

Jenkins Freestyle Project with GitHub Webhook Automation

Author: Oluwaseun Osunsola

LinkedIn: <https://www.linkedin.com/in/oluwaseun-osunsola-95539b175/>

Project Repository: <https://github.com/Oluwaseunoa/DevOps-Projects/tree/main/Jenkins-Projects>

Environment & Tools: AWS EC2 (Ubuntu 22.04 LTS), Jenkins 2.462.x (LTS), GitHub

Project Overview

This project demonstrates how to configure a **Jenkins Freestyle Job** integrated with **GitHub Source Code Management (SCM)** and automate build execution using **GitHub Webhooks**. The goal is to implement a **basic Continuous Integration (CI) pipeline** where Jenkins automatically triggers builds whenever changes are pushed to a GitHub repository.

Project Objectives

- Create and configure a Jenkins Freestyle Job
 - Integrate Jenkins with a GitHub repository using Git SCM
 - Trigger Jenkins builds manually and automatically
 - Configure GitHub Webhooks to enable event-driven builds
 - Validate automated build execution via console logs
-

Tools & Technologies Used

- **Jenkins** – Automation Server
 - **GitHub** – Source Code Repository
 - **Git** – Version Control System
 - **GitHub Webhooks** – Build Trigger Mechanism
 - **Web Browser** – Jenkins & GitHub Access
-

Project Architecture (Logical Flow)

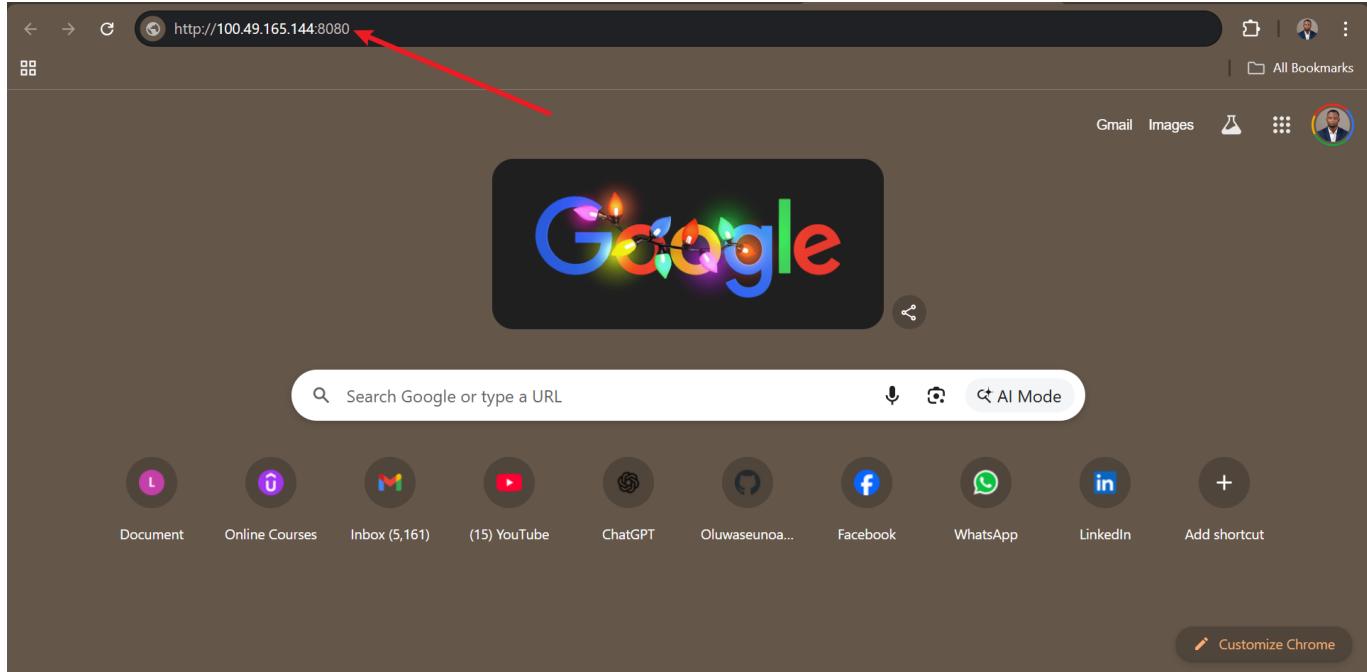
1. Developer pushes code to GitHub
 2. GitHub sends a webhook event to Jenkins
 3. Jenkins pulls the latest code
 4. Jenkins executes the configured job
 5. Build status and logs are generated
-

Step-by-Step Project Implementation

Step 1: Open Jenkins in the Browser

Navigate to the Jenkins server URL using a web browser.

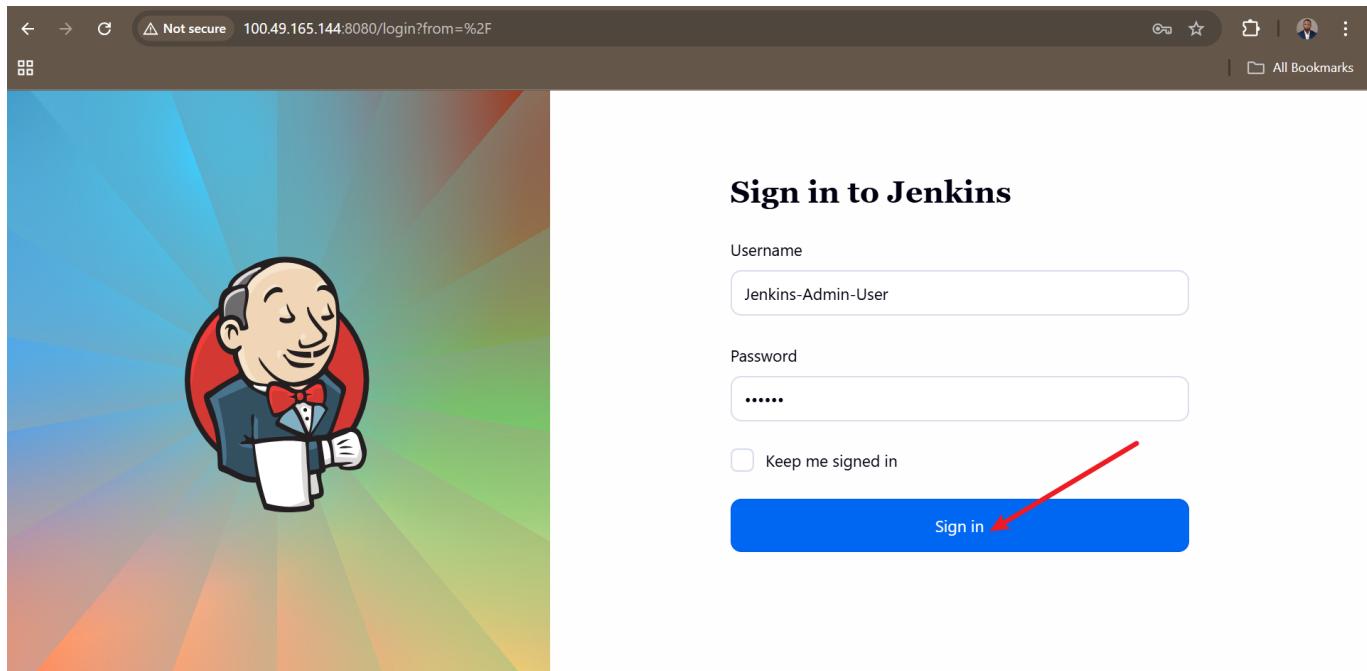
Screenshot:



Step 2: Sign In to Jenkins

Authenticate with valid Jenkins credentials.

