**A Step-By-Step Guide On How To Integrate Jenkins With Selenium**

## Integrating Jenkins With Selenium

For many developers, this may imply that their work could be at risk or unreliable without automated testing. Together with an open-source automation server Jenkins and popular Web testing framework Selenium constitute potent tools in handling of testing automation. In this article I’ll guide you, step by step, how to integrate and use Jenkins with Selenium, for a smooth continuous integration and testing process flow.

### Prerequisites:

Before diving into the integration, ensure you have the following prerequisites in place:

**Jenkins Installation:**You must install and run Jenkins on a system. You can download it from the official website (<https://www.jenkins.io/download/> ) and follow instructions given therein.

**Selenium WebDriver**: You should also install Selenium WebDriver on your machine since it is a browser automation tool. You can install it using a package manager or download it from the official Selenium website (https: //www.selenium.dev/downloads/).

**Java Development Kit (JDK):** Both Jenkins and Selenium are in Java. Ensure that the appropriate JDK is loaded onto your system.

Now that you have the prerequisites in place, let’s proceed with the integration:

## Step 1: Install Jenkins Plugins

The functionalities of Jenkins are enhanced using numerous plugins. To work with Selenium, you’ll need to install a couple of plugins:

**Selenium Plugin:** Go to Jenkins Plugin Manager and install “Selenium Plugin”. Use this plugin for running Selenium tests on a Jenkins pipeline.

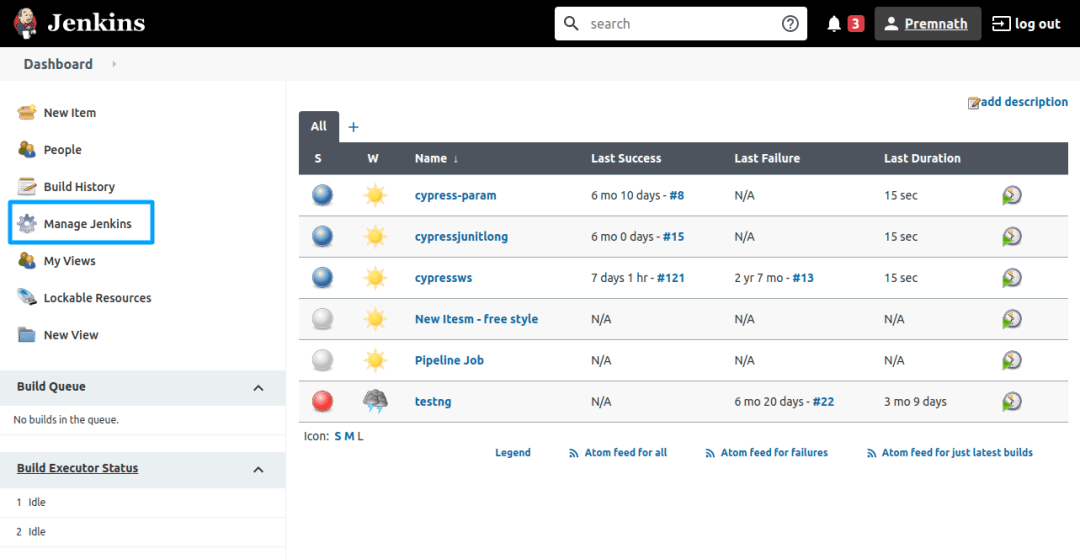
**WebDriver Automation Plugin:** Install the “WebDriver Automation Plugin” to enable Selenium WebDriver integration with Jenkins.

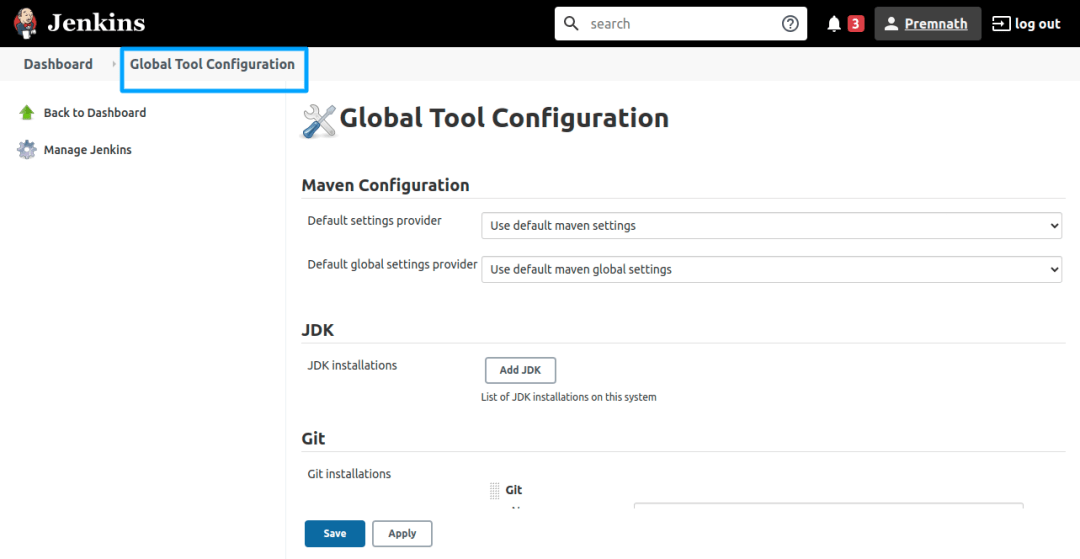
## Step 2: Configure Jenkins

Once the plugins are installed, you need to configure Jenkins:

Open Jenkins in your web browser.

On the left sidebar, click “Manage Jenkins”.

Navigate to “**Global Tool Configuration**.”

Under “**Selenium WebDriver**,” add the path to your Selenium WebDriver executable.

Save the configuration.

## Step 3: Create A Jenkins Job

To automate Selenium tests using Jenkins, you need to create a Jenkins job:

Navigate to the ‘New Item’ option on the Jenkins dashboard.

Give your job a name and select “**Freestyle project**.”

Under the “**Build**” section, click on “**Add build step**” and select “**SeleniumHQ htmlSuite Run.**“

Configure your Selenium tests by providing the path to your test suite, browser settings, and other relevant details.

Save the job configuration.

## Create a Jenkins Job

## Create a Jenkins Job 1

## Step 4: Configure Jenkins Build Triggers

To trigger your Selenium tests automatically, you can set up build triggers. The most common trigger is to run the tests when code is pushed to a specific branch in your version control system (e.g., GitHub). To set up GitHub webhook triggers:

In your Jenkins job configuration, go to the “**Build Triggers**” section.

Check the option “**GitHub hook trigger for GITScm polling**.”

Save your job configuration.

## Configure Jenkins Build Triggers

## Step 5: Build And Test

With everything set up, you can now build and test your Selenium integration with Jenkins:

Click on your Jenkins job from the dashboard.

Click “**Build Now**” to start a manual build, or wait for automatic triggers to initiate the tests.

Jenkins will execute your Selenium tests and display the results in the job console output.

## Build and TestConclusion

By integrating Jenkins with Selenium, you will have a seamless testing process in your software development life cycle, making it efficient and automated when it comes to quality assurance. Through this article, we have shown you how this integration can be successful through various important ways. How do you automate your web testing? Set up Jenkins, install needed plugins, configure your environment, create jenkins jobs for web application reliable. Selenium when merged with Jenkins makes it the ultimate tool in Continuous Integration and Continuous Testing (CI/CT), thereby saving you time and reducing the hassle involved in delivering high quality software. Join The AutoMation Journey With Jenkins & Selenium Today!