**Jenkins Integration with Git (Cucumber and Selenium)**

**Created by : Anjali**

**Initial Steps**

1. Create a repository in Git and commit your code in the branch
2. Raise a Secure request for your jenkin instance -> which means you are requesting for the separate instance of your project in Jenkins and url should be like *https://jenkins.optum.com/modernization/*

*Approaches for running your UI tests through Jenkins*

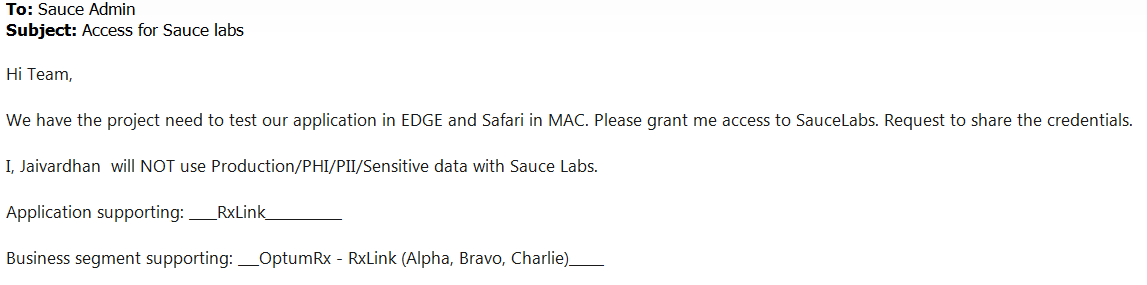
1. Through Docker

We can execute our UI test using Docker with the help of Sauce Lab but in this case either our application should be in public domain or this application should be open in Sauce Lab tunnel ,for ex *http://hub.ugh.com/* is not in public domain and so we cannot automate this using Sauce Lab.

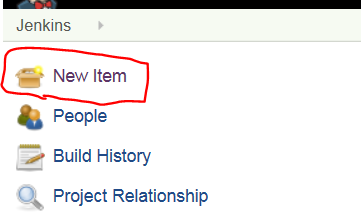
Steps :

1. Raise a request for Sauce lab account

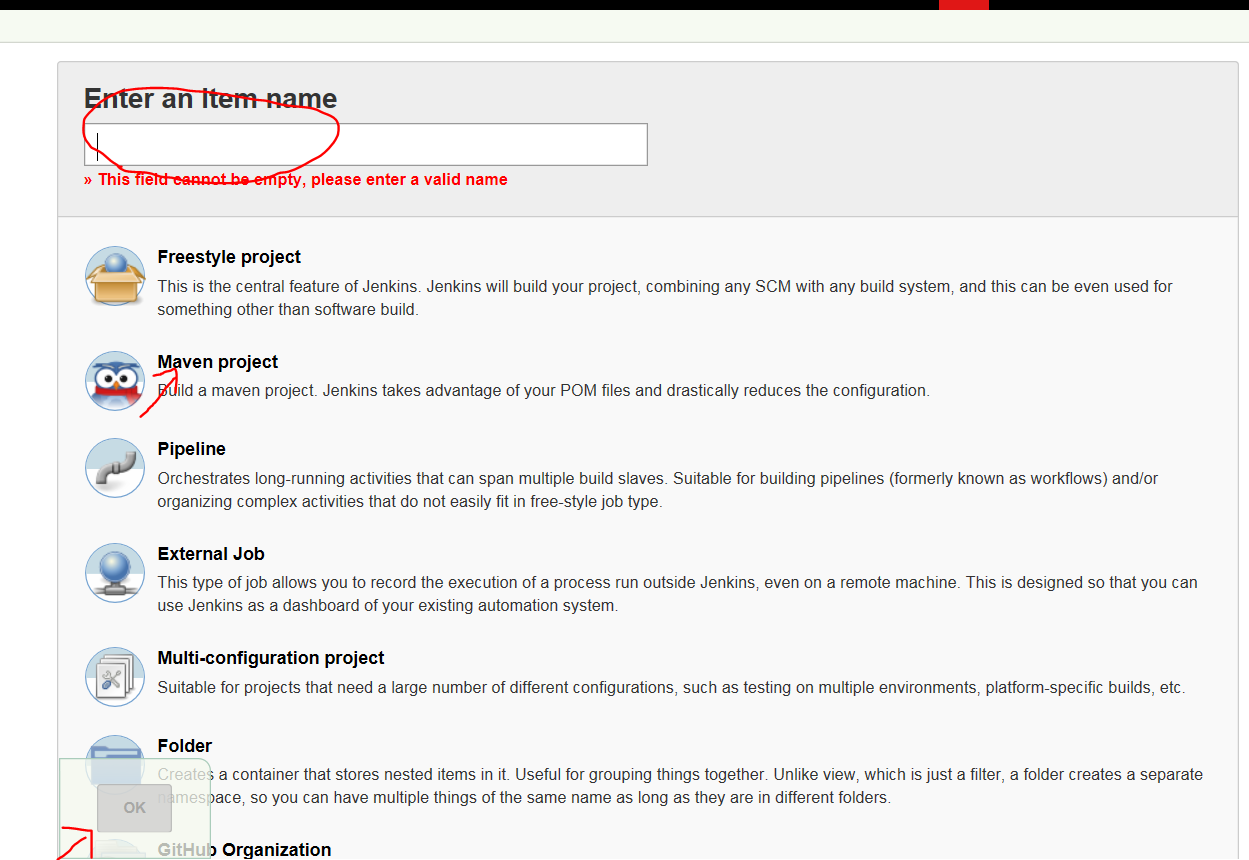
For this you need to send an email to the Sauce admin, see below screenshot for reference



