

# Install JDK

[Java Archive Downloads - Java SE 21](#)

[https://download.oracle.com/java/21/archive/jdk-21.0.7\\_windows-x64\\_bin.msi](https://download.oracle.com/java/21/archive/jdk-21.0.7_windows-x64_bin.msi) (sha256)

Also make sure Python is also installed in your system

Windows installer (64-bit) — Recommended

## Install Jenkins: Jenkins.war

Open Command Prompt and check the Java version

```
C:\Users\sweta>java -version
java version "21.0.7" 2025-04-15 LTS
Java(TM) SE Runtime Environment (build 21.0.7+8-LTS-245)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.7+8-LTS-245, mixed mode, sharing)
```

Then Type: java -jar "C:\Program Files\jenkins.war"

```
C:\Users\sweta>java -jar "C:\Program Files\jenkins.war"
Running from: C:\Program Files\jenkins.war
webroot: C:\Users\sweta\jenkins\war
2026-01-02 14:31:32.142+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2026-01-02 14:31:34.049+0000 [id=1] WARNING o.e.jee9.nested.ContextHandler#setContextPath: Empty contextPath
2026-01-02 14:31:34.932+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.25; built: 2025-08-11T23:52:37.219Z; git: a862b76d8372e24205765182d9ae1dd333ce2ea; jvm 21.0.7+8-LTS-245
2026-01-02 14:31:36.154+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.ee9.jsp.JettyJspServlet
2026-01-02 14:31:36.245+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2026-01-02 14:31:37.030+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\sweta\jenkins found at: $user.home/.jenkins
2026-01-02 14:31:37.191+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@2ce45a7b{Jenkins v2.528.3,/,b=file:///C:/Users/sweta/.jenkins/war/,a=AVAILABLE,h=oeje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@153d4abb{STARTED}}
2026-01-02 14:31:37.252+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@7bbb6a8{HTTP/1.1,(http/1.1)}{0.0.0.0:8080}
2026-01-02 14:31:37.305+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server@6853425f{STARTING}[12.0.25,sto=0] @6118ms
2026-01-02 14:31:37.387+0000 [id=45] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2026-01-02 14:31:37.472+0000 [id=41] INFO jenkins.model.Jenkins#<init>: Starting version 2.528.3
2026-01-02 14:31:37.606+0000 [id=52] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2026-01-02 14:31:37.633+0000 [id=72] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2026-01-02 14:31:39.019+0000 [id=65] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2026-01-02 14:31:39.029+0000 [id=58] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2026-01-02 14:31:39.030+0000 [id=68] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2026-01-02 14:31:39.358+0000 [id=52] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2026-01-02 14:31:39.359+0000 [id=65] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2026-01-02 14:31:39.360+0000 [id=71] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2026-01-02 14:31:39.362+0000 [id=55] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2026-01-02 14:31:39.400+0000 [id=87] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2026-01-02 14:31:39.912+0000 [id=64] INFO jenkins.install.SetupWizard#init:

[CRLF]>
[CRLF]> *****
[CRLF]> *****
[CRLF]> *****
[CRLF]>
[CRLF]> Jenkins initial setup is required. An admin user has been created and a password generated.
[CRLF]> Please use the following password to proceed to installation:
[CRLF]>
[CRLF]> 2a8107bf414a4ea8b729cf86c9573781
```

```
[CRLF]>
[CRLF]> This may also be found at: C:\Users\sweta\.jenkins\secrets\initialAdminPassword
[CRLF]>
[CRLF]> *****
[CRLF]> *****
[CRLF]> *****

2026-01-02 14:31:45.394+0000 [id=64] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2026-01-02 14:31:45.429+0000 [id=41] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2026-01-02 14:31:46.966+0000 [id=87] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2026-01-02 14:31:46.967+0000 [id=87] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
```

Password: **2a8107bf414a4ea8b729cf86c9573781**

```
This may also be found at: C:\Users\sweta\.jenkins\secrets\initialAdminPassword
```

Save this password


Open Local Host: <http://localhost:8080>

Follow the steps, Add User name password, email and continue. By default local host 8080 will be there

# Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

 **Jenkins**

[+ New Item](#)

[Build History](#)

Build Queue

No builds in the queue.

