**Jenkins tool setup and execution**

**# yum install wget unzip tree git -y**

**# yum install java -y**

To install jenkins, we have add the package to repo and then install it.

**#** **wget -O /etc/yum.repos.d/jenkins.repo** [**https://pkg.jenkins.io/redhat/jenkins.repo**](https://pkg.jenkins.io/redhat/jenkins.repo)

**# rpm --import** [**https://pkg.jenkins.io/redhat/jenkins.io.key**](https://pkg.jenkins.io/redhat/jenkins.io.key)

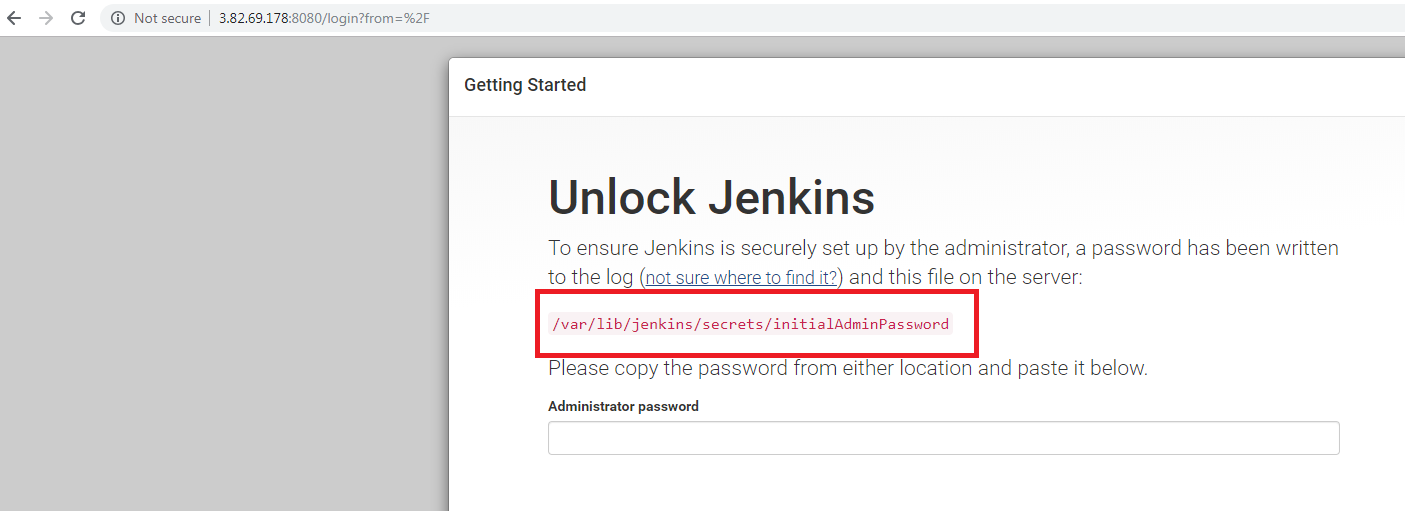
**# yum install jenkins -y**

**# service jenkins start**

* Browse URL with public IP with port number (8080)

**Ex:** <http://54.243.5.107:8080/>

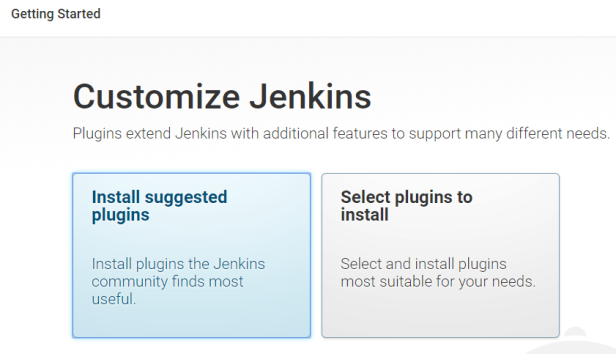
You can see the Jenkins page first time login as below



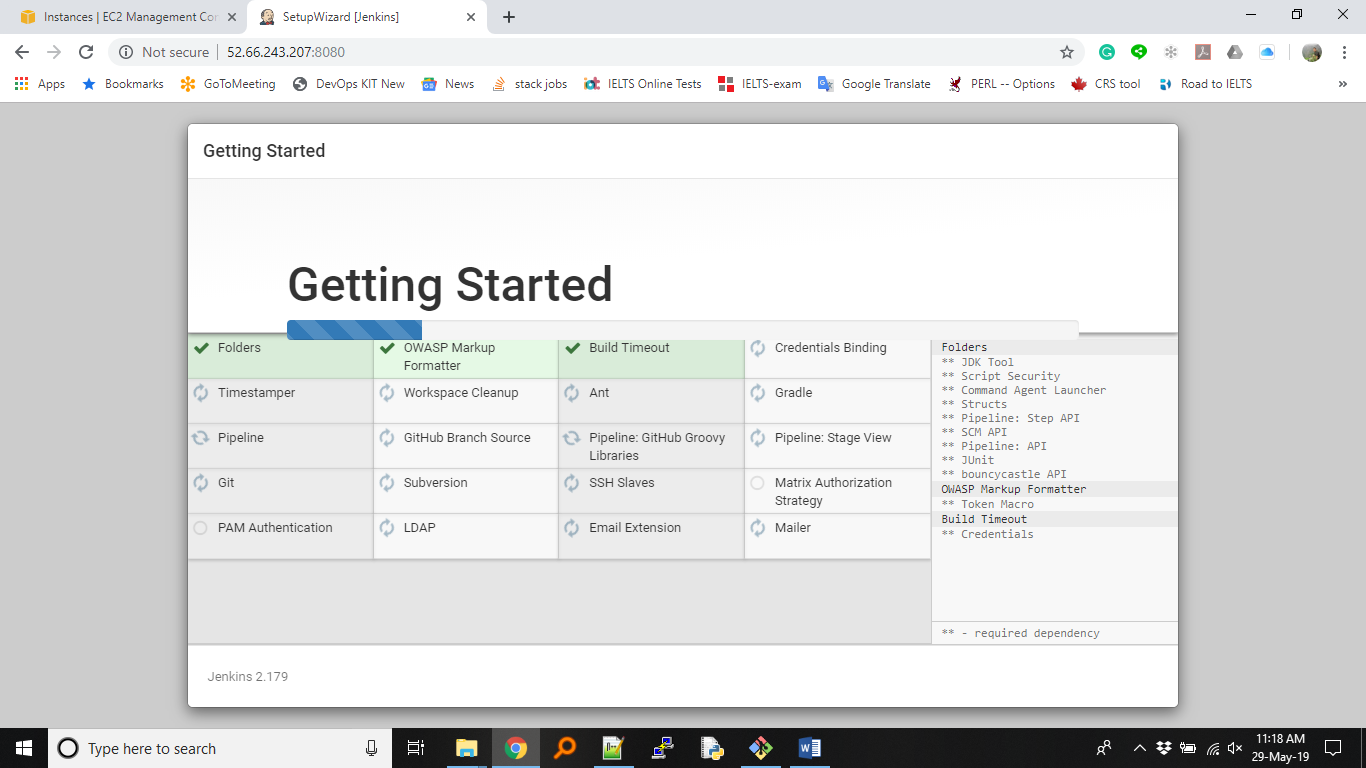
Copy the initial admin password from linux machine from this file (**initialAdminPassword**) and paste on the UI at prompting place.