1. Go to your Jenkin’s instance and create a new job by clicking on *New Item*

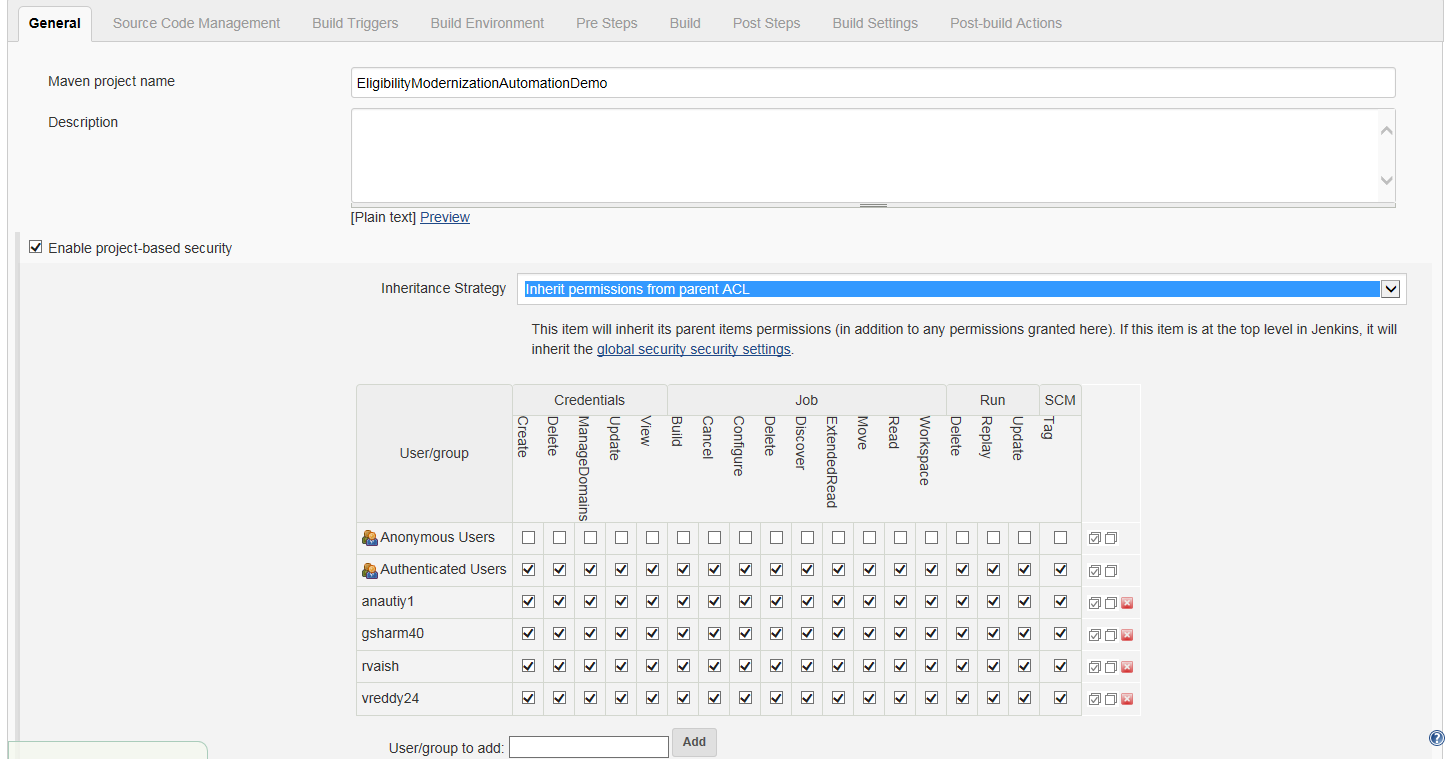


1. Enter the name of your job and click on *Maven project* and click on *OK*

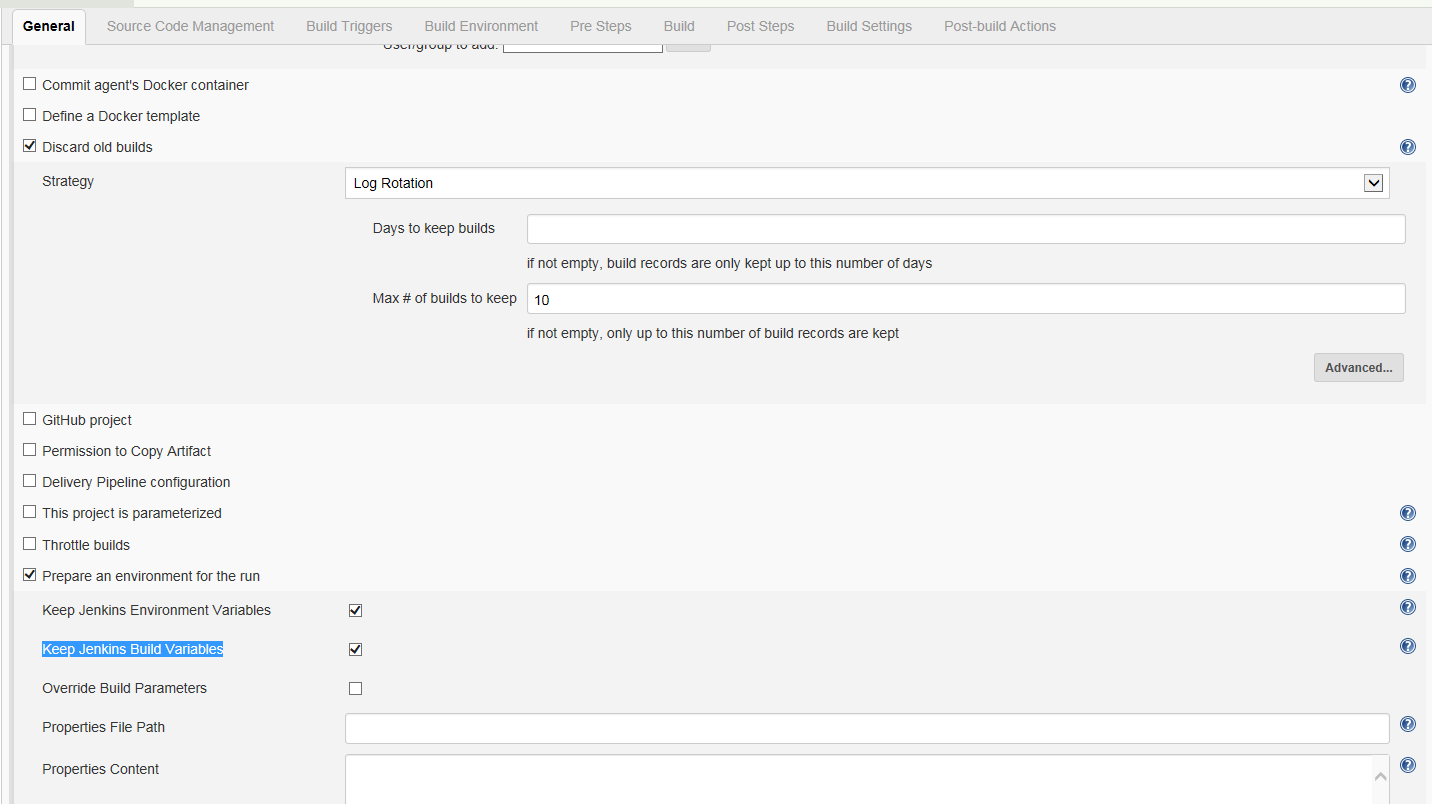


1. Configure your job as mention below

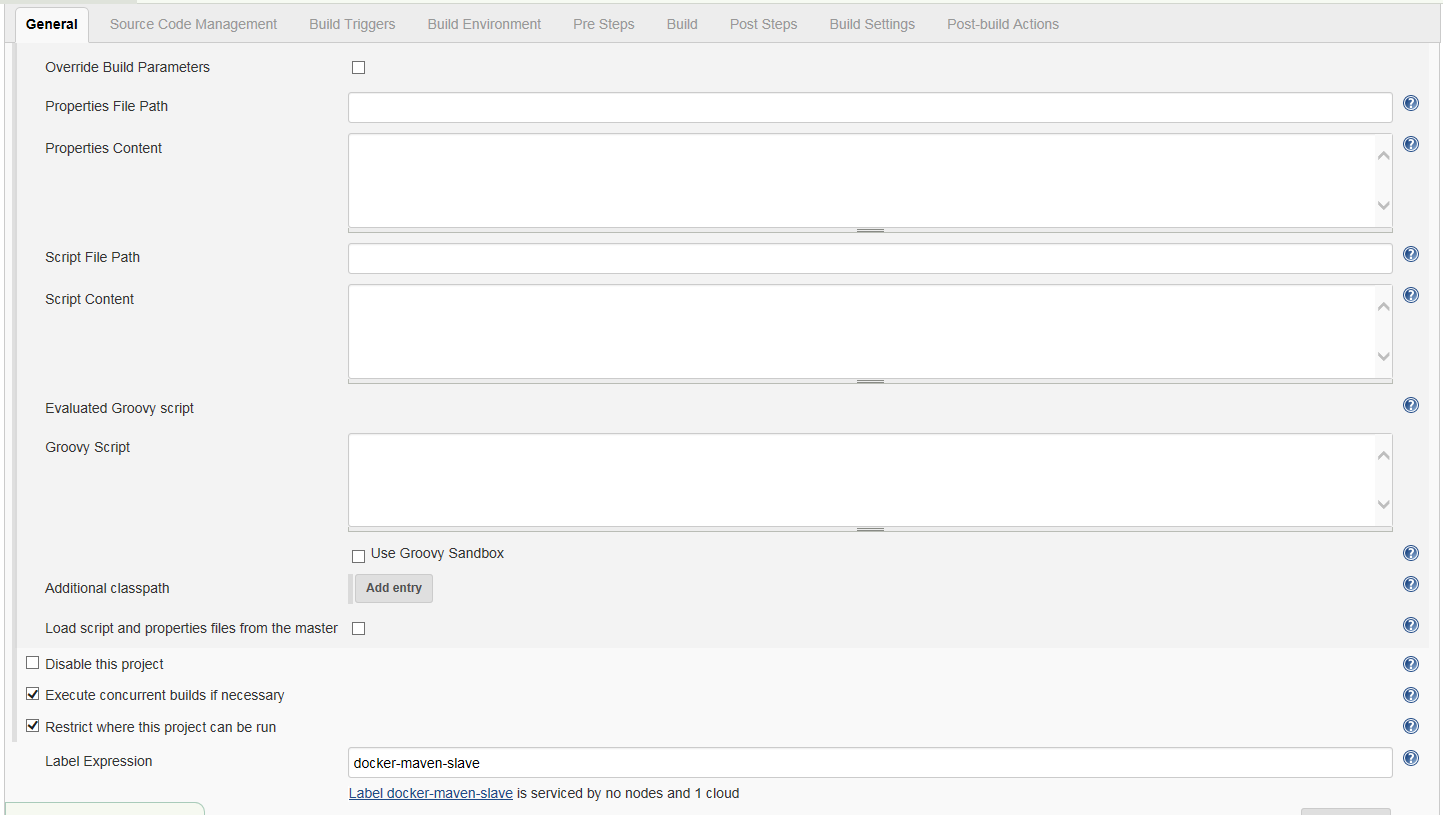
* Maven Project Name – your project or item name
