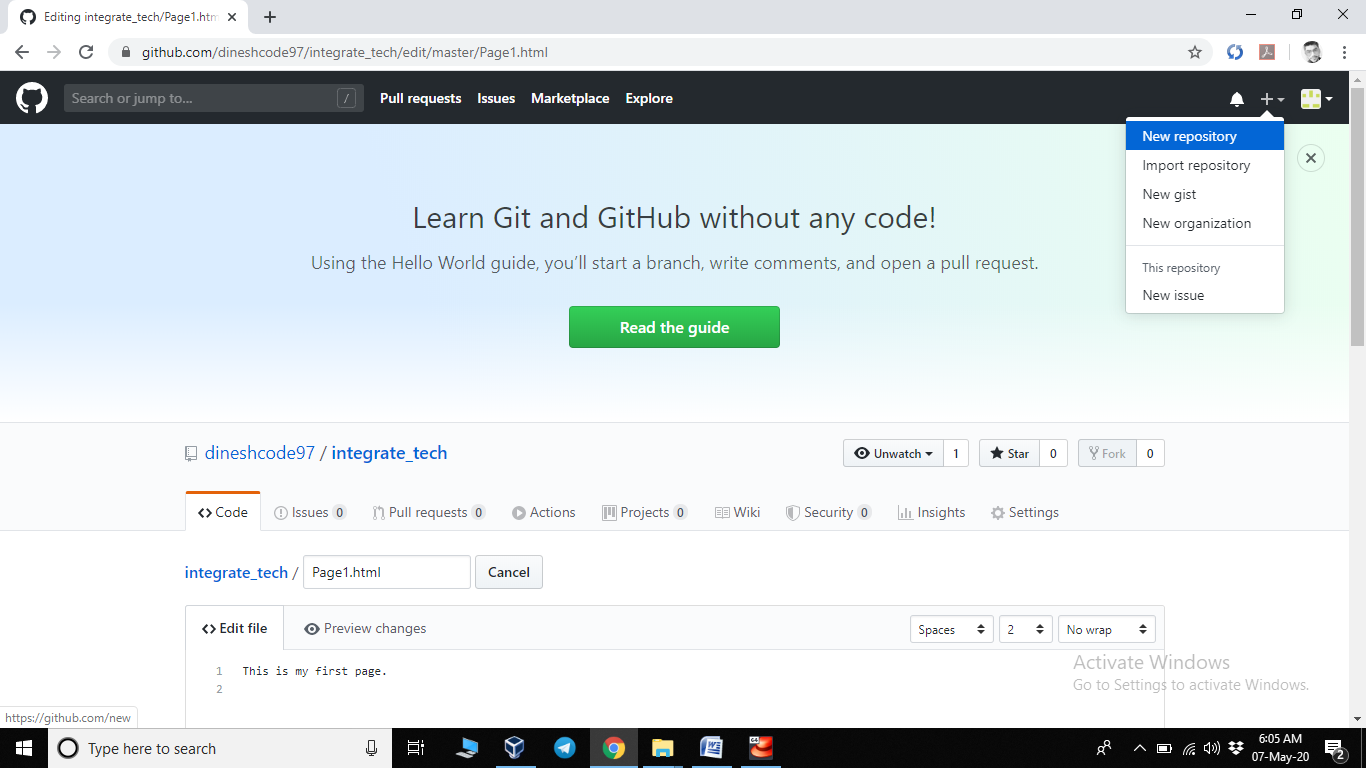
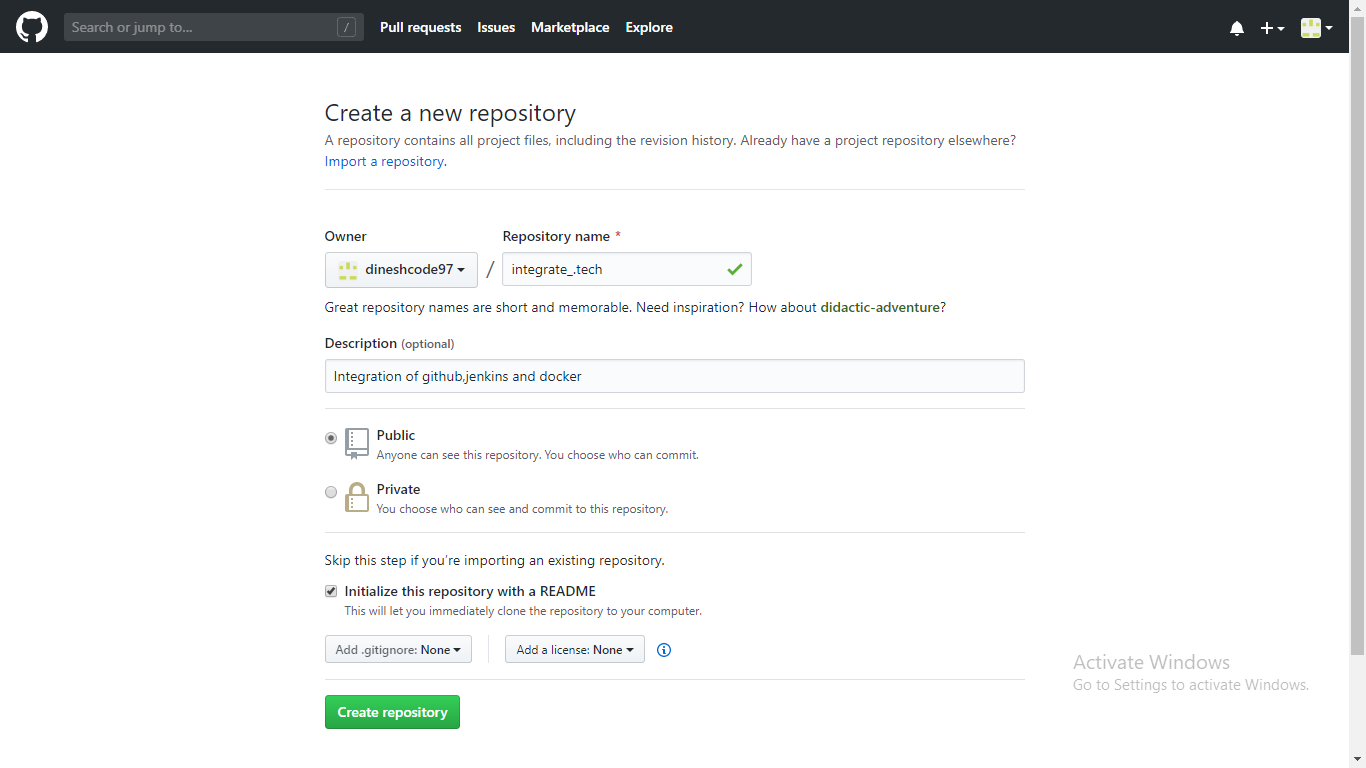
Documentation

Github:

* Create a new repository



* Give name to repository and add description(Optional)
* Select Initialize READ.md
* Click on create repository



