

# Lab Exercise 9

## Setting up Poll SCM Configuration in Jenkins

Name: Viraj Bhidola

Sap id: 500121825

Enrolment no.: R2142231731

Batch 2 DevOps

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

**Tools required:** Jenkins

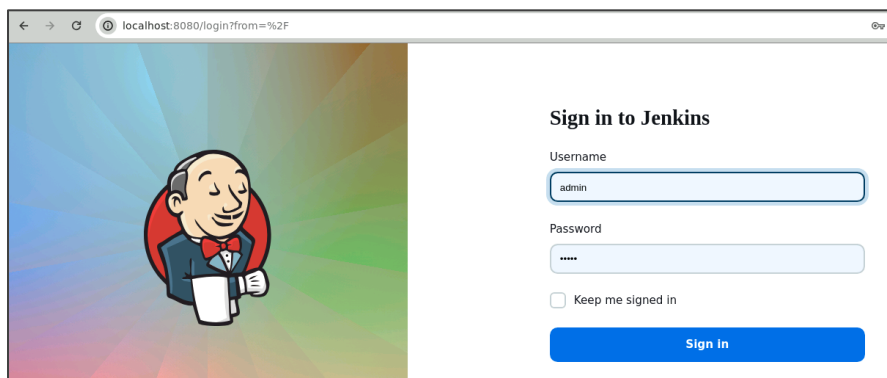
**Prerequisites:** You need to have a Jenkins up and running.

Steps to be followed:

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

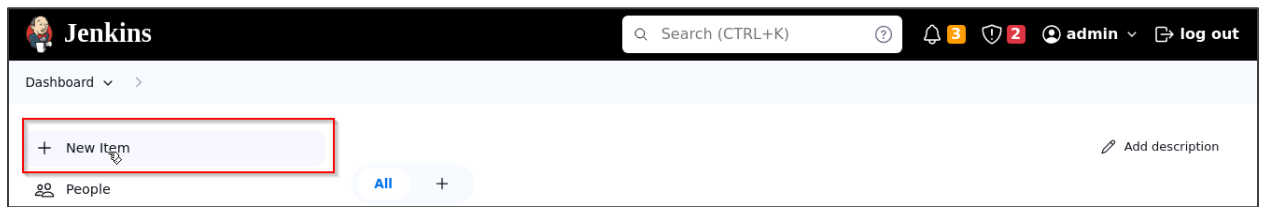
### Step 1: Log in to the Jenkins CI tool and set up poll SCM configuration

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

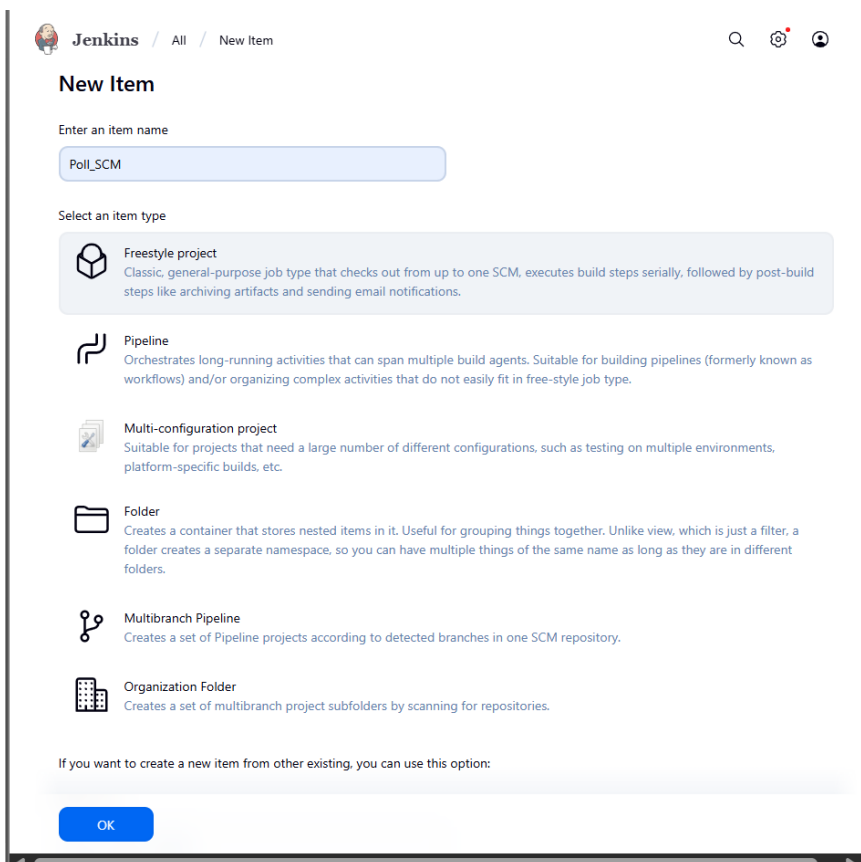


**Note:** The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

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**



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

**Source Code Management**

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

Repository URL ? ✕

`https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git`

Credentials ?

