

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

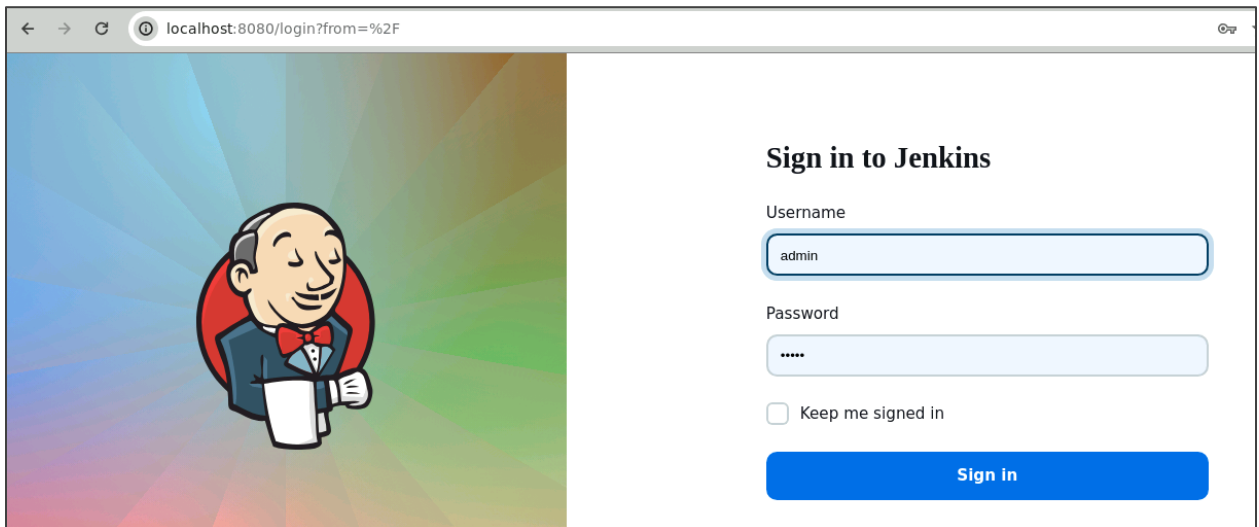
Prerequisites: You need to have a Jenkins up and running.

Steps to be followed:

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

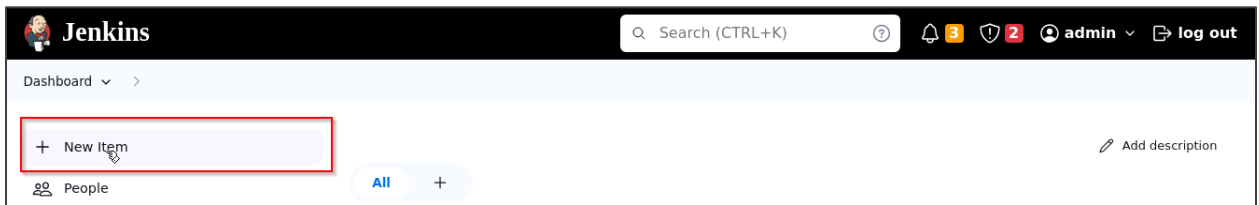
Step 1: Log in to the Jenkins CI tool and set up poll SCM configuration