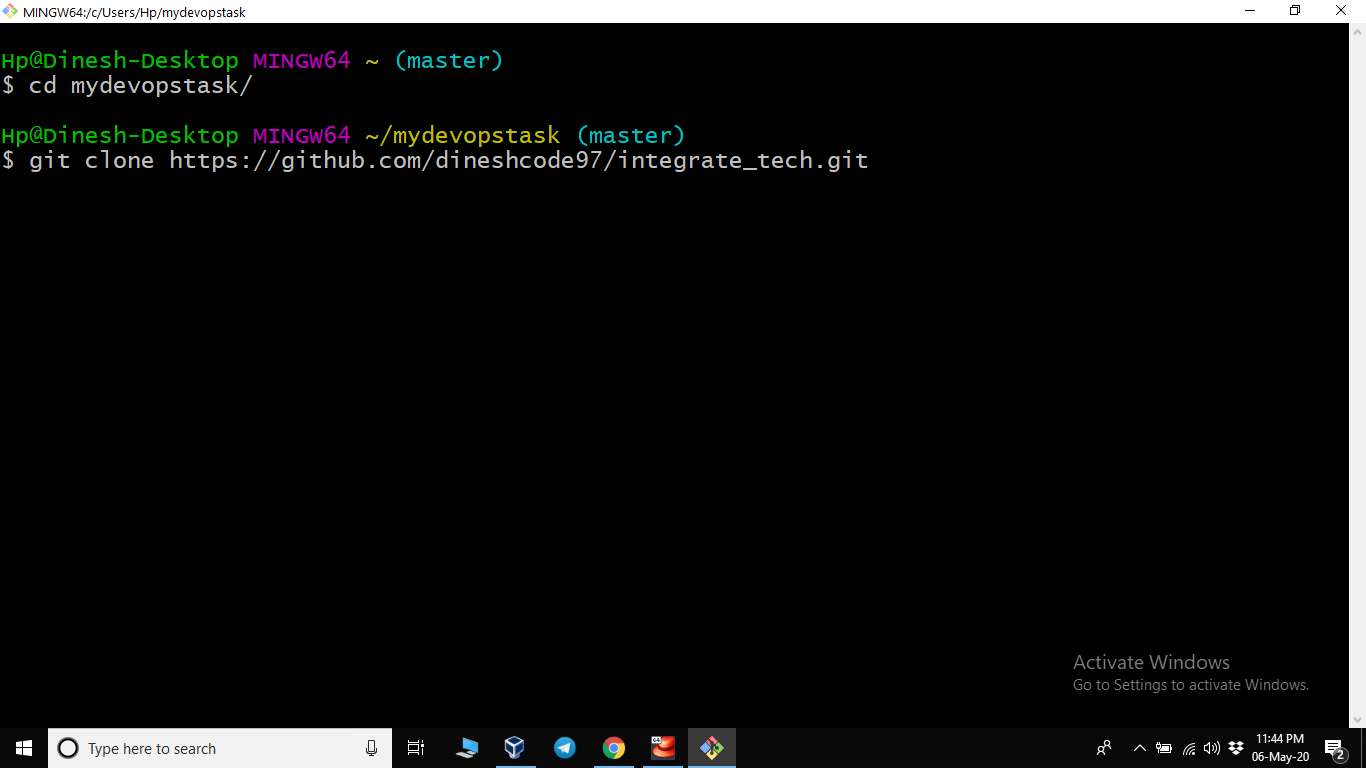
Git:

