

AIR UNIVERSITY, ISLAMABAD

Department of Cyber Security

Secure Software Design & Development Lab (CY- 256L)

---

REG ID: 231312

---

# ASSIGNMENT III

---

Kiran Hashmi

---

BS-CYS 4 A

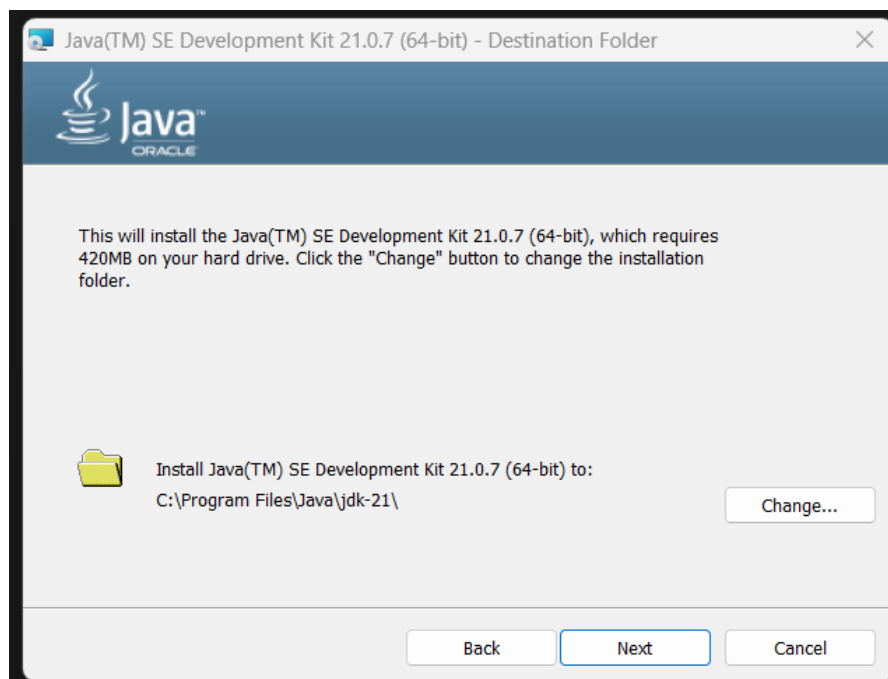
## What is Jenkins?

Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on. It is a free source that can handle any kind of build or continuous integration. You can integrate Jenkins with a number of testing and deployment

### Pre requisites:

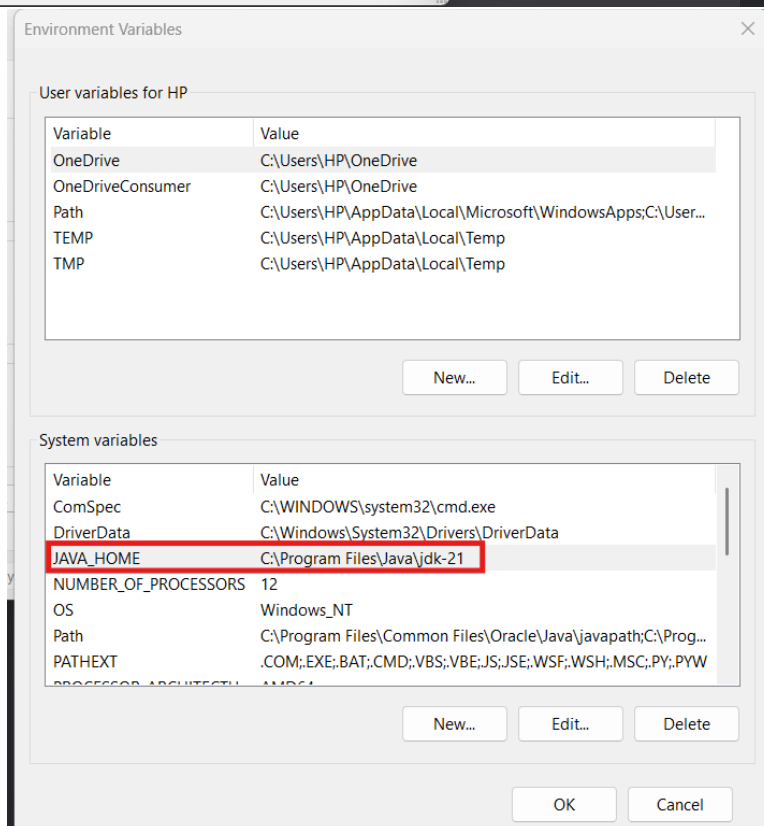
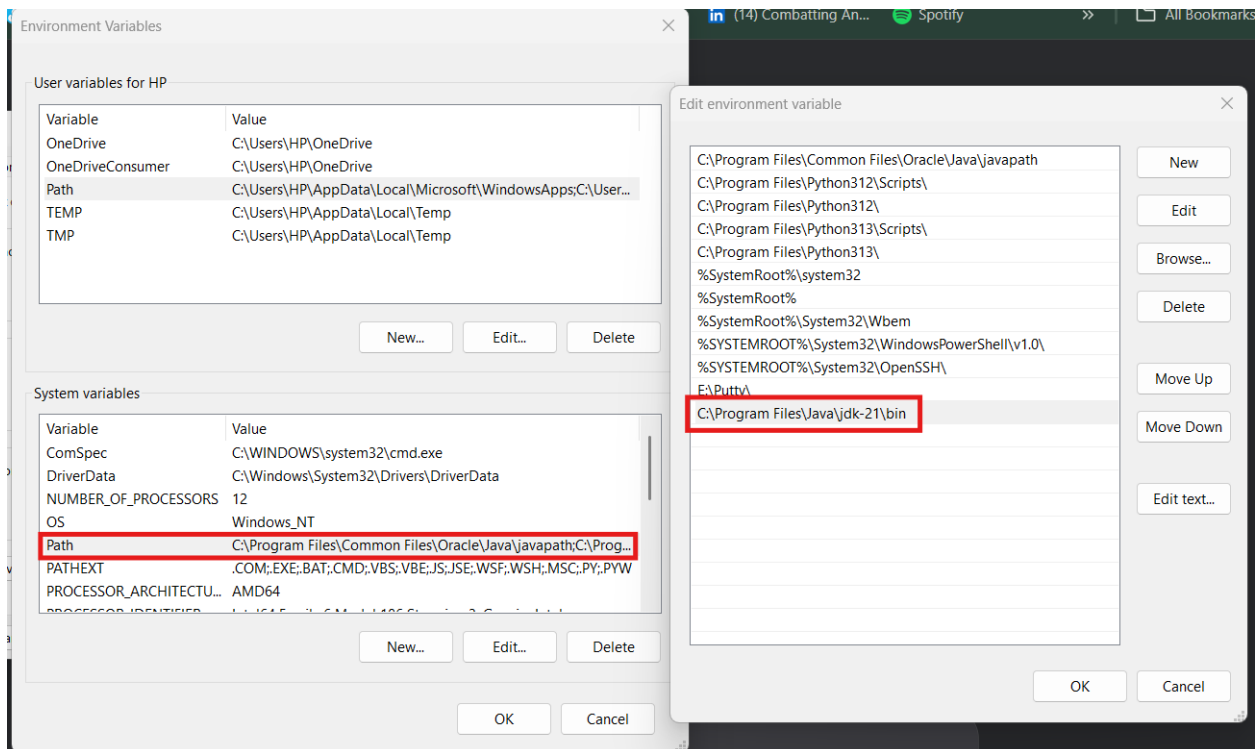
Java should be installed on your machine.

### Installation of JAVA:



### Configuring java:

Submitted to :Mr. Mahaz Khan



Installed Java:

Date of Submission: 17/06/2025

Submitted to :Mr. Mahaz Khan

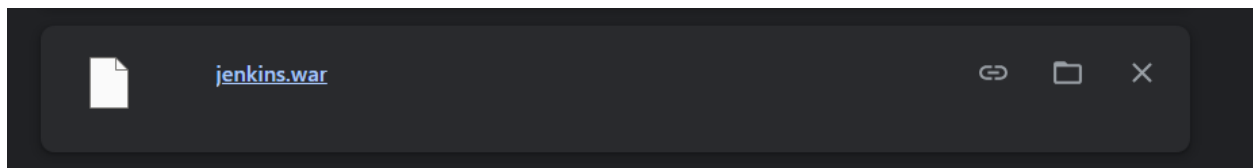
```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>java --version
java 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\HP>javac --version
javac 21.0.7

C:\Users\HP>
```

## Download Jenkins:



## Starting Jenkins

Open the command prompt. From the command prompt, browse to the directory where the jenkins.war file is present. Run the following command

## Java -jar Jenkins.war

```
C:\Users\HP>cd Downloads

C:\Users\HP\Downloads>Java -jar jenkins.war
Running from: C:\Users\HP\Downloads\jenkins.war
webroot: C:\Users\HP\jenkins\war
2025-06-13 12:53:28.924+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2025-06-13 12:53:29.906+0000 [id=1] WARNING o.e.jee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-06-13 12:53:29.954+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.22; built: 2025-06-02T15:25:31.946Z; git: 335c9ab44a5591f0ea941bf350e139b8c4f5537c; jvm 21.0.7+8-LTS-245
2025-06-13 12:53:30.498+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.ee9.jsp.JettyJspServlet
2025-06-13 12:53:30.558+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-06-13 12:53:30.987+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\HP\jenkins found at: $user.home/.jenkins
2025-06-13 12:53:31.067+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@451f87af{Jenkins v2.514.0, b-file:///C:/Users/HP/Downloads/jenkins.war, a=AVAILABLE, h=oeje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@2b556bb2{STARTED}}
2025-06-13 12:53:31.083+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@6a9950f1[HTTP/1.1, (http/1.1){0.0.0.0:8080}]
2025-06-13 12:53:31.095+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server@1fb19a0{STARTING}[12.0.22,sto=0] @2616ms
2025-06-13 12:53:31.241+0000 [id=42] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2025-06-13 12:53:31.334+0000 [id=52] INFO jenkins.model.Jenkins#<init>: Starting version 2.514
2025-06-13 12:53:31.334+0000 [id=52] INFO jenkins.InitReactorRunner$1onAttained: Started initialization
2025-06-13 12:53:31.358+0000 [id=72] INFO jenkins.InitReactorRunner$1onAttained: Listed all plugins
2025-06-13 12:53:32.890+0000 [id=62] INFO jenkins.InitReactorRunner$1onAttained: Prepared all plugins
2025-06-13 12:53:32.895+0000 [id=57] INFO jenkins.InitReactorRunner$1onAttained: Started all plugins
2025-06-13 12:53:32.895+0000 [id=61] INFO jenkins.InitReactorRunner$1onAttained: Augmented all extensions
2025-06-13 12:53:33.350+0000 [id=54] INFO jenkins.InitReactorRunner$1onAttained: System config loaded
2025-06-13 12:53:33.352+0000 [id=66] INFO jenkins.InitReactorRunner$1onAttained: System config adapted
2025-06-13 12:53:33.352+0000 [id=66] INFO jenkins.InitReactorRunner$1onAttained: Loaded all jobs
2025-06-13 12:53:33.352+0000 [id=51] INFO jenkins.InitReactorRunner$1onAttained: Configuration for all jobs updated
2025-06-13 12:53:33.470+0000 [id=87] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2025-06-13 12:53:34.260+0000 [id=69] INFO jenkins.install.SetupWizard#init:

*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
```

After the command is run, various tasks will run, one of which is the extraction of the war file which is done by an embedded webserver called winstone.

