***Jenkins Ant build with FreeStyle Project***

***Step 1: Need to install Apache Ant in server:***

***Installing Ant on Linux***

Enter the URL: <http://ant.apache.org/bindownload.cgi>

# cd /opt/

wget <https://dlcdn.apache.org//ant/binaries/apache-ant-1.10.12-bin.zip>

unzip apache-ant-1.10.12-bin.zip

# export ANT\_OPTS="-Xmx256M"

# ANT\_HOME=/opt/apache-ant-1.10.12

# export PATH=${ANT\_HOME}/bin:${JAVA\_HOME}/bin:${PATH}

======================================================

On the Apache Ant Project page, find the heading Current Release of Ant.

Select apache-ant-1.6.5-bin.tar.gz [PGP] [SHA1] [MD5].

Save and extract the package file into a Linux home directory.

Set the ANT\_OPTS environment variable: export ANT\_OPTS="-Xmx256M"

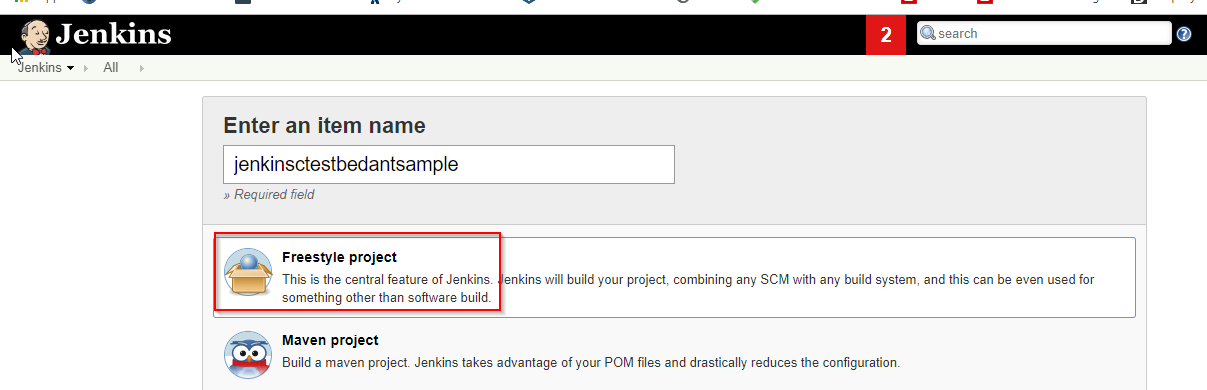
Set the ANT\_HOME environment variable to the directory where you installed Ant: export ANT\_HOME=${ant\_dir}

Set the PATH environment variable to include the directory where you installed the Ant bin directory: export PATH=${ANT\_HOME}/bin:${JAVA\_HOME}/bin:${PATH}

Parent topic: Installing the small package on Linux

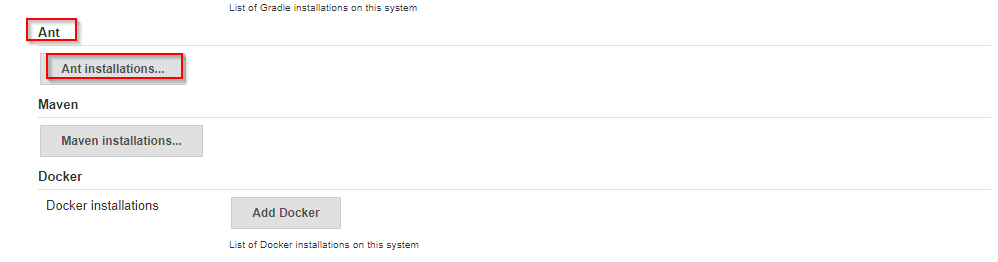
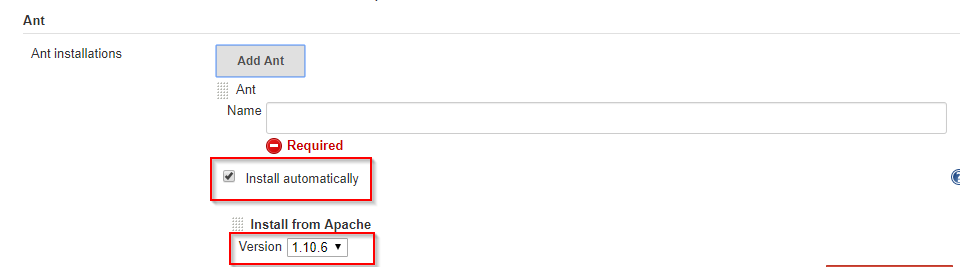
***Step 2: install all Junit related Jenkins plugin for Junit Testing…………..***

