

Lab Exercise 6.1

Managing Jenkins Freestyle Jobs

Name Pulkit

Sap id 500125835

Devops_b3

Objective: To manage Jenkins job by creating, renaming, and deleting freestyle job, facilitating streamlined job configuration and maintenance within the Jenkins environment

Tools required: Jenkins

Prerequisites: None

Steps to be followed:

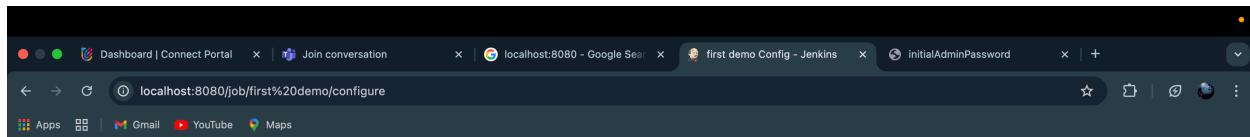
1. Create a new Freestyle Job
2. Rename and delete the FreeStyle Job

Step 1: Create a new Freestyle Job

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

The screenshot shows the Jenkins dashboard at localhost:8080. The top navigation bar includes links for 'Dashboard | Connect Portal', 'Join conversation', 'localhost:8080 - Google Search', 'Dashboard - Jenkins', and 'initialAdminPassword'. The main content area features the Jenkins logo and the heading 'Welcome to Jenkins!'. It displays a message about managing jobs and provides links to 'Create a job', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. On the left, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2). A search bar and a 'Add description' button are located in the top right.

The screenshot shows the 'New Item' dialog in Jenkins at localhost:8080/view/all/newJob. The title is 'New Item'. It asks for an item name ('first demo') and allows selecting an item type. The 'Freestyle project' option is selected and described as a 'Classic, general-purpose job type'. Other options shown include 'Pipeline', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'. At the bottom is an 'OK' button.



Jenkins / first demo / Configuration

Configure General Enabled

General

Description

Plain text [Preview](#)

Discard old builds ?

