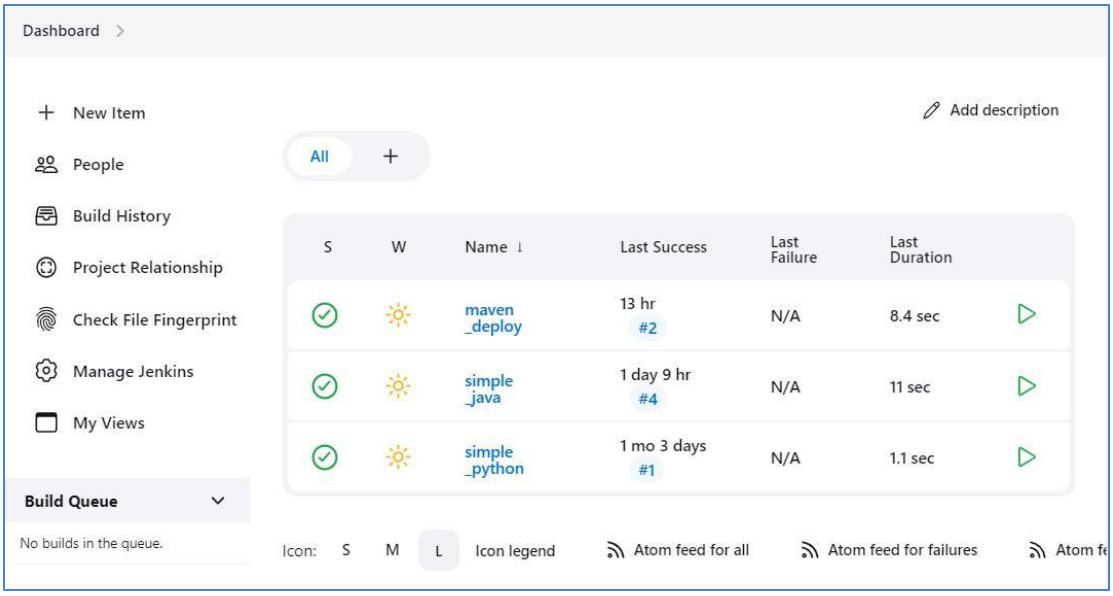
Assignment No. 7

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

Pre-requisites: Jenkins

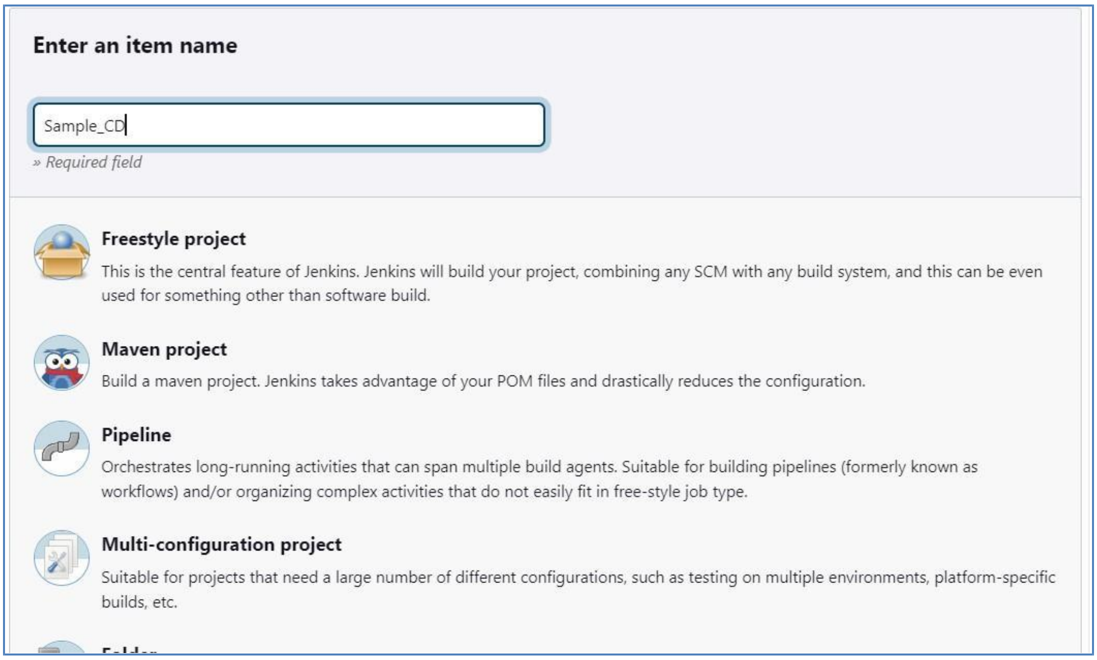
Step 1: Go to the Jenkins Dashboard and select **New Item**.



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	▷

No builds in the queue.

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.

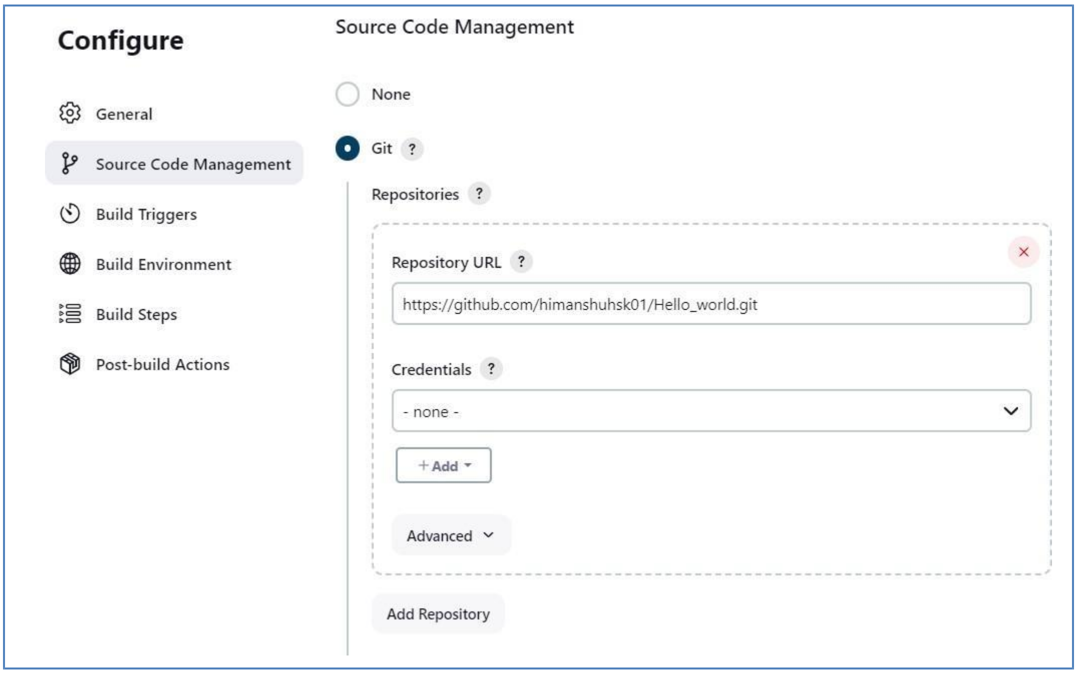
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

