

## Lab Exercise 9

### Setting up Poll SCM Configuration in Jenkins

**Objective:** To set up poll SCM configuration in Jenkins for automating build triggering and optimizing resource usage on Jenkins servers

**Tools required:** Jenkins

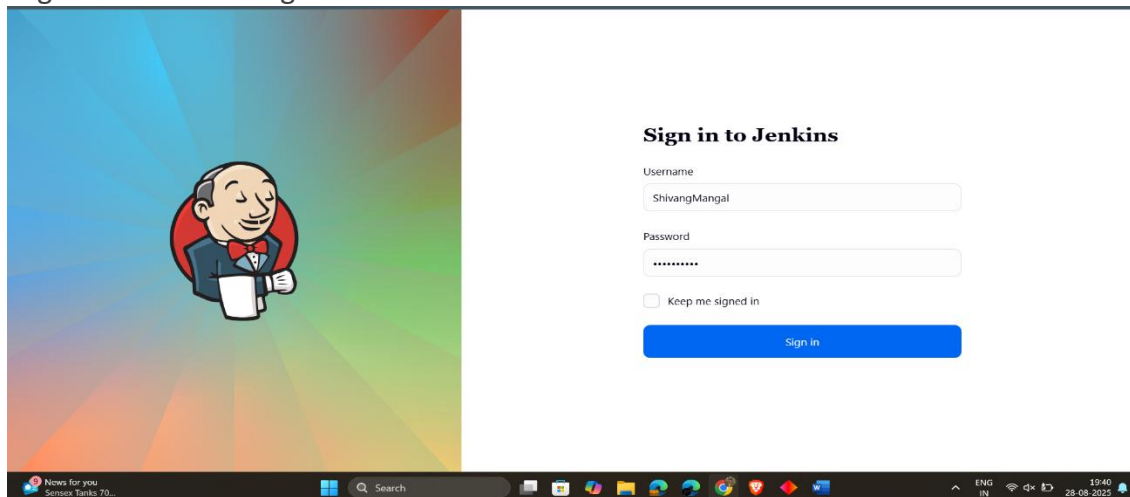
**Prerequisites:** You need to have a Jenkins up and running.

Steps to be followed:

1. Log in to the Jenkins CI tool and set up poll SCM configuration

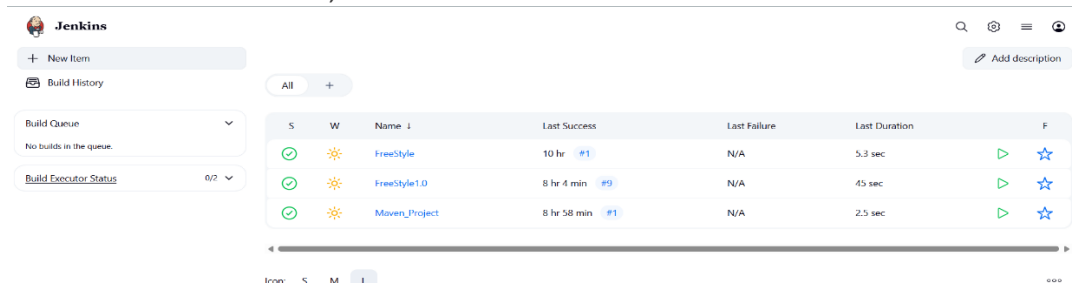
#### Step 1: Log in to the Jenkins CI tool and set up poll SCM configuration

