# SEPM EXP: 7

*Name: Atharva Khewle*
*Batch : T13*
*RollNo : 48*

**Aim** : To Install and Configure Jenkins on Windows

## THEORY

Jenkins is a widely used open-source automation server that helps in automating various tasks in the software development lifecycle, such as building, testing, and deploying applications. It is primarily used for Continuous Integration (CI) and Continuous Deployment (CD) processes. Jenkins can be installed on various platforms, including Windows, making it accessible for a wide range of developers and DevOps teams.
In this theory, we will discuss the steps and prerequisites needed for installing Jenkins on a Windows machine. The installation process involves setting up Java, downloading the Jenkins WAR file, and configuring Jenkins as a service on Windows.

### Prerequisites for Installing Jenkins on Windows

Before you can install Jenkins on Windows, ensure the following prerequisites are met:

1. Java Development Kit (JDK) Installation: Jenkins is written in Java, so the Java Runtime Environment (JRE) or the Java Development Kit (JDK) is required for running Jenkins. It is recommended to use JDK 8 or a later version for compatibility with Jenkins.
2. Windows Operating System: Jenkins can be installed on any version of Windows, including Windows 7, 8, 10, or Windows Server editions. Ensure that your Windows machine meets the minimum system requirements for running Jenkins.
3. Network Connectivity: Jenkins may require internet access for downloading plugins and updates, so make sure your Windows system has internet access.

### Installation Steps for Jenkins on Windows

1. Install Java Development Kit (JDK):
    - Download the JDK installer from the official Oracle website (or an OpenJDK version).
    - Run the installer and follow the installation instructions.
    - Set the JAVA_HOME environment variable:
        - Open the System Properties (right-click My Computer → Properties → Advanced system settings).
        - Click on the Environment Variables button.
        - Add a new system variable:
            - Name: JAVA_HOME
            - Value: Path to the installed JDK directory (e.g., C:\Program Files\Java\jdk-11.0.9)
    - Add the bin folder of JDK to the PATH environment variable:

        - For example: C:\Program Files\Java\jdk-11.0.9\bin
2. Download Jenkins WAR File:
    - Go to the official Jenkins website: https://www.jenkins.io/download/
    - Download the jenkins.war file, which is a Java Web Application Archive file that contains Jenkins. You can download the latest stable version or a specific version, depending on your requirements.
3. Run Jenkins Using Command Line:
    - Open a command prompt (cmd) as an administrator.
    - Navigate to the directory where you downloaded the jenkins.war file.

Run the following command to start Jenkins:

nginx

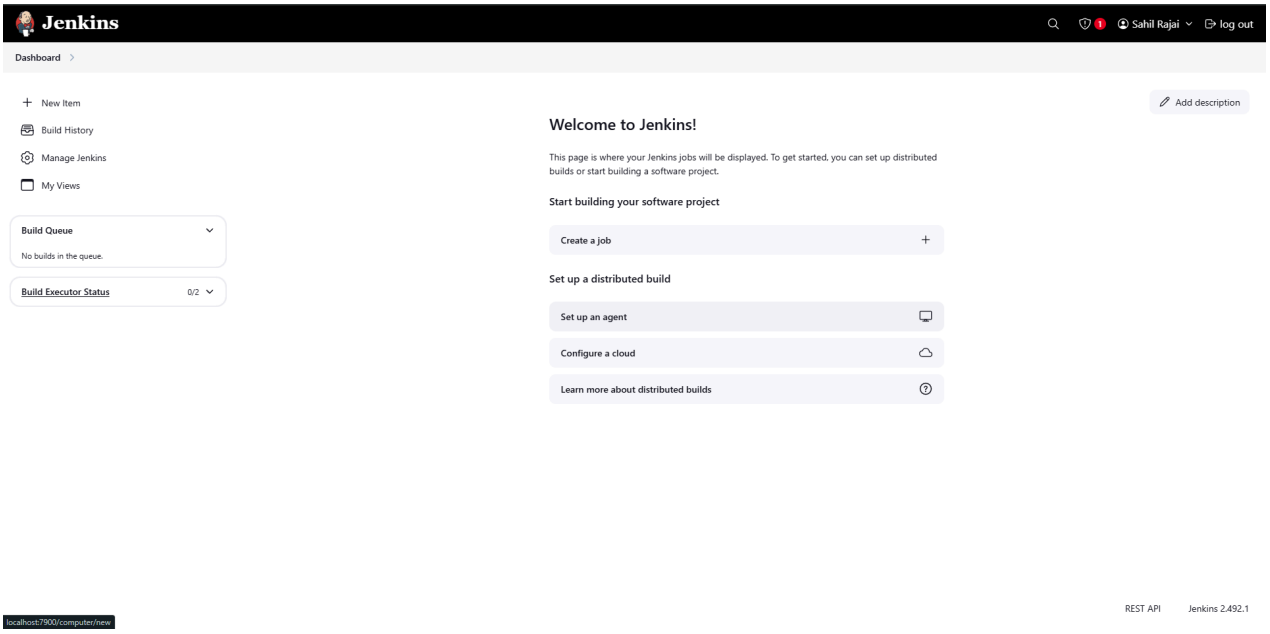
Copy

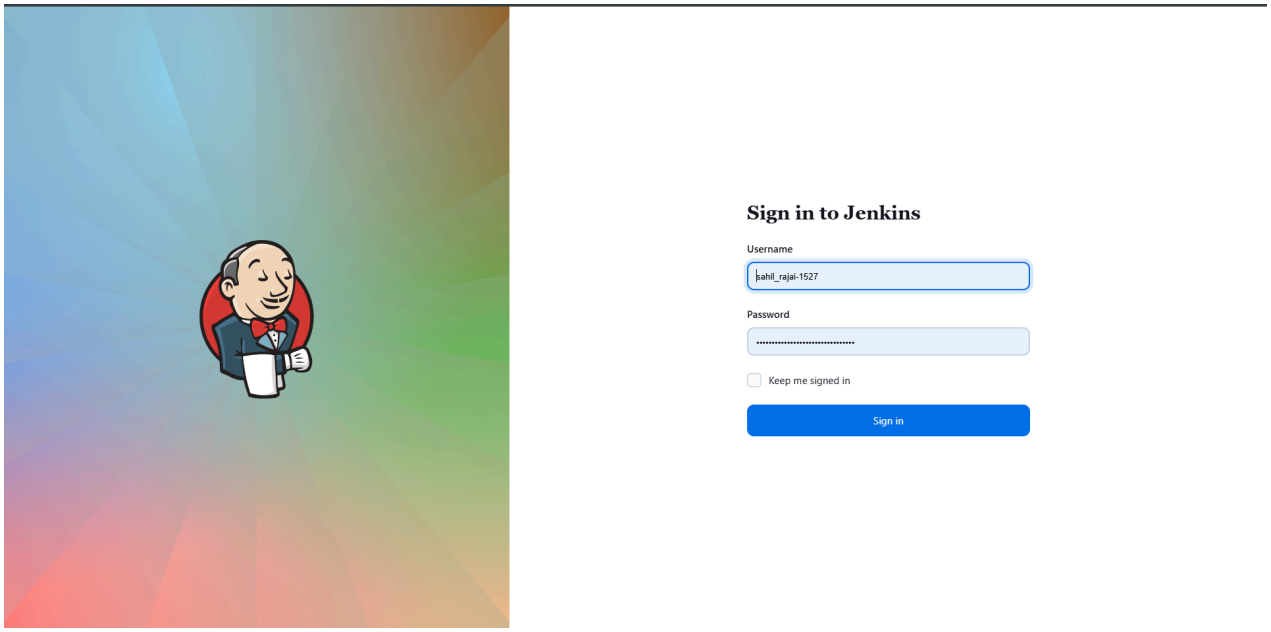
java -jar jenkins.war

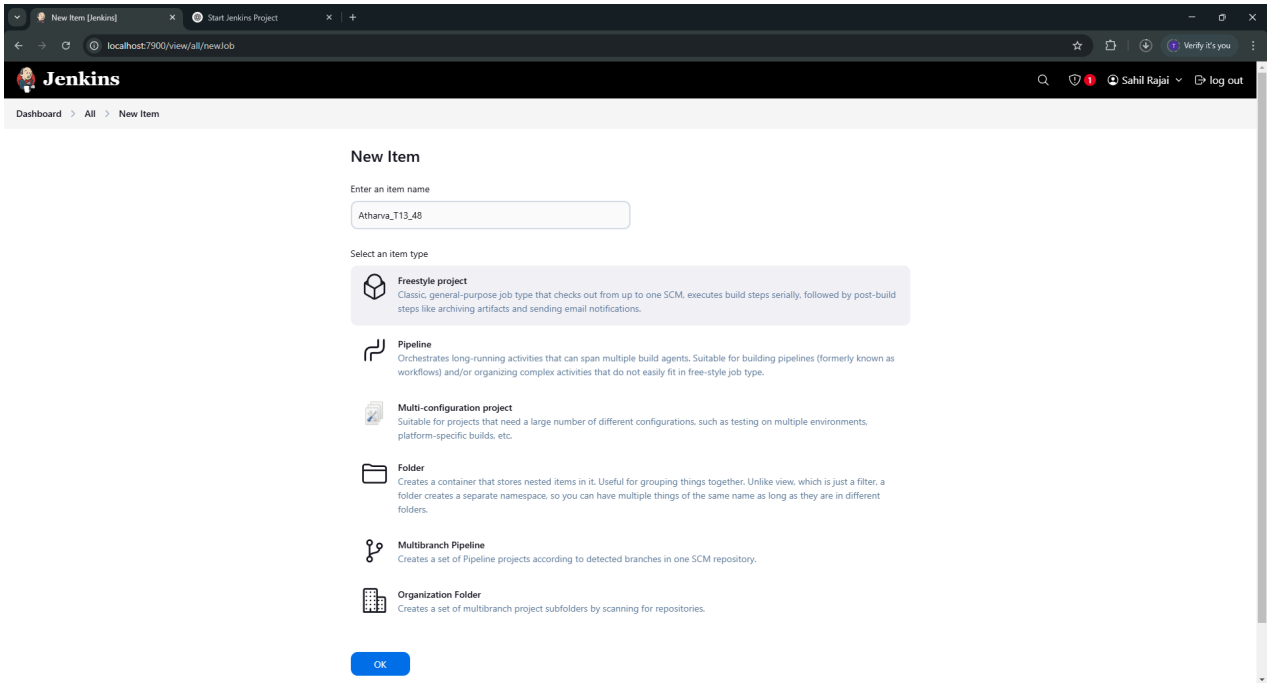
    -
    - Jenkins will start running on the default port 8080. To access Jenkins, open a web browser and navigate to http://localhost:8080.
4. Unlock Jenkins:
    - Upon accessing Jenkins for the first time, you will be prompted to unlock Jenkins by providing an initial setup password.
    - The password can be found in a file named secrets/initialAdminPassword located in the Jenkins home directory (C:\Users\<YourUser>\.jenkins by default).
    - Copy the password and paste it into the unlock field on the Jenkins setup page.
5. Install Suggested Plugins:
    - After unlocking Jenkins, you will be prompted to install plugins.
    - Select the "Install suggested plugins" option to install a set of recommended plugins that will enhance Jenkins' functionality.
    - This process may take a few minutes, depending on your internet speed.
6. Create Admin User:
    - After the plugins are installed, Jenkins will ask you to create an admin user. This will be the primary user to manage Jenkins.
    - Enter a username, password, full name, and email address for the admin user.
7. Jenkins Setup Complete:
    - Once the admin user is created, you will be taken to the Jenkins dashboard.
    - From here, you can start configuring Jenkins, such as setting up jobs, pipelines, and integrating other tools like Git, Maven, or Docker.

