

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**



Sign in to Jenkins

Username

mohdanas


Password

.....

☐ Keep me signed in

Sign in

2. Create a new Jenkins job by clicking on **New Item**
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


 / All / New Item 🔍 ⚙️ 👤

New Item


Enter an item name

DEVSECOPS_LAB6.2


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.

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

 / DEVSECOPS_LAB6.2 / Configuration   


Source Code Management

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


☐ None

☒ Git ?


Repositories ?

Repository URL ? 

Credentials ?

- none - 

+ Add

Advanced 

Add Repository

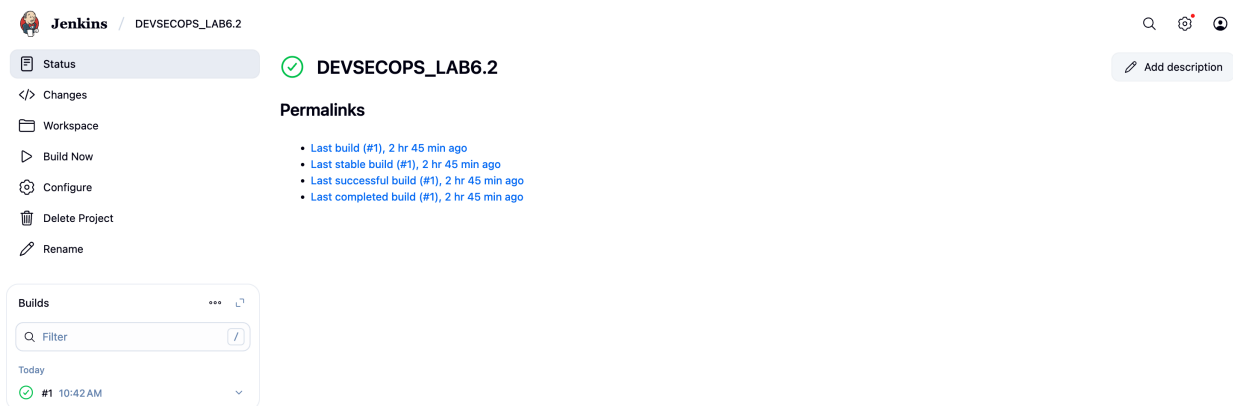
Branches to build ?

Save

Apply

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

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



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.

✓ Console Output

[Download](#)[Copy](#)[View as plain text](#)

```
Started by user Mohd Anas
Running as SYSTEM
Building in workspace /Users/mohdanas/.jenkins/workspace/DEVSECOPS_LAB6.2
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO
> git init /Users/mohdanas/.jenkins/workspace/DEVSECOPS_LAB6.2 # timeout=10
Fetching upstream changes from https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO
> git --version # timeout=10
> git --version # 'git version 2.39.5 (Apple Git-154)'
> git fetch --tags --force --progress -- https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 04a6890beda18d9a0604c2a45835912c4cdb966b (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 04a6890beda18d9a0604c2a45835912c4cdb966b # timeout=10
Commit message: "div() added"
First time build. Skipping changelog.
Finished: SUCCESS
```