

Name – Rachit Shrivastava

Sap Id – 500119571

Batch 3 – DevOps

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

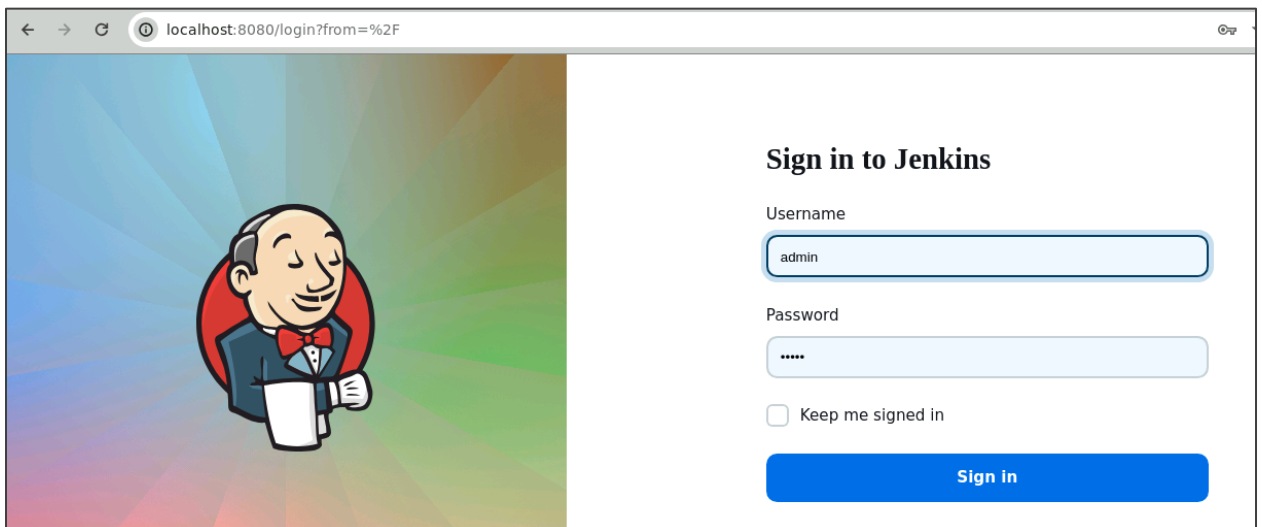
Prerequisites: You need to have a Jenkins up and running.

Steps to be followed:

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

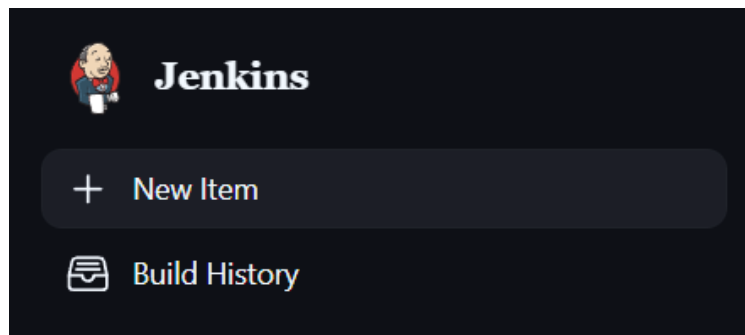
Step 1: Log in to the Jenkins CI tool and set up poll SCM configuration

1.1 Log in to Jenkins using the credentials

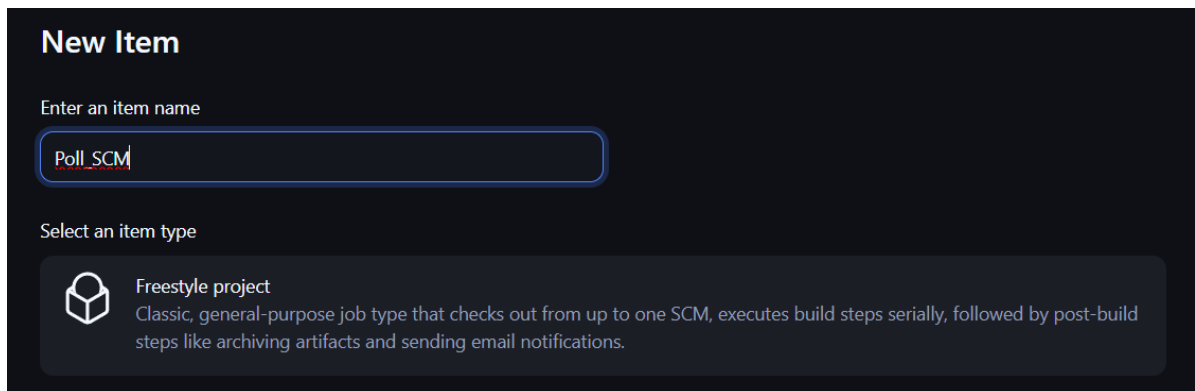
A screenshot of a web browser window showing the Jenkins login page. The browser's address bar displays 'localhost:8080/login?from=%2F'. The page features a large, colorful background image of the Jenkins mascot, a cartoon man in a tuxedo, holding a white cup. To the right of the mascot, the text 'Sign in to Jenkins' is displayed. Below this, there are two input fields: 'Username' with the value 'admin' and 'Password' with masked characters '.....'. A checkbox labeled 'Keep me signed in' is positioned below the password field. At the bottom right, there is a blue button labeled 'Sign in'.

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**

A screenshot of the 'New Item' configuration page in Jenkins. The page has a dark blue header with the title 'New Item'. Below the header, there is a section 'Enter an item name' with a text input field containing 'Poll_SCM'. Below this is a section 'Select an item type' with a list of options. The first option is 'Freestyle project', which is highlighted. To the left of the text is a small icon of a cube. To the right of the text is a description: 'Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.'

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 ▾

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**

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 ?

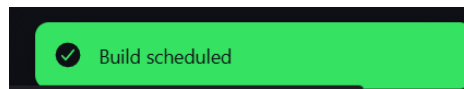
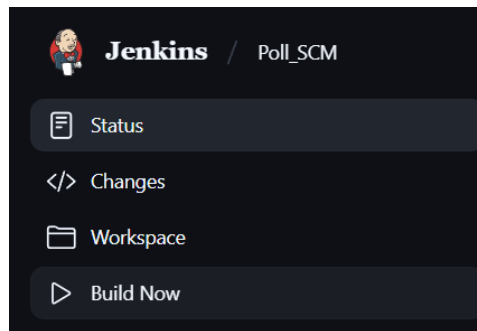
Schedule ?

`H/2 ****`

Would last have run at Thursday, 28 August, 2025 at 6:53:00 pm India Standard Time; would next run at Thursday, 28 August, 2025 at 6:55:00 pm India Standard Time.

☐ Ignore post-commit hooks ?

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.



```
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"
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.