GitHub project

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

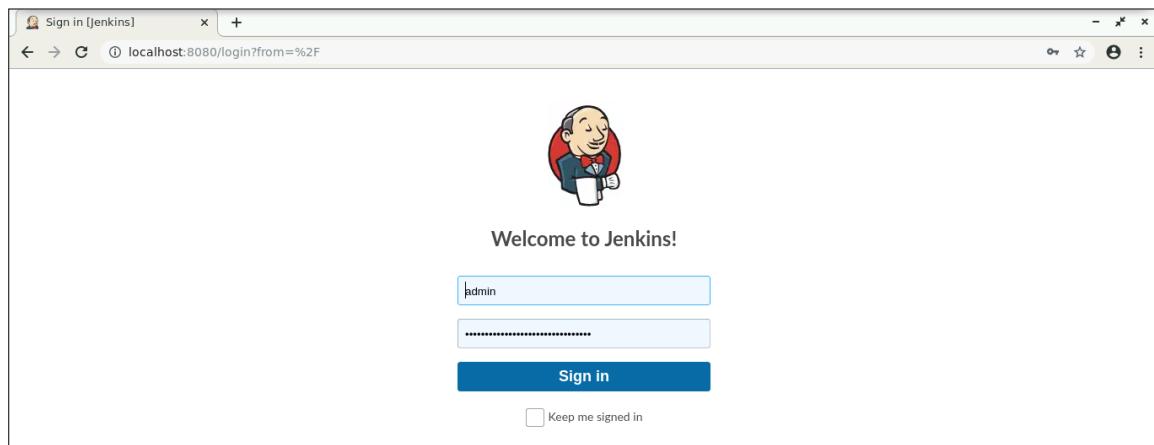
[Advanced](#) ▾

Source Code Management

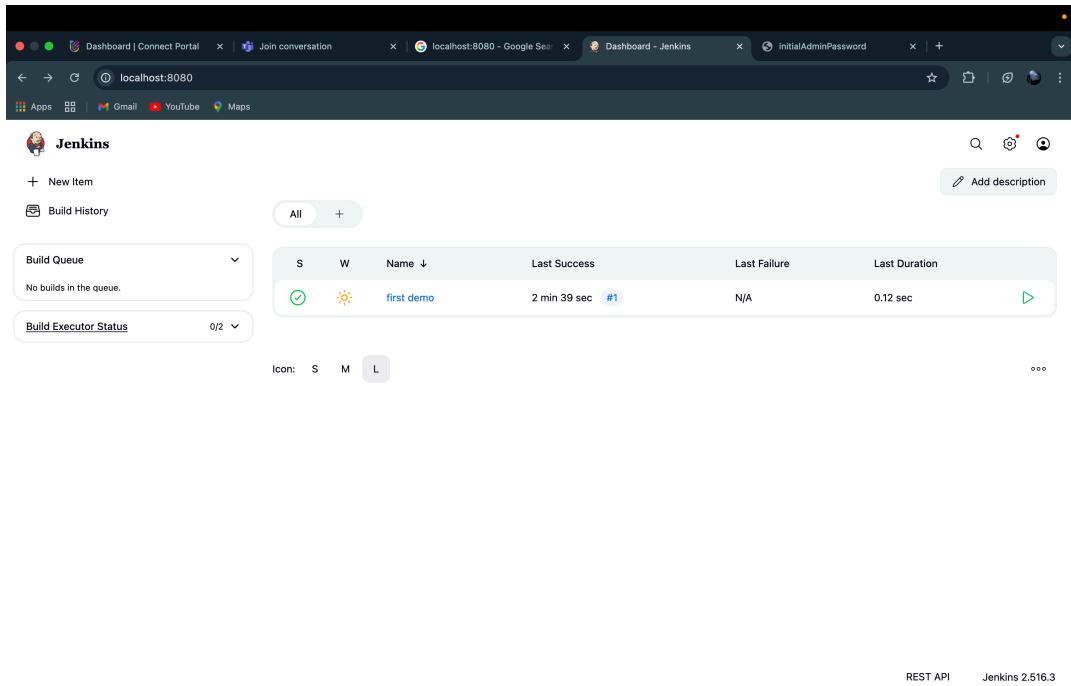
Connect and manage your code repository to automatically pull the latest code for your builds.

None

[Save](#) [Apply](#)



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

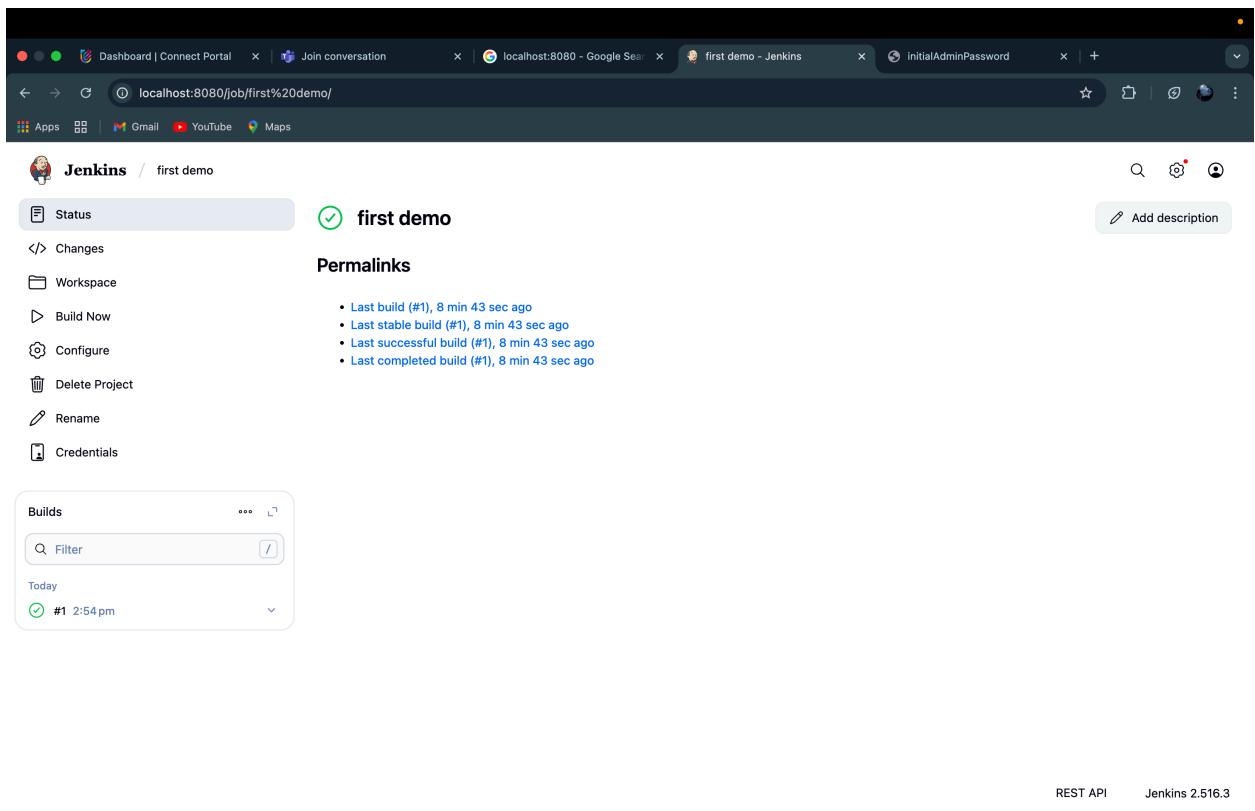


2. Click on **New Item** in the Jenkins Dashboard
3. Enter a name for your project, select **Freestyle project** as the build job type, and click on **OK**
4. Click on **Save**