##### 1.1 Log in to Jenkins using the credentials

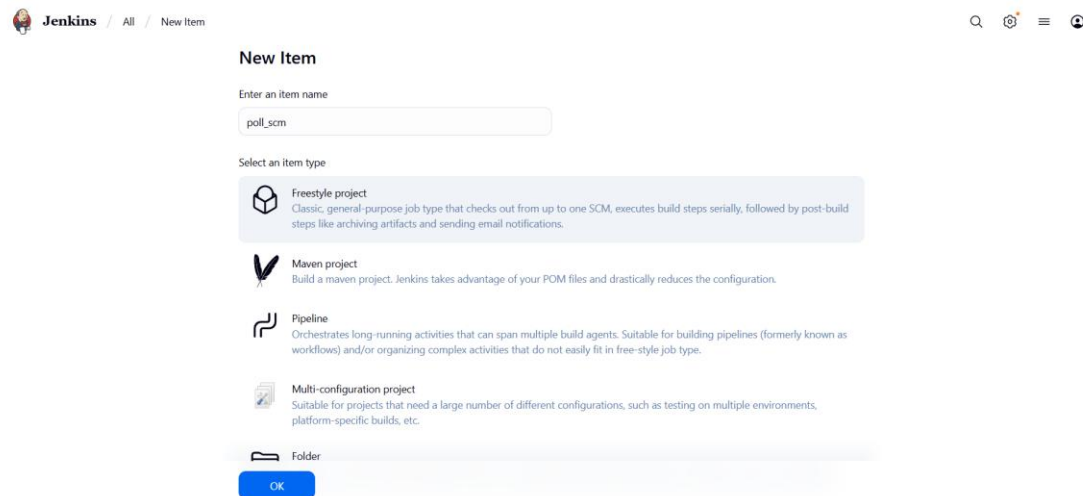


**Note:** The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

##### 1.2 In the Jenkins dashboard, click on **New Item**

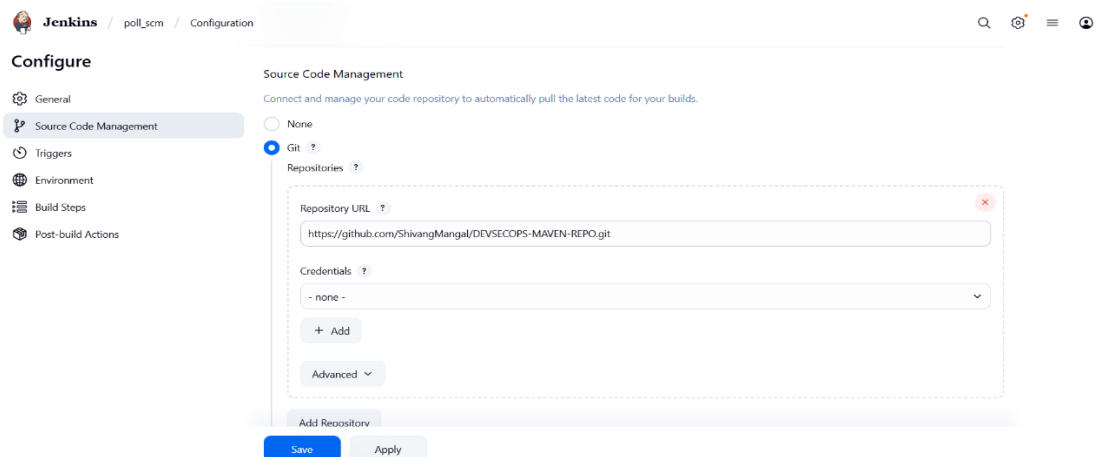


### 1.3 Select the **Freestyle project** while creating a Jenkins job, provide a custom job name, and click on **OK**



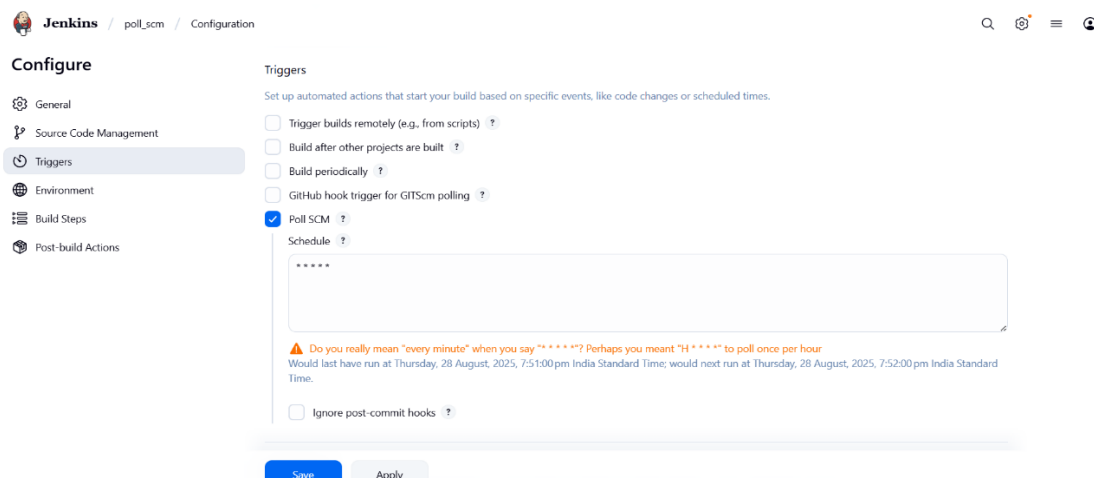
The screenshot shows the Jenkins 'New Item' page. At the top, there's a breadcrumb 'Jenkins / All / New Item'. On the right, there are icons for search, settings, menu, and user. The main heading is 'New Item'. Below it, there's a text input field 'Enter an item name' with the value 'poll\_scm'. Underneath, there's a section 'Select an item type' with four options: 'Freestyle project' (selected), 'Maven project', 'Pipeline', and 'Multi-configuration project'. Each option has a brief description. At the bottom, there's a 'Folder' option and an 'OK' button.

