Task: Setting up Jenkins for Continuous Integration/Continuous Deployment (CI/CD)

Description: Configure Jenkins to automate the build, test, and deployment processes for your

application.

1. Automated Testing in CI:

• Test Case 1: Integrate a sample application with automated unit tests into the CI

pipeline.

Prerequisites:

* Install Jenkins:
  + Follow the official Jenkins installation instructions for your platform: Jenkins Installation

# Update package lists

sudo apt-get update

# Install Java 11

sudo apt-get install -y openjdk-11-jdk

# Download Jenkins key and add it to system

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

# Add Jenkins to system package source list

sudo sh -c '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'

# Update package lists again to include Jenkins and install it

sudo apt-get update

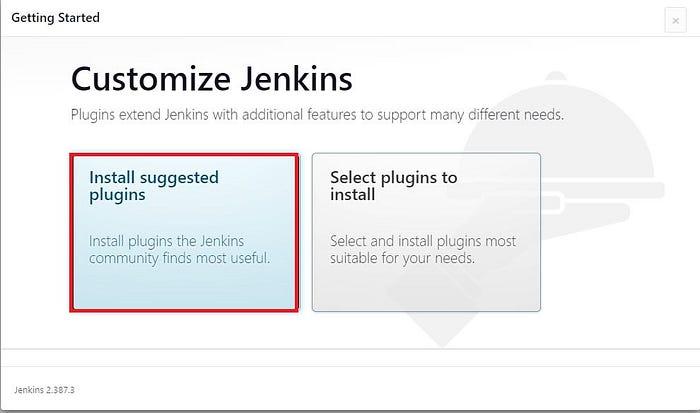
sudo apt-get install -y jenkins

# Verify Jenkins is installed and working

sudo systemctl status jenkins

Install Required Plugins:

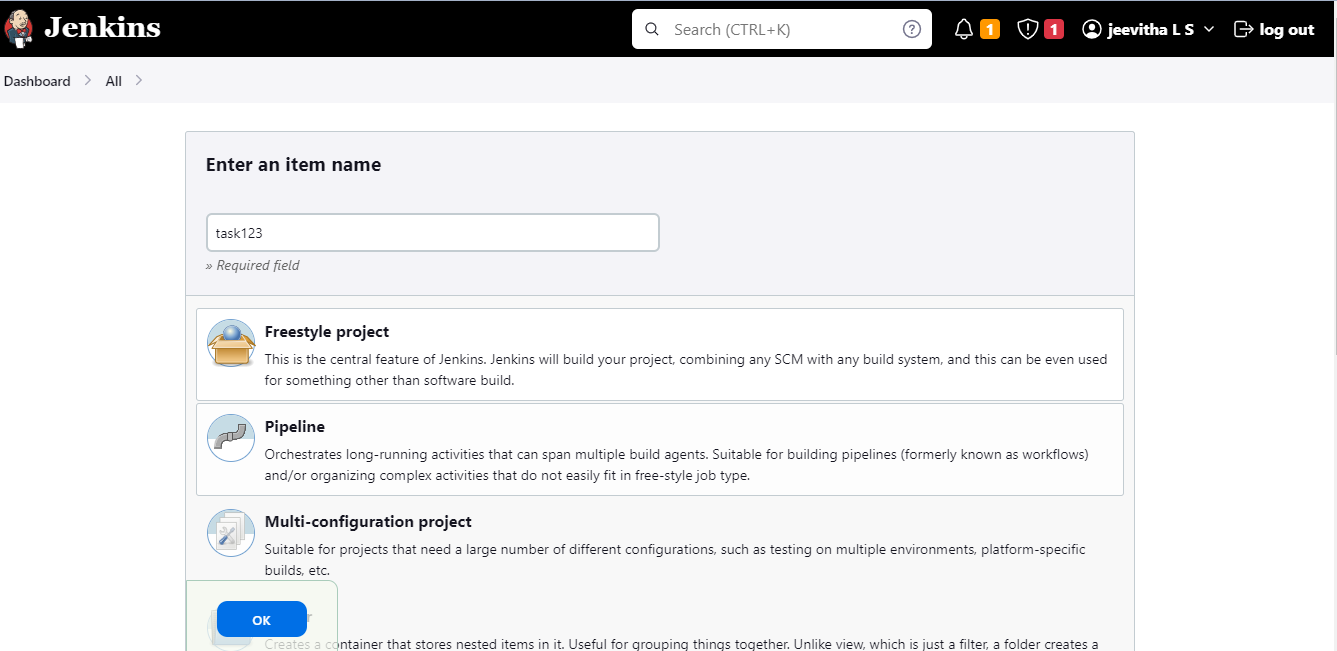
* + Install necessary plugins for version control systems (e.g., Git), build tools, and deployment platforms.