* Check *Enable project base security* and enter the Inheritance Strategy as *Inherit permission from parent ACL*
* Give the permission to the users (as per the project requirement)



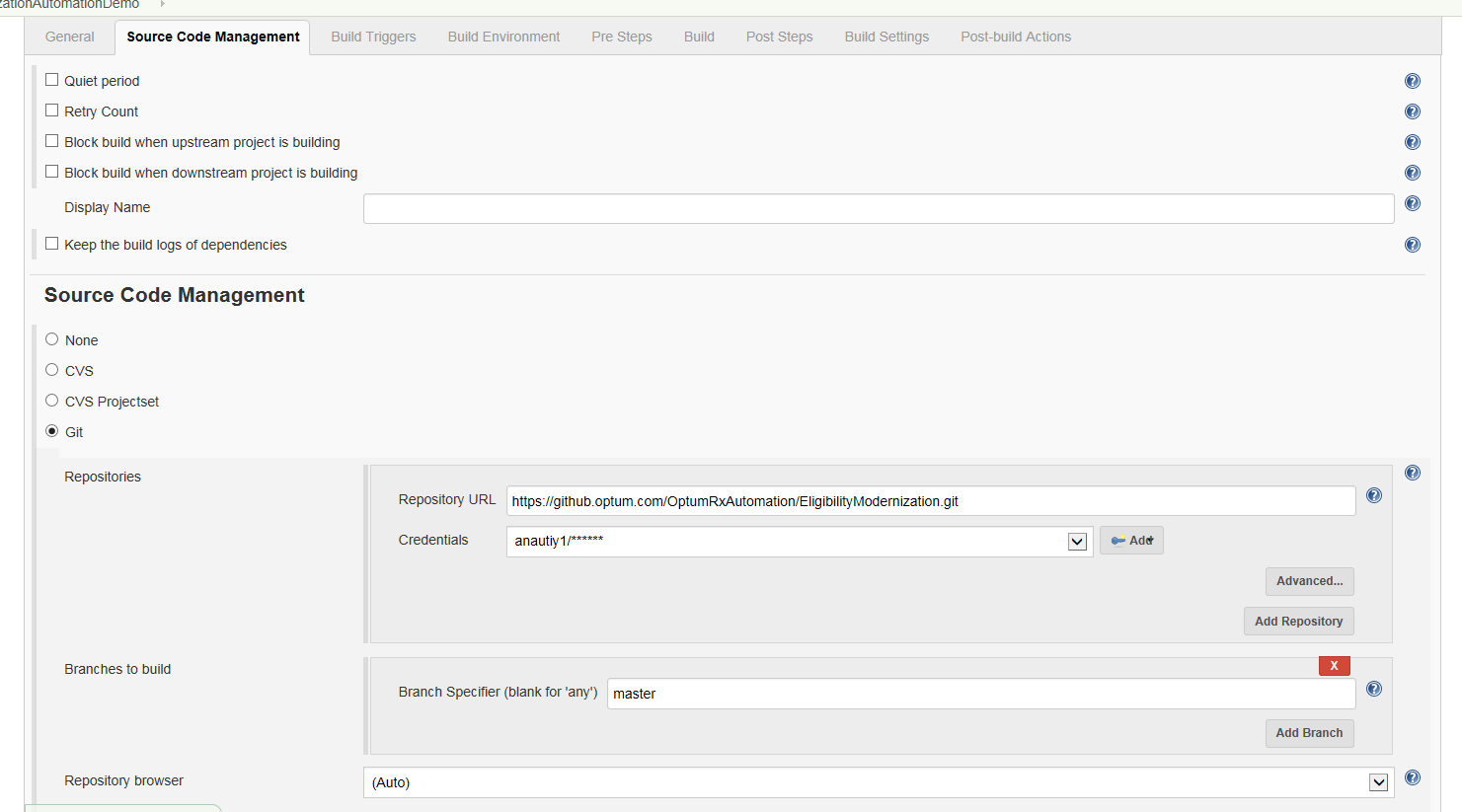
* Check *Discard old builds*, enter strategy as *Log Rotation and Max # of builds to keep* as 10
* Check *Prepare an environment for the run* , *Keep Jenkins Environment Variables and Keep Jenkins Build Variables*



