Lab: Configuring global tools

The goal of this exercise is to install two different versions of Maven:

- Maven 3.5.4
- Maven 2.2.1

Install two versions of Maven

From your lab's home page, select **Jenkins (LTS version)** to access your Jenkins controller. Sign in using the following credentials:

- Username: **butler**
- Password: **butler**

Go to **Jenkins Manage Jenkins → Global Tools Configuration** and scroll down to the **Maven** section near the bottom of the page:

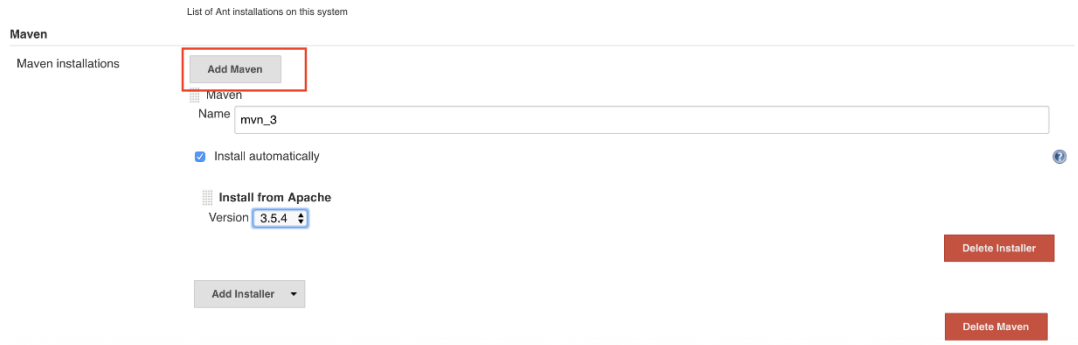

Figure 1. Install Maven Versions

To configure the Maven 3 version:

- Under **Maven Installations**, click **Add Maven** to install Maven.
- In the **Name** field, enter `mvn3`.
- Make sure the **Install automatically** box is checked.
- In the **Version** field, enter `3.5.4`.

To add the Maven 2 version:

- Click **Add Maven** to install another Maven version.
- In **Name**, enter `mvn2`.
- Make sure the **Install automatically** box is checked.
- In **Version**, enter `2.2.1`.
- Click **Save** at the bottom of the screen to save the changes.

You have now installed two versions of Maven.

Create a Maven build project

The goal of this lab exercise is to configure a simple Maven build job on your Jenkins server.

Step 1. Create a new build job

To create a new build job, select **New Item** from the menu in the left frame. Name this project `pipeline-job`, select **Maven project**, and then click **OK**.

