# **dLab 1: Installation of Jenkins**

1. Jenkins installation needs Java 11 or Java 17. Download Java 17 and install it.

<https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe>

1. Download Jenkins and install it.

<https://get.jenkins.io/windows-stable/2.414.1/jenkins.msi>

1. Run executable file which is downloaded.



1. Click Next and specify path where Jenkins to be installed

A screenshot of a computer

Description automatically generated

1. Select ‘Run service as LocalSystem’ option though it is less secured. But as we don’t have admin rights we need to go for this option.

A screenshot of a computer

Description automatically generated

1. Mention port number on which Jenkins to run. Click on Test Port

A screenshot of a computer

Description automatically generated

1. Select path where Java is installed.

A screenshot of a computer program

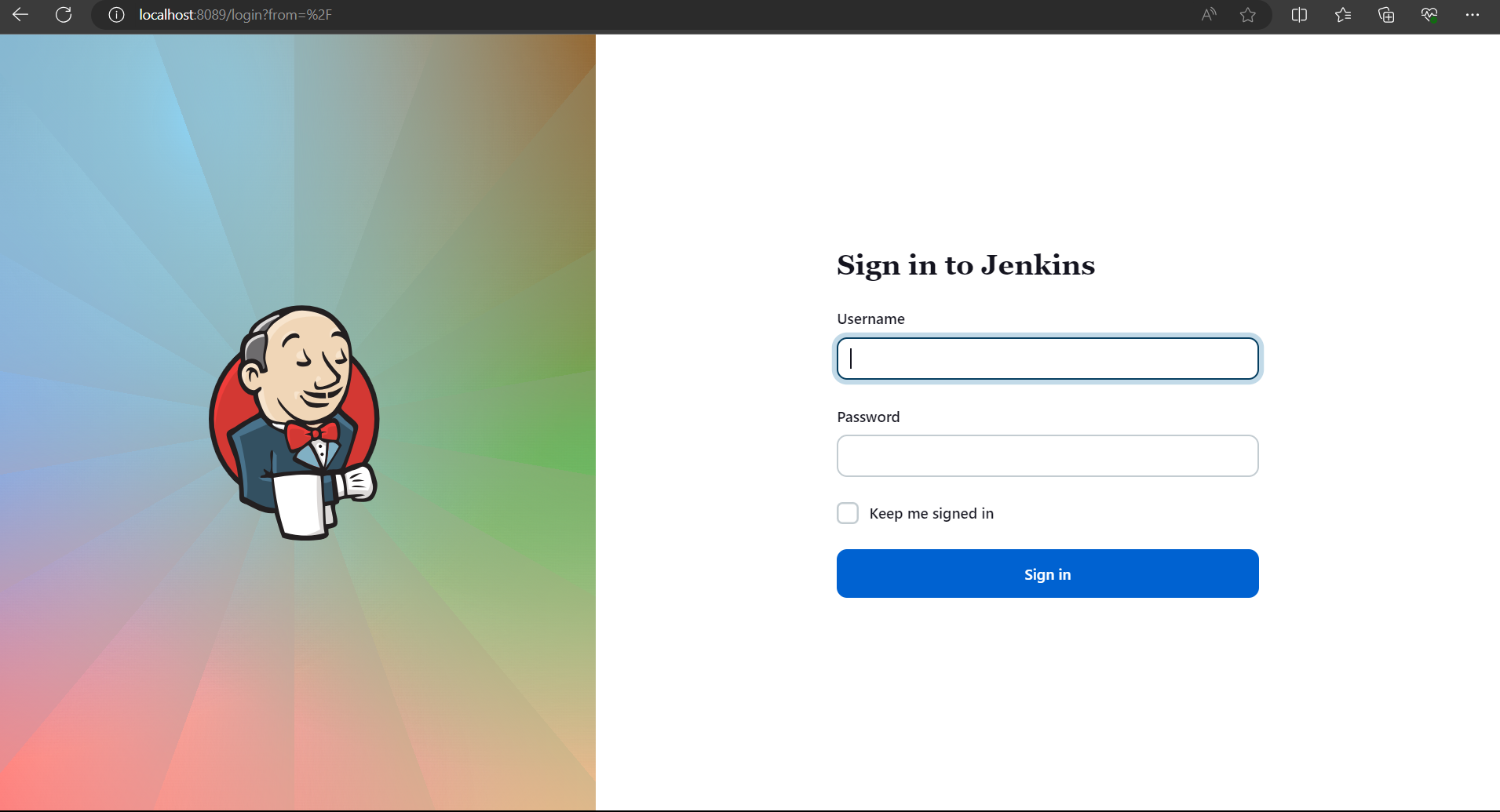
Description automatically generated

1. Next 🡪 Next 🡪 and install. Successful installation

A screenshot of a computer

Description automatically generated

1. Open the browser and type the URL **localhost:8089.** Home page of Jenkins.



1. Specify username as Admin and get password from the location where Jenkins installation is done.

A screenshot of a computer

Description automatically generated

1. Get the password

A screenshot of a computer

Description automatically generated

1. Login into Jenkins

A screenshot of a computer

Description automatically generated

# **Lab 2 : Install git on local system. Add Java program to GitHub repository.**

1. Download git. And install it

<https://git-scm.com/download/win>

1. Login to your GitHub account, create new repository and add HelloWorld.java program to it.

Refer this

<https://github.com/priyankak02/helloWorld/blob/7b962f7012dec3b74e2f6c7b99d2d23116942577/HelloWorld.java>