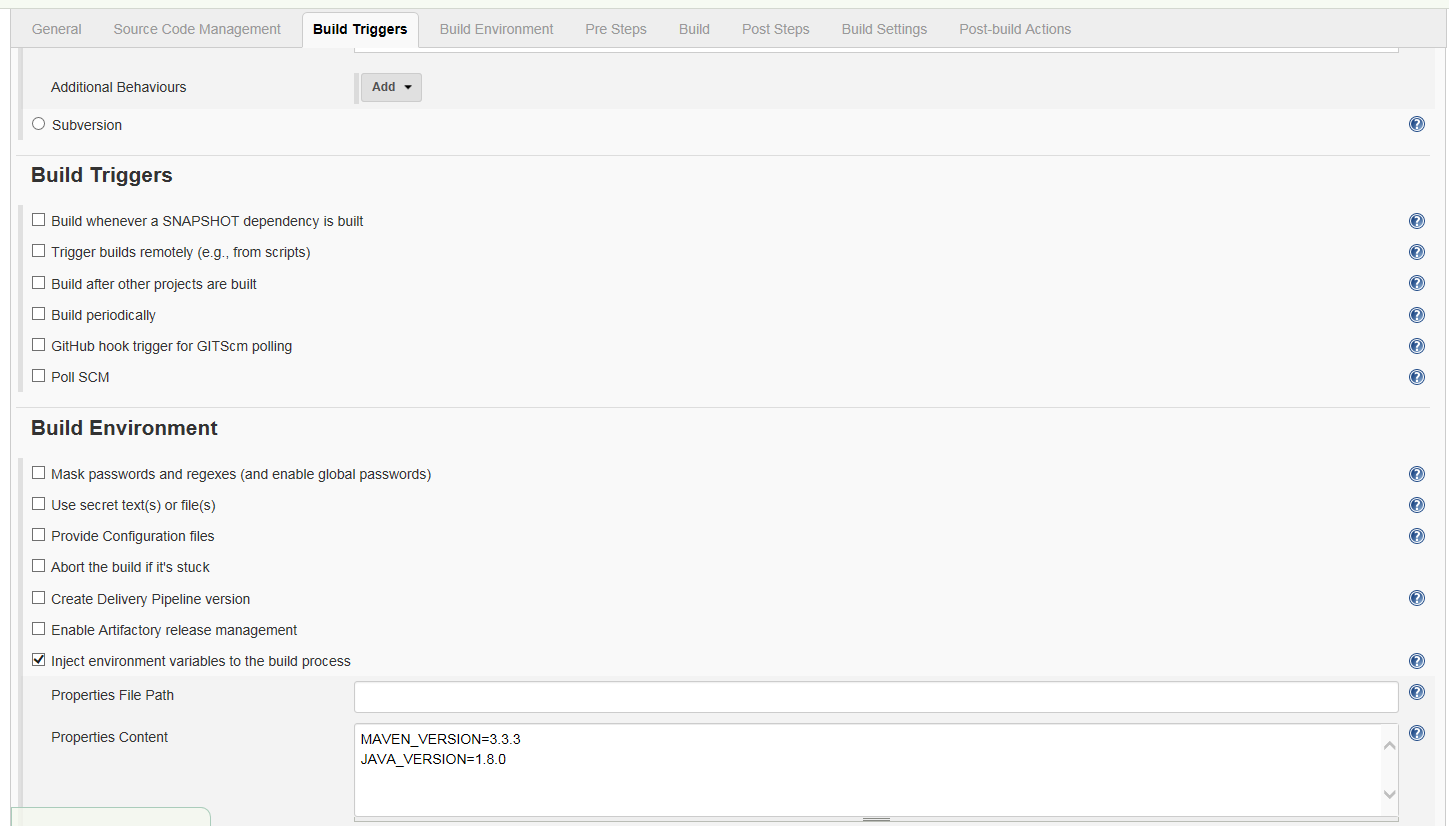
* Check *Execute concurrent builds if necessary* and *Restrict where this project can be run*
* Enter *Label expression* as per your label you mentioned



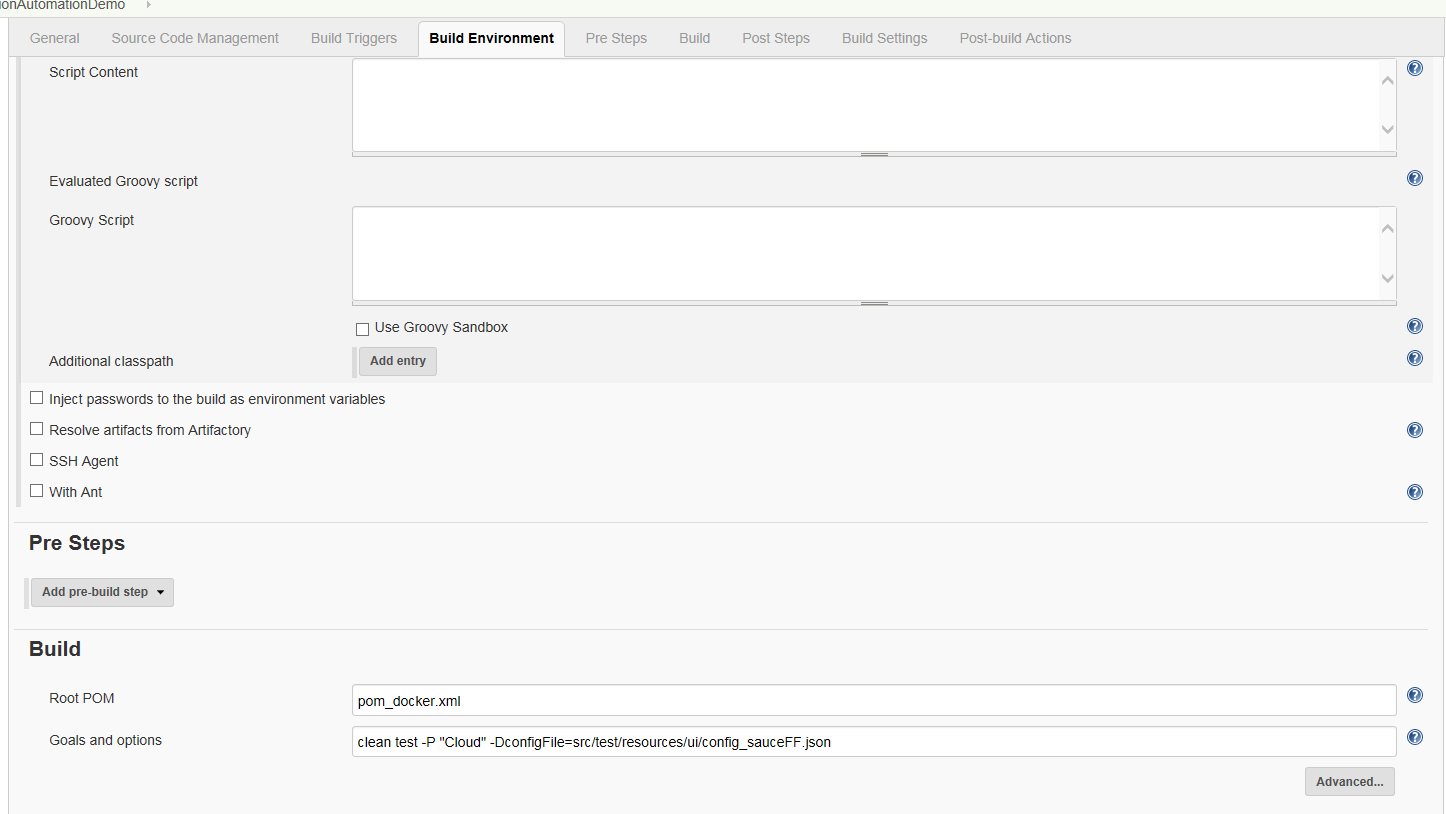
* Check *Git* from source code management and enter your *Repository URL* and *enter the credentials*
* Mention your branch (Where your code deployed in Git) and *Repository browser* should be *auto*.



