

# Jenkins Installation & Configuration

## System Requirements

Before installing Jenkins, the following software and system requirements were ensured:

- **Operating System:** Windows 10 (64-bit)
- **Java Development Kit (JDK):** Version 11 or later
- **Jenkins Version:** Jenkins Long-Term Support (LTS)
- **Git for Windows:** Installed for source code management
- **Web Browser:** Google Chrome / Microsoft Edge

Java was verified using the command:

```
cmd  
java -version
```

Git was verified using:

```
cmd  
git --version
```

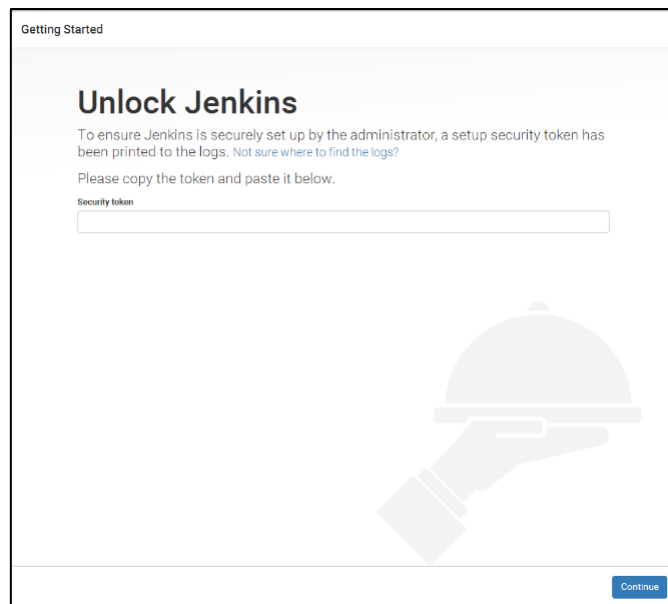
## Jenkins Installation on Windows

The Jenkins installation process was carried out using the official Jenkins LTS installer.

### Steps Followed:

1. The Jenkins LTS installer was downloaded from the official Jenkins website.
2. The installer was executed and Jenkins was installed as a Windows service.
3. The default installation path was selected.
4. Jenkins was configured to run on port **8080**.
5. After installation, Jenkins was accessed using the URL:

```
http://localhost:8080
```



**Figure 1.** Jenkins initial “Unlock Jenkins” page

## Initial Setup Wizard

After launching Jenkins for the first time, the Initial Setup Wizard appeared.

### Steps Performed:

1. The initial administrator password was retrieved from:

`C:\Program Files\Jenkins\secrets\initialAdminPassword`

2. The password was entered on the Unlock Jenkins screen.
3. The option “**Install suggested plugins**” was selected.
4. Jenkins automatically installed the recommended plugins such as Git, Pipeline, and Email Extension plugins.



**Figure 2.** “Install suggested plugins” screen

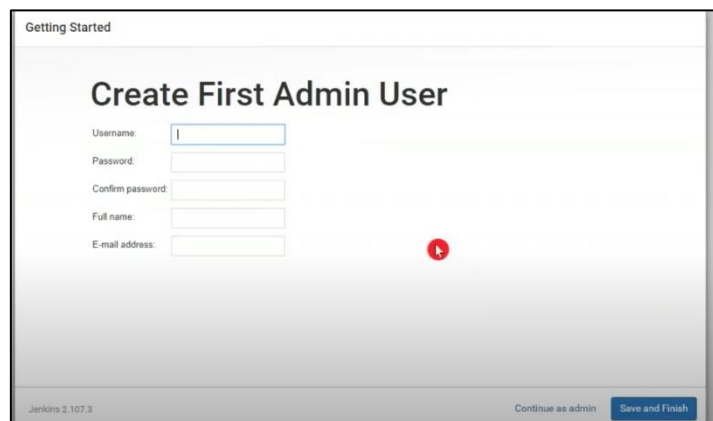
## Admin User Creation

After plugin installation, Jenkins prompted for user creation.

### Configuration:

- Username: Enrolment Number
- Password: (secure password set)
- Full name and email ID were entered

This account was created as the Jenkins administrator.



