**Jenkins Installation**

**Step 1: Add Jenkins Repository Jenkins package is not available in the default CentOS and RHEL repositories. So we need to add jenkins repository using the beneath commands.**

**# sudowget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo**

sudowget -O /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

**# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key**

**Step 2: Install Jenkins and Java Run the below yum command to install Jenkins and java.**

**# yum install java-1.8.0-openjdk –y**

**For Redhat 8:**

**# dnf update**

**# dnf install java-1.8.0-openjdk-devel**

**Java –version**

**# yum install jenkins**

**# yum install maven (used to compile the code for java application ,if it is ndejs application install npm installer)**

**Ip Mvn–version**

**:8080 try in browser**

**# yum install git**

**# git clone url**

**Step 3: Start and Enable Jenkins Service Run the following systemctl commands to start and enable the jenkins service**

**# systemctl start jenkins**

**# systemctl enable jenkins**

**Step 4: Open the ports (80 and 8080) in OS firewall. In case firewall is enabled on your Linux server then run the following commands to open jenkins related ports like 80 and 8080.**

**# firewall-cmd --zone=public --add-port=8080/tcp --permanent**

**success**

**# firewall-cmd --zone=public --add-service=http --permanent**

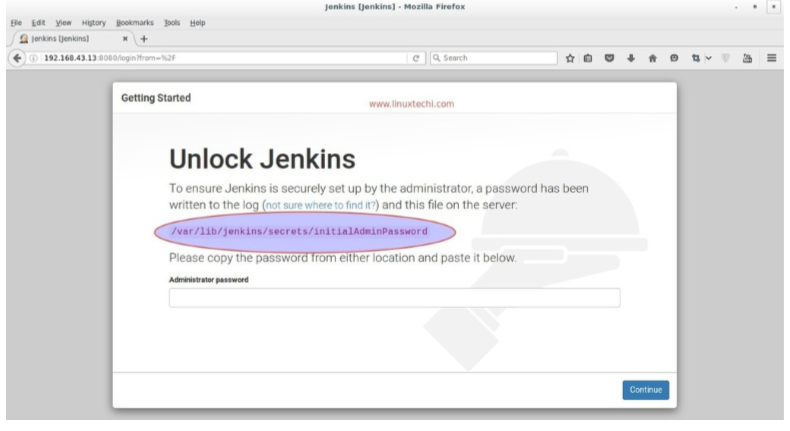
**success**

**# firewall-cmd --reload**

**success**

**Step:5 Access the Jenkins Web portal Access the**

**URL :http://<Ip-Address-of-your-Server>:8080**

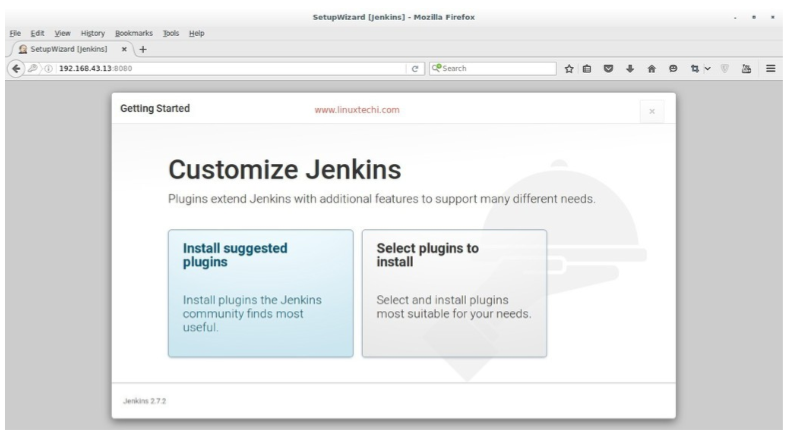


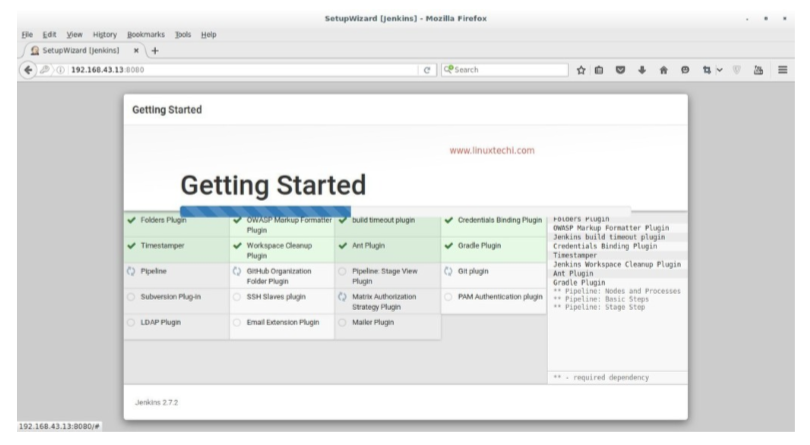
**Admin password is created and stored in the log file “/var/log/jenkins/jenkins.log“. Run the below command to get the password.`**

**# grep -A 5 password /var/log/jenkins/jenkins.log**

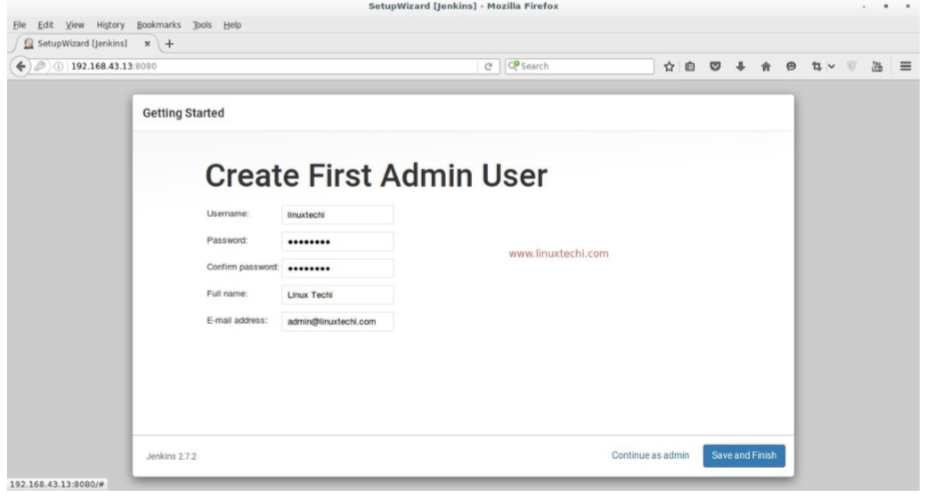
**Copy the password and paste it in above windows and click on Continue..**