Screenshot:



Step 3: Jenkins Dashboard Displayed

After login, the Jenkins dashboard is displayed, showing existing jobs and options.

Screenshot:

The screenshot shows the Jenkins dashboard at the URL 100.49.165.144:8080. The top navigation bar indicates "Not secure". The dashboard features a sidebar with "New Item" and "Build History" buttons, and a search bar with an "Add description" button. A main table displays the "Java-Pipeline" job, which is marked as "Success" (green checkmark icon) and "Warning" (yellow sun icon). The table includes columns for Status (S), Warning (W), Name (Java-Pipeline), Last Success (2 days 19 hr #3), Last Failure (N/A), and Last Duration (8.1 sec). Below the table is a horizontal scroll bar. At the bottom, there are links for "REST API" and "Jenkins 2.528.3".

Step 4: Create a New Jenkins Job

Click **New Item** to create a new job.

Screenshot:

This screenshot is identical to the one above, showing the Jenkins dashboard at 100.49.165.144:8080. A red arrow points to the "New Item" button in the sidebar. The rest of the interface, including the table with the Java-Pipeline job and the bottom links, remains the same.

Step 5: Name the Job and Select Freestyle Project

- Job Name: **my-first-job**
- Job Type: **Freestyle Project**

Screenshot:

The screenshot shows the Jenkins 'New Item' creation interface. At the top, there's a search bar with 'Not secure' and the URL '100.49.165.144:8080/view/all/newJob'. Below it, the Jenkins logo and 'Jenkins / All / New Item' are visible. The main area is titled 'New Item' with a sub-instruction 'Enter an item name' and a text input field containing 'my-first-job'. Under 'Select an item type', three options are listed: 'Freestyle project' (selected and highlighted with a red border), 'Pipeline', and 'Multi-configuration project'. A red arrow points from the bottom-left towards the 'OK' button. The 'OK' button is blue with white text.

Step 6: Access Project Configuration Page

The job configuration page opens in a new tab. Log in to GitHub in preparation for repository setup.

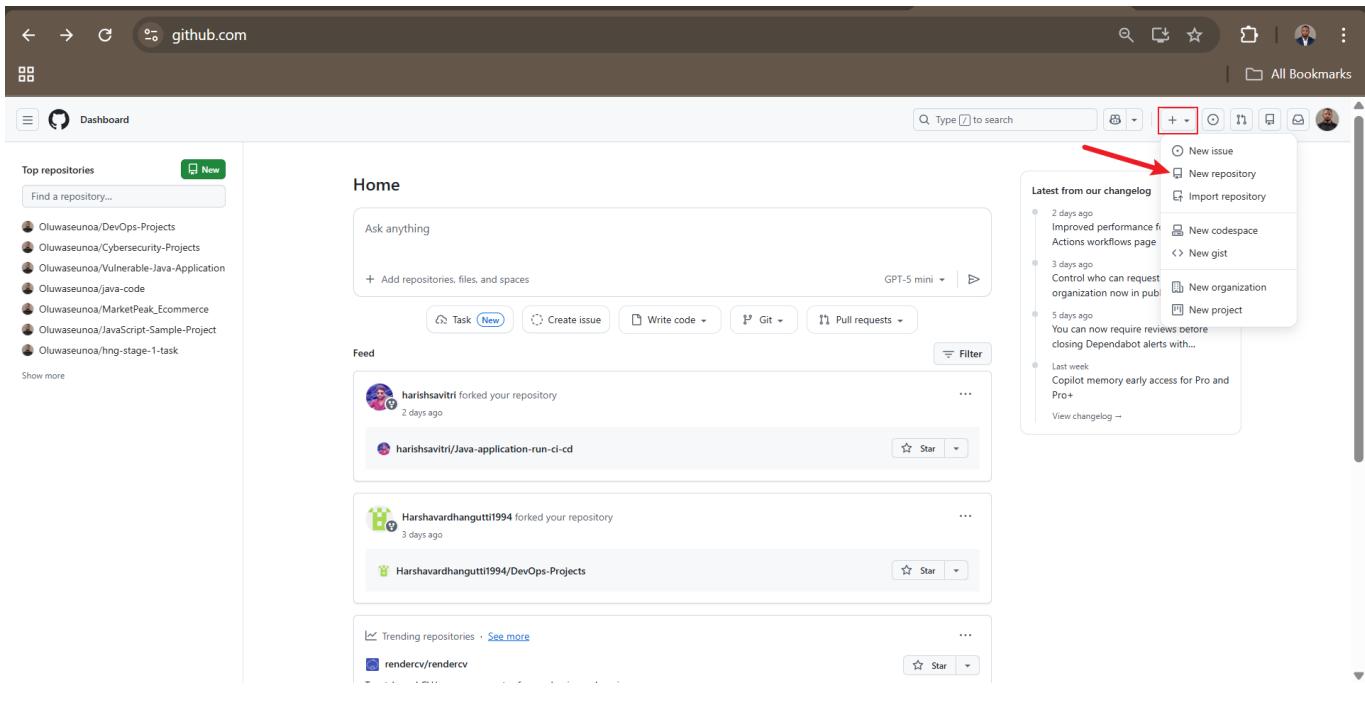
Screenshot:

The screenshot shows the Jenkins 'Configuration' page for the 'my-first-job' project. The URL is '100.49.165.144:8080/job/my-first-job/configure'. The 'General' tab is selected. On the left, a sidebar lists 'Configure' (selected) and 'General'. The 'General' section contains fields for 'Description' (empty) and 'Plain text Preview' (disabled). It also includes checkboxes for 'Discard old builds', 'GitHub project', 'This project is parameterized', and 'Throttle builds'. At the bottom are 'Save' and 'Apply' buttons. A red box highlights the 'Enabled' checkbox, which is checked. The Jenkins logo and 'Jenkins / my-first-job / Configuration' are at the top left.

Step 7: Create a New GitHub Repository

Click the + icon on GitHub and select **New repository**.

