

Name – Rachit Shrivastava

Sap Id – 500119571

Batch 3 – DevOps

Lab Exercise 20

Creating a Pipeline Script

Objective: To create a pipeline script for automating build processes in Jenkins

Tools required: Jenkins

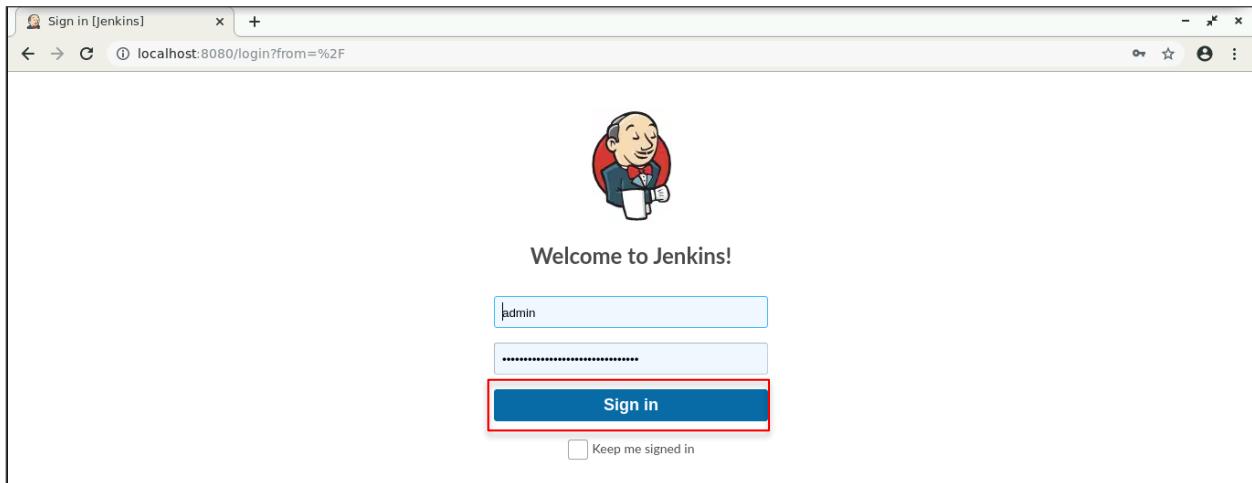
Prerequisites: None

Steps to be followed:

1. Log in to the Jenkins CI tool and create a pipeline script

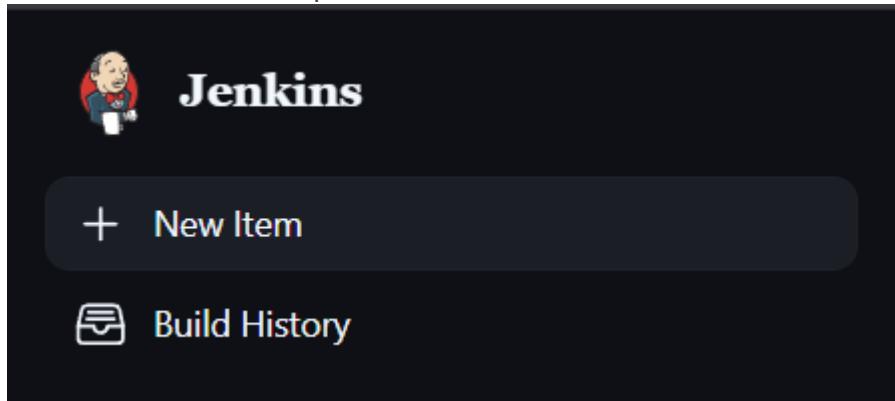
Step 1: Log in to the Jenkins CI tool and create a pipeline script

- 1.1 Open the browser, go to the Jenkins **Dashboard** by typing **localhost:8080** in your browser, provide the credentials, and click the **Sign in** button

