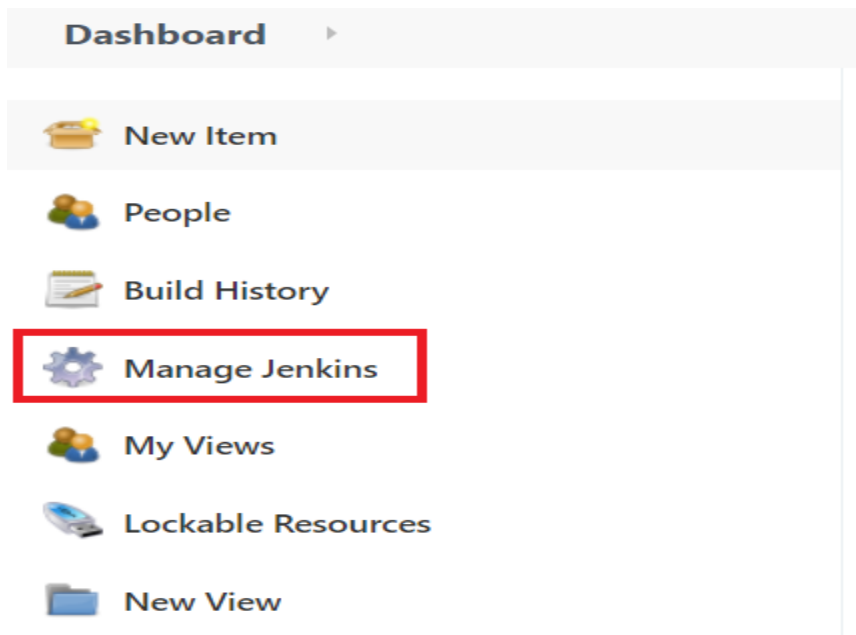


## How to Configure Poll SCM in Jenkins

This document explains how to integrate Jenkins with Git and use Poll SCM to trigger continuous downloads. So, let's see what are the steps involved in triggering builds using Poll SCM in Jenkins, once you have launched Jenkins in your Ubuntu instance on port no 8080.

1. In the Jenkins dashboard, click on the **Manage Jenkins** option.



2. Then click on **Manage Plugins**.

### System Configuration



#### Configure System

Configure global settings and paths.



#### Global Tool Configuration

Configure tools, their locations and automatic installers.



#### Manage Nodes and Clouds

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



#### Manage Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

🔴 There are updates available


3. Go into the **Installed tab** and check if the Git and GitHub plugins are installed or not. If the plugins are not already installed, then install them.

<input checked="" type="checkbox"/>	<b>Docker API Plugin</b> This plugin provides <a href="#">docker-java</a> API for other plugins.	3.1.5.2	<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<b>Git</b> This plugin integrates <a href="#">Git</a> with Jenkins.	4.6.0	<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<b>GitHub API Plugin</b> This plugin provides <a href="#">GitHub API</a> for other plugins.	1.123	<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<b>GitHub Branch Source Plugin</b> Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.10.2	<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<b>GitHub plugin</b> This plugin integrates <a href="#">GitHub</a> to Jenkins.	1.33.1	<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<b>Jackson 2 API Plugin</b> This plugin exposes the Jackson 2 JSON APIs to other Jenkins plugins.	2.12.1	<a href="#">Uninstall</a>

4. Then create a new **Freestyle project**. Here, we have created a project named GitTrigger. Click on the **OK** button to finish.


