.

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

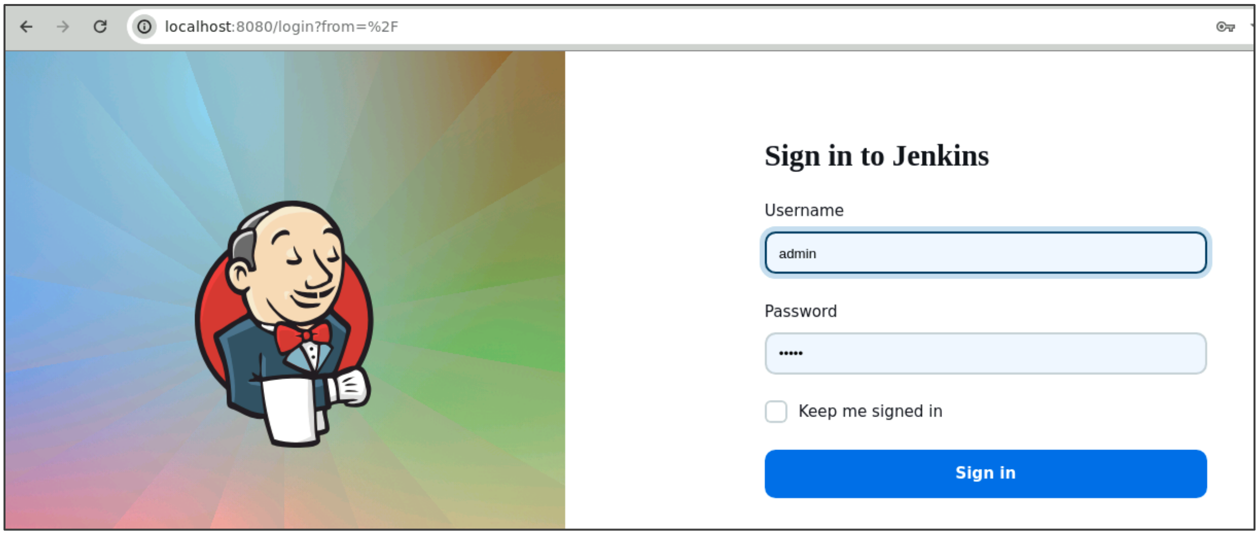
Prerequisites: You need to have a Jenkins up and running.

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



localhost:8080/login?from=%2F

Sign in to Jenkins

Username: admin

Password: •••••

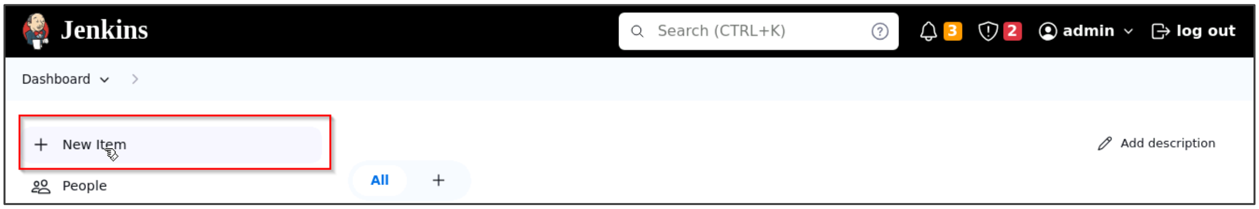
☐ Keep me signed in

Sign in

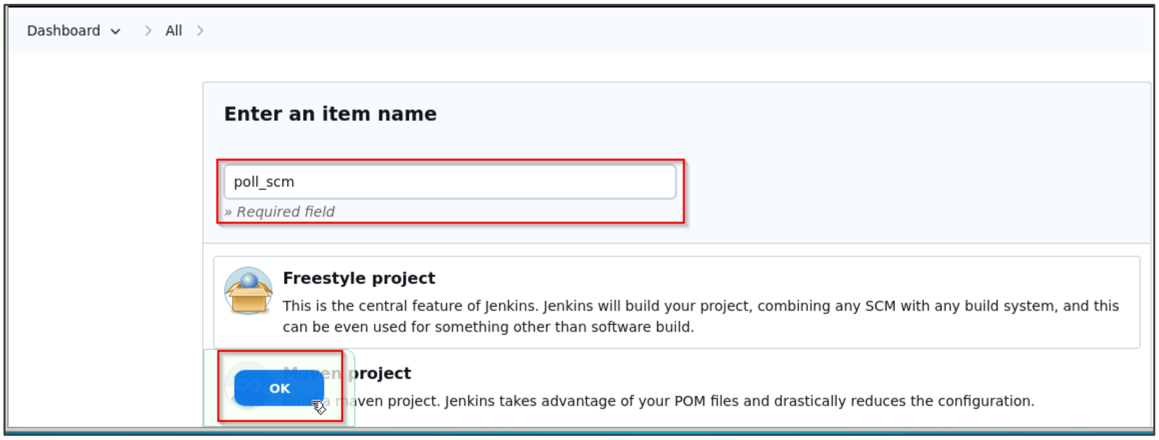
Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