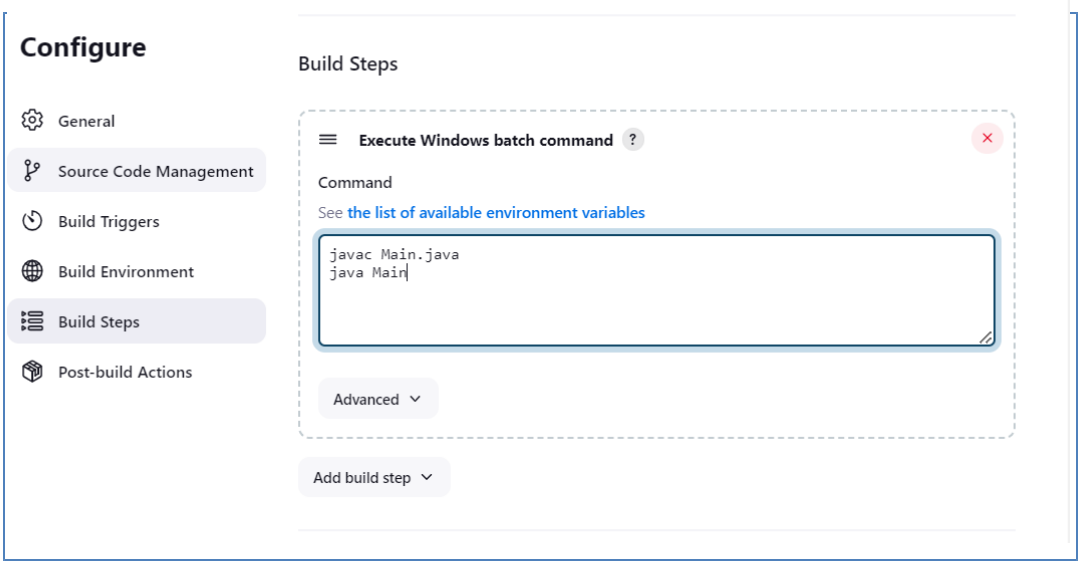
● Git

Repositories

Repository URL: https://github.com/himanshuhsk01/Hello_world.git

Credentials: - none -

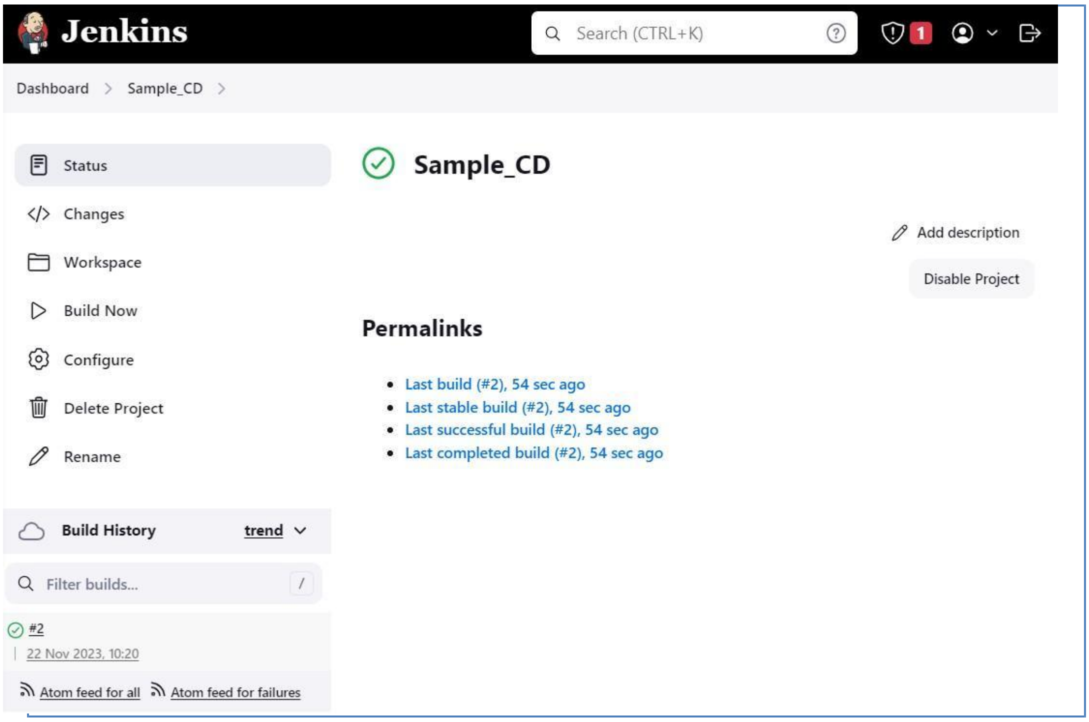
Step 4: Select the **Execute Windows batch command** option from the **add build step** button and give the command to run your java program.



```
javac Main.java
java Main
```