# **Lab 3: Integrate Jenkins with GitHub**

1. Home screen for Jenkins

A screenshot of a computer

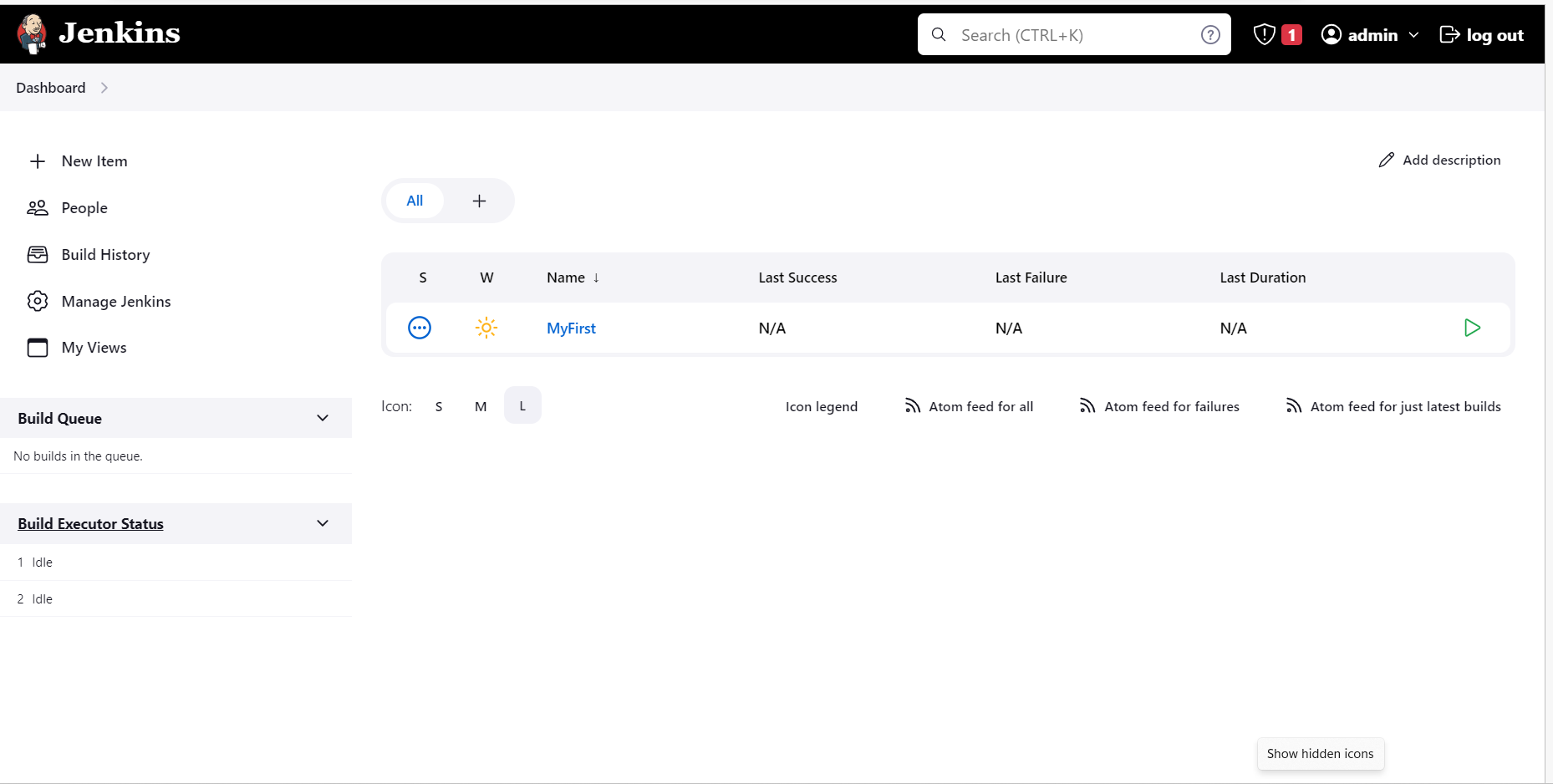
Description automatically generated

1. New Item 🡪 Specify item name 🡪 Select Freestyle project 🡪 Click OK

A screenshot of a computer

Description automatically generated

1. Click on Manage Jenkins for adding GitHub plugins



1. Select Plugins option.

A screenshot of a computer

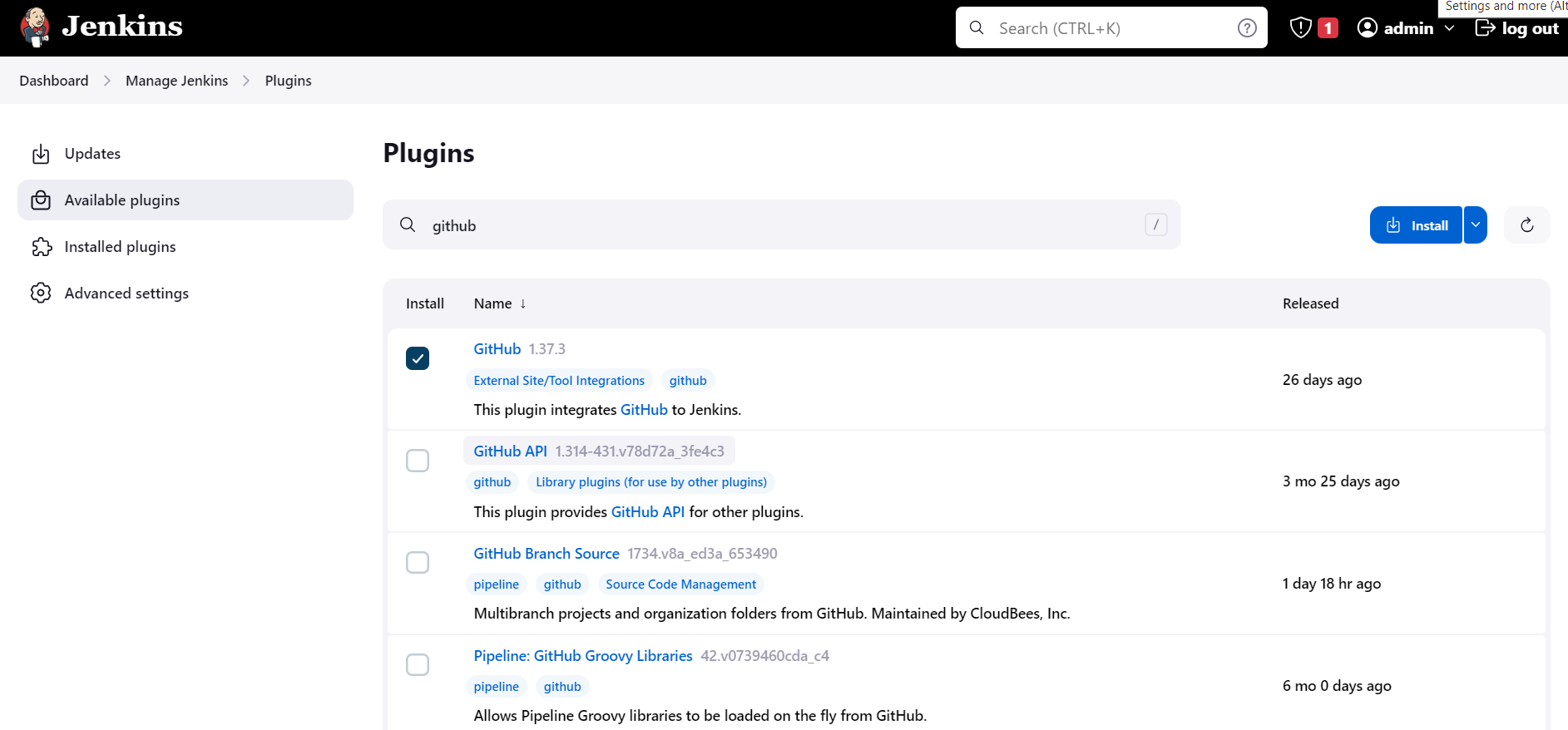
Description automatically generated

1. Select Available Plugins option