1.1 Log in to Jenkins using the credentials

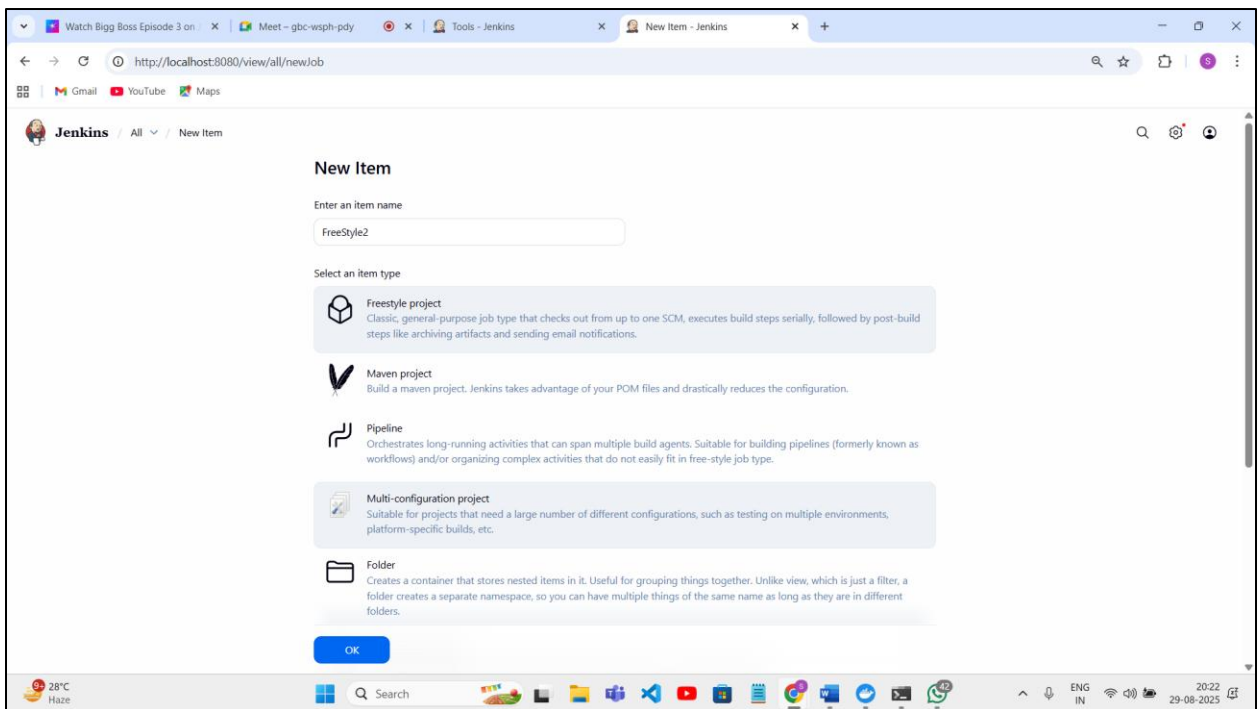
A screenshot of a web browser window showing the Jenkins login page. The browser's address bar displays 'localhost:8080/login?from=%2F'. The page features a large illustration of the Jenkins mascot, a blue elephant wearing a red bow tie and holding a white document, set against a colorful, radial background. To the right of the illustration, the text 'Sign in to Jenkins' is displayed. Below this, there are two input fields: 'Username' with the text 'admin' entered, and 'Password' with masked characters '.....'. A checkbox labeled 'Keep me signed in' is present below the password field. At the bottom right, there is a prominent blue button labeled 'Sign in'.

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

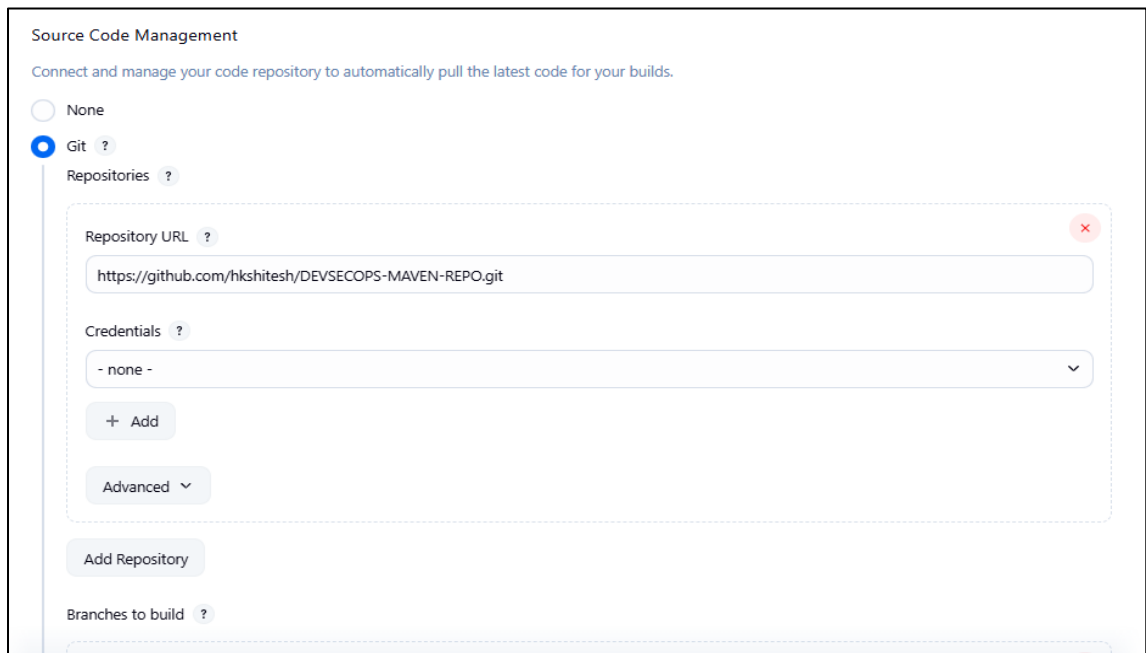
1.2 In the Jenkins dashboard, click on **New Item**



