

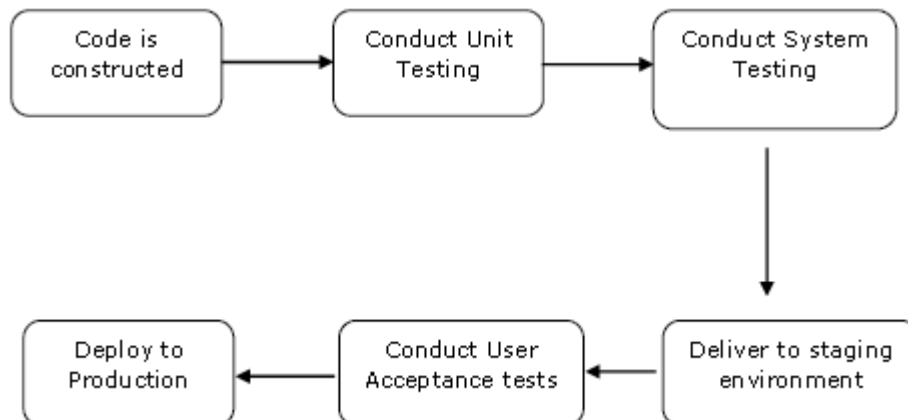
## ASSIGNMENT 7 DevOps

IT\_BE\_41\_Himanshu Shukla

### Use Jenkins “Deploy to container” plugin and "Build pipeline" plugin to implement continuous deployment and delivery of a project

Pre-requisites: Jenkins

Jenkins is used in providing good support for continuous deployment and delivery. The flow of a software development till the deployment is shown below:



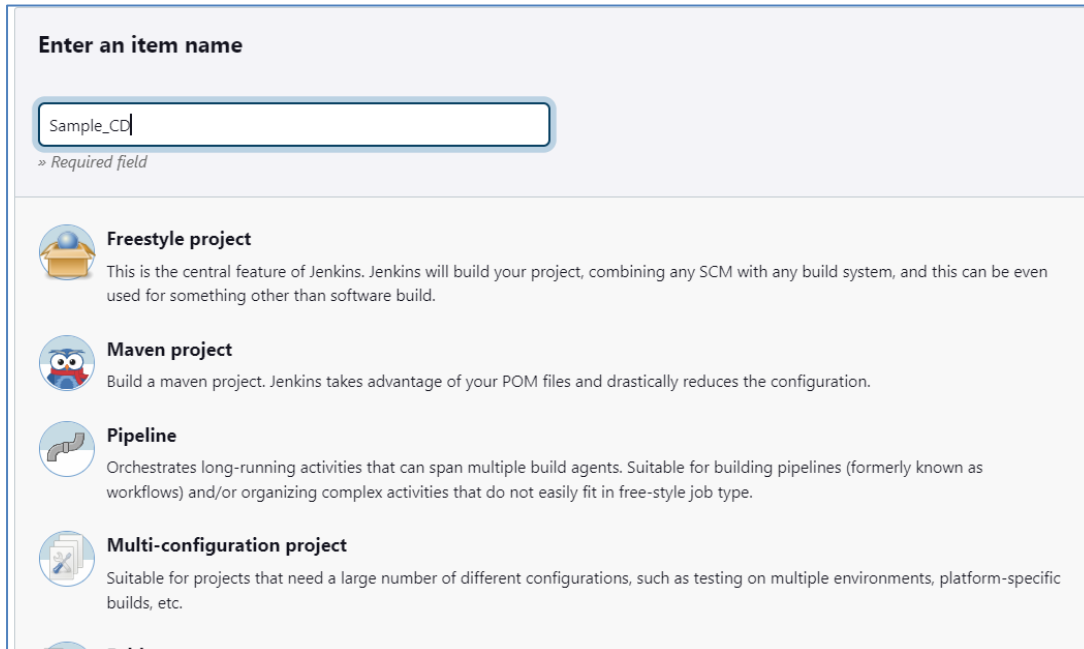
The main part of continuous deployment to make sure that the above entire process is automated. Jenkins provides various plugins for all these things. One of them is "**Deploy to container**" plugin, which was seen in earlier sections.