**# cat /var/lib/jenkins/secrets/initialAdminPassword**

f735142601ba4d71bd32692c4d08946c



Click on install suggested plugins



So now, you are able to see the getting started page.

To check the Jenkins server running or not

**# service jenkins status**

**or**

**# ps -ef | grep jenkins**

[root@jenkinserver ~]# ps -ef |grep jenkins

jenkins 4119 1 10 01:55 ? 00:00:36 /etc/alternatives/java -Dcom.sun.akuma.Daemon=daemonized -Djava.awt.headless=true -DJENKINS\_HOME=/var/lib/jenk

ins -jar /usr/lib/jenkins/jenkins.war --logfile=/var/log/jenkins/jenkins.log --webroot=/var/cache/jenkins/war --daemon --httpPort=8080 --debug=5 --handlerCoun

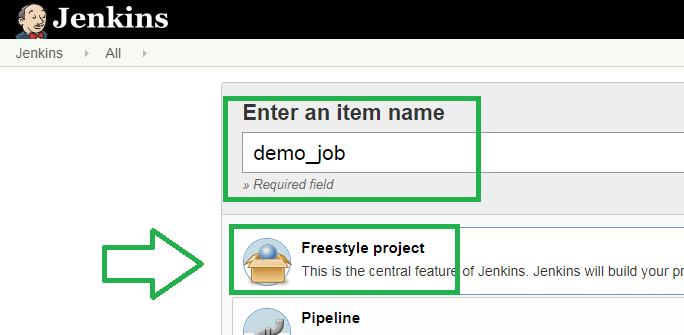
tMax=100 --handlerCountMaxIdle=20

root 4256 4221 0 02:01 pts/0 00:00:00 grep --color=auto jenkins

[root@jenkinserver ~]#

**Exercise 1**

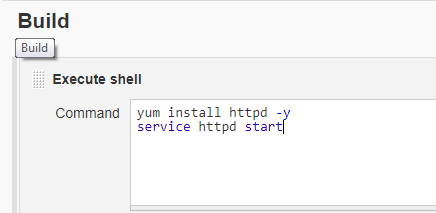
Please create new jobs to get started as demo\_job1

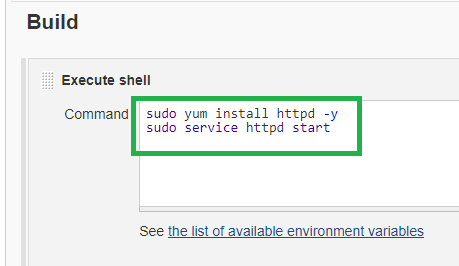


Click on Build tab

Select the option “Execute shell”

Write the below content in the box and save





It will fail even we use sudo command, because, all process running by “jenkins” user.

So, add the “jenkins” user in “sudoers” file at the bottom line to execute commands as a root user.