Build Executor Status

0/2

## Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

### Start building your software project

Create a job

### Set up a distributed build

Set up an agent

Configure a cloud

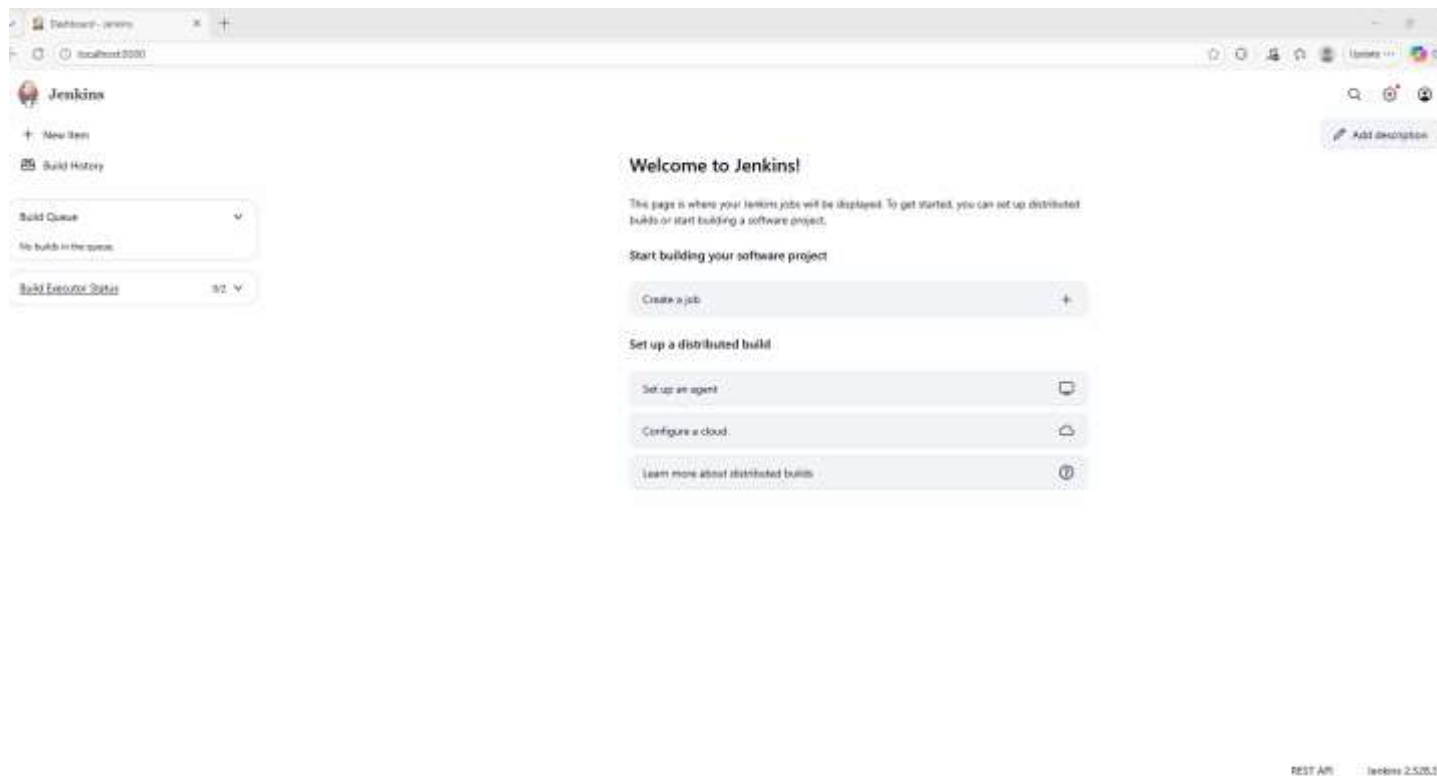
Learn more about distributed builds

[Add description](#)


## Dashboard - Jenkins

## Jenkins for Beginners

Jenkins UI First Screen after logging in:




## Goto Settings and then Systems

Jenkins / Manage Jenkins / SystemQ⚙️

### System

Home directory ?

By default, Jenkins stores all of its data in this directory on the file system

C:\Users\sweta\.jenkins 

System Message

This message will be displayed at the top of the Jenkins main page. This can be useful for posting notifications to your users

```
<b>This Jenkins server is for learning</b>
<h1>Hello Sweta</h1>
```

# of executors

5

This Jenkins server is for learning

Hello Sweta

Build Queue



Build Executor Status



(0 of 5 executors busy)

## Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

### Start building your software project

Create a job



### Set up a distributed build

Set up an agent



Configure a cloud



Learn more about distributed builds





## Tools

### Maven Configuration

Default settings provider

Use default maven settings ▾

Default global settings provider

Use default maven global settings ▾

### JDK installations

JDK installations ▴

Edited

+ Add JDK

≡ **JDK**



Name

JAVA\_HOME

JAVA\_HOME

C:\Program Files\Java\jdk-21

☐ Install automatically ?

## CREATING A JOB:

1. GO TO CREATE A JOB
2. AS AN EXAMPLE YOU MAY SELECT FREESTYLE PROJECT

### New Item

Enter an item name

JenkinsFirstJob

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

3. CLICK OK

Jenkins JenkinsFirstJob Configuration

Configure

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

General

Enabled

Description

Safe HTML Preview

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Advanced

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☒ None

☐ Git ?