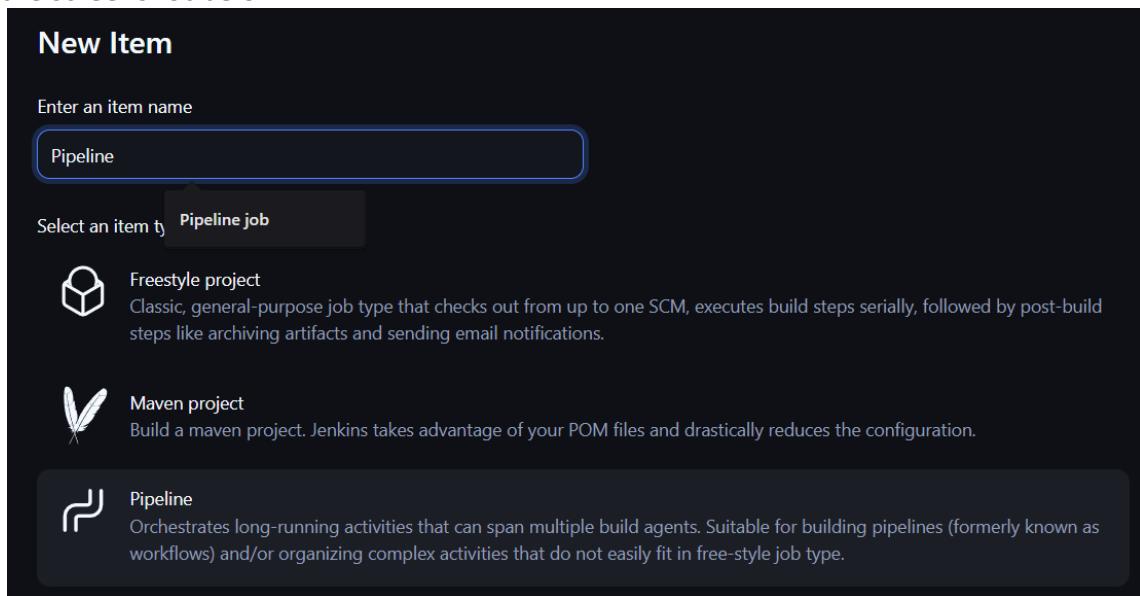


Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

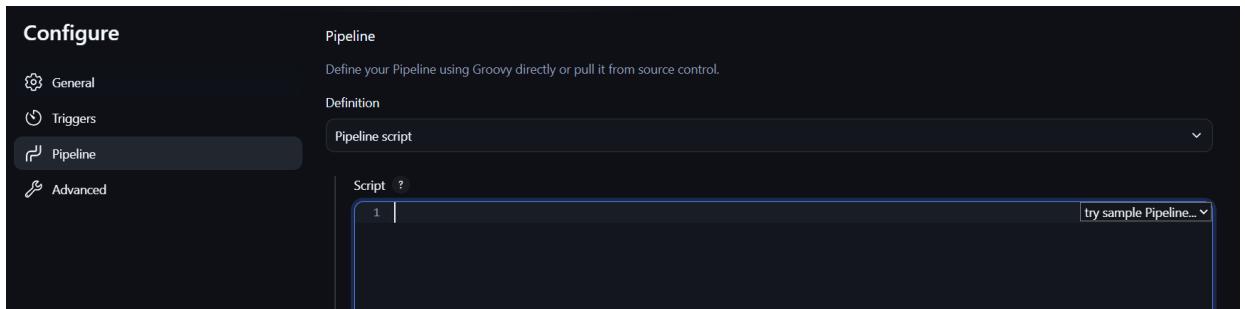
1.2 Click on the **New Item** option as shown in the screenshot below:



1.3 Enter a desired name for the project, select **Pipeline**, and then click on **OK** as shown in the screenshot below:

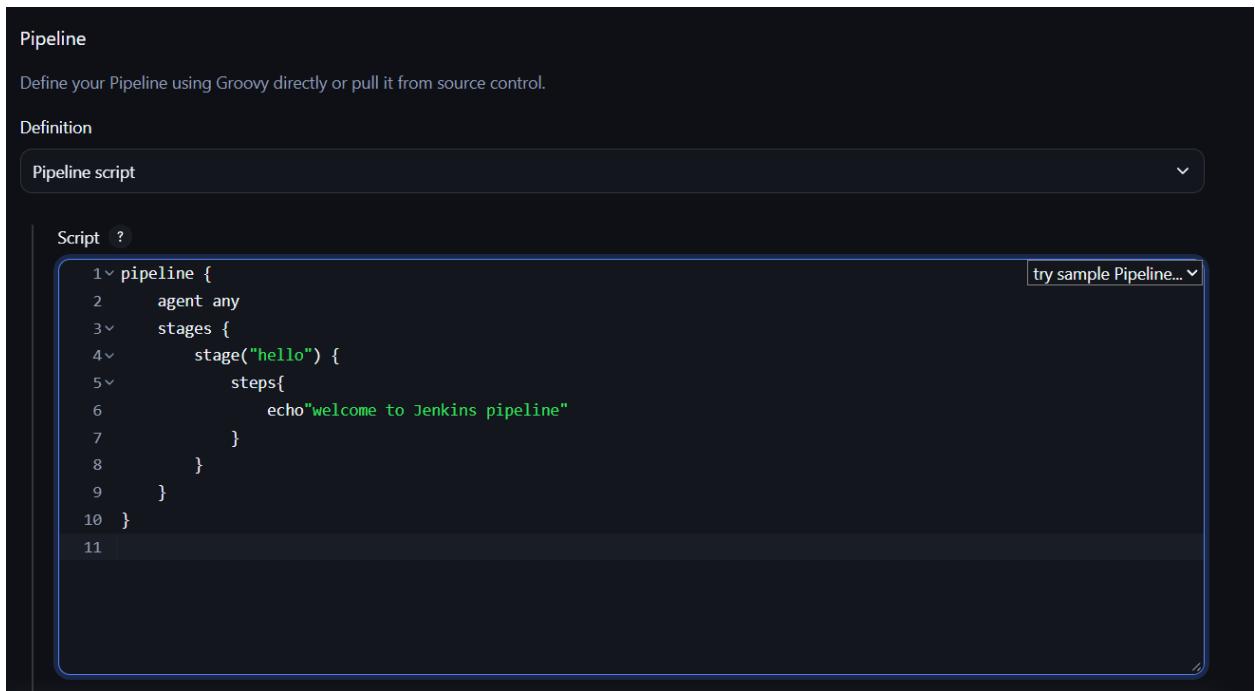


1.4 Click on **Pipeline** as shown in the screenshot below:

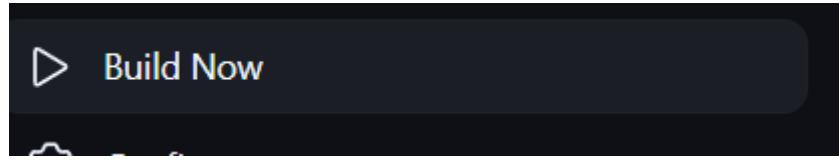


1.5 Enter the following pipeline script in the script editor and click on **Save** as shown in the screenshot below:

```
pipeline {
    agent any
    stages {
        stage("hello") {
            steps{
                echo"welcome to Jenkins pipeline"
            }
        }
    }
}
```