### Enter an item name

Required field




**Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.




**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**

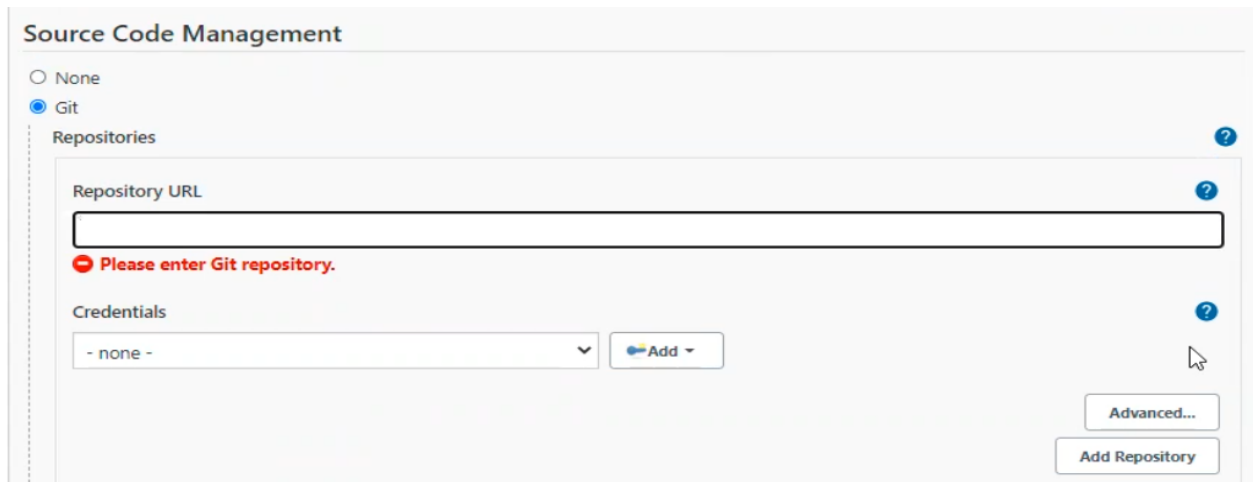
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



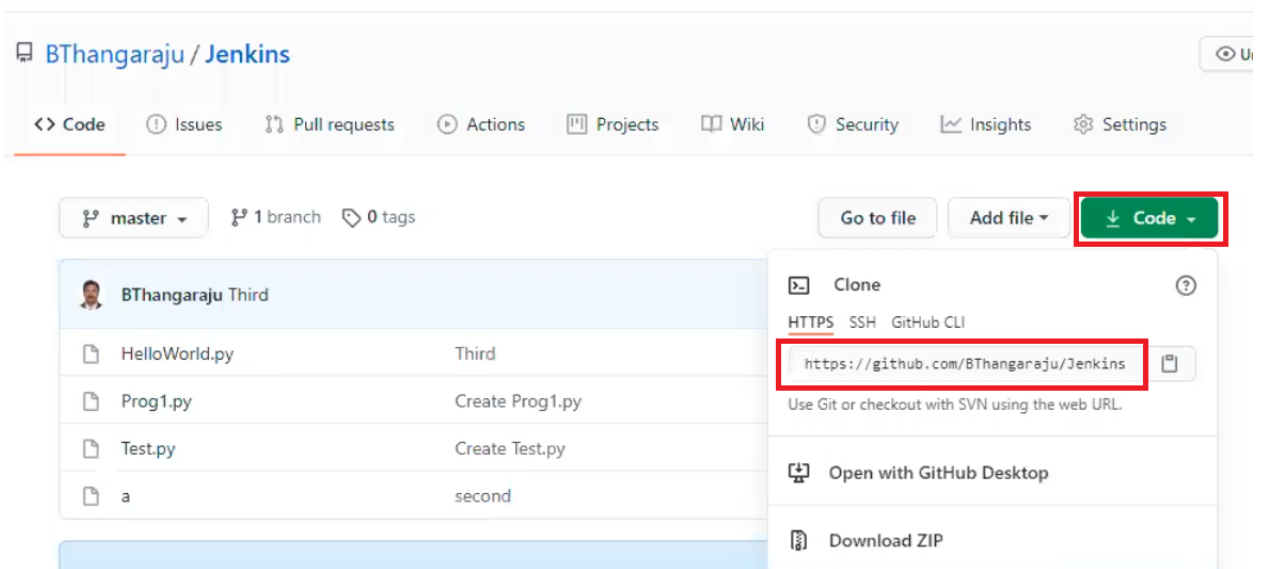
**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate name space, so you can have multiple things of the same name as long as they are in different folders.

5. Next, we need to configure the project.
6. Now scroll down and select **Git** in the **Source Code Management (SCM)**.



7. Here, you need to enter the URL of the repository.
8. Therefore, go to your repository and copy the repository URL, as shown in the image below.



9. Paste the copied URL into the **Repository URL**, as shown below, and leave the other options as it is (default setting).

**Source Code Management**

☐ None  
☒ Git

Repositories

Repository URL

⛔ Please enter Git repository.