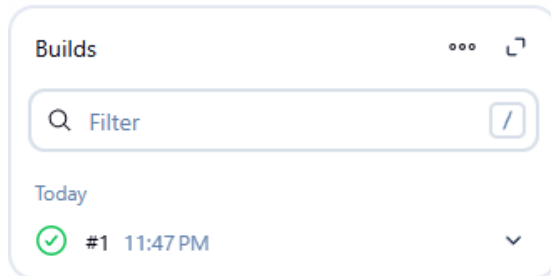
Save Apply

Click on Build Steps:  
Select Execute Windows Batch Command

Inside the box - you can write java -version  
Apply



Click on Build now  
Click on the green tick:



You can view the console output:

#### Console Output

```
Started by user sweta bhattacharya
Running as SYSTEM
Building in workspace C:\Users\sweta\.jenkins\workspace\JenkinsFirstJob
[JenkinsFirstJob] $ cmd /c call C:\Users\sweta\AppData\Local\Temp\jenkins16118012837783788400.bat

C:\Users\sweta\.jenkins\workspace\JenkinsFirstJob>java -version
java version "21.0.7" 2025-04-15 LTS
Java(TM) SE Runtime Environment (build 21.0.7+8-LTS-245)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.7+8-LTS-245, mixed mode, sharing)

C:\Users\sweta\.jenkins\workspace\JenkinsFirstJob>exit 0
Finished: SUCCESS
```

You can go to Dashboard by clicking on Jenkins on Left side

It will display the first job in the dashboard

This Jenkins server is for learning  
Hello Sweta

[Add description](#)

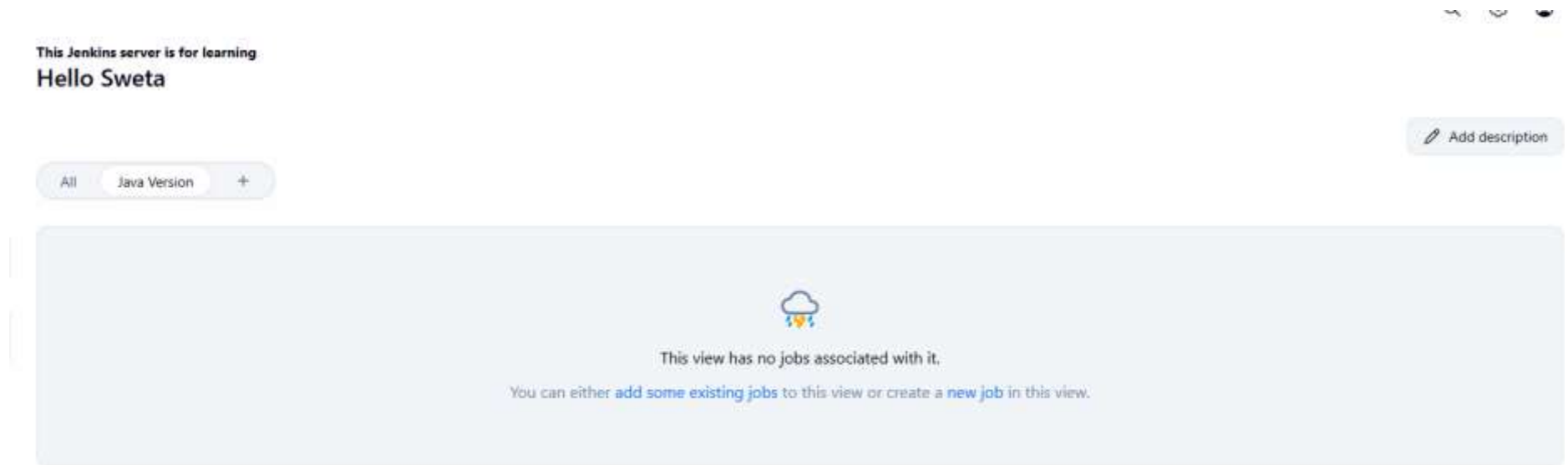
All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	JenkinsFirstJob	4 min 5 sec #1	N/A	1.2 sec	▶

Icons: S M L

You can change the view

1. Click on Add View
2. Give a name of the view – Eg: Java Version
3. You can remove weather by clicking on the “x”
4. Apply and Save



Click on Add some existing jobs:

Select the JenkinsFirstJob

Apply and save – The view without weather will appear as shown below:

All

Java Version

+

Add description

S	Name	Last Success	Last Failure	Last Duration	
	<a href="#">JenkinsFirstJob</a>	9 min 49 sec <a href="#">#1</a>	N/A	1.2 sec	