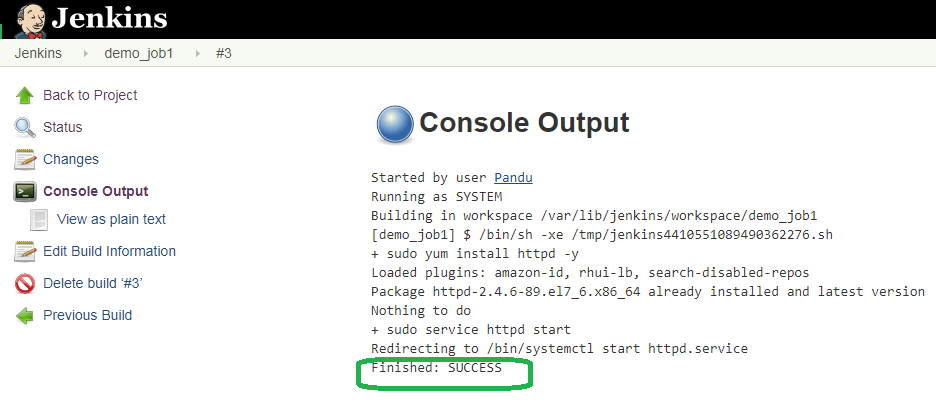
**# vi /etc/sudoers**

#includedir /etc/sudoers.d

ec2-user ALL=(ALL) NOPASSWD: ALL

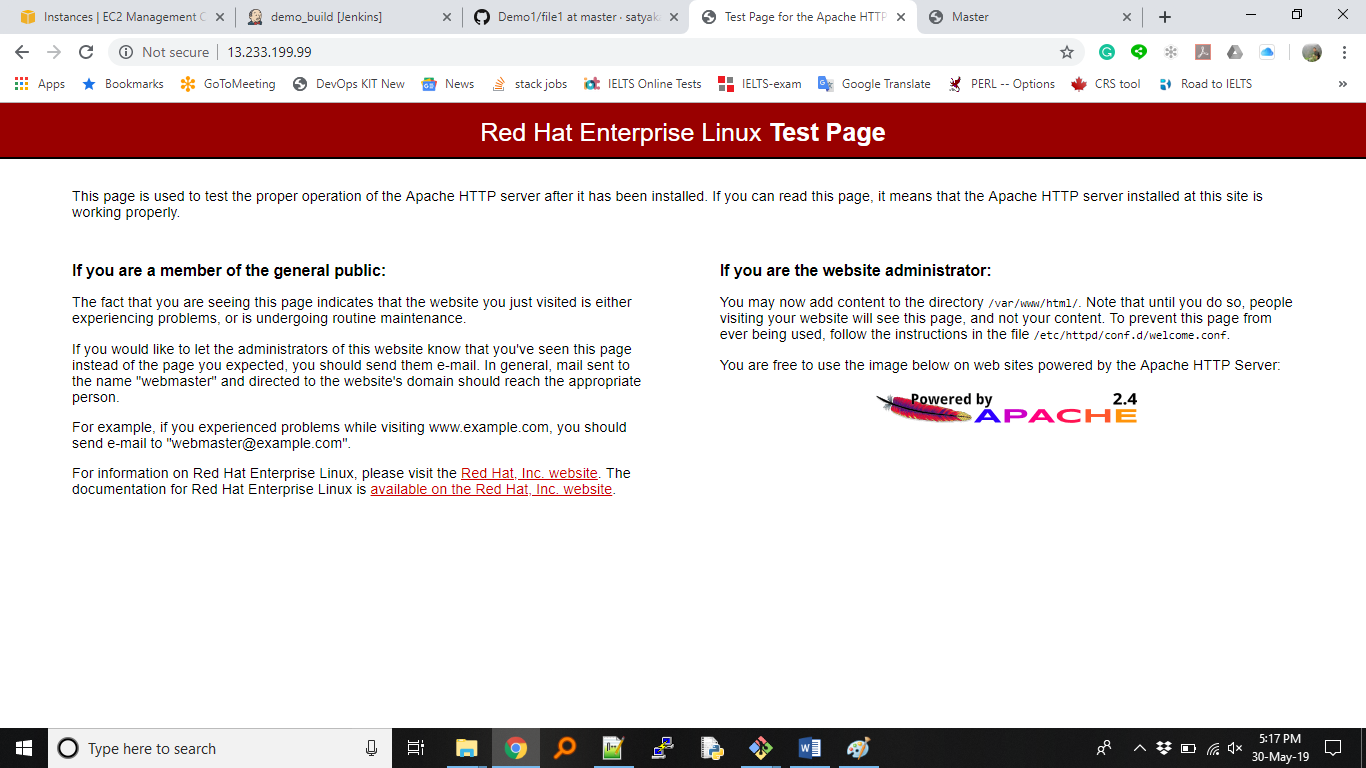
jenkins ALL=(ALL) NOPASSWD: ALL

Then build the job now, it will be successful



Browse the URL with public IP and will get http Apache test page

**Ex:** <http://13.233.199.99>



**Exercise 2**

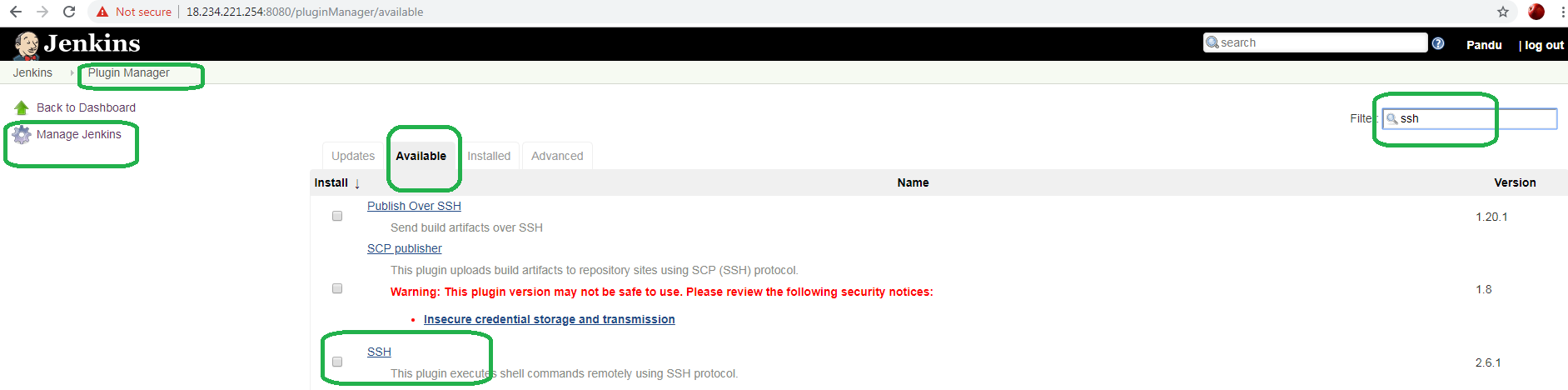
Now, we will try to do activity remote machine using ssh or executing commands on remote machine.

**Step1:** Create another RHEL machine

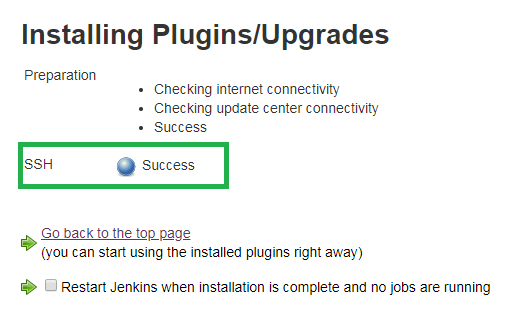
**Step2:** Install SSH plugins in Jenkins server, show that it can talk with remote machine

Jenkins 🡪 Manage Jenkins 🡪 Manage Plugins 🡪 Click on "Available” tab

In filter engine search respective plugin example: SSH

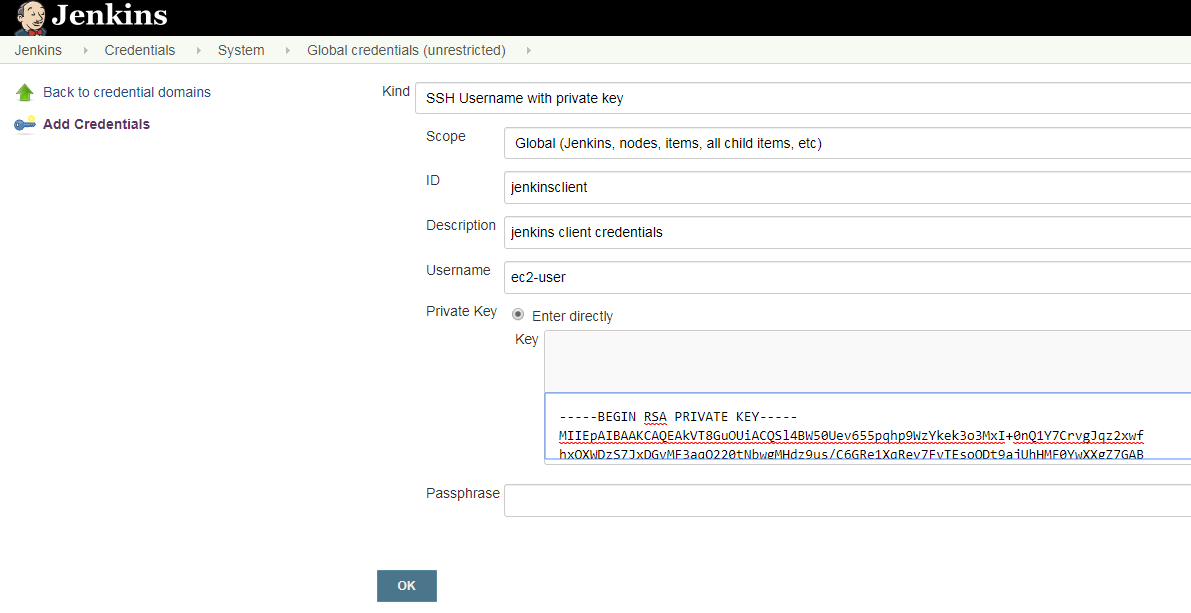


Select the “SSH” plugin -> Click on “Install without restart”



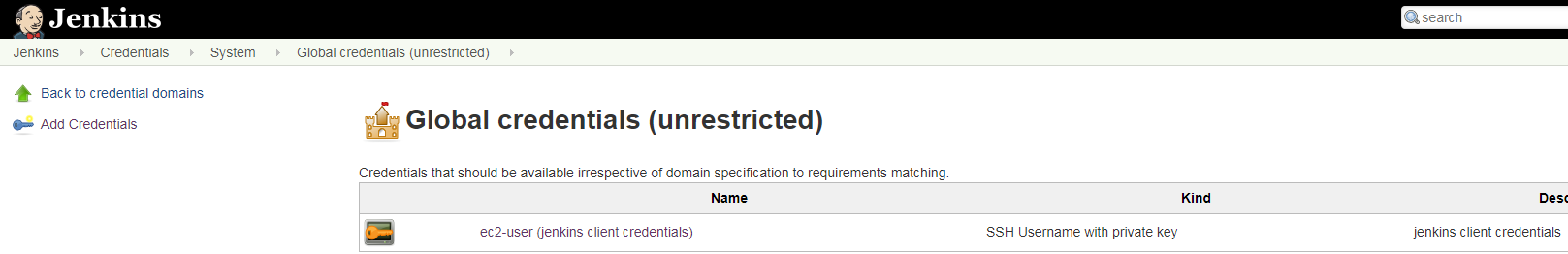
**Step3:** To access remote machine and execute commands on remote machine, need to be setting up credentials in Jenkins server.

