.

Lab Exercise 9
Setting up Poll SCM Configuration in Jenkins

Name: Vishal Pandey
Sap id: 500125280
Enrolment no.: R2142231906
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

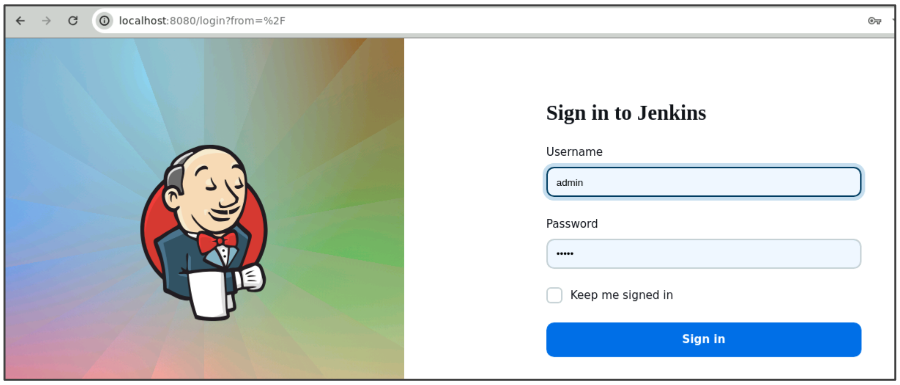
Prerequisites: You need to have a Jenkins up and running.

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



localhost:8080/login?from=%2F

Sign in to Jenkins

Username: admin

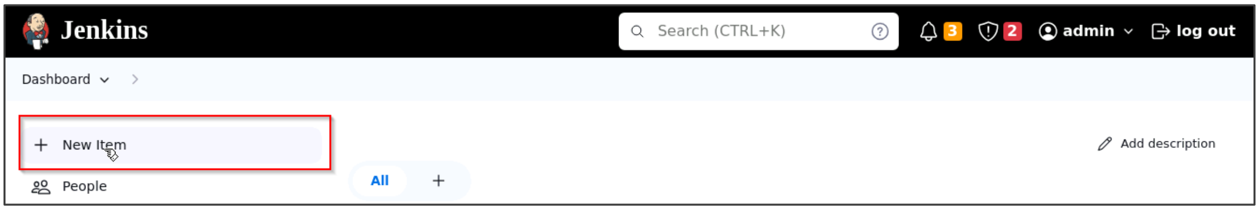
Password: •••••

□ Keep me signed in

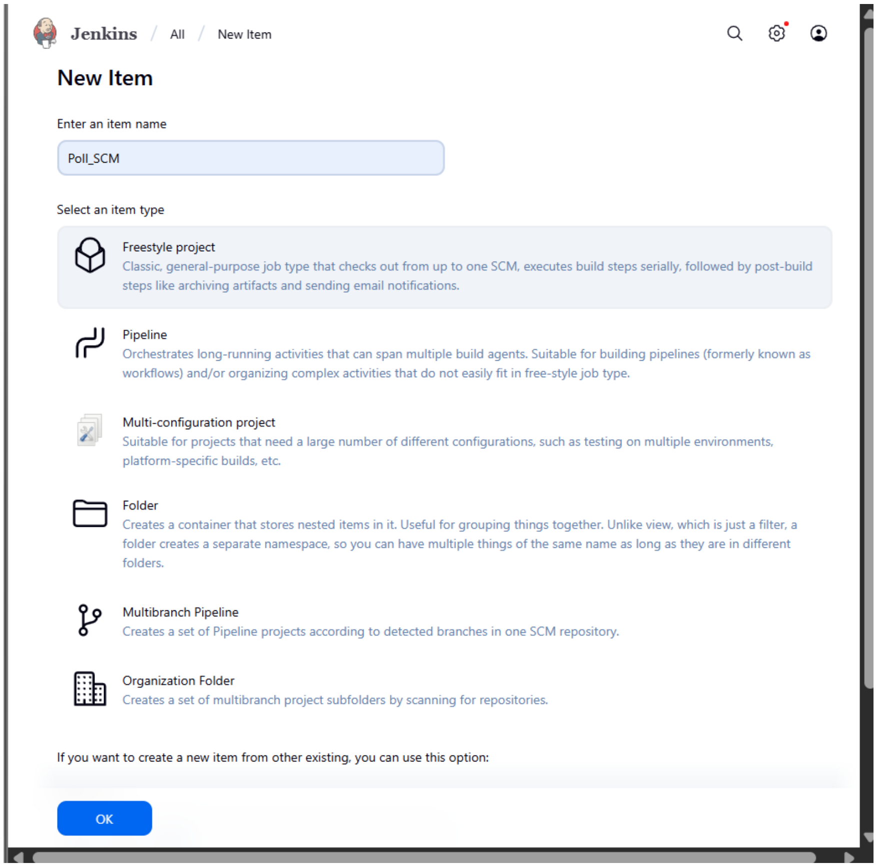
Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

.