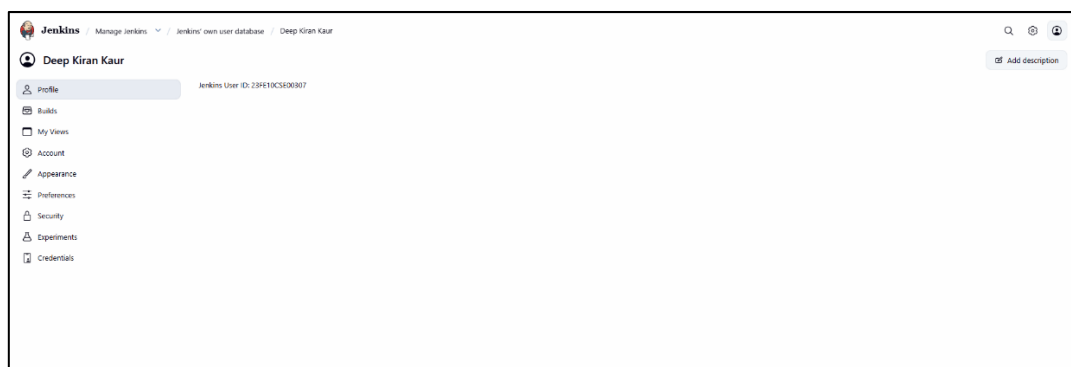
**Figure 3.** Create First Admin User page

## Jenkins Dashboard Verification

Once the setup was completed, the Jenkins dashboard was displayed successfully.

The dashboard provides access to:

- Job creation
- Plugin management
- System configuration
- Build history and logs



**Figure 4.** Jenkins Dashboard

## Global Tool Configuration

Jenkins tools were configured to support build and testing operations.

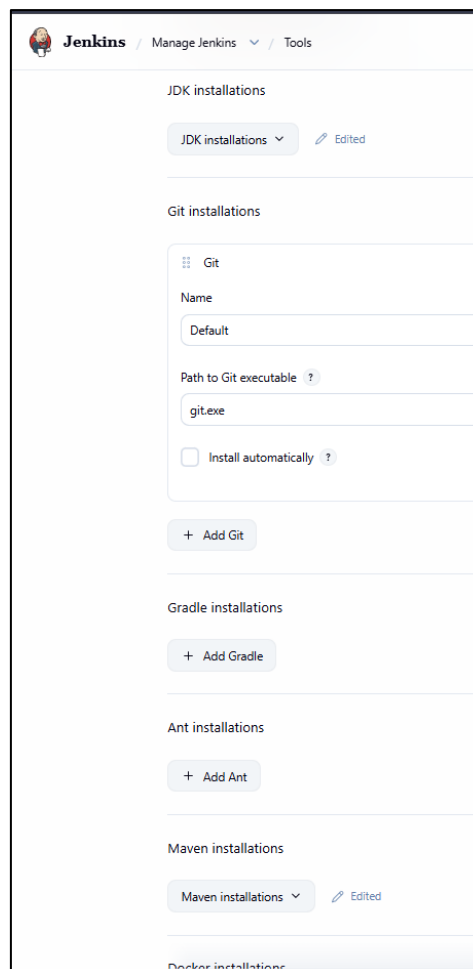
## Tools Configured:

- **JDK:** Path to installed Java JDK configured
- **Git:** Path to Git executable configured
- **Maven:** Apache Maven configured for build automation

These configurations allow Jenkins to compile and test Java projects automatically.

## Steps:

1. Navigated to **Manage Jenkins → Global Tool Configuration**
2. Added JDK, Git, and Maven installations
3. Saved the configuration



**Figure 5.** Global Tool Configuration page showing JDK, Git, Maven

## Configure System Settings

System-level configurations were completed.