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

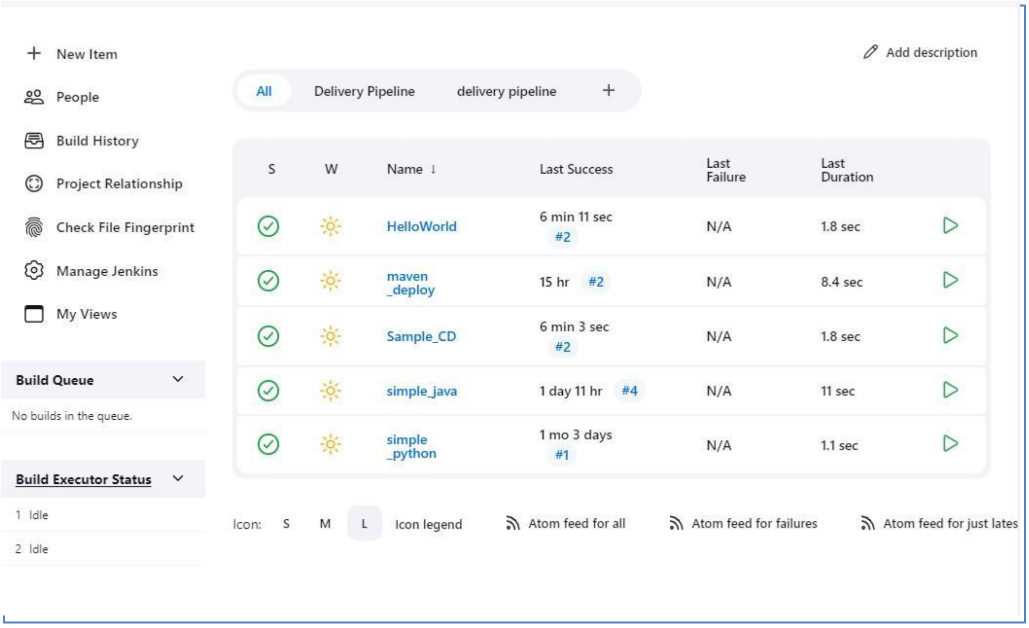


Permalinks

- Last build (#2), 54 sec ago
- Last stable build (#2), 54 sec ago
- Last successful build (#2), 54 sec ago
- Last completed build (#2), 54 sec ago

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



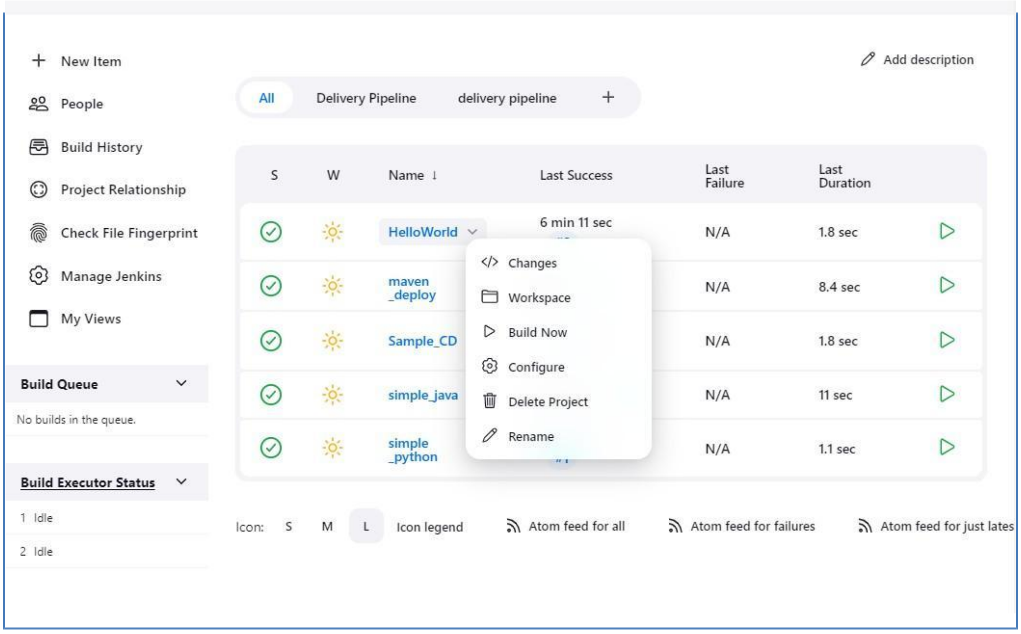
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	HelloWorld	6 min 11 sec #2	N/A	1.8 sec
✓	☀	maven _deploy	15 hr #2	N/A	8.4 sec
✓	☀	Sample_CD	6 min 3 sec #2	N/A	1.8 sec
✓	☀	simple_java	1 day 11 hr #4	N/A	11 sec
✓	☀	simple _python	1 mo 3 days #1	N/A	1.1 sec

No builds in the queue.

1 Idle

2 Idle

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



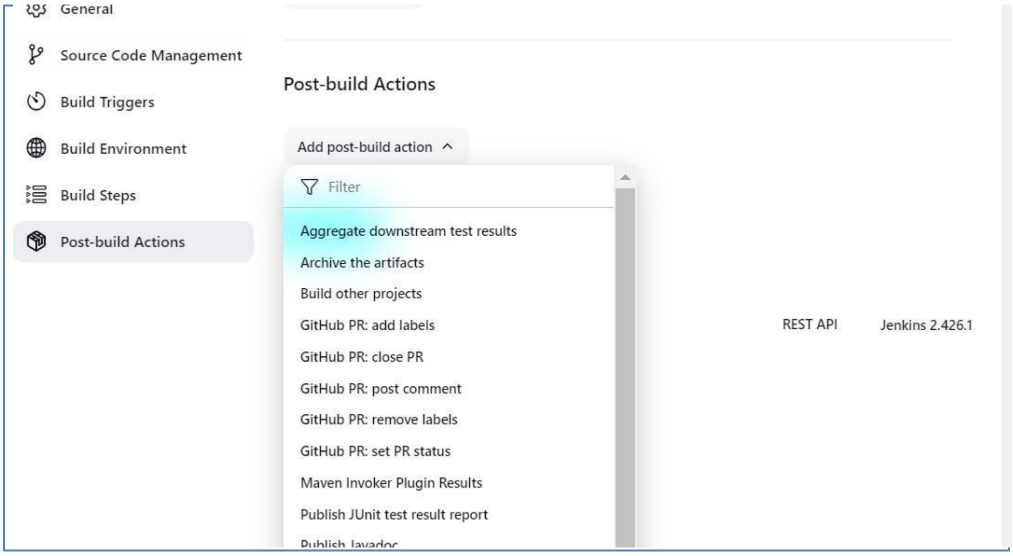
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	HelloWorld	6 min 11 sec	N/A	1.8 sec
✓	☀	maven _deploy		N/A	8.4 sec
✓	☀	Sample_CD		N/A	1.8 sec
✓	☀	simple_java		N/A	11 sec
✓	☀	simple _python		N/A	1.1 sec

No builds in the queue.