***Step 3 : Create a free style Project……….***

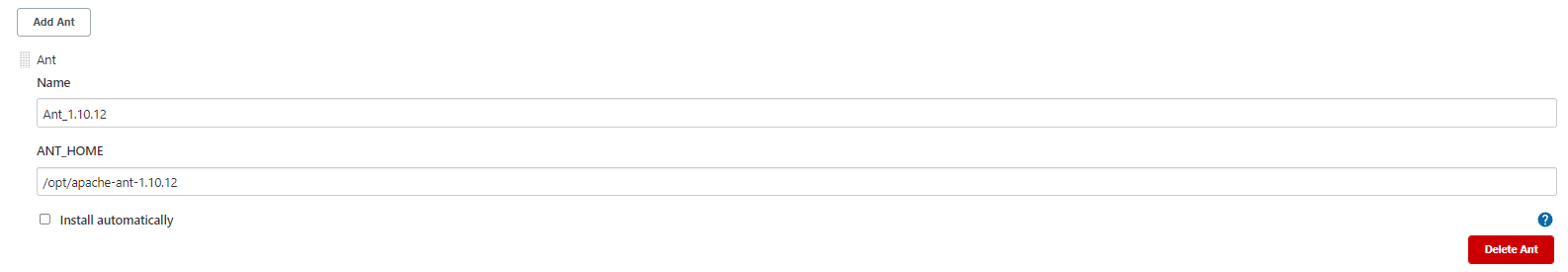


Now before going to configure that above project first we need to configure some options for making ANT workable in Jenkins. Please go to **Jenkins > Manage Jenkins > Global Tool Configuration.**

There you will find one section named **ANT.** Click on **Ant Installations:**

Here in Name section need to give your any preferred name for ANT with version info so that you can track that later. Below that check **install automatically** function**.** This will automatically install ANT in the corresponding Jenkins server. You also can choose your preferred ANT version from below Version drop down menu option. After successfully configured click Apply and Save. Or if we install Manually then we have to set just ant Home Path like below