**In the next windows Select the option : Install suggested plugins**

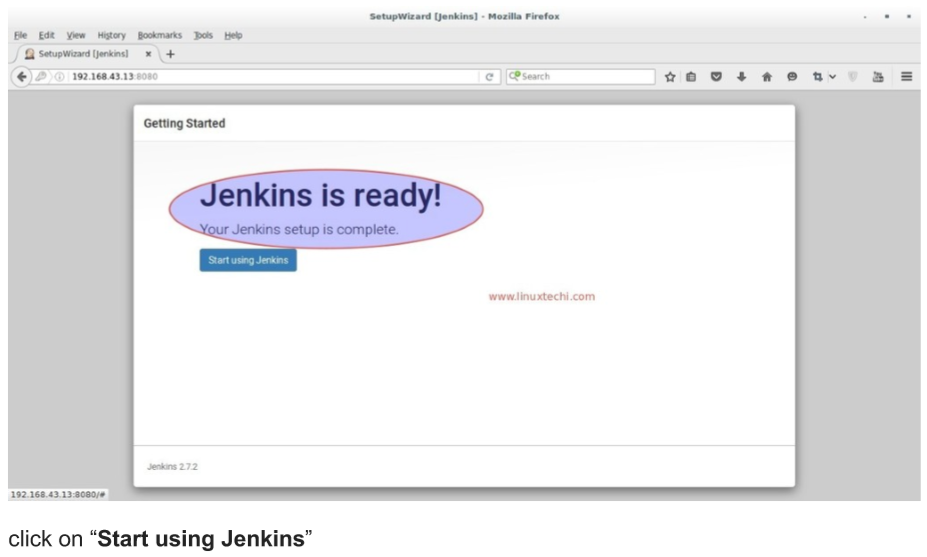


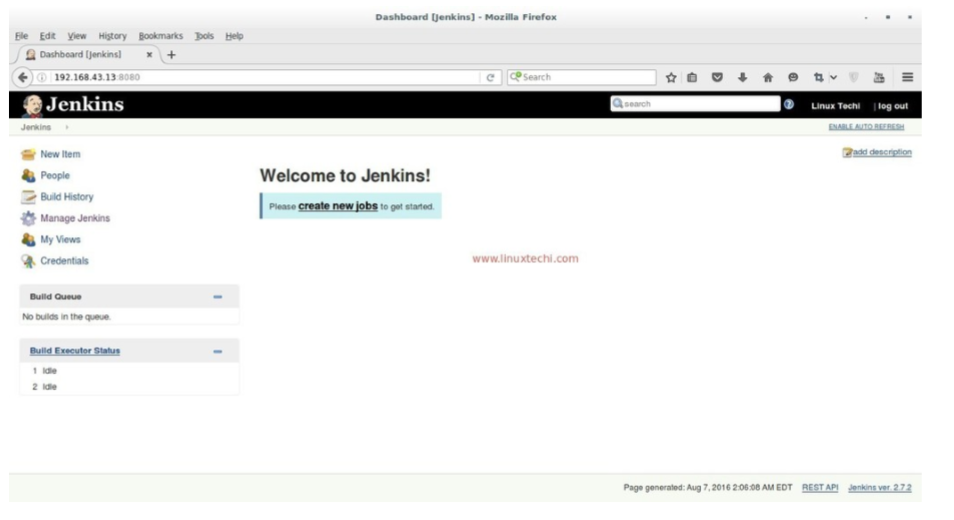


**As we can see required plugin installation is in progress for Jenkins. Once it is done with plugin installation. It will ask to create Admin User**



**Click on Save and Finish**





**To configure the git/gitlab project install git package in Jenkins server.**

**# yum install git -y**

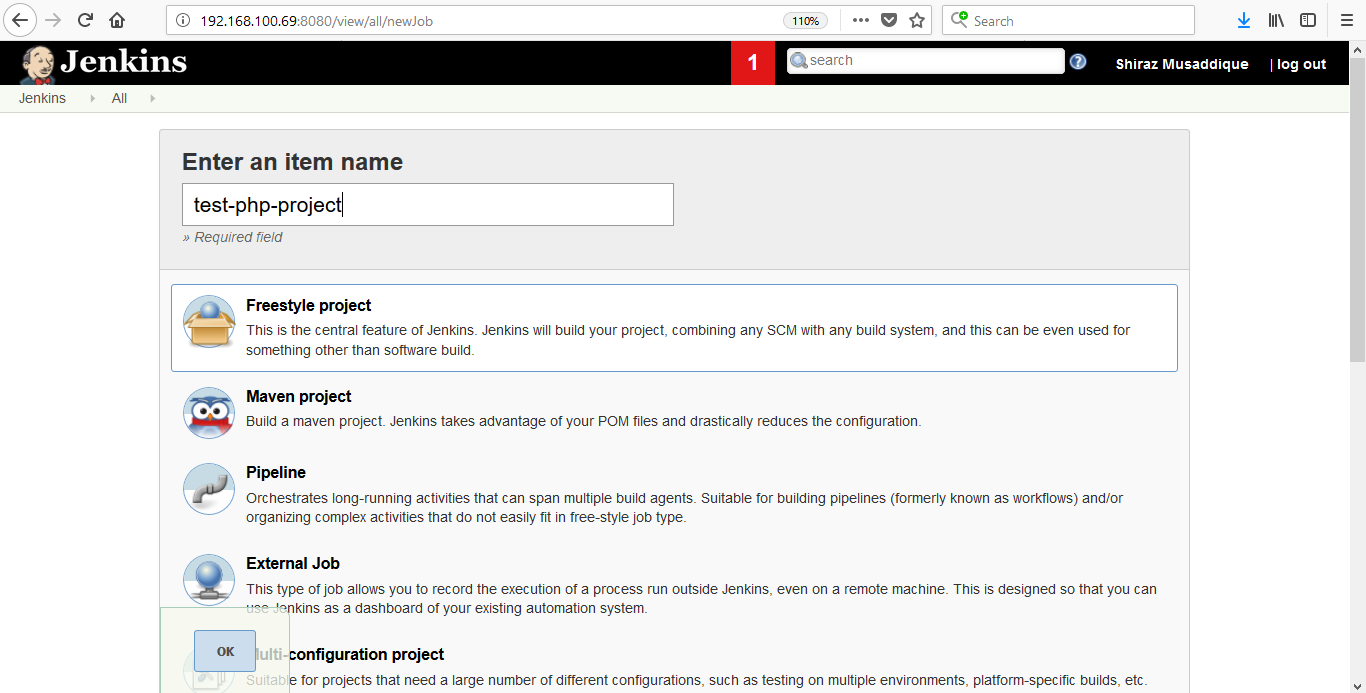
**To build the PHP project refer Jenkins plugins document.**

