DevOps Training-Day-1

Installing and Setting Up WSL with Ubuntu on Windows 10

Step 1: Enable WSL

Before installing Ubuntu, ensure that WSL is enabled on your Windows system.

Enable WSL Feature

1. Open **PowerShell** as Administrator and run:
2. wsl --install

This installs the default Linux distribution and enables necessary components.

3. If WSL is already installed but not enabled, use:
4. dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
5. Enable the Virtual Machine Platform feature (required for WSL 2):
6. dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
7. Restart your computer to apply changes.

Step 2: Install Ubuntu

1. Open **Command Prompt** or **PowerShell** and run:
2. wsl --install -d Ubuntu

If the installation fails due to timeout issues, retry the command after shutting down WSL:

wsl --shutdown

wsl --install -d Ubuntu

3. Once installed, start Ubuntu:
4. wsl.exe -d Ubuntu

Step 3: Set Up Ubuntu

When Ubuntu runs for the first time, it will ask you to create a new user account.

1. **Enter a username** (must start with a lowercase letter or underscore, and contain only lowercase letters, digits, underscores, and dashes).
2. **Set a password** (enter and confirm the password). If passwords do not match, you will need to retry.
3. Once successful, Ubuntu will be set up and ready to use.

Step 4: Verify Installation

To check the installed distributions and their versions:

wsl -l -v

To verify Ubuntu is running:

wsl -d Ubuntu

Step 5: Configure Ubuntu

Update System Packages

After logging in, update the package list and upgrade installed packages:

sudo apt update && sudo apt upgrade -y

Set Default WSL Version

To use WSL 2 as the default version for future installations:

wsl --set-default-version 2

To check the current WSL version:

wsl -l -v

To convert an existing installation to WSL 2:

wsl --set-version Ubuntu 2

Step 6: Enable .hushlogin to Suppress Login Message

To disable the daily login message, create a .hushlogin file in your home directory:

touch ~/.hushlogin

Additional Commands

Restart WSL:

wsl --shutdown

Uninstall a Distribution:

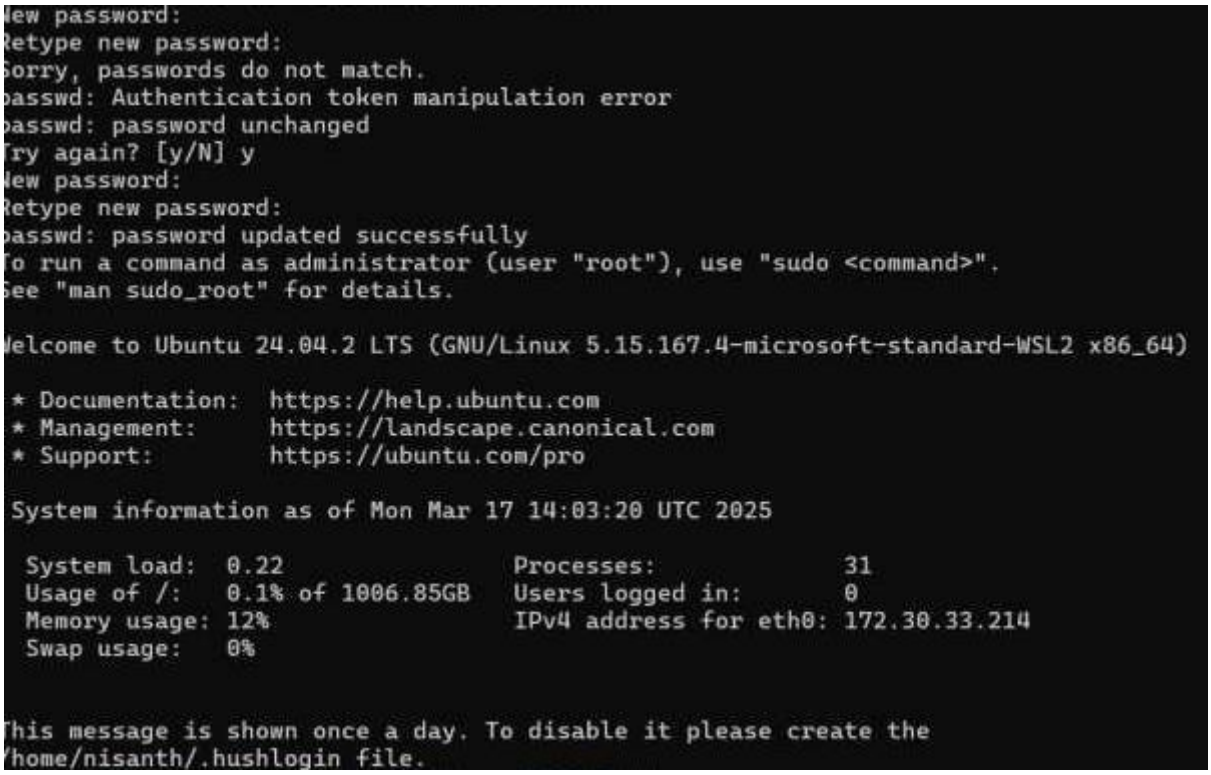
wsl --unregister Ubuntu

Access Windows Files in WSL:

cd /mnt/c

Conclusion

You have successfully installed and set up WSL with Ubuntu on Windows 10. You can now use the Ubuntu terminal to run Linux commands and manage your system efficiently.