Jenkins 🡪 Credentials 🡪 global 🡪 Add Credentials



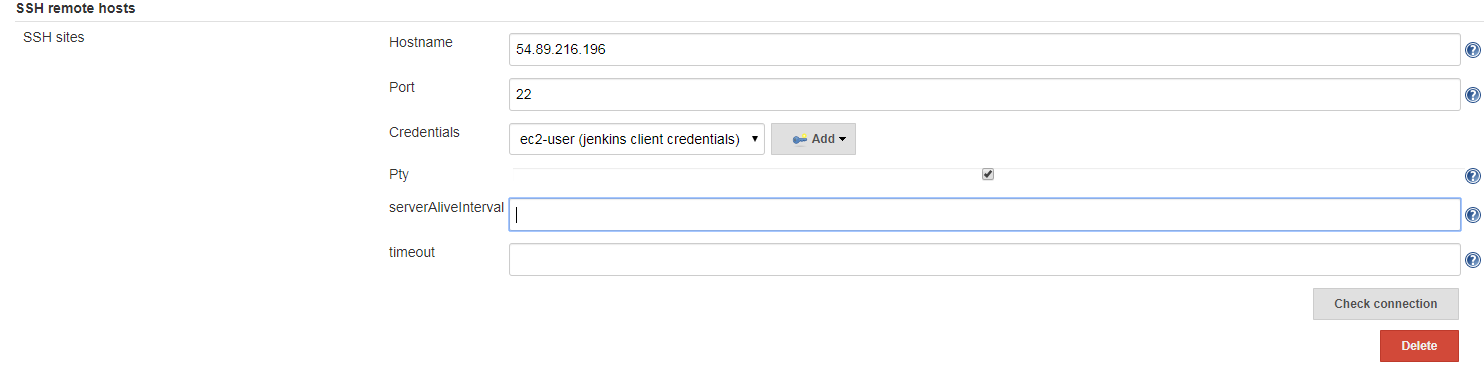
In the Private key box space, paste the pem file contents (Ex: Mumbai23May.pem).

Open the pem file in a notepad, select all the lines(Ctrl+a), copy it then paste it in that Private key box space.



**Step4:** Adding SSH sites,

Jenkins 🡪 Manage Jenkins 🡪 Configure System

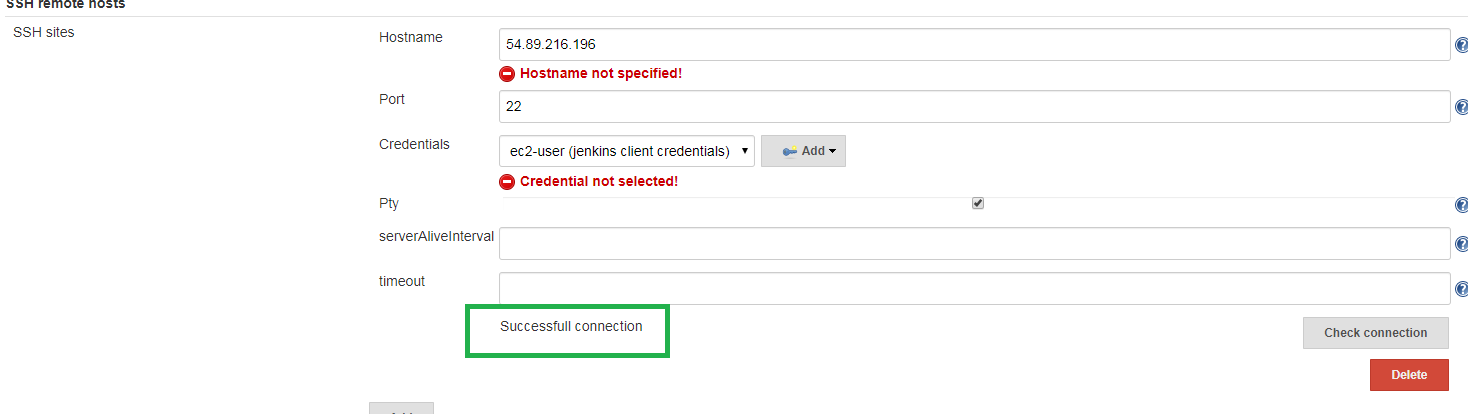


Hostname: 54.89.216.196 (Private IP of newly created remote machine)

Port: 22

Credentials: ec2-user (Jenkins client credentials)

Click on Check connection 🡪 you will see Successful connection

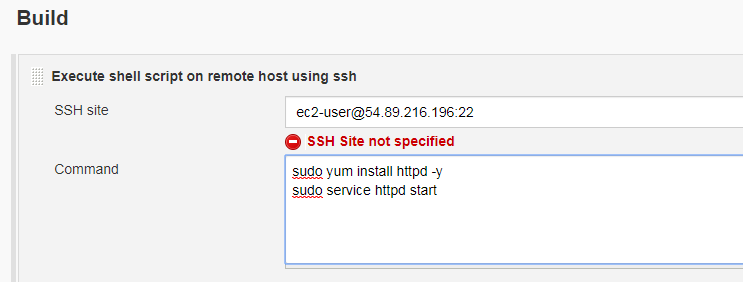


Click on the "save" button at the bottom.

**Step5:** Create a job to execute on remote machine.

Jenkins 🡪 New item 🡪 RemoteJob1 🡪 Freestyle project

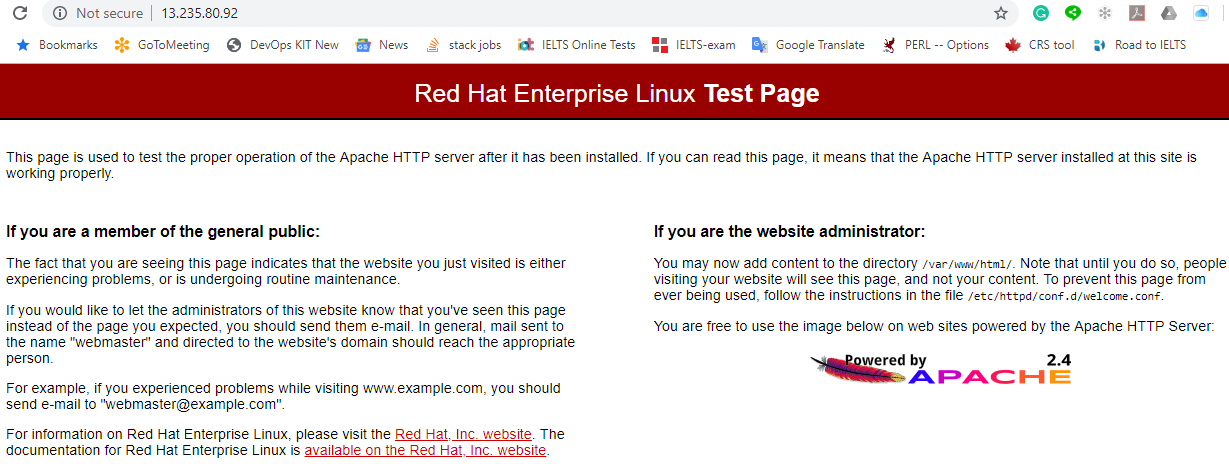
Create new job as remote\_job1



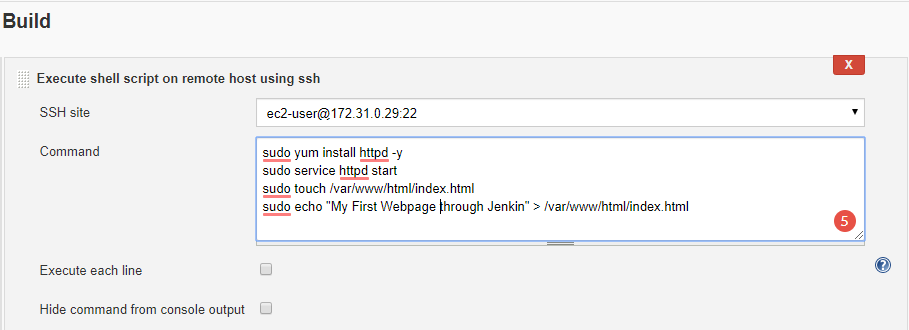
**Save => Build Now.**

It should be successful.