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**



New Item

Enter an item name

Poll_SCM

Select an item type

Freestyle project
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.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

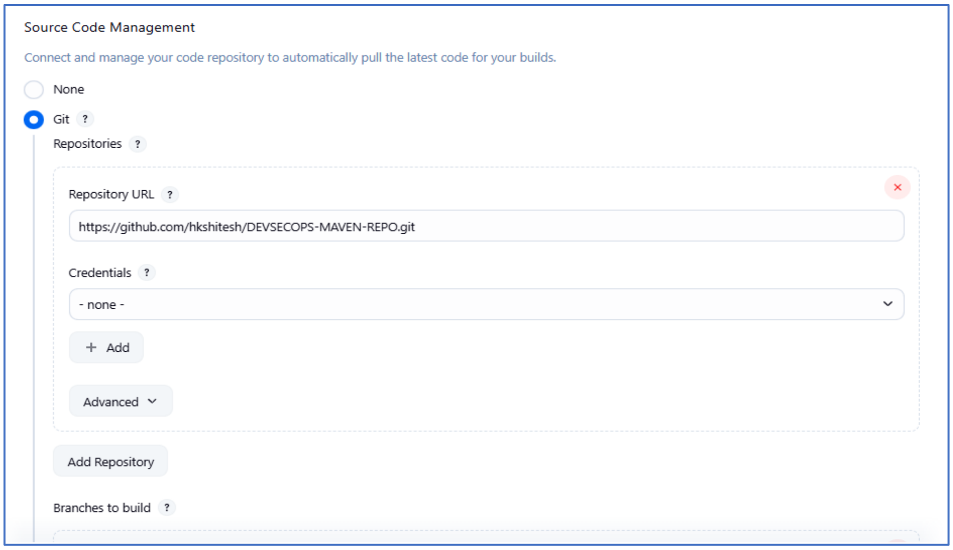
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

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 ?

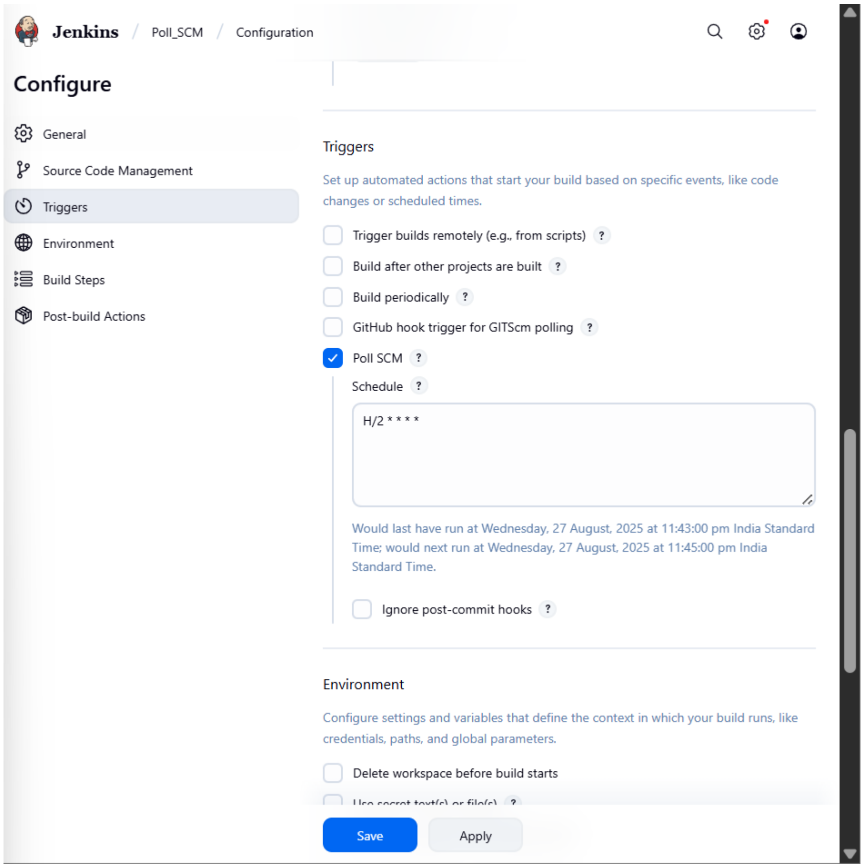
- none -

Branches to build ?

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 Wednesday, 27 August, 2025 at 11:43:00 pm India Standard Time; would next run at Wednesday, 27 August, 2025 at 11:45:00 pm India Standard Time.

