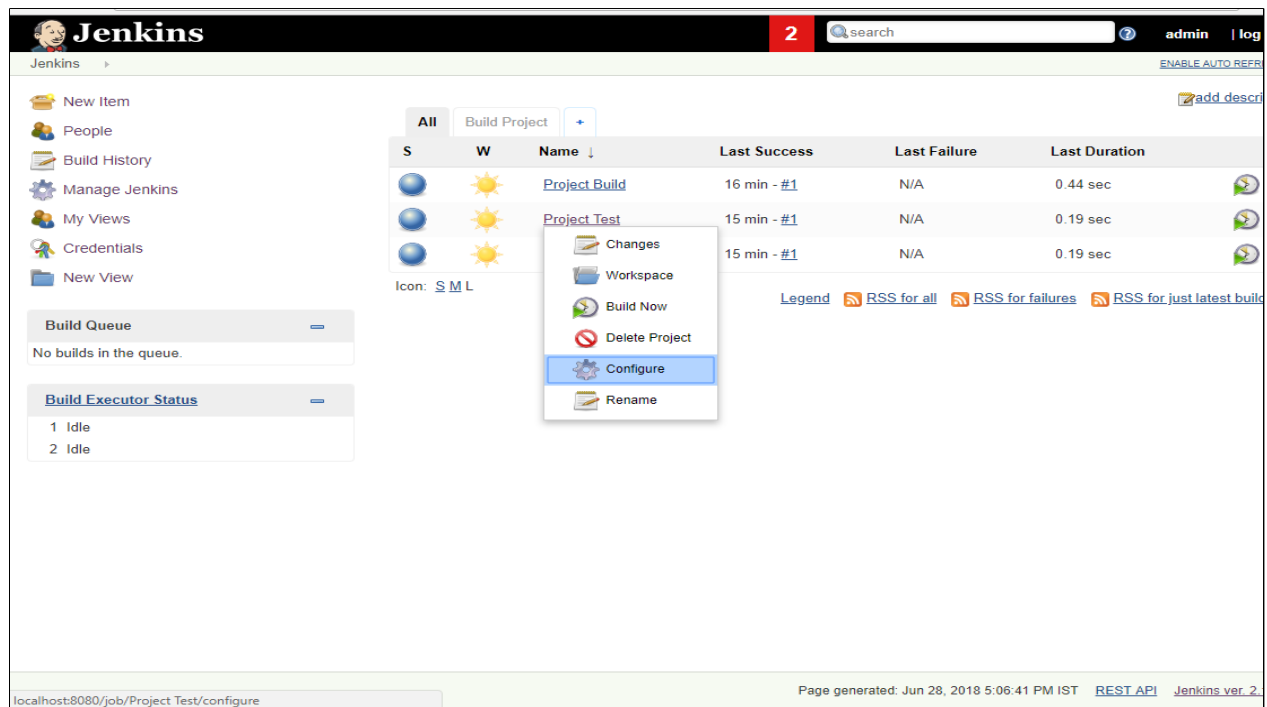
Steps to build a pipeline in Jenkins:

1. One need to add the jobs in the following sequence i.e Build, Test and Deploy.
2. In order to achieve the desired pipeline flow the jobs need to be configured one after the other i.e; one need to chain the sequence of jobs.

As the jobs are already been build all one needs to do is configure them

Following below are the steps to configure the jobs to create a pipeline:

1. Go to the respective job and click on the configure option.



The screenshot displays the Jenkins dashboard. On the left, there is a sidebar with navigation links: New Item, People, Build History, Manage Jenkins, My Views, Credentials, and New View. Below these are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing two 'Idle' executors). The main area shows a table of jobs. A context menu is open over the 'Project Test' job, with the 'Configure' option highlighted. The table has columns for 'S' (status), 'W' (weather icon), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'Project Test' job shows a last success of '15 min - #1' and a last failure of 'N/A'.