.

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**



Enter an item name

poll_scm

» Required field

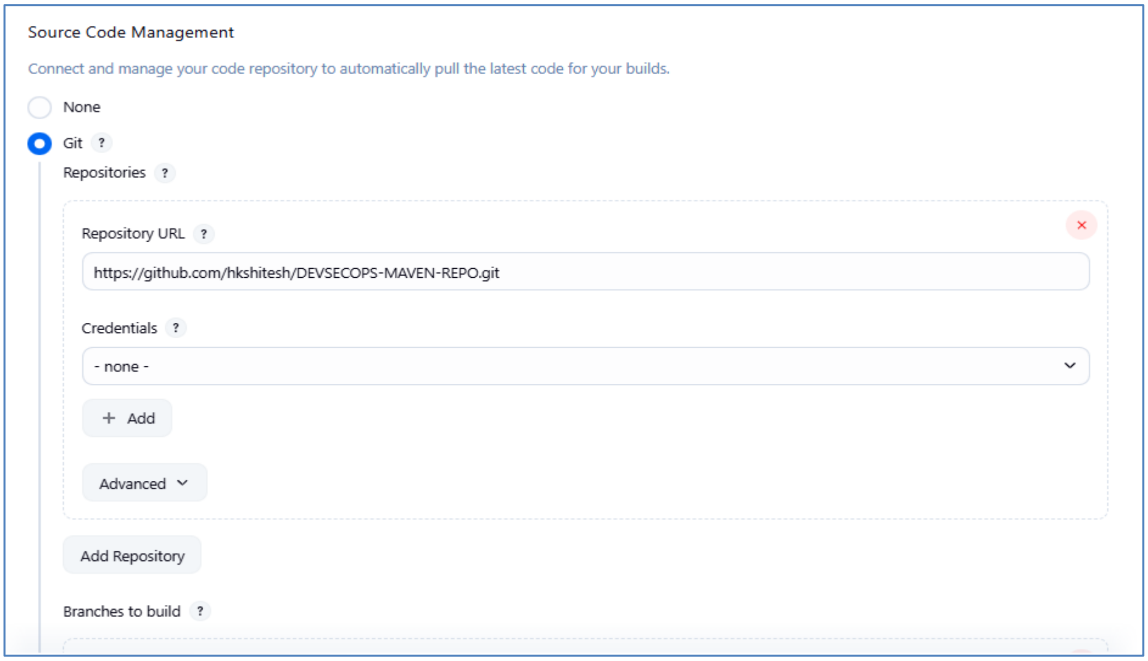
Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project

...ven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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