**To create the Job in Jenkins follow below steps.**

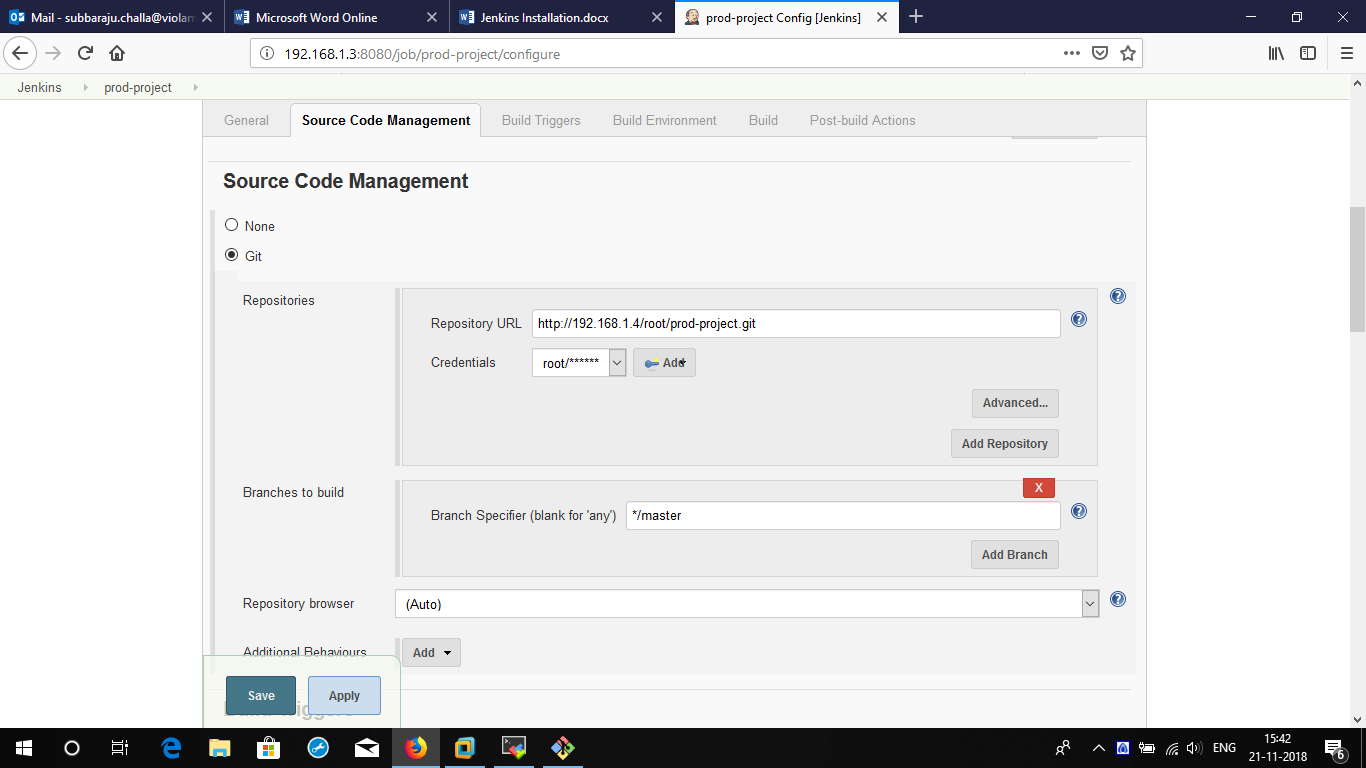
**Select new Item in left pan**



**Enter Item Name and select Free Style Project**



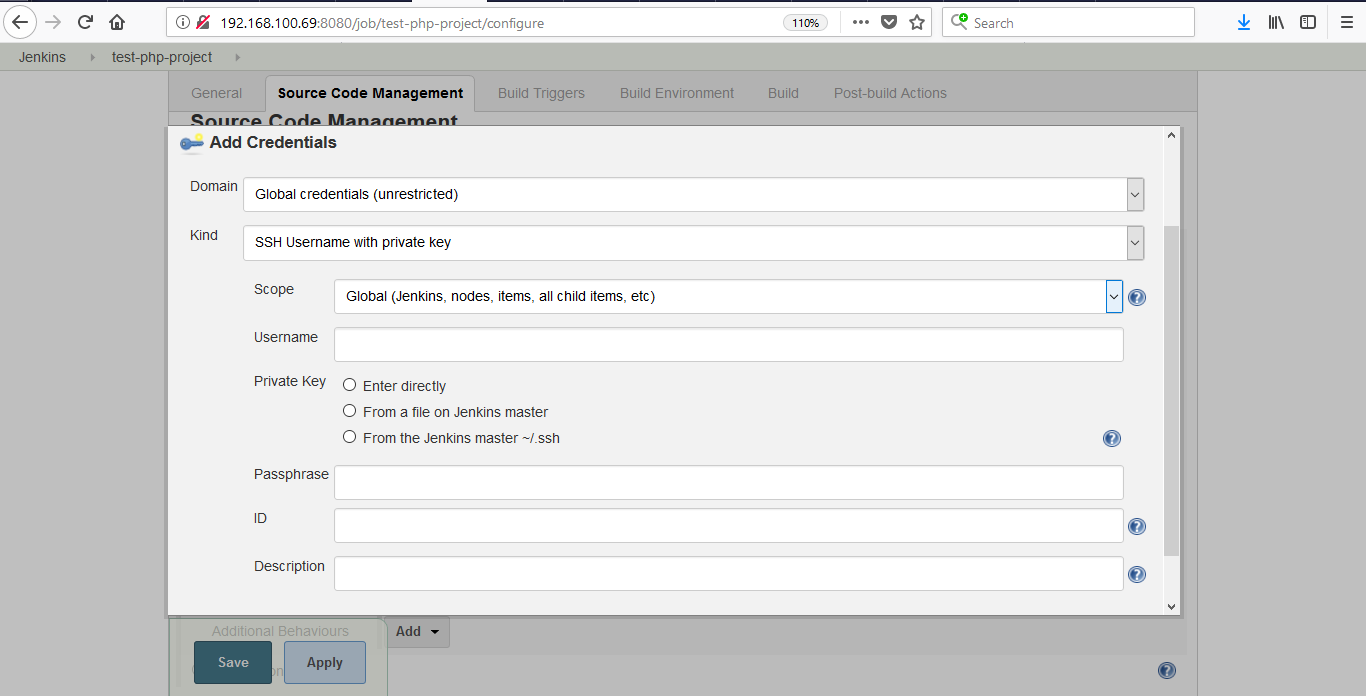
**Provide Repository URL and also add credential to click on add button to access that repository**



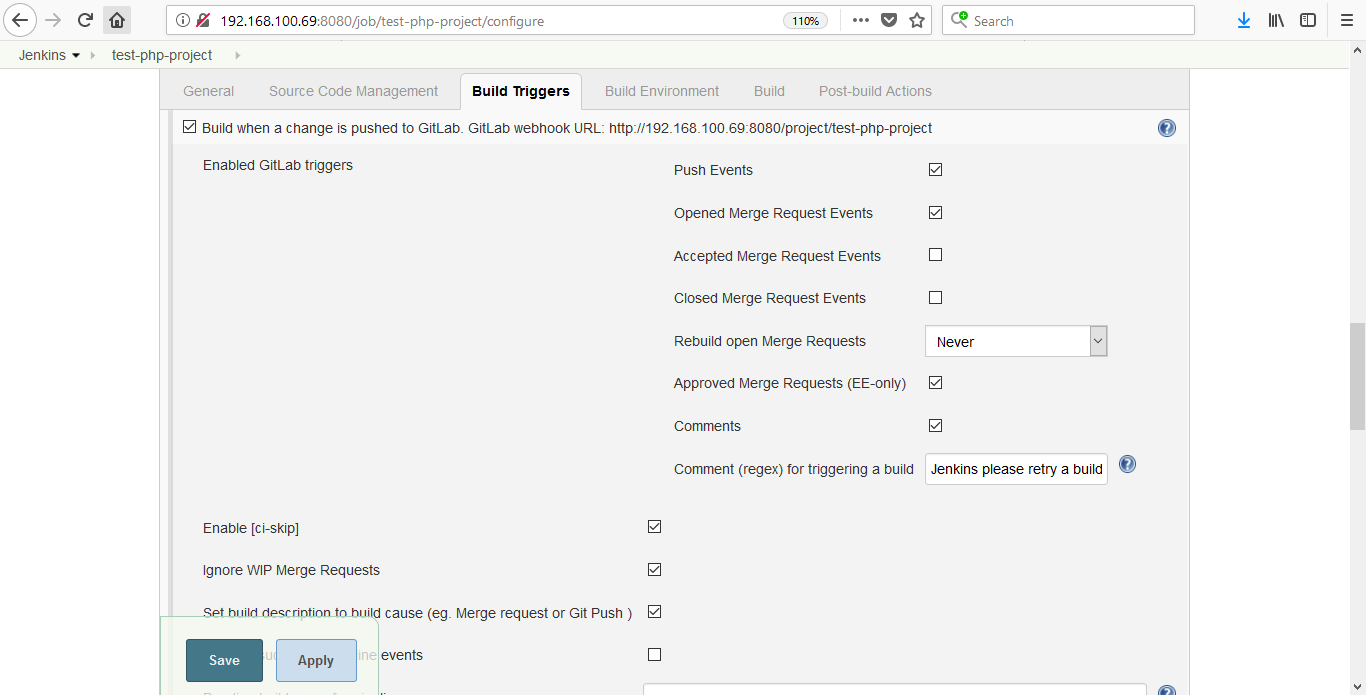
**Provide the Credntial and select SSH username with private key Select enter it directly and paste your root rsa\_idssh key into it**

**Or select user name with password and give git user name and password.**

**>> create ssh keys in jenkins and copy id\_rsa.pub key in to gitlabssh keys**



**In Build Triggers select this option Build when a change is pushed to GitLab. GitLab webhook URL:** [**http://192.168.100.69:8080/project/Jobname**](http://192.168.100.69:8080/project/Jobname)