S	W	Name	Last Success	Last Failure	Last Duration
		Project Build	16 min - #1	N/A	0.44 sec
		Project Test	15 min - #1	N/A	0.19 sec
		Project Deploy	15 min - #1	N/A	0.19 sec

Legend: RSS for all RSS for failures RSS for just latest build

Page generated: Jun 28, 2018 5:06:41 PM IST [REST API](#) Jenkins ver. 2.10.1

2. Go to the Build Triggers Option and select the option “Build after other projects are build” and specify the name of the project which is to be build after the Build stage. The Build job is the default job one cannot configure the build job after another job. Because it is only after the build stage the job will be moved on to Test and Deploy stage.

The screenshot shows the Jenkins configuration interface for a job named 'Project Test'. The browser address bar indicates the URL is 'localhost:8080/job/Project%20Test/configure'. The configuration page has several tabs: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'Build Triggers' tab is currently selected. Under 'Source Code Management', 'None' is selected. The 'Build Triggers' section includes options for 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built' (which is checked), 'Build periodically', and 'Poll SCM'. The 'Build after other projects are built' option has a 'Projects to watch' field containing 'Project Build,' and three radio button options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. The 'Build Environment' section has options for 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', and 'Create Delivery Pipeline version'. The 'Build' section is partially visible at the bottom, showing a 'Save' button and an 'Apply' button. A red 'X' icon and a help icon are also visible in the bottom right corner of the configuration area.

Similarly we need to configure the Deploy Job.