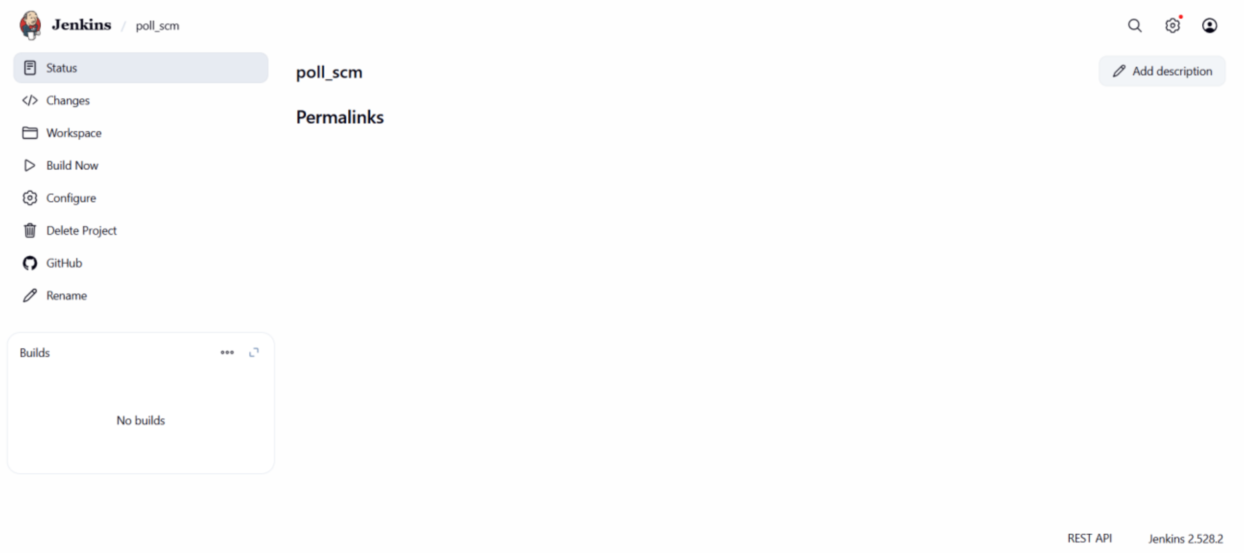
Repositories

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

Credentials: - none -

Branches to build

.



