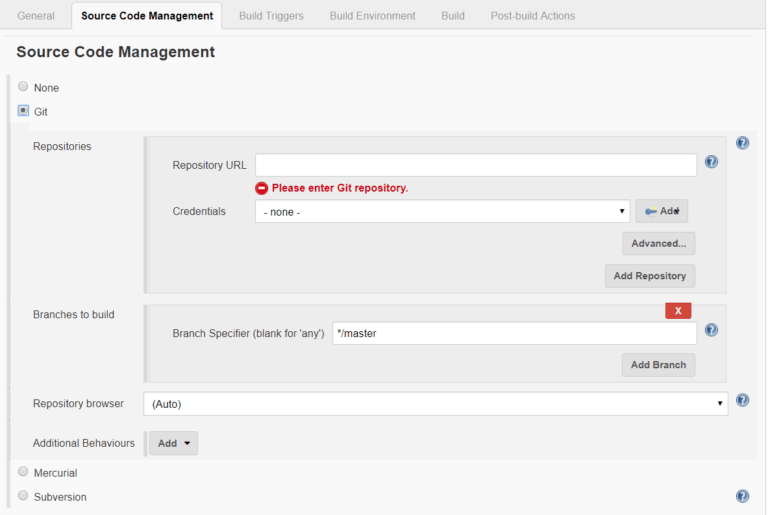
**Working with CronJobs**

1. Let's scroll to *Source Code Management* section. Upon selecting the radio button next to *Git*, a new section appears labeled as *Repositories*.

**This is where we need to configure details of our SCM repository**. Let's type the URL of the SCM repository in the *Repository URL* text field:

Update Repo URL as **https://github.com/chaitanyagaajula/simple-java-maven-app.git**



2. **3. We also need to provide user credentials so that Jenkins can access the repository.**

Let's click the *Add* button next to Credentials, which will display a pop-up screen to create the user credentials.

Let's select the *Kind* as *Username with Password*. We'll have to type the username and password in the designated text fields:

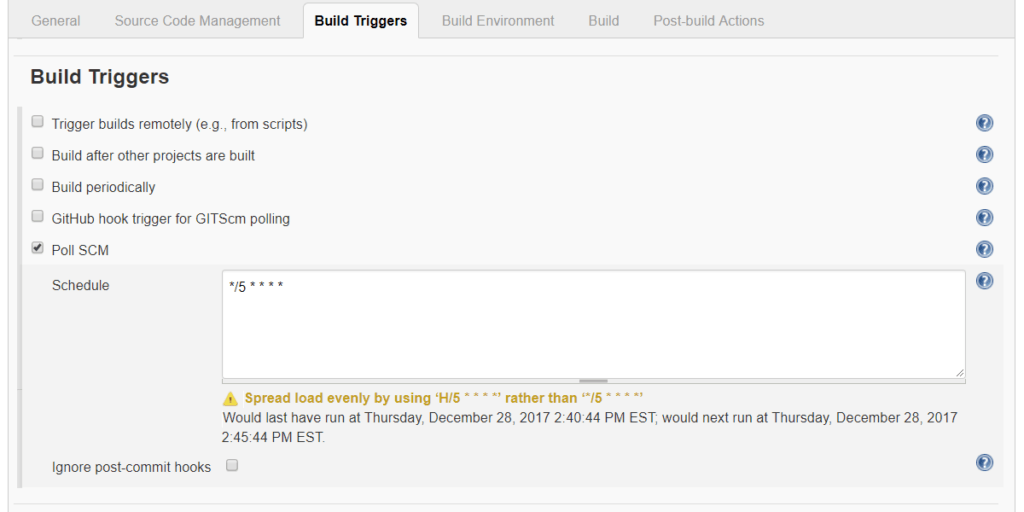


**3. Creating a Job That Polls SCM**

Let's create a new job as explained in the previous section, with a few modifications.