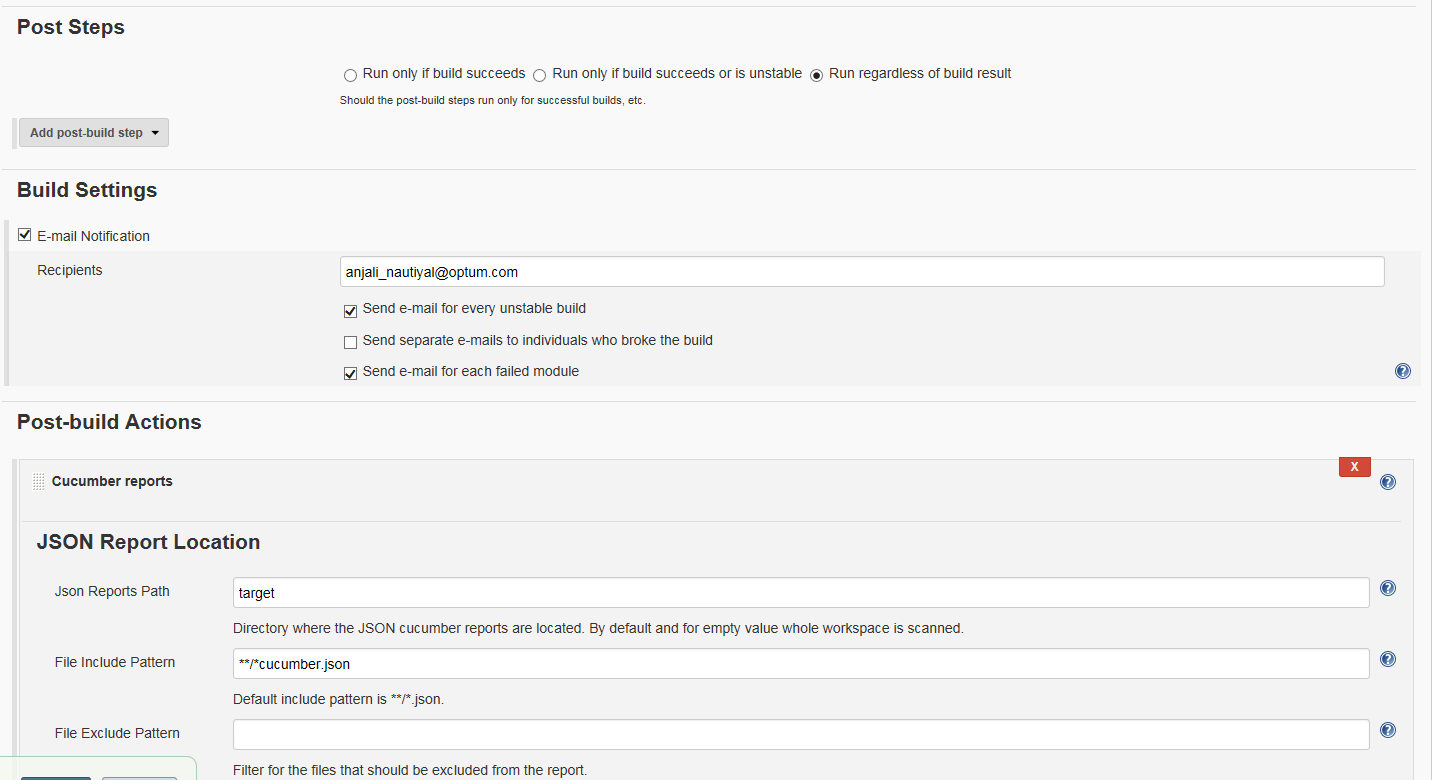
* *Check Inject environment variables to the build process* and add your *Properties Content*



* Mention the *Root POM* and *Goals and option as clean test -P "Cloud" -DconfigFile=****src/test/resources/ui/config\_sauceFF.json*** *(this is path of your sauceFF.json)*



* Check *Run regardless of build result* and enter the email in the *Build setting* (Where you want build result and it is an optional)
* Add *Post build action* (for this you need to add plugins from *Manage plugin* for the reports whatever you want)



* Click on *Apply* and click on *Save*.

Your job has been configured and you can trigger the build by clicking on *Build now* option.

1. Through Virtual Machine

*We can execute our UI test by integrating our VMs (which is in UHG domain) with the Jenkins job.*

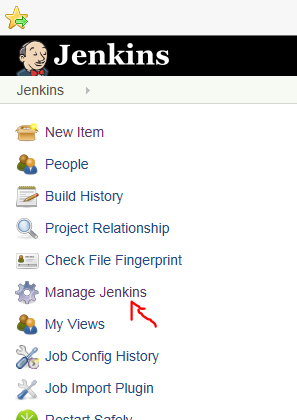
*Please Note : Our VM should be Test Virtual Machine For ex. Java Developer Virtual Machine*

Steps :

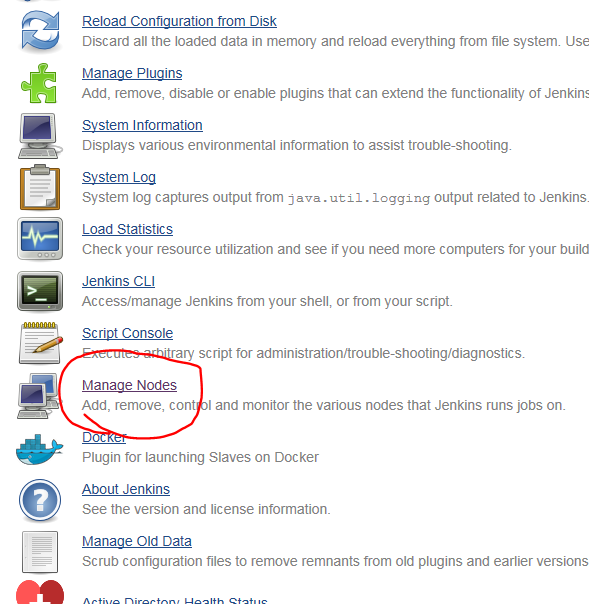
1. Raise a request for Java virtual desktop .
2. Once your VDI will be ready , please be ensure that your VDI should have Git, Maven version and JDK 8 with the path set in your environment variable.
3. Now open your jenkin’s instance and go to the *Manage Jenkins*

