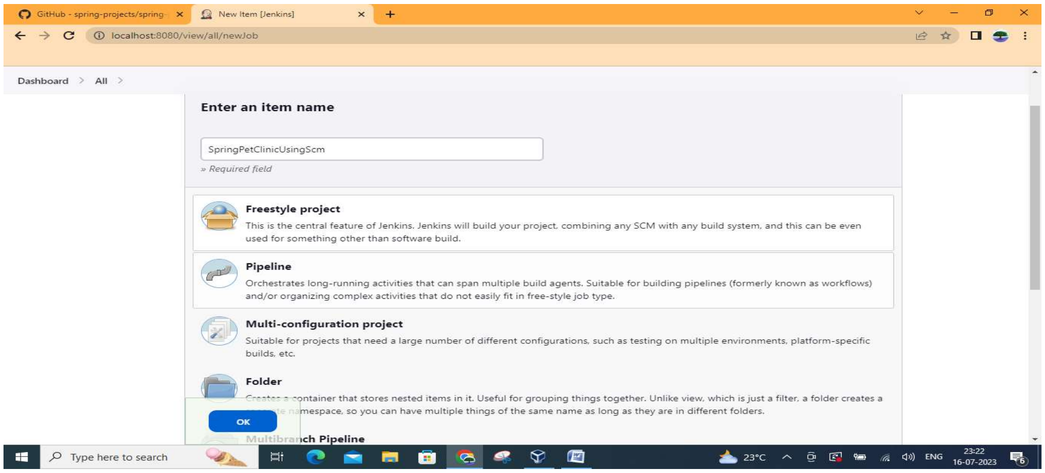
Assignment 5

Build Spring Pet Clinic using Jenkins with Poll SCM

Step 1: Set up the Spring Pet Clinic project in Jenkins

1. Create a new Jenkins job:

- Go to your Jenkins dashboard and click on "New Item."
- Enter a name for your job (e.g., "SpringPetClinicUsingScm").
- Select "Freestyle project" and click "OK."



Enter an item name

SpringPetClinicUsingScm

» 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.

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.

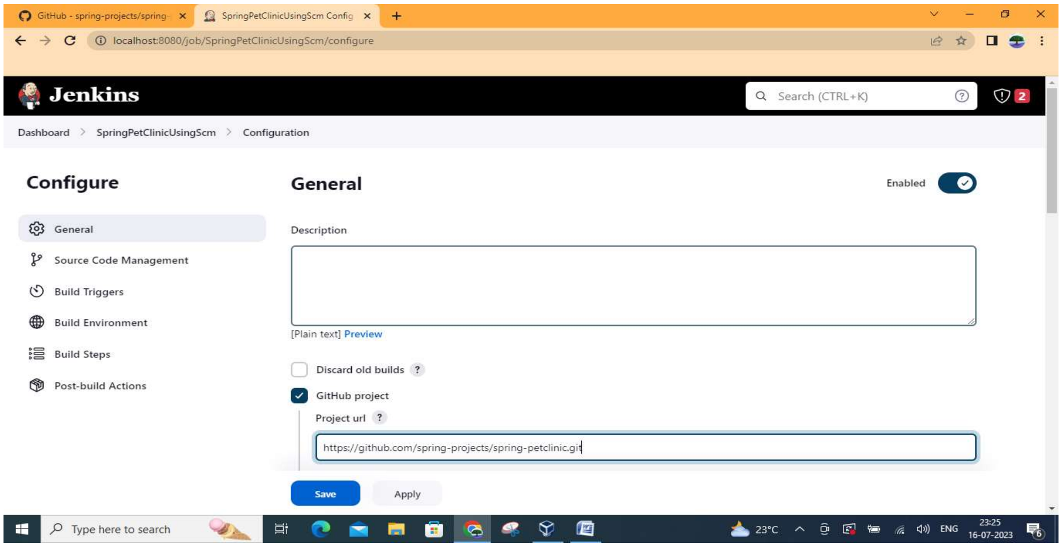
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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

2. Configure the job:

- Under the "General" section, select "GitHub project" and provide the URL of the Spring Pet Clinic repository.
- Repository URL: https://github.com/spring-projects/spring-petclinic.git