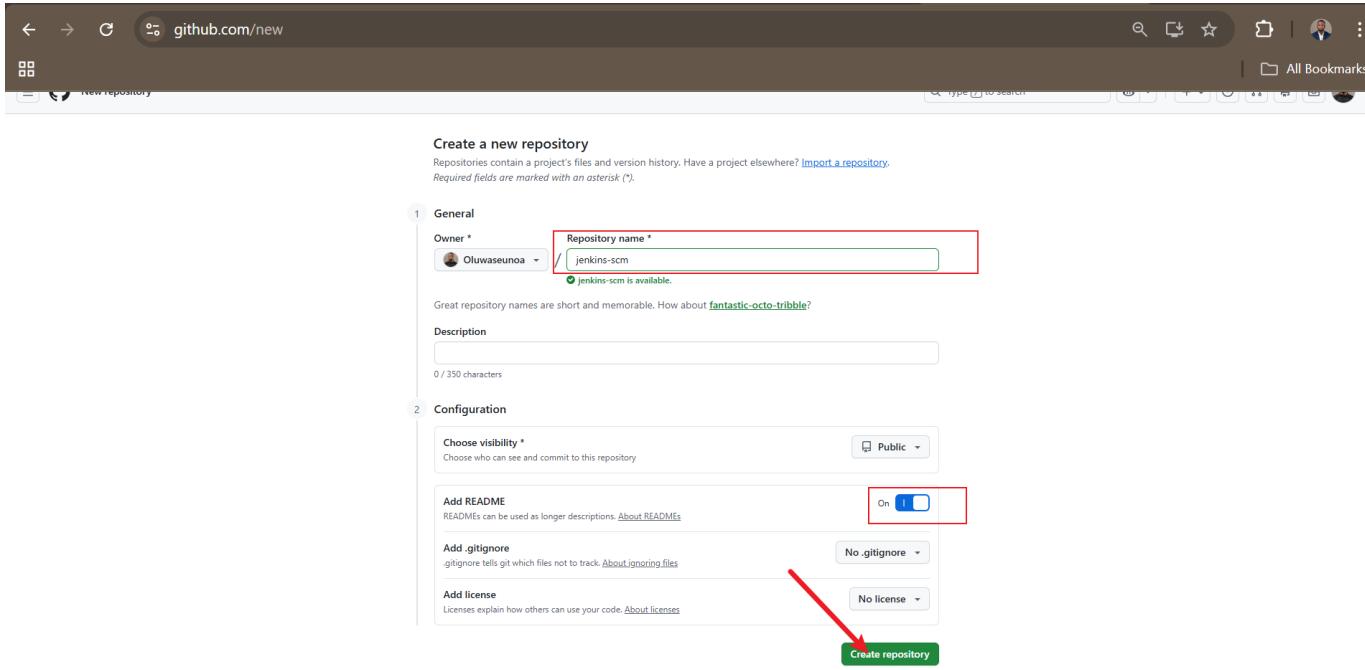
Screenshot:



Step 8: Initialize the Repository

- Repository Name: **jenkins-scm**
- Initialize with **README.md**

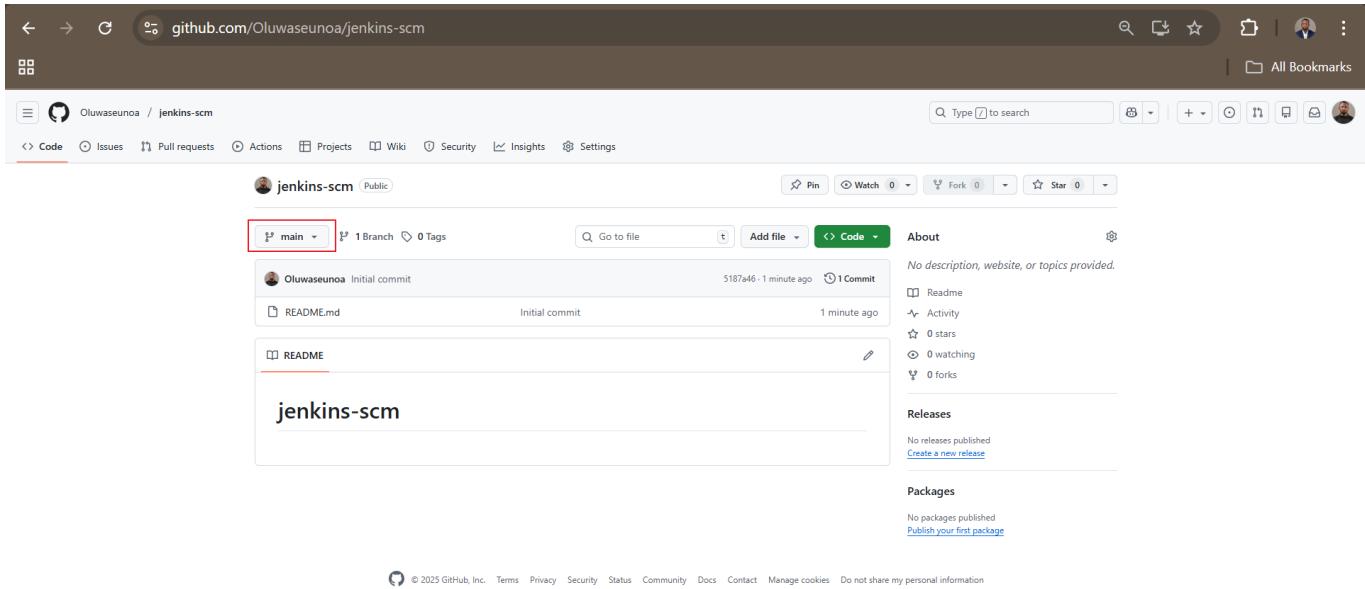
Screenshot:



Step 9: Repository Successfully Created

The repository is created with the default branch set to **main**.