A screenshot of a computer

Description automatically generated

1. Search for Github, select it and install



Restart Jenkins when installation is complete

1. Specify path of git installed locally. Go to Manage Jenkins and select Tools option.

Specify path of git.exe

A screenshot of a computer

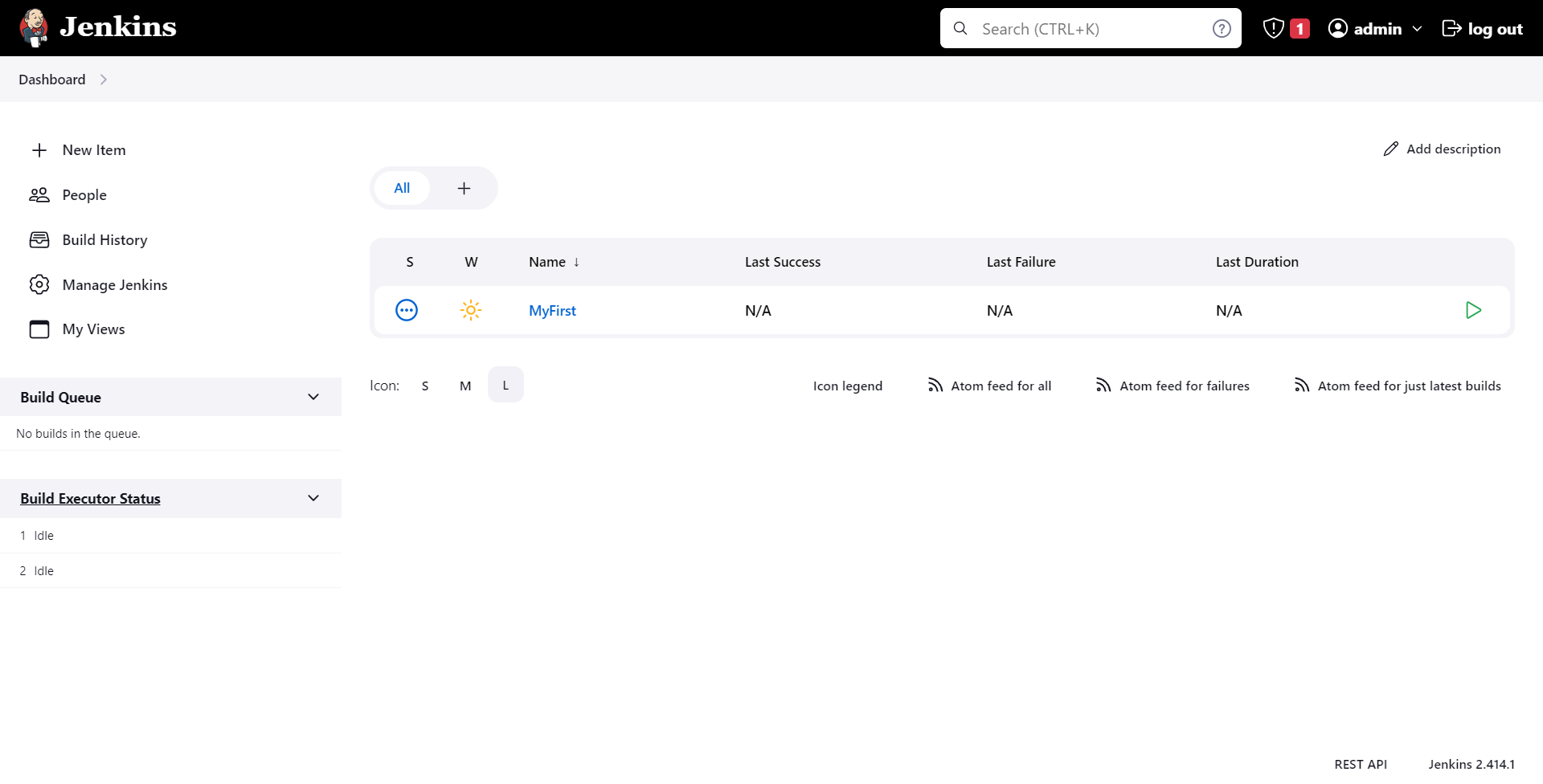
Description automatically generated

1. Under JDK installations, specify path for Jdk 17. Click Apply and then Save

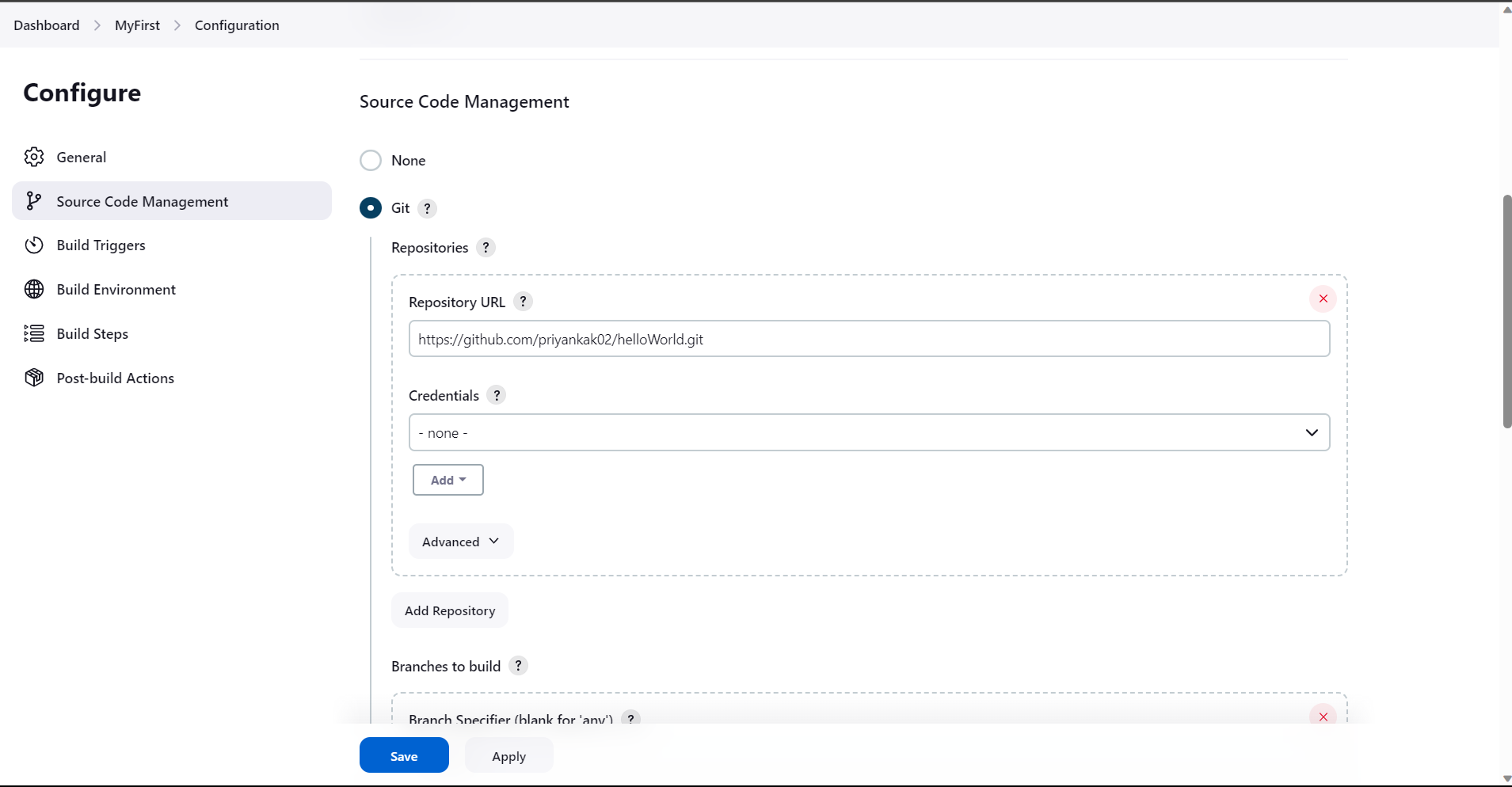
A screenshot of a computer

Description automatically generated