General

Enabled

Description

[Plain text] Preview

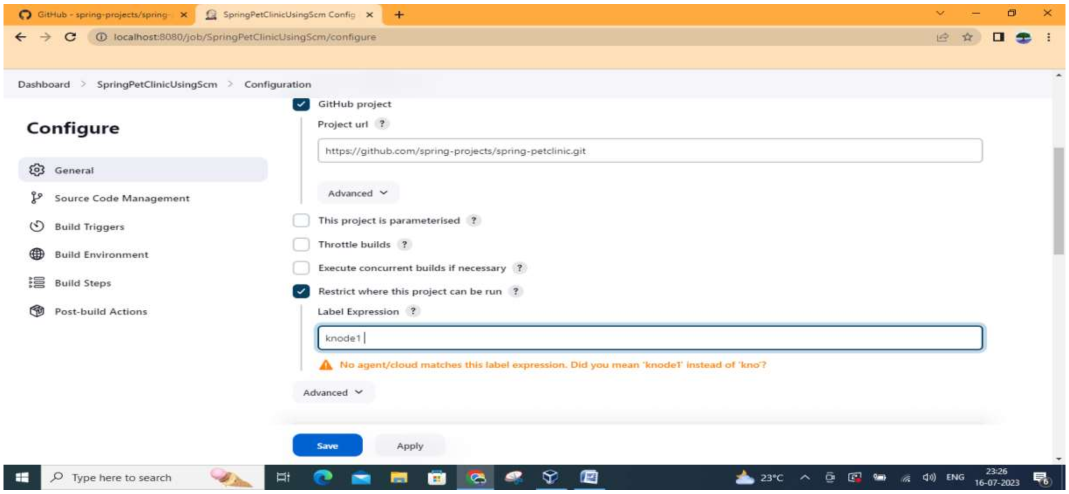
☐ Discard old builds

☑ GitHub project

Project url: https://github.com/spring-projects/spring-petclinic.git

3. Specify the Jenkins slave node (knode1) for the job:

- Under the "Restrict where this project can be run" section, enter "knode1" in the "Label Expression" field. This ensures the job runs on the specified slave node.


GitHub - spring-projects/spring
SpringPetClinicUsingScm Config
localhost:8080/job/SpringPetClinicUsingScm/configure
Dashboard > SpringPetClinicUsingScm > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Advanced
Advanced
Save
Apply
Type here to search
23°C
ENG
23:26
16-07-2023

☑ GitHub project

Project url: https://github.com/spring-projects/spring-petclinic.git

☐ This project is parameterised

☐ Throttle builds

☐ Execute concurrent builds if necessary

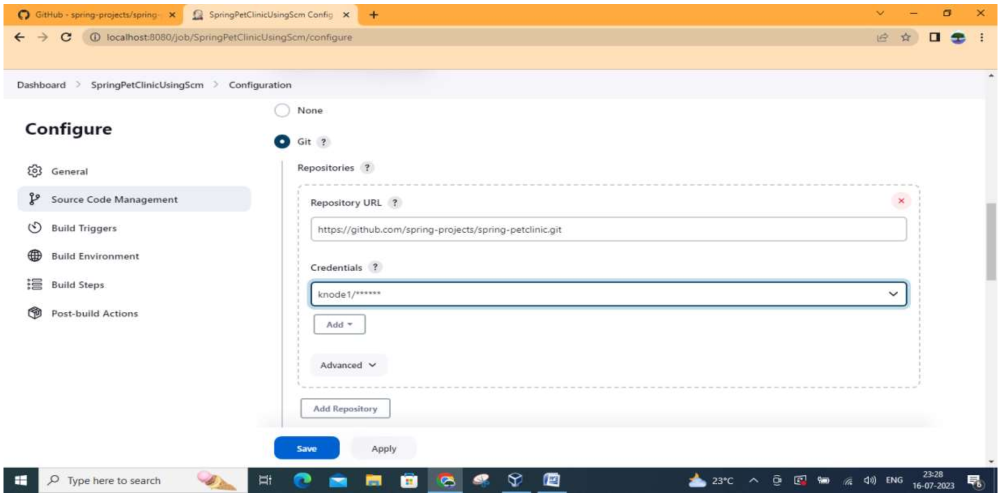
☑ Restrict where this project can be run

Label Expression: knode1

⚠ No agent/cloud matches this label expression. Did you mean 'knode1' instead of 'kno'?

4. Configure source code management:

- Under the "Source Code Management" section, choose "Git." Repository URL: https://github.com/spring-projects/spring-petclinic.git
- Optionally, you can specify credentials if your repository requires authentication.