Icons:

S

M

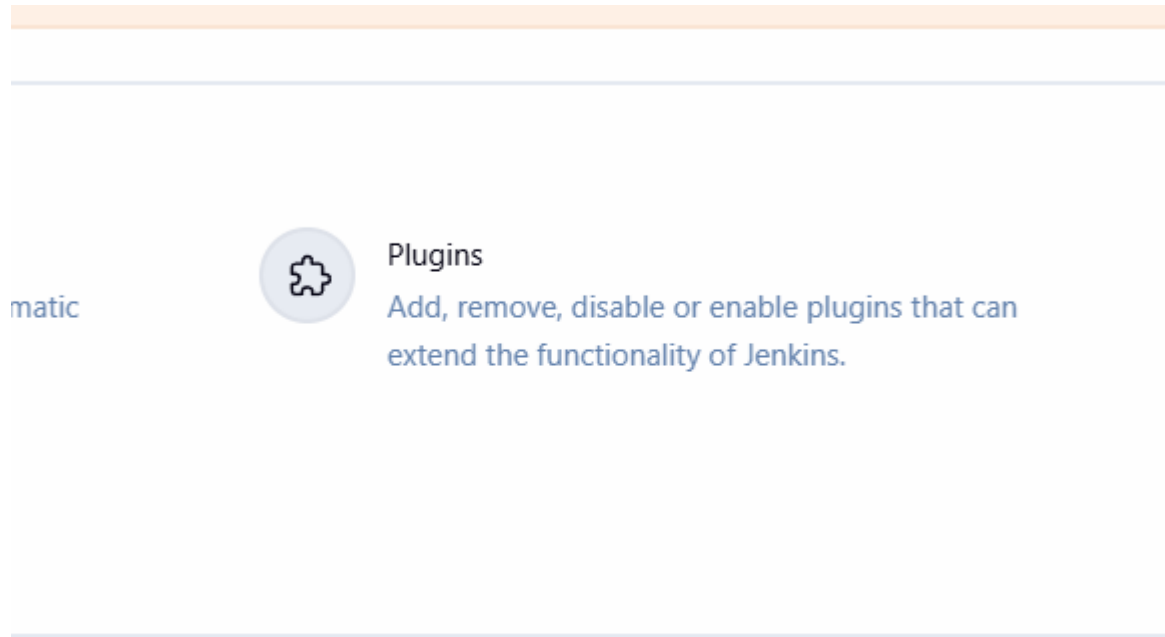
L

...

Add Pluggins:



Click on Plugins:



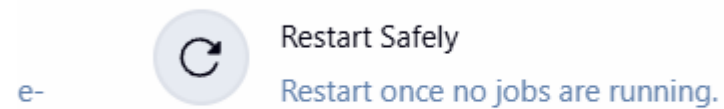
You can check installed plugins

You can click on available plugins – Type Safe restart

Select the check box on left of “Safe Restart” and select Install on Right

The plugin gets installed

The icon will appear in the Manage Jenkins



Now you can also Uninstall

Then restart Jenkins from the CMD prompt

Refresh the page

Then it will reappear under the Available Plugins



Jenkins

Manage Jenkins

Plugins

## Plugins



Updates



Available plugins



Installed plugins



Advanced settings

safe

Install

Name

Released



[Safe Restart](#) 102.v4dc1b\_9636a\_ee

[Miscellaneous](#)

This plugin allows you to restart Jenkins safely.

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

11 mo ago


## Advanced settings

The Proxy configuration form has been moved to [Configure System page](#)

### Deploy Plugin

You can select a plugin file from your local system or provide a URL to install a plugin from outside the configured update site(s).

File

 Choose File No file chosen

Or

URL

Deploy

### Update Site

URL

Submit

You can also install the plugin in your local machine and click on choose file

## Create a Java Program Job:

### New Item

Enter an item name

FirstJavaJob

Select an item type



#### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



#### 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.



#### 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.



#### Multibranch Pipeline

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



#### Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

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

Copy from

OK

Click OK

## Step 1: Add Description

### General

Enabled



#### Description

Jenkins Job running periodically - Java app

## Add Trigger:

### Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

\* \* \* \* \*

⚠ Do you really mean "every minute" when you say "\* \* \* \* \*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour

Would last have run at Sunday, 4 January, 2026, 1:44:00 am India Standard Time; would next run at Sunday, 4 January, 2026, 1:45:00 am India Standard Time.

## Add Build Steps – Windows Batch command

### Build Steps



Automate your build process with ordered tasks like code compilation, testing, and deployment.


 **Execute Windows batch command** 



### Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.