Date of Submission: 17/06/2025

Submitted to :Mr. Mahaz Khan

Once the processing is complete without major errors, the following line will come in the output of the command prompt.

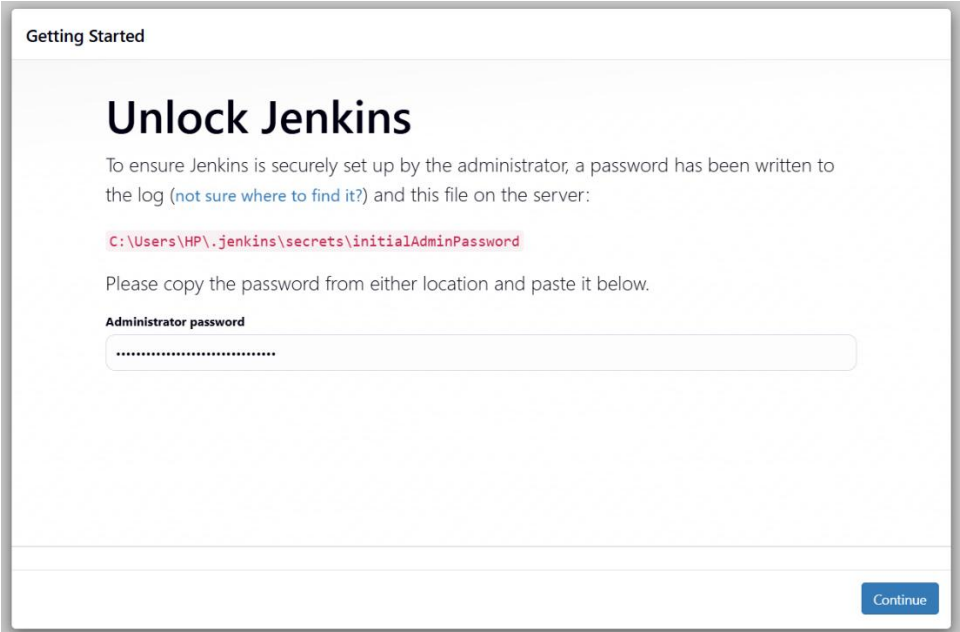
Copy the password and save it somewhere for later use.

## Accessing Jenkins

Once Jenkins is up and running, one can access Jenkins from the

link – <http://localhost:8080>

This link will bring up the Jenkins dashboard. Write password that you saved previously.

The image shows the 'Getting Started' screen of Jenkins. It has a title 'Unlock Jenkins' and a paragraph explaining that a password has been written to the log and a file on the server. The file path 'C:\Users\HP\.jenkins\secrets\initialAdminPassword' is highlighted in red. Below this, it says 'Please copy the password from either location and paste it below.' There is a label 'Administrator password' above a text input field that contains a series of dots. A 'Continue' button is located at the bottom right of the form.

Getting Started

## Unlock Jenkins

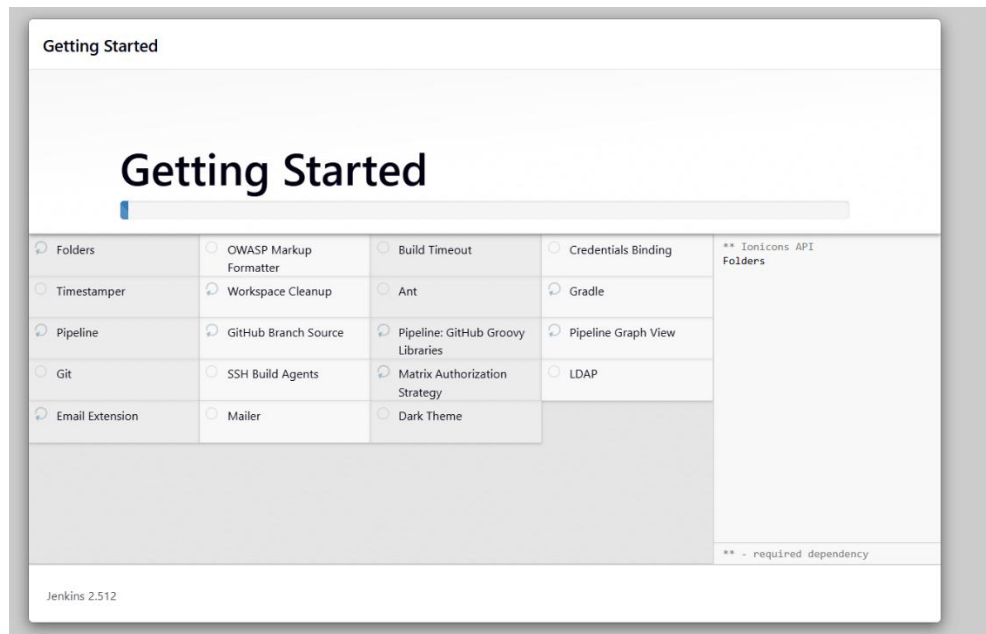
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\Users\HP\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Continue



Getting Started

## Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.514

[Skip and continue as admin](#) [Save and Continue](#)

## Getting Started

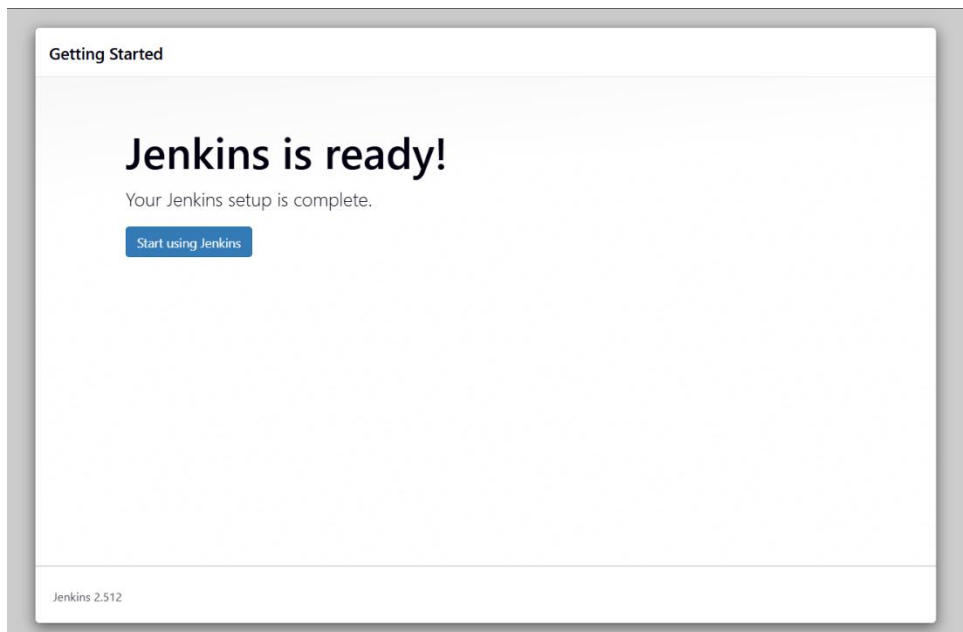
# Instance Configuration

Jenkins URL:

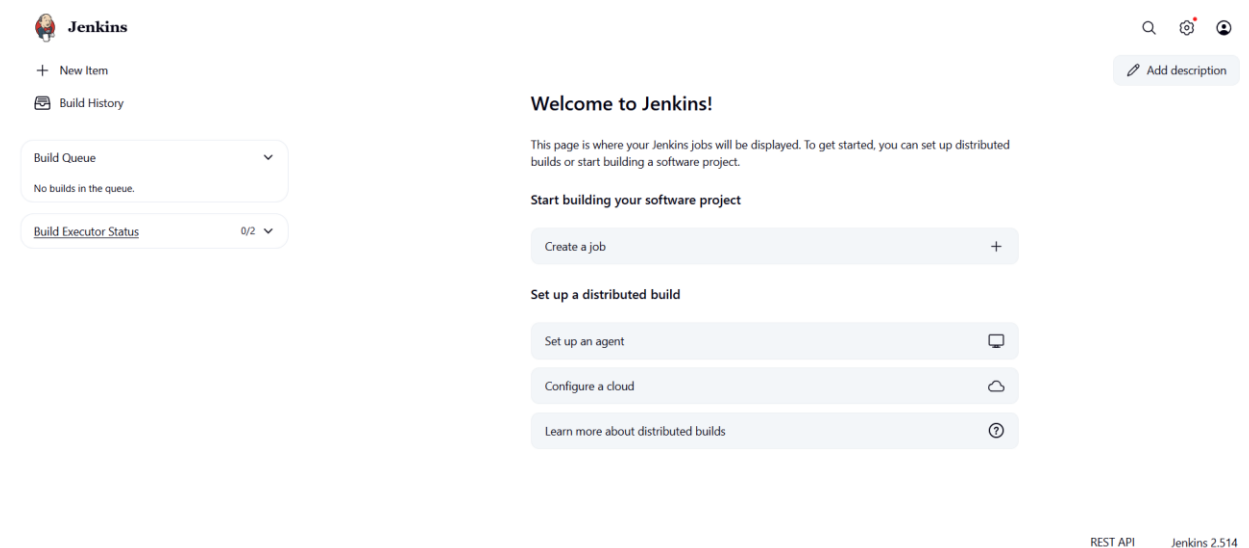
http://127.0.0.1:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD\_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.