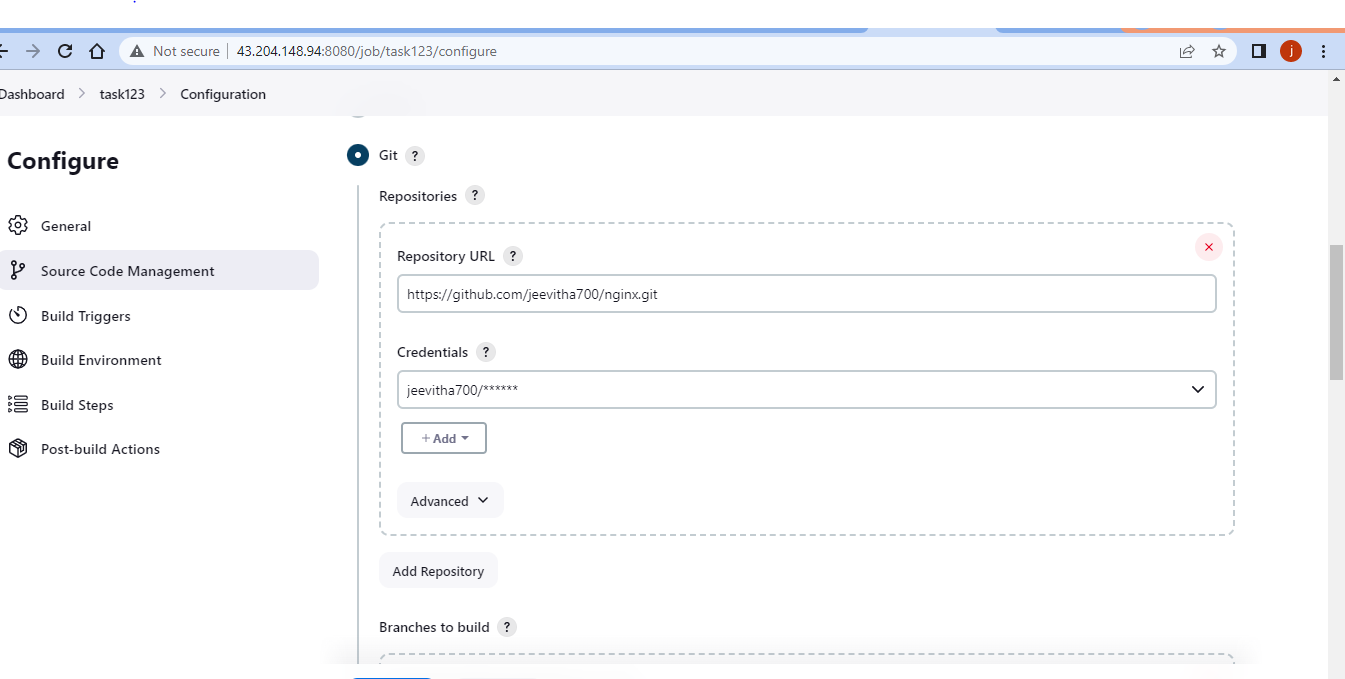


Configuration Steps:

1. Automated Testing in CI:

* Configure Jenkins Job for CI:
  + Open Jenkins and create a new Freestyle project or Pipeline job.
  + Set up source code management (e.g., Git).
  + Configure build triggers to run the job on code changes.
* Integrate Automated Tests:
  + Add build steps to run automated unit tests.
  + Use testing frameworks suitable for your application (e.g., JUnit for Java).
  + Configure post-build actions to report test results.