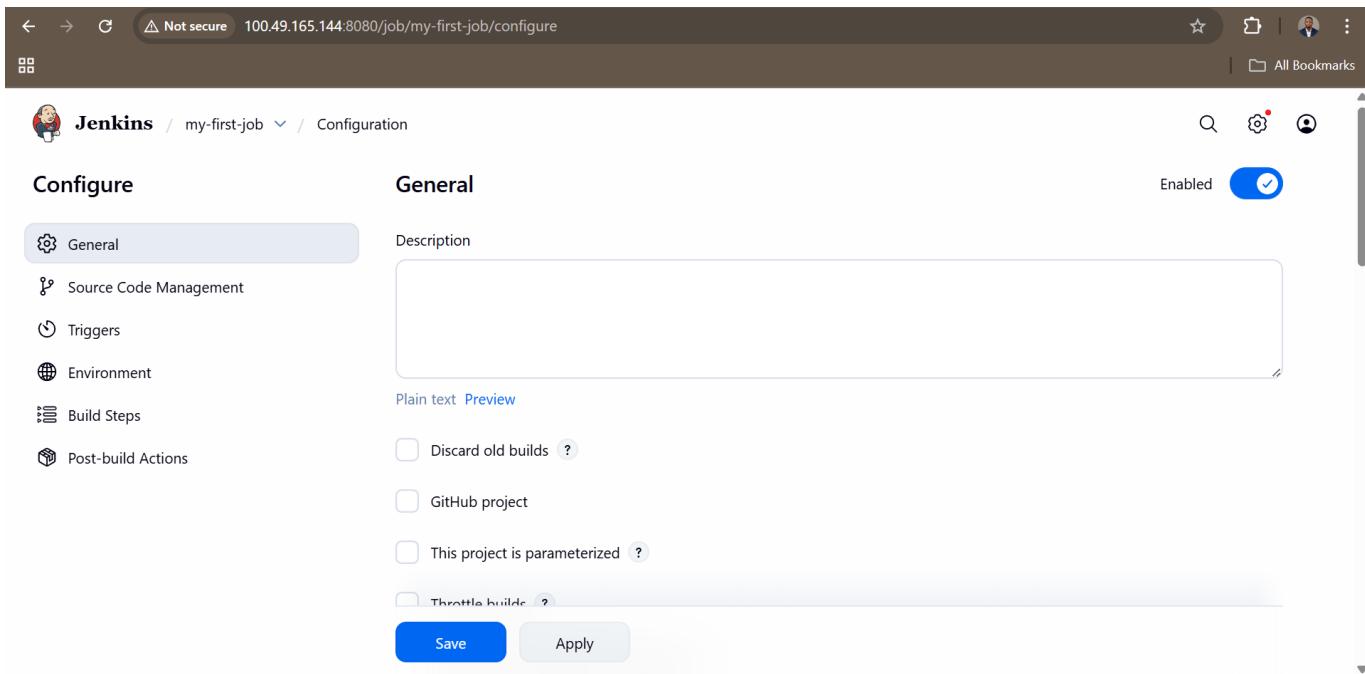
Screenshot:



Step 10: Return to Jenkins Job Configuration

Switch back to the Jenkins job configuration page.

Screenshot:



Step 11: Configure Source Code Management

- Scroll to **Source Code Management**
- Select **Git**

Screenshot:

The screenshot shows the Jenkins job configuration page for 'my-first-job'. On the left, a sidebar lists 'General', 'Source Code Management' (which is selected), 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The main content area is titled 'Configure' and has a sub-section 'Connect and manage your code repository to automatically pull the latest code for your builds.' It shows a 'Git' configuration with a 'Repository URL' field containing a placeholder 'https://github.com/Oluwaseunao/jenkins-scm.git' and a note 'Copied!'. Below it is a 'Credentials' dropdown set to '- none -'. At the bottom are 'Save' and 'Apply' buttons.

Step 12: Copy GitHub Repository URL

Click **Code** in GitHub and copy the HTTPS repository URL.

Screenshot:

The screenshot shows a GitHub repository page for 'jenkins-scm'. In the top right, there's a 'Code' button with a dropdown menu open. The 'HTTPS' option is highlighted and has a red border around its URL: 'https://github.com/Oluwaseunao/jenkins-scm.git'. Other options in the menu include 'Local', 'Codespaces', 'SSH', and 'GitHub CLI'. To the right of the code menu, there's a note 'Copied!' with a checkmark icon. The repository details on the right show 1 branch, 0 tags, 0 forks, and 0 releases. At the bottom, there's a copyright notice for GitHub, Inc. and links for Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and Do not share my personal information.

Step 13: Paste Repository URL into Jenkins

Paste the copied GitHub URL into **Repository URL** field.

Screenshot:

The screenshot shows the Jenkins job configuration page for 'my-first-job'. The 'Source Code Management' section is selected. Under 'Git', the 'Repository URL' is set to `https://github.com/Oluwaseunoa/jenkins-scm.git`. The 'Save' button is highlighted with a red box.

Step 14: Specify Branch and Save

- Branch Specifier: `*/main`
- Click **Save**

Screenshot:

The screenshot shows the Jenkins job configuration page for 'my-first-job'. The 'Source Code Management' section is selected. Under 'Branches to build', the 'Branch Specifier' is set to `*/main`. A red arrow points to the 'Save' button at the bottom.

Step 15: Trigger Manual Build

Click **Build Now** to run the first build manually.

Screenshot:

The screenshot shows the Jenkins interface for the 'my-first-job' project. The top navigation bar indicates the URL is 100.49.165.144:8080/job/my-first-job/. The left sidebar contains links for Status, Changes, Workspace, Build Now (which has a red arrow pointing to it), Configure, Delete Project, and Rename. The main content area is titled 'my-first-job' and 'Permalinks'. Below this is a 'Builds' section with a sub-section for 'No builds'. The status bar at the bottom right includes icons for search, settings, and user profile.

Step 16: Observe Build Status

The build appears in the **Build History** section and completes successfully.

Screenshot:

The screenshot shows the Jenkins interface for the 'my-first-job' project after a build has been triggered. The top navigation bar and sidebar are identical to the previous screenshot. The main content area now shows a green checkmark icon next to 'my-first-job'. The 'Builds' section is highlighted with a red box and displays the following information: 'Builds' (dropdown menu), 'Filter' input field, 'Today' date range, and a list item '#1 6:16AM' with a green checkmark. The status bar at the bottom right includes icons for search, settings, and user profile.

Step 17: View Console Output

Click the build number and view the console output.

Screenshot:

The screenshot shows the Jenkins interface for a job named "my-first-job". The top navigation bar indicates the URL is 100.49.165.144:8080/job/my-first-job/. The left sidebar has links for Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. Under Builds, there is a list for Today with entry #1 6:16AM. A context menu is open over this entry, with "Console Output" highlighted. Red arrows point from the "Console Output" option in the menu to the "Console Output" link in the sidebar and to the "Console Output" section of the main content area.

Permalinks

- Last build (#1), 5 min 24 sec ago
- Last stable build (#1), 5 min 24 sec ago
- Last successful build (#1), 5 min 24 sec ago
- Last completed build (#1), 5 min 24 sec ago

Builds

Today

#1 6:16AM

Console Output

Step 18: Build Execution Completed Successfully

The job finishes successfully. Navigate back to the job page.

Screenshot:

The screenshot shows the Jenkins interface for the "Console Output" of build #1. The top navigation bar indicates the URL is 100.49.165.144:8080/job/my-first-job/1/console. The left sidebar has links for Status, Changes, Console Output, Edit Build Information, Delete build #1, Timings, and Git Build Data. The main content area displays the build log, which starts by cloning a Git repository from https://github.com/Oluwaseunoa/jenkins-scm.git. It shows the execution of git commands like init, fetch, config, and checkout. The log concludes with "Finished: SUCCESS". Red arrows point from the "Console Output" link in the sidebar to the "Console Output" link in the top navigation bar and to the "Console Output" section of the main content area.