**Now copy above urland check on accept merge**

**Now go to gitlab server. :**

**This is to trigger webook**

**Go to project >>intigration and selcetwebhook**

**Now paste above url>> copy api token from jenkins>> for taking api token**

**Go to jenkins user and configure >> create api token >> copy and paste it in**

**Webhookapi tokens**

**Copying access token from git lab server to jenkins:**

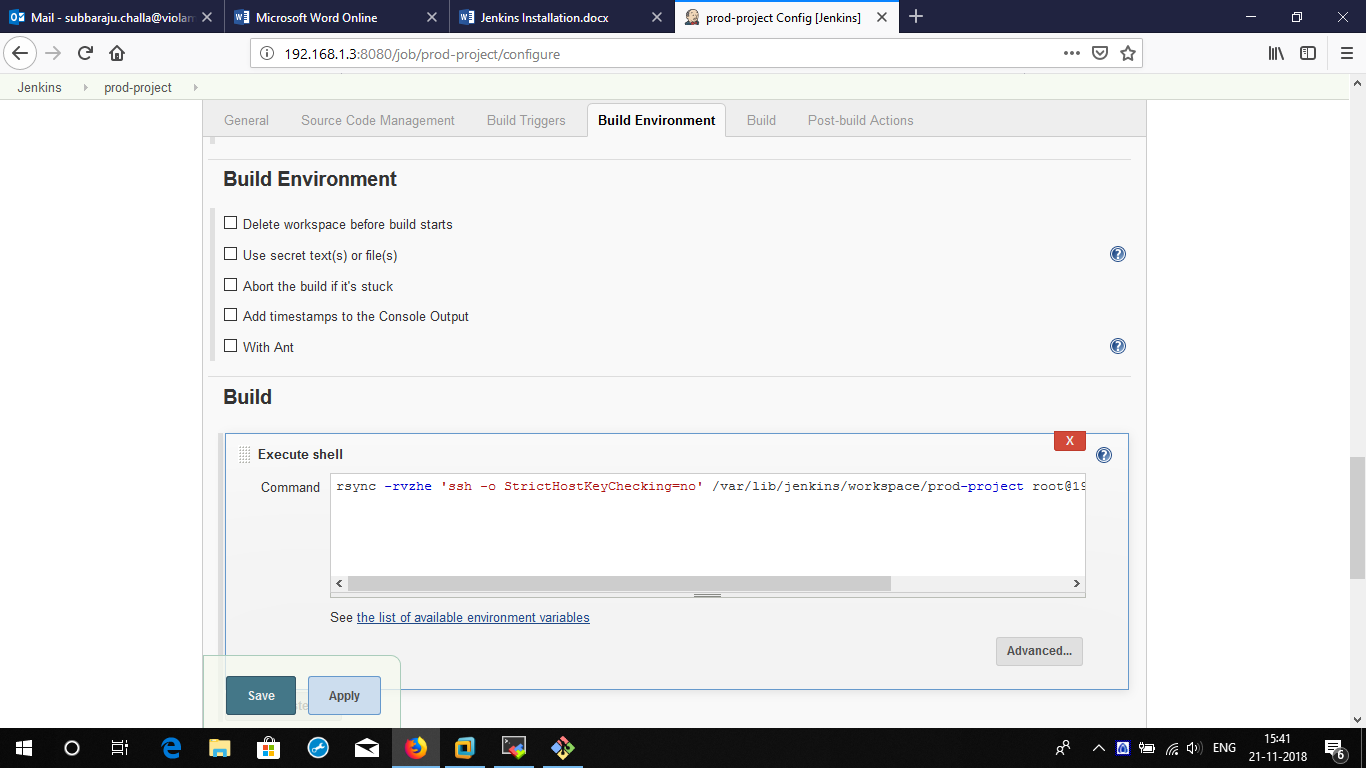
**This is to enable job from url from jenkins to gitlab**

**Go to setting in gitlab server >> access tokens >> create token and copy >>**

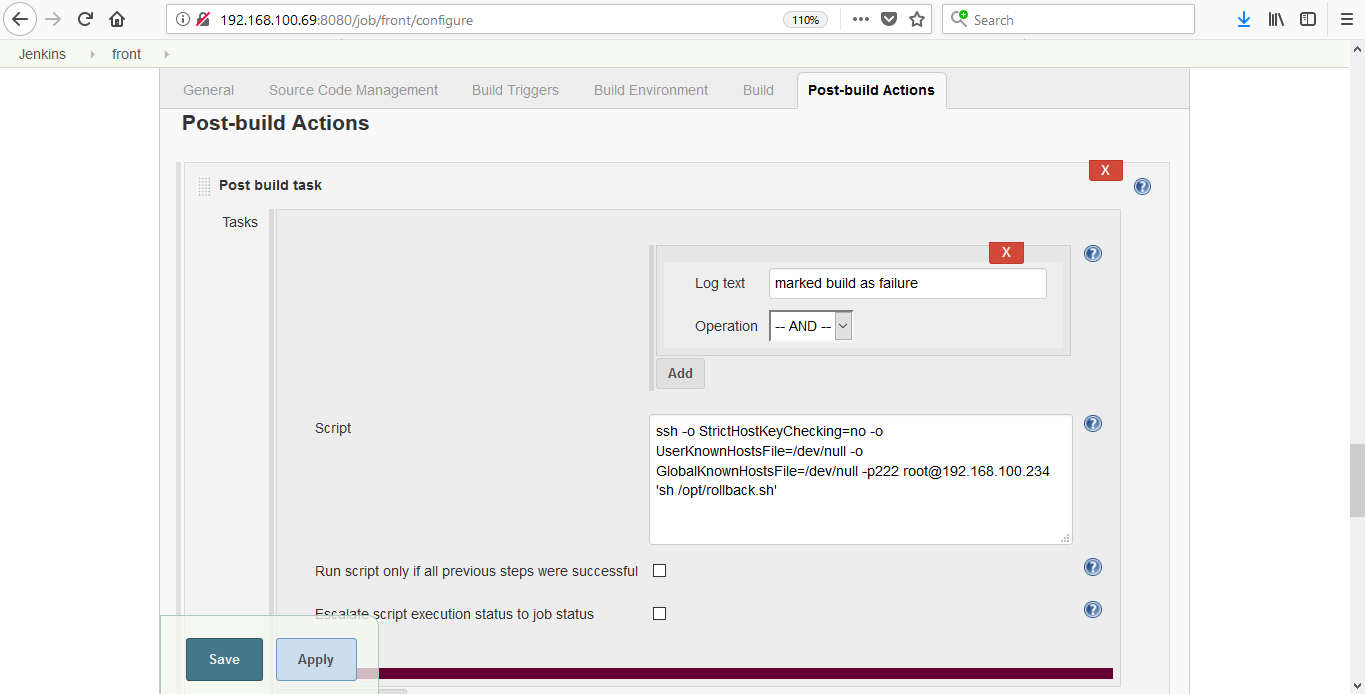
**Go to jenkins server >> Manage jenkins>> configure system >> in git lab field**