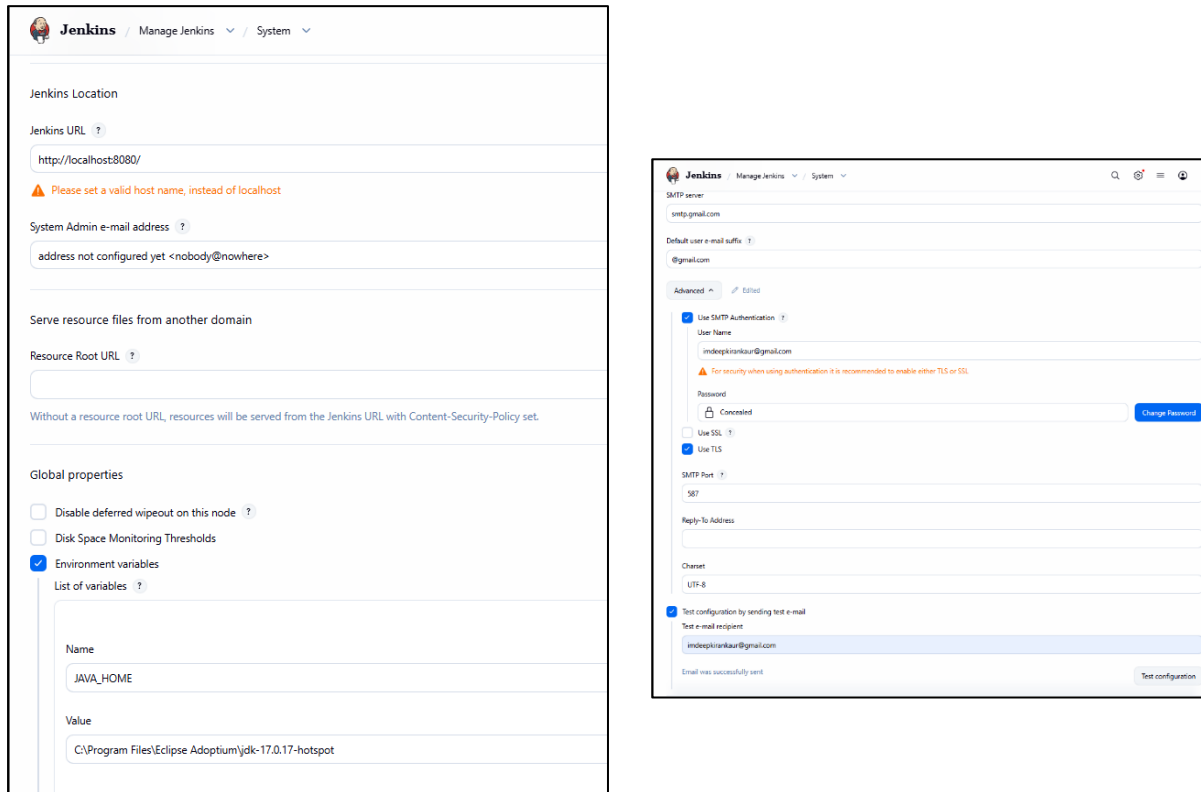
## Settings Configured:

- **Jenkins URL:** Set to <http://localhost:8080>

- **SMTP Server:** Configured for email notifications
- **System Environment Variables:** Configured as required

Steps:

1. Navigated to **Manage Jenkins → Configure System**
2. Set Jenkins URL
3. Entered SMTP credentials
4. Saved settings



**Figure 6.** Configure System page showing Jenkins URL and SMTP configuration

## Security Configuration

Security was configured to prevent unauthorized access.

### Security Measures Implemented:

- **Security Realm:** Jenkins' own user database
- **Authorization Strategy:** Matrix-based security
- **CSRF Protection:** Enabled

Steps:

1. Navigated to **Manage Jenkins → Configure Global Security**
2. Enabled user authentication
3. Assigned permissions using matrix-based strategy
4. Enabled CSRF protection
5. Saved configuration



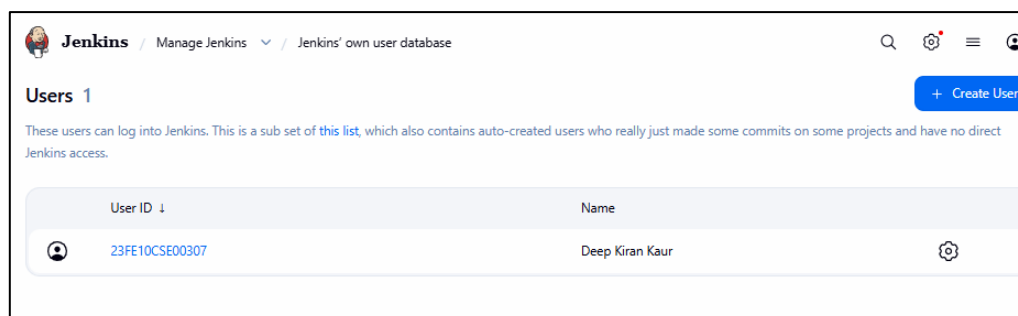
**Figure 7.** Configure Global Security page

## User Management

User accounts were managed using Jenkins user management.

### Actions Performed:

- Admin user created
- Permissions assigned
- User details verified



**Figure 8.** User Management Screen

## Summary

Jenkins was successfully installed and configured on the local Windows system.

All essential tools such as JDK, Git, and Maven were integrated.

Security settings were applied and administrative access was established.

This setup provided a stable environment for creating Jenkins jobs and pipelines for Continuous Integration.