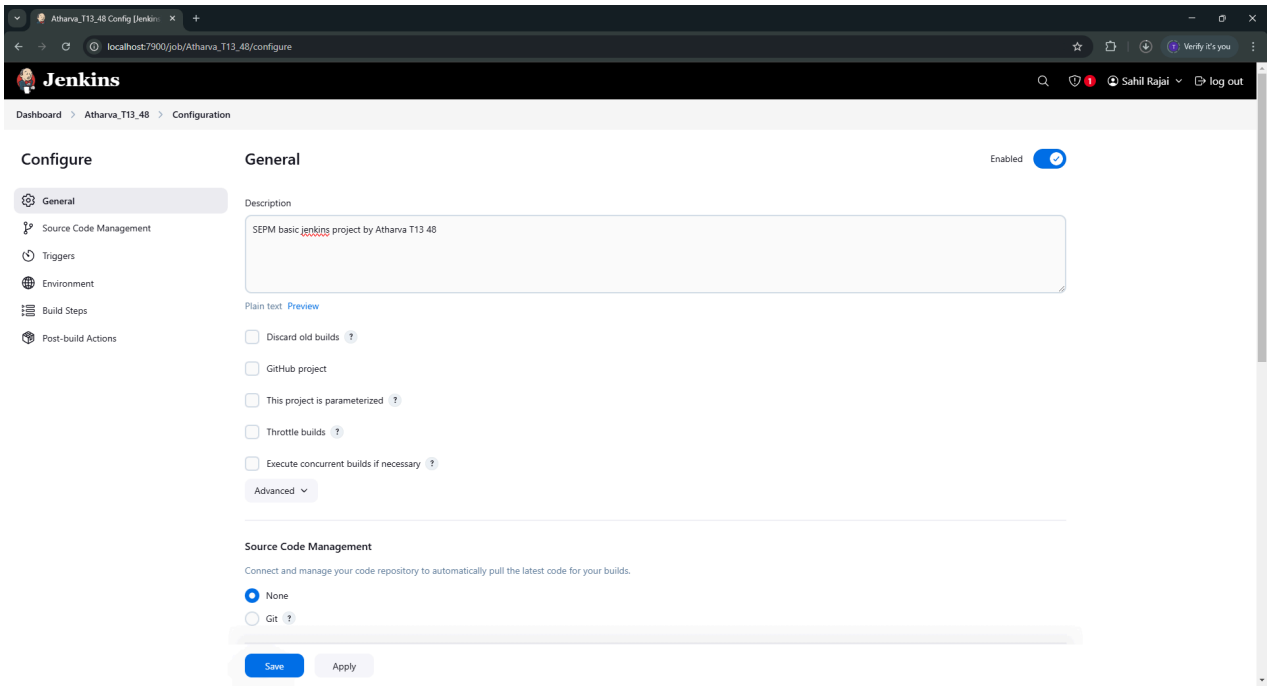
**Work:**

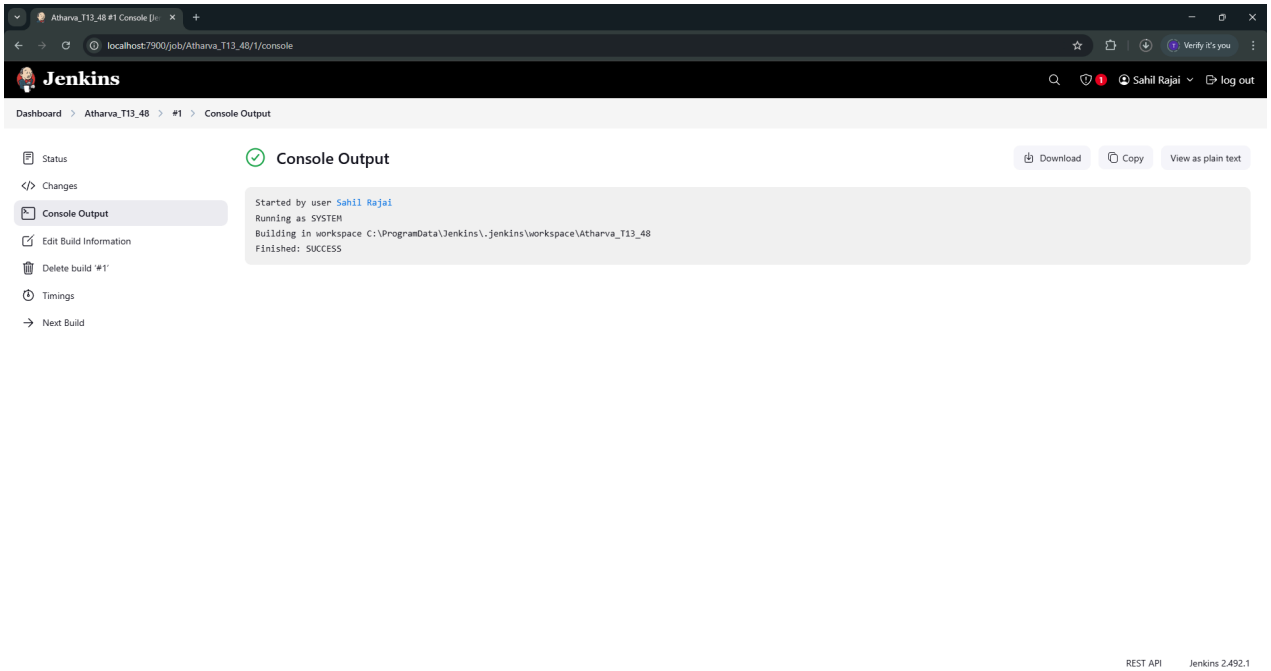

Jenkins 2.492.1 Setup
Welcome to the Jenkins 2.492.1 Setup Wizard
The Setup Wizard allows you to change the way Jenkins 2.492.1 features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel

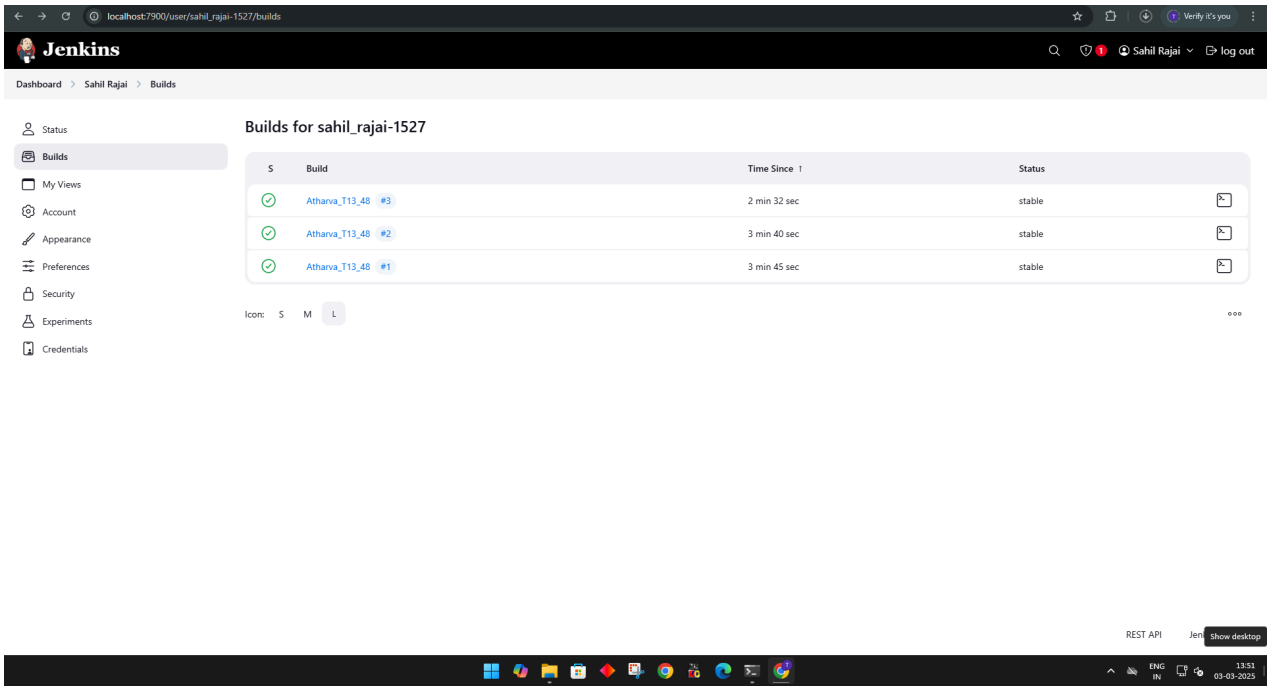

Jenkins
Sahil Rajai
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
0/2
Add description
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building 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
REST API
Jenkins 2.492.1
localhost:7900/computer/new


Sign in to Jenkins
Username
sahil_rajai-1527
Password
Keep me signed in
Sign in


New Item [Jenkins]
Start Jenkins Project
localhost:7900/view/all/newJob
Verify it's you
Jenkins
Sahil Rajai
log out
Dashboard
All
New Item
New Item
Enter an item name
Atharva_T13_48
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.
OK


Atharva_T13_48 Config [Jenkins
localhost:7900/job/Atharva_T13_48/configure
Jenkins
Sahil Rajai
log out
Dashboard
Atharva_T13_48
Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
General
Enabled
Description
SEPM basic jenkins project by Atharva T13 48
Plain text 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


Atharva_T13_48 #1 Console [Je
localhost:7900/job/Atharva_T13_48/1/console
Jenkins
Sahil Rajai
log out
Dashboard
Atharva_T13_48
#1
Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Timings
Next Build
Console Output
Download
Copy
View as plain text
Started by user Sahil Rajai
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Atharva_T13_48
Finished: SUCCESS
REST API
Jenkins 2.492.1




## Conclusion

Installing Jenkins on Windows is a straightforward process that sets up a powerful automation server for continuous integration and delivery (CI/CD). By installing Java, downloading the Jenkins WAR file, and configuring it to run as a service, you can quickly get Jenkins up and running. Once installed, Jenkins streamlines software development tasks, automating builds, tests, and deployments, while offering flexibility through plugins and integrations. With Jenkins running as a service, it ensures a reliable and efficient automation environment.