* Change Directory where you want to clone github repository using

cd mydevopstask/ or create a new directory using mkdir dirname

* Clone using git clone giturl

Eg. git clone <https://github.com/dineshcode97/integrate_tech.git>



* Create a hook

cd .git/hooks/

cat > post-commit

#!/bin/bash/

git push

press ctrl+z

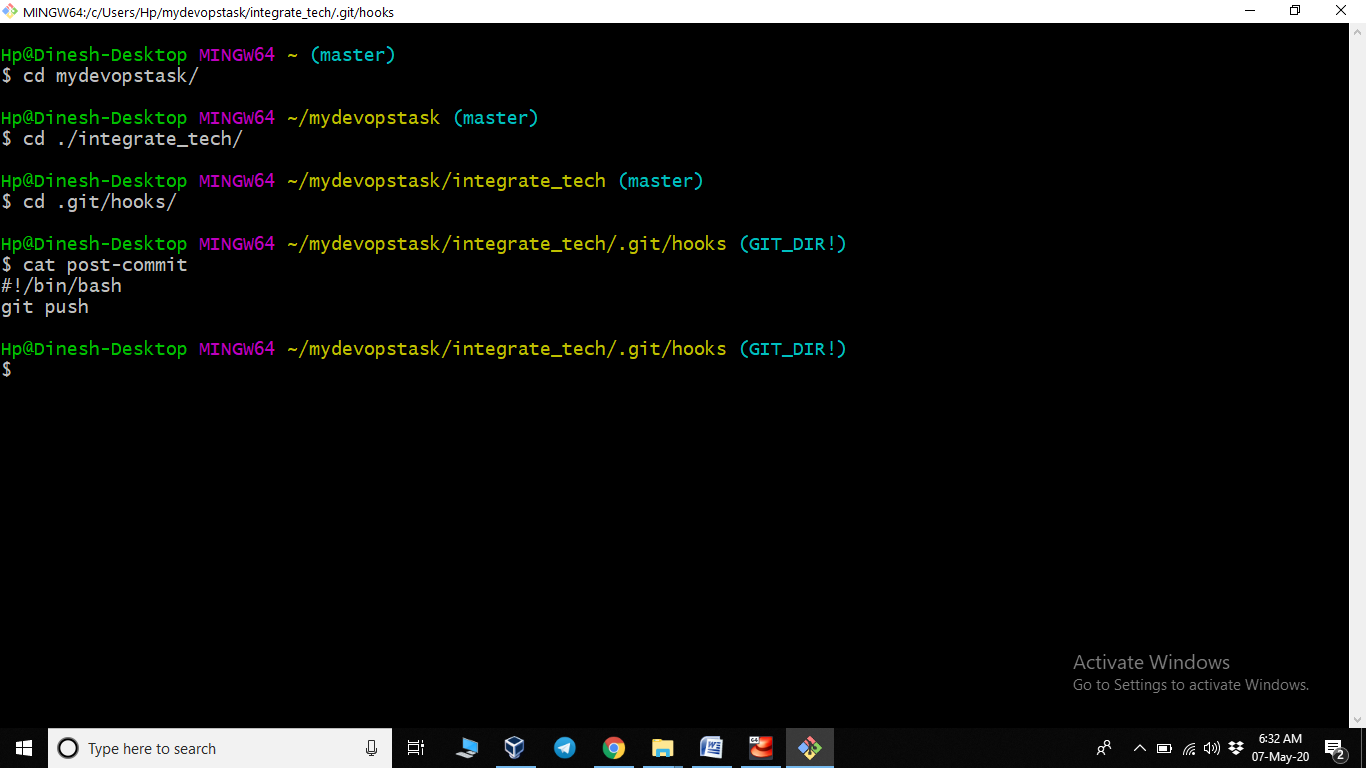
* Add and Commit some files

git add Page1.html

git commit –m “First Page” Page1.html

Note:- Now your hook automatically Push files to github

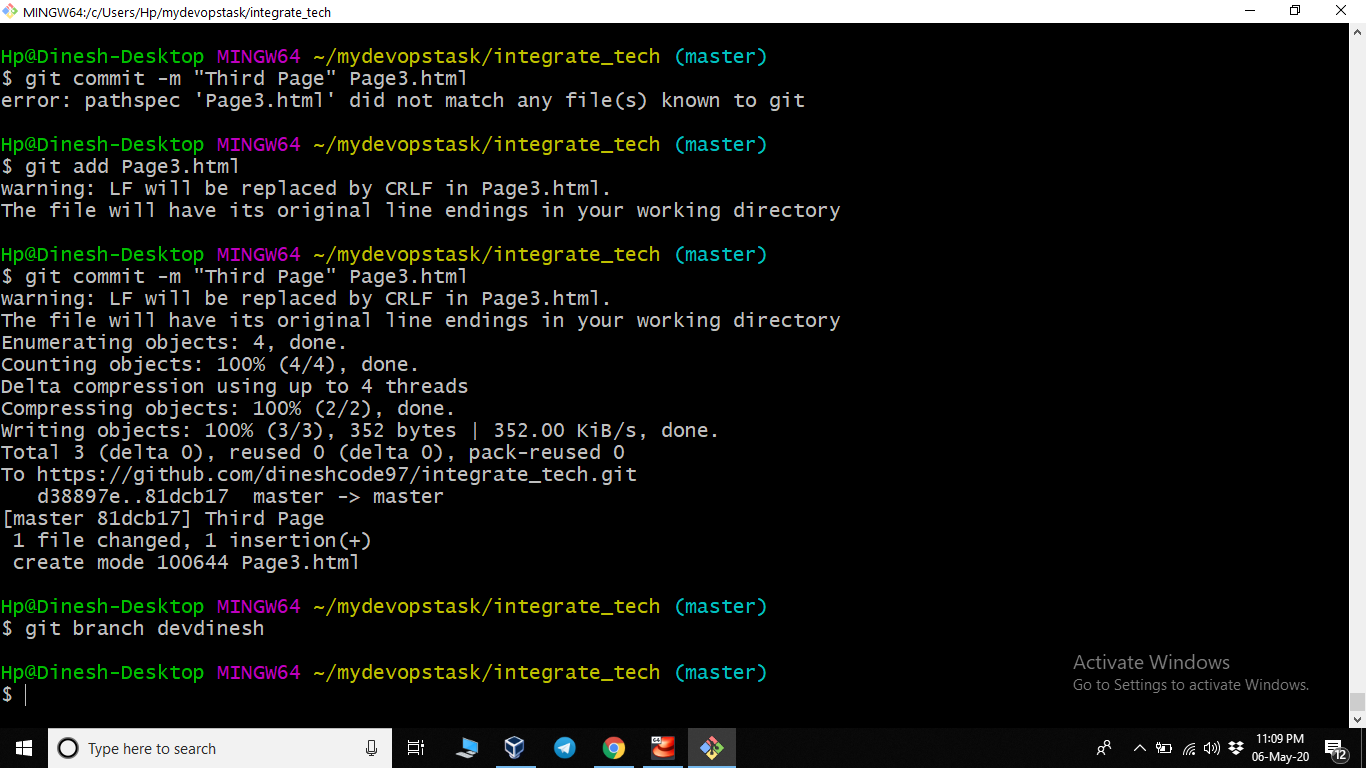
(I created the hooks before so follow above steps to create hook)



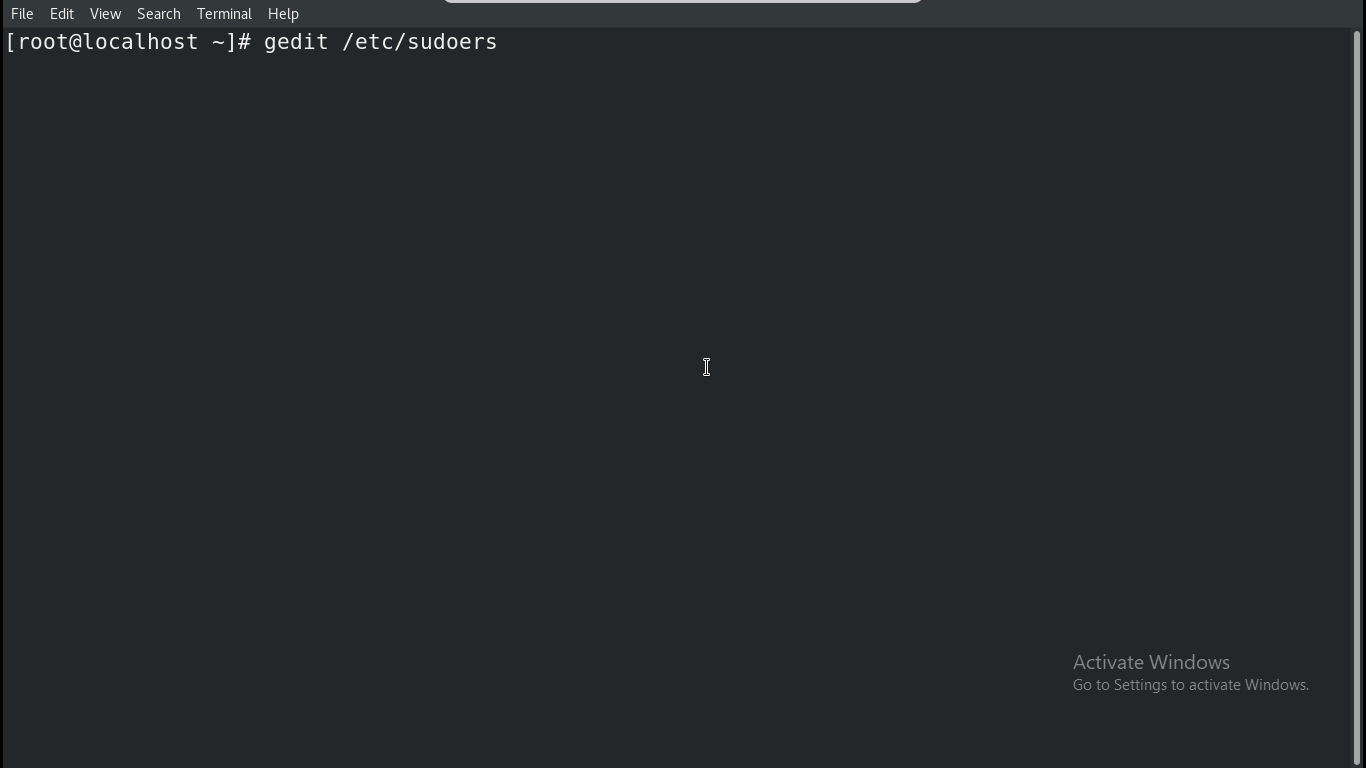
* Create a new branch using git branch branchname

->git branch devdinesh

Linux(VM):