To verify, go to the browser, put the public IP of remote machine in the URL section. It will display http apache page like below.

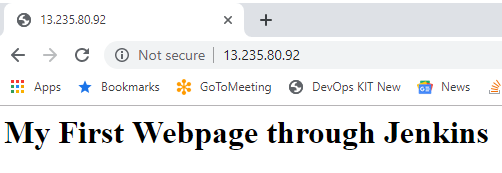


**Step6:** Now, try to create index file in remote machine location with some content.



* save and Build it again. The latest job should be successful (blue)

Refresh the URL and check it now.



**Exercise 3**

Here we will see a simple war file configuration in Jenkins.

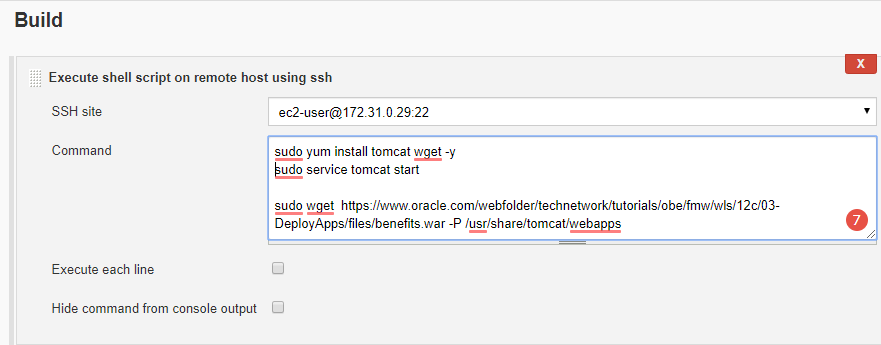
Now download the “benefits.war” from the oracle website through command and access it in webpage.

Command:

**# sudo yum install tomcat wget -y**

**# sudo service tomcat start**

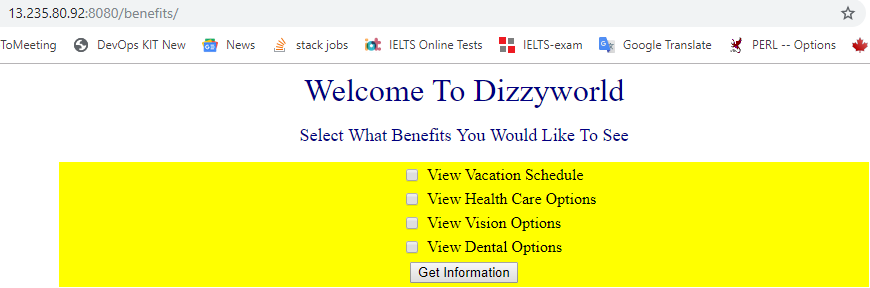
**# sudo wget https://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/wls/12c/03-DeployApps/files/benefits.war -P /usr/share/tomcat/webapp**

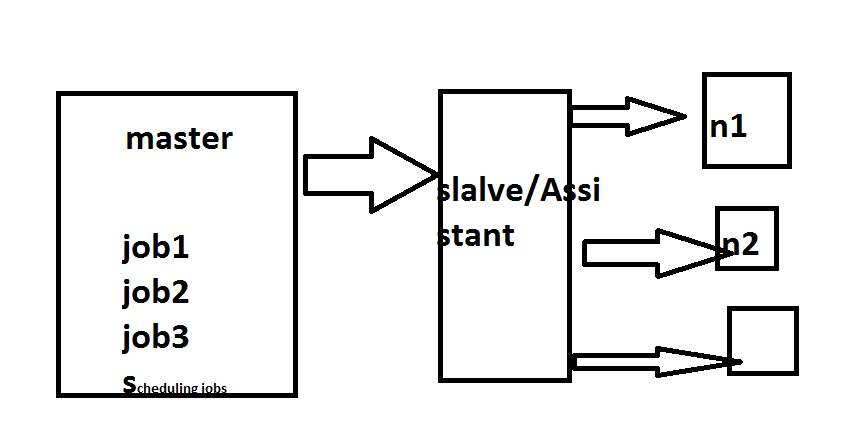


save -> Build now -> You can monitor the console output -> The build might be successful.

Now open the browser and check the benefits war file.

**Ex:** [**http://13.235.80.92:8080/benefits/**](http://13.235.80.92:8080/benefits/)



**===========\*\*\*\*\*\*\*\*\* Master & Slave setup \*\*\*\*\*\*\*\*\*=============** 

**Create one RHEL machine for slave setup.**

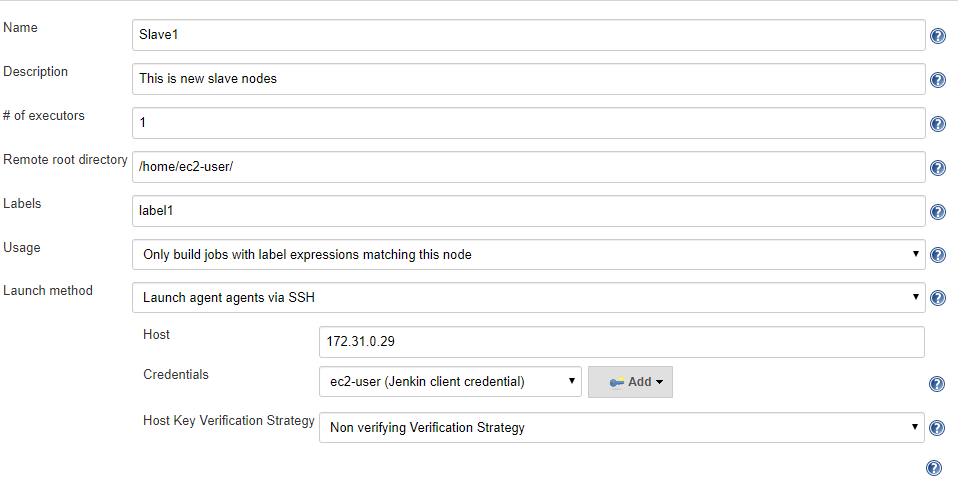
**Here we use same credentials (.pem file) for practice, else add the credentials first for the new slave machine.**