Step 2: Rename and delete the Freestyle Job

1. Click on the name of the project that you have created as shown in the screenshot below:

2. Click on the **Rename** button to rename the project



3. Enter a desired name to the project and click on **Rename**

The screenshot shows a Jenkins build summary page for a project named 'part 2'. The build number is #2, and it was last updated on 20-Sept-2025 at 4:20:03 pm. The build was started by a user named 'pullkit'. The status bar indicates the build took 93 ms. A sidebar on the left provides links for Status, Changes, Console Output, Edit Build Information, Delete build '#2', and Timings. The main content area shows a message indicating '</> No changes.' The footer of the page includes links for REST API and Jenkins 2.516.3.

4. Click on the renamed project and then click on **Delete Project** to delete the project

By following these steps, you have successfully managed Jenkins jobs by creating, renaming, and deleting Freestyle Job, streamlining your project workflows within the Jenkins environment.

Lab Exercise 6.2

Creating a New Jenkins Job to Checkout Source Code

Objective: To set up a Jenkins job to manage source code, specifically by configuring the Source Code Management section to check out code from a Git repository

Tools required: Jenkins

Prerequisites: Jenkins must be operational.

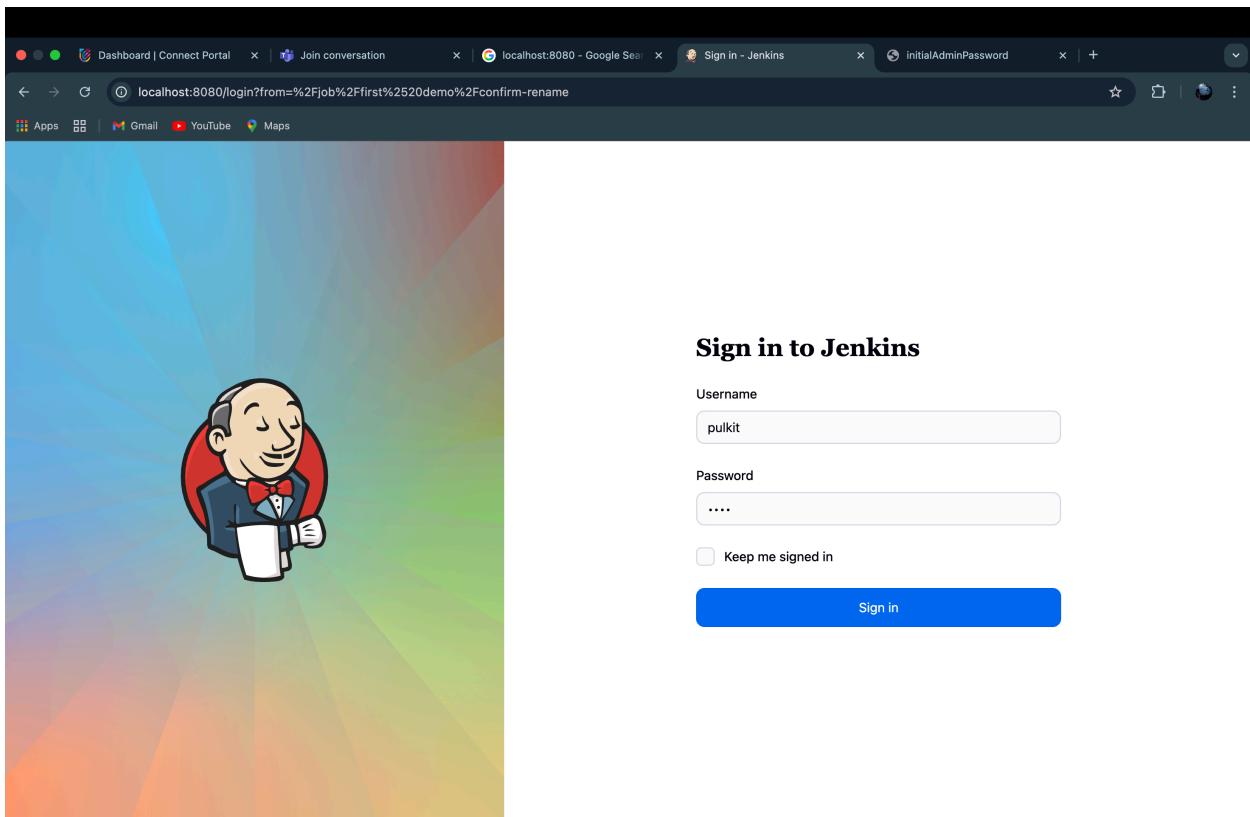
Steps to be followed:

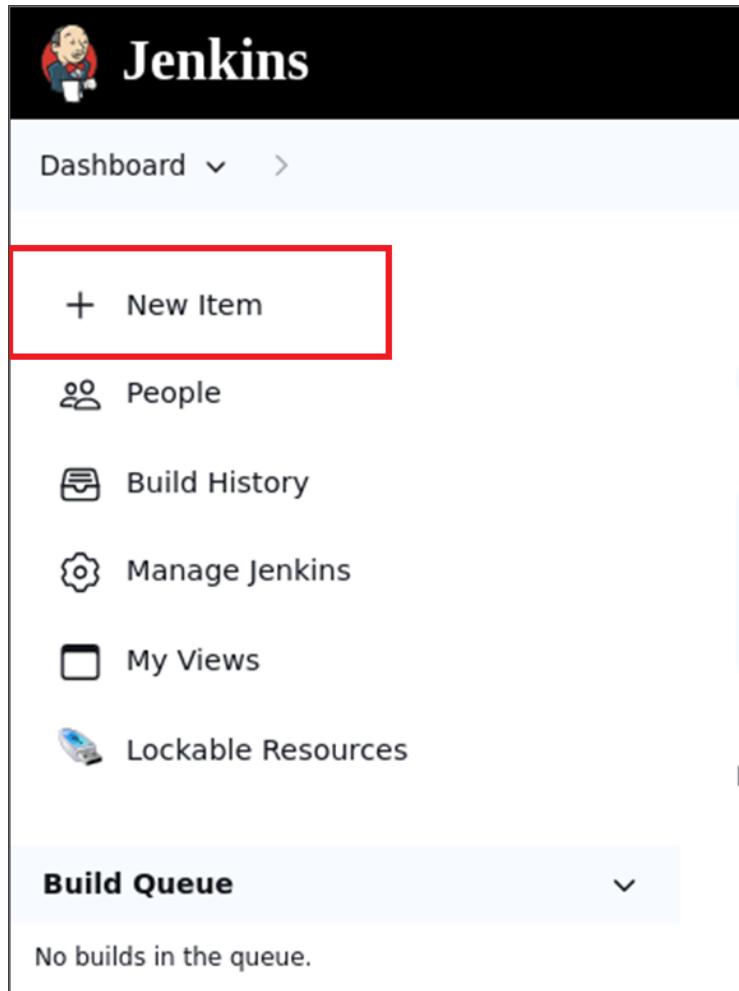
1. Log in and create a Jenkins job
2. Configure source code management

Step 1: Log in and create a Jenkins job

1. Navigate to **localhost:8080** in your web browser, enter your credentials, and click on **Sign In**
2. Create a new Jenkins job by clicking on **New Item**

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





3. Provide custom job name inside the field **Enter an item name**, select the **Freestyle project** option, and click on the **OK** button to save the job

Enter an item name

FreeStyle
» 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.