Credentials

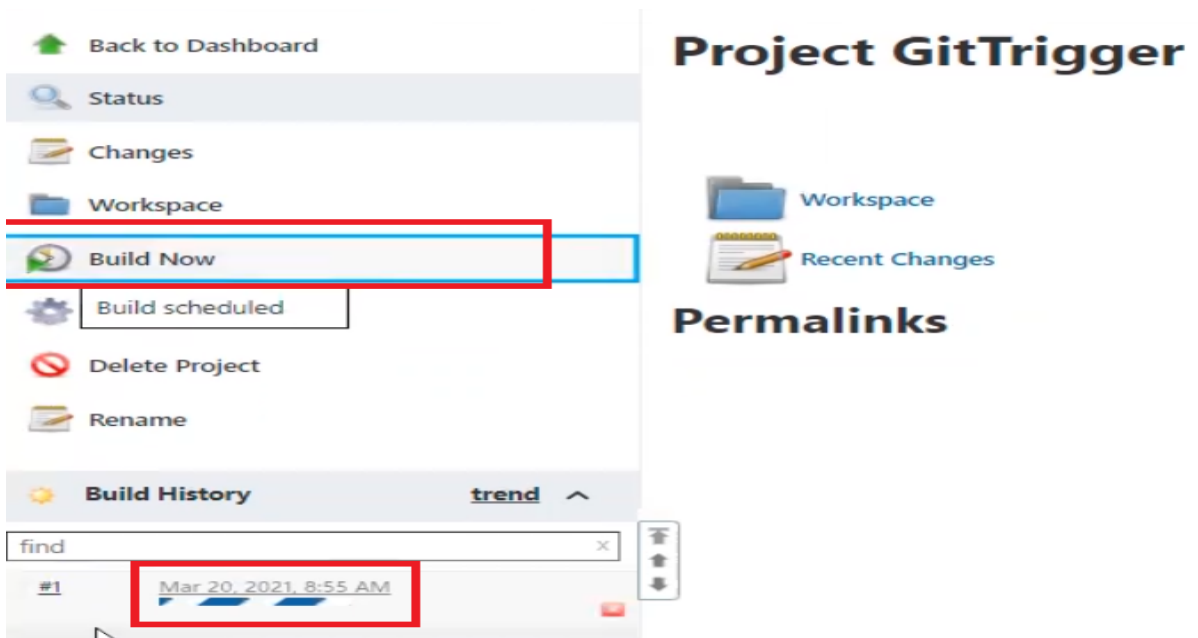
10. Now scroll down to the **Build** section and click on **Add build step**.  
 Select the **Execute shell** option from that list.

scans

11. After writing the necessary commands, click on **Apply** and then **Save** the command, as shown in the image below.



12. Now, click on **Build Now** in the dashboard. In the build history, you can see that the project is being built.



13. Once the project has been successfully built, you can view the **Console output** as shown in the image below.

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

```

Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/GitTrigger
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/BThangaraju/Jenkins.git
> git init /var/lib/jenkins/workspace/GitTrigger # timeout=10
Fetching upstream changes from https://github.com/BThangaraju/Jenkins.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/BThangaraju/Jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/BThangaraju/Jenkins.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/master^{commit} # timeout=10
Checking out Revision a54a864f579168424403f5e26d1010e98ed57170 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f a54a864f579168424403f5e26d1010e98ed57170 # timeout=10
Commit message: "Third"
First time build. Skipping changelog.
[GitTrigger] $ /bin/sh -xe /tmp/jenkins2204856867948303638.sh
+ ./HelloWorld.py
Hello World Demo for GitHub hook trigger for GITSCM polling...
Hello World Demo for GitHub hook trigger for GITSCM polling...

```

14. Next, we will configure Poll SCM to trigger this job automatically. Therefore select the **Back to Project** option on the dashboard.