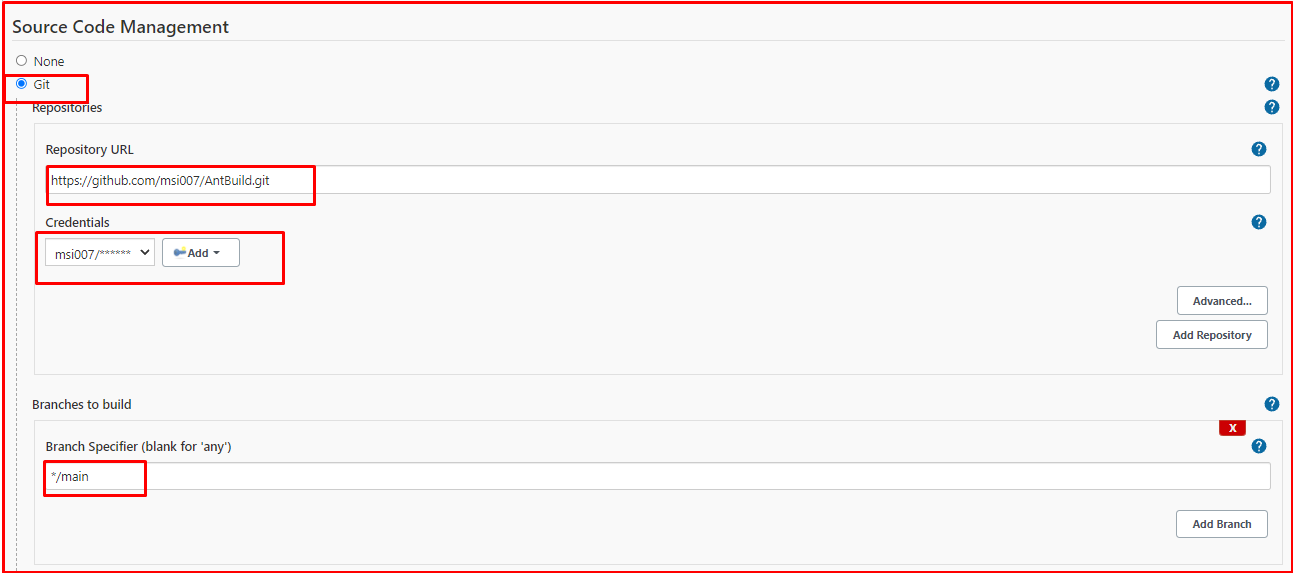


Need to ensure that Junit plugin is installed in Jenkins

**Step 4 :**

Now we need to configure our newly created job to build code using ANT……………

Here in **Source Code Management** we will use option Git and will provide our corresponding ANT based projects Repository URL.



Now in the Build section need to add one build step from “**Add build step**” and need to select “**Invoke Ant**” option



Here I select MyAnt1.10.12 from “Ant Version” which we had already configured in **Jenkins > Manage Jenkins > Global Tool Configuration.**

Target section is for specifying any specific build action from **build.xml**. For example, from below part of **build.xml** we can specifically define the target to take action. For below one if we use “test” in Target section only clean target will be built. But if it remains empty that means build should take place for all targets



After this Click Apply and Save. Now if we run this Jenkins Job it will first fetch all codes from Github and then will build that code with corresponding **build.xml** using ANT. After successful build you will see below in console output:

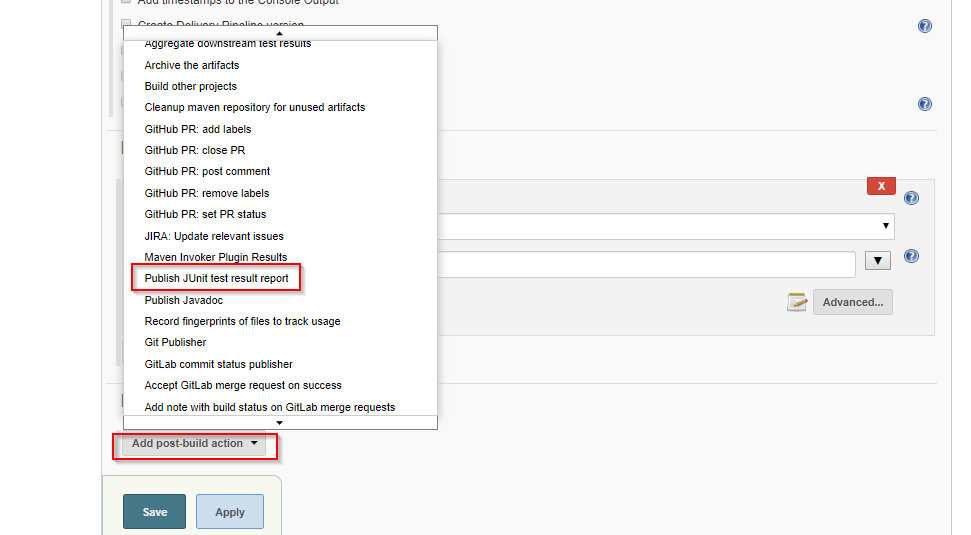


Step 5:

**JUnit Report generation from Jenkins:**

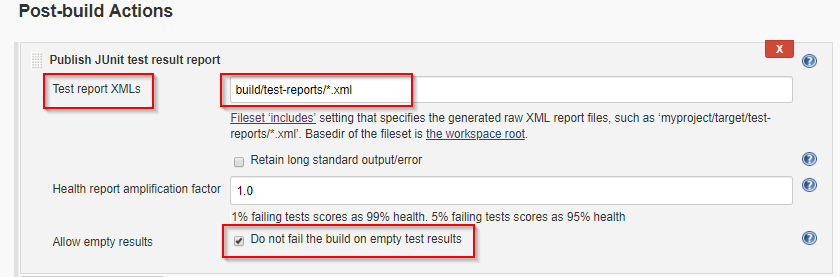
We also can visualize our JUnit testing report from Jenkins GUI. Jenkins has plugin named JUnit which enabled Jenkins to publish Unit testing in report format from a xml file.