1.6 Click on **Build Now** to run the pipeline script as shown in the screenshot below:

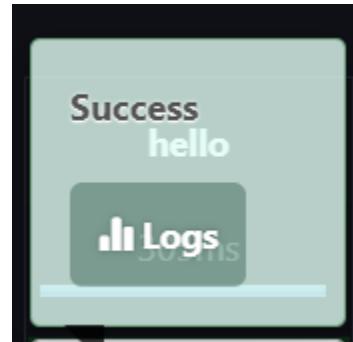


1.7 Hover over the milliseconds number next to the build stage name as shown in the screenshot below:

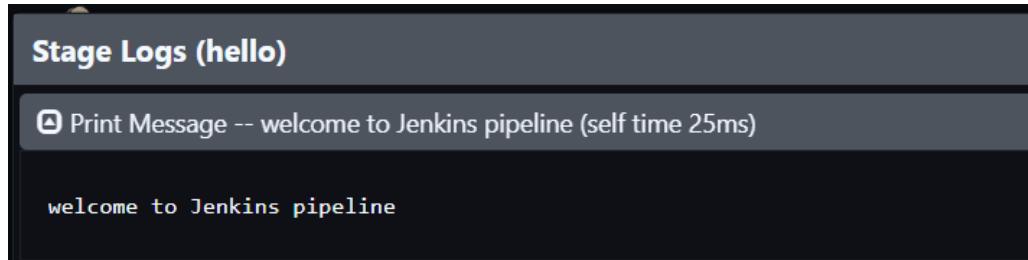


Note: Ensure that you hover the cursor over the milliseconds number without clicking on it

1.8 Click on **Logs** as shown in the screenshot below:



1.9 Check for the message in the top-left corner to confirm the successful execution of the pipeline stage as shown in the screenshot below:

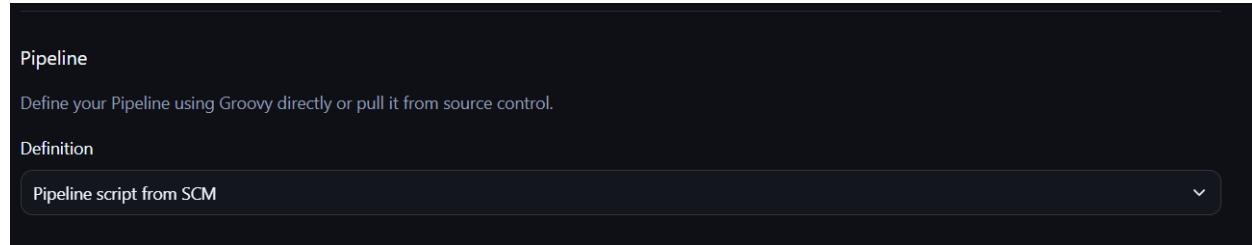


The screenshot shows the Jenkins Stage Logs interface for a pipeline stage named "hello". It displays a single log entry: "Print Message -- welcome to Jenkins pipeline (self time 25ms)" followed by the output "welcome to Jenkins pipeline".

Implementing Pipeline Script from SCM

Step 1: Change Pipeline Definition

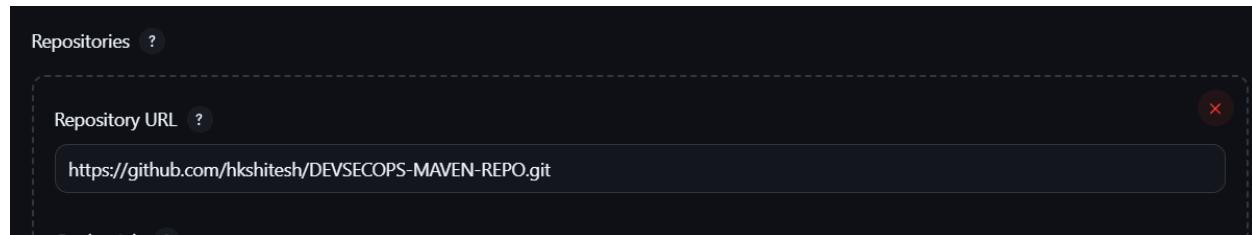
In the pipeline configuration screen, select "Pipeline script from SCM" from the **Definition** dropdown menu. This configures the Jenkins job to retrieve the pipeline script from a version control system like Git.



Step 2: Configure Repository

After selecting "Pipeline script from SCM", new options will appear. In the **Repositories** section, enter the **Repository URL** where the `JenkinsFile` is located.

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



Step 3: Configure Build Triggers

Set up an automated action to start the build based on specific events.