**Step 1:** Go to the Jenkins Dashboard and select **New Item**.

The screenshot shows the Jenkins Dashboard. The 'New Item' button is highlighted with an orange box. The dashboard includes a sidebar with navigation links: People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area displays a table of build jobs with columns for status (S), warning (W), name, last success, last failure, and last duration. The 'Build Queue' section at the bottom indicates 'No builds in the queue.'

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	maven_deploy	13 hr #2	N/A	8.4 sec
✓	☀	simple_java	1 day 9 hr #4	N/A	11 sec
✓	☀	simple_python	1 mo 3 days #1	N/A	1.1 sec

**Step 2:** Give the Item name and choose **Freestyle project** option. Here I have given the item name "**Sample\_CD**". Click on OK button.



**Enter an item name**

Sample\_CD

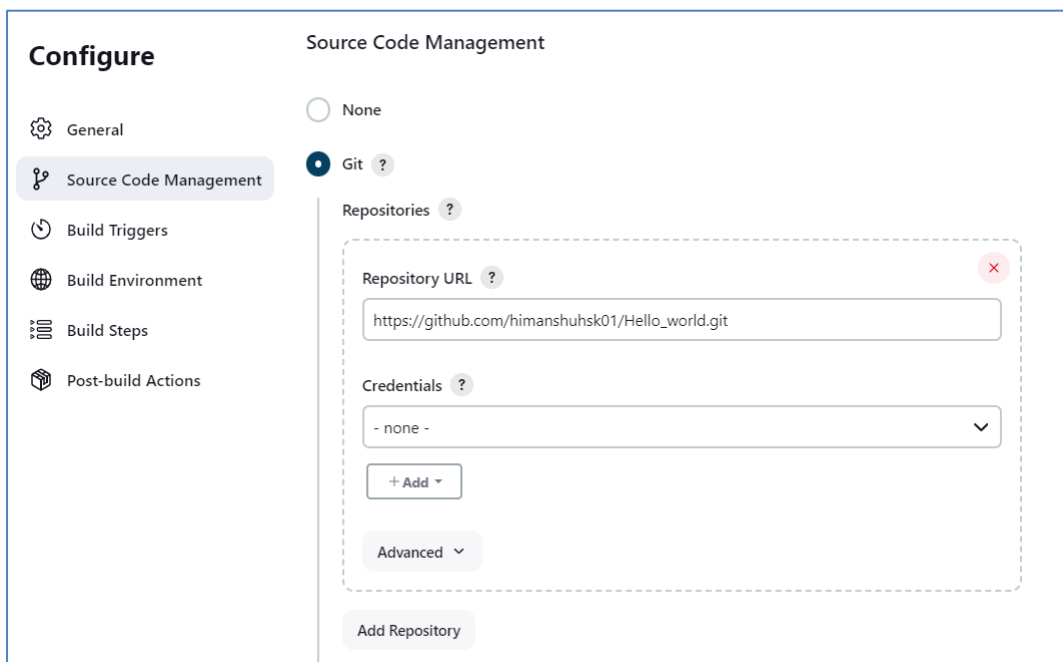
» Required field

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

**Step 3:** In this example, we are keeping it simple and just using to print HelloWorld.

Select the Git option and enter the GitHub repository of your program in the **Repository URL** section.([click](#))

note:your branch name should to be master because jenkins use it as default



**Configure**

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/himanshuhs01/Hello\_world.git

Credentials ?