Steps for Adding Virtual machine in your jenkin’s instance

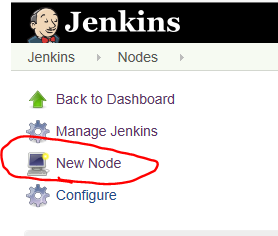
1. Go to *Manage Jenkins.*



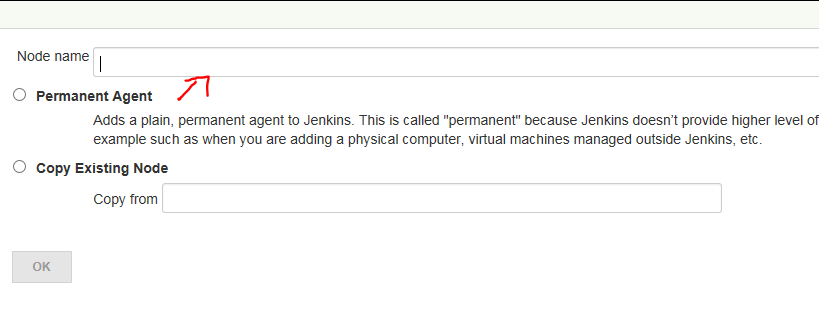
1. Go to the *Manage nodes.*



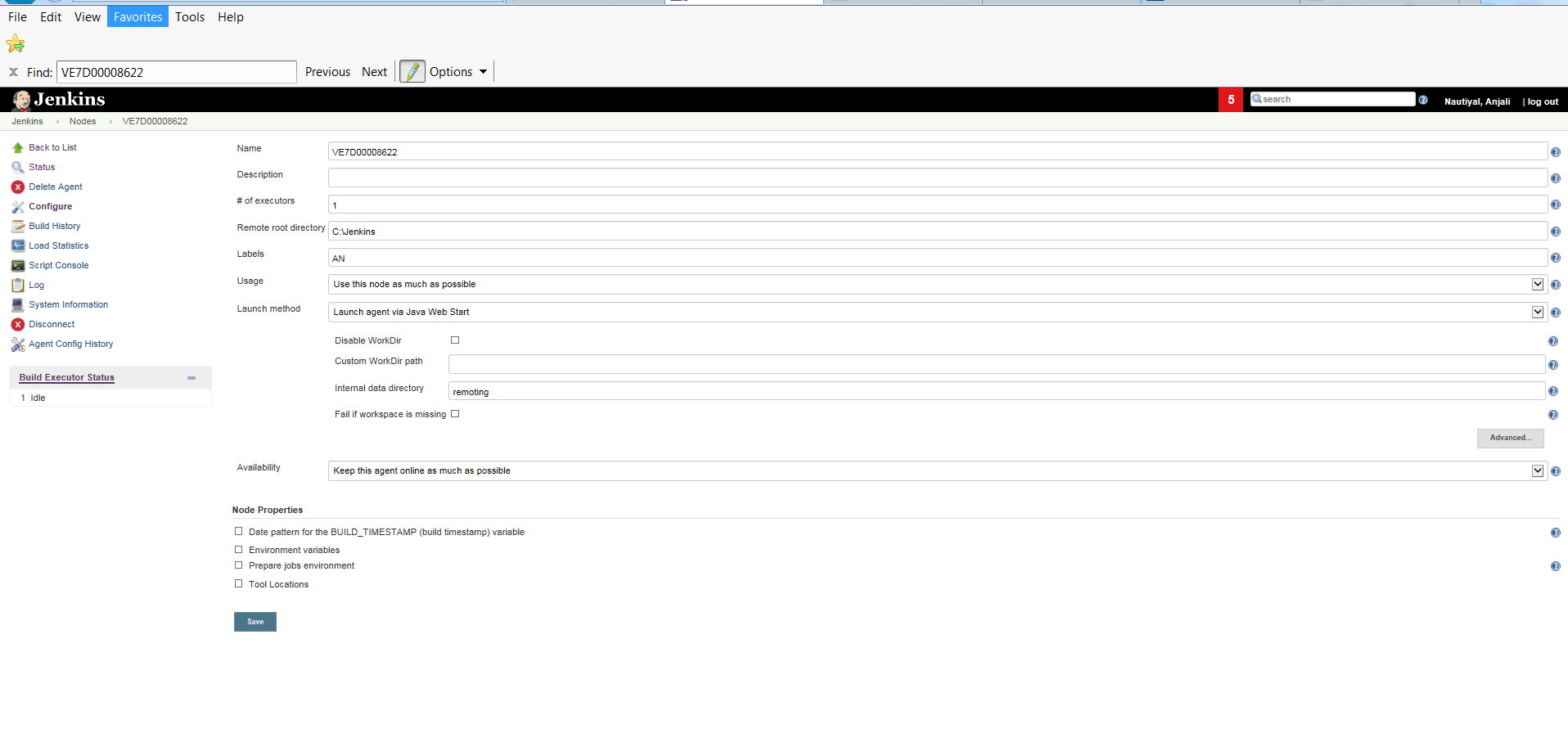
1. Click on *New Node.*



1. Enter the *Node name* (your VM name) and check *Permanent Agent* option.

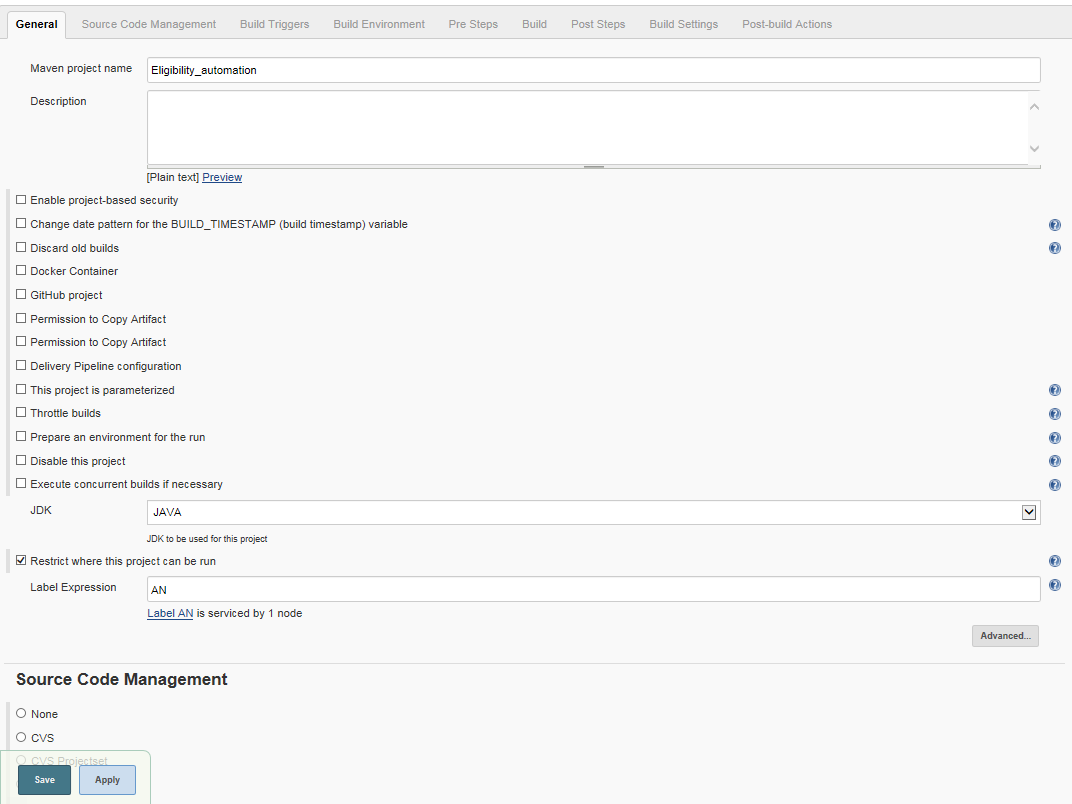


1. Enter the *Name* of your VM
2. Enter the *# of executors*
3. Mention the *Remote root directory*
4. Mention the *Label* of your VM