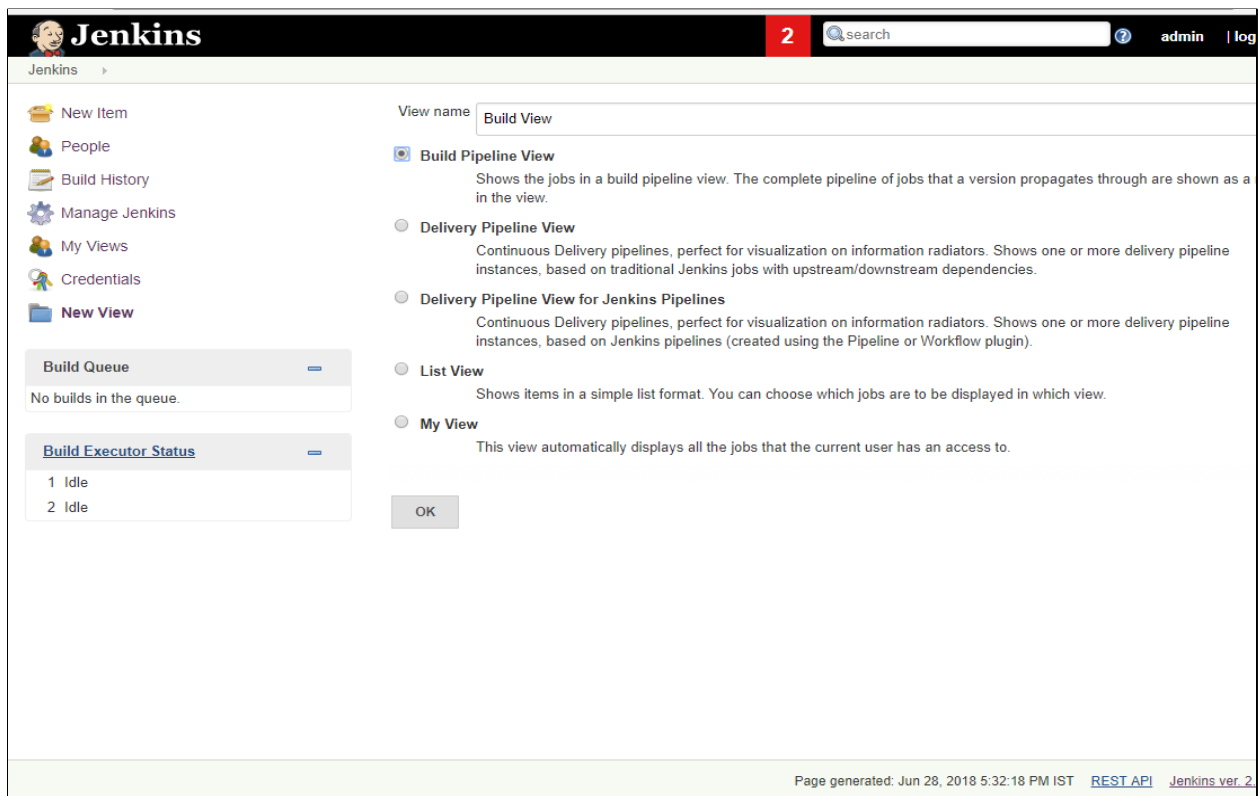
View

In order to display the sequence of jobs and the flow for various kinds of plugin what comes into picture is view. Basically view is used for representation of pipeline.

In order to create a view we need to follow the following steps:

1. Build View

a. On the Jenkins dashboard on the left side of the screen one can see the option of “New View” click on the respective button and you will be navigated to the following page.



The screenshot shows the Jenkins 'New View' configuration page. The top navigation bar includes the Jenkins logo, a red tab labeled '2', a search bar, and links for 'admin' and 'log'. The left sidebar contains a list of navigation items: 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Credentials', and 'New View' (highlighted). Below the sidebar, there are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two idle executors). The main content area is titled 'View name' and 'Build View'. It lists five view options with radio buttons: 'Build Pipeline View' (selected), 'Delivery Pipeline View', 'Delivery Pipeline View for Jenkins Pipelines', 'List View', and 'My View'. Each option has a brief description. An 'OK' button is at the bottom right. The footer indicates the page was generated on Jun 28, 2018 5:32:18 PM IST, with links to the REST API and Jenkins version 2.

Jenkins

2

search

admin | log

Jenkins

New Item

People

Build History

Manage Jenkins

My Views

Credentials

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

View name

Build View

☒ Build Pipeline View

Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a in the view.

☐ 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.

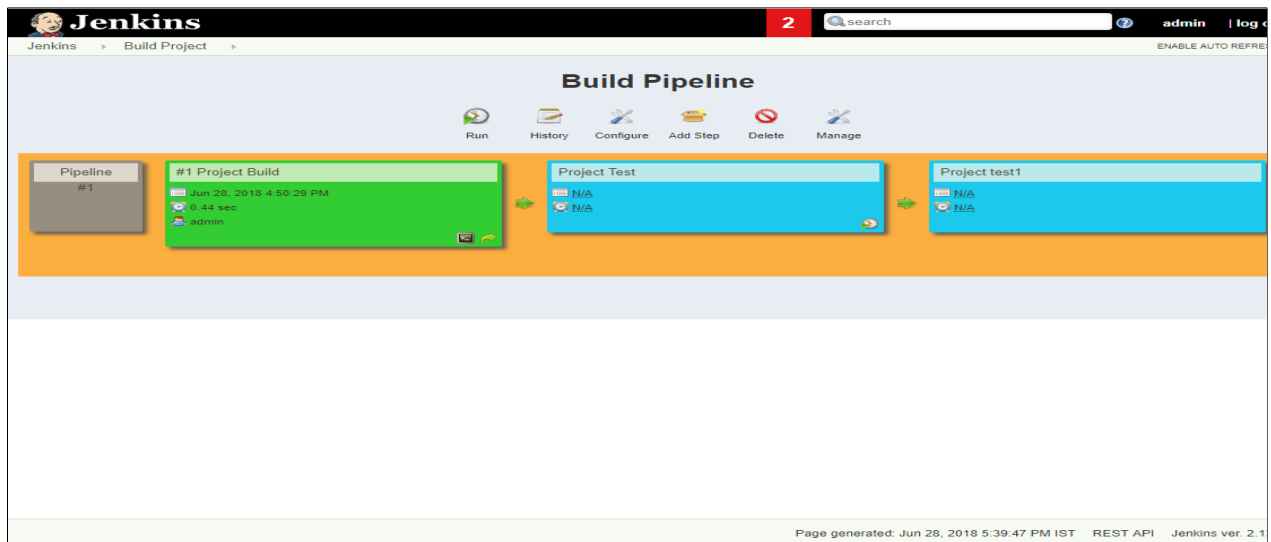
OK

Page generated: Jun 28, 2018 5:32:18 PM IST [REST API](#) [Jenkins ver. 2](#)

After one has created the view, with the help of view we can see the jobs that were been configured in the desired sequence.

