Building Ruby on Rails Applications with Jenkins

This guide is meant to be a step by step walk through to create a build job in Jenkins running on a Linux server. That will work to continuously build a Ruby on Rails application. It will also expect a git server more specifically a private Gitlab server hosted at MSU Denver.

Installation of Jenkins

First you have to install Jenkins. There are many great guides on the internet for many common operating systems. To install on ubuntu you can refer to this guide https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+on+Ubuntu

Likewise to install on a RHEL or similar distribution you can refer to this guide https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+on+Ubuntu

Installing Necessary Plugins

There are two plugins that are necessary for the Jenkins server to communicate with the Gitlab server at MSU Denver.

1. Log into Jenkins
2. From the main page click the Manage Jenkins button on the right hand side of the screen.
3. Then select the Manage Plugins button from the next page.
4. Click the Available tab at the top of the list.
5. Find the Git Plugin and check the box.
6. Find the EnvInject Plugin and check the box.
7. Now click the Download and Install button.
8. Wait for the two plugins to be installed and restart Jenkins if it doesn't auto restart.

Creating The Build Job

Jenkins Jobs are what is used to build the application. We will need a specific job set up for the project.

1. Log into Jenkins.
2. From the main page click the new item button on the left hand side of the screen.
3. Now you can select Freestyle project
4. Give it a meaningful name in the Item name text field.
5. Click the OK button near the bottom of the page. This should take you to the configuration page for the job.

Configuring the Build Job

All the configuration for this job will occur on the configuration page. First, you must connect Jenkins to the Gitlab Repository and provide it access to the code. Next, to automate the process it is very easy to set up Jenkins to poll the Gitlab server for changes. If changes are detected then it will build the project. Finally, you have to set a Environment Variable to be used during the build process.

Connecting to the Gitlab Repository

1. Find the Source Code Management section on the page and select Git.
2. For the Repository URL you have to enter the full URL to the project repository on the git server.
3. Click the Add button next to the credentials drop down list. You need to enter in credentials for a user account on the Gitlab server that the Jenkins server should use.
    a. Using a username and password is very easy to enter.
4. Now select the credentials you just created from the drop down list labeled Credentials.
5. You probably don't want Jenkins to build each and every branch of code in the projects repository. In the Branches to build text field enter the name of the branch you would like built by Jenkins. You can add multiple branches by clicking the Add Branch button under the text field.

Jenkins Can Check for Changes to the Gitlab Repository

1. Find the Build Triggers section right below Source Code Management section.
2. Click the checkbox for Poll SCM.
3. A big box will appear labeled Schedule. This box is where you will enter a schedule for the Cron Job that will run to perform the polling of the Gitlab repository. To set up the CI server to poll every 5 minutes you can enter the following into the box. There are spaces between every symbol except the H/5.
   H/5 * * * *
   To learn more about what that means above you can click the help dialog button next to the field in Jenkins.

Setting an Environment Variable

1. The final configuration for the Job is a environment variable that Jenkins will need to have in order to access the code. Find the Build Environment section.
2. Check the Inject environment variables to the build process checkbox.
3. In the Properties Content text area enter the following. GIT_SSL_NO_VERIFY=1

Create Build Script

The last step is to create a Build Script for this project. This Build Script is a shell script that when executed will perform all the steps necessary to build and execute the tests on this Rails application.

1. Find the Build section right underneath the Build Environment section.
2. Click the Add build step and select Execute shell.
3. In the Command box enter the following.

```
echo "Started Build" $BUILD_NUMBER
source ~/.bashrc
rvm use --create ruby-2.2.0@my_app
rvm --force gemset empty
gem install bundler --no-rdoc --no-ri
bundle install
bundle exec rake db:drop:all
bundle exec rake db:create:all
bundle exec rake db:migrate
bundle exec rake db:test:prepare
bundle exec rake test
```

4. Finally click the Apply button at the bottom of the page.

Now the Build Job is complete. The Jenkins CI server will poll the Gitlab server and your project for changes. Upon detecting changes it will get the newest code from the repository and build then test the Rails application.