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

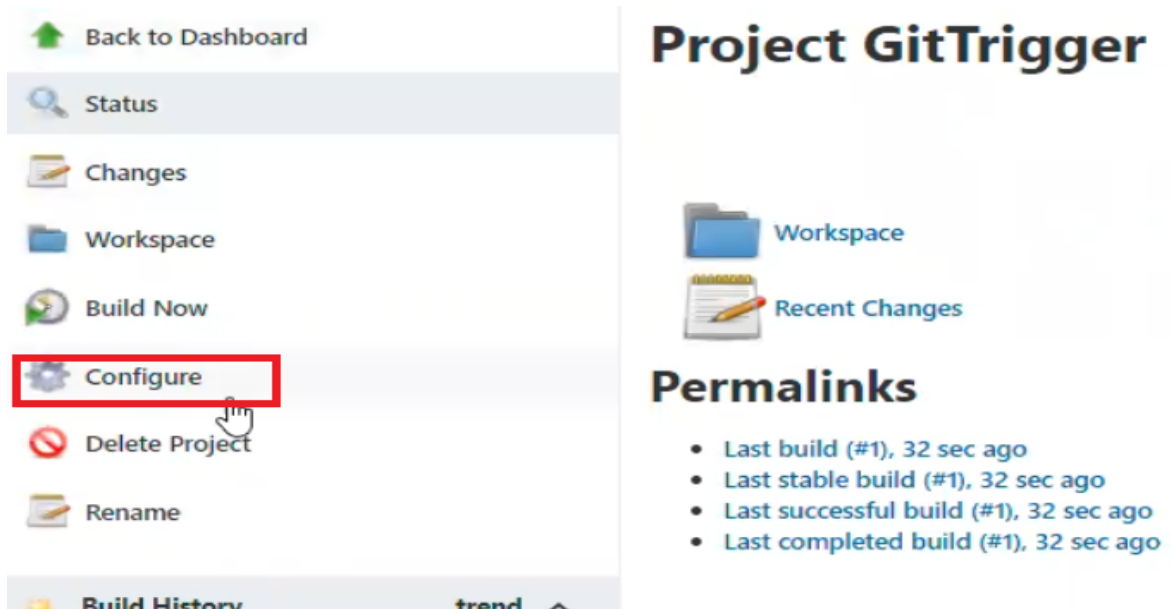
## Console Output

```

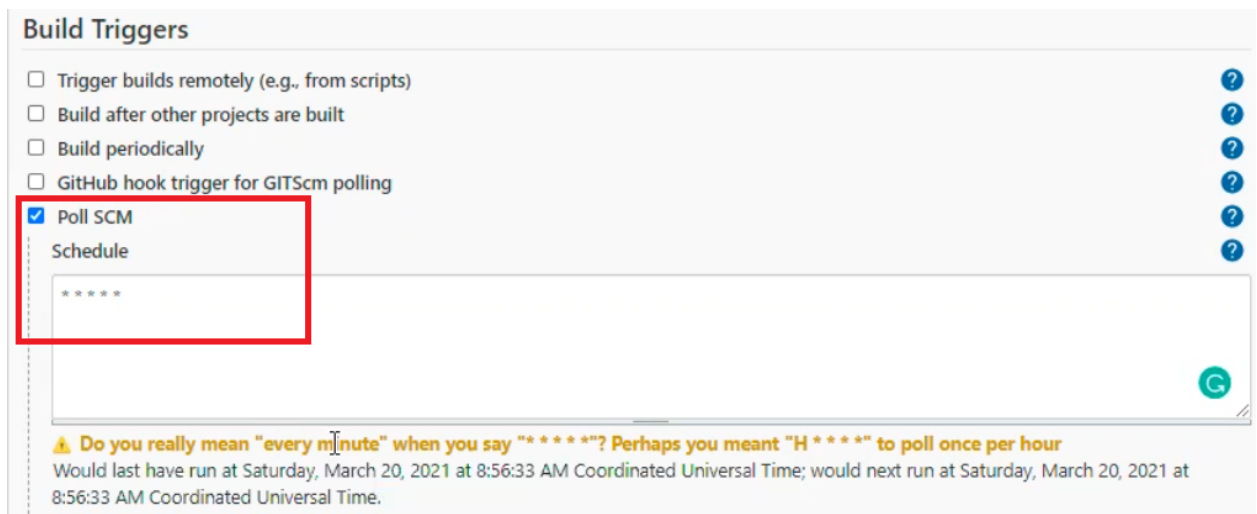
Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/GitTrigger
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/BThangaraju/Jenkins.git
> git init /var/lib/jenkins/workspace/GitTrigger # timeout=10
Fetching upstream changes from https://github.com/BThangaraju/Jenkins.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/BThangaraju/Jenkins.
> git config remote.origin.url https://github.com/BThangaraju/Jenkins.gi
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision a54a864f579168424403f5e26d1010e98ed57170 (refs/remo

```

15. Then click on the **Configure** option.



16. Scroll down and select the **Poll SCM** option in the **Build Triggers** section.
17. In the **Schedule** box, you need to specify the time interval using the cron schedule expression. Here, we have entered “\* \* \* \* \*”, which means to check for new changes every minute.