In this view one can see the actual flow of the jobs which altogether comprise of a pipeline.

If one wants to check for a particular job and not for the entire pipeline, this can be achieved with the help of “Trigger” option.



3. Delivery Pipeline

Visualization of Continuous Delivery pipelines. Renders pipelines based on upstream/downstream jobs or Jenkins pipelines. Provides a full screen view for information radiators.

In Continuous Delivery feedback and visualization of the delivery process is one of the most important areas. When using Jenkins as a build server it is with the Delivery Pipeline plugin possible to visualize one or more delivery pipelines in the same view, even in full screen.

Same steps needs to be followed in order to create a Delivery Pipeline as it is been followed in Build Pipeline:

1. Set the view name

The screenshot shows the Jenkins 'New View' configuration interface. The 'View name' field is set to 'Delivery'. The 'Delivery Pipeline View' option is selected. The left sidebar shows navigation links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Credentials', and 'New View'. The bottom status bar shows 'Page generated: Jun 28, 2018 5:54:44 PM IST'.

View name:

☐ Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as in the view.

☒ 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

Page generated: Jun 28, 2018 5:54:44 PM IST [REST API](#) [Jenkins ver.](#)

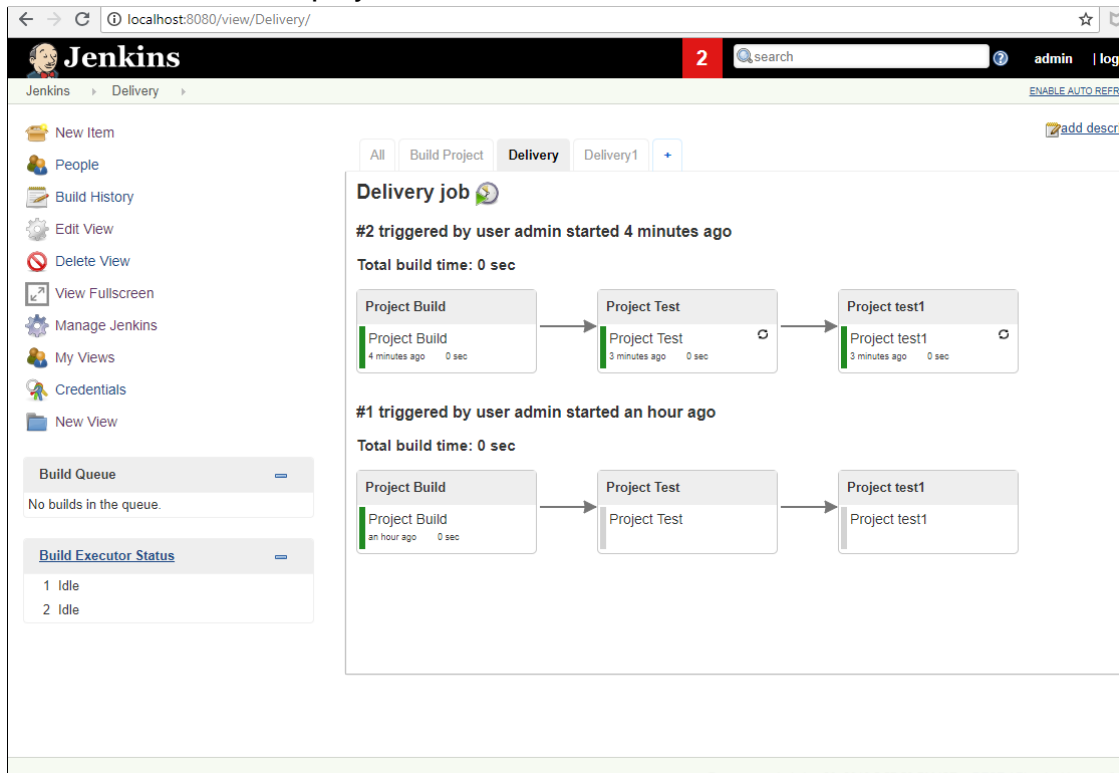
2. Do the configuration settings such as:

The screenshot shows the Jenkins configuration page for a job named 'Delivery1'. The left sidebar contains two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two idle executors). The main configuration area includes the following settings:

- Max number of pipelines: -1
- Update interval: 2
- Enable start of new pipeline build: ☒
- Enable manual triggers: ☒
- Enable rebuild: ☒
- Show avatars: ☐
- Show commit messages: ☐
- Show job description: ☐
- Show job promotions: ☐
- Show test results: ☐
- Show static analysis results: ☐
- Show total build time: ☒
- Use relative links for easier navigation: ☐
- Use direct link to console log: ☐
- URL for custom CSS file: (empty)
- URL for custom CSS file (fullscreen): (empty)
- Enable paging: ☐
- Theme: Default

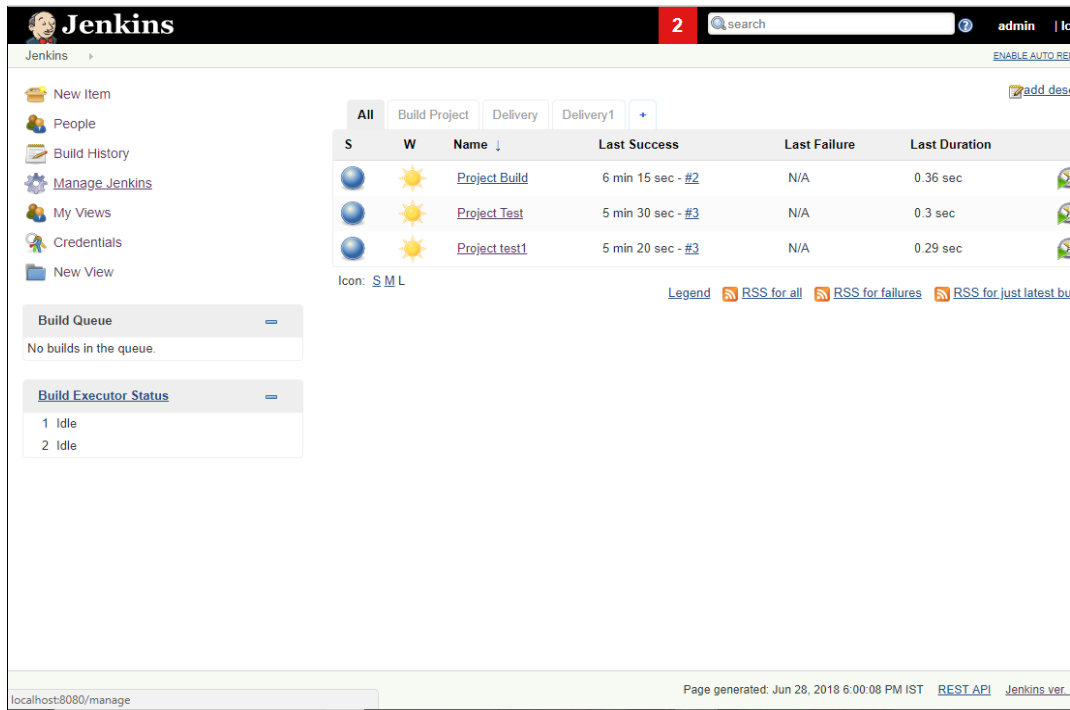
At the bottom, there are 'OK' and 'Apply' buttons.

3. The view is displayed



How to install the Plugins?