 **Execute Windows batch command** 



Command

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

```
javac Launch.java
java Launch
```

Advanced 

Click Apply ok

Come to – FirstJavaJob screen and click on Build Now

So the Build runs every second

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Builds

Filter

Today

#3 1:50 AM

#2 1:50 AM

#1 1:50 AM

✖

FirstJavaJob

Jenkins Job running periodically - Java app

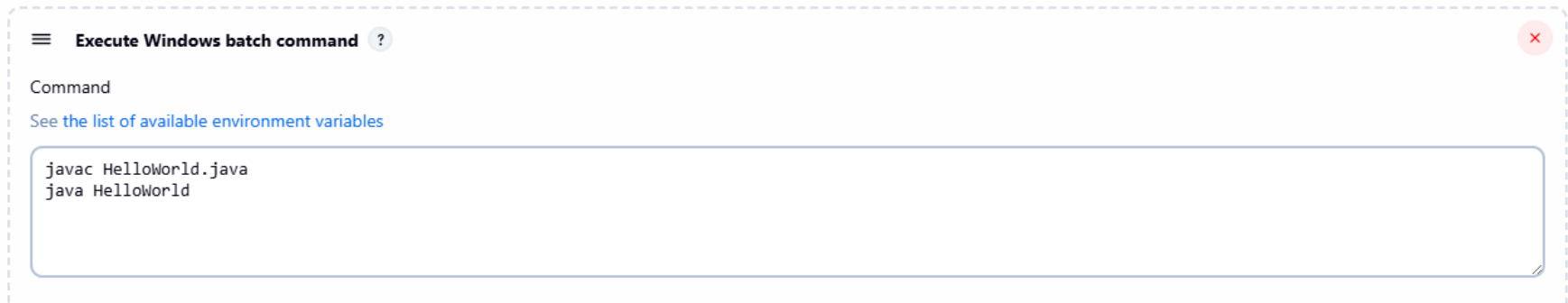
Permalinks

- Last build (#3), 14 sec ago
- Last failed build (#3), 14 sec ago
- Last unsuccessful build (#3), 14 sec ago
- Last completed build (#3), 14 sec ago

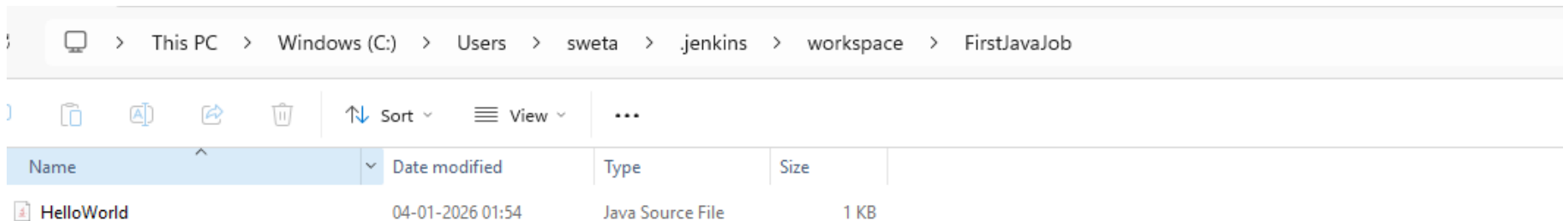
Reason:

```
C:\Users\sweta\.jenkins\workspace\FirstJavaJob>java Launch
Error: Could not find or load main class Launch
Caused by: java.lang.ClassNotFoundException: Launch
```

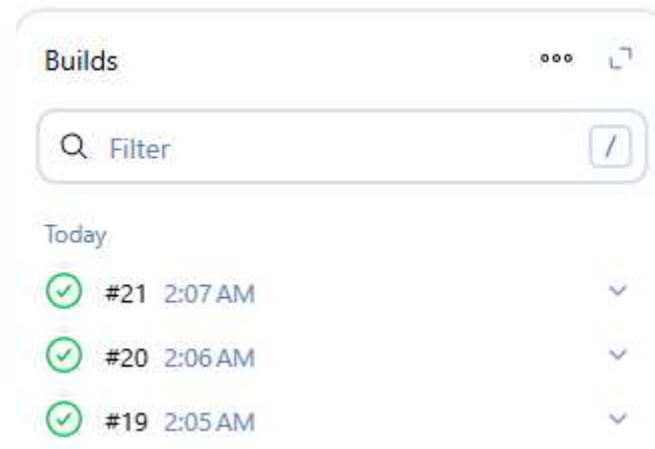
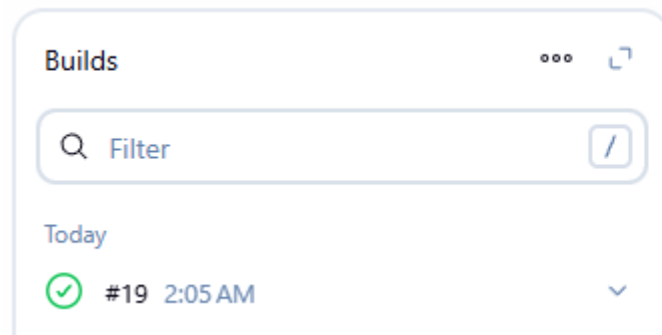
Now I create a helloworld program