To generate JUnit testing report for our above ANT based code, need to go to **Project > jenkinscitestbedantsample < Configure**. From there we need to add one post build option which will generate JUnit testing report after build takes place. Need to select “**Publish JUnit test result report**”



There is one filed named “**Test report XMLs**” where we need to provide the location of JUnit xml report from where Jenkins will show us in GUI. If we look at specially in JUnit section in our **build.xml** we can see the test reports location.





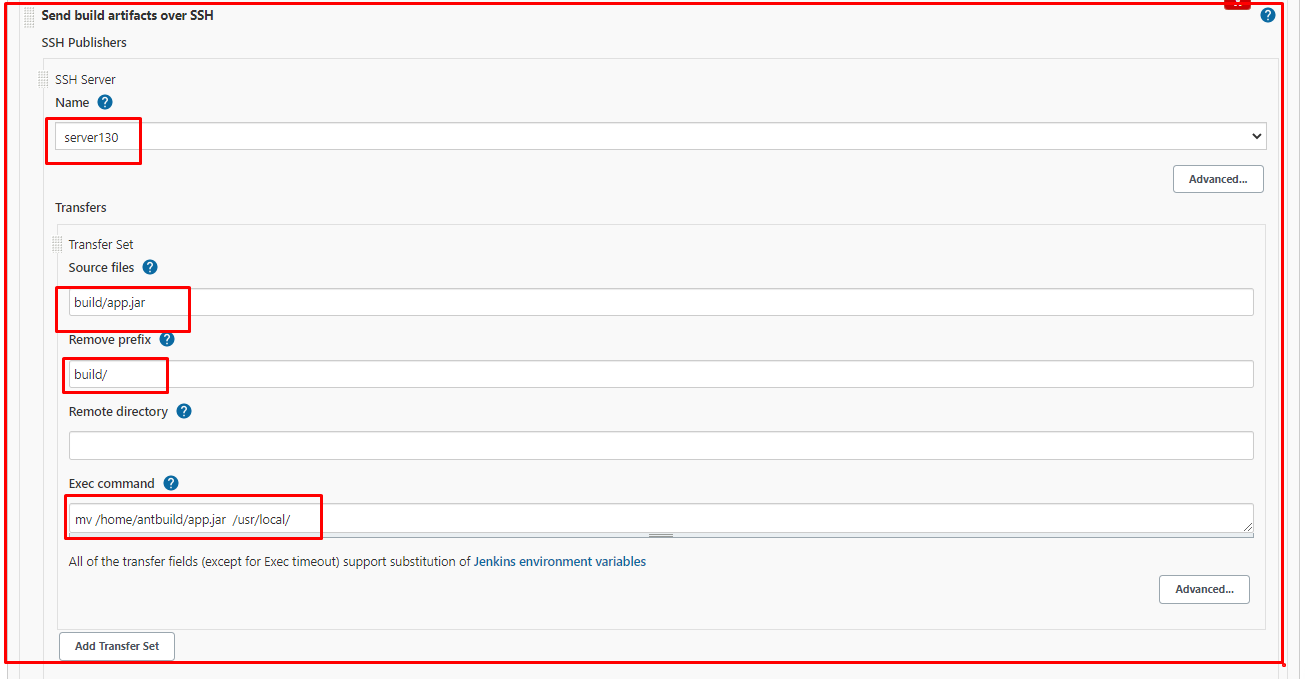
There is another option called “**Allow empty results**” which enables this particular Jenkins build not to get failed if our targeted location doesn’t contain any **\*.xml** report file. You can **uncheck/check** based on your requirements. After making above changes click apply and save. Build again & see the test result.

Now if need we can transfer created app.jar file to another server (production server) by another post build action.

At first we have to add server **Jenkins > Manage Jenkins > Configuration Systems.** Then add server by **Publish over SSH**



**Post Build Action:**



*Prepared by ……………….*

*Md. Saiful Islam*