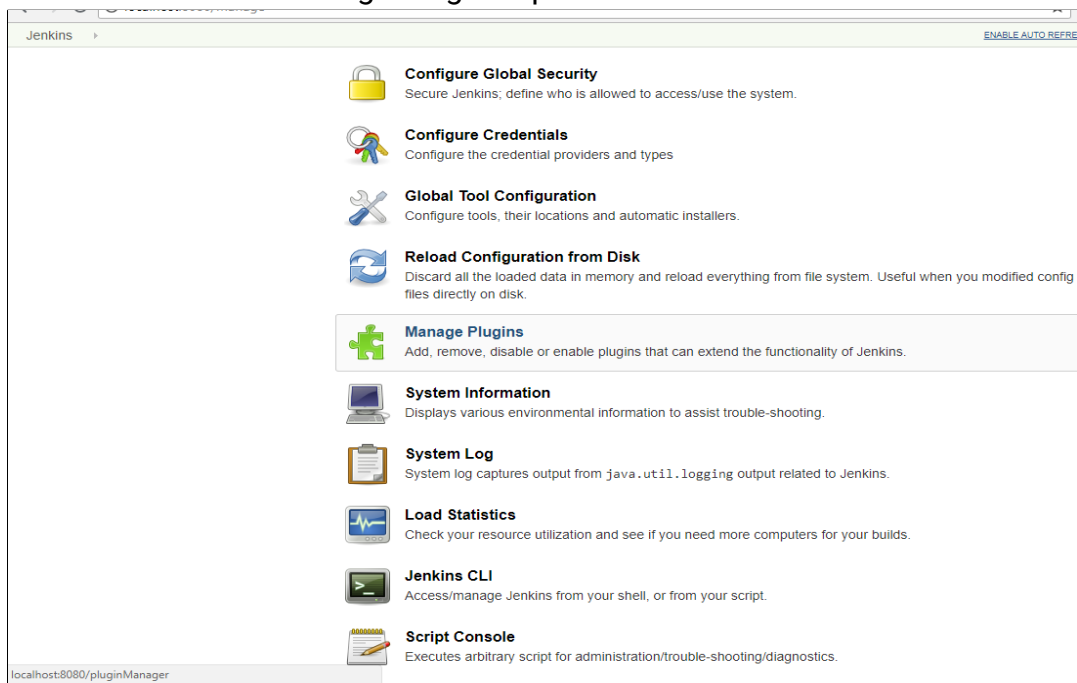
1. Go to the “Manage Jenkins” option present on the Jenkins Dashboard.



The screenshot shows the Jenkins Dashboard. The left sidebar contains the following menu items: New Item, People, Build History, **Manage Jenkins** (highlighted), My Views, Credentials, and New View. The main content area displays a table of build history for the 'Project Build' filter. The table has columns for Status (S), Weather (W), Name, Last Success, Last Failure, and Last Duration. Below the table, there are links for 'Icon: S M L', 'Legend', 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'. On the left, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The bottom status bar shows 'localhost:8080/manage' and 'Page generated: Jun 28, 2018 6:00:08 PM IST'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
🌤️	☀️	Project Build	6 min 15 sec - #2	N/A	0.36 sec
🌤️	☀️	Project Test	5 min 30 sec - #3	N/A	0.3 sec
🌤️	☀️	Project test1	5 min 20 sec - #3	N/A	0.29 sec