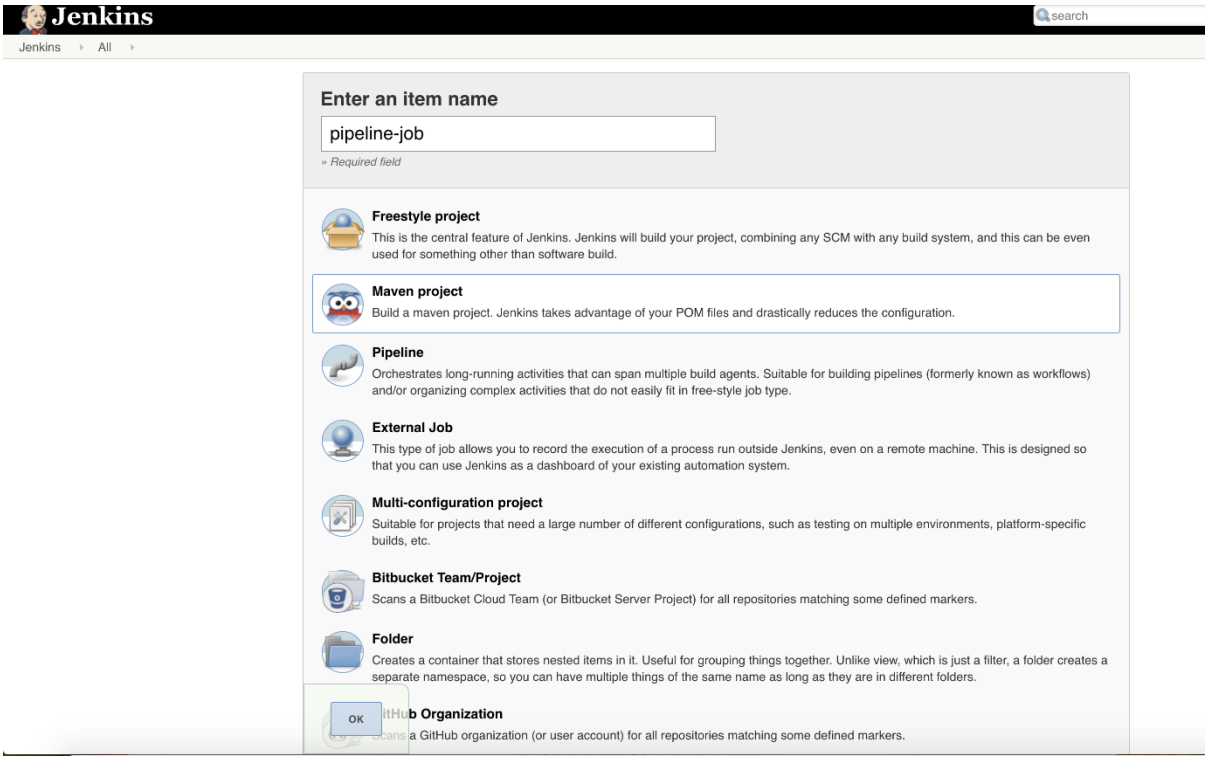

Figure 2. Create Pipeline Maven Job

Step 2. Configure the SCM details

When you click **OK**, a configuration page is displayed for this build job. Scroll down to the **Source Code Management** section (or click the **Source Code Management** tab at the top of the screen to jump to that section) and select **Git**. This opens a window where you can configure the SCM access for your build:

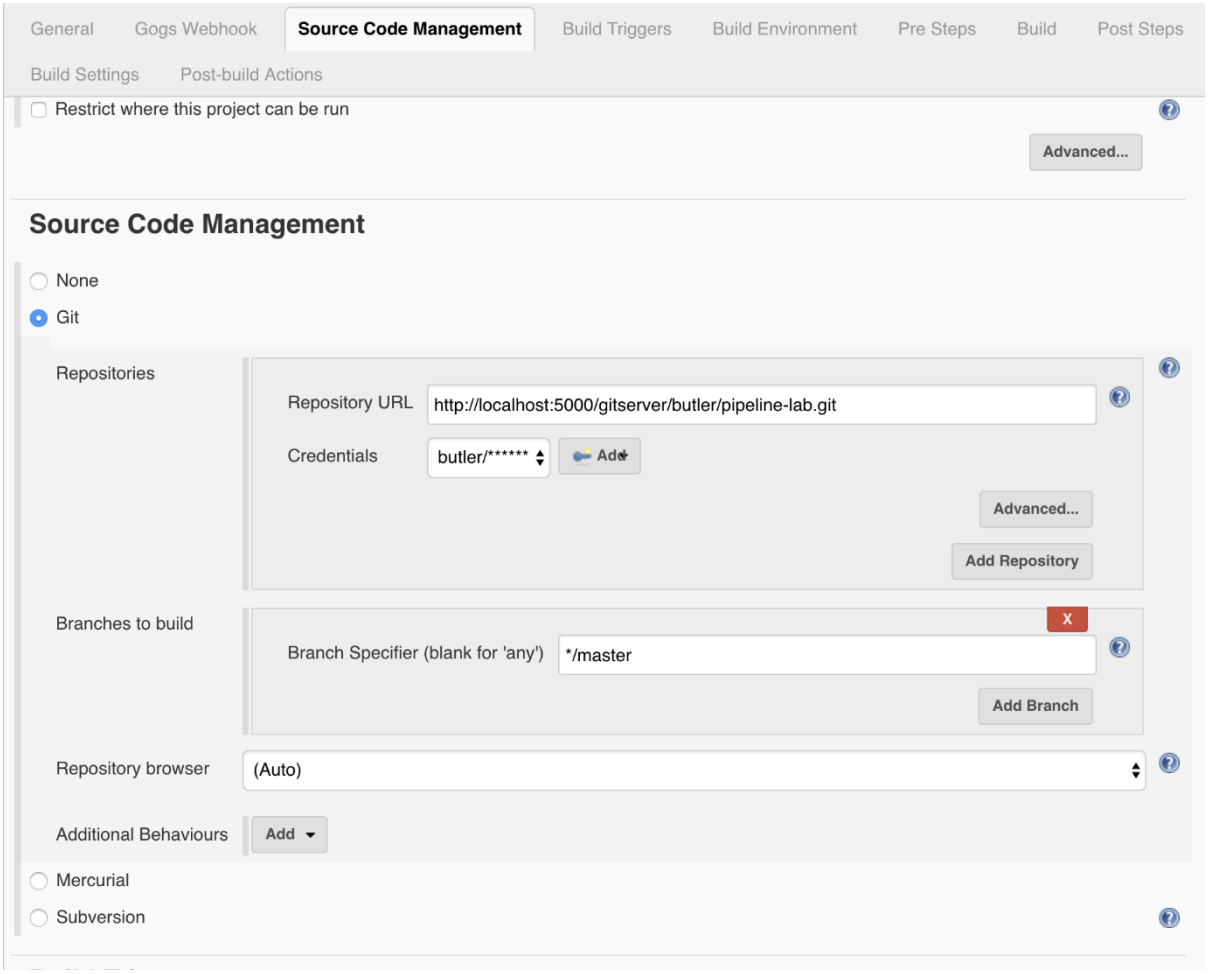

Figure 3. Configure SCM

- In the **Repository URL** field, enter your Git repository address. To get this address:
  - Return to your lab's home page.
  - Right-click on **Gitserver** (A local Git service) and sign in.
  - This opens a page like the following that has links to the repositories on this server:

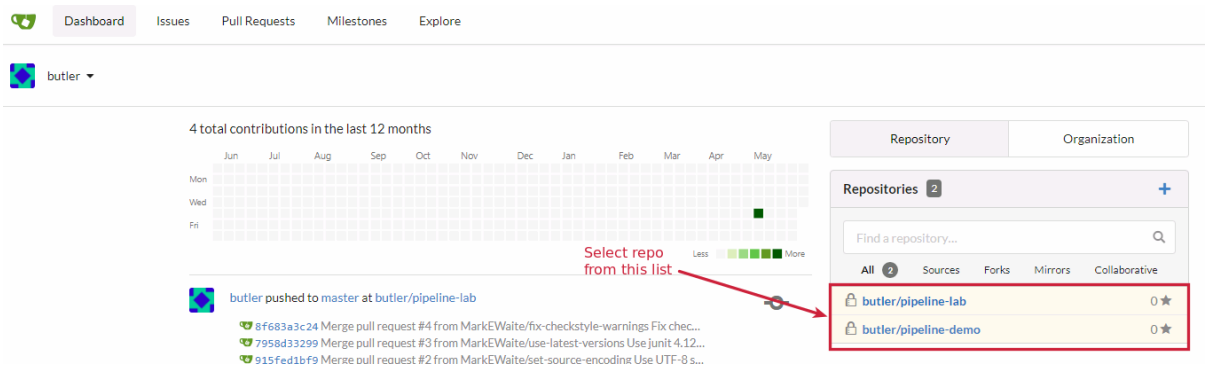

Figure 4. Configure SCM

  - Select the pipeline-lab repo.
  - Copy the HTTP URL for that repo:

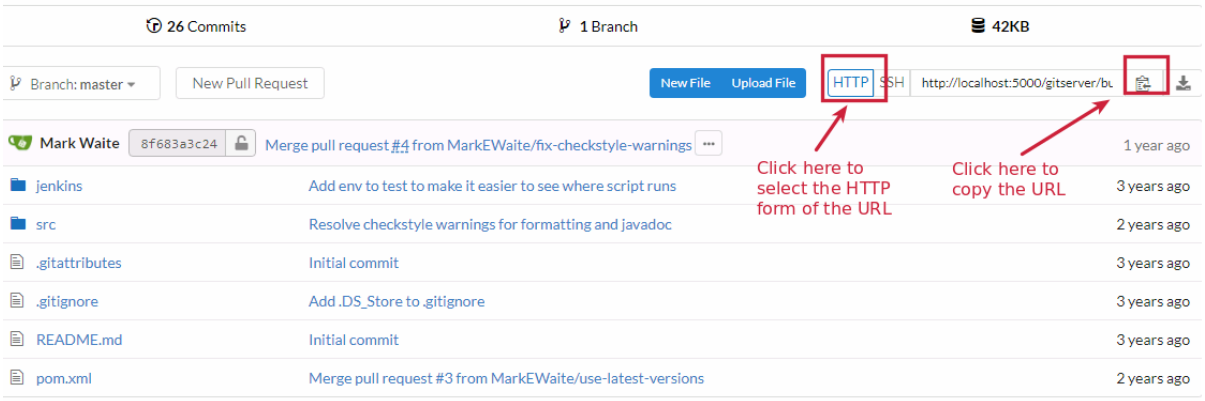

Figure 5. Configure SCM

- Go back to the **Configure SCM** section for your build job and paste the URL into the **Repository URL** field.

This repository needs the `butler` credential to authorize access to this repository. If you had already defined this credential (we will learn how to do that later in this class), you could just select it from the drop-down menu. For this exercise, we will instead add the credential here:

- Select **Add Jenkins** next to the **Credentials** field.
- This displays a screen that you can populate:

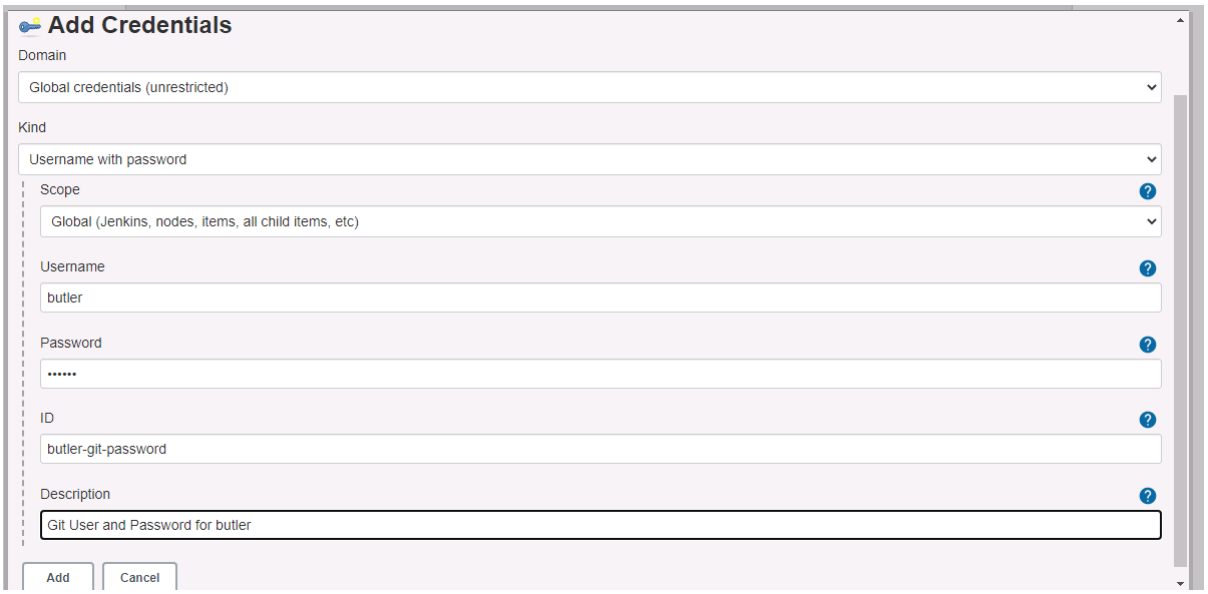

Figure 6. Configure SCM

- Use the following properties:
  - **Kind**: `Username with password`
  - **Scope**: `Global` (this allows it to be reused for other jobs)
  - **Username**: `butler`
  - **Password**: `butler`
  - **ID**: `butler-git-password`
  - **Description**: `Git User and Password for butler`

- When all fields are filled in, select **Add** to add the credential.
- Select the `butler` credential from the drop-down list.
- Leave the other fields set to their default values for now, although you may want to quickly look at the **Additional Behaviors** that you could specify. These are beyond the scope of this class but you should be aware that they exist.