### 1.4 Now, in the Configure page, navigate to **Source Code Management** from the left navigation bar, select **Git**, and then provide the Git repository URL



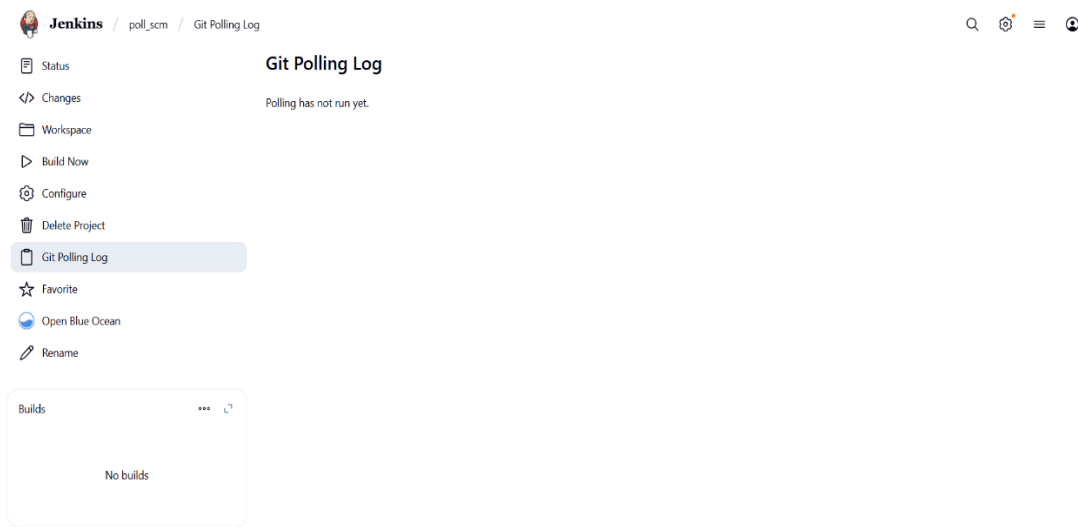
The screenshot shows the Jenkins 'Configure' page for the 'poll\_scm' job. The left navigation bar has 'Source Code Management' selected. The main section is 'Source Code Management' with a description: 'Connect and manage your code repository to automatically pull the latest code for your builds.' There are two radio buttons: 'None' and 'Git' (selected). Below 'Git', there's a 'Repositories' section with a 'Repository URL' field containing 'https://github.com/ShivangMangal/DEVSECOPS-MAVEN-REPO.git'. There's also a 'Credentials' dropdown menu set to 'none'. At the bottom, there are 'Add Repository', 'Save', and 'Apply' buttons.