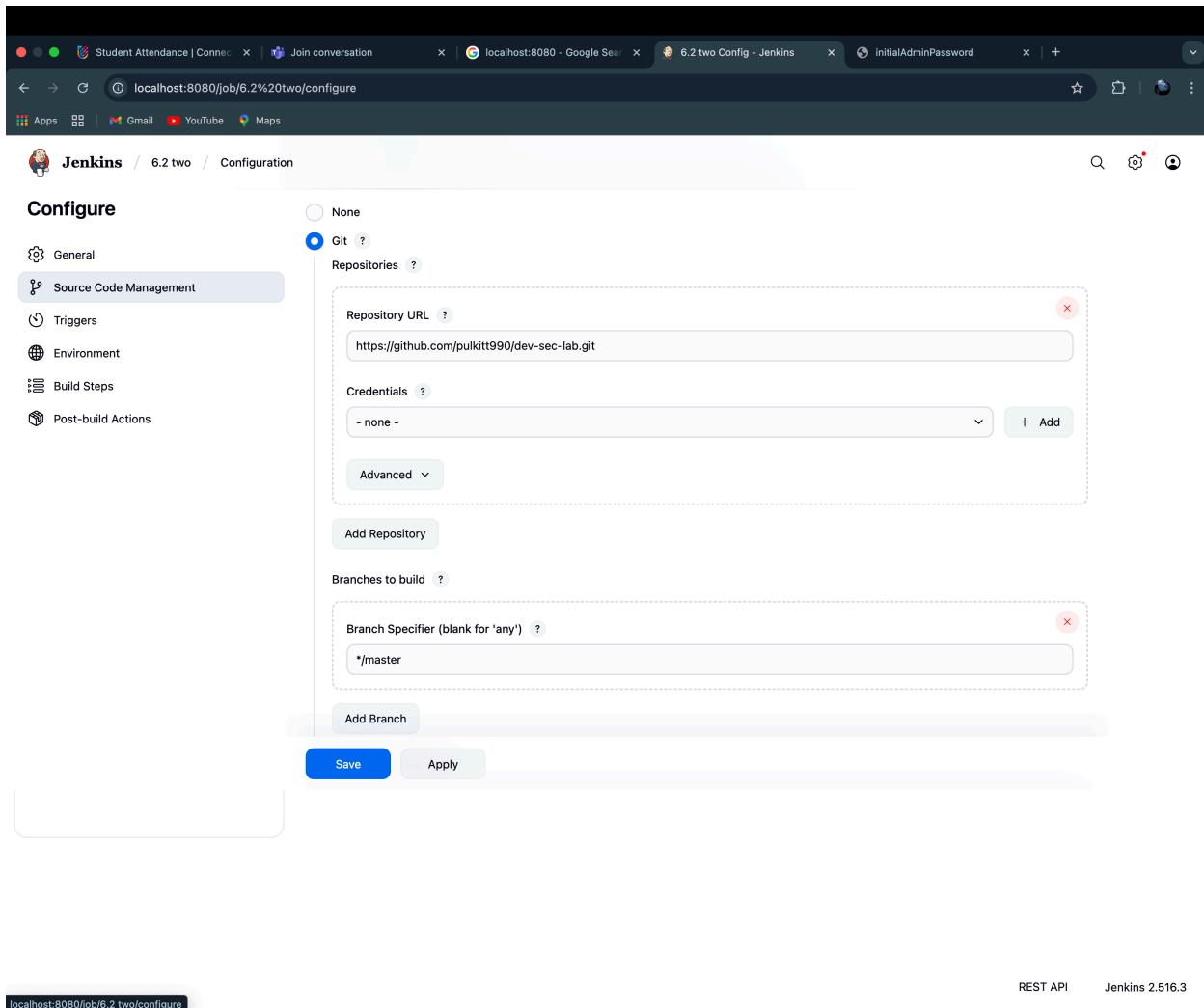
 **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

 **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.

 **GitHub Organization**
Creates a GitHub organization (or user account) for all repositories matching some defined markers.
OK

Step 2: Configure source code management

1. Access the newly created job's configuration screen by clicking on **Configure**



2. Navigate to the **Source Code Management** tab, provide Git repository configuration inside the **Repository URL** field, and click on the **Save** button

3. Then, click on the **Build Now** option to schedule a build

The screenshot shows the Jenkins interface for the 'part 2' job. The top navigation bar includes tabs for 'Student Attendance | Connect', 'Join conversation', and 'part 2 - Jenkins'. The main content area displays the job configuration with the following details:

- Status:** Green checkmark icon, indicating the job is healthy.
- Changes:** Shows a recent change.
- Workspace:** Shows a recent workspace update.
- Build Now:** A button to trigger a manual build.
- Configure:** A link to edit the job configuration.
- Delete Project:** A link to delete the project.
- Rename:** A link to rename the project.
- Credentials:** A link to manage credentials.

Permalinks: A section listing recent builds:

- Last build (#2), 1 min 13 sec ago
- Last stable build (#2), 1 min 13 sec ago
- Last successful build (#2), 1 min 13 sec ago
- Last completed build (#2), 1 min 13 sec ago

Builds: A summary table showing one build entry:

Filter	#2	4:20pm
Today	Green circle icon	4:20pm

At the bottom right, there are links for 'REST API' and 'Jenkins 2.516.3'.

4. To schedule the build, click the required link under **Permalinks**

This screenshot is identical to the previous one, showing the Jenkins 'part 2' configuration page. However, the 'Last build (#2), 2 min 14 sec ago' link under the 'Permalinks' section is now highlighted with a red box, indicating it has been selected or is the target of a click action.

5. Click on **Console Output** to check out the process during the build process

By following these steps, you have successfully set up a Jenkins job to automatically check out source code from a Git repository, enabling seamless integration and automation in your CI/CD pipeline.