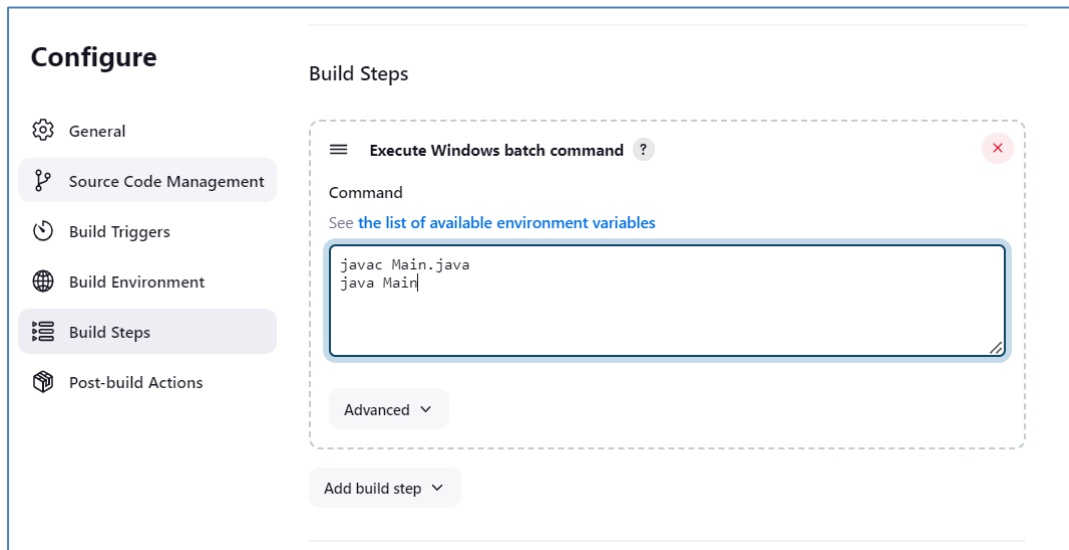
- none -

+ Add

Advanced

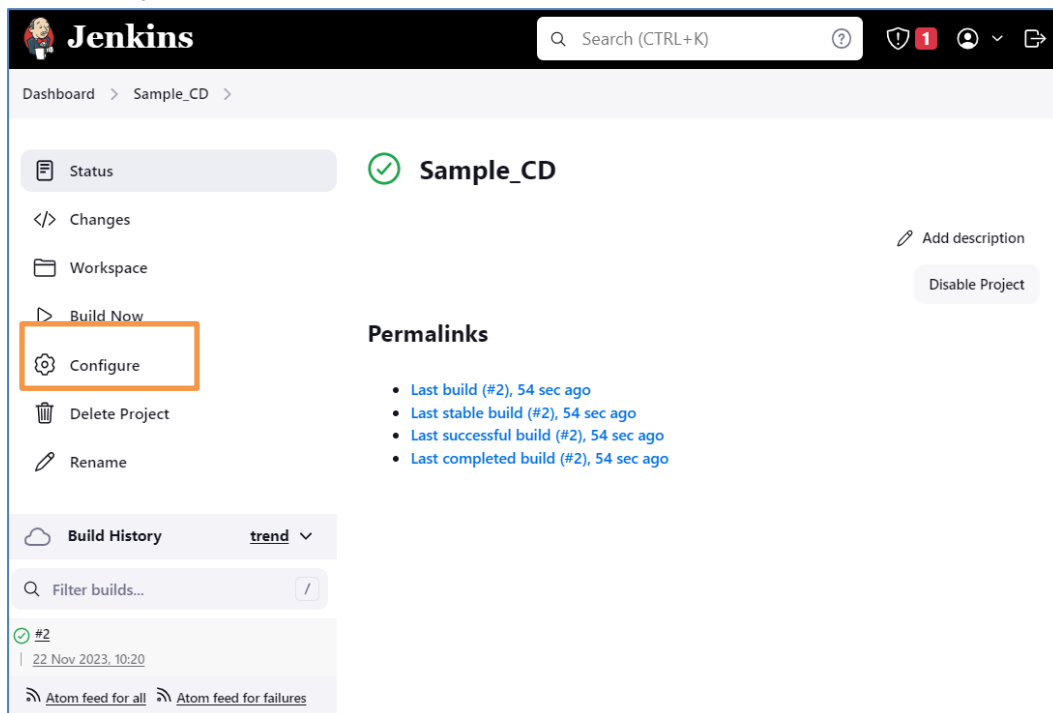
Add Repository

**Step 4:** Select the **Execute Windows batch command** option from the **add build step** button and give the command to run your java program.