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



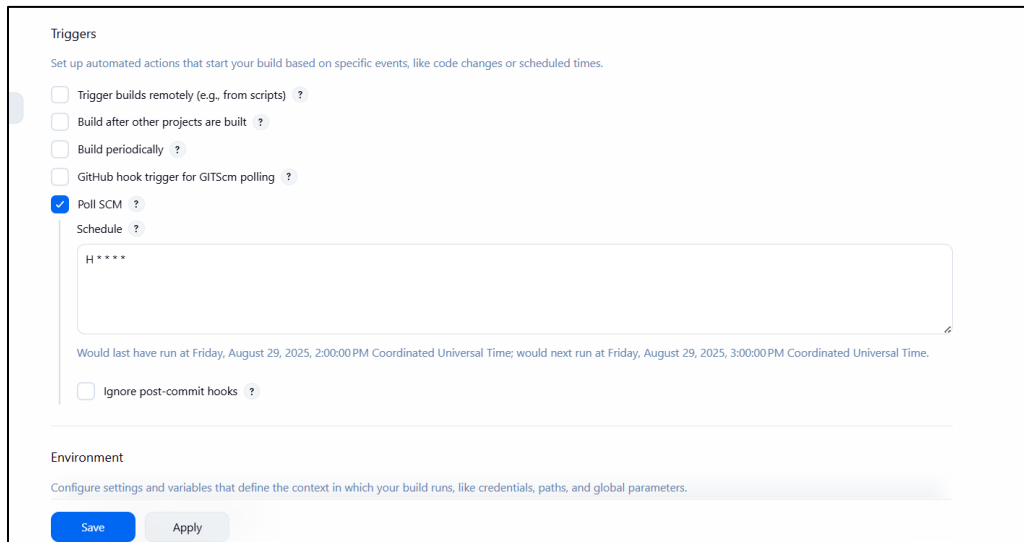
1.4 Now, in the Configure page, navigate to **Source Code Management** from the left navigation bar, select **Git**, and then provide the Git repository URL



The screenshot shows the 'Source Code Management' configuration page. At the top, it says 'Connect and manage your code repository to automatically pull the latest code for your builds.' Below this, there are two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Under the 'Git' section, there is a 'Repositories' heading. A dashed box contains a 'Repository URL' field with the value 'https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git', a 'Credentials' dropdown menu set to '- none -', an '+ Add' button, and an 'Advanced' dropdown menu. Below the dashed box is an 'Add Repository' button. At the bottom, there is a 'Branches to build' section.

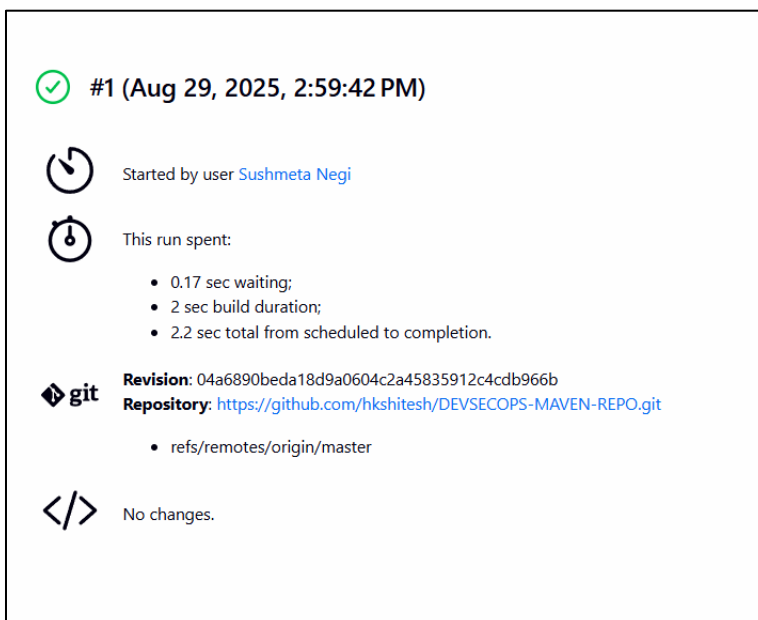
Note: Here, the repository URL is <https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git>

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



The screenshot shows the 'Triggers' configuration page. It starts with the heading 'Triggers' and a sub-heading 'Set up automated actions that start your build based on specific events, like code changes or scheduled times.' Below this, there are five checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. The 'Poll SCM' checkbox is checked. Below the 'Poll SCM' checkbox, there is a 'Schedule' section with a text input field containing 'H/2****'. Below the input field, there is a note: 'Would last have run at Friday, August 29, 2025, 2:00:00 PM Coordinated Universal Time; would next run at Friday, August 29, 2025, 3:00:00 PM Coordinated Universal Time.' At the bottom, there is an 'Environment' section with a sub-heading 'Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.' Below the 'Environment' section, there are two buttons: 'Save' and 'Apply'.

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.



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.