○ None

● Git

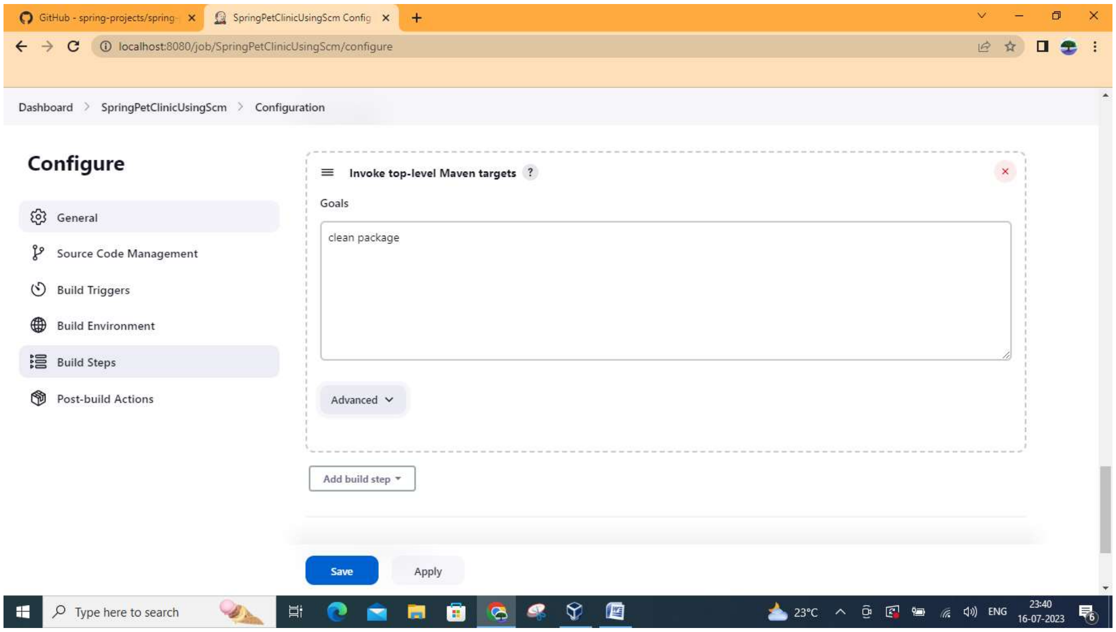
Repositories

Repository URL: https://github.com/spring-projects/spring-petclinic.git

Credentials: knode1/******

5. Build the project:

- In the "Build" section, click "Add build step" and choose "Invoke top-level Maven targets."
- Set the "Goals" field to `clean package` to build the Spring Pet Clinic application.



Invoke top-level Maven targets

Goals: clean package

- This command will clean the project and package it into a deployable artifact (usually a JAR or WAR file).

6. **Save the job configuration.**

Step 2: Configure Poll SCM

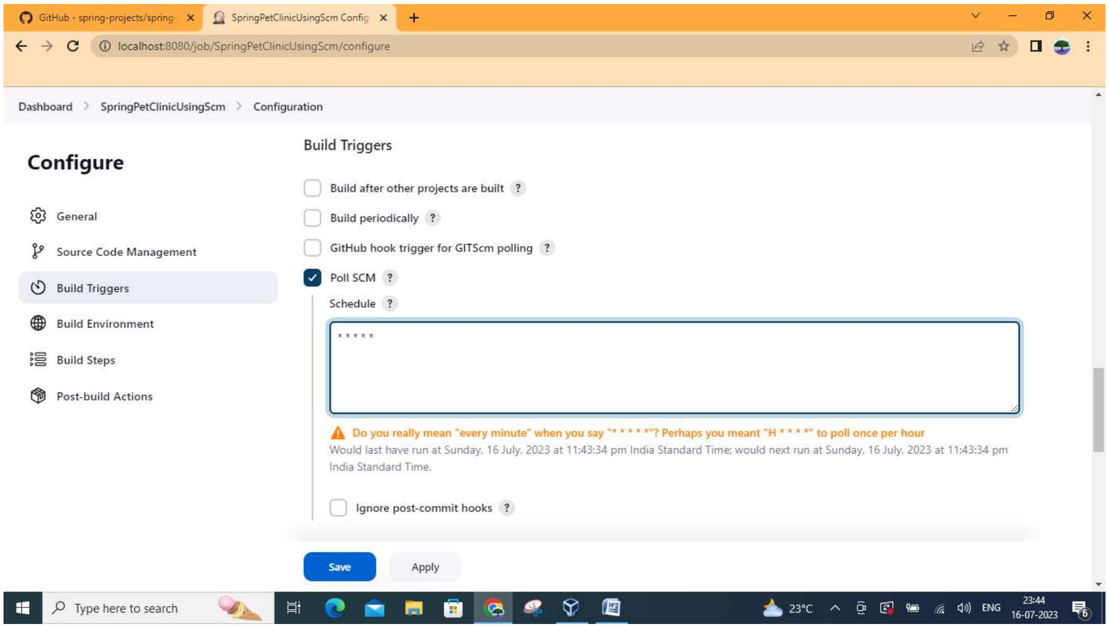
1. **Enable Poll SCM:**

- Go to the job configuration page.
- In the "Build Triggers" section, check the option "Poll SCM."

2. **Set Polling Schedule:**

- In the "Schedule" field, provide the polling schedule in cron format. For example, to poll every 1 minute, use `* * * * *`.
- This means Jenkins will check for changes in the repository every 5 minutes and trigger a build if there are any new changes.