### 1.5 Now, navigate to **Build Environment**, click on **Poll SCM**, provide **H/2\*\*\*\*** under the **Schedule** section, and then click on **Save**

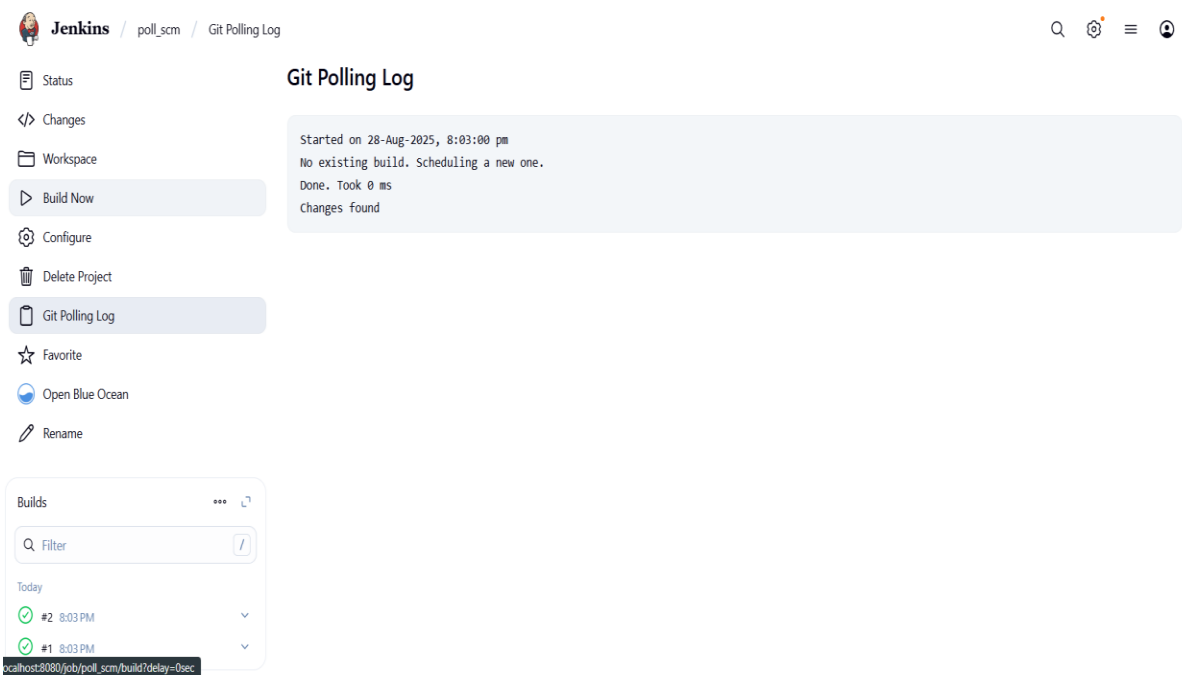


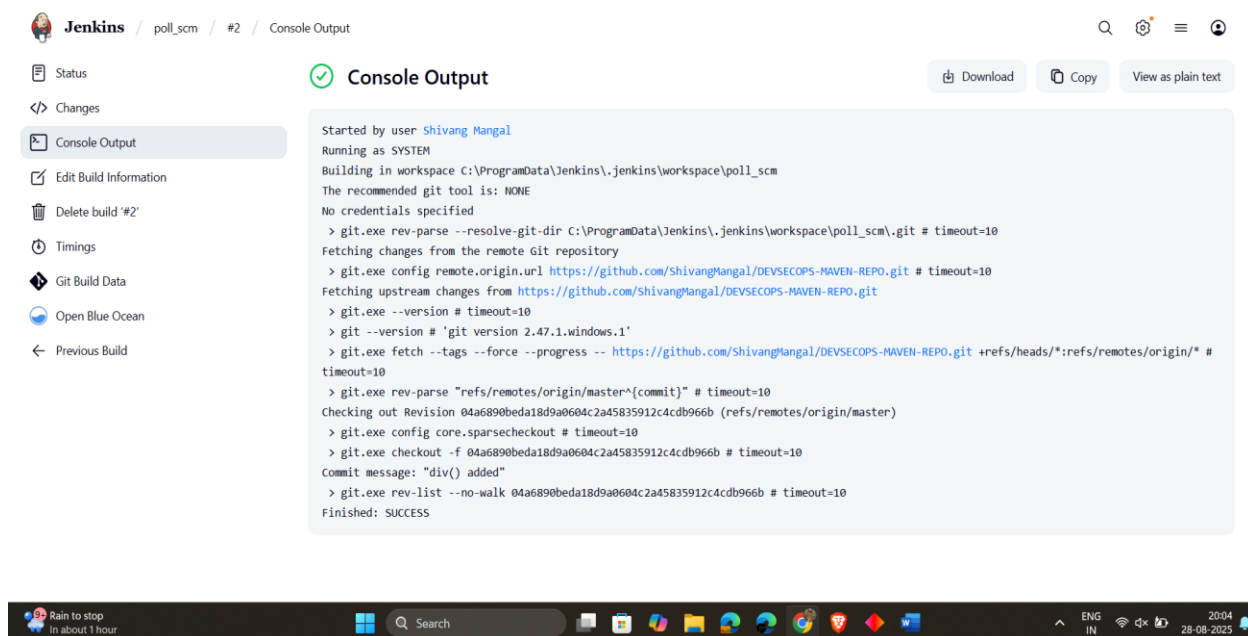
The screenshot shows the Jenkins 'Configure' page for the 'poll\_scm' job, specifically the 'Triggers' section. The left navigation bar has 'Triggers' selected. The main section is 'Triggers' with a description: 'Set up automated actions that start your build based on specific events, like code changes or scheduled times.' There are four checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', and 'Poll SCM' (selected). Below 'Poll SCM', there's a 'Schedule' field containing 'H/2\*\*\*\*'. There's a warning message: 'Do you really mean "every minute" when you say "H/2\*\*\*\*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour. Would last have run at Thursday, 28 August, 2025, 7:51:00pm India Standard Time; would next run at Thursday, 28 August, 2025, 7:52:00pm India Standard Time.' At the bottom, there are 'Save' and 'Apply' buttons.

1.6 You will be directed to the poll scm project after clicking on Save. Now, click on **Git Polling Log** from the left navigation bar.



1.7 Then, click on **Build Now** to configure a build, and you will be able to view the successful build in the **Console output** section.