* Start Your VM and open the terminal
* Enter command gedit /etc/sudoers



* Text editor open the file sudoers

By going down you will find

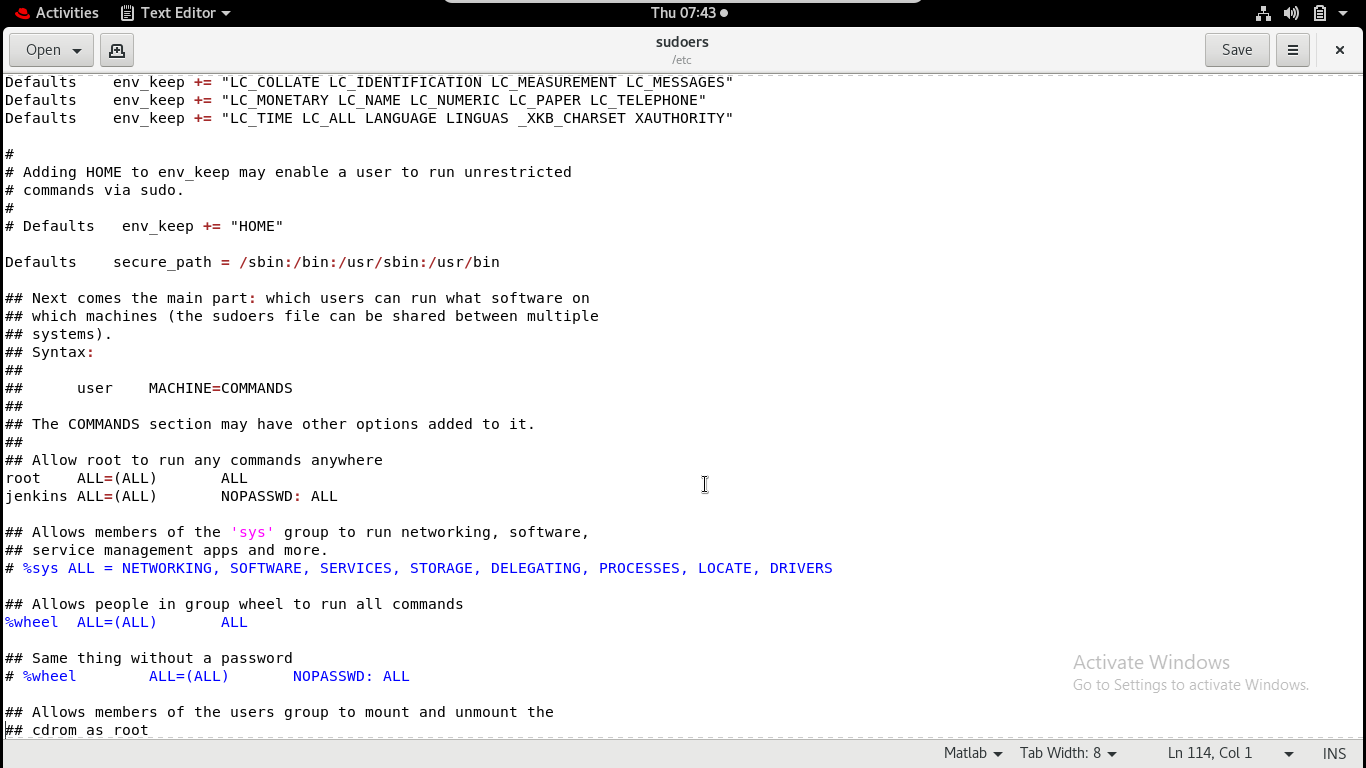
#Allow root to run command anywhere

root ALL=(ALL) ALL

* Add next line there

jenkins ALL=(ALL) NOPASSWD:ALL