Console Output

```

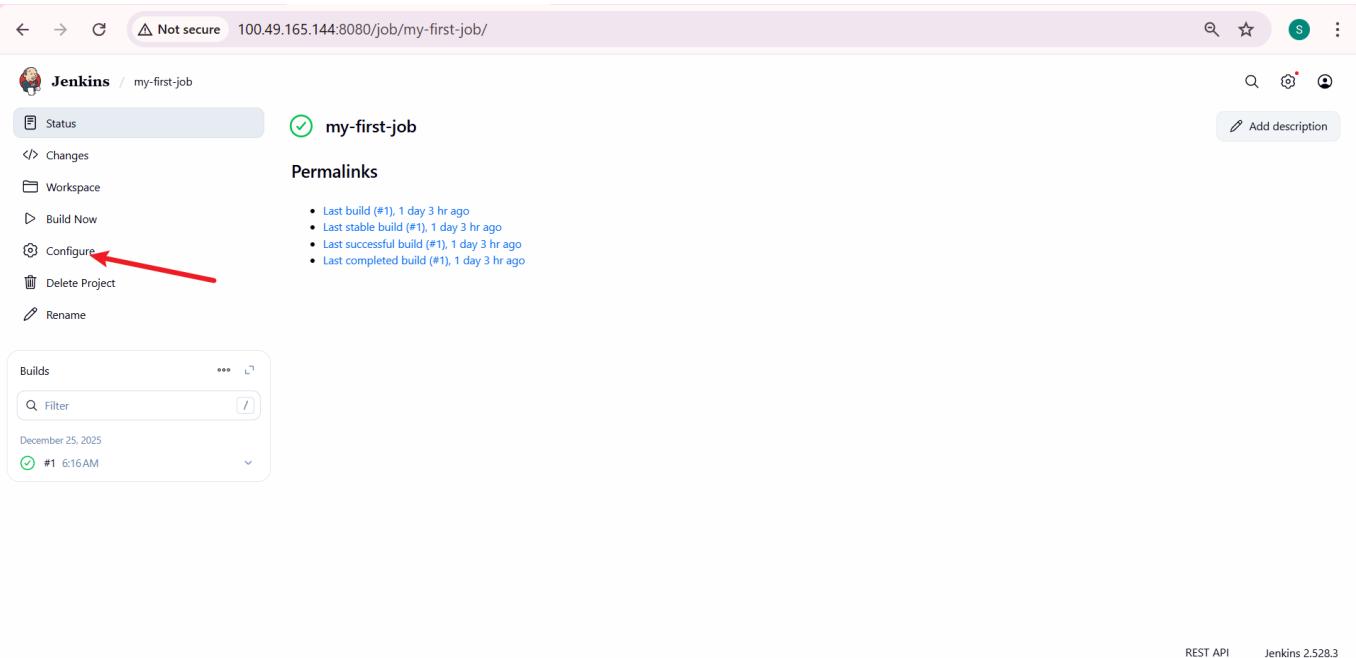
Started by user Jenkins-Admin-User
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/my-first-job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Oluwaseunoa/jenkins-scm.git
> git init /var/lib/jenkins/workspace/my-first-job # timeout=10
Fetching upstream changes from https://github.com/Oluwaseunoa/jenkins-scm.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch -tags --force --progress -- https://github.com/Oluwaseunoa/jenkins-scm.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/Oluwaseunoa/jenkins-scm.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 5187a46d647c5012fb916019c374c8226a9855f2 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 5187a46d647c5012fb916019c374c8226a9855f2 # timeout=10
Commit message: "Initial commit"
First time build. Skipping changelog.
Finished: SUCCESS

```

Step 19: Re-open Job Configuration

Click **Configure** to enable automation.

Screenshot:

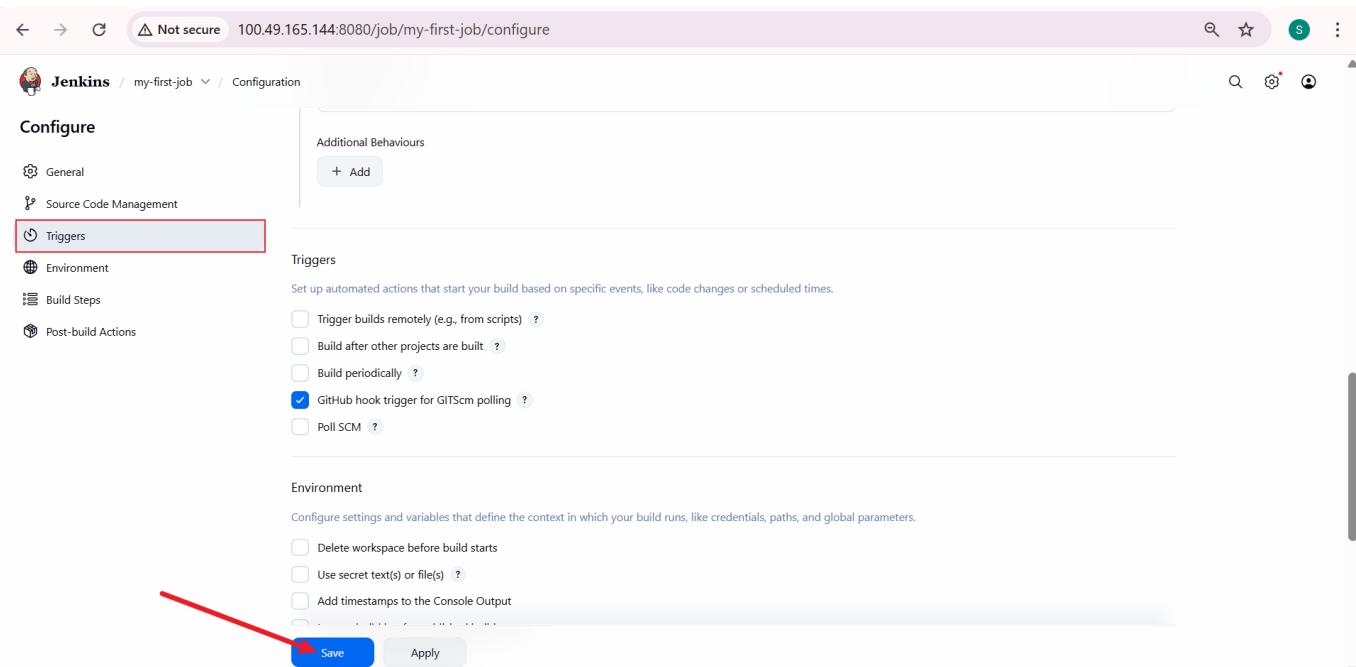


The screenshot shows the Jenkins interface for a job named "my-first-job". On the left, there's a sidebar with options like Status, Changes, Workspace, Build Now, Configure (which has a red arrow pointing to it), Delete Project, and Rename. The main area shows the job name "my-first-job" with a green checkmark. Below it is a section titled "Permalinks" listing four build logs. Under "Builds", there's a table with one entry: "#1 6:16AM". At the bottom right, it says "REST API Jenkins 2.528.3".