18. In order to trigger Poll SCM, we need to make a few changes. Therefore, we will go back to the repository.

master
Jenkins / HelloWorld.py
<> Jump to


BThangaraju Third

1 contributor

Executable File
5 lines (4 sloc)
219 Bytes

```

1  #! /usr/bin/python3
2  # This Python program will print Hello World...
3  print("Hello World Demo for GitHub hook trigger for GITScm polling...\n")
4  print("Hello World Demo for GitHub hook trigger for GITScm polling...\n")

```


19. Then edit the file.

```

1  #! /usr/bin/python3
2  # This Python program will print Hello World...
3  print("Hello World Demo for GitHub hook trigger for GITScm polling...\n")
4  print("Hello World Demo for GitHub hook trigger for GITScm polling...\n")
5  print("Hello GitTrigger Project\n");|
6

```

20. After making the necessary changes, commit the changes.



### Commit changes

Update HelloWorld.py

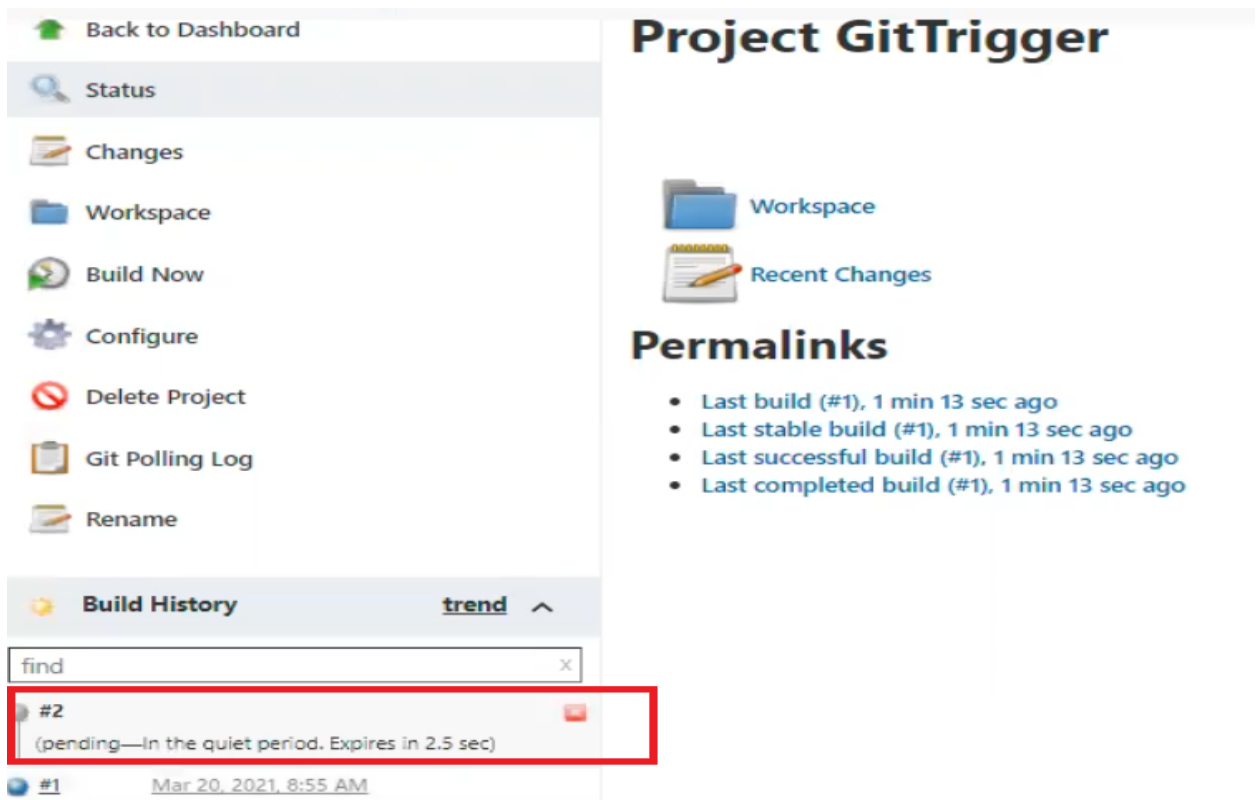
Add an optional extended description...

☒ Commit directly to the master branch.
 ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes
Cancel



21. Now return to the Jenkins dashboard. We can see that Jenkins will start to build the job in less than a minute, as shown in the image below.



22. Once the job is built, click on the 'build number' to see the **log output**.