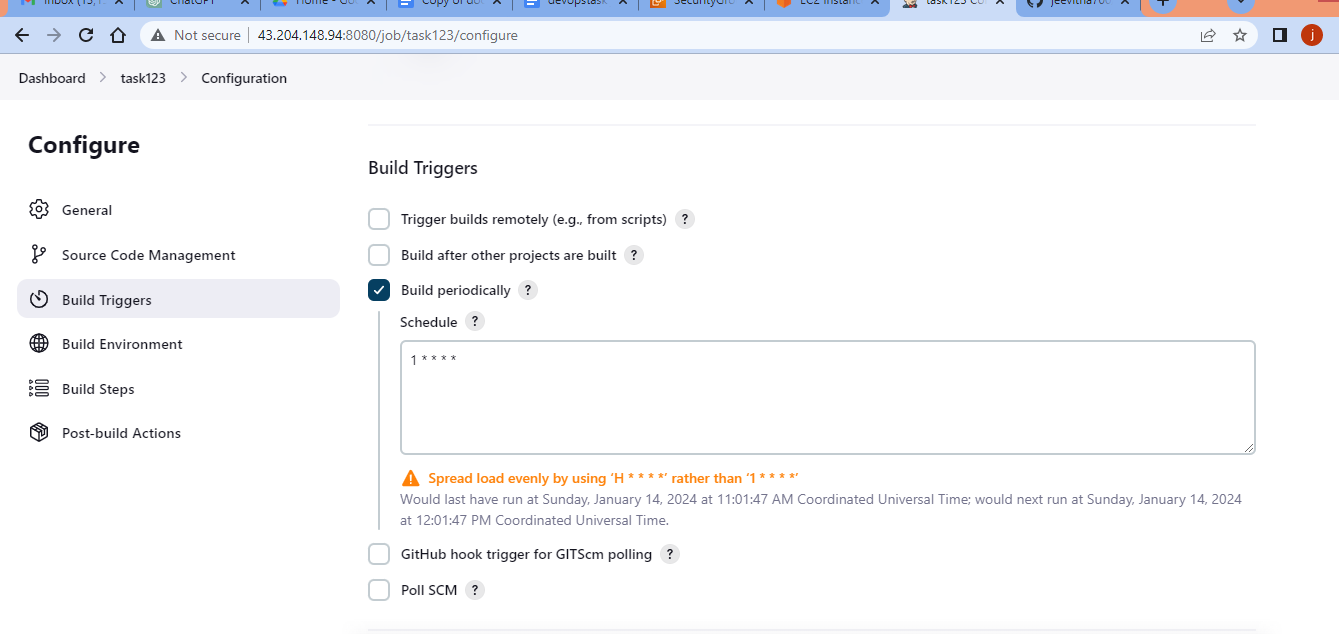




2. Scheduled Builds:

• Test Case 2: Schedule a periodic build (e.g., nightly build) and verify that the CI pipeline

Scheduled Builds:

* Configure Periodic Builds:
  + In your Jenkins job configuration, enable periodic builds.Use the Build periodically option to specify the schedule (e.g., nightly).
  + 

3. Rollback Mechanism in CD:

• Test Case 3: Deploy a faulty version intentionally in the CD pipeline, trigger a rollback,

and confirm that the3. Rollback Mechanism in CD:

* Configure CD Pipeline:
  + Create a new Jenkins job for deployment (CD).
  + Use plugins or scripts to deploy your application to the target environment.
* Introduce Faulty Deployment:
  + Modify the deployment script or intentionally introduce an error in the code.
  + Run the CD pipeline.
* Implement Rollback:
  + Configure the CD pipeline to detect deployment failures.
  + Add a post-build action or script to trigger a rollback (e.g., revert to the last stable version).