Step-by-Step Guide to Creating a Freestyle Job in Jenkins to Install Nginx on a Local Ubuntu VM

Prerequisites for Setting Up a Freestyle Job to Install Nginx in Jenkins

Before creating the Freestyle Job, ensure that the following prerequisites are met:

1. Install Jenkins on Ubuntu (If Not Installed)

If Jenkins is not installed on your Ubuntu VM, follow these steps:

Step 1: Update Package Lists

sudo apt update -y

Step 2: Install Java (Required for Jenkins)

sudo apt install -y openjdk-17-jdk

Step 3: Verify Java Version

java -version

Step 4: Add Jenkins Repository Key

(Note: The apt-key add command is deprecated in newer Ubuntu versions. Use the correct method below.)

Correct Way to Add Jenkins Repository (Without apt-key)

Step 4.1: Add Jenkins GPG Key

wget -q -O- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkinskeyring.asc > /dev/null

Step 4.2: Add Jenkins Repository

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" |

sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

Step 5: Install Jenkins

sudo apt update -y

sudo apt install -y jenkins

Step 6: Start and Enable Jenkins Service

sudo systemctl start jenkins

sudo systemctl enable jenkins

Step 7: Check Jenkins Status

sudo systemctl status jenkins

2. Access Jenkins Web Interface

Jenkins will be available at http://<VM_IP>:8080

To Get the Jenkins Server URL, Follow These Steps:

Method 1: Check the Default URL

By default, Jenkins runs on port 8080. Open in a browser:

http://<your-server-ip>:8080

If you're on the same machine as Jenkins, use:

http://localhost:8080

Method 2: Get Server IP Address

hostname -I

or

ip a | grep inet

Method 3: Check Jenkins Logs (If Unable to Access)

sudo journalctl -u jenkins --no-pager --lines=50

Look for lines mentioning *"Jenkins is fully up and running"* and the URL.

3. Access Jenkins Web Interface and Log In

1. Open a browser and go to http://<JENKINS_SERVER_IP>:8080
2. Enter the username (admin) and the admin password retrieved from the following command:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

3. Choose *Install Suggested Plugins* (recommended) or manually select plugins.

4. Ensure Sudo Access for Jenkins User

Jenkins runs as a system user (jenkins). If your script requires sudo, allow Jenkins to execute commands without a password:

sudo visudo

Add the following line at the end of the file:

jenkins ALL=(ALL) NOPASSWD: ALL

Save and exit.

Step-by-Step Guide to Creating a Freestyle Job in Jenkins to Install Nginx

Step 1: Create a New Freestyle Job

1. Click on **New Item** from the Jenkins Dashboard.
2. Enter a name for the job, e.g., *Install-Nginx*.
3. Select **Freestyle project**.
4. Click **OK**.

Step 2: Configure the Job

Add Build Step

1. Scroll down to **Build** → Click *Add build step* → Select **Execute shell**.
2. Paste the following script in the command box:

```
#!/bin/bash

echo "Updating package lists..."

sudo apt update -y


echo "Installing Nginx..."

sudo apt install -y nginx


echo "Starting Nginx service..."

sudo systemctl start nginx
```

```
echo "Enabling Nginx to start on boot..."
sudo systemctl enable nginx

echo "Nginx Installation Completed!"
```

Step 3: Save and Run the Job

1. Click **Save**.
2. Click **Build Now**.
3. Check the **Console Output** to verify the installation.

Step 4: Verify the Installation

1. Check Nginx Status

```
systemctl status nginx
```

If running, you should see output like *"active (running)"*.