And then paste the .java file in the location where jenkins searches for the file –



Click on Build Now



Console Output:



## Console Output

```
Started by timer
Running as SYSTEM
Building in workspace C:\Users\sweta\.jenkins\workspace\FirstJavaJob
[FirstJavaJob] $ cmd /c call C:\Users\sweta\AppData\Local\Temp\jenkins6552117785408533484.bat

C:\Users\sweta\.jenkins\workspace\FirstJavaJob>javac HelloWorld.java

C:\Users\sweta\.jenkins\workspace\FirstJavaJob>java HelloWorld
Hello, World!

C:\Users\sweta\.jenkins\workspace\FirstJavaJob>exit 0
Finished: SUCCESS
```

Create a Repository and add a program in GitHub

We have created a Program-1 repository

We have further created a hello.py program inside it

Copy the link of the repository in Github

The screenshot shows the GitHub interface for a repository named 'Program-1', which is public. At the top, there are buttons for 'Pin', 'Watch' (0), 'Fork' (0), and 'Star' (0). Below the repository name, it shows 'main' branch, '1 Branch', and '0 Tags'. A search bar 'Go to file' and buttons 'Add file' and '<> Code' are present. The commit history shows two commits: 'Initial commit' for 'README.md' and 'Add Hello World print statement' for 'hello.py', both made 1 minute ago. The 'About' section on the right states 'No description, website, or topics provided.' and lists 'Readme', 'Activity', '0 stars', '0 watching', and '0 forks'. The 'Releases' section shows 'No releases published' and a link to 'Create a new release'. The main content area displays the 'README' file with the title 'Program-1'.

Program-1 Public

Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file <> Code

swetab-max Add Hello World print statement 552770e · 1 minute ago 2 Commits

README.md	Initial commit	1 minute ago
hello.py	Add Hello World print statement	1 minute ago

README

## Program-1

About

No description, website, or topics provided.

- Readme
- Activity
- 0 stars
- 0 watching
- 0 forks

Releases

No releases published

[Create a new release](#)

Then in Jenkins:

Create New job as done earlier

Configure the job as shown below:

1. Add Description:

General

Enabled



Description

GitHub Project

2. Select GitHub Project – Paste the GitHub Repo URL : <https://github.com/swetab-max/Program-1.git/>

3. Source Code Management – Paste the GitHub Repo URL : <https://github.com/swetab-max/Program-1.git/>

Git ?

Repositories ?

Repository URL ?

[https://github.com/swetab-max/Program-1.git](https://github.com/swetab-max/Program-1.git/)

4. For Credentials – Click on Add and Select Jenkins

5. In the Jenkins Credential Provider Screen: Add Jenkins Username and Password

Jenkins Credentials Provider: Jenkins

### Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

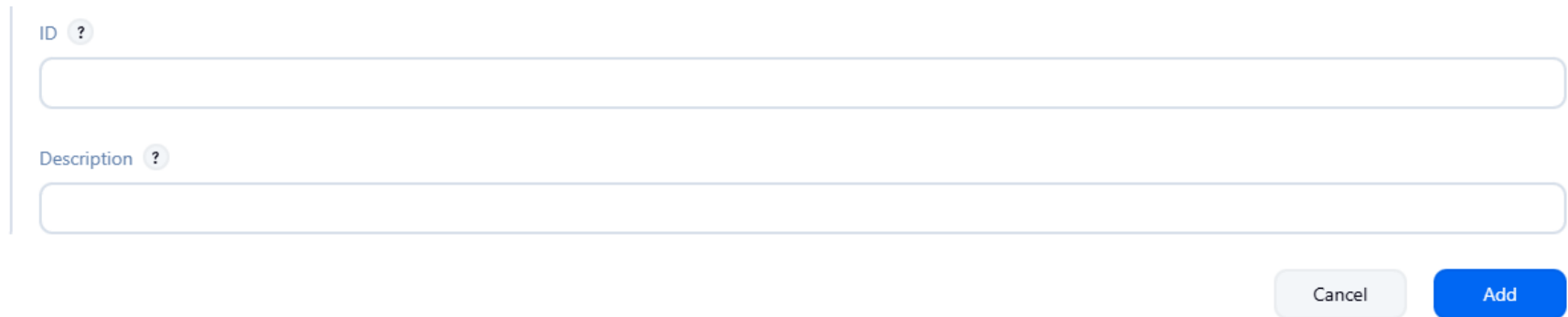
Username

Blank username; did you mean to use secret text credentials instead?