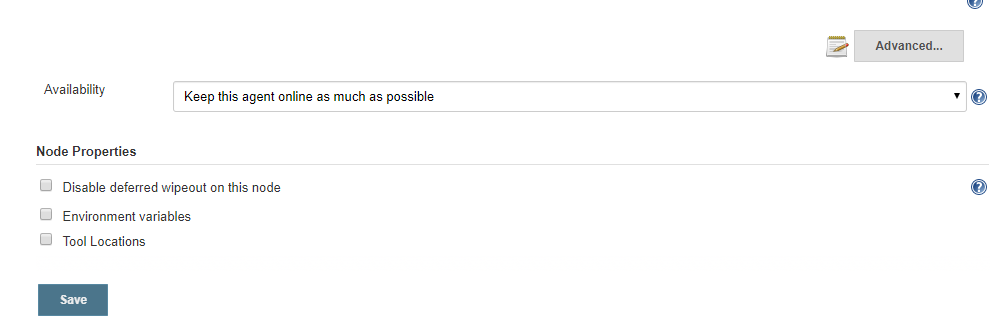
**Setup the credentials for slave machine in jenkins:**

Jenkins --> Manage Jenkins --> Manage Nodes

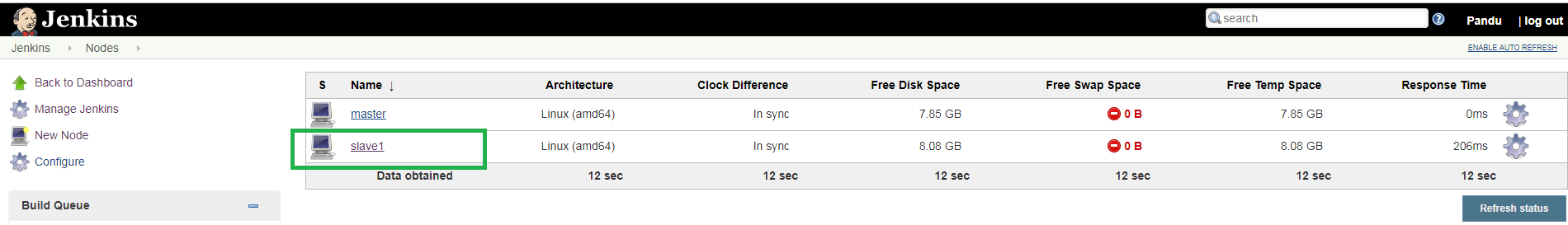
Click on New node --> Provide Node name (Slave1) --> Select Permanent Agent option.



****

****

Provide the details properly and Save.

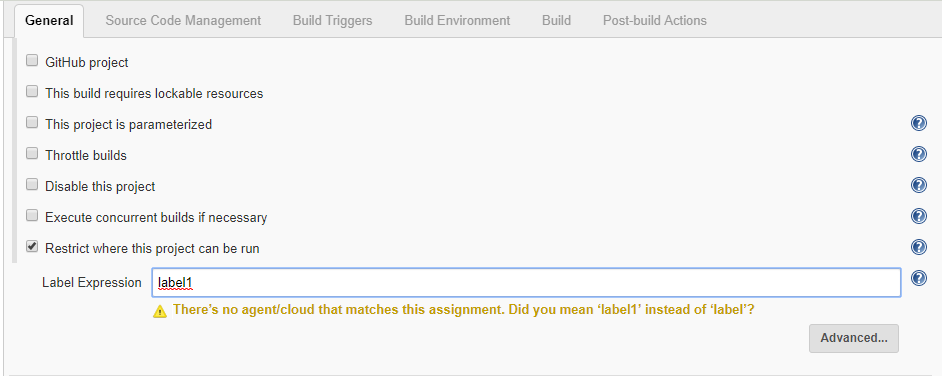


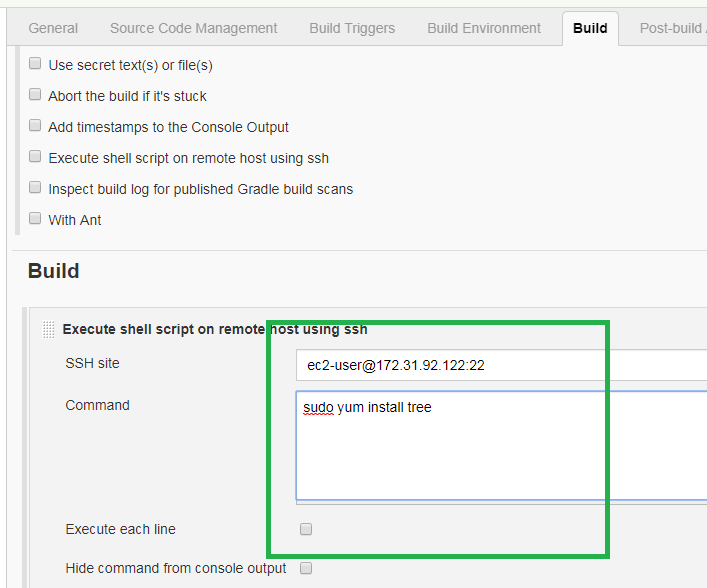
Now, create a job for slave machine.

Jenkins--> New Item-->

Enter an item name "Slavejob1"

Freestyle Project --> click on ok





save and Build Now.

So far we have ran job on slave machine.

**========= Parameterized Tasks ==========**

**Tasks1:**

Step1: Create a new job "demo\_job5"

**In General section**, check the option "This project is parameterized"

Select "Add Parameter" option as "string parameter"

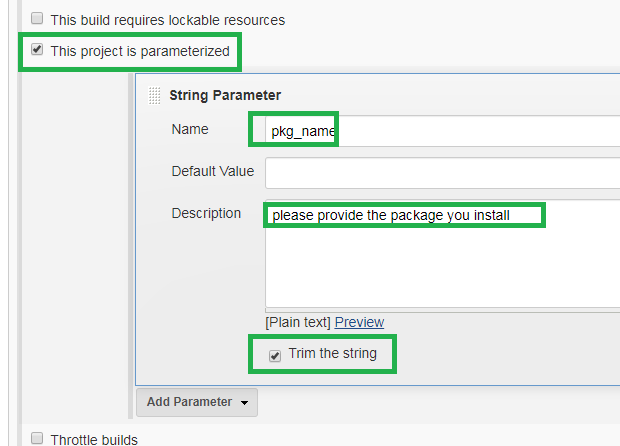
Provide name and description

Example:

Name pkg\_name

Description "please provide the package you install"

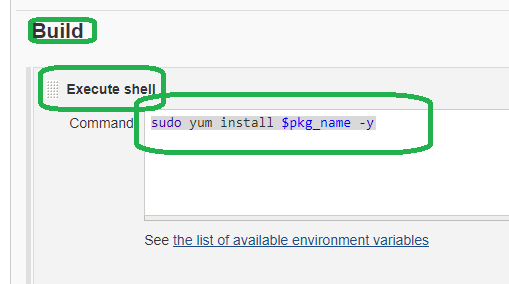
Select "Trim the string"



Now, go to “Build” tab and select "Execute Shell"

Provide this command: "sudo yum install $pkg\_name –y”

Save and build.



Click on “Build with Parameters”

* the job should run successfully.