Click on **Apply** then **Save** button.

**Step 5:** So our project is now created. You can check a build to see if the build is successfully created or not. To check a build, click on the **Build Now** option



Just like this ,Create a one more free style Project name **HelloWorld** of same github url([click](#))

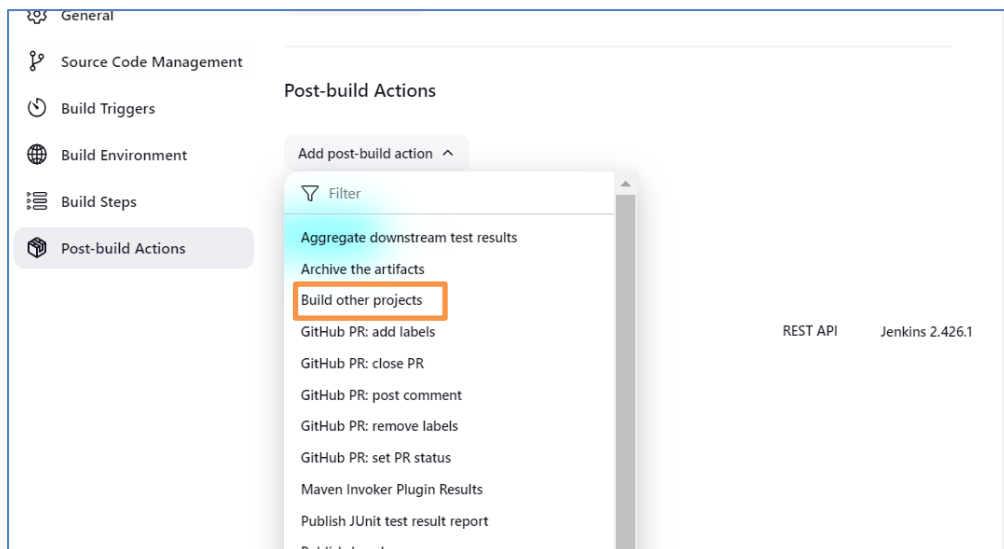
Now you can see my both Project HelloWorld and Sample\_CD

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	HelloWorld	6 min 11 sec #2	N/A	1.8 sec
✓	☀	maven_deploy	15 hr #2	N/A	8.4 sec
✓	☀	Sample_CD	6 min 3 sec #2	N/A	1.8 sec
✓	☀	simple_java	1 day 11 hr #4	N/A	11 sec
✓	☀	simple_python	1 mo 3 days #1	N/A	1.1 sec

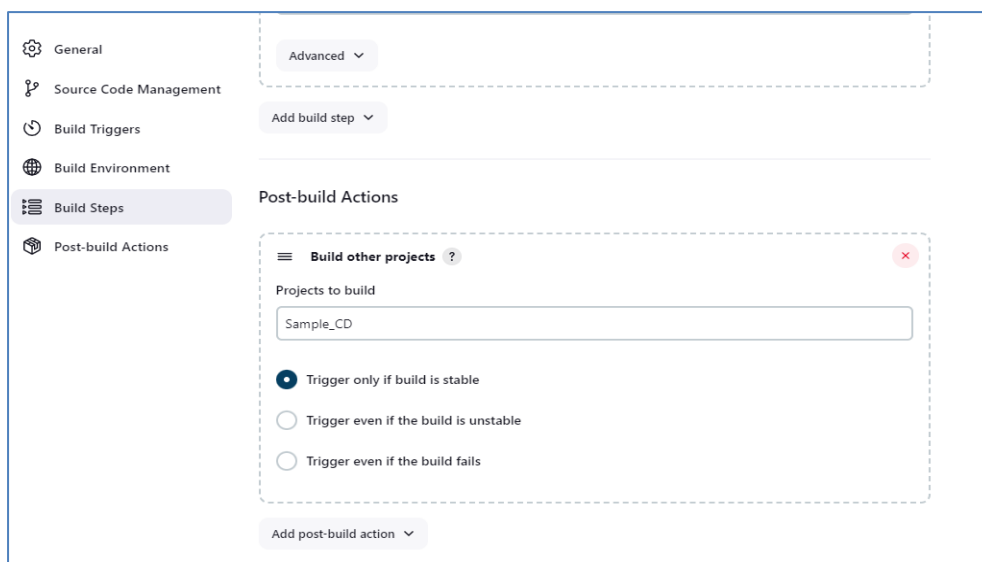
**Step 6:** Now, go to your previously created **Helloworld** project and click on the **Configure** option.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	HelloWorld	6 min 11 sec	N/A	1.8 sec
✓	☀	maven_deploy		N/A	8.4 sec
✓	☀	Sample_CD		N/A	1.8 sec
✓	☀	simple_java		N/A	11 sec
✓	☀	simple_python		N/A	1.1 sec