The screenshot shows the Jenkins web interface for a build named 'poll\_scm' with ID '#2'. The 'Console Output' tab is selected, displaying a series of Git commands and their successful execution. The output indicates that the build is running as 'SYSTEM' in the workspace 'C:\ProgramData\Jenkins\jenkins\workspace\poll\_scm'. It shows the process of fetching changes from a remote repository, configuring the remote origin URL, and checking out the latest revision. The build concludes with a commit message 'div() added' and a successful status.

**Jenkins** / poll\_scm / #2 / Console Output

Search, Settings, Menu, User icons

Download, Copy, View as plain text

**Console Output**

```
Started by user Shivang Mangal
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\poll_scm
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\poll_scm\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/ShivangMangal/DEVSECOPS-MAVEN-REPO.git # timeout=10
Fetching upstream changes from https://github.com/ShivangMangal/DEVSECOPS-MAVEN-REPO.git
> git.exe --version # timeout=10
> git --version # 'git version 2.47.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/ShivangMangal/DEVSECOPS-MAVEN-REPO.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 04a6890beda18d9a0604c2a45835912c4cdb966b (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 04a6890beda18d9a0604c2a45835912c4cdb966b # timeout=10
Commit message: "div() added"
> git.exe rev-list --no-walk 04a6890beda18d9a0604c2a45835912c4cdb966b # timeout=10
Finished: SUCCESS
```

Rain to stop In about 1 hour

Search

ENG IN 20:04 28-08-2025

You can see that the build is configured successfully.

By following these steps, you have successfully set up poll SCM configuration in Jenkins for automating build triggering and optimizing resource usage on Jenkins servers.