Submitted to :Mr. Mahaz Khan



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with the Jenkins logo, a 'New Item' button, a 'Build History' button, and two status boxes: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). The main area has a 'Welcome to Jenkins!' message, followed by instructions on how to get started. Below this, there's a 'Start building your software project' section with a 'Create a job' button. Further down, there's a 'Set up a distributed build' section with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. In the top right corner, there are search, settings, and user icons, along with an 'Add description' button. At the bottom right, it says 'REST API' and 'Jenkins 2.514'.

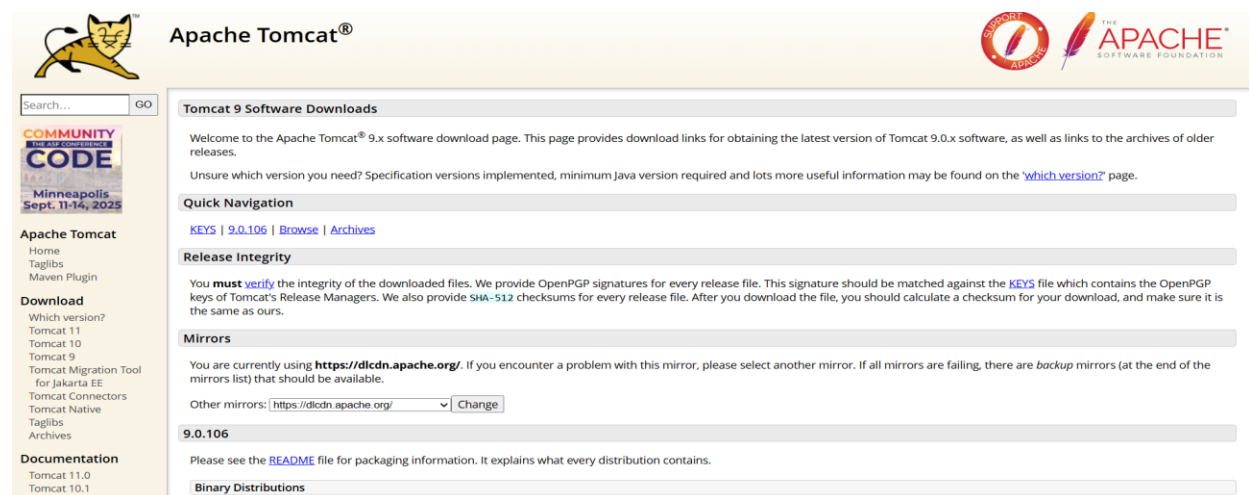
## What is Tomcat?

Tomcat is an open-source web server and servlet. The Apache Software Foundation has developed it. It is used widely for hosting Java-based applications on the web. It is built on Java technologies and implements the Java Servlet and JavaServer Pages (JSP) specifications. Tomcat acts as a bridge between web servers and Java-based applications, facilitating the execution of dynamic content and processing client requests.

## Download Tomcat

The official website for tomcat is Tomcat. If you click the given link, you can get the home page of the tomcat official website as shown below.

<https://tomcat.apache.org/download-90.cgi>



The screenshot shows the Apache Tomcat 9.0.106 download page. The header features the Apache Tomcat logo and the Apache Software Foundation logo. Below the header, there's a search bar and a 'GO' button. The main content area is titled 'Tomcat 9 Software Downloads' and includes a welcome message, a link to the 'which version?' page, and a 'Quick Navigation' section with links to 'KEYS', '9.0.106', 'Browse', and 'Archives'. There's also a 'Release Integrity' section with instructions on how to verify the integrity of the downloaded files. A 'Mirrors' section lists the current mirror and provides a link to the 'README' file. At the bottom, there's a 'Binary Distributions' section.

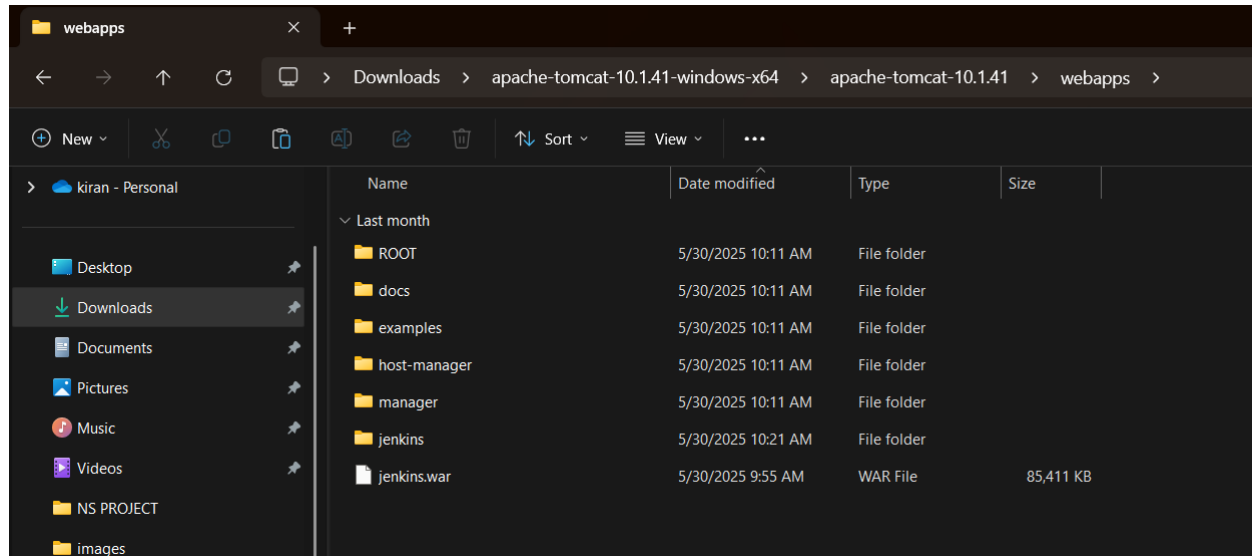
Date of Submission: 17/06/2025



Submitted to :Mr. Mahaz Khan

Go to the 'Binary Distributions' section. Download the 32-bit Windows zip file.

Then unzip the contents of the downloaded zip file.



## Jenkins and Tomcat Setup

Copy the Jenkins.war file which was downloaded from the previous section and copy it to the webapps folder in the tomcat folder.

Now open the command prompt. From the command prompt, browse to the directory where the tomcat7 folder is location. Browse to the bin directory in this folder and run the start.bat file

```
C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\bin>startup.bat
```

Date of Submission: 17/06/2025

```
C:\Users\HP>cd Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\bin

C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\bin>startup.bat
Using CATALINA_BASE:   "C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41"
Using CATALINA_HOME:   "C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41"
Using CATALINA_TMPDIR: "C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk-21"
Using CLASSPATH:        "C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\bin\bootstrap.jar;
C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""
C:\Users\HP\Downloads\apache-tomcat-10.1.41-windows-x64\apache-tomcat-10.1.41\bin>

Tomcat
at hudson.PluginManager$2$1$1.run(PluginManager.java:582)
at org.jvnet.hudson.reactor.TaskGraphBuilder$TaskImpl.run(TaskGraphBuilder.java:175)
at org.jvnet.hudson.reactor.Reactor.runTask(Reactor.java:304)
at jenkins.model.Jenkins$5.runTask(Jenkins.java:1149)
at org.jvnet.hudson.reactor.Reactor$2.run(Reactor.java:221)
at org.jvnet.hudson.reactor.Reactor$Node.run(Reactor.java:120)
at jenkins.security.ImpersonatingExecutorService$1.run(ImpersonatingExecutorService.java:68)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1144)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:642)
at java.base/java.lang.Thread.run(Thread.java:1583)
13-Jun-2025 18:10:33.859 INFO [pool-7-thread-22] jenkins.InitReactorRunner$1.onAttained Prepared all plugins
13-Jun-2025 18:10:33.950 INFO [pool-7-thread-17] jenkins.InitReactorRunner$1.onAttained Started all plugins
13-Jun-2025 18:10:33.962 INFO [pool-7-thread-14] jenkins.InitReactorRunner$1.onAttained Augmented all extensions
13-Jun-2025 18:10:35.024 INFO [GitSCM.onLoaded] hudson.plugins.build_timeout.global.GlobalTimeoutConfiguration.load glob
al timeout not set
13-Jun-2025 18:10:38.660 INFO [pool-7-thread-7] jenkins.InitReactorRunner$1.onAttained System config loaded
13-Jun-2025 18:10:38.660 INFO [pool-7-thread-7] jenkins.InitReactorRunner$1.onAttained System config adapted
13-Jun-2025 18:10:38.771 INFO [pool-7-thread-4] jenkins.InitReactorRunner$1.onAttained Loaded all jobs
13-Jun-2025 18:10:38.812 INFO [pool-7-thread-1] jenkins.InitReactorRunner$1.onAttained Configuration for all jobs update
d
13-Jun-2025 18:10:38.919 INFO [pool-7-thread-20] jenkins.InitReactorRunner$1.onAttained Completed initialization
13-Jun-2025 18:10:39.063 INFO [Jenkins initialization thread] hudson.lifecycle.Lifecycle.onReady Jenkins is fully up and
running
```