- none - ▼

+ Add

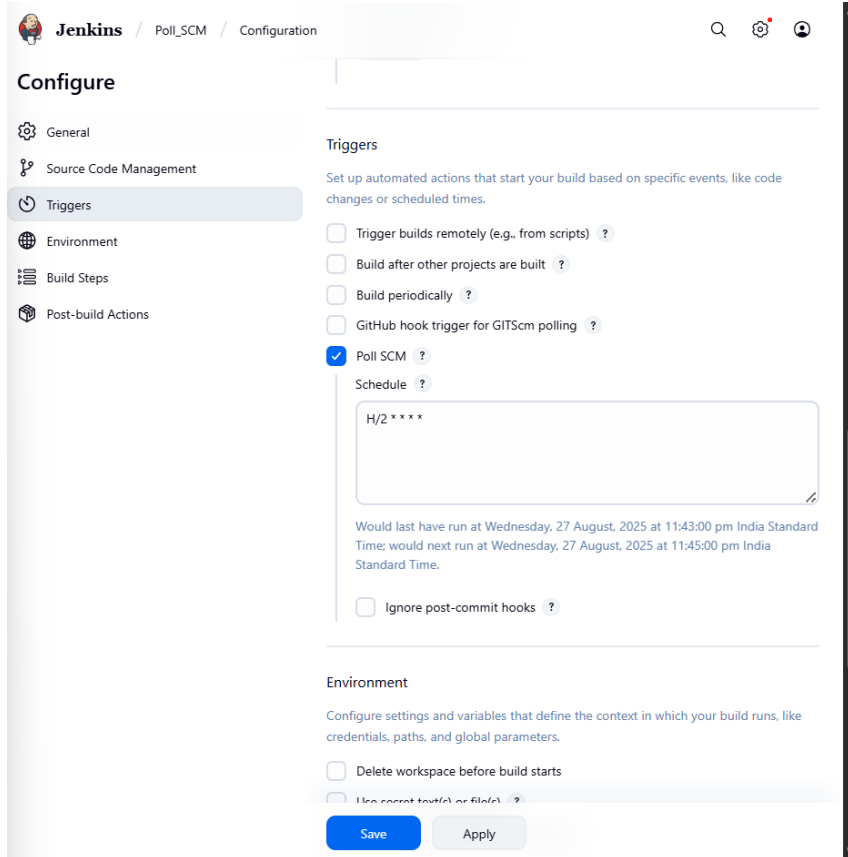
Advanced ▼

Add Repository

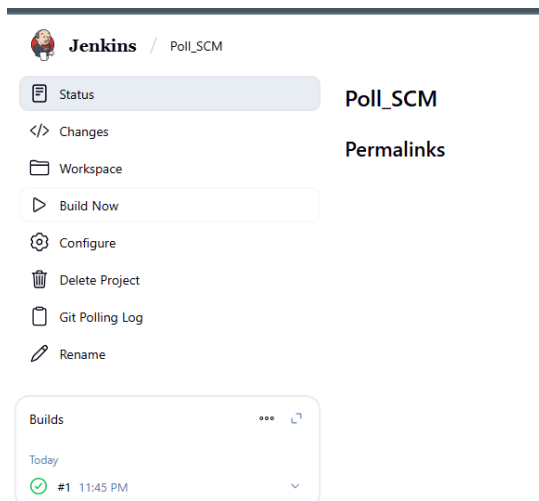
Branches to build ?


**Note:** Here, the repository URL is **`https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git`**

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



1.6 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

Download

Copy

View as plain text

Started by user [Viraj Bhidola](#)

Running as SYSTEM

Building in workspace

C:\ProgramData\Jenkins\.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.exe init

C:\ProgramData\Jenkins\.jenkins\workspace\Poll\_SCM # timeout=10

Fetching upstream changes from

<https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git>

> git.exe --version # timeout=10

> git --version # 'git version 2.47.1.windows.2'

> git.exe fetch --tags --force --progress --

<https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git>

+refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git.exe config remote.origin.url

<https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git> #

timeout=10

> git.exe config --add remote.origin.fetch

+refs/heads/\*:refs/remotes/origin/\* # timeout=10

Avoid second fetch

> 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"

localhost:8080

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.