2. Click on the Manage Plugins option.

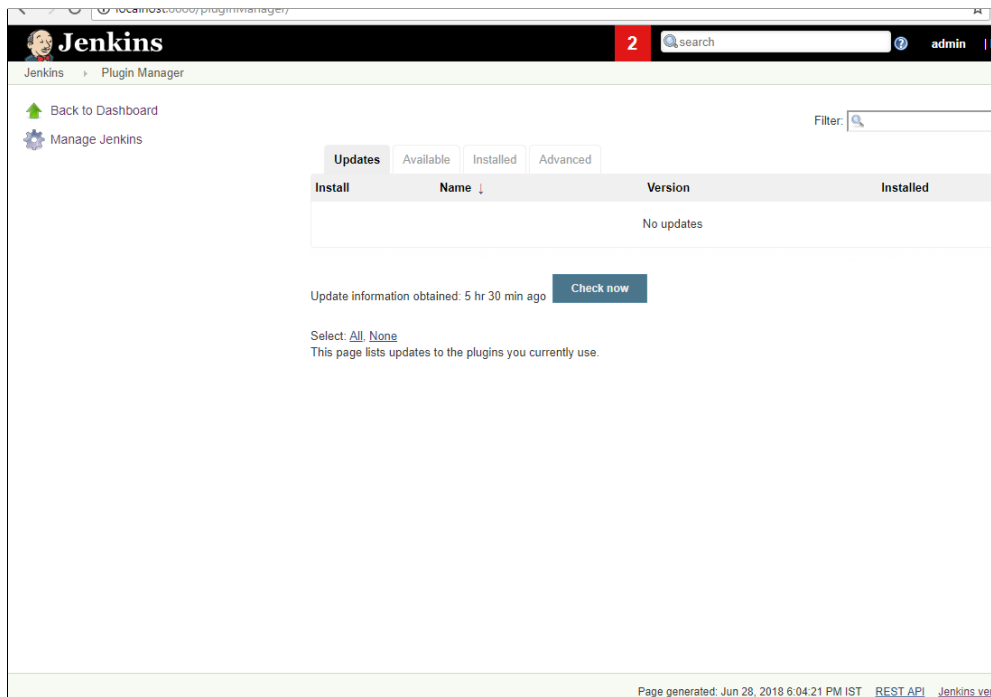


The screenshot shows the 'Manage Plugins' page in Jenkins. It features a list of configuration options, each with an icon and a description:

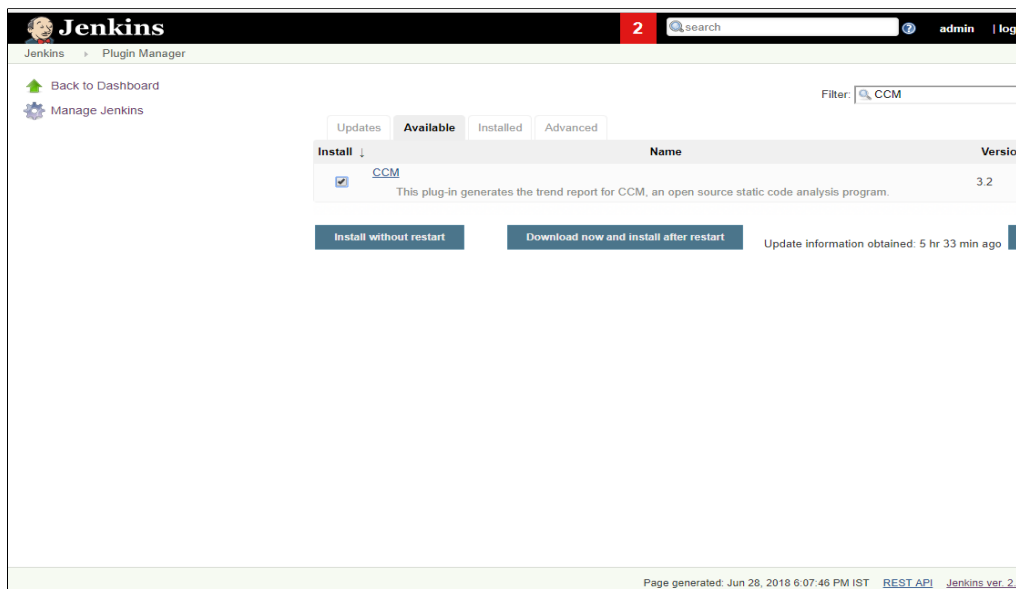
- Configure Global Security**: Secure Jenkins; define who is allowed to access/use the system.
- Configure Credentials**: Configure the credential providers and types.
- Global Tool Configuration**: Configure tools, their locations and automatic installers.
- Reload Configuration from Disk**: Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.
- Manage Plugins** (highlighted): Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from `java.util.logging` output related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- Jenkins CLI**: Access/manage Jenkins from your shell, or from your script.
- Script Console**: Executes arbitrary script for administration/trouble-shooting/diagnostics.

The bottom status bar shows 'localhost:8080/pluginManager'.