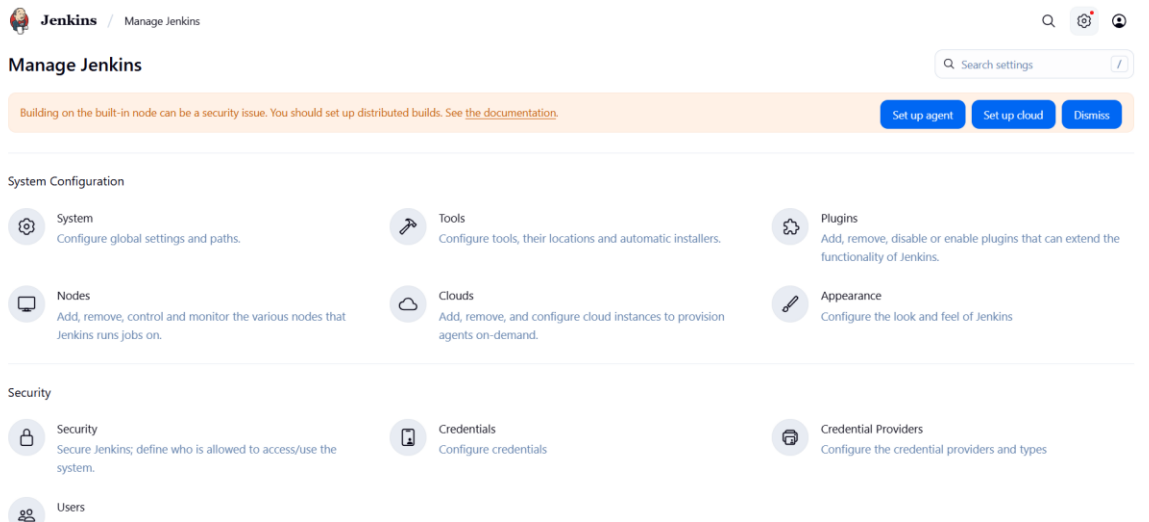
Open the browser and go to the link – <http://localhost:8080/jenkins>.

Jenkins will be up and running on tomcat.

## Jenkins - Git Setup

For this exercise, you have to ensure that Internet connectivity is present from the machine on which Jenkins is installed. In your Jenkins Dashboard (Home screen), click the Manage Jenkins option on the left hand side

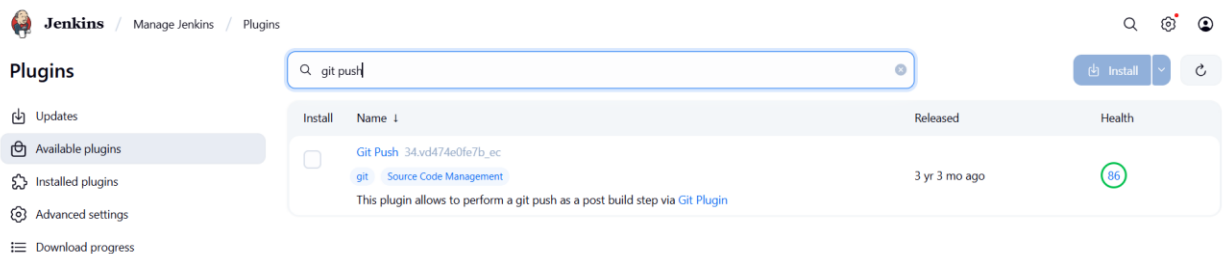
Submitted to :Mr. Mahaz Khan



The screenshot shows the Jenkins 'Manage Jenkins' interface. At the top, there's a navigation bar with the Jenkins logo and 'Manage Jenkins' text. Below this, a search bar is visible. A prominent orange warning banner states: 'Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.' To the right of the banner are buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'. The main content area is divided into two sections: 'System Configuration' and 'Security'. Under 'System Configuration', there are six options: System (Configure global settings and paths), Tools (Configure tools, their locations and automatic installers), Plugins (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), Nodes (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), Clouds (Add, remove, and configure cloud instances to provision agents on-demand), and Appearance (Configure the look and feel of Jenkins). Under the 'Security' section, there are three options: Security (Secure Jenkins; define who is allowed to access/use the system), Credentials (Configure credentials), and Credential Providers (Configure the credential providers and types). A 'Users' link is also visible at the bottom left of the security section.

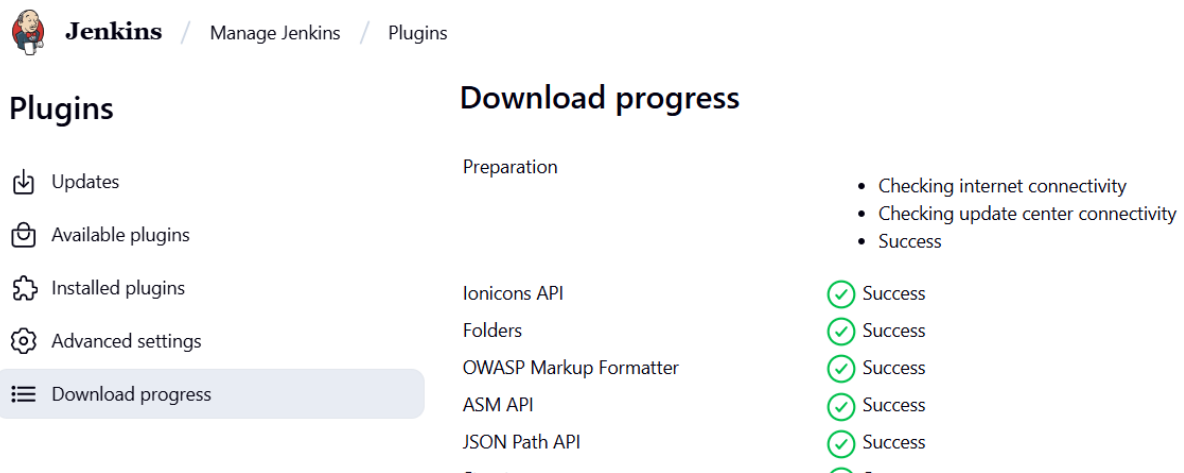
In the next screen, click the 'Plugins' option.

In the next screen, click the Available tab. This tab will give a list of plugins which are available for downloading. In the 'Filter' tab type 'Git Push'



The screenshot shows the Jenkins 'Plugins' page. The breadcrumb navigation at the top reads 'Jenkins / Manage Jenkins / Plugins'. On the left, there's a sidebar with tabs: 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area features a search bar with 'git push' entered. Below the search bar is a table of available plugins. The table has columns for 'Install', 'Name', 'Released', and 'Health'. One plugin is listed: 'Git Push' with version '34.vd474e0fe7b\_ec', released '3 yr 3 mo ago', and a health status of '86'. A description below the table states: 'This plugin allows to perform a git push as a post build step via Git Plugin'. There are 'Install' and 'Refresh' buttons at the top right of the plugin list.

The installation will then begin and the screen will be refreshed to show the status of the download.



The screenshot shows the Jenkins 'Plugins' and 'Download progress' page. The breadcrumb navigation at the top reads 'Jenkins / Manage Jenkins / Plugins'. On the left, the same sidebar as the previous screen is visible, with 'Download progress' now selected. The main area is split into two columns. The left column, titled 'Plugins', lists the following items: 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (which is highlighted). The right column, titled 'Download progress', shows the status of various components. Under the 'Preparation' section, there are three items: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this, a list of plugins is shown with their status: 'Ionicons API' (Success), 'Folders' (Success), 'OWASP Markup Formatter' (Success), 'ASM API' (Success), and 'JSON Path API' (Success). Each item is preceded by a green checkmark icon.

Date of Submission: 17/06/2025

Once all installations are complete, restart Jenkins by issue the following command in the

browser. `http://localhost:8080/jenkins/restart`

## New Item

Enter an item name

Kiran-Job

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

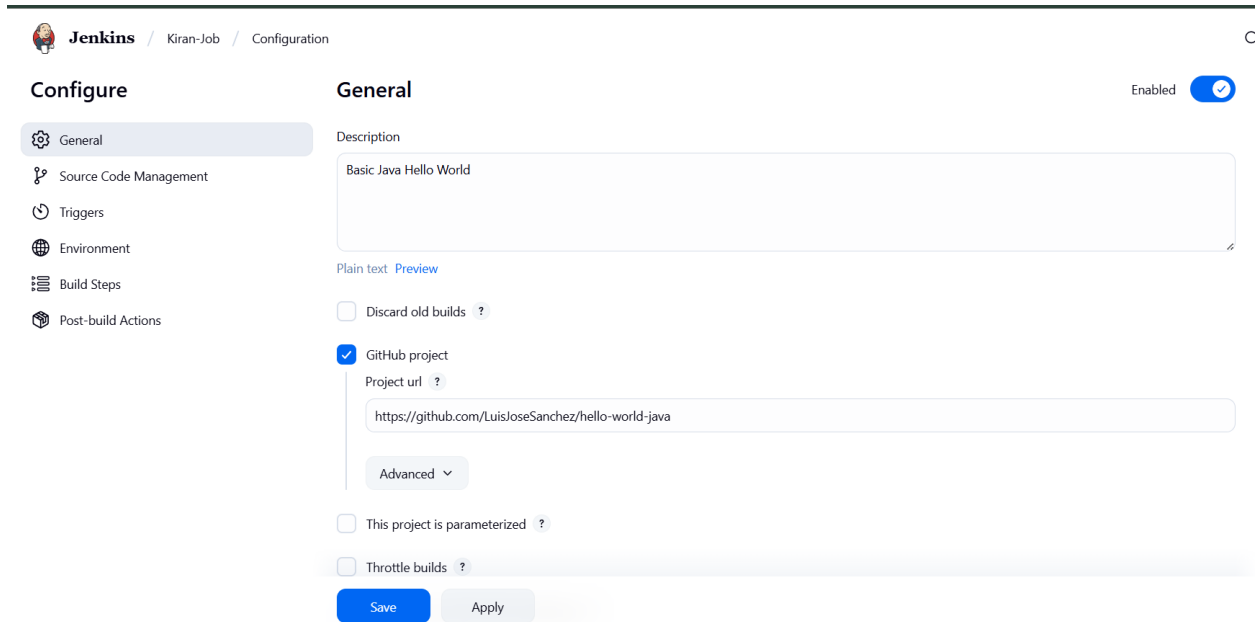
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

## Jenkins - Setup Build Jobs

For this exercise, we will create a job in Jenkins which picks up a simple HelloWorld application, builds and runs the java program.

The following screen will come up in which you can specify the details of the job.