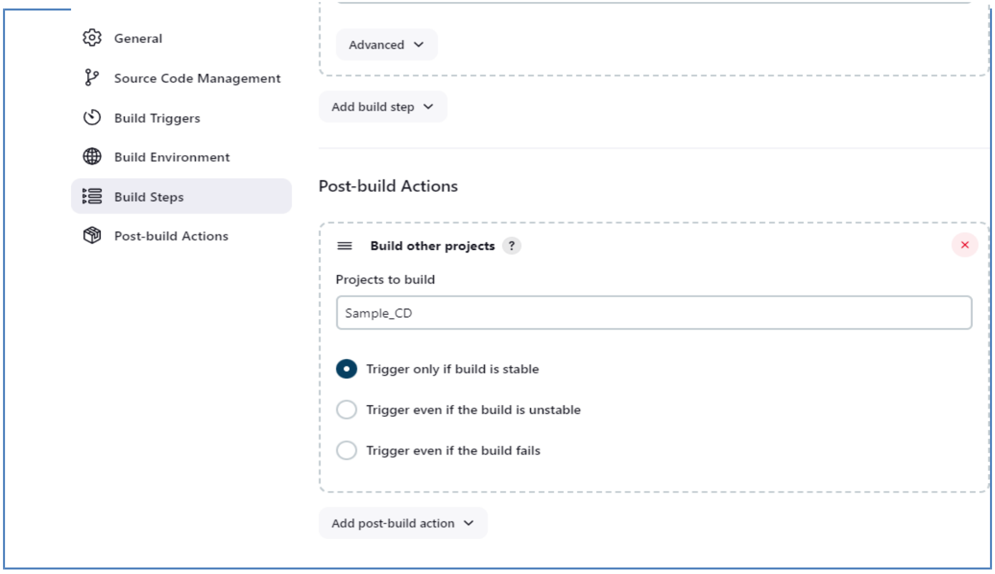
1 Idle

2 Idle

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.



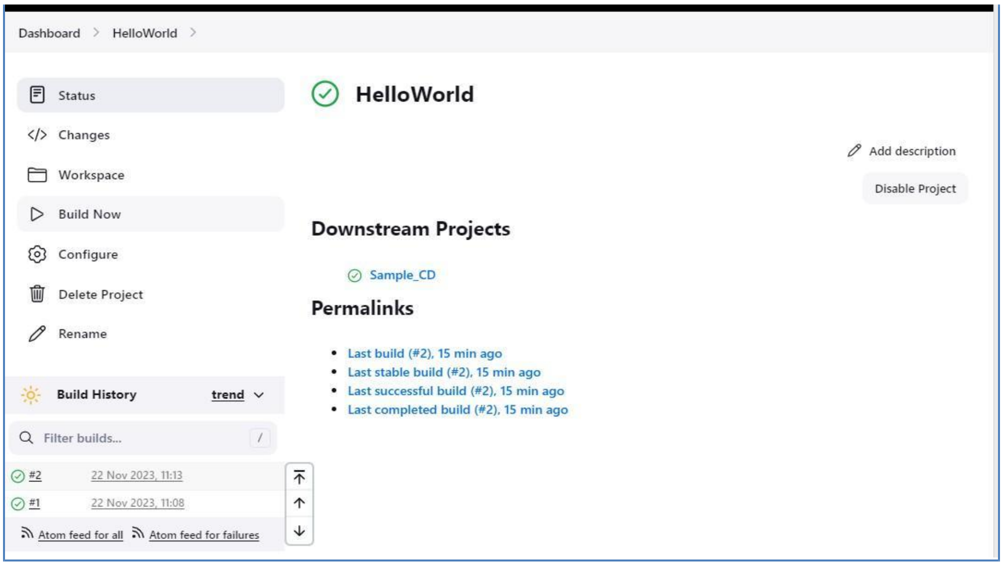
Projects to build: Sample_CD

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

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



Downstream Projects

Sample_CD

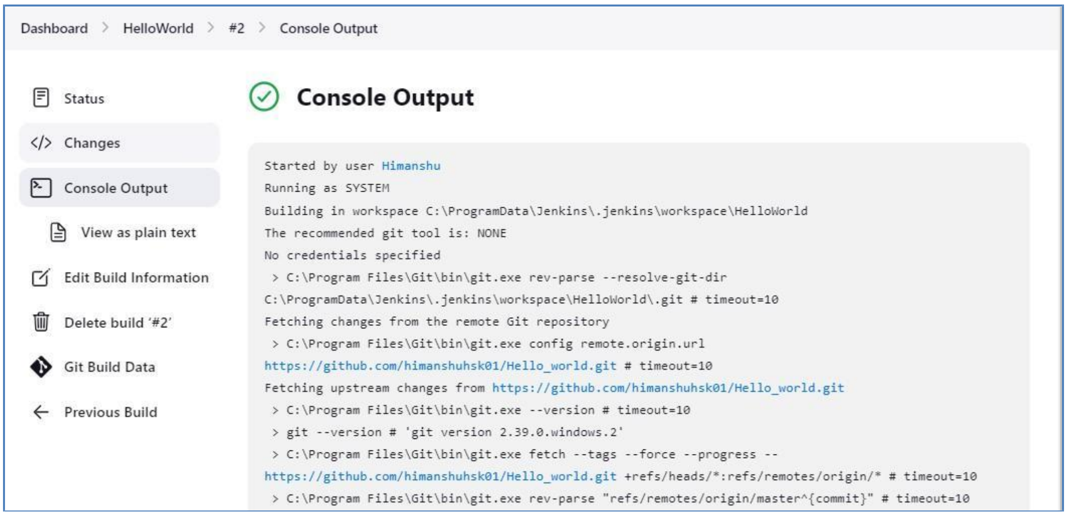
Permalinks

- Last build (#2), 15 min ago
- Last stable build (#2), 15 min ago
- Last successful build (#2), 15 min ago
- Last completed build (#2), 15 min ago