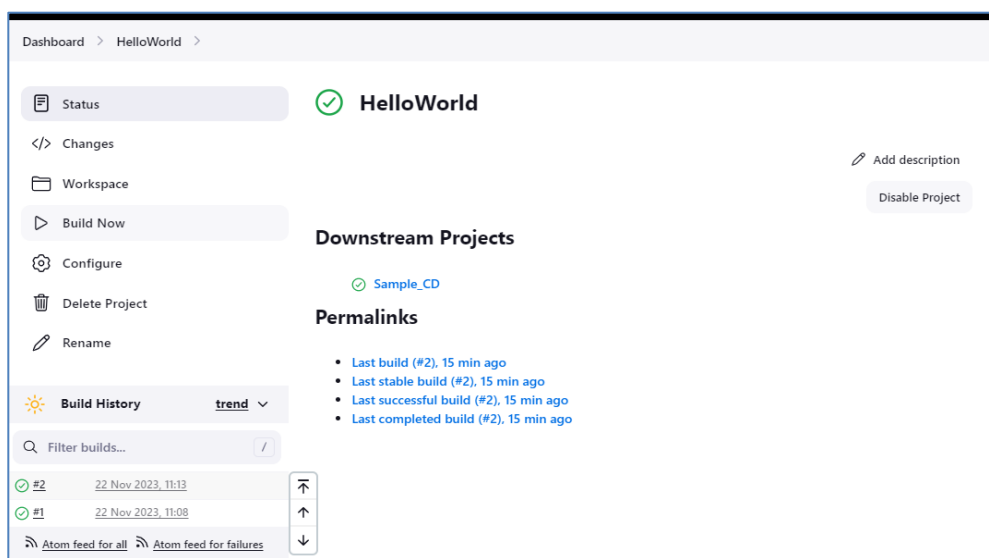
**Step 7:** In the Project configuration, select the **Add post-build** action and choose **Build other projects** option.



**Step 8:** In the **Projects to build** option, enter the "SAMPLE\_CD" as the project name to build. You can leave the other option as the default. Click on **Apply** then the **Save** button.



**Step 9:** Now, build the HelloWorld project. To do that, click on the **Build Now** option.



**Step 10:** Now, if you see the Console output, you will also see that after the **HelloWorld** project is successfully built, the build of the demo project will also happen.

The screenshot shows the Jenkins console output for build #2 of the HelloWorld project. The left sidebar contains navigation links: Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#2', Git Build Data, and Previous Build. The main area displays the console output, which shows the build process starting with user Himanshu, running as SYSTEM, and building in the workspace C:\ProgramData\Jenkins\workspace\HelloWorld. It then shows the recommended git tool is NONE, no credentials specified, and the build fetching changes from the remote Git repository. The output ends with the command: > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10.