2. Open Nginx in Browser

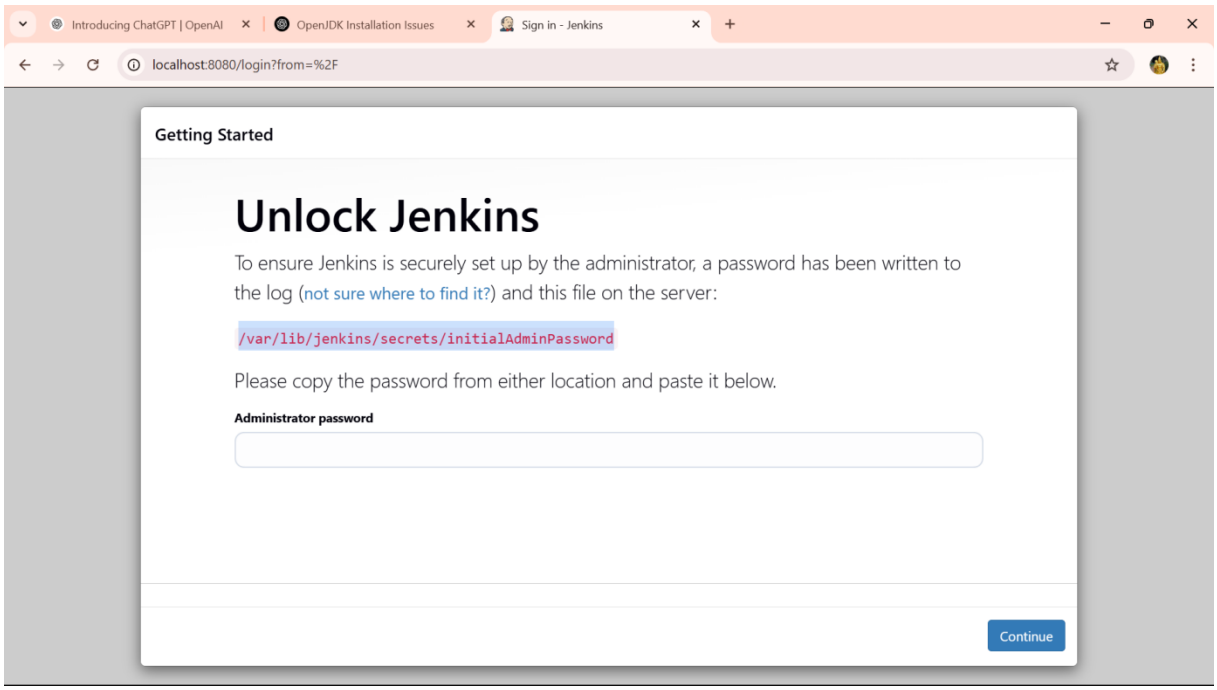
http://<VM_IP>

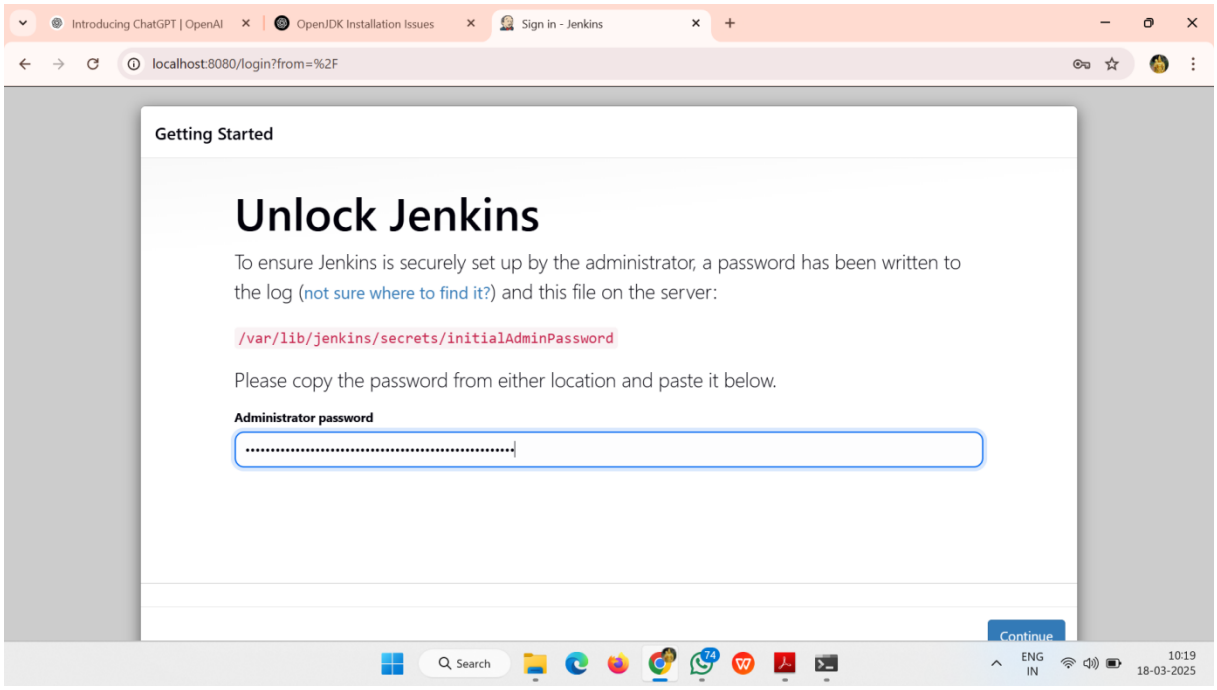
You should see the default Nginx welcome page.