**Give git lab url**[**http://192.168.100.4**](http://192.168.100.4)**>> credentials >>git lab api token**.

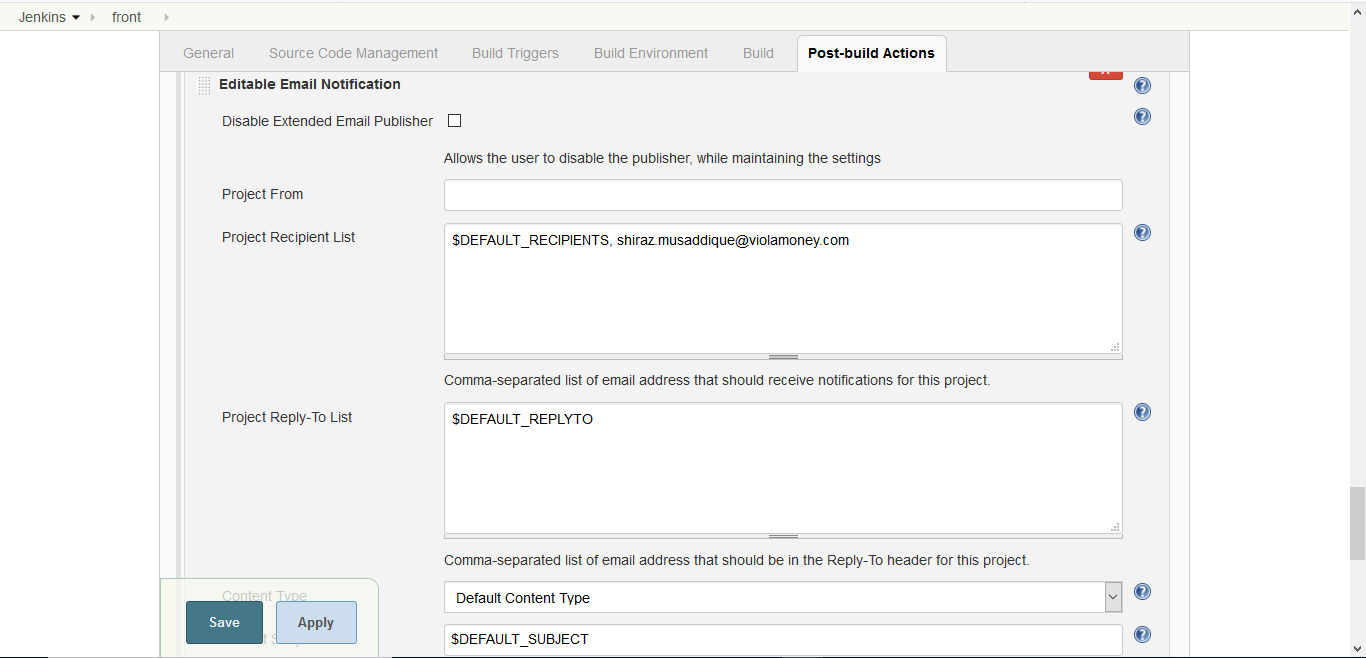
**Then click on add build steps and select execute shell and provide your scripts or command to run**



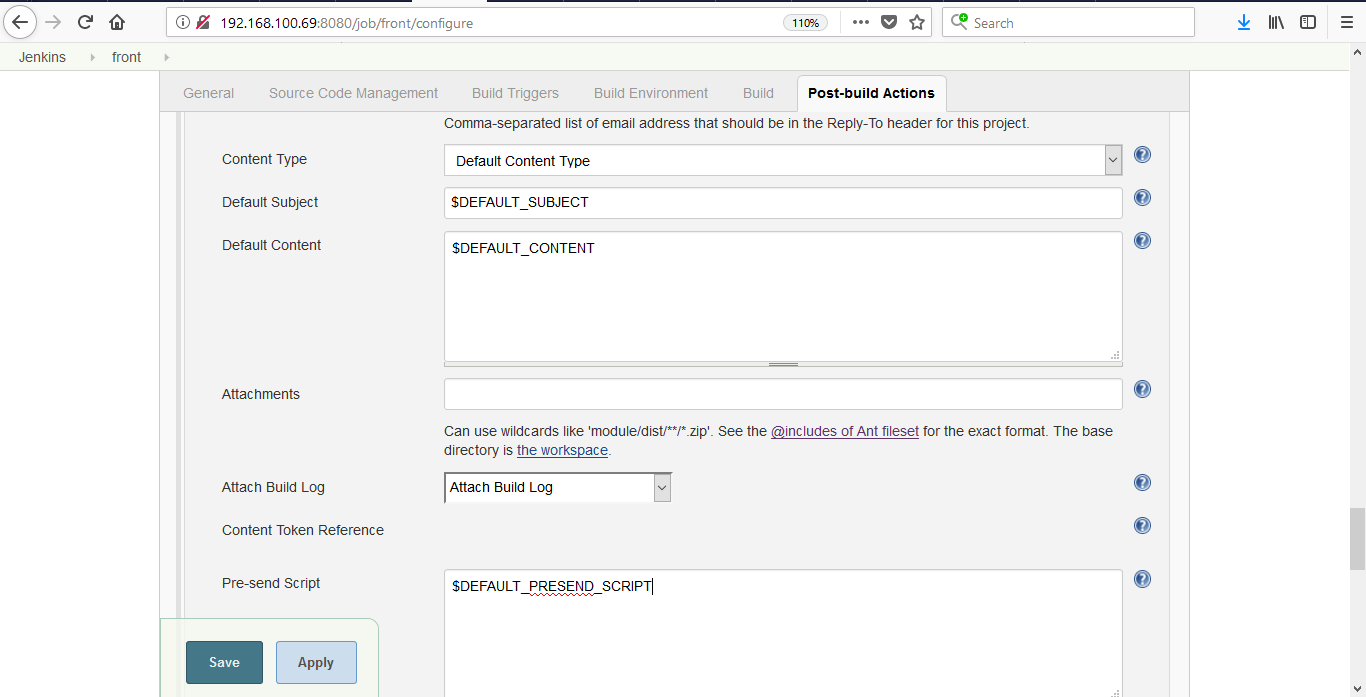
**Click on Post Build action and select post build task Provide the Log text which you want to search in Build log and run the script. You can use this for roll back steps. You can add this as many as you can.**