**Tasks2:**

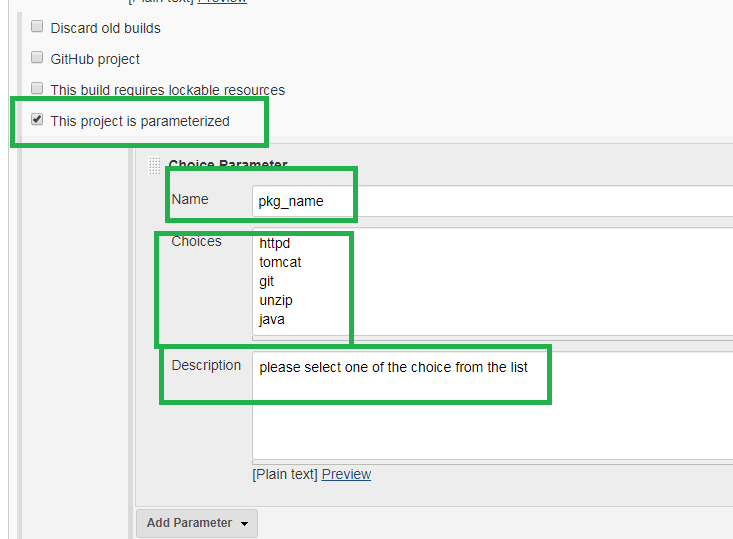
Step1: Create a new job "demo\_job6"

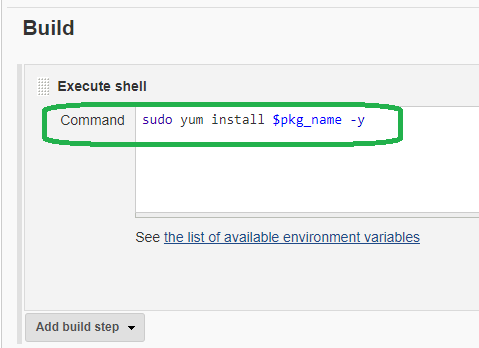
**In General section**, select option "This project is parameterized"

Select "Add Parameter" option as "choice Parameter"

Save and Build.

Provide details as below





[Build with Parameters](http://18.212.167.139:8080/job/demo_job6/build?delay=0sec)

* the job should run successfully

**Home Assignment:**

Q. How to restore jenkins?

1. Take backup of jenkins folder

* # **cp /var/lib/jenkins /tmp/jenkins\_ddmmyy**

1. Stop jenkins

* # **service jenkins stop**

1. Uninstall jenkins

* # **yum remove jenkins**

1. Delete jenkins directory

* # **rm –rf /var/lib/jenkins**

1. Install jenkins tool in Remote host

* Commands are available in this note at the starting point.

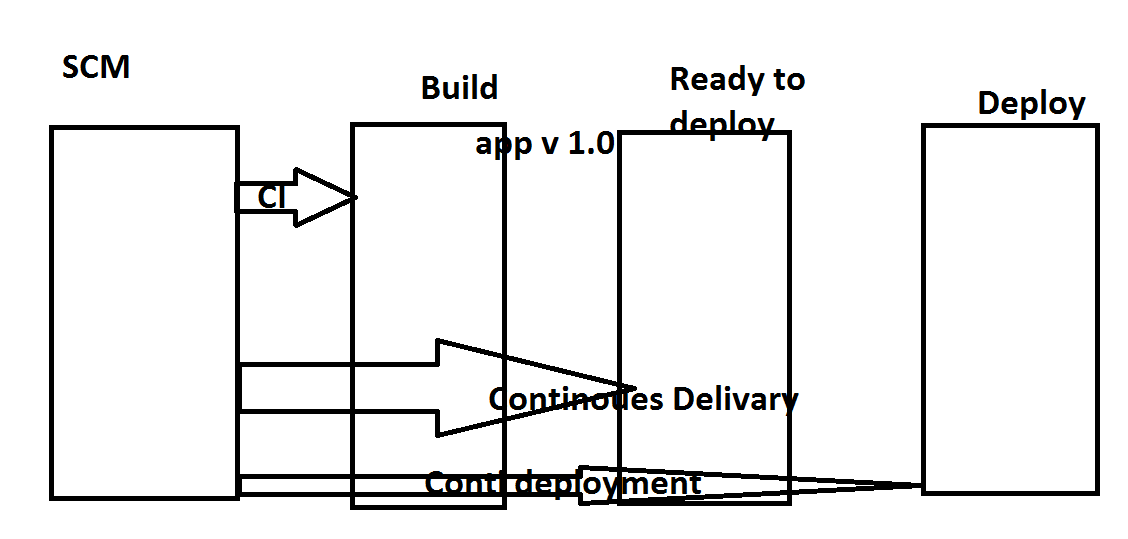
1. Copy the jenkins directory from host to remote host

* # **scp /tmp/jenkins\_ddmmyy ec2\_user@RemoteHost\_publicIP**

1. Open the jenkins page in web browser.

* **publicIP:8080**

**===========CI/CD Pipeline=========**



We can create pipeline job in two ways:

* Pipeline code
* Build pipeline.

Pipeline code: It uses Groovy language.

We need to write code in groovy language and in stages mode.

**Exercise1:**

Jenkins --> New item --> Select pipeline and name something as “pipelinejob2”

Build pipeline plugin

**Step1:**

Jenkins-->Manage Jenkins-->Manage Plugins-->

In filter section, search for “build pipeline”

Select “Build Pipeline Plugin”

Install without restart

Create multiple jobs (3 now) like as below

job1: echo "continuous integration getting done"

job2: echo "continuous delivery getting done"

job3: echo "continuous deployment getting done"

go to each job "Configure" --> Build Triggers --> Build after other projects are built -->

Configure each job should trigger one after another job, then run the job.

It will run successfully all jobs.

**Home Exercise:**

Configure build periodically and check once it is running or not periodically.

**Github webhooking:**

Login into github with your login credentials.

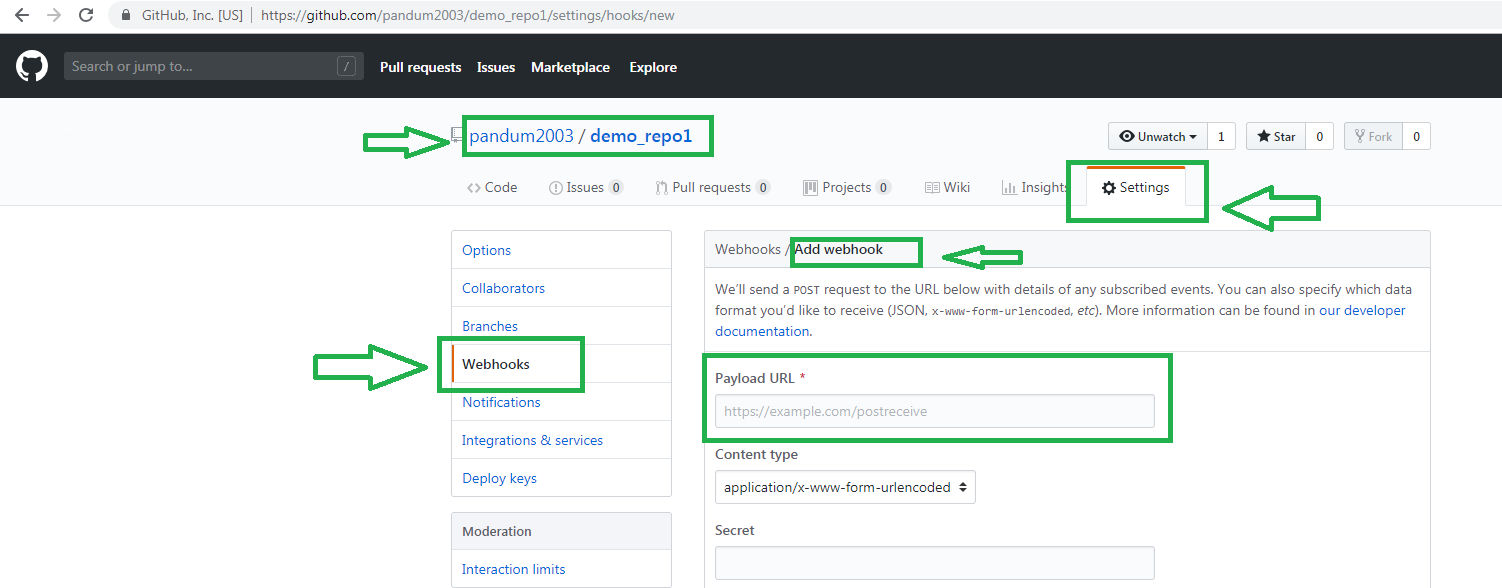
Create one repo as demo\_repo1

Create one file as file2 in above repo and commit it.