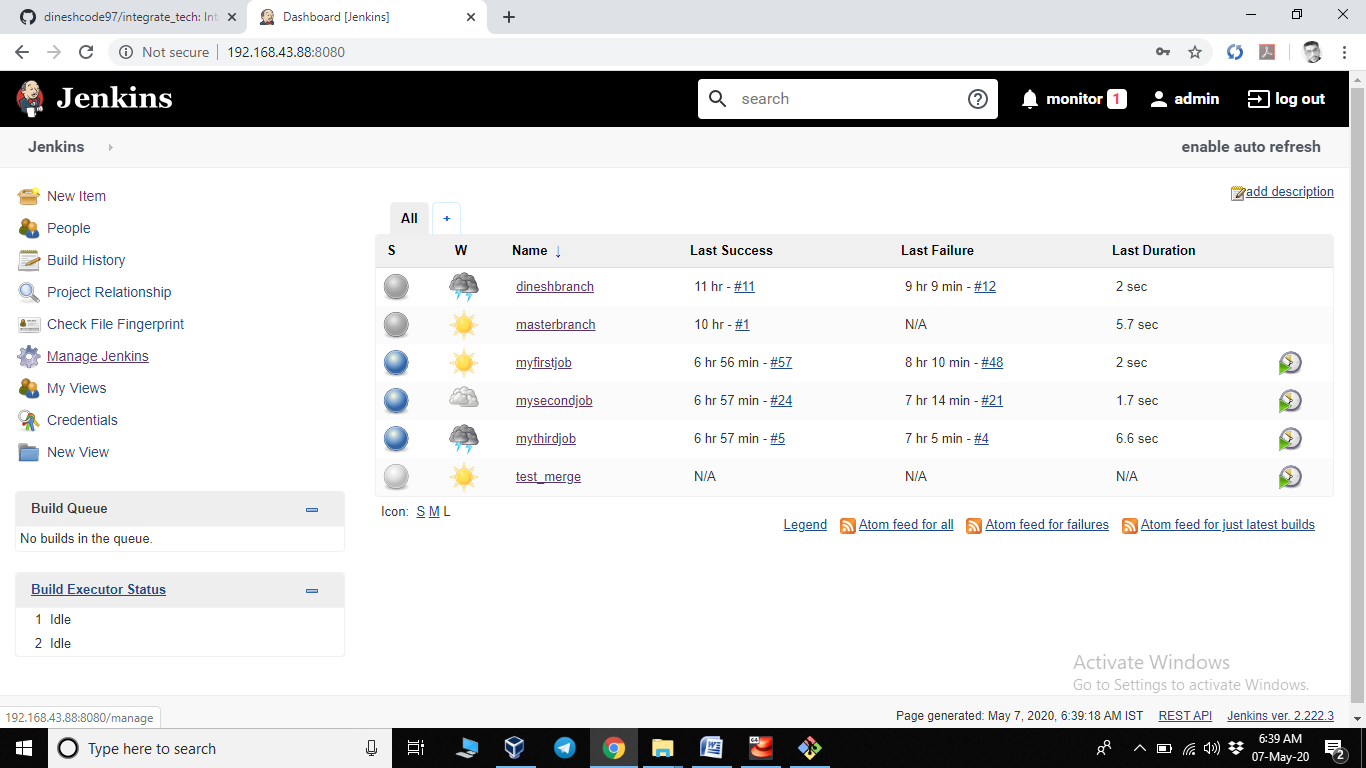
* Save the file and then close the file

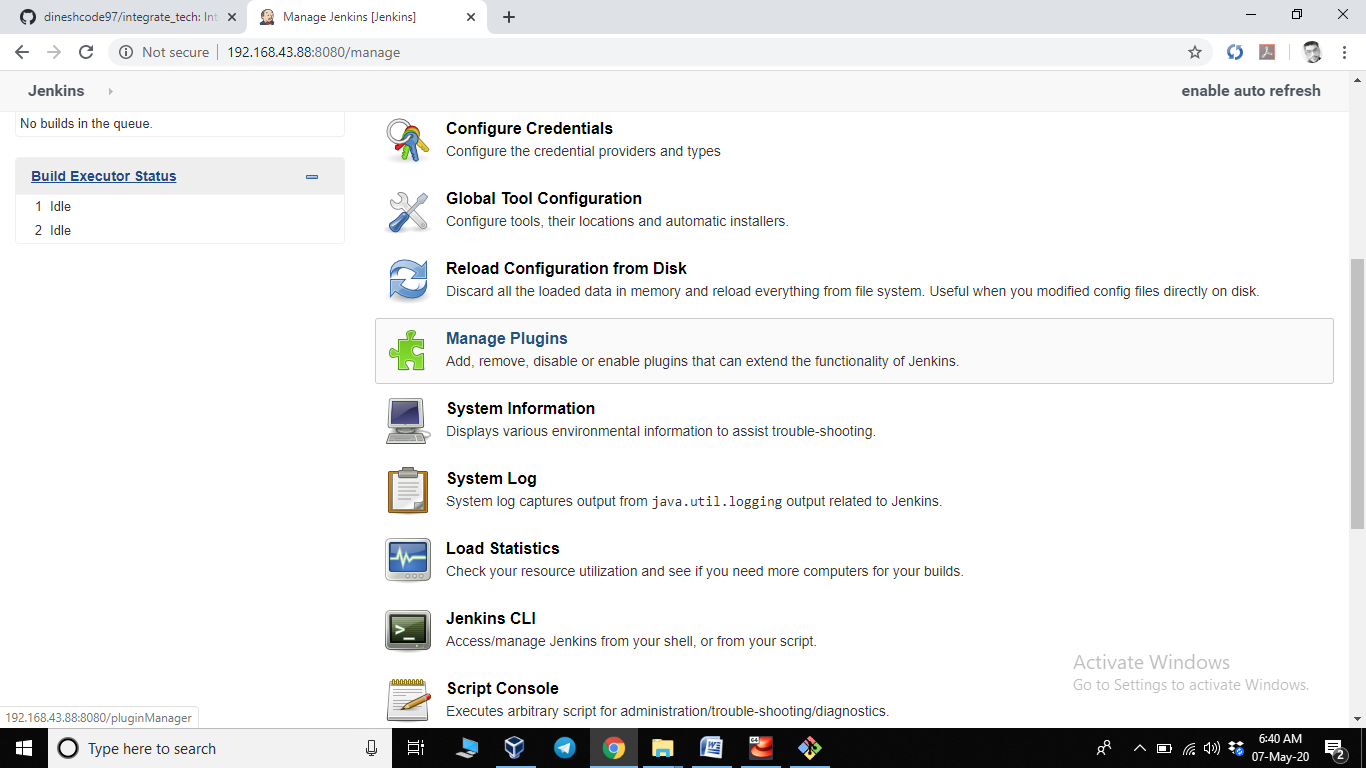


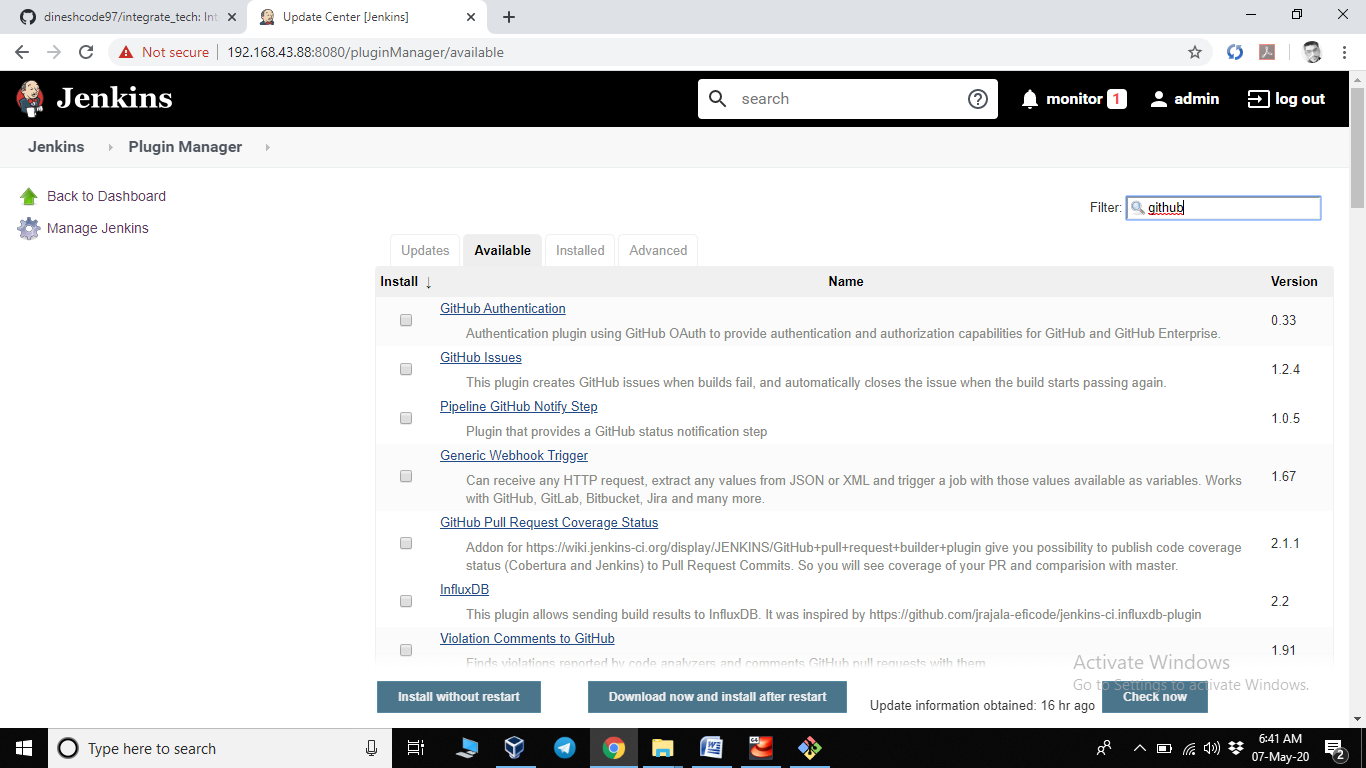
Jenkins:

* Goto Jenkins open your browser use # yourIP:8080
* Enter username and password
* Goto Manage Jenkins->manage plugin-> search Github