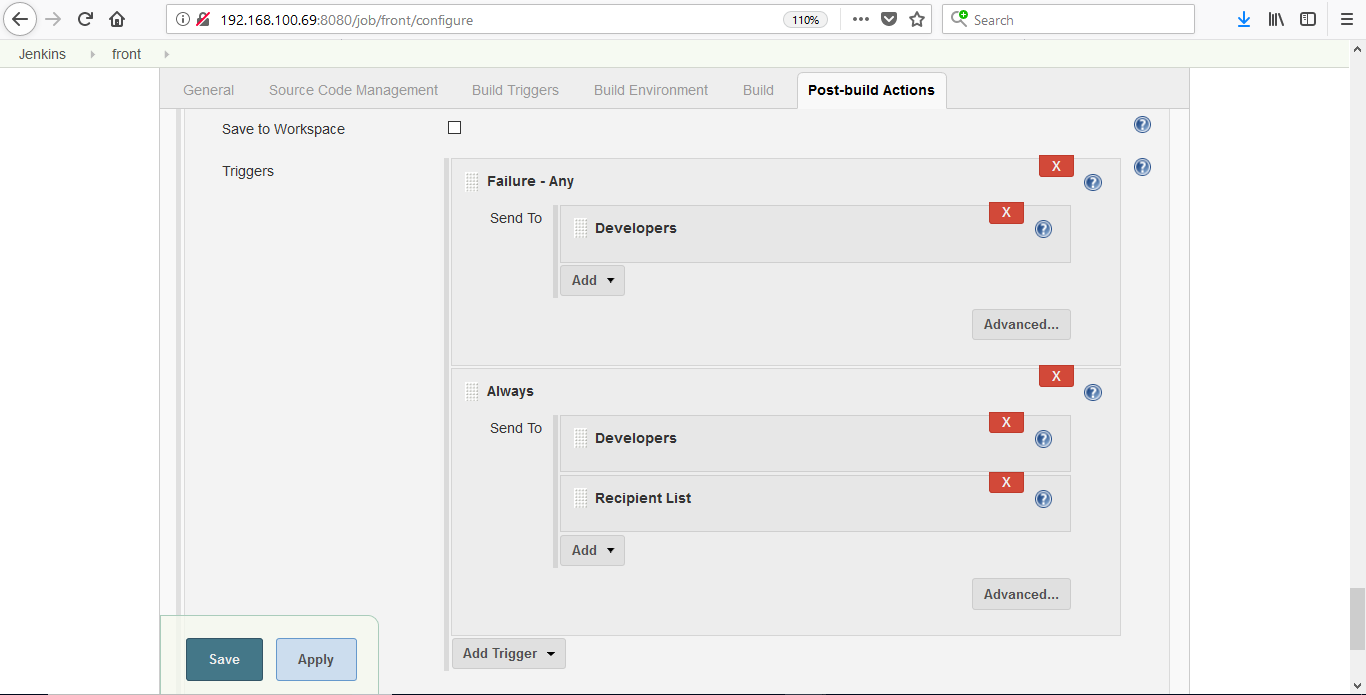
**Click on Add Post Build Action and select Editable Notification Provide your email address or concerns email address.**



