

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. Log in to Jenkins using the credentials



Sign in to Jenkins

Username

mohdanas

Password

.....

☐ Keep me signed in

Sign in

2. In the Jenkins dashboard, click on **New Item**
3. Select the **Freestyle project** while creating a Jenkins job, provide a custom job name, and click on **OK**
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 Configuration page for a job named 'POLL_SCM'. The left sidebar contains a 'Configure' section with a list of tabs: General, Source Code Management (selected), Triggers, Environment, Build Steps, and Post-build Actions. The main content area is titled 'Source Code Management' and includes a description: 'Connect and manage your code repository to automatically pull the latest code for your builds.' Below this, there are radio buttons for 'None' and 'Git' (selected). A 'Repositories' section is shown with a 'Repository URL' field containing 'https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git', a 'Credentials' dropdown menu set to '- none -', and an 'Add' button. An 'Advanced' dropdown menu is also visible. At the bottom of the configuration area, there is an 'Add Repository' button and a 'Branches to build' section. The page concludes with 'Save' and 'Apply' buttons.

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

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule ?


* * * * *

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * * *" to poll once per hour

Would last have run at Wednesday, August 27, 2025, 12:24:00 PM India Standard Time; would next run at Wednesday, August 27, 2025, 12:25:00 PM India Standard Time.

☐ Ignore post-commit hooks ?

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

 **Jenkins** / POLL_SCM / #1 / Console Output

Status

</> Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Git Build Data

✓ Console Output

Download

Copy

View as plain text

```
Started by user Mohd Anas
Running as SYSTEM
Building in workspace /Users/mohdanas/.jenkins/workspace/POLL_SCM
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git
> git init /Users/mohdanas/.jenkins/workspace/POLL_SCM # timeout=10
Fetching upstream changes from https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git
> git --version # timeout=10
> git --version # 'git version 2.39.5 (Apple Git-154)'
> git fetch --tags --force --progress -- https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.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 04a6890beda18d9a0604c2a45835912c4cdb966b (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 04a6890beda18d9a0604c2a45835912c4cdb966b # timeout=10
Commit message: "div() added"
First time build. Skipping changelog.
Finished: SUCCESS
```

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.