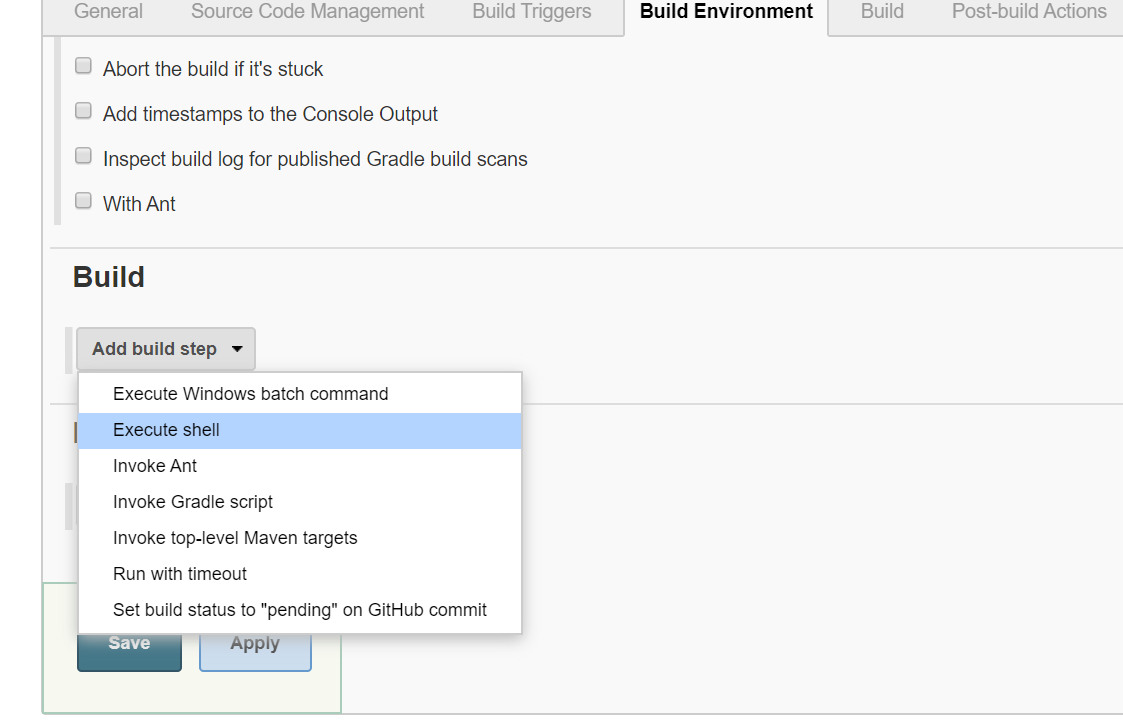
In the *Build Triggers* section, **instead of selecting *Build Periodically*, let's select *Poll SCM***. As soon as we do that, we should see a text box with Label *Schedule*.

Let's type *\*/5 \* \* \* \** in this box, which means we want to schedule the job to run every 5 minutes:



4. Now, the last thing we need to do is to set up the build script.

Let's scroll down to *Build* section, click *Add build step* and select *Execute Shell*. Since we're working on a Maven project in the SCM repository, **we need to type *mvn clean install*, which will perform a Maven build.**



Let's try to understand what we've done here.

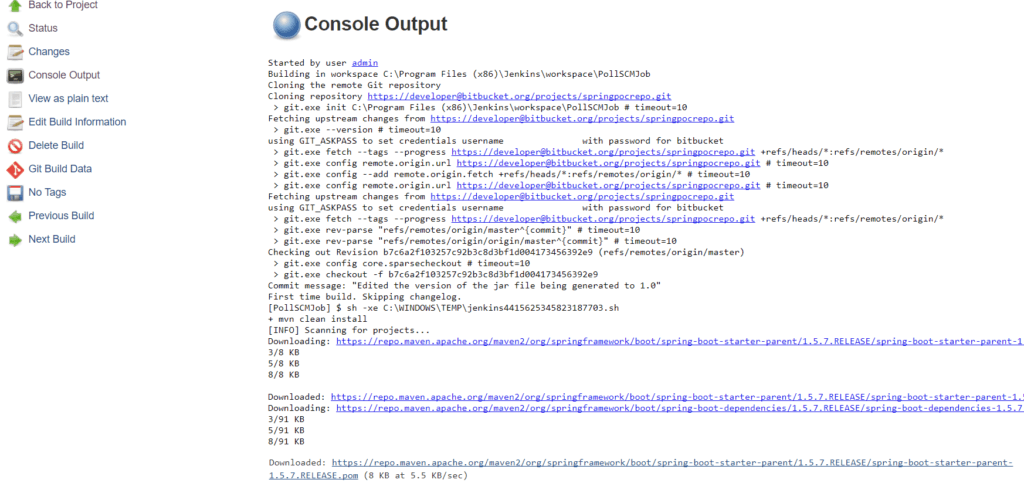
We've created a job that is scheduled to run every 5 minutes. **The job's been configured to pull source code from the master branch of the given github repository**.

It will use provided user credentials to log in to github.

**After pulling the source code, the job will execute the script containing a provided Maven command.**

Now, if we save and wait approximately five minutes, we should see the build execution in *Build History* section on the job dashboard.

The *Console Output* should show the output of the Maven build. We can see in the console output that the source code has been pulled from Bitbucket and the command *mvn clean install* has been executed:

[](https://www.baeldung.com/wp-content/uploads/2018/01/maven-console-output-1-1.png)

Since it's a Maven build, **the console output might be very long depending on the number of Maven dependencies that are downloaded**.

But at the end of the output, we should see *BUILD SUCCESS* message.