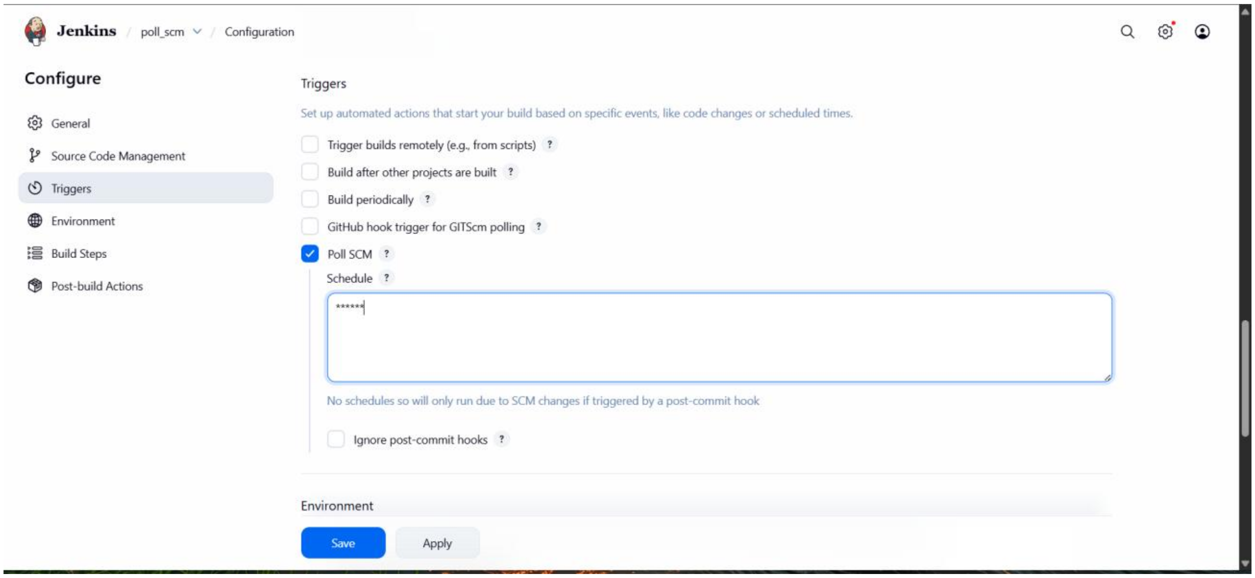
poll_scm

Permalinks

Builds

No builds

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 ?

```
******
```

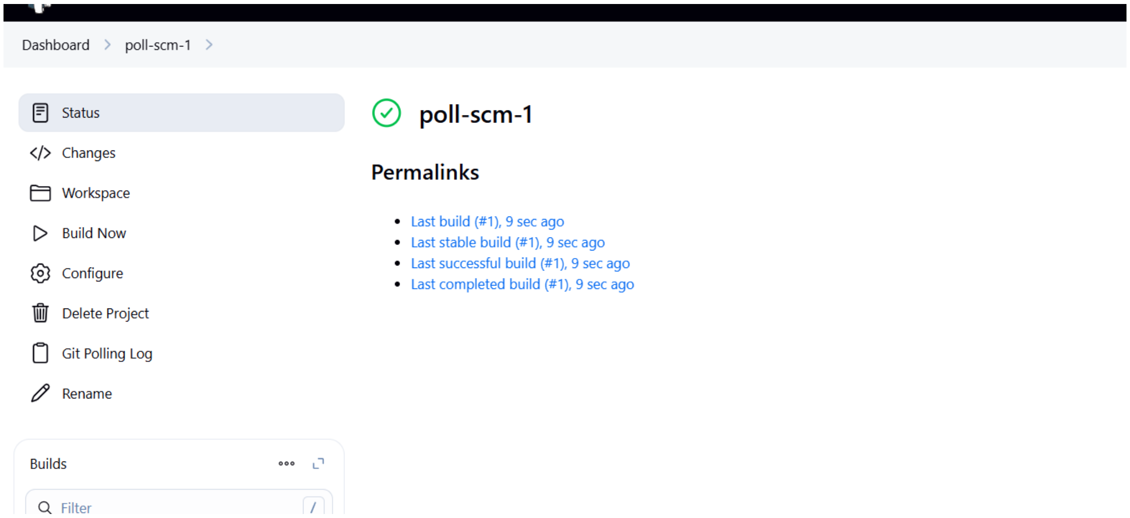
No schedules so will only run due to SCM changes if triggered by a post-commit hook

- □ Ignore post-commit hooks ?

Environment

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.

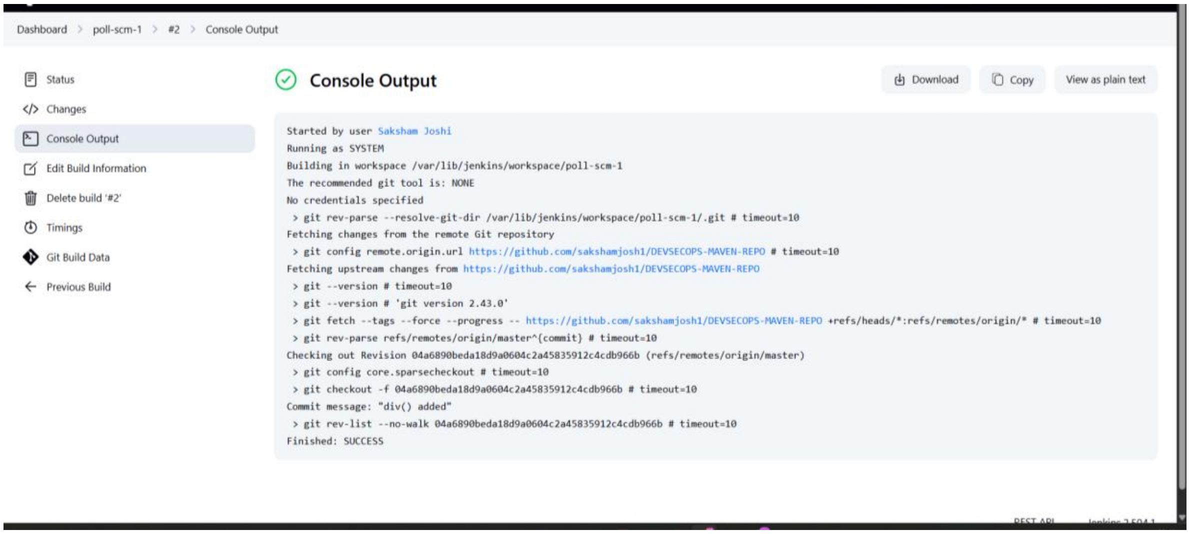
.



poll-scm-1

Permalinks

- Last build (#1), 9 sec ago
- Last stable build (#1), 9 sec ago
- Last successful build (#1), 9 sec ago
- Last completed build (#1), 9 sec ago



Console Output

```
Started by user Saksham Joshi
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/poll-scm-1
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/poll-scm-1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/sakshamjosh1/DEVSECOPS-MAVEN-REPO # timeout=10
Fetching upstream changes from https://github.com/sakshamjosh1/DEVSECOPS-MAVEN-REPO
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
 > git fetch --tags --force --progress -- https://github.com/sakshamjosh1/DEVSECOPS-MAVEN-REPO +refs/heads/*:refs/remotes/origin/* # timeout=10
 > 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"
 > git rev-list --no-walk 04a6890beda18d9a0604c2a45835912c4cdb966b # timeout=10
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.