3. **Save the job configuration**


GitHub - spring-projects/spring-
SpringPetClinicUsingScm Config
localhost:8080/job/SpringPetClinicUsingScm/configure
Dashboard > SpringPetClinicUsingScm > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Save
Apply
Type here to search
23°C
ENG
23:44
16-07-2023

Build Triggers

□ Build after other projects are built ?

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

☑ Poll SCM ?

Schedule ?

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Sunday, 16 July, 2023 at 11:43:34 pm India Standard Time; would next run at Sunday, 16 July, 2023 at 11:43:34 pm India Standard Time.

□ Ignore post-commit hooks ?

.

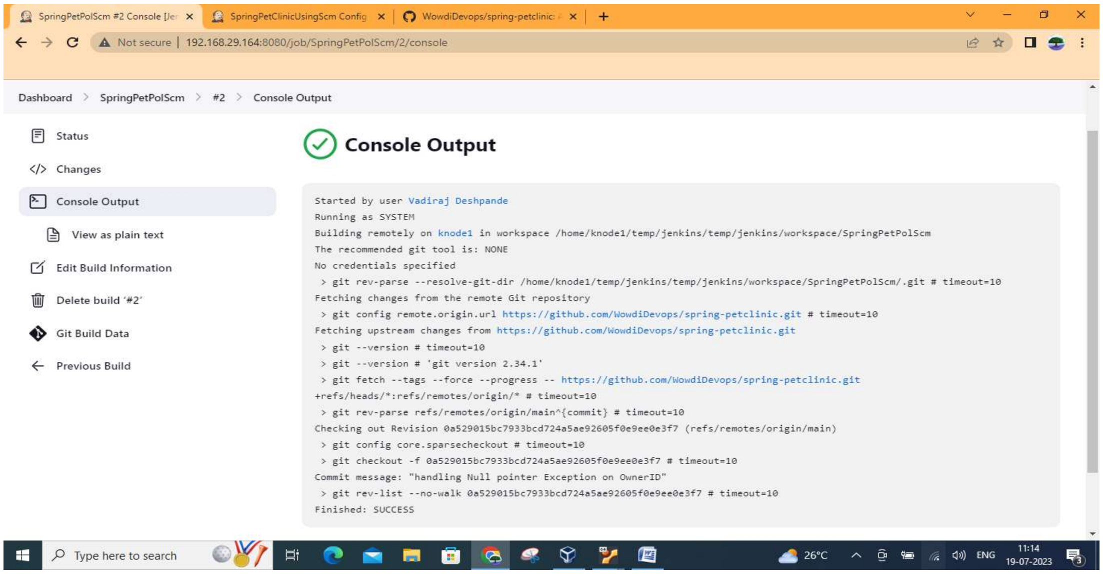
Step 3: Test the configuration

1. **Manually run the Jenkins job:**

- Go back to the Jenkins dashboard and click on your "SpringPetClinic" job.
- Click on "Build Now" to trigger a manual build.

2. **Verify the build status:**

- Once the build is completed, check the "Console Output" to ensure that the project was built successfully.



SpringPetPolScm #2 Console [Je
SpringPetClinicUsingScm Config
WowdiDevops/spring-petclinic
Not secure | 192.168.29.164:8080/job/SpringPetPolScm/2/console
Dashboard > SpringPetPolScm > #2 > Console Output
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#2'
Git Build Data
Previous Build
Type here to search
26°C
ENG
11:14
19-07-2023

Console Output

```
Started by user Vadiraj Deshpande
Running as SYSTEM
Building remotely on knode1 in workspace /home/knode1/temp/jenkins/temp/jenkins/workspace/SpringPetPolScm
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /home/knode1/temp/jenkins/temp/jenkins/workspace/SpringPetPolScm/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/WowdiDevops/spring-petclinic.git # timeout=10
Fetching upstream changes from https://github.com/WowdiDevops/spring-petclinic.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/WowdiDevops/spring-petclinic.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 0a529015bc7933bcd724a5ae92605f0e9ee0e3f7 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 0a529015bc7933bcd724a5ae92605f0e9ee0e3f7 # timeout=10
Commit message: "handling Null pointer Exception on OwnerID"
 > git rev-list --no-walk 0a529015bc7933bcd724a5ae92605f0e9ee0e3f7 # timeout=10
Finished: SUCCESS
```

3. **Polling SCM:**

- Wait for the scheduled polling (based on the configured schedule) to trigger an automatic build. Jenkins will check for changes in the Spring Pet Clinic repository and build the project if there are any new commits.

  With these steps, you have set up Jenkins to build the Spring Pet Clinic application using the Jenkins slave node "knode1" and configured poll SCM to trigger builds automatically based on the specified schedule. The Spring Pet Clinic repository link is: https://github.com/spring-projects/spring-petclinic.git