Step 20: Enable GitHub Webhook Trigger

- Scroll to **Build Triggers**
- Select **GitHub hook trigger for GITScm polling**
- Save configuration

Screenshot:



The screenshot shows the Jenkins configuration page for the "my-first-job" job. The left sidebar has "Triggers" selected, indicated by a red box and arrow. The main panel shows the "Triggers" section with a list of options: "Trigger builds remotely (e.g., from scripts)", "Build after other projects are built", "Build periodically", "GitHub hook trigger for GITScm polling" (which is checked), and "Poll SCM". Below this is the "Environment" section with options like "Delete workspace before build starts", "Use secret text(s) or file(s)", and "Add timestamps to the Console Output". At the bottom are "Save" and "Apply" buttons, with "Save" highlighted by a red arrow.

Step 21: Configuration Saved Successfully

Screenshot:

The screenshot shows the Jenkins interface for the job 'my-first-job'. The left sidebar contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The main content area displays the build history with four entries: Last build (#1), Last stable build (#1), Last successful build (#1), and Last completed build (#1), all from 1 day 3 hours ago. A 'Builds' section shows one build (#1) from December 25, 2025, at 6:16AM. At the bottom right, there are links for REST API and Jenkins 2.528.3.

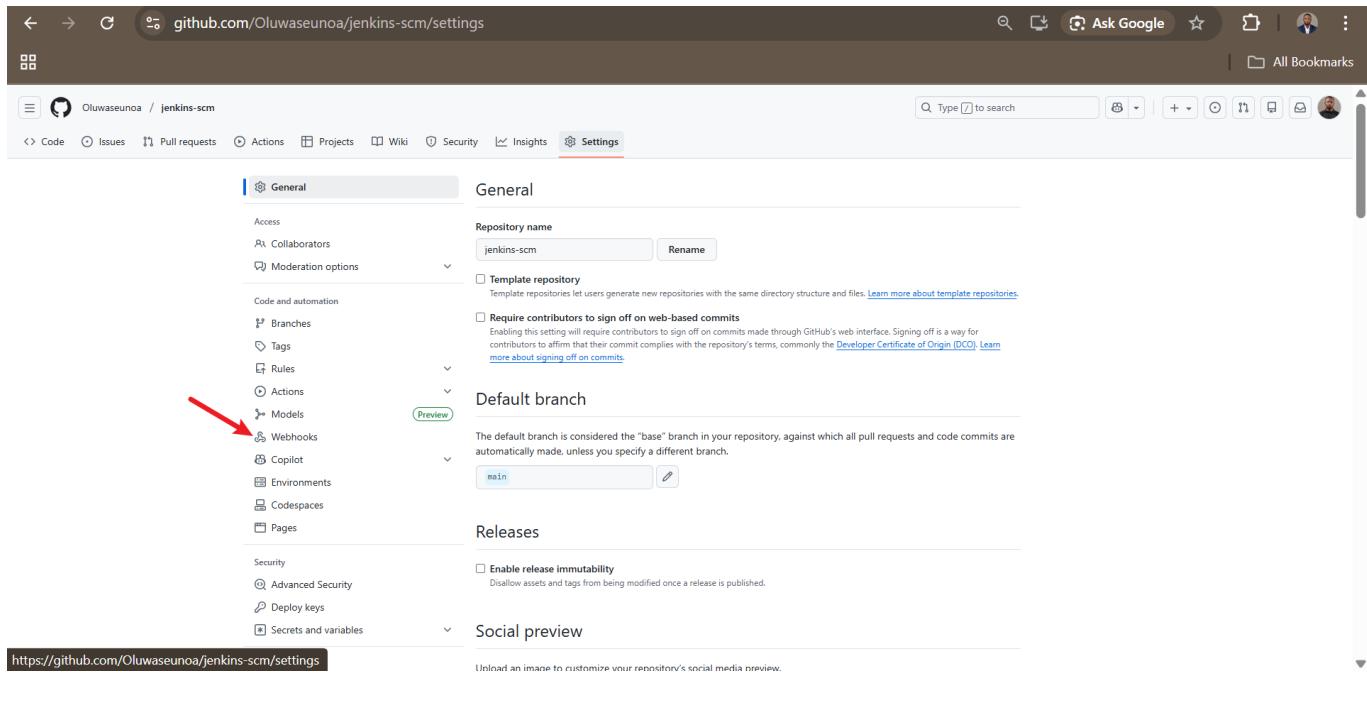
Step 22: Open GitHub Repository Settings

Screenshot:

The screenshot shows the GitHub repository page for 'jenkins-scm'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. An arrow points to the 'Settings' link. The main content area shows the repository details: 'main' branch, 1 branch, 0 tags, 1 commit by 'Oluwaseunao' (yesterday), and a README.md file (initial commit yesterday). To the right, sections include 'About' (No description, website, or topics provided), 'About' (Readme, Activity, 0 stars, 0 forks), 'Releases' (No releases published, Create a new release), and 'Packages' (No packages published, Publish your first package). The footer contains copyright information for GitHub, Inc. and links for Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and Do not share my personal information.