#2 22 Nov 2023, 11:13

#1 22 Nov 2023, 11:08

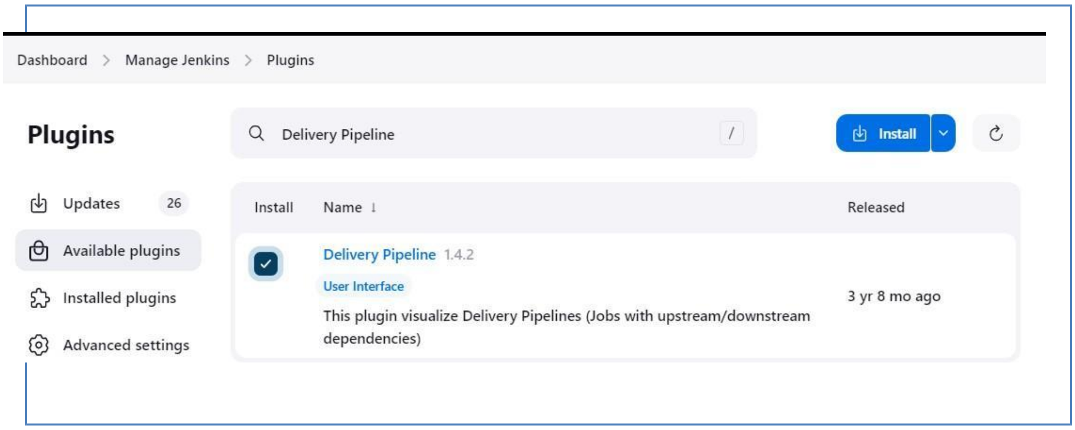
Step 10: Now, if you see the Console output, you will also see that after the **HelloWorld** project is successfully built, the build of the demo project will also happen.



Console Output

```
Started by user Himanshu
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\HelloWorld
The recommended git tool is: NONE
No credentials specified
 > C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir
C:\ProgramData\Jenkins\.jenkins\workspace\HelloWorld\.git # timeout=10
Fetching changes from the remote Git repository
 > C:\Program Files\Git\bin\git.exe config remote.origin.url
https://github.com/himanshuhsk01/Hello_world.git # timeout=10
Fetching upstream changes from https://github.com/himanshuhsk01/Hello_world.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > git --version # 'git version 2.39.0.windows.2'
 > C:\Program Files\Git\bin\git.exe fetch --tags --force --progress --
https://github.com/himanshuhsk01/Hello_world.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
```

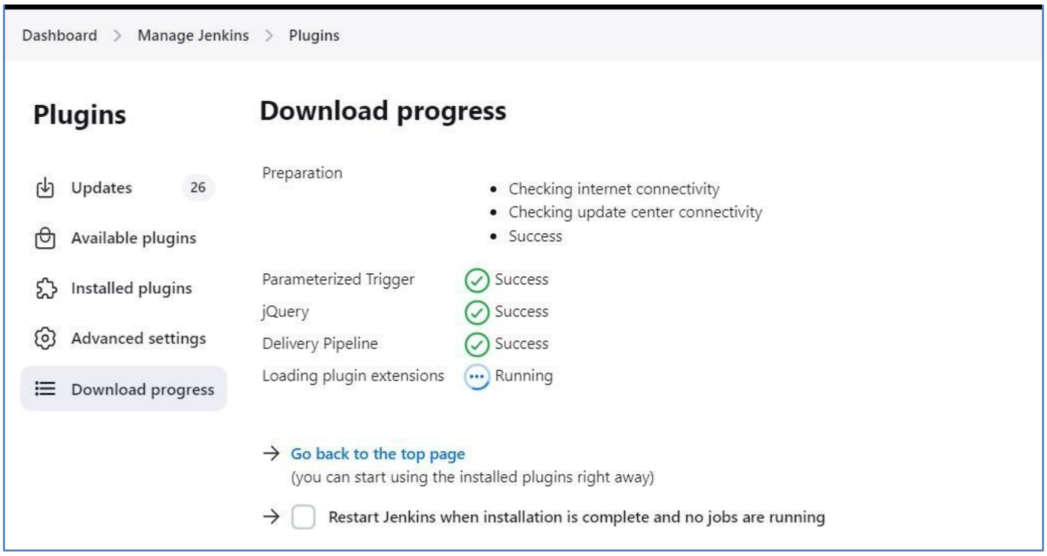
Step 11: Let's now install the **Delivery pipeline plugin**. Go to DashBoard >Manage Jenkins>plugins and install the "delivery Plugin".



Search: Delivery Pipeline

Install	Name ↓	Released
☑	Delivery Pipeline 1.4.2 User Interface This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)	3 yr 8 mo ago

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



Download progress

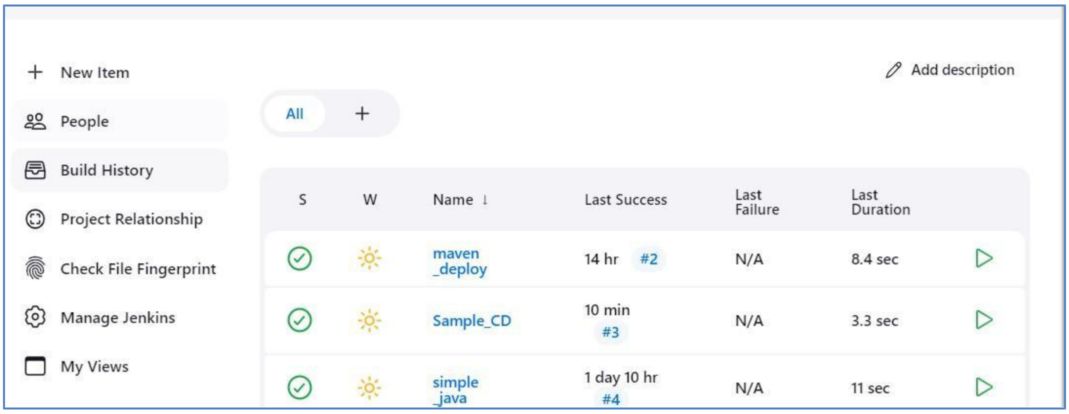
Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

Parameterized Trigger	✓ Success
jQuery	✓ Success
Delivery Pipeline	✓ Success
Loading plugin extensions	⋯ Running