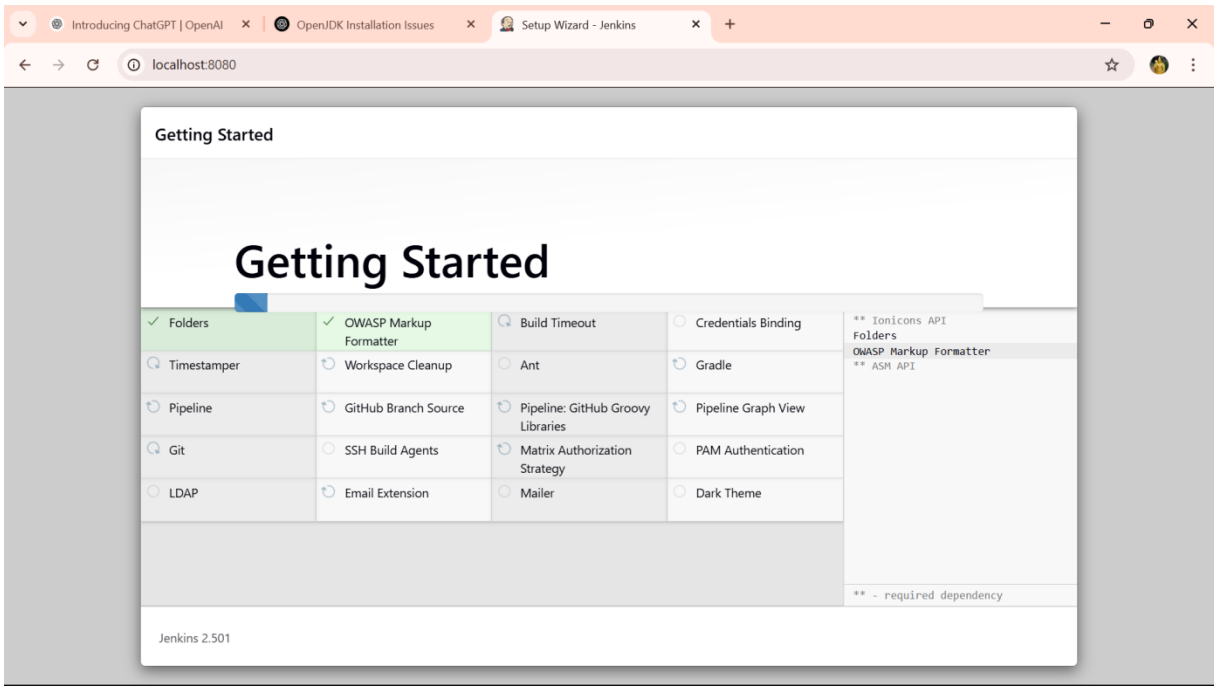
Conclusion

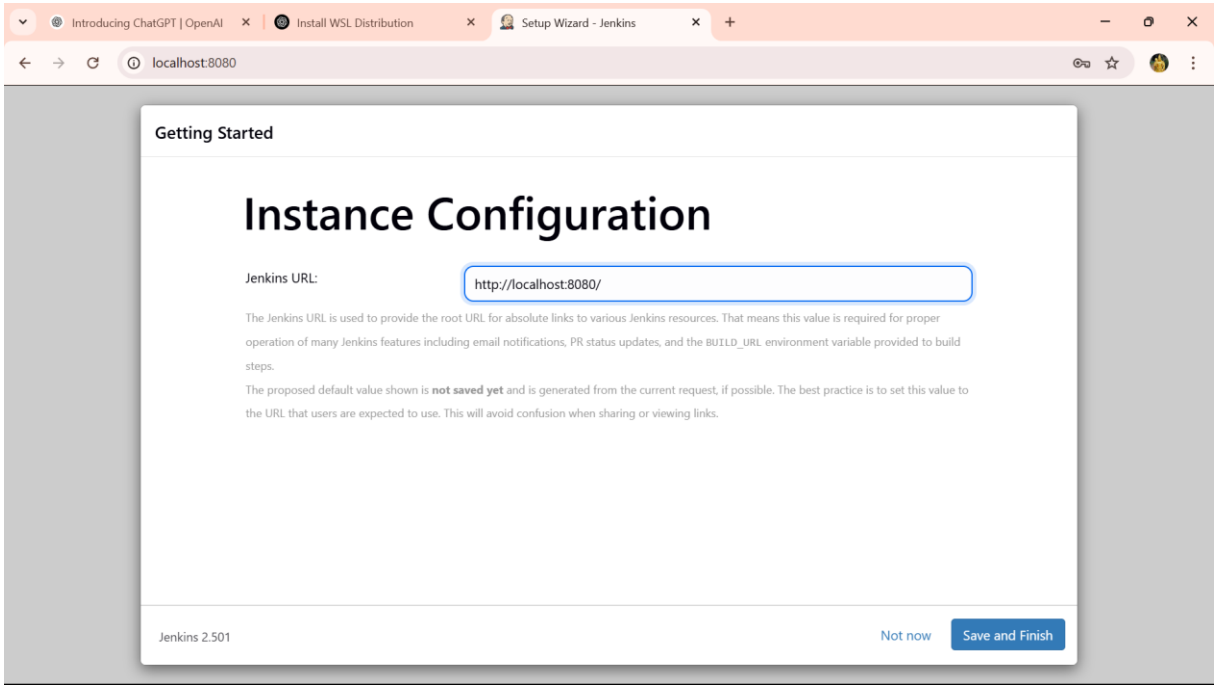
You have successfully set up a Jenkins Freestyle Job to install Nginx on a local Ubuntu VM. This guide covers everything from Jenkins installation, configuration, and running the job to verify that Nginx is installed and running correctly.

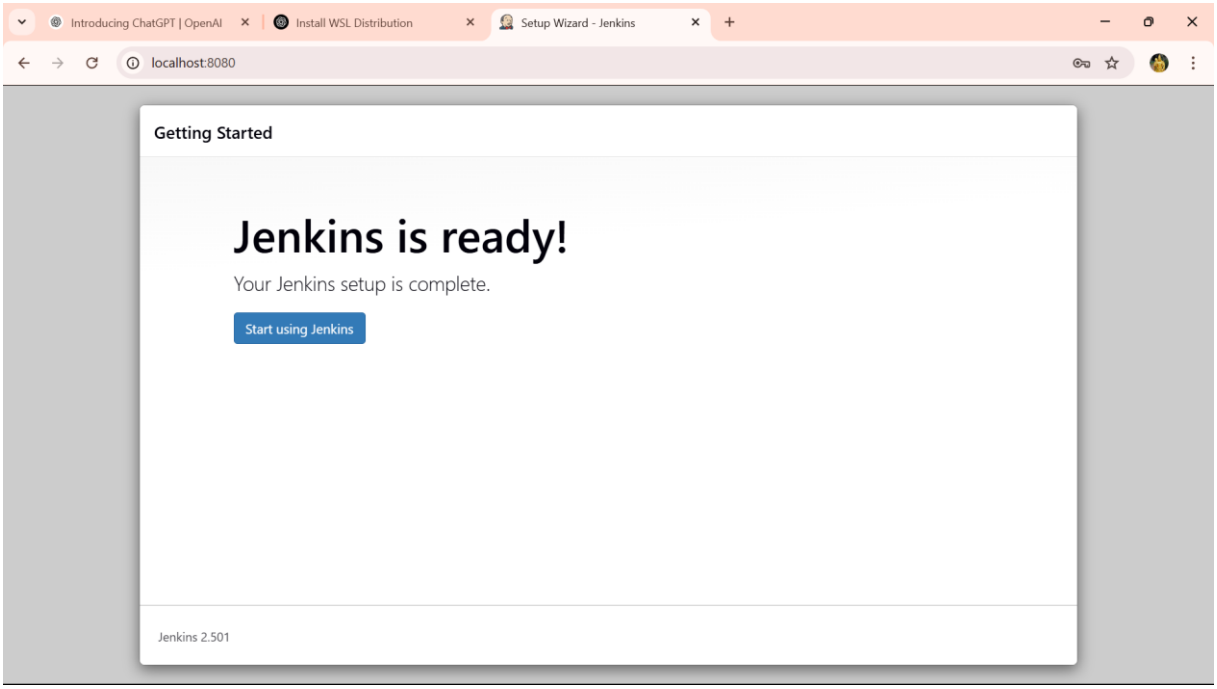
Now, your Jenkins automation is ready to deploy Nginx effortlessly!