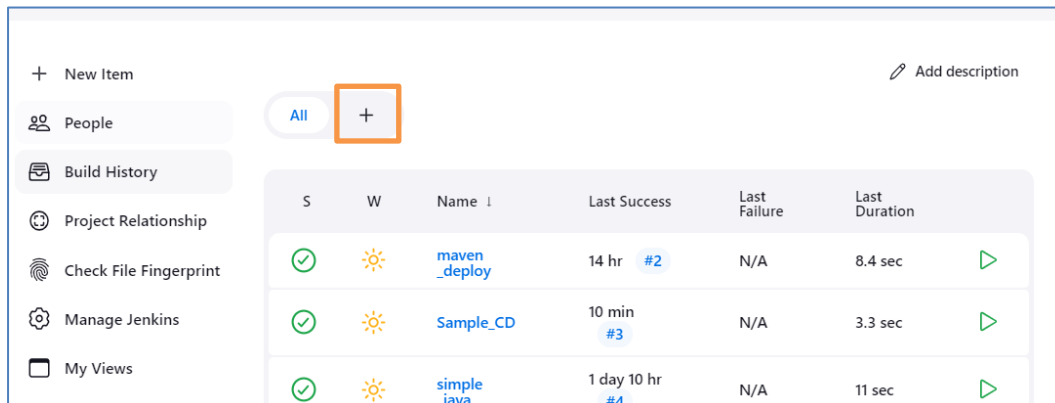
**Step 11:** Let's now install the **Delivery pipeline plugin**. Go to Dashboard > Manage Jenkins > plugins and install the "delivery Plugin".

The screenshot shows the Jenkins Manage Jenkins > Plugins page. The left sidebar contains navigation links: Updates (26), Available plugins, Installed plugins, Advanced settings, and Download progress. The main area shows the search results for "Delivery Pipeline". The "Install" button is highlighted with an orange box. The table below shows the "Delivery Pipeline" plugin (1.4.2) with a "User Interface" link and a description: "This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)". The "Released" column shows "3 yr 8 mo ago".

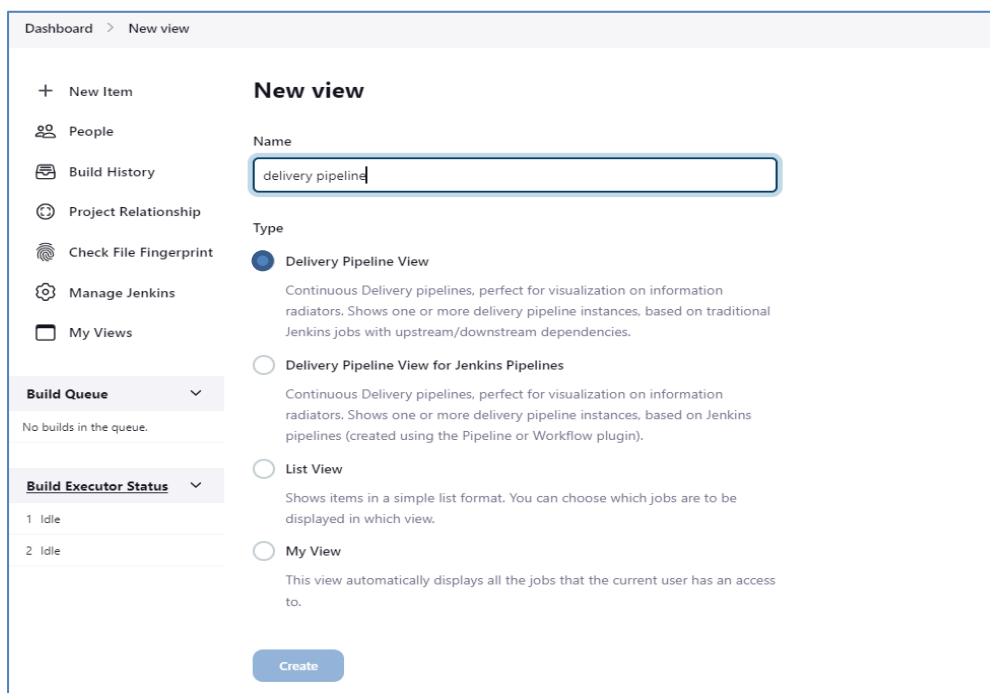
After successful installation of plugins, click on **Go back to the top page** link