The screenshot shows the Jenkins configuration page for a job named 'Kiran-Job'. The 'General' tab is selected in the left sidebar. The 'Description' field contains 'Basic Java Hello World'. The 'GitHub project' checkbox is checked, and the 'Project url' is set to 'https://github.com/LuisJoseSanchez/hello-world-java'. The 'Enabled' toggle is turned on. At the bottom, there are 'Save' and 'Apply' buttons.

Jenkins / Kiran-Job / Configuration

Configure

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

General

Enabled ☒

Description

Basic Java Hello World

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ?

https://github.com/LuisJoseSanchez/hello-world-java

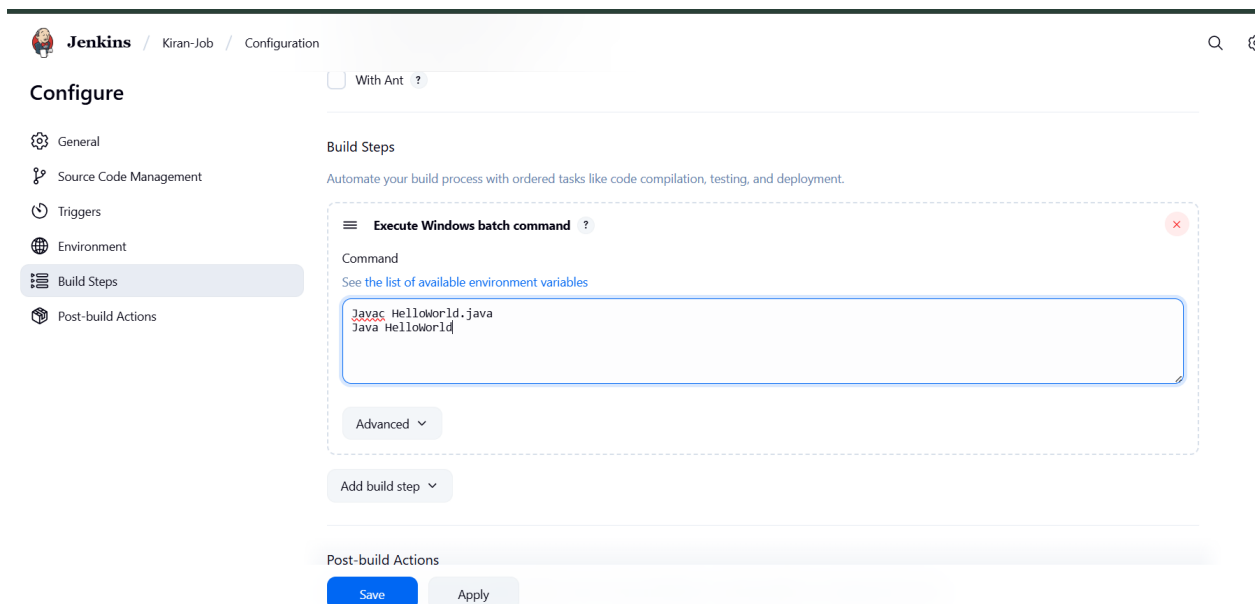
Advanced ▾

☐ This project is parameterized ?

☐ Throttle builds ?

Save Apply

We need to specify the location of files which need to be built. If you repository is hosted on Github, you can also enter the url of that repository here:



The screenshot shows the Jenkins configuration page for the same job, but the 'Build Steps' tab is selected. A new build step 'Execute Windows batch command' has been added. The 'Command' field contains 'javac HelloWorld.java' and 'java HelloWorld'. The 'Add build step' button is visible at the bottom.

Jenkins / Kiran-Job / Configuration

Configure

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

☐ With Ant ?

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 HelloWorld.java  
java HelloWorld

Advanced ▾

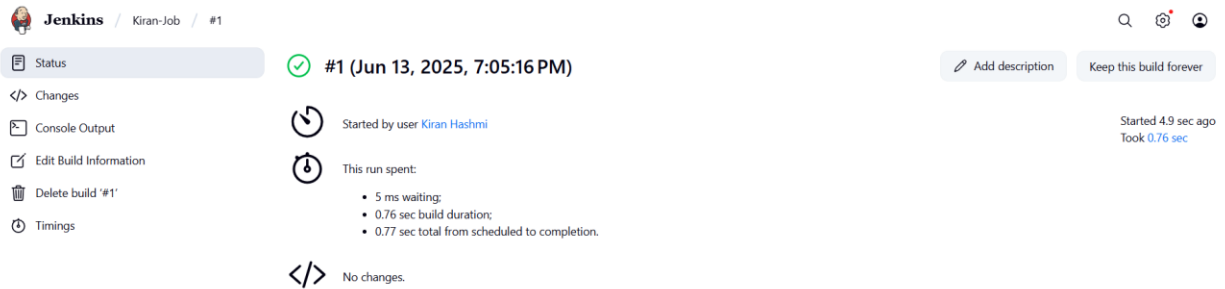
Add build step ▾

Post-build Actions

Save Apply

Now go to the Build section and click on Add build step → Execute Windows batch command

Submitted to :Mr. Mahaz Khan



This screenshot shows the Jenkins build summary for build #1 of the 'Kiran-Job'. The build is successful, indicated by a green checkmark. It was started by user 'Kiran Hashmi' on June 13, 2025, at 7:05:16 PM. The console output tab is selected, showing the build process. The build took 0.76 seconds to complete. The console output shows the build starting in the workspace, running as SYSTEM, and executing a command to call a batch file. The batch file then runs 'javac HelloWorld.java' and 'java HelloWorld', outputting 'Hello world!'. The build finishes successfully with exit code 0.

Jenkins / Kiran-Job / #1

Status #1 (Jun 13, 2025, 7:05:16 PM)

Started by user [Kiran Hashmi](#)

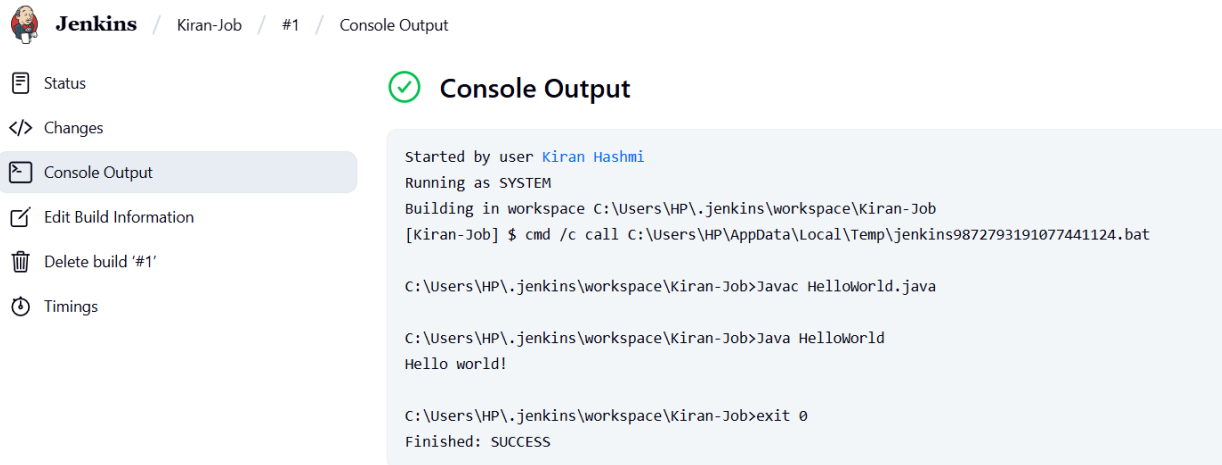
This run spent:

- 5 ms waiting;
- 0.76 sec build duration;
- 0.77 sec total from scheduled to completion.

No changes.

Started 4.9 sec ago  
Took 0.76 sec

Add description Keep this build forever



This screenshot shows the Jenkins console output for build #1 of the 'Kiran-Job'. The console output is displayed in a light blue box. It shows the build starting in the workspace, running as SYSTEM, and executing a command to call a batch file. The batch file then runs 'javac HelloWorld.java' and 'java HelloWorld', outputting 'Hello world!'. The build finishes successfully with exit code 0.

Jenkins / Kiran-Job / #1 / Console Output

Status Console Output

Started by user [Kiran Hashmi](#)

Running as SYSTEM

Building in workspace C:\Users\HP\.jenkins\workspace\Kiran-Job

[Kiran-Job] \$ cmd /c call C:\Users\HP\AppData\Local\Temp\jenkins9872793191077441124.bat

C:\Users\HP\.jenkins\workspace\Kiran-Job>javac HelloWorld.java

C:\Users\HP\.jenkins\workspace\Kiran-Job>java HelloWorld

Hello world!

C:\Users\HP\.jenkins\workspace\Kiran-Job>exit 0

Finished: SUCCESS

successfully defined the job.