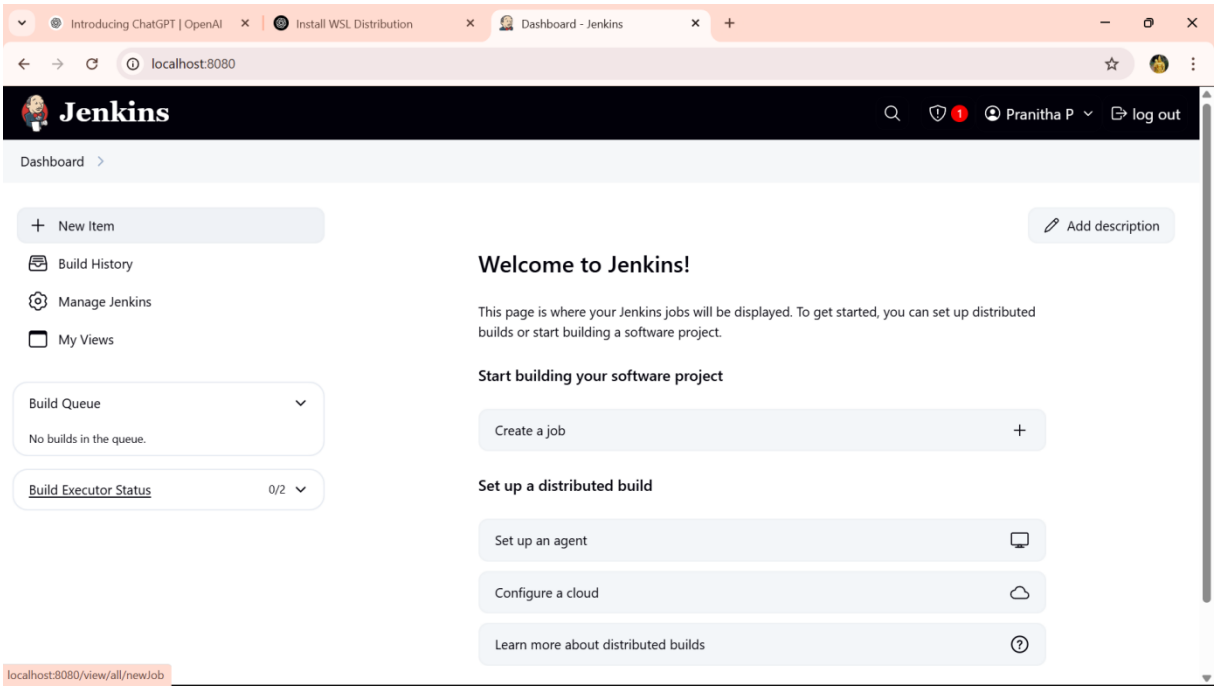

Introducing ChatGPT | OpenAI
OpenJDK Installation Issues
Sign in - Jenkins
localhost:8080/login?from=%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue

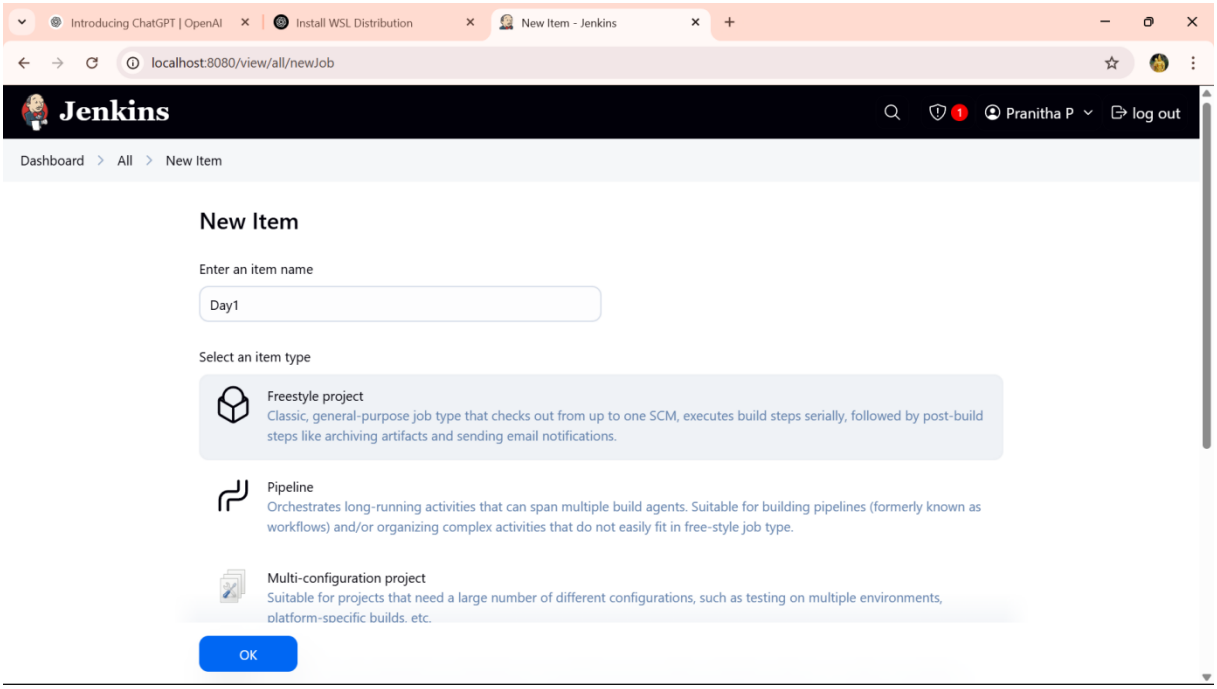

Introducing ChatGPT | OpenAI
OpenJDK Installation Issues
Sign in - Jenkins
localhost:8080/login?from=%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue
Search
ENG
IN
10:19
18-03-2025

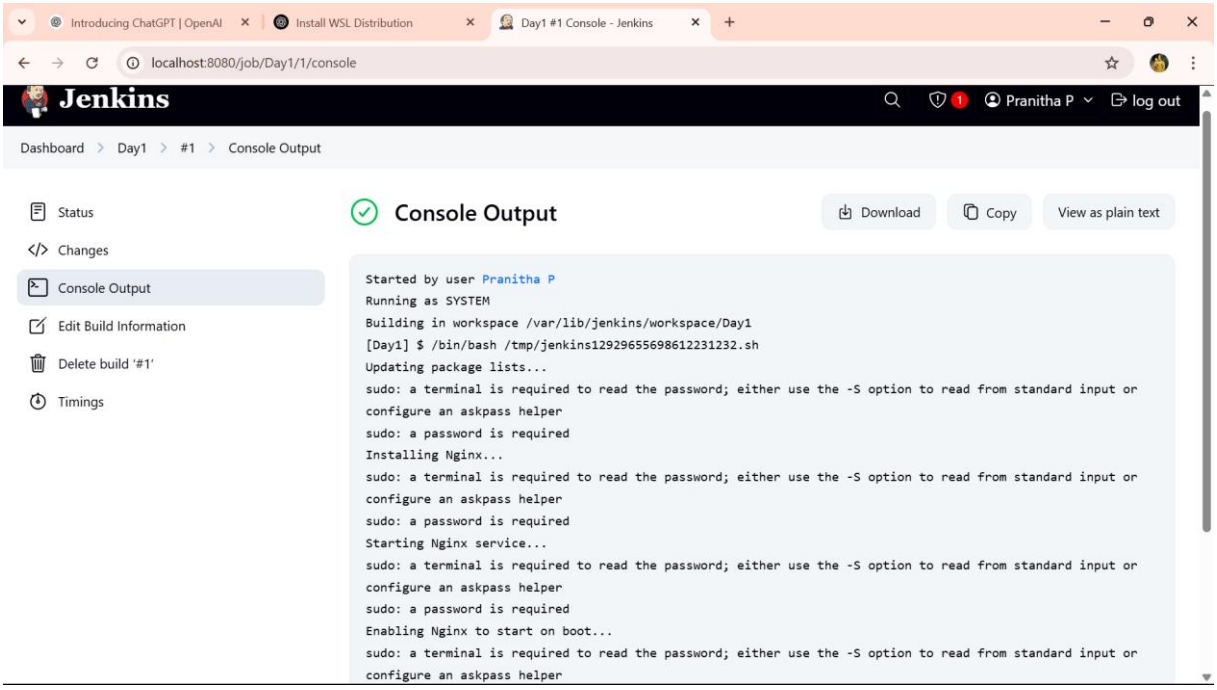

Introducing ChatGPT | OpenAI
OpenJDK Installation Issues
Setup Wizard - Jenkins
localhost:8080
Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
Dark Theme
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** - required dependency
Jenkins 2.501