5. Select Usage as *Use this node as much as possible*
6. Select Launch Method as *Launch agent Via Java Web Start*
7. Select Availability as *Keep data online as much as possible*
8. Click on *Save*

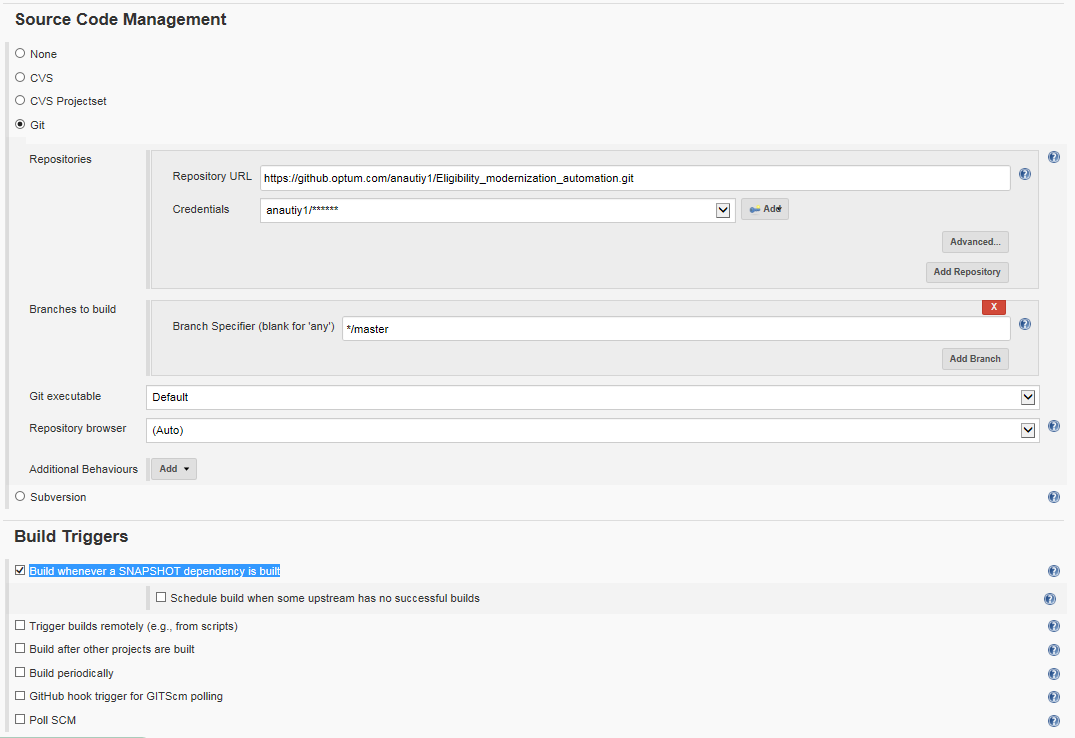


Configure your job by using below steps

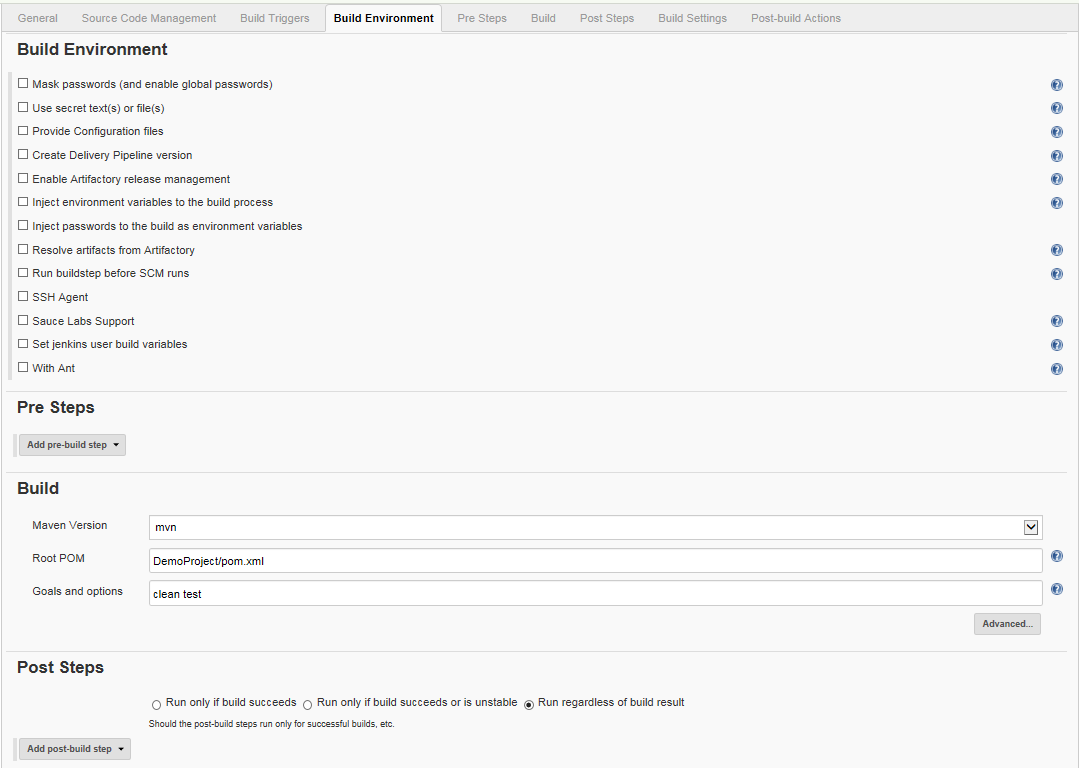
1. *JDK* should be your JDK version name which you mentioned in configuration (Via Configure System)
2. Check *Restrict where the project can* *be run* and mention your *VM Label Name* where you want to run this particular job