**Step1:**

Go to repo on your github --> settings (top right)

--> webhooks(left side) --> addwebhook --> provide jenkins server url and select application/json, select everything option

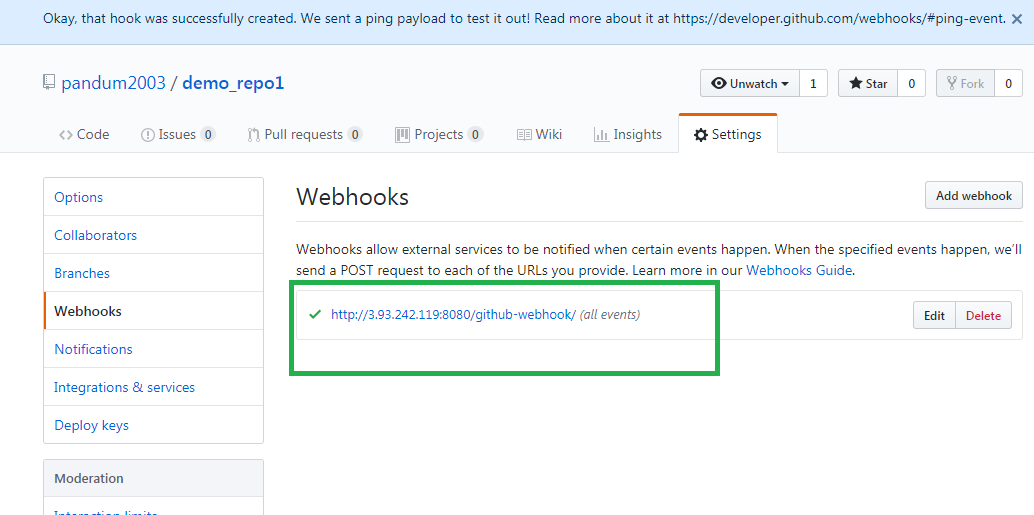


Note: In Payload URL don’t forget to add “github-webhook/” after the Jenkins URL.

Ex: <http://13.233.155.148:8080/github-webhook/>

Which event

Send me everything

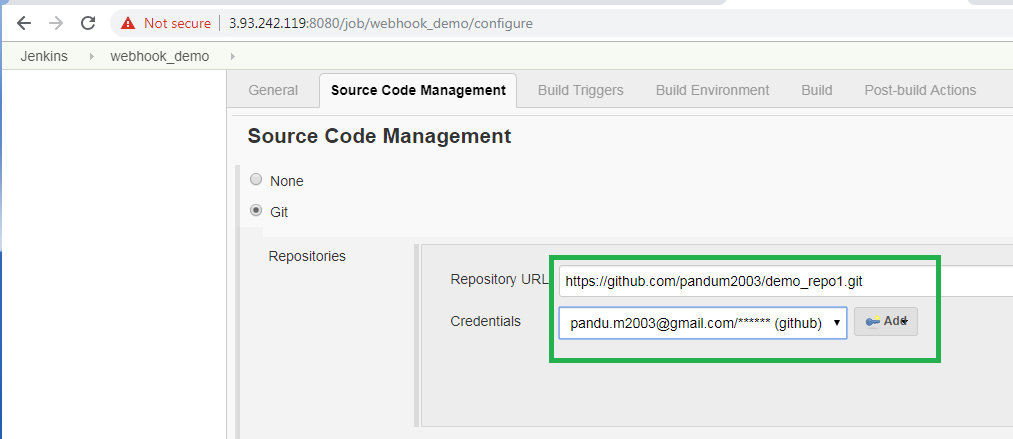


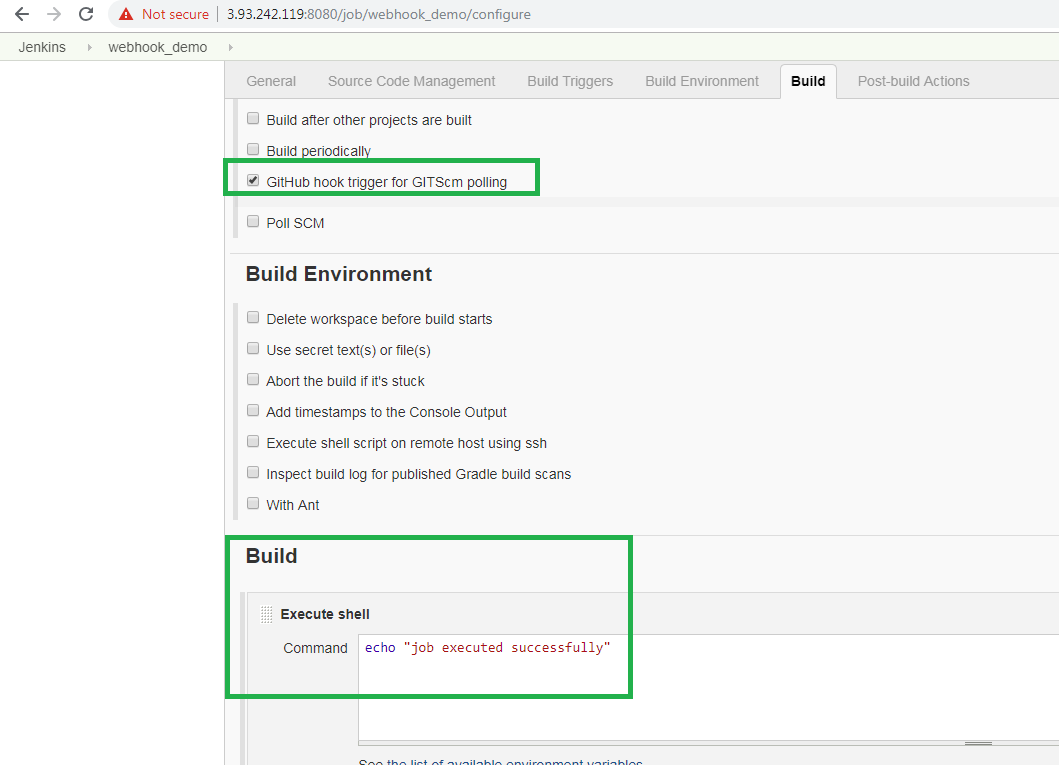
**step3:** Create job to use with github-webhook

Create a new job

jenkins-->new-item--><name as: webhook\_demo>

Provide gthub details URL and credentials





Save the configuration

Now, change the github repo file.

Once you modify the file, build will trigger automatically on jenkins.