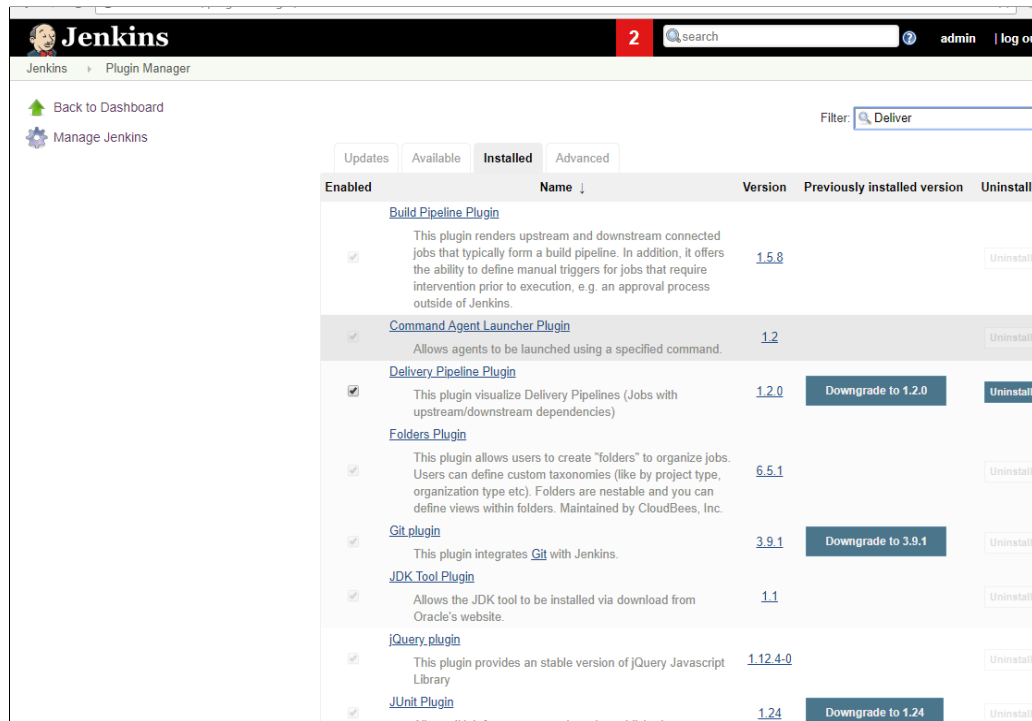
3. After clicking on the “Manage Plugins” option, it further comprises of 4 other options i. i.e Updates, Available, Installed and Advanced.



4. Go to the Available tab to see if the required plugin. If the plugin is already installed one can directly go to the installed section and find for the required plugin.



In case the Plugin is already installed.



The screenshot shows the Jenkins Plugin Manager interface. The top navigation bar includes the Jenkins logo, a red notification badge with the number '2', a search bar, and user links for 'admin' and 'log out'. The breadcrumb trail shows 'Jenkins' > 'Plugin Manager'. On the left sidebar, there are links for 'Back to Dashboard' and 'Manage Jenkins'. The main content area has a filter set to 'Deliver' and tabs for 'Updates', 'Available', 'Installed', and 'Advanced'. The 'Installed' tab is active, displaying a table of installed plugins.

Enabled	Name ↓	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Build Pipeline Plugin This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.	1.5.8		Uninstall
<input checked="" type="checkbox"/>	Command Agent Launcher Plugin Allows agents to be launched using a specified command.	1.2		Uninstall
<input checked="" type="checkbox"/>	Delivery Pipeline Plugin This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)	1.2.0	Downgrade to 1.2.0	Uninstall
<input checked="" type="checkbox"/>	Folders Plugin This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.	6.5.1		Uninstall
<input checked="" type="checkbox"/>	Git plugin This plugin integrates Git with Jenkins.	3.9.1	Downgrade to 3.9.1	Uninstall
<input checked="" type="checkbox"/>	JDK Tool Plugin Allows the JDK tool to be installed via download from Oracle's website.	1.1		Uninstall
<input checked="" type="checkbox"/>	jQuery plugin This plugin provides a stable version of jQuery Javascript Library	1.12.4-0		Uninstall
<input checked="" type="checkbox"/>	JUnit Plugin	1.24	Downgrade to 1.24	Uninstall