- Select the **Poll SCM** checkbox.
- In the **Schedule** text box, enter `H/15 * * * *` to instruct Jenkins to check the repository for any changes approximately every 15 minutes.

The screenshot shows the 'Triggers' configuration page for a Jenkins job. The title 'Triggers' is at the top. Below it is a descriptive text: 'Set up automated actions that start your build based on specific events, like code changes or scheduled times.' There are several checkboxes:

- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

The 'Poll SCM' checkbox is checked. To its right is a 'Schedule' field containing the cron expression `H/15 * * * *`. A note below the schedule says: 'Would last have run at Monday, 13 October, 2025 at 8:27:00 am India Standard Time; would next run at Monday, 13 October, 2025 at 8:42:00 am India Standard Time.' At the bottom of the list are two more checkboxes:

- Ignore post-commit hooks ?
- Trigger builds remotely (e.g., from scripts) ?

JenkinsFile used:

```
pipeline {
    agent any
    tools
    {
        maven 'MAVEN_HOME'
    }

    stages
    {
        stage('Welcome Stage')
        {
            steps
            {
                echo 'Welcome to Jenkins Pipeline'
            }
        }

        stage('Clean Stage')
        {
            steps
            {
                bat 'mvn clean'
            }
        }
        stage('Clean Success Stage')
        {
            steps
            {
                echo 'clean success'
            }
        }
        stage('Build & Install Stage')
        {
            steps
            {
                bat 'mvn install'
            }
        }

        stage('Build Success')
        {
            steps
            {
                echo 'Build successful'
            }
        }
    }
}
```

```

        }
        stage('Final Success')
        {
            steps
            {
                echo 'Final successful'
            }
        }
    }}

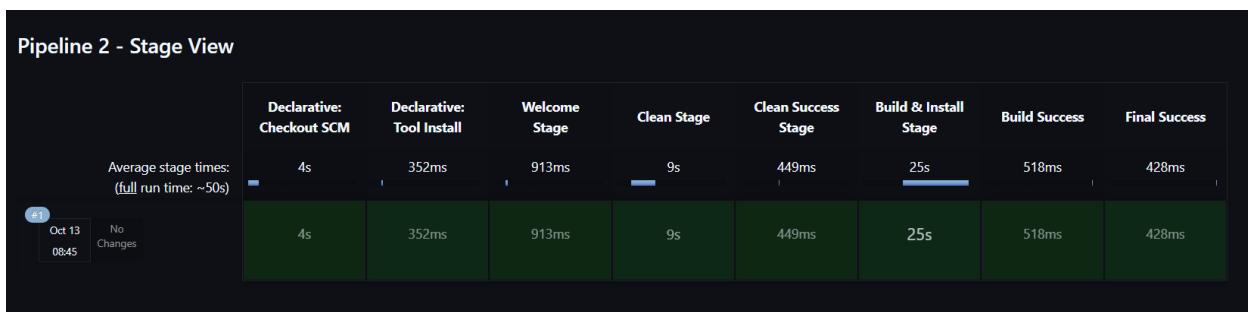
```

Step 4: Review the JenkinsFile

The repository contains a `JenkinsFile` that defines the pipeline's stages and steps. The script used in this exercise includes the following stages:

- **Welcome Stage**
- **Clean Stage**
- **Clean Success Stage**
- **Build & Install Stage**
- **Build Success**
- **Final Success**

The script also defines `maven` as a required tool and uses the `bat` step to run maven commands like `mvn clean` and `mvn install`.



Pipeline 2

 Add description

Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	Welcome Stage	Clean Stage	Clean Success Stage	Build & Install Stage	Build Success	Final Success
Average stage times: (full run time: ~44s)	3s	304ms	685ms	9s	425ms	21s	461ms	424ms
#2 Oct 13 08:48 No Changes	2s	257ms	457ms	8s	401ms	18s	405ms	421ms
#1 Oct 13 08:45 No Changes	4s	352ms	913ms	9s	449ms	25s	518ms	428ms