

# Lab Exercise 9 – Setting Up Poll SCM

## Configurations In Jenkins

**Name:-Vansh Bhatt**

**SapId:- 500125395**

**R.No:- R2142231689**

**Batch:- DevOps B1**

**To:- Hitesh Kumar Sharma Sir**

**Objective:** To set up poll SCM configuration in Jenkins for automating build triggering and optimizing resource usage on Jenkins servers

**Tools required:** Jenkins

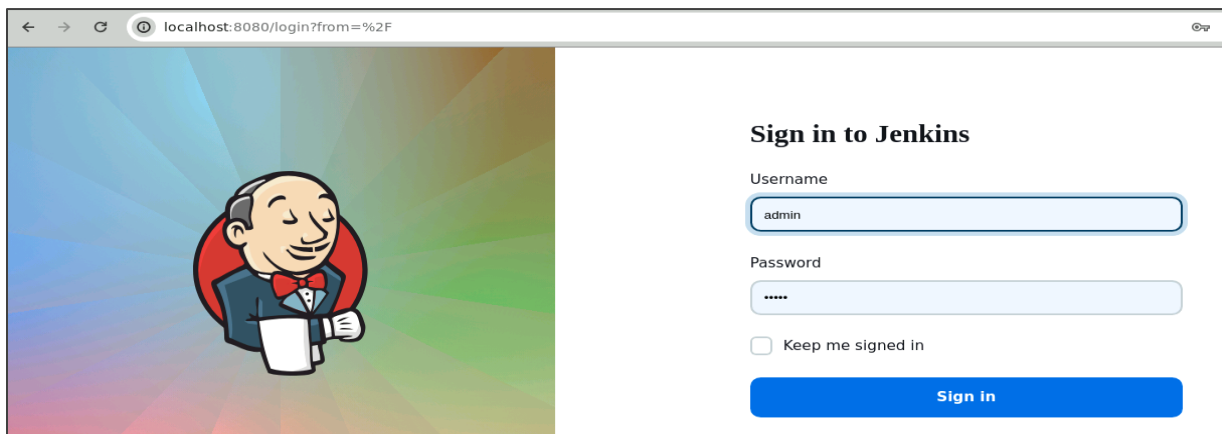
**Prerequisites:** You need to have a Jenkins up and running.

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**

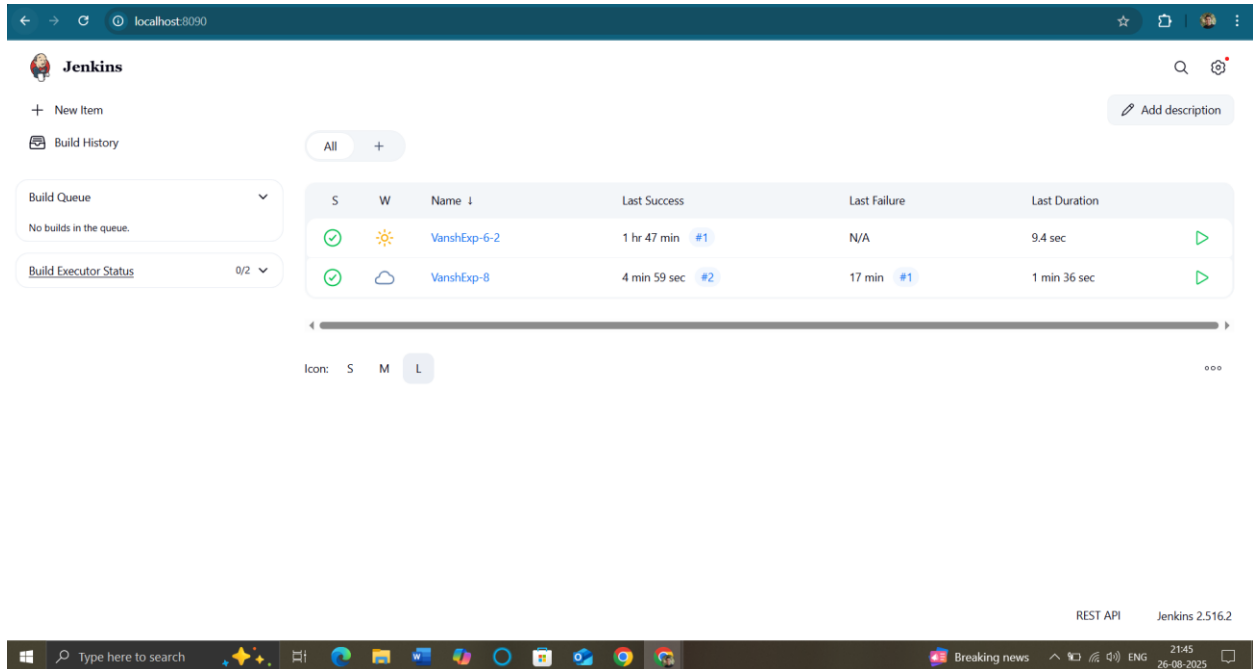
#### **1.1 Log in to Jenkins using the credentials**