☐ Treat username as secret ?

Password

6. Give any ID and Description of your choice and click on Add



A form for adding a new credential. It features two input fields: 'ID' and 'Description', each with a question mark icon to its left. Below the fields are two buttons: a light gray 'Cancel' button and a blue 'Add' button.

7. Select the Credential from the drop down. The credential you just created will be visible, just click on it and select

8. For Branches to Build - Since my program is under main so update to main removing master



A form for specifying branches to build. It has a label 'Branches to build' with a question mark icon. Below it is a dashed border containing a 'Branch Specifier (blank for 'any')' label with a question mark icon and an input field containing the text '\*/main'. A red 'X' icon is in the top right corner of the dashed box.

9. For Triggers – Select Poll SCM and specify the duration / frequency of running the build. In this case, any commit when done in Git will trigger the Build run here in Jenkins.

#### Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

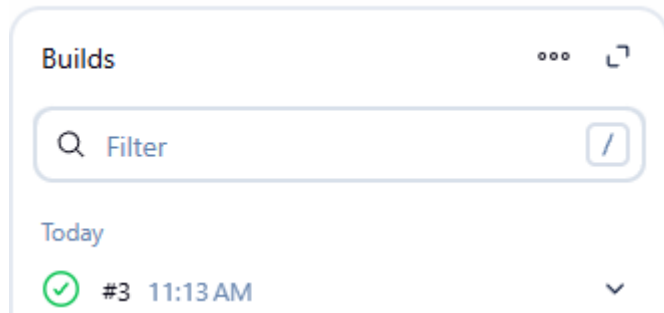
- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☒ Poll SCM ?

Schedule ?

\*\*\*\*\*

10. For Environment select Delete workspace before build starts
11. For Build Steps – Select Execute Windows Batch Command – Write the execution command for your program in Git – python hello.py
12. Apply and Save
13. Click on Build Now

## 14. Build will run successfully and console Output will show Success



```
C:\Users\sweta\.jenkins\workspace\GITProject1>python hello.py  
Hello, World!
```

```
C:\Users\sweta\.jenkins\workspace\GITProject1>exit 0  
Finished: SUCCESS
```

# Build CI/CD Pipeline

1. Go to Manage Jenkins – Plugins – Available plugins
2. Search for Build Pipeline Plugin and install

Install

Name ↓

Released

Health

<input checked="" type="checkbox"/>	<div><div>Build Pipeline 2.1.0</div><div>User Interface   Build Tools   Other Post-Build Actions</div><div>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.</div><div>This plugin is up for adoption! We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.</div></div>	1 day 11 hr ago	<div>80</div>
<input type="checkbox"/>	<div><div>Webhook Step 3.42.v620877fe1e14</div><div>Allows build pipelines to wait for notification from an external system before continuing.</div></div>	1 yr 11 mo ago	<div>89</div>
<input type="checkbox"/>	<div><div>Pipeline timeline 1.0.3</div><div>An interactive build timeline to help you visualize your build pipeline and identify bottlenecks.</div></div>	6 yr 11 mo ago	<div>82</div>

## Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Parameterized Trigger 

Success

Build Pipeline 

Success

Loading plugin extensions 

Success

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

## Then Create a New Job

1. Create Freestyle Job: DEV\_Job
2. Follow the same steps in the previous section for adding the Git Repository URL and same URL in Source Code Management.
3. Select the same credentials that you selected earlier
4. Update the Branch Specifier to main
5. For Triggers – Select Poll SCM and add the five stars
6. For Environment – Delete workspace before build starts needs to be selected
7. For Build Steps select Execute Windows batch command
8. Type - python hello.py
9. Apply and Ok
10. The DEV\_Job will be displayed in the dashboard

Then create another job – Name it: QA\_Job, follow the same steps until Step 4.

5. For Triggers – Since this job should run only after DEV\_Job runs successfully, like a pipeline, so select “Build after other projects are built”

### Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Trigger builds remotely (e.g., from scripts) ?

☒ Build after other projects are built ?

Select DEV\_Job for Projects to watch

☒ Build after other projects are built ?

Projects to watch

DEV Job,

6. For Environment – Delete workspace before build starts needs to be selected

7. For Build Steps select Execute Windows batch command

8. Type - python add.py

9. Apply and Ok

10. The QA\_Job will be displayed in the dashboard

Now for the Dev\_Job add:

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

☰ Build other projects ?

Projects to build

QA\_Job,

Now click on + sign to add a new View

All Java Version +

Select Build Pipeline View

## New view

Name

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.