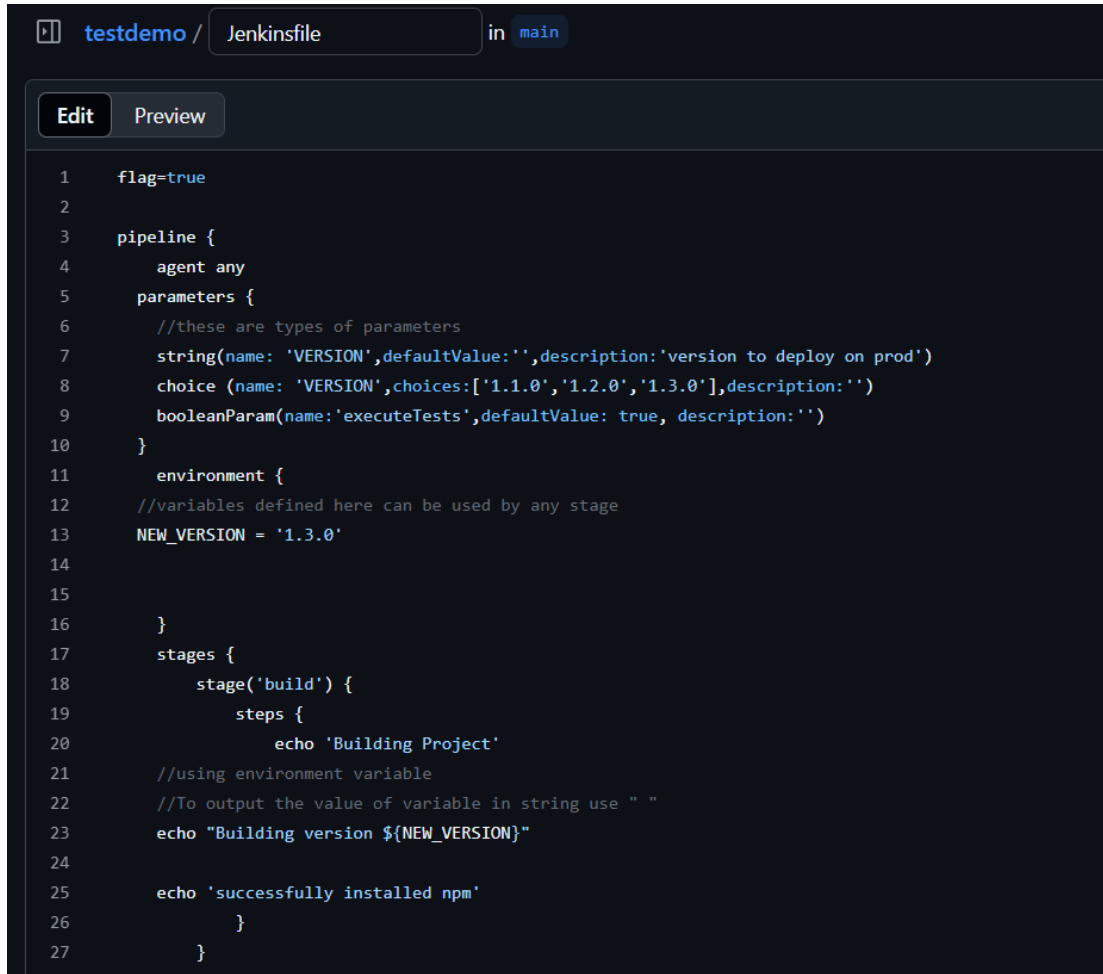
## What is Continuous Integration?

Continuous Integration is a development practice that requires developers to integrate code into a shared repository at regular intervals. This concept was meant to remove the problem of finding later 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.

## What is Jenkins File?

A Jenkins file is a script written in the Groovy programming language that defines the steps to be executed by a Jenkins pipeline. The pipeline is a series of steps executed in a particular order.

Make a new file named Jenkinsfile in any of your existing GitHub repository and paste the code given above:



```
1  flag=true
2
3  pipeline {
4      agent any
5      parameters {
6          //these are types of parameters
7          string(name: 'VERSION',defaultValue:'',description:'version to deploy on prod')
8          choice (name: 'VERSION',choices:['1.1.0','1.2.0','1.3.0'],description:'')
9          booleanParam(name:'executeTests',defaultValue: true, description:'')
10     }
11     environment {
12         //variables defined here can be used by any stage
13         NEW_VERSION = '1.3.0'
14
15     }
16     stages {
17         stage('build') {
18             steps {
19                 echo 'Building Project'
20
21                 //using environment variable
22                 //To output the value of variable in string use " "
23                 echo "Building version ${NEW_VERSION}"
24
25                 echo 'successfully installed npm'
26             }
27         }
28     }
29 }
```

flag=true

```
pipeline {
    agent any
    parameters {
        //these are types of parameters
        string(name: 'VERSION',defaultValue:"",description:'version to deploy
on prod')
        choice (name: 'VERSION',choices:['1.1.0','1.2.0','1.3.0'],description:")
        booleanParam(name:'executeTests',defaultValue: true, description:")
    }
}
```

```
environment {
    //variables defined here can be used by any stage
    NEW_VERSION = '1.3.0'
}
stages {
    stage('build') {
        steps {
            echo 'Building Project'
            //using environment variable
            //To output the value of variable in string use " "
            echo "Building version ${NEW_VERSION}"

            echo 'successfully installed npm'
        }
    }

    stage('test') {
        when {
            expression {
                params.executeTests
            }
        }
        steps {

            echo 'Testing Project'
        }
    }
    stage('deploy') {
        steps {
            echo 'Deploying Project'
            echo "DEploying version ${params.VERSION}"
        }
    }
}


post {
    // the conditions here will execute after the build is done
```






```
always {  
    //this action will happen always regardless of the result of build  
    echo 'Post build condition running'  
}  
failure {  
    //this action will happen only if the build has failed  
    echo 'Post Action if Build Failed'  
}  
}  
}
```

Now start Jenkins with this command:


- Java -jar Jenkins.war
- Go to Jenkins dashboard.
- Click on new item.
- Add name of your pipeline as myfirstpipeline.
- Select multibranch pipeline option:

 **Jenkins** / All / New Item




myFirstPipeline1


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**

OK

- A new pipeline will be created:

The screenshot shows the Jenkins web interface. At the top, the breadcrumb navigation reads 'Jenkins / myFirstPipeline1 / Configuration'. The 'Configuration' title is prominently displayed. Below it, a list of configuration categories is shown with gear icons: General (highlighted), Branch Sources, Build Configuration, Scan Multibranch Pipeline Triggers, Orphaned Item Strategy, Appearance, Health metrics, and Properties. The 'General' section is expanded, showing a toggle switch for 'Enabled' which is currently turned on. Below the toggle is a 'Display Name' label with a help icon and an empty text input field. At the bottom of the configuration area are two buttons: 'Save' and 'Apply'.

Now click on add source option in branch sources select github

Generate a github token

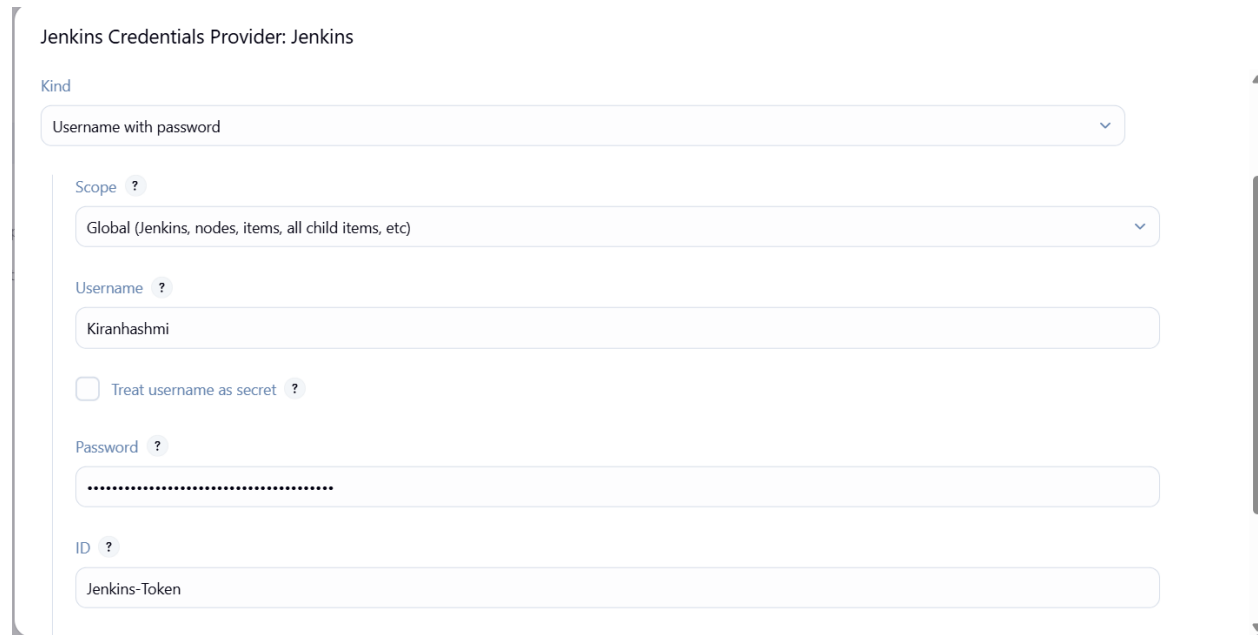
1. Generate a GitHub Personal Access Token (PAT)
  1. Go to: <https://github.com/settings/tokens>
  2. Click "Generate new token" → Select Classic token
  3. Name it like: JenkinsIntegrationToken
  4. Select scopes:
    - repo (to allow Jenkins to read your private/public repos)
    - admin:repo\_hook (to allow webhooks)
  5. Generate the token
  6. Copy the token

Add Credentials in Jenkins

1. In Jenkins → in the screen you're on → click + Add
2. Choose "Jenkins" scope (global is fine)

3. In the credentials form:

- Kind: Username with password
- Username: Your GitHub username (e.g., kiranshashmi)
- Password: Paste the GitHub token here
- ID (optional): e.g., github-token
- Description: e.g., GitHub PAT for multibranch



The screenshot shows the Jenkins Credentials Provider configuration page for a provider named 'Jenkins'. The 'Kind' is set to 'Username with password'. The 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'Kiranhashmi'. There is an unchecked checkbox for 'Treat username as secret'. The 'Password' field is masked with dots. The 'ID' field contains 'Jenkins-Token'.

Jenkins Credentials Provider: Jenkins

Kind

Username with password

Scope ?

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

Username ?

Kiranhashmi

☐ Treat username as secret ?

Password ?