Step 23: Navigate to Webhooks

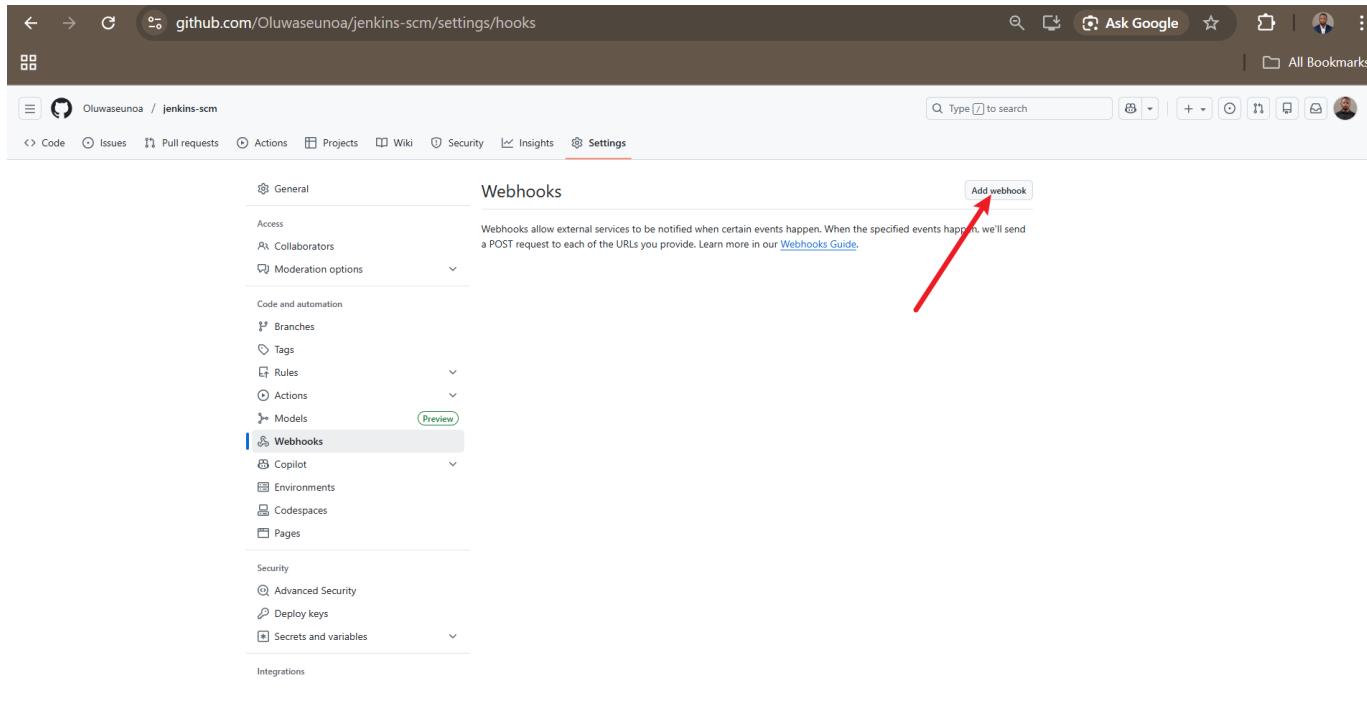
Screenshot:



The screenshot shows the GitHub repository settings page for 'jenkins-scm'. The left sidebar has 'Webhooks' selected under 'Code and automation'. The main area is titled 'General' and contains sections for 'Repository name' (set to 'jenkins-scm'), 'Template repository', 'Require contributors to sign off on web-based commits', 'Default branch' (set to 'main'), 'Releases', 'Social preview', and 'Integrations'. A red arrow points to the 'Webhooks' link in the sidebar.

Step 24: Add New Webhook

Screenshot:

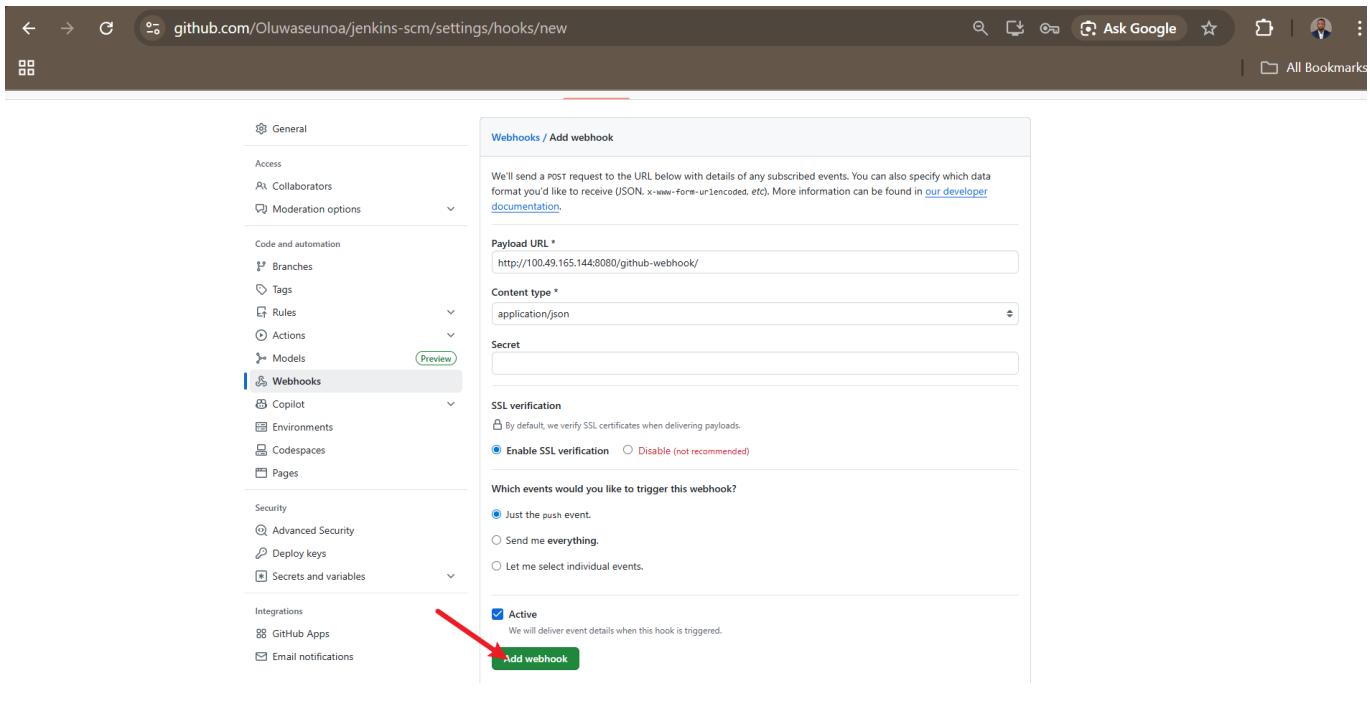


The screenshot shows the GitHub repository settings page for 'jenkins-scm'. The left sidebar has 'Webhooks' selected under 'Code and automation'. The main area is titled 'Webhooks' and contains a description of what webhooks do. A red arrow points to the 'Add webhook' button at the top right of the main area.