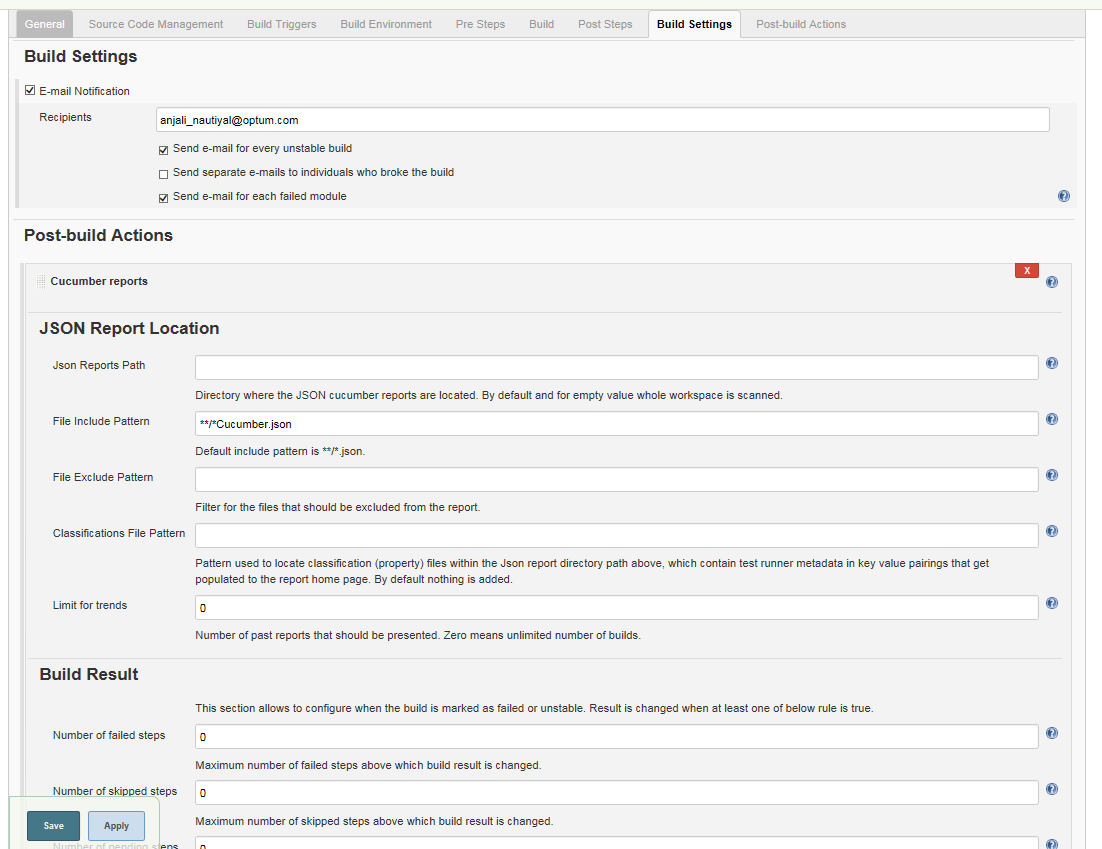
1. In the *Source code management* ,select *Git* and mention the *Repository URL* , *Credentials* and *Branch to build*.
2. Check *Build whenever a SNAPSHOT dependency is built*



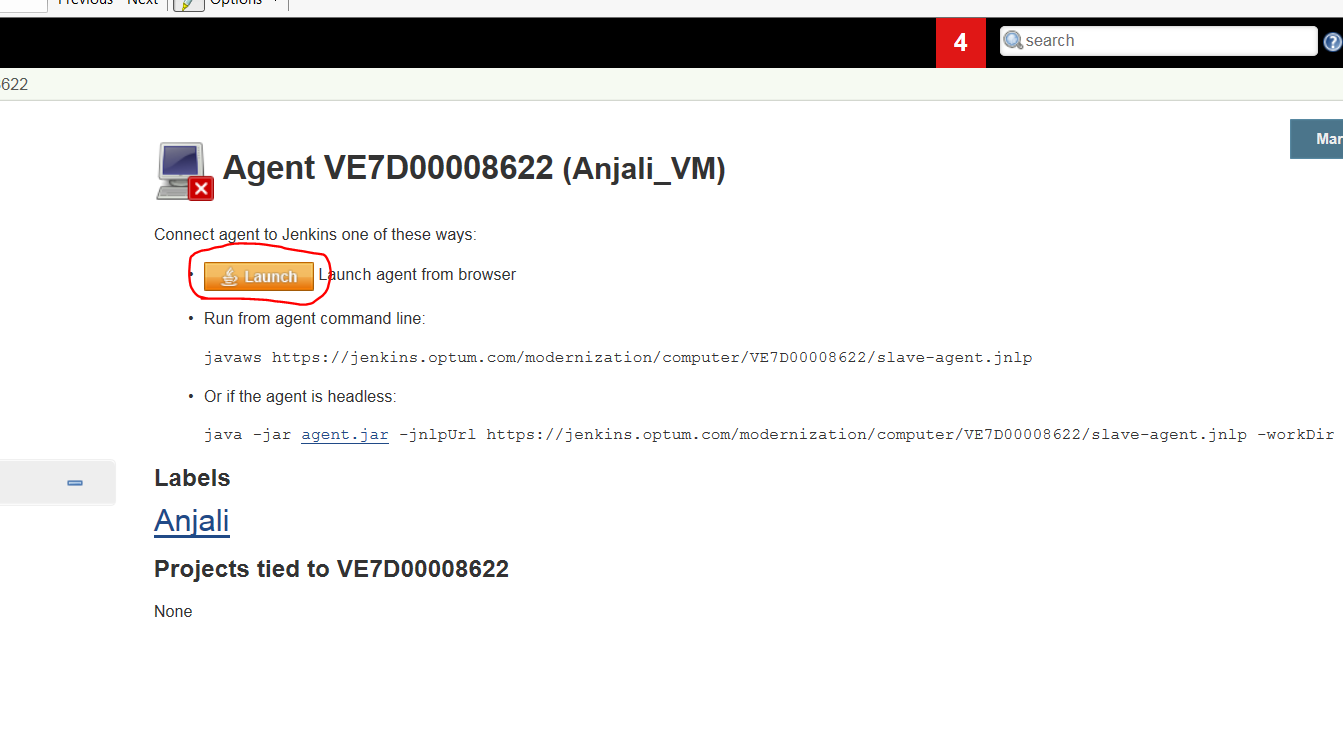
1. Mention *Maven Version, Root POM and Goal and Version*
2. Select *Post Steps as Regardless of build results*



1. Mention *Build setting as email notification* for build results
2. Select reports plugin in *Post build action* as per your project requirement



1. Click on *Save* button
2. Now Launch your Virtual Machine by clicking on the VM and then click on *Launch.*



*You will notice the popup (having connected message).*

Once your VM will be connected you can build your job by clicking on build now