The screenshot shows the Jenkins Manage Jenkins > Plugins page, specifically the "Download progress" section. The left sidebar contains navigation links: Updates (26), Available plugins, Installed plugins, Advanced settings, and Download progress (selected). The main area shows the "Download progress" section with a "Preparation" status and a list of plugins being installed: Parameterized Trigger (Success), jQuery (Success), Delivery Pipeline (Success), and Loading plugin extensions (Running). A "Go back to the top page" link is highlighted with an orange box, with the text "(you can start using the installed plugins right away)". Below this link is a checkbox labeled "Restart Jenkins when installation is complete and no jobs are running".

**Step 12:** To see the Delivery Pipeline in action, click on + symbol in the tab next to the **All** tab on the Jenkins Dashboard screen.



**Step 13:** Give the **View name** and select **Delivery Pipeline View**. Click on create button.



**Step 14:** In the next page, leave the default options. Scroll down and change the following settings:

- Make sure "Show static analysis results" option is checked.
- Make sure the option "Show total build time" is checked.
- In the Pipelines section for the Initial job enter the Helloworld project as the first job which should build.
- Give any name for the Pipeline
- Click the Apply and OK button.

Update interval ?

- ☐ Enable start of new pipeline build ?
- ☐ Enable manual triggers ?
- ☐ Enable rebuild ?
- ☐ Allow cancelling pipeline builds ?
- ☐ Show avatars ?
- ☐ Show commit messages ?
- ☐ Show absolute date and time ?
- ☐ Show job description ?
- ☐ Show job promotions ?
- ☐ Show test results ?
- ☒ Show static analysis results ?
- ☒ Show total build time ?
- ☐ Use relative links for easier navigation ?

Dashboard > delivery pipeline >

### Pipelines

Components

≡ Component

Name ?

Initial Job ?

HelloWorld

Final Job (optional) ?