Step 25: Configure Webhook

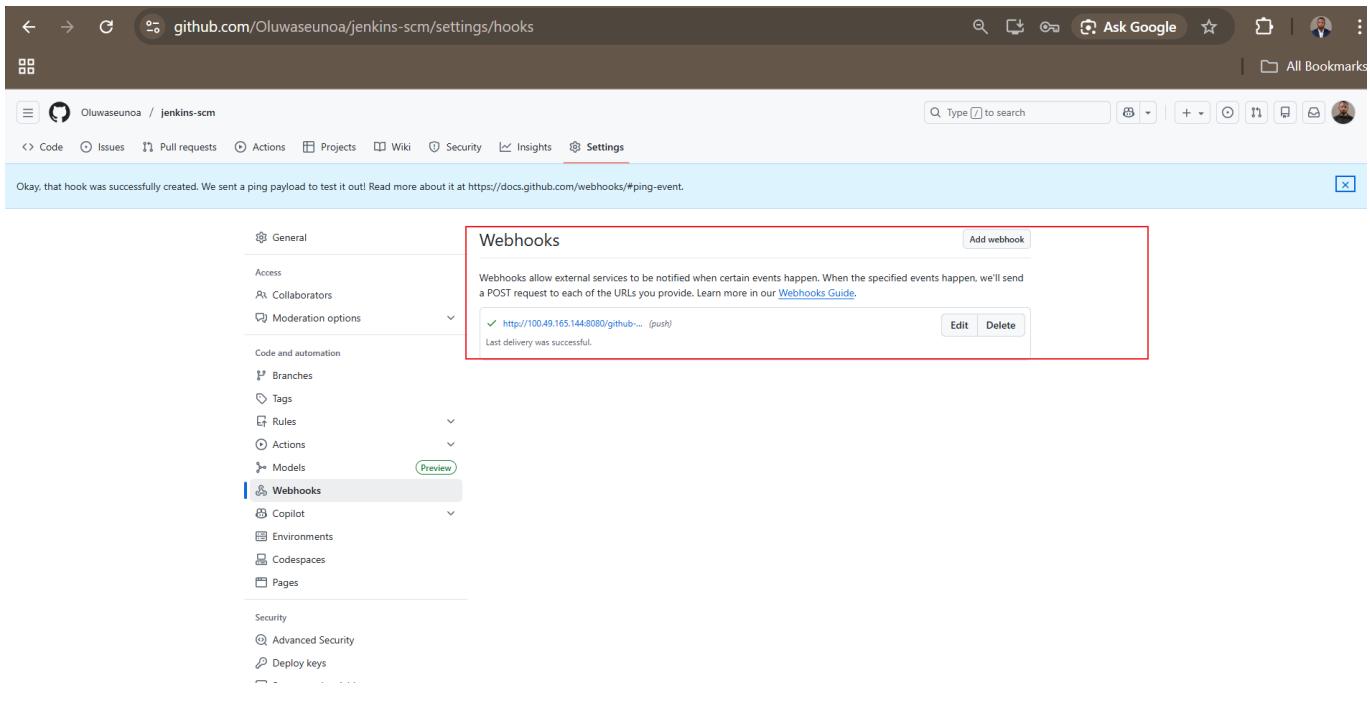
- Payload URL: <http://<jenkins-ip>:8080/github-webhook/>
- Content Type: `application/json`
- Event: **Just the push event**

Screenshot:



Step 26: Webhook Added Successfully

Screenshot:



Step 27: Clone GitHub Repository Locally

Screenshot:

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Automating Builds with Jenkins Freestyle Job and Github Webhooks (main)
$ git clone https://github.com/Oluwaseunoa/jenkins-scm.git
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Step 28: Open README File

Screenshot:

The screenshot shows a terminal window in the VS Code interface. The title bar says "DevOps-Projects". The terminal tab is selected, showing the command line history:

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Automating Builds with Jenkins Freestyle Job and Github Webhooks (main)
$ cd ./jenkins-scm/
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Automating Builds with Jenkins Freestyle Job and Github Webhooks/jenkins-scm (main)
$
```

The status bar at the bottom indicates "Ln 1, Col 1" and other settings like "Spaces: 4", "UTF-8", "CRLF", "Markdown", "Prettier", and a bell icon.

Step 29: Modify README Content

Screenshot:

The screenshot shows a terminal window in the VS Code interface. The title bar says "DevOps-Projects". The tab bar shows "README.md ...jenkins-scm X" and "README.md ...Getting Started with Jenkins with Github- Build X". The status bar at the bottom indicates "Ln 4, Col 110" and "Spaces: 4".

The terminal content is as follows:

```
1 # jenkins-scm
2
3
4 Hello, the purpose of this newly added line is to see if build trigger work when changes
are pushed to github
```

A red box highlights the last four lines of the terminal output.

Below the terminal, the status bar shows "Ln 4, Col 110" and "Spaces: 4". It also includes icons for Markdown, Prettier, and a bell.

Step 30: Commit and Push Changes

Screenshot:

The screenshot shows a terminal window in VS Code displaying a GitHub commit history and a README file. The terminal output shows a commit from 'Oluwaseunoa' adding a new line to the README file. The README file content is as follows:

```

1  # jenkins-scm
2
3
4 Hello, the purpose of this newly added line is to see if build trigger work when changes
are pushed to github

```

The terminal also shows the commit message and the resulting object statistics.

Step 31: Verify Changes on GitHub

Screenshot:

The screenshot shows a GitHub repository page for 'jenkins-scm'. The README file content is displayed as follows:

```

Hello, the purpose of this newly added line is to see if build trigger work when changes are pushed to github

```

A red box highlights the last line of the README file content.

Step 32: Observe Automatic Build Trigger

Jenkins automatically triggers a new build without manual action.

Screenshot:

The screenshot shows the Jenkins interface for the 'my-first-job'. The left sidebar has links for Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The main area shows a 'Builds' table with two entries: '#2 10:19 AM' (highlighted with a red box) and '#1 6:16 AM'. The 'Permalinks' section lists four recent builds. At the bottom right, it says 'REST API Jenkins 2.528.3'.

Step 33: View Console Output for Webhook Build

Screenshot:

The screenshot shows the Jenkins interface for the 'my-first-job' build #2. The left sidebar has links for Status, Changes, Console Output (highlighted with a red box), Edit Build Information, Delete build '#2', Polling Log, Timings, Git Build Data, and Previous Build. The main area shows the 'Console Output' page with the following log:

```

Started by GitHub push by Oluwaseunoa
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/my-first-job
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/my-first-job/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Oluwaseunoa/jenkins-scm.git # timeout=10
Fetching upstream changes from https://github.com/Oluwaseunoa/jenkins-scm.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/Oluwaseunoa/jenkins-scm.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 61b9755855c6e43ba6bafe0e3cea3d0347db4bd65 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 61b9755855c6e43ba6bafe0e3cea3d0347db4bd65 # timeout=10
Commit message: "Added a new line to README"
> git rev-list --no-walk 5187a46d647c5012fb916019c374c8226a9855f2 # timeout=10
Finished: SUCCESS

```

At the bottom right, it says 'REST API Jenkins 2.528.3'.

Project Outcome

- Jenkins Freestyle Job successfully configured
- GitHub repository integrated via SCM
- Manual and automatic builds validated

- GitHub Webhook successfully triggers Jenkins builds
 - CI workflow fully operational
-

Key Learnings

- Jenkins Freestyle job creation and configuration
 - GitHub SCM integration
 - Webhook-based automation
 - Core CI/CD principles
 - Build monitoring and log analysis
-

Future Improvements

- Add build steps (shell scripts or Maven/Node builds)
 - Integrate automated testing
 - Implement Jenkins Pipelines (Jenkinsfile)
 - Add notifications (Slack / Email)
 - Secure Jenkins with credentials management
-

Author

Seun DevOps / Cloud Engineering Project