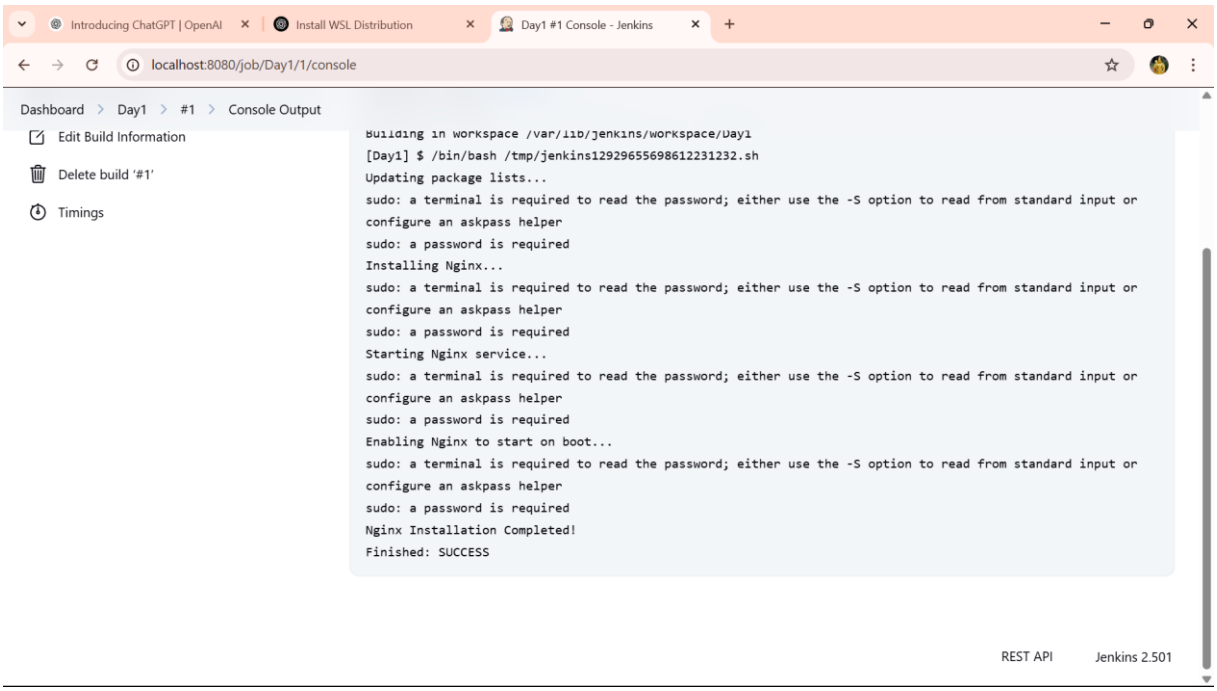

Introducing ChatGPT | OpenAI
Install WSL Distribution
Setup Wizard - Jenkins
localhost:8080
Getting Started
Instance Configuration
Jenkins URL:
http://localhost:8080/
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for 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.
Jenkins 2.501
Not now
Save and Finish

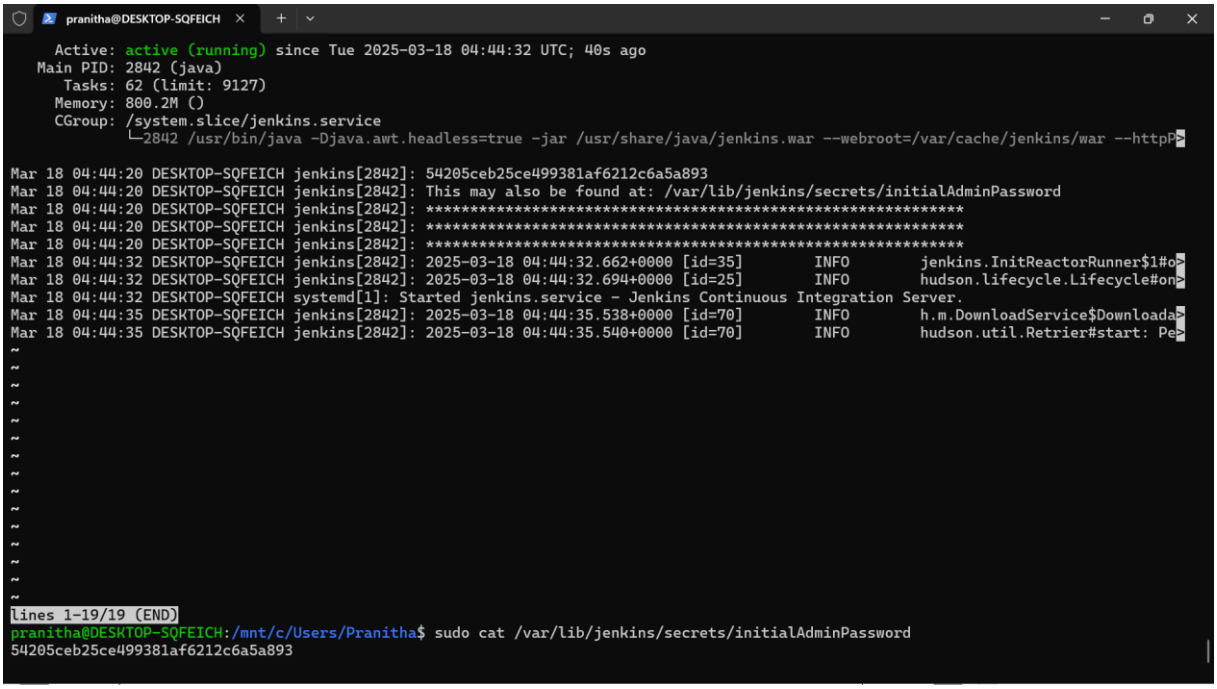

Introducing ChatGPT | OpenAI
Install WSL Distribution
Setup Wizard - Jenkins
localhost:8080
Getting Started
Jenkins is ready!
Your Jenkins setup is complete.
Start using Jenkins
Jenkins 2.501

