

JENKINS INSTALLATION AND SETUP, EXPLORE THE ENVIRONMENT

Installing and setting up Jenkins on a Windows machine is straightforward and allows you to take advantage of its continuous integration and continuous deployment (CI/CD) capabilities. This guide walks through the installation process, initial configuration, and a brief exploration of the Jenkins environment to help you get started.

Step 1: Download Jenkins

To begin, download Jenkins from its official website.

1. **Visit the Jenkins Website:** Go to Jenkins' official download page.
2. **Select the Windows Installer:** Jenkins offers a Windows installer, a .msi file, which simplifies the installation process by setting up Jenkins as a Windows service. Click on the installer link to begin the download.

Step 2: Install Jenkins on Windows

Once the .msi file is downloaded, you're ready to install Jenkins on your Windows machine.

1. **Run the Installer:** Locate the .msi file in your downloads folder, and double-click it to start the installation.
2. **Follow Installation Prompts:**
 - **Select Installation Location:** You'll be prompted to choose an installation folder. You may choose the default path or specify a custom location.
 - **Choose Java Version:** Jenkins requires Java to run, so ensure Java 11 or Java 17 is installed on your system before proceeding. If you don't have Java installed, install it by visiting [Java's official site](#).

3. **Complete the Installation:** Once the installer completes, Jenkins will be set up as a Windows service, which allows it to start automatically whenever your computer boots.

Step 3: Access Jenkins and Initial Setup

After the installation, you need to set up Jenkins by accessing it through your web browser.

1. **Open Jenkins in a Browser:**

- By default, Jenkins runs on port 8080. To access it, open your browser and go to:

`http://localhost:8080`

2. **Unlock Jenkins:**

- Jenkins will initially be in a locked state. To unlock it, you'll need to locate the **initial administrator password**.
- Open the file at the following location in a text editor (e.g., Notepad):

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

- Copy the password from this file and paste it into the Jenkins setup page to unlock Jenkins.

3. **Install Suggested Plugins:**

- Jenkins will prompt you to install plugins. Choose **Install suggested plugins** to install a set of commonly used plugins, which can enhance your Jenkins environment with features like Git integration, build pipelines, and notifications.

4. **Create an Administrator Account:**

- After installing plugins, create your first admin user by providing a username, password, and email address.

- This account will be your main login for managing Jenkins.

5. Instance Configuration:

- In the final setup screen, you'll see the Jenkins URL. Keep the default value (<http://localhost:8080>) unless you plan to access Jenkins from other devices on your network. Click **Save and Finish** to complete the setup.

Step 4: Exploring the Jenkins Environment

With Jenkins now set up, let's explore its interface to understand its structure and features.

1. Dashboard:

- The Jenkins dashboard is the main control center where you can create and manage jobs, configure system settings, and monitor build statuses.
- From here, you can create new projects, view recent builds, and access the system's configuration options.

2. Creating Your First Job:

- To create a job, click on **New Item** on the left sidebar.
- Give your job a name, select a job type (such as a Freestyle project or a Pipeline), and click **OK**. This opens a configuration page for the job, where you can specify the source code repository, build triggers, and build steps.

3. Manage Jenkins:

- The **Manage Jenkins** menu offers settings and tools to configure Jenkins as an administrator.
- Here, you can install new plugins, manage credentials, update Jenkins, view system logs, and configure global security settings.

4. Plugins and Configuration:

- Plugins are essential for expanding Jenkins' functionality. Jenkins has an extensive plugin library, allowing you to integrate with various tools and services like Git, Docker, and cloud providers.
- To manage plugins, navigate to **Manage Jenkins > Manage Plugins**, where you can install or update plugins and view installed ones.

Step 5: Setting Up Jenkins for Source Code Management

With the environment set up, you can connect Jenkins to your source code repositories.

1. Integrating Git with Jenkins:

- Most projects use Git for source code management. To connect Jenkins to Git repositories, ensure the **Git plugin** is installed. You can check this by going to **Manage Plugins**.
- When configuring a job, enter the repository URL in the **Source Code Management** section and provide authentication credentials if required.

2. Configuring Build Triggers:

- Jenkins can be configured to build automatically whenever changes are pushed to your repository. In the job configuration, select **Poll SCM** or **Build when a change is pushed to GitHub** to enable automated builds.

Conclusion

Installing and setting up Jenkins on Windows is a foundational step towards implementing automated build and deployment pipelines. By exploring Jenkins' environment, including the dashboard, job creation, and plugin management, you can configure and scale Jenkins to meet your development and deployment needs. With Jenkins set up and your environment configured, you are now ready

to streamline project builds, tests, and deployments in a more efficient, automated way.

HIMA BINDU

23BTRCL228

AIML/C