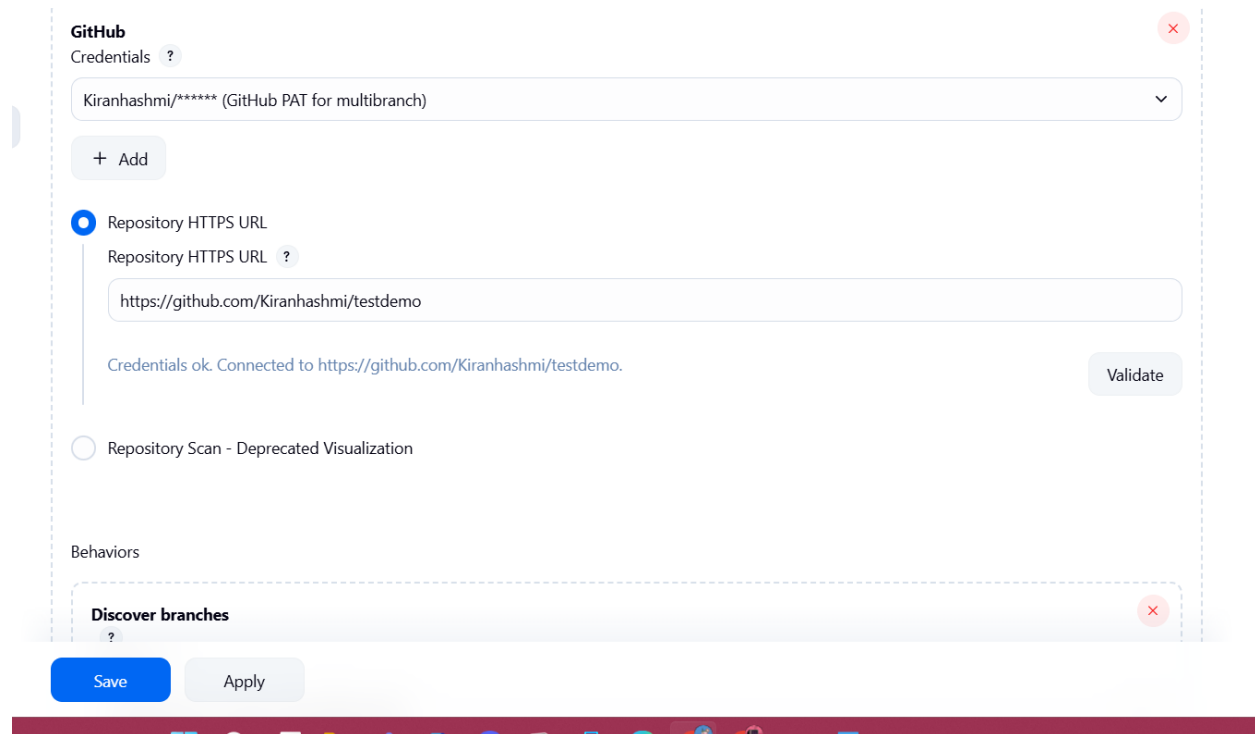
.....

ID ?

Jenkins-Token

- Paste the link of your github repository where you created the JenkinsFile

Submitted to :Mr. Mahaz Khan



The screenshot shows the Jenkins configuration page for a GitHub integration. At the top, it says "Submitted to :Mr. Mahaz Khan". The main configuration area is titled "GitHub" and contains several sections:

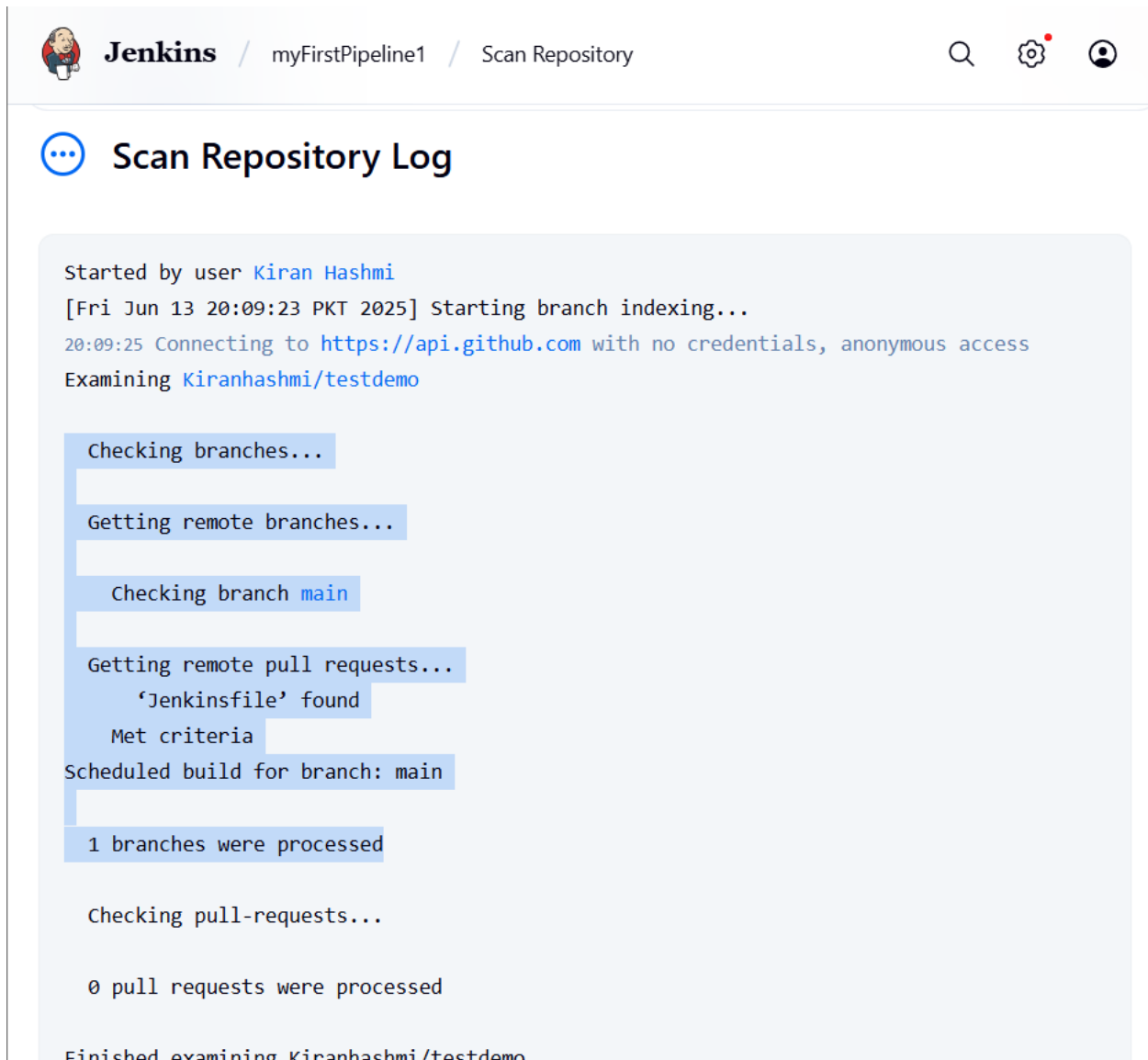
- Credentials**: A dropdown menu showing "Kiranhashmi/\*\*\*\*\* (GitHub PAT for multibranch)". Below it is a "+ Add" button.
- Repository HTTPS URL**: A radio button is selected. Below it is a text field containing "https://github.com/Kiranhashmi/testdemo". A message below the field says "Credentials ok. Connected to https://github.com/Kiranhashmi/testdemo." and a "Validate" button is to the right.
- Repository Scan - Depreciated Visualization**: An unselected radio button.
- Behaviors**: A section with a "Discover branches" option.

At the bottom of the configuration area are "Save" and "Apply" buttons. The entire configuration area is enclosed in a dashed border with a red 'X' icon in the top right corner.

Validate the link to see if it is correct.  
And click on save.

It will start scanning the repository.  
It will scan all the branches one by one to look for Jenkinsfile  
When it is found, the scanning stops and the exits successfully.

Date of Submission: 17/06/2025



The screenshot shows the Jenkins interface for a pipeline named 'myFirstPipeline1' with the step 'Scan Repository'. The log for 'Scan Repository Log' is displayed, showing the following steps:

- Started by user [Kiran Hashmi](#)
- [Fri Jun 13 20:09:23 PKT 2025] Starting branch indexing...
- 20:09:25 Connecting to <https://api.github.com> with no credentials, anonymous access
- Examining [Kiranhashmi/testdemo](#)
- Checking branches...
- Getting remote branches...
- Checking branch [main](#)
- Getting remote pull requests...
- 'Jenkinsfile' found
- Met criteria
- Scheduled build for branch: [main](#)
- 1 branches were processed
- Checking pull-requests...
- 0 pull requests were processed
- Finished examining [Kiranhashmi/testdemo](#)

## Why Did We Install Git for Jenkins?

Because Jenkins needs Git to clone your GitHub repository (i.e., download your code and Jenkinsfile) before it can run the pipeline.

### 1. Install Git (if not already installed)

- Go to: <https://git-scm.com/download/win>
- Download and install Git for Windows.
- After installing, verify it's available:
  - Open Command Prompt
  - Type: `git --version`
  - You should see something like: `git version 2.xx.x`

### 2. Configure Git in Jenkins

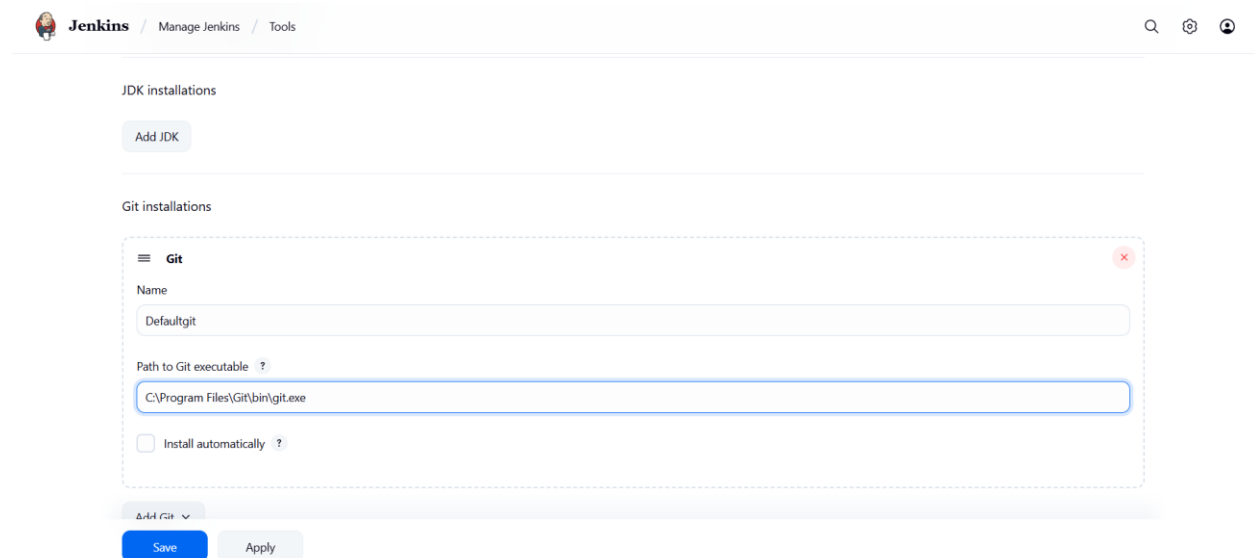
Step A: Open Jenkins settings

Submitted to :Mr. Mahaz Khan

- Go to Jenkins Dashboard
- Click “Manage Jenkins”
- Click “Global Tool Configuration”