List View

Give the Name – I have given DEV\_QA\_Pipeline and click on Create

## New view

Name

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.

## In the Edit View Window Add Description

### Edit View

Name

DEV\_QA\_Pipeline

Description

Describe the purpose of this view.

The pipeline is for DEV and QA Jobs

In the pipeline flow: The first job or initial job should be selected which is our DEV\_Job

## 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 ?

DEV\_Job



No of Displayed Builds can be made as 5 or as per your requirement

No Of Displayed Builds ?

5



Row Headers

All build variables and parameters

Show all variables in the current build, and obfuscate sensitive values. Variables are drawn from Build#getBuildVariables().

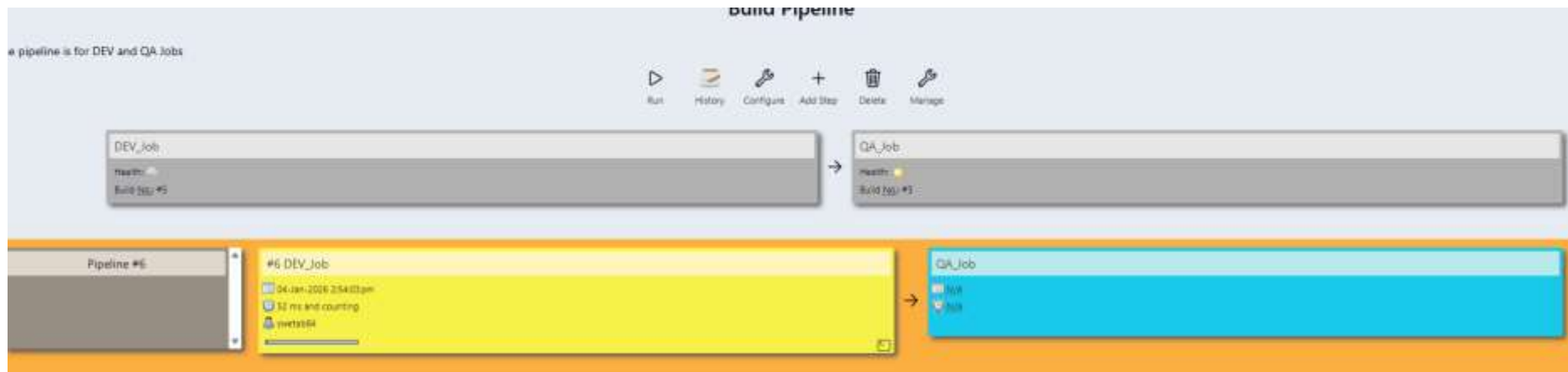
Column Headers

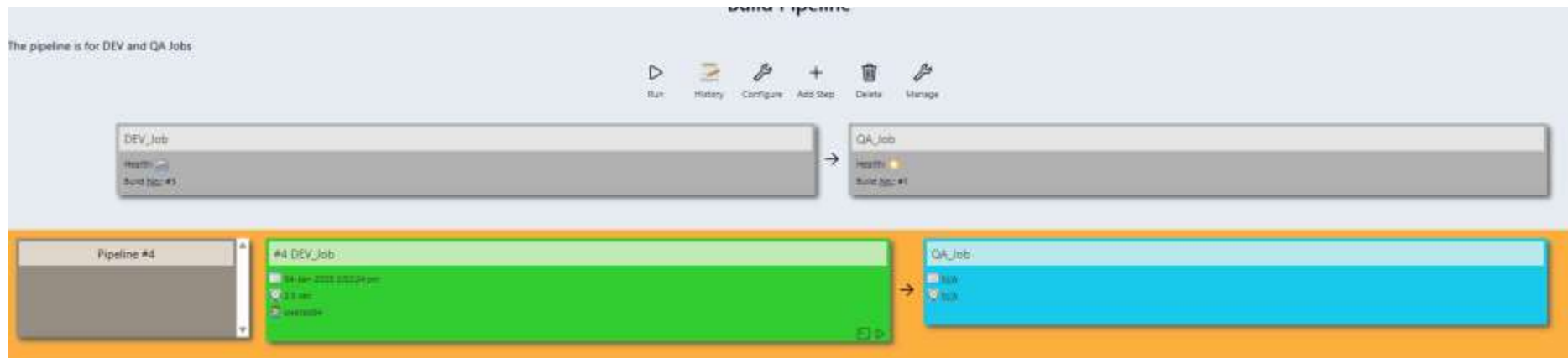
All build variables and parameters

Show all variables in the current build, and obfuscate sensitive values. Variables are drawn from Build#getBuildVariables().

Click on Apply and Save

The Build Pipeline will be shown click on run





Only when Dev\_Job is executed then only QA\_Job will run