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

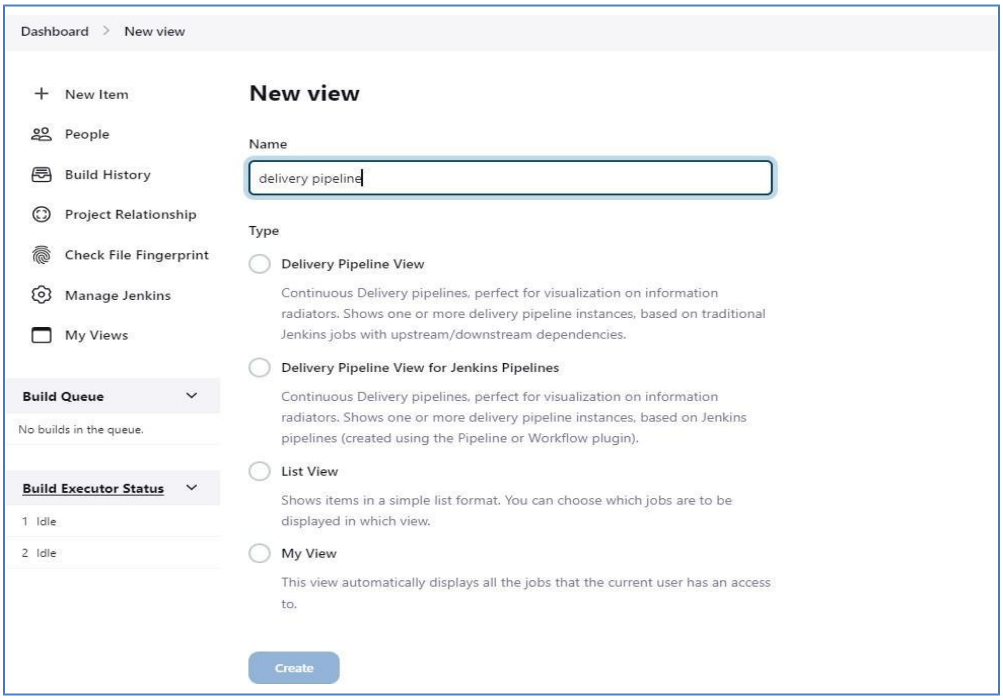
→ □ Restart Jenkins when installation is complete and no jobs are running

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.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☼	maven _deploy	14 hr #2	N/A	8.4 sec	▷
✓	☼	Sample_CD	10 min #3	N/A	3.3 sec	▷
✓	☼	simple _java	1 day 10 hr #4	N/A	11 sec	▷

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



New view

Name: delivery pipeline

Type

○ 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

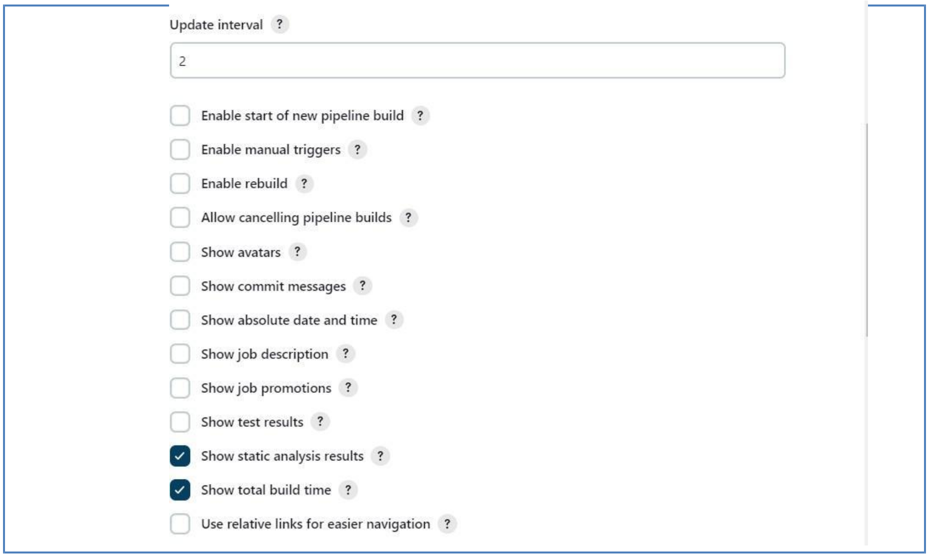
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

○ My View

This view automatically displays all the jobs that the current user has an access to.

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 ?

2

□ 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 ?

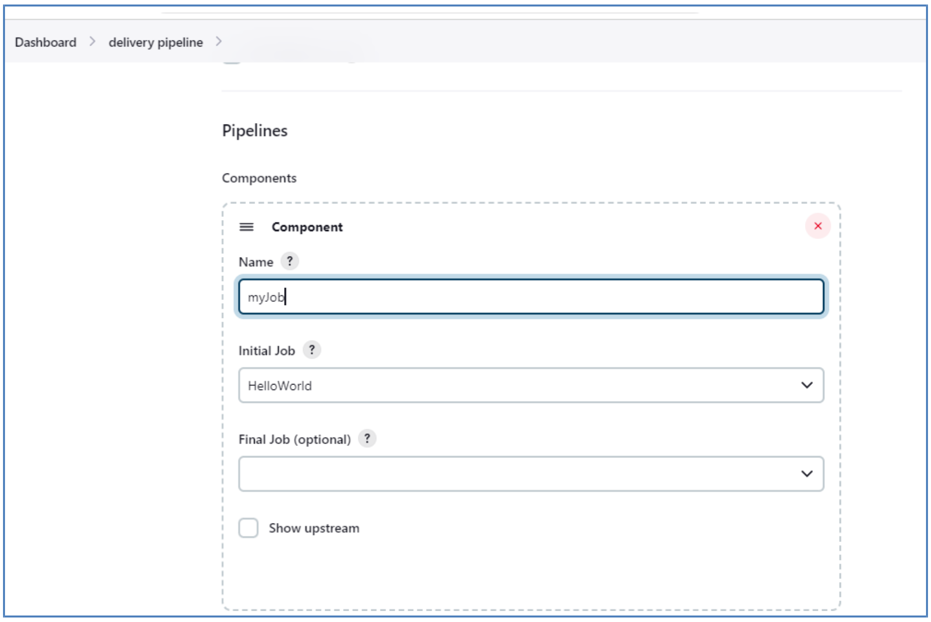
□ Show job promotions ?