**Select the Attach build log when mail has sent.**



**Select Advanced tab in the plugin and click on Add Trigger and select Always.**

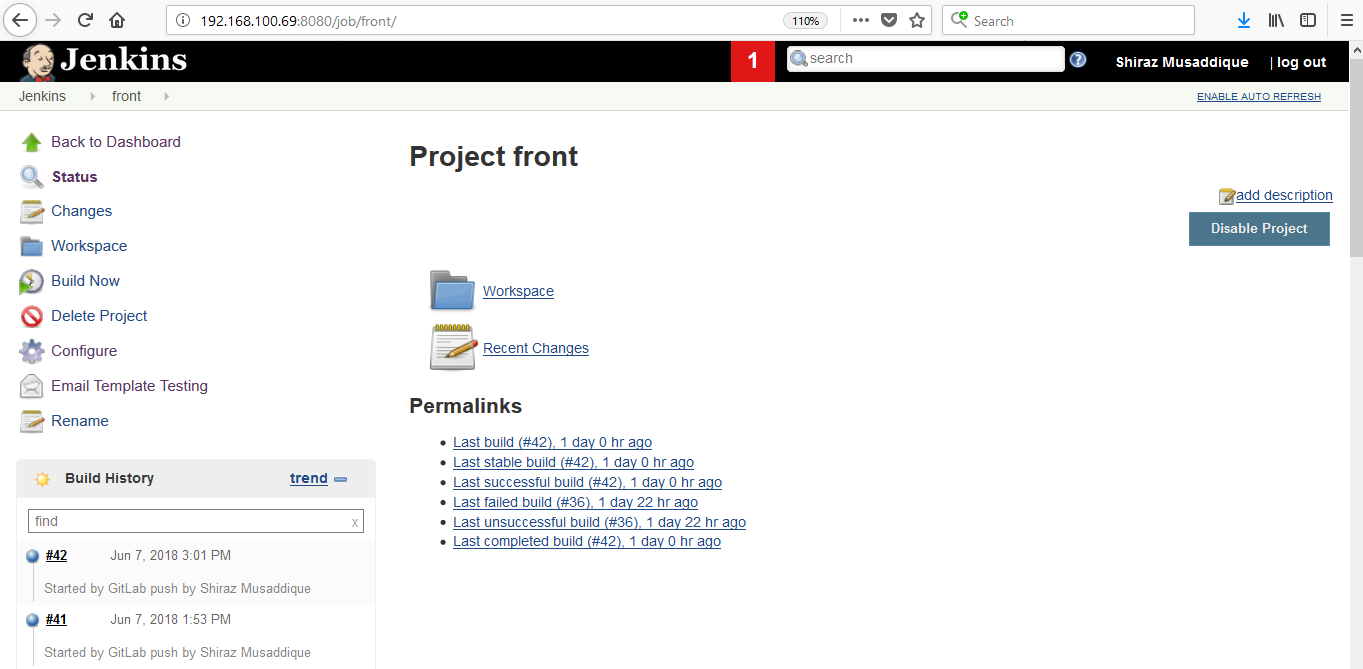


**Click apply and save.**

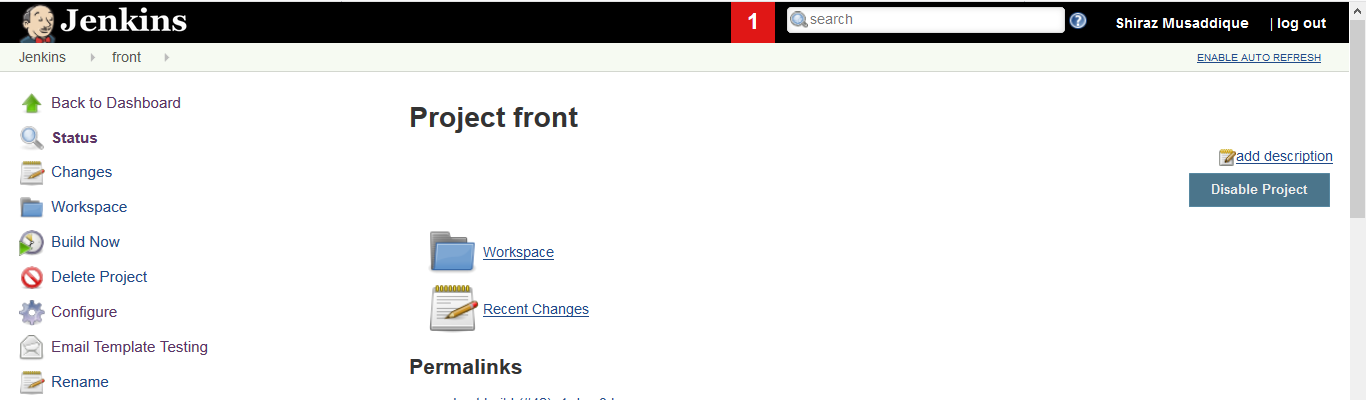
**Now when anyone will push the code into repository the build will automatically trigger**

**OR**

**You can build it Manually by clicking build now.**



**OR**



**Execute Shell Commands**

**cd /var/lib/jenkins/workspace/front**

**echo "Click on this url to see your project http://192.168.102.176/dashboard?id=frontv" | mail -s "code scan has started" testmail@violamoney.com**

**/opt/sonar-scanner/bin/sonar-scanner -X -Dsonar.projectKey=frontv -Dsonar.sources=. -Dsonar.host.url=http://192.168.102.176:9000 -Dsonar.login=466ffeb6378d359bf56f7579ea25f40fffdb373c**

**The above commands are to send an email at the starting time when sonar-scanner start the scan and send the report to SonarQube.**

**rm -rf /var/lib/jenkins/workspace/front/vcard-front-\***

**zip -r vcard-${JOB\_NAME}-${BUILD\_NUMBER}.zip /var/lib/jenkins/workspace/front/**

**scp -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o GlobalKnownHostsFile=/dev/null -rvp -P222 /var/lib/jenkins/workspace/front/vcard-front-\* root@192.168.100.234:/mnt/frontfiles/**

**ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o GlobalKnownHostsFile=/dev/null -p222 root@192.168.100.234 'sh /opt/front.sh'**

**w3m** [**https://testvc.violamoney.com/frontv**](https://testvc.violamoney.com/frontv)

**Front.sh file**

**#!/bin/bash**

**zip -r /var/www/html/bkp/front-violacard`date +%Y-%m-%d.%H:%M:%S`.zip /var/www/html/frontv ; rm -rf /var/www/html/frontv/\* ; rm -rf /mnt/frontfiles/var**

**cd /mnt/frontfiles ; unzip `ls -t /mnt/frontfiles/\*zip | head -1 | awk {print'$1'}`**

**cp -r /mnt/frontfiles/var/lib/jenkins/workspace/front/\* /var/www/html/frontv/ ; chown -R apache:apache /var/www/html/frontv/ ; chmod -R 755 /var/www/html/frontv**

**mv /var/www/html/frontv/data/application/config/constants.php /var/www/html/frontv/data/application/config/constants.php`date +%Y-%m-%d.%H:%M:%S`**

**cp -rvp /mnt/frontfiles/constants.php /var/www/html/frontv/data/application/config/**

**Above commands will connect to the server and copy the workspace code zip in to the server path /mnt/frontfiles**

**Other ssh command is used for run the script on server which is stored in /opt/front.sh**

**Post Build Task Script command**

**This script will run when the plugin search the specific string like failure in Build log and roll back the deployment on the server.**

**ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o GlobalKnownHostsFile=/dev/null -p222 root@192.168.100.234 'sh /opt/rollback.sh'**

**rollback.sh file**

**#!/bin/bash**

**zip -r /var/www/html/failed/failed-front-violacard`date +%Y-%m-%d.%H:%M:%S`.zip /var/www/html/frontv**

**rm -rf /var/www/html/frontv/\***

**rm -rf /var/www/html/bkp/var**

**cd /var/www/html/bkp/ ; unzip `ls -t /var/www/html/bkp/\*zip | head -1 | awk {print'$1'}`**

**cp -rp /var/www/html/bkp/var/www/html/frontv/\* /var/www/html/frontv/**

**chown -R apache:apache /var/www/html/frontv ; chmod -R 755 /var/www/html/frontv**

**echo "Roll back has completed to the test server" | mail -s "Roll back has completed" shiraz.musaddique@violamoney.com**