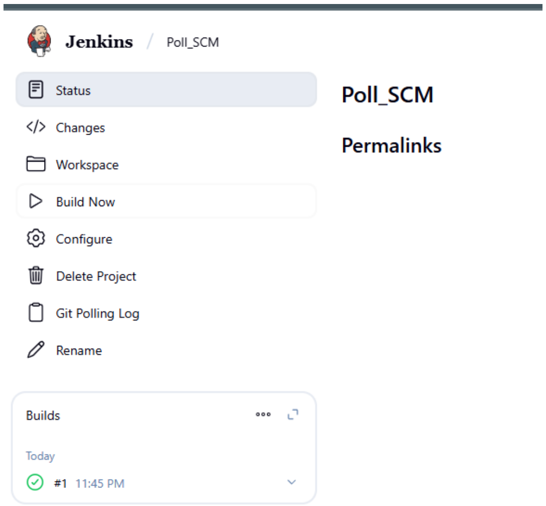
□ Ignore post-commit hooks ?

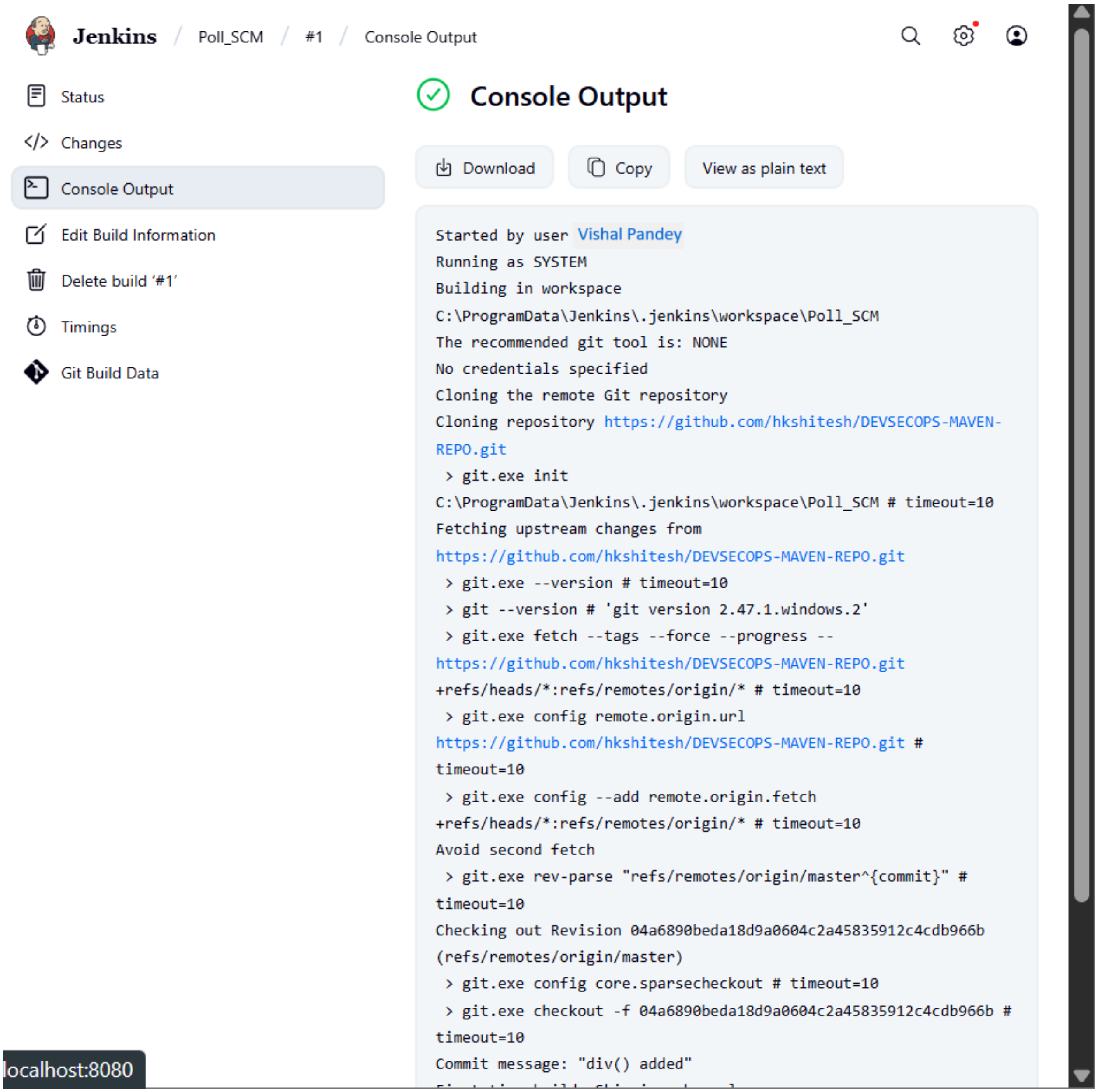
Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

□ Delete workspace before build starts

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.





```
Started by user Vishal Pandey
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"
```

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.