The screenshot shows the Jenkins login interface in a web browser. The address bar displays 'localhost:8080/login?from=%2F'. On the left, there is a cartoon illustration of a man in a tuxedo. On the right, the title 'Sign in to Jenkins' is followed by two input fields: 'Username' with the value 'admin' and 'Password' with masked characters '.....'. Below these fields is a checkbox labeled 'Keep me signed in' which is unchecked. At the bottom right is a blue 'Sign in' button.

**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**

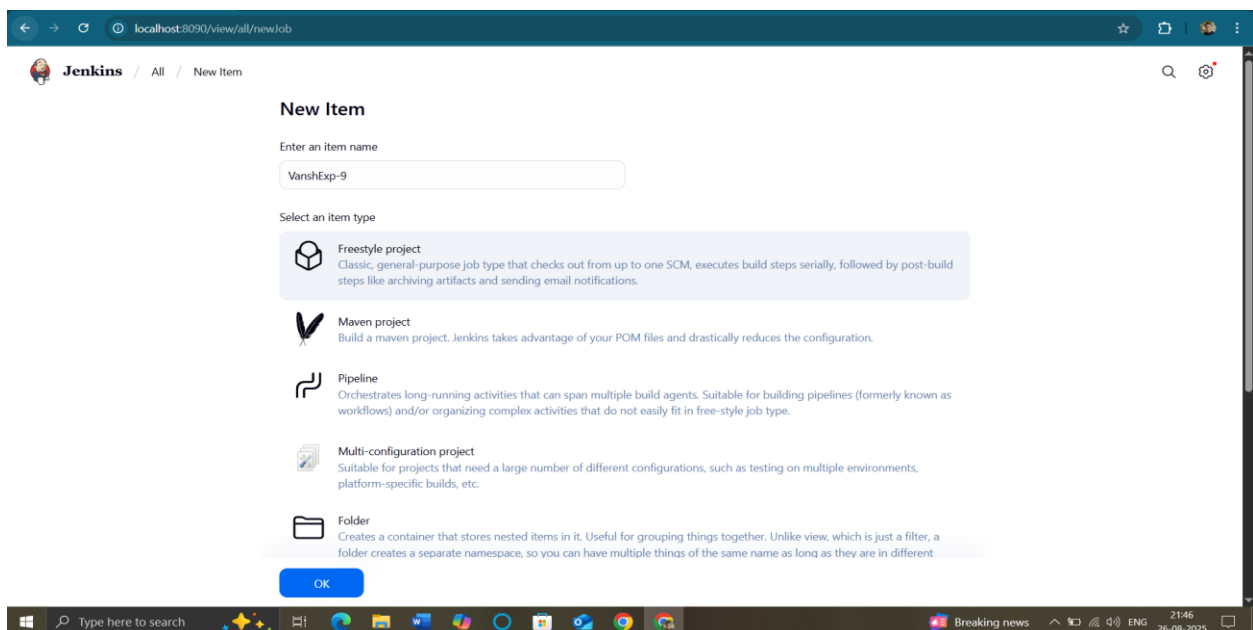


The screenshot shows the Jenkins dashboard at localhost:8090. The 'New Item' button is visible in the top left. A table lists existing jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	VanshExp-6-2	1 hr 47 min #1	N/A	9.4 sec
✓	☁	VanshExp-8	4 min 59 sec #2	17 min #1	1 min 36 sec