Install pulgin which integrated github and Jenkins

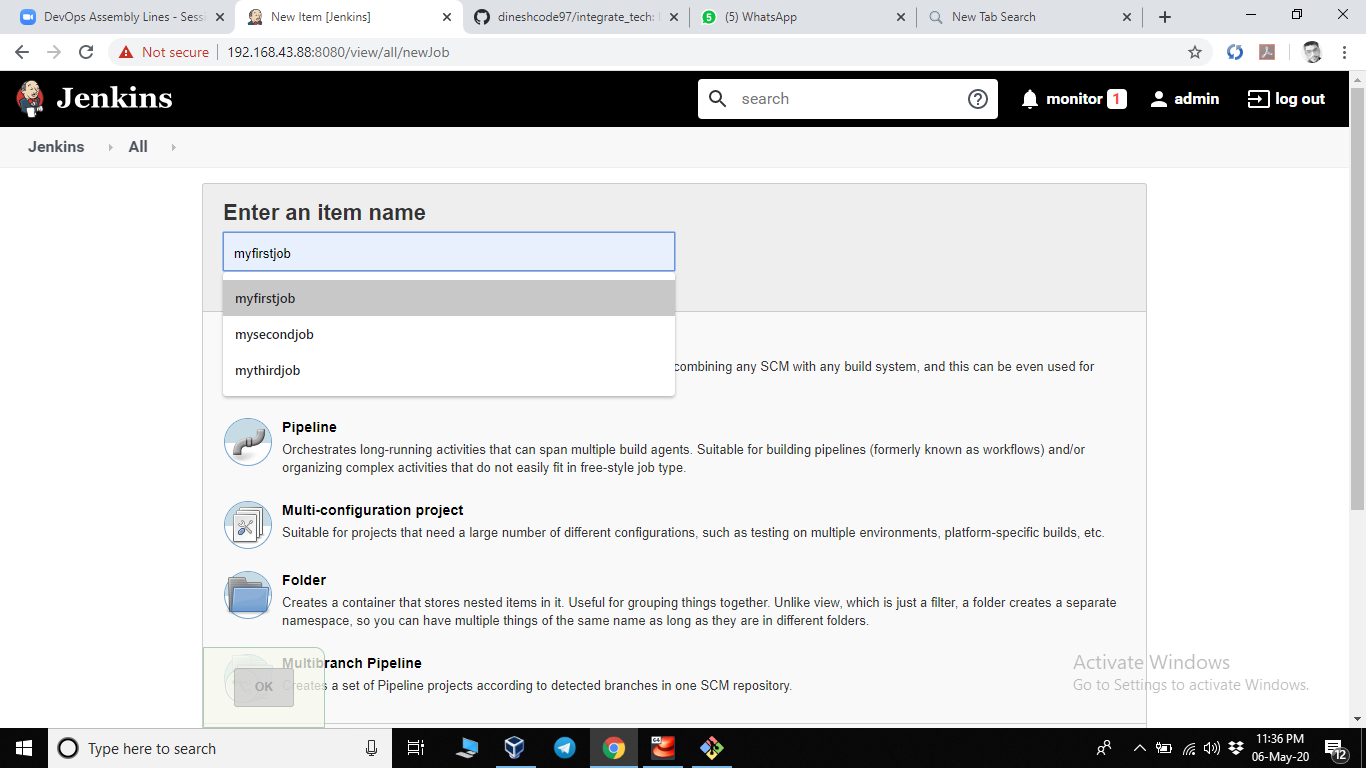




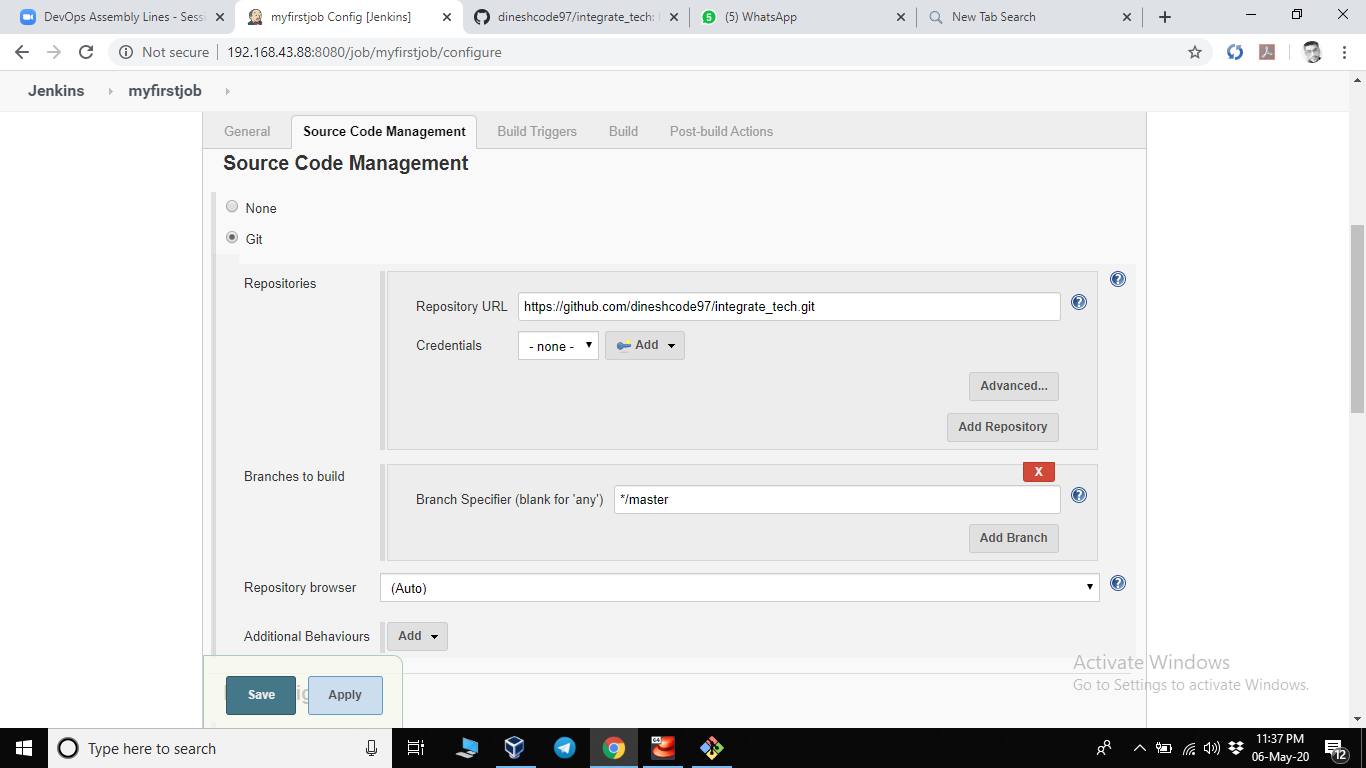


* Create Job1 by clicking new item->Enter job name->select freestyle project

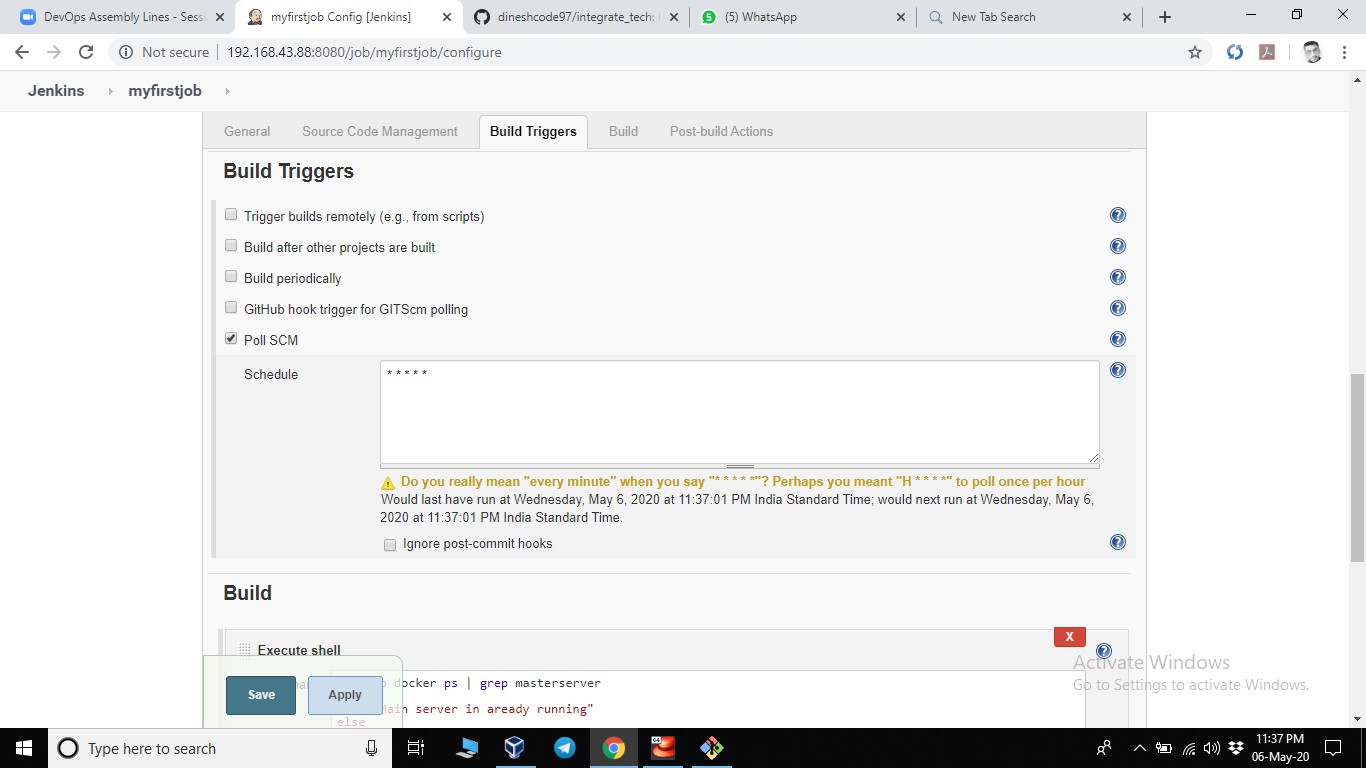
(job name: myfirst job)