☐ Show upstream

Now you can view the delivery Pipeline, and **project delivery is successful**

+ New Item

People

Build History

Edit View

Delete View

Project Relationship

Check File Fingerprint

View Fullscreen

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

All

Delivery Pipeline

delivery pipeline

+

myJob

#2 triggered by user Himanshu started 28 minutes ago

Total build time: 3 sec

HelloWorld

HelloWorld

28 minutes ago

1 sec

Sample\_CD

Sample\_CD

27 minutes ago

1 sec

#1 triggered by user Himanshu started 32 minutes ago

Total build time: 3 sec

HelloWorld

HelloWorld

32 minutes ago

3 sec

Sample\_CD

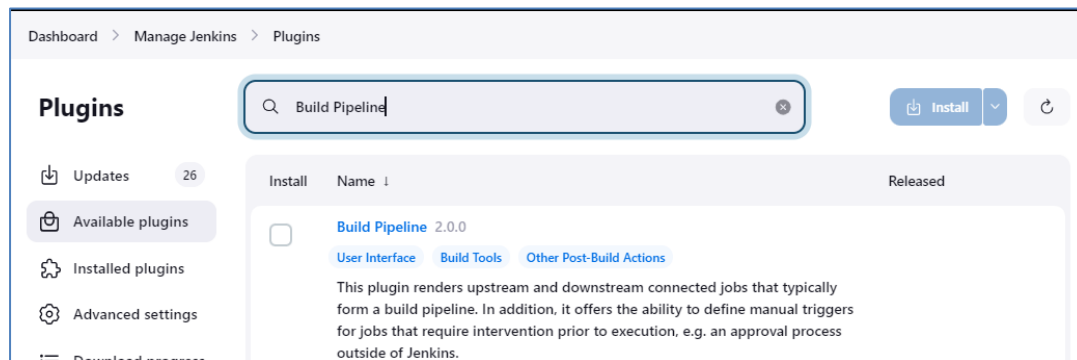
Sample\_CD



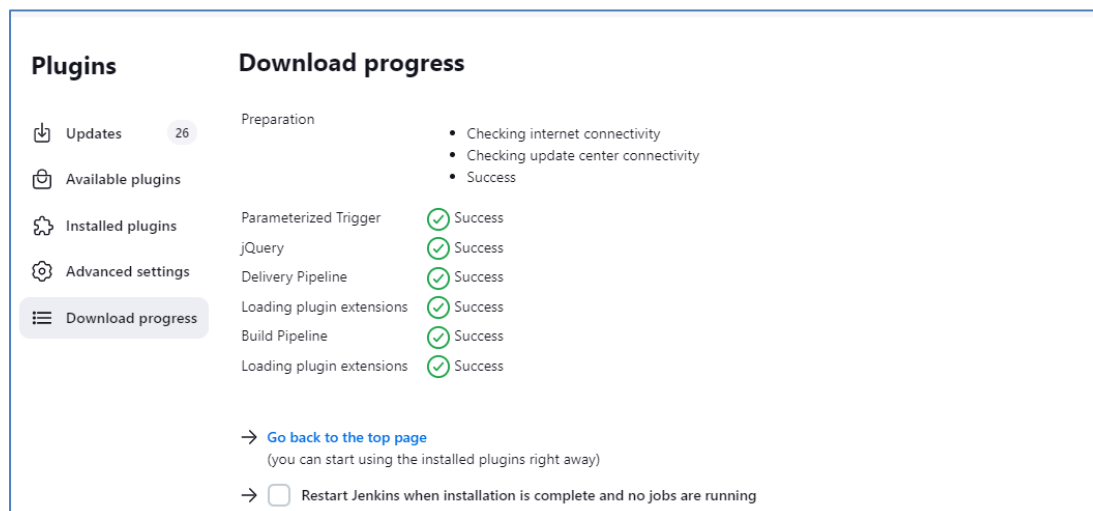
## Build Pipeline Plugin

Another important plugin of Jenkins is the "**Build pipeline**" plugin. Let's take a look at this plugin:

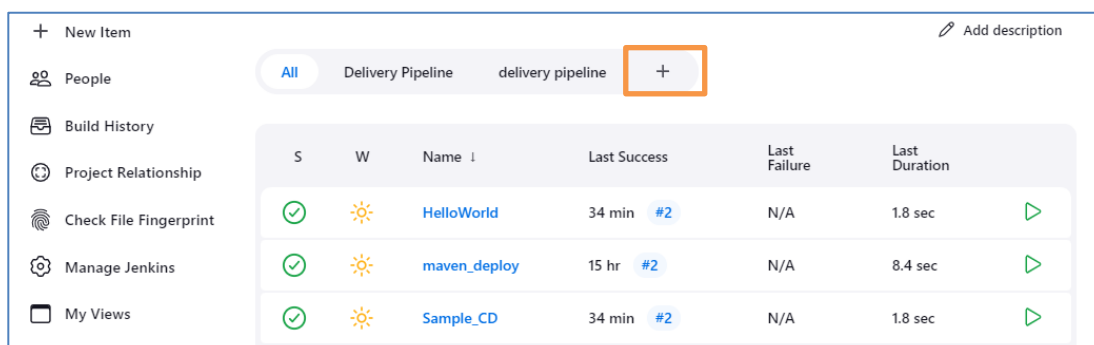
**Step 1:** On the Jenkins Dashboard, select **Manage Jenkins>Plugin** and install the "**Build Pipeline plugin**" under available plugin.



**Step 2:** Once the installation is completed successfully, click on **Go back to the top page** link.



**Step 3:** To see the Build pipeline in action, click on the + symbol in the Tab next to the **All** tab in the Jenkins Dashboard.



**Step 4:** In the View name option, enter any name and choose the Build **Pipeline View** Option.

**Step 7:** Leave all the default option and scroll down. In the **Upstream/downstream config** section enter the name of the HelloWorld project for the select initial job option. Then click on the **OK** button.

Now you can see a view of the entire delivery pipeline, and you will be able to see the status of each project in the whole pipeline.