□ Show test results ?

☑ Show static analysis results ?

☑ Show total build time ?

□ Use relative links for easier navigation ?



Dashboard > delivery pipeline >

Pipelines

Components

Component ×

Name ?

myJob

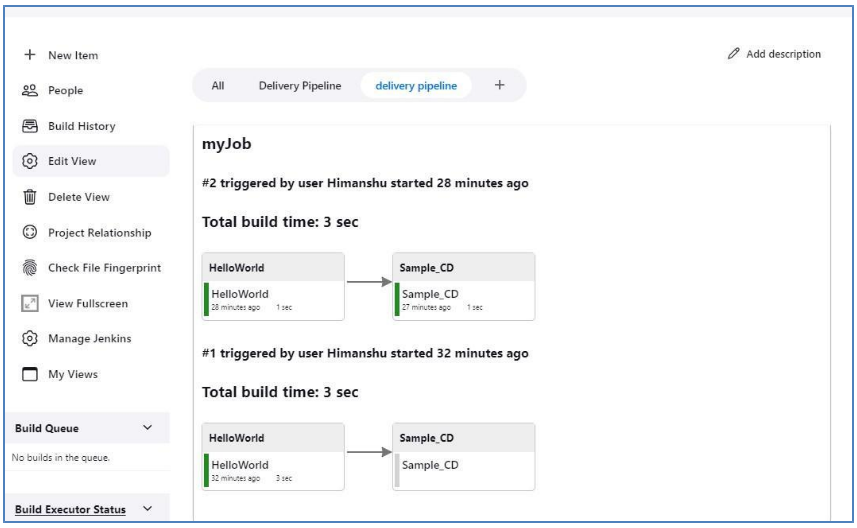
Initial Job ?

HelloWorld

Final Job (optional) ?

□ Show upstream

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



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

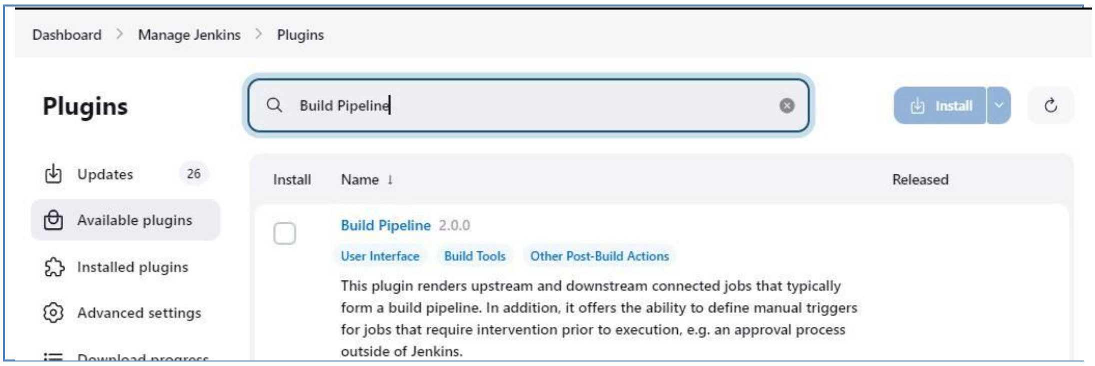
Total build time: 3 sec

HelloWorld — HelloWorld, 32 minutes ago, 3 sec → Sample_CD — Sample_CD

Build Pipeline Plugin

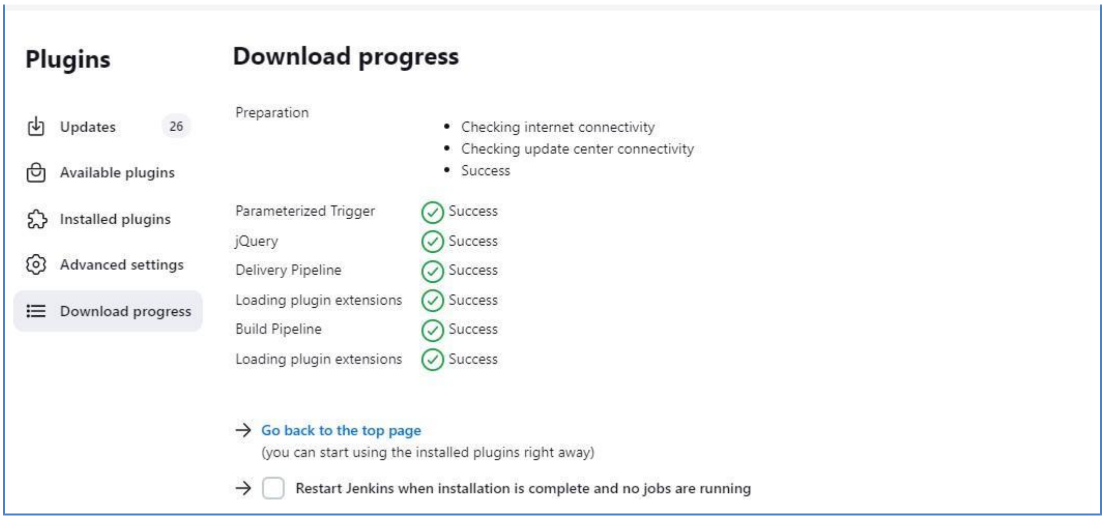
Another important plugin of Jenkins is the "**Build pipeline**" plugin. Let's take a look at this plugin:

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



Install	Name ↓	Released
☐	Build Pipeline 2.0.0 User Interface · Build Tools · Other Post-Build Actions 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.	