Step 3. Configure the build goals

Scroll down to the **Build** section and select `mvn3` for **Maven Version**.

Finally, you need to configure the Maven build goals in the **Goals and options** field: `clean install`

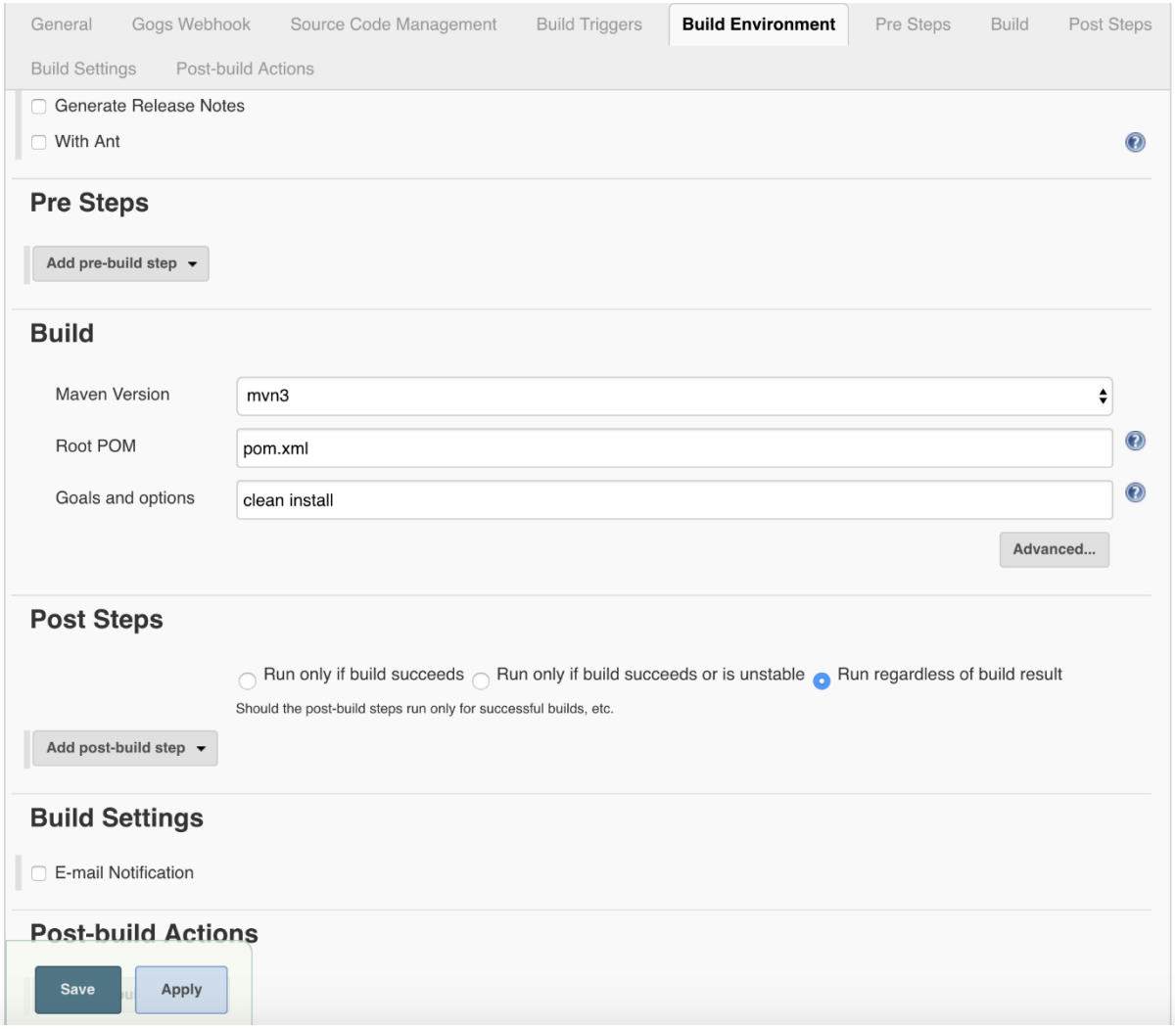

Figure 7. Specify Maven Goals

- Now click **Save** to save this project.
- Click **Build Now** to start a new build.

Go back to the dashboard to watch the build in progress.

That is all for this exercise.