Step B: Add Git installation

1. Scroll to Git section
2. Click “Add Git”
3. Give it a name (e.g., DefaultGit)
4. Uncheck "Install automatically"
5. In the Path to Git executable, enter:

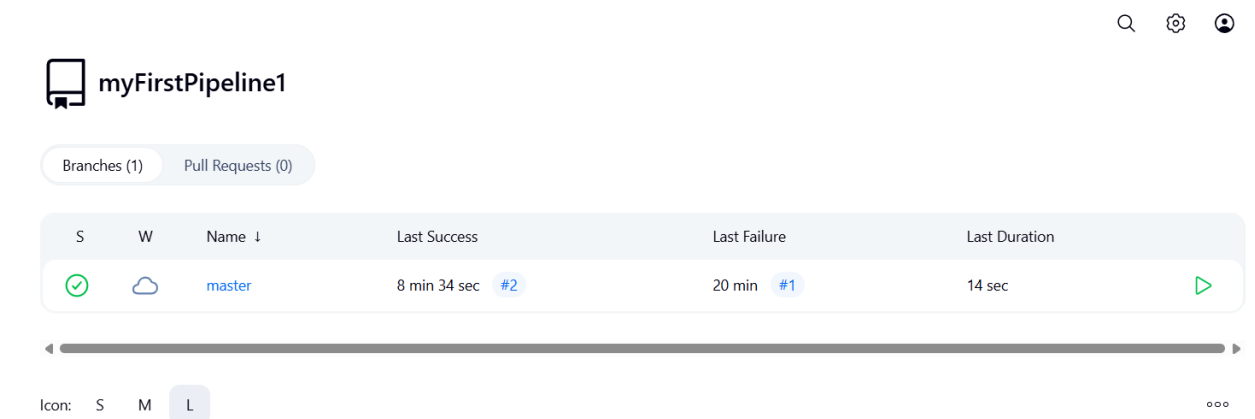


The screenshot shows the Jenkins 'Manage Jenkins' page, specifically the 'Tools' section. Under 'Git installations', there is a form to add a new Git installation. The form is titled 'Git' and has a red close button in the top right corner. It contains the following fields and options:

- Name:** A text input field with the value 'Defaultgit'.
- Path to Git executable:** A text input field with the value 'C:\Program Files\Git\bin\git.exe'.
- Install automatically:** A checkbox that is currently unchecked.

Below the form, there is a 'Save' button and an 'Apply' button. The 'Add Git' button is also visible above the form.

Now go back to the pipeline page:



The screenshot shows the Jenkins Pipeline page for a pipeline named 'myFirstPipeline1'. The page has a header with a search icon, a settings icon, and a user icon. Below the header, there is a section for 'myFirstPipeline1' with a 'Branches (1)' tab selected. The table below shows the pipeline's history:


S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	master	8 min 34 sec #2	20 min #1	14 sec


Below the table, there is a horizontal timeline bar. At the bottom, there is a section for 'Icon: S M L' with a 'L' button selected.


Date of Submission: 17/06/2025


- Click on master and you will be able to all the stages successfully build.


---


 **Jenkins** / myFirstPipeline1 / master


 Status


 Changes


 Build with Parameters

 View Configuration

 Stages

 GitHub

 Pipeline Syntax

 **master**


Full project name: myFirstPipeline1/master


**Permalinks**

- [Last build \(#3\), 17 min ago](#)
- [Last stable build \(#3\), 17 min ago](#)
- [Last successful build \(#3\), 17 min ago](#)
- [Last failed build \(#1\), 38 min ago](#)
- [Last unsuccessful build \(#1\), 38 min ago](#)
- [Last completed build \(#3\), 17 min ago](#)

Builds


Today

 #3 10:11 PM

 **Jenkins** / myFirstPipeline1 / master / Stages

**Stages**

June 13, 2025

 #3  
22:11 - 9.1 sec

Start

Checkout SCM

build

test

deploy

Post Actions

End

3 sec

0.23 sec

0.14 sec

0.15 sec

0.14 sec

- From build history, click on Console Output:

Here you will see how the whole pipeline has been executed from building till Deploying



Submitted to :Mr. Mahaz Khan

Jenkins / myFirstPipeline1 / master / #3

Console Output

Started by user Kiran Hashmi

22:11:40 Connecting to <https://api.github.com> using Kiranhashmi/\*\*\*\*\* (GitHub PAT for multibranch)

Obtained Jenkinsfile from 1a4c41d9fdb6a7ca9a918038d69ed3f424c103a6

[Pipeline] Start of Pipeline

Did you forget the 'def' keyword? WorkflowScript seems to be setting a field named flag (to a value of type Boolean) which could lead to memory leaks or other issues.

[Pipeline] node

Running on Jenkins in C:\Users\HP\.jenkins\workspace\myFirstPipeline1\_master

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

The recommended git tool is: NONE  
using credential Jenkins-Token

> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\Users\HP\.jenkins\workspace\myFirstPipeline1\_master\.git # timeout=10

Fetching changes from the remote Git repository

> C:\Program Files\Git\bin\git.exe config remote.origin.url <https://github.com/Kiranhashmi/testdemo.git> # timeout=10

Fetching without tags

Fetching upstream changes from <https://github.com/Kiranhashmi/testdemo.git>

> C:\Program Files\Git\bin\git.exe --version # timeout=10

> git --version # 'git version 2.49.0.windows.1'

using GIT\_ASKPASS to set credentials GitHub PAT for multibranch

```
Building Project
[Pipeline] echo
Building version 1.3.0
[Pipeline] echo
successfully installed npm
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (test)
[Pipeline] echo
Testing Project
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deploy)
[Pipeline] echo
Deploying Project
```

### Post Build Actions:

- ☐ You can also perform some actions after the build is complete. You just need to add another attribute named post in Jenkins file.

So, edit your Jenkins file from GitHub and write the post attribute like this, and commit changes.

```
48     post {
49         // the conditions here will execute after the build is done
50
51         always {
52             //this action will happen always regardless of the result of build
53             echo 'Post build condition running'
54         }
55         failure {
56             //this action will happen only if the build has failed
57             echo 'Post Action if Build Failed'
58         }
59     }
60 }
61 }
```

- Build the pipeline again and see the console output:

```
Deploying Project
[Pipeline] echo
Deploying version 1.2.0
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Post build condition running
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

GitHub has been notified of this commit's build result

Finished: SUCCESS
```

### Environment Variables:

- You can also use or change the environment variable available in Jenkins.
- What variables are available in Jenkins?

Type this: <http://localhost:8080/env-vars.html/>. You will see a list of all the environment variables.



Jenkins /



The following variables are available to shell and batch build steps:

**BRANCH\_NAME**

For a multibranch project, this will be set to the name of the branch being built, for example in case you wish to deploy to production from master but not from feature branches; if corresponding to some kind of change request, the name is generally arbitrary (refer to `CHANGE_ID` and `CHANGE_TARGET`).

**BRANCH\_IS\_PRIMARY**

For a multibranch project, if the SCM source reports that the branch being built is a primary branch, this will be set to "true"; else unset. Some SCM sources may report more than one branch as a primary branch while others may not supply this information.

**CHANGE\_ID**

For a multibranch project corresponding to some kind of change request, this will be set to the change ID, such as a pull request number, if supported; else unset.

**CHANGE\_URL**

For a multibranch project corresponding to some kind of change request, this will be set to the change URL, if supported; else unset.

**CHANGE\_TITLE**

For a multibranch project corresponding to some kind of change request, this will be set to the title of the change, if supported; else unset.

**CHANGE\_AUTHOR**

For a multibranch project corresponding to some kind of change request, this will be set to the username of the author of the proposed change, if supported; else unset.

**CHANGE\_AUTHOR\_DISPLAY\_NAME**

For a multibranch project corresponding to some kind of change request, this will be set to the human name of the author, if supported; else unset.

**CHANGE\_AUTHOR\_EMAIL**

For a multibranch project corresponding to some kind of change request, this will be set to the email address of the author, if supported; else unset.

**CHANGE\_TARGET**

You can also define your own environment variables in the Jenkins file.

The attribute for this purpose is named as `environment`. It is defined before the stages so that it is accessible by all stages.

Suppose we need a specific version of something in all stages:

```
12 //variables defined here can be used by any stage
13 NEW_VERSION = '1.3.0'
14
15
16 }
17 stages {
18     stage('build') {
19         steps {
20             echo 'Building Project'
21             //using environment variable
22             //To output the value of variable in string use " "
23             echo "Building version ${NEW_VERSION}"
24
25             echo 'successfully installed npm'
26         }
27     }
28 }
```

Here first we defined the variable in the environment attribute and the used it in the build stage.

- Now build the pipeline again and see the console output.




**Jenkins** / myFirstPipeline1 / master / #3

```
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] echo
Building Project
[Pipeline] echo
Building version 1.3.0
[Pipeline] echo
successfully installed npm
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (test)
[Pipeline] echo
Testing Project
```

### **Tools attribute for build tools:**

Using tools attribute, you can access different tools for your projects.

- Jenkins file support 3 tools right now that are:
- Maven, gradle, and JDK
- We will include Maven (A build automation tool used primarily for Java projects, mainly in
- downloading dependencies etc.)
- Edit your Jenkinsfile to add the following lines.
- Here first we listed Maven in the tools list.
- Then we used a command to install maven during build.

 **Jenkins**

Manage Jenkins

Tools

≡

Maven

Name

Maven

☒ Install automatically

?

≡

Install from Apache

Version

3.9.10

Add Installer

▼

Add Maven

Save

Apply