REST API Jenkins 2.516.2

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



The screenshot shows the 'New Item' form in Jenkins. The item name is 'VanshExp-9'. The 'Freestyle project' option is selected under 'Select an item type'.

**New Item**

Enter an item name

VanshExp-9

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.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- 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

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

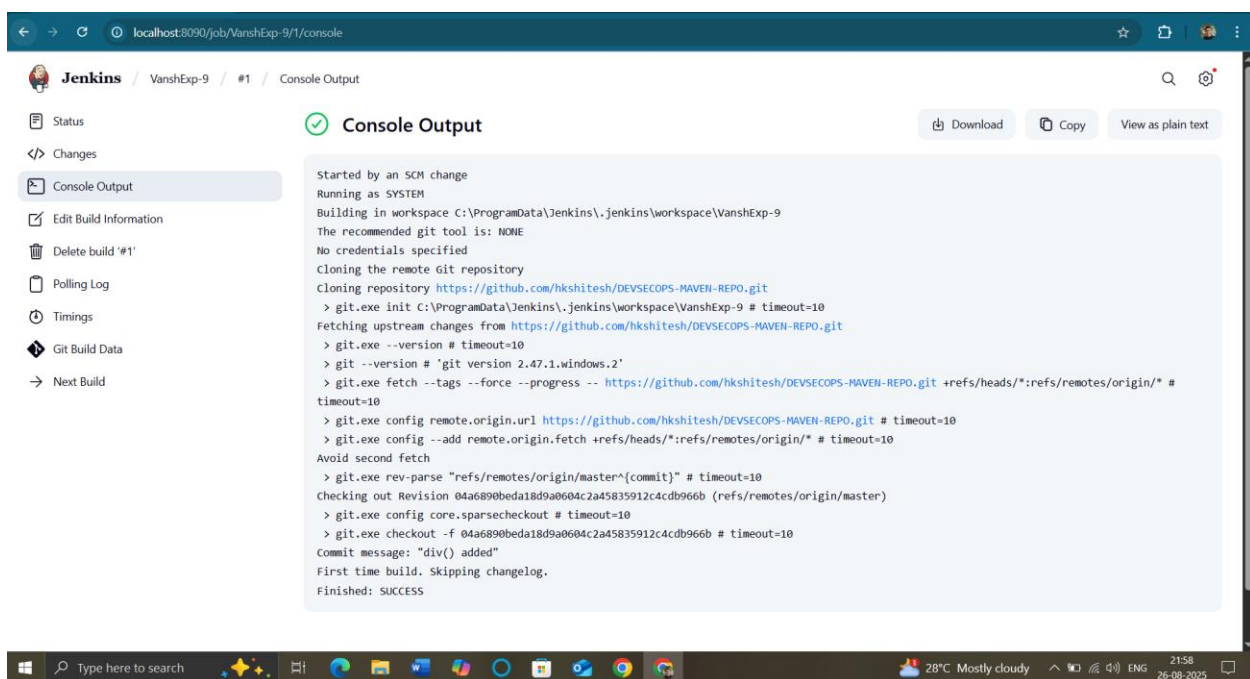
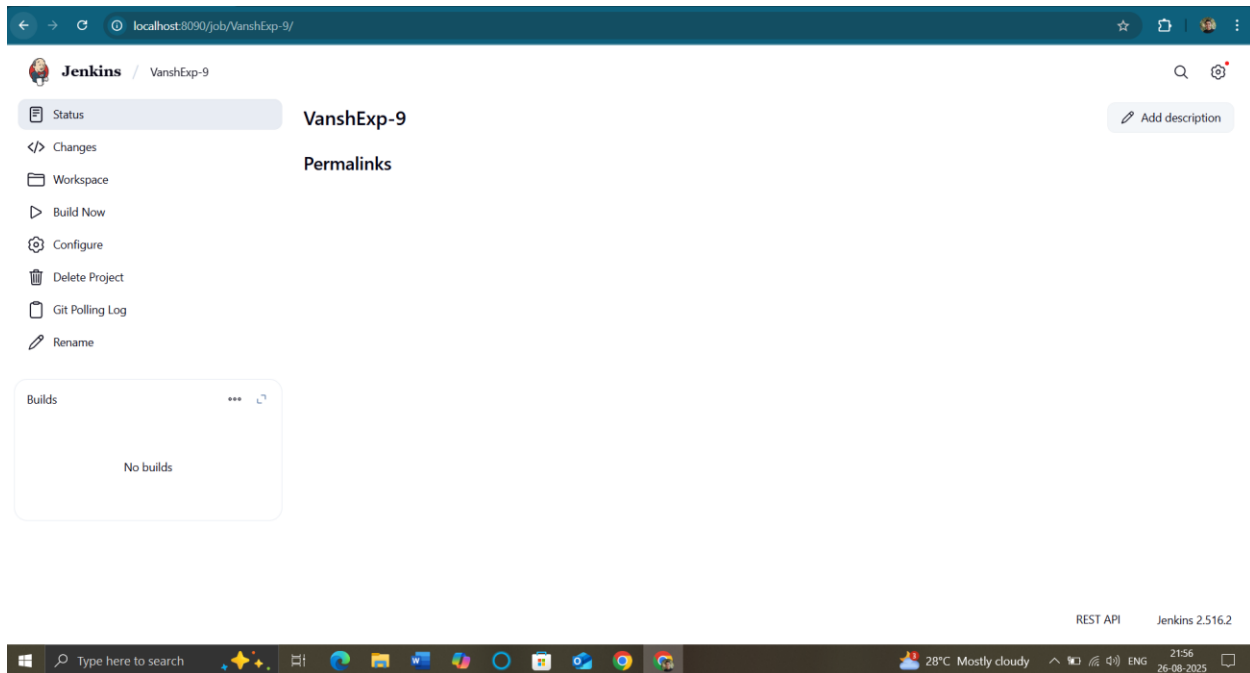
The screenshot shows the 'Source Code Management' configuration page in Jenkins. At the top, it says 'Connect and manage your code repository to automatically pull the latest code for your builds.' Below this, there are two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Under the 'Git' section, there is a 'Repositories' section with a red 'X' icon. Inside this section, there is a 'Repository URL' field containing 'https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git'. Below the URL field is a 'Credentials' dropdown menu showing '- none -'. There are also '+ Add' and 'Advanced' buttons. At the bottom of the 'Repositories' section is an 'Add Repository' button. Below the 'Repositories' section 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 Jenkins Configuration page for 'VanshExp-9'. The left sidebar shows the configuration tree with 'Triggers' selected. The main content area is titled 'Triggers' and contains the text 'Set up automated actions that start your build based on specific events, like code changes or scheduled times.' There are four checkboxes: '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 checkboxes is a 'Schedule' field containing 'H/2\*\*\*\*'. A warning message is displayed below the schedule field: 'Do you really mean "every minute" when you say "H/2\*\*\*\*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour. Would last have run at Tuesday, August 26, 2025, 9:54:00 PM India Standard Time; would next run at Tuesday, August 26, 2025, 9:55:00 PM India Standard Time.' There is also an 'Ignore post-commit hooks' checkbox. At the bottom of the page are 'Save' and 'Apply' buttons. The bottom status bar shows the system clock as 21:55 on 26-08-2025.

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.