1. Lets configure our Job. Go back to Dashboard



1. Select Job previously created. Click Configure. Select Git under source code management. Specify Git Repository URL.



1. Specify blank to Branch specifier.

A screenshot of a computer

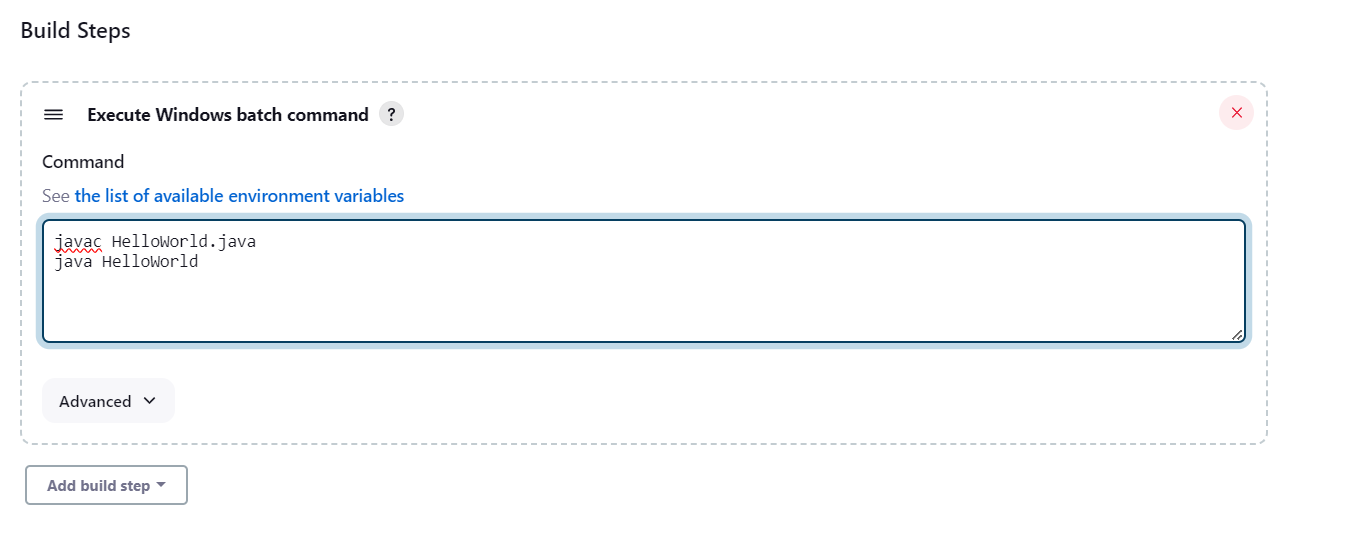
Description automatically generated

1. Select Execute Windows Batch Command under Add Build Step option.

A screenshot of a computer

Description automatically generated

1. Mention commands to build (compile + execute) our java program which is stored in git. Click Apply and then save



1. Click Build Now. Observer ongoing build process.

A screenshot of a computer

Description automatically generated

1. Click on build number

A screenshot of a computer

Description automatically generated

1. Select console output

A screenshot of a computer

Description automatically generated

# **Lab 4: Schedule build process**

Here need to write cron expression.

Refer [Trigger Jenkins Job using "Build periodically" and "Poll SCM" - DevOpsSchool.com](https://www.devopsschool.com/blog/setting-up-the-cron-jobs-in-jenkins-using-build-periodically-scheduling-the-jenins-job/) for writing cron expression.

1. Go to Dashboard, select job, click configure. Select Build periodically option under Build Triggers.

A screenshot of a computer

Description automatically generated

1. Specify cron expression as, click apply and save

A screenshot of a computer

Description automatically generated

Now build will automatically happen after every 2 minutes.

1. Observe build is happening after every 2 minutes.

A screenshot of a computer

Description automatically generated