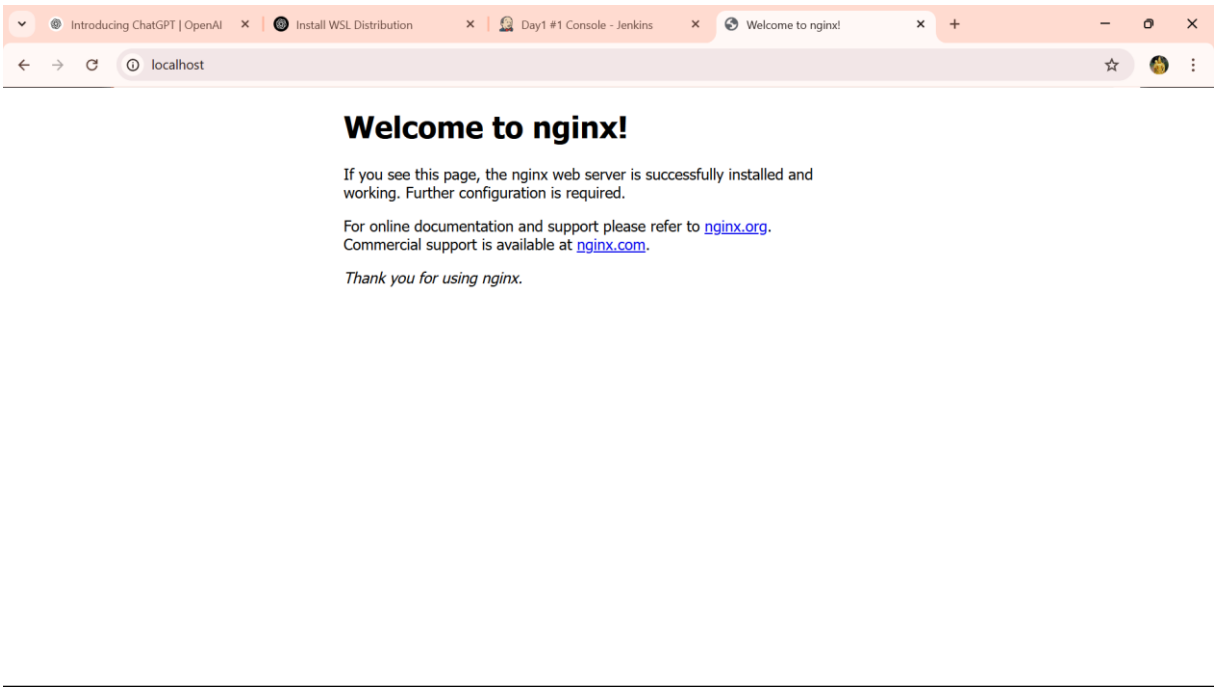

Introducing ChatGPT | OpenAI
Install WSL Distribution
Dashboard - Jenkins
localhost:8080
Jenkins
Pranitha P
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
0/2
Add description
Welcome to Jenkins!
This page is where your Jenkins 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
localhost:8080/view/all/newJob


Introducing ChatGPT | OpenAI
Install WSL Distribution
New Item - Jenkins
localhost:8080/view/all/newJob
Jenkins
Pranitha P
log out
Dashboard > All > New Item
New Item
Enter an item name
Day1
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.
OK


Introducing ChatGPT | OpenAI
Install WSL Distribution
Day1 #1 Console - Jenkins
localhost:8080/job/Day1/1/console
Jenkins
Pranitha P
log out
Dashboard > Day1 > #1 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Timings
Console Output
Download
Copy
View as plain text
Started by user Pranitha P
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Day1
[Day1] $ /bin/bash /tmp/jenkins1292965569861223 1232.sh
Updating package lists...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Installing Nginx...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Starting Nginx service...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Enabling Nginx to start on boot...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper


Introducing ChatGPT | OpenAI
Install WSL Distribution
Day1 #1 Console - Jenkins
localhost:8080/job/Day1/1/console
Dashboard > Day1 > #1 > Console Output
Edit Build Information
Delete build '#1'
Timings
Building in workspace /var/lib/jenkins/workspace/Day1
[Day1] $ /bin/bash /tmp/jenkins1292965569861223 1232.sh
Updating package lists...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Installing Nginx...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Starting Nginx service...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Enabling Nginx to start on boot...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Nginx Installation Completed!
Finished: SUCCESS
REST API
Jenkins 2.501


pranitha@DESKTOP-SQFEICH
Active: active (running) since Tue 2025-03-18 04:44:32 UTC; 40s ago
Main PID: 2842 (java)
Tasks: 62 (limit: 9127)
Memory: 800.2M ()
CGroup: /system.slice/jenkins.service
└─2842 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpP>
Mar 18 04:44:20 DESKTOP-SQFEICH jenkins[2842]: 54205ceb25ce499381af6212c6a5a893
Mar 18 04:44:20 DESKTOP-SQFEICH jenkins[2842]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 18 04:44:20 DESKTOP-SQFEICH jenkins[2842]: **
Mar 18 04:44:20 DESKTOP-SQFEICH jenkins[2842]: **
Mar 18 04:44:20 DESKTOP-SQFEICH jenkins[2842]: **
Mar 18 04:44:32 DESKTOP-SQFEICH jenkins[2842]: 2025-03-18 04:44:32.662+0000 [id=35] INFO jenkins.InitReactorRunner$1#o>
Mar 18 04:44:32 DESKTOP-SQFEICH jenkins[2842]: 2025-03-18 04:44:32.694+0000 [id=25] INFO hudson.lifecycle.Lifecycle#on>
Mar 18 04:44:32 DESKTOP-SQFEICH systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 18 04:44:35 DESKTOP-SQFEICH jenkins[2842]: 2025-03-18 04:44:35.538+0000 [id=70] INFO h.m.DownloadService$Downloada>
Mar 18 04:44:35 DESKTOP-SQFEICH jenkins[2842]: 2025-03-18 04:44:35.540+0000 [id=70] INFO hudson.util.Retrier#start: Pe>
lines 1-19/19 (END)
pranitha@DESKTOP-SQFEICH:/mnt/c/Users/Pranitha$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
54205ceb25ce499381af6212c6a5a893


Introducing ChatGPT | OpenAI
Install WSL Distribution
Day1 #1 Console - Jenkins
Welcome to nginx!
localhost
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.