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



Download progress

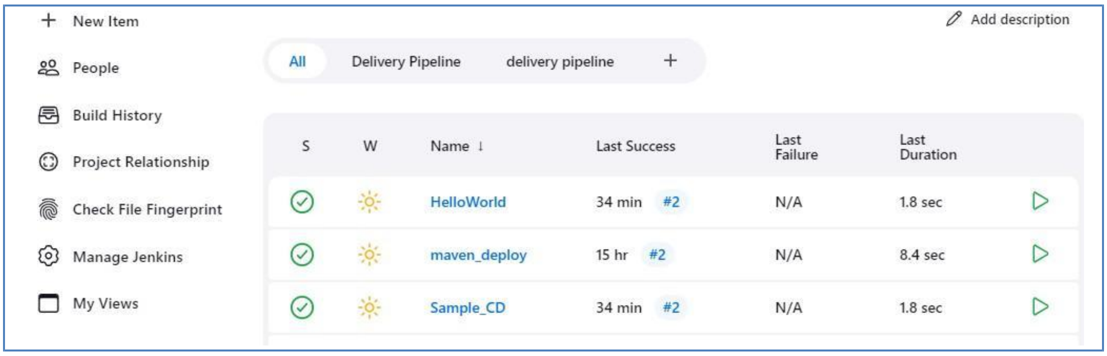
Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

Parameterized Trigger	✓ Success
jQuery	✓ Success
Delivery Pipeline	✓ Success
Loading plugin extensions	✓ Success
Build Pipeline	✓ Success
Loading plugin extensions	✓ Success

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

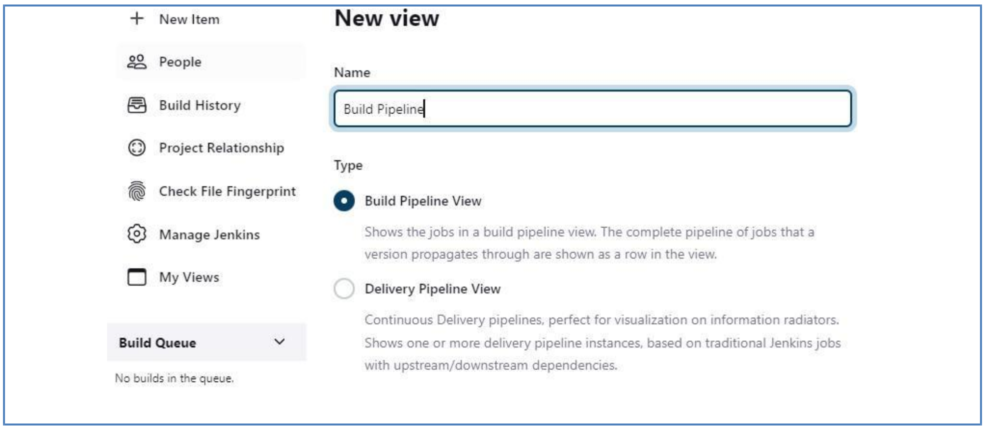
→ ☐ Restart Jenkins when installation is complete and no jobs are running

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.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	HelloWorld	34 min #2	N/A	1.8 sec	▷
✓	☀	maven_deploy	15 hr #2	N/A	8.4 sec	▷
✓	☀	Sample_CD	34 min #2	N/A	1.8 sec	▷

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



New view

Name

Build Pipeline

Type

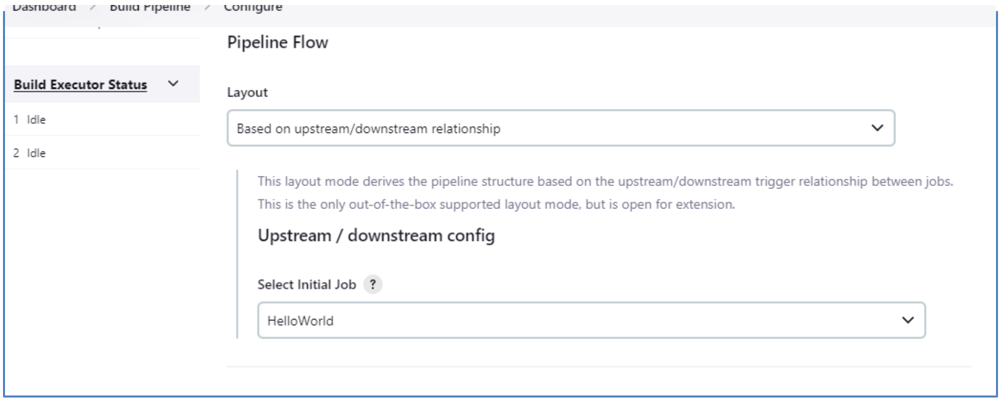
● Build Pipeline View

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

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.



Pipeline Flow

Layout

Based on upstream/downstream relationship

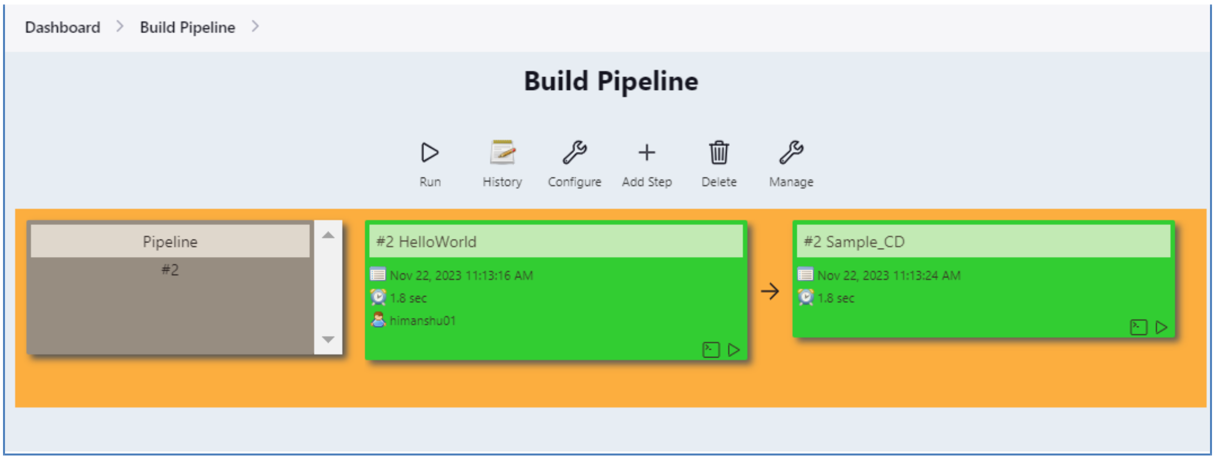
This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job ?

HelloWorld

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.



#2 HelloWorld
Nov 22, 2023 11:13:16 AM
1.8 sec
himanshu01

#2 Sample_CD
Nov 22, 2023 11:13:24 AM
1.8 sec