* Click on your job1 1->configure
* Goto source code management and select git
* Copy your github repository URl and paste in Repository URl of job



* Goto bulid triggers and select poll scm fill \* \* \* \* \*(to run every min)



* Goto build and select execute shell
* Add this instruction or code in box
* If sudo docker ps | grep masterserver

then

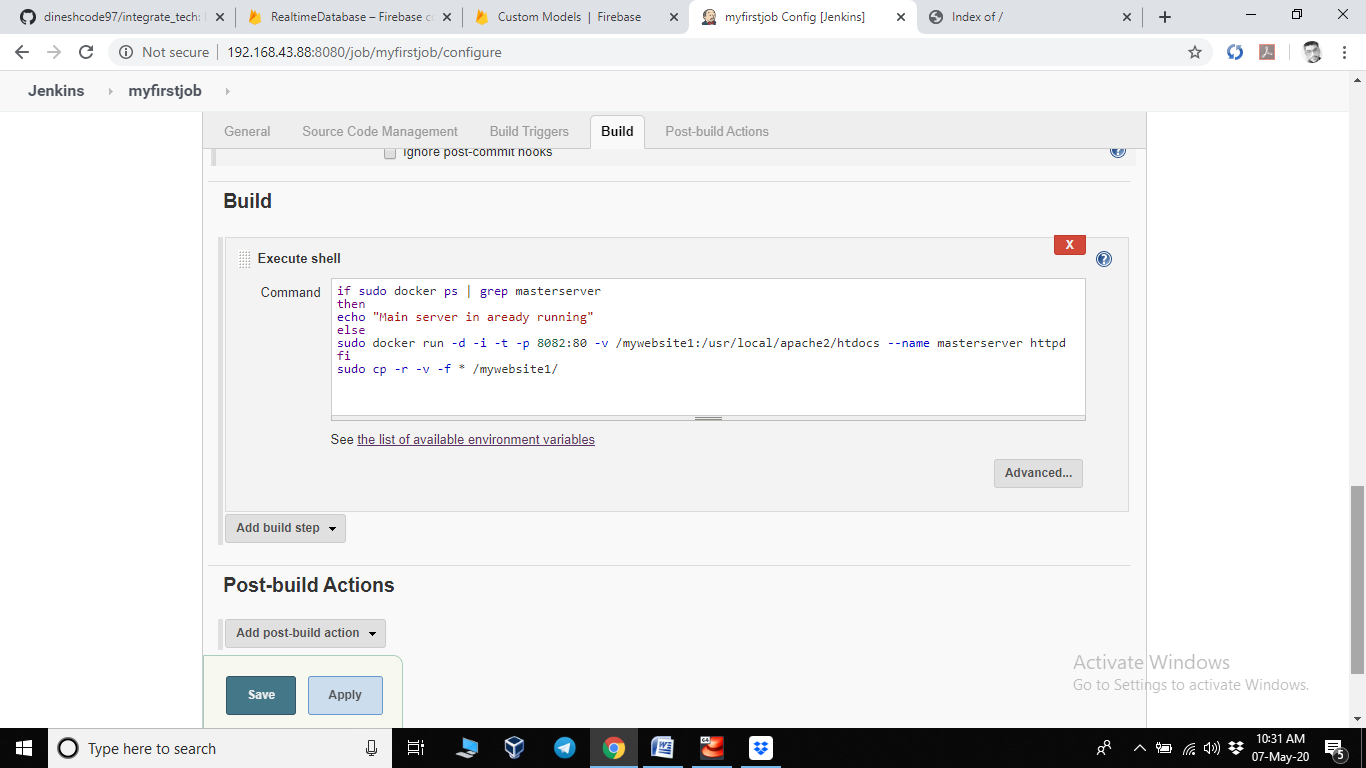
echo “Already running”

else

sudo docker run –d –i –t –p 8082:80 –v /mywebsite1:/usr/local/apache2/htdocs –name masterserver httpd

fi

* sudo cp –r –v –f \* /testingsite/



In this code if master server is running already it execute else part else

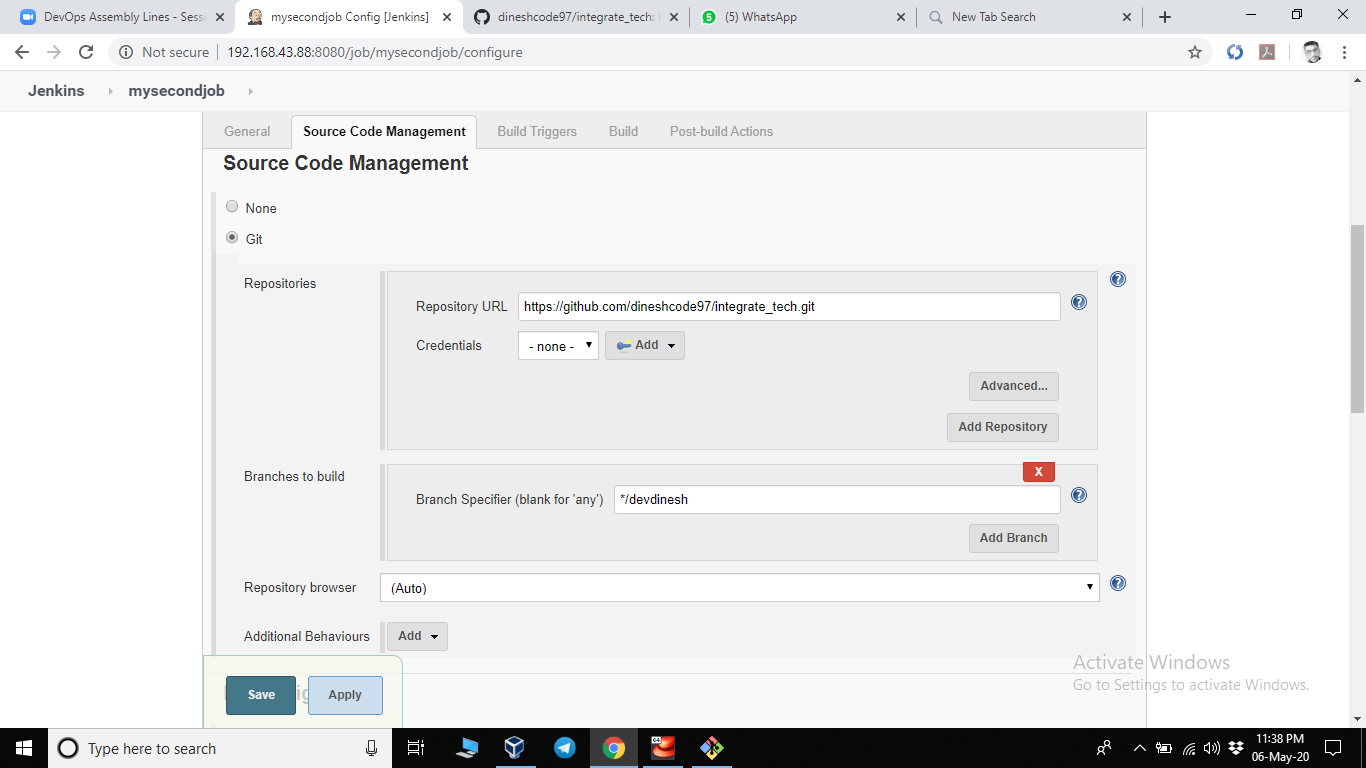
Its start a web server mount it in a dir and pat the port(expose port)

JOB2:

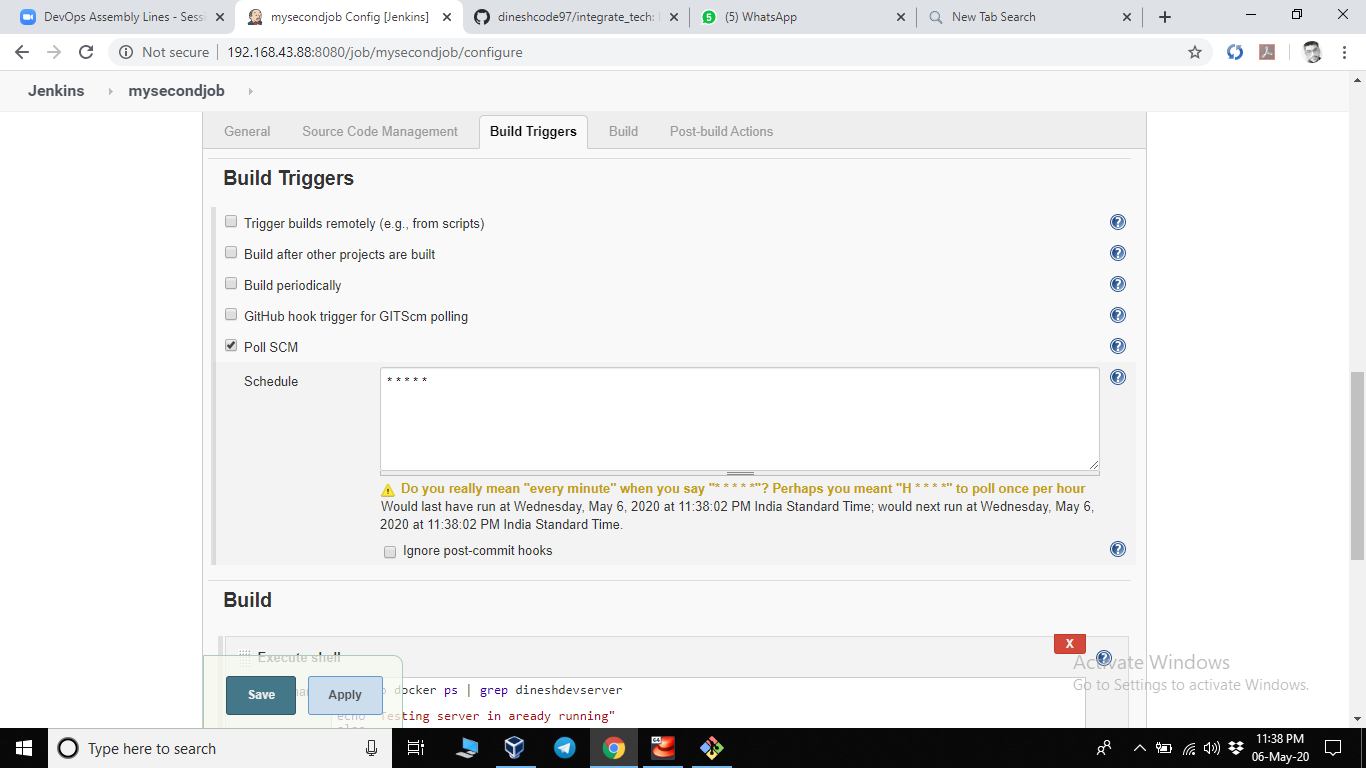
* Same process to create a job as in job1
* Give another name to your job 2(like mysecond job)
* Create and go to configure
* Same as job 1 goto source code management section select git

And enter your github url in repository url.

* Give your 2nd branch in brances specifier which you create in git



* Goto bulid triggers and select poll scm fill \* \* \* \* \*(to run every min)



* Goto bulid and select execute shell
* Enter the code in box
* If sudo docker ps | grep dineshserver

then

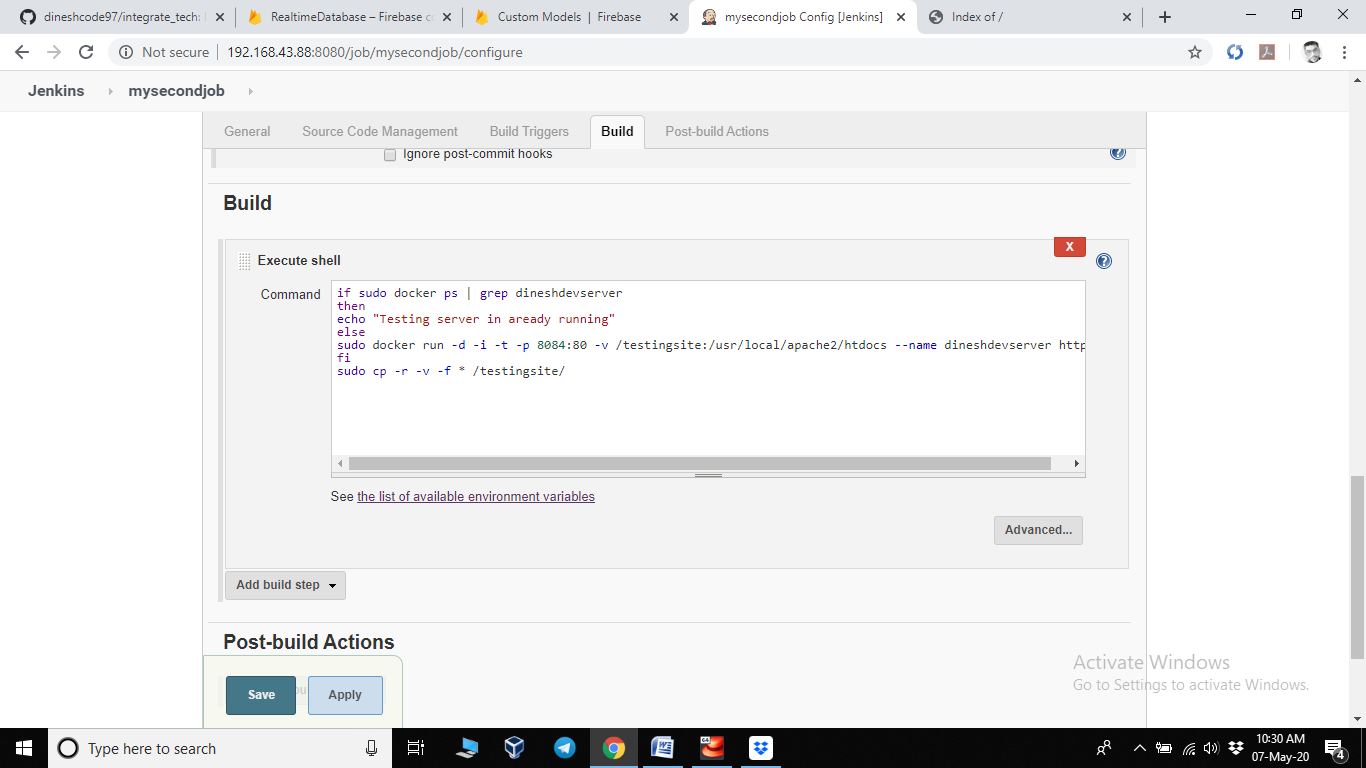
echo “Already running”

else

sudo docker run –d –i –t –p 8084:80 –v /testingsite:/usr/local/apache2/htdocs –name dineshserver httpd

fi

sudo cp –r –v –f \* /testingsite/



In this code if dineshserver is running it will not run again else

It runs a httpd server with name dineshserver

And mount all files from /usr/local/apache1/htdocs to /testingsite/

JOB3:

* Same process to create a job as in job1
* Give another name to your job 3(like mythird job)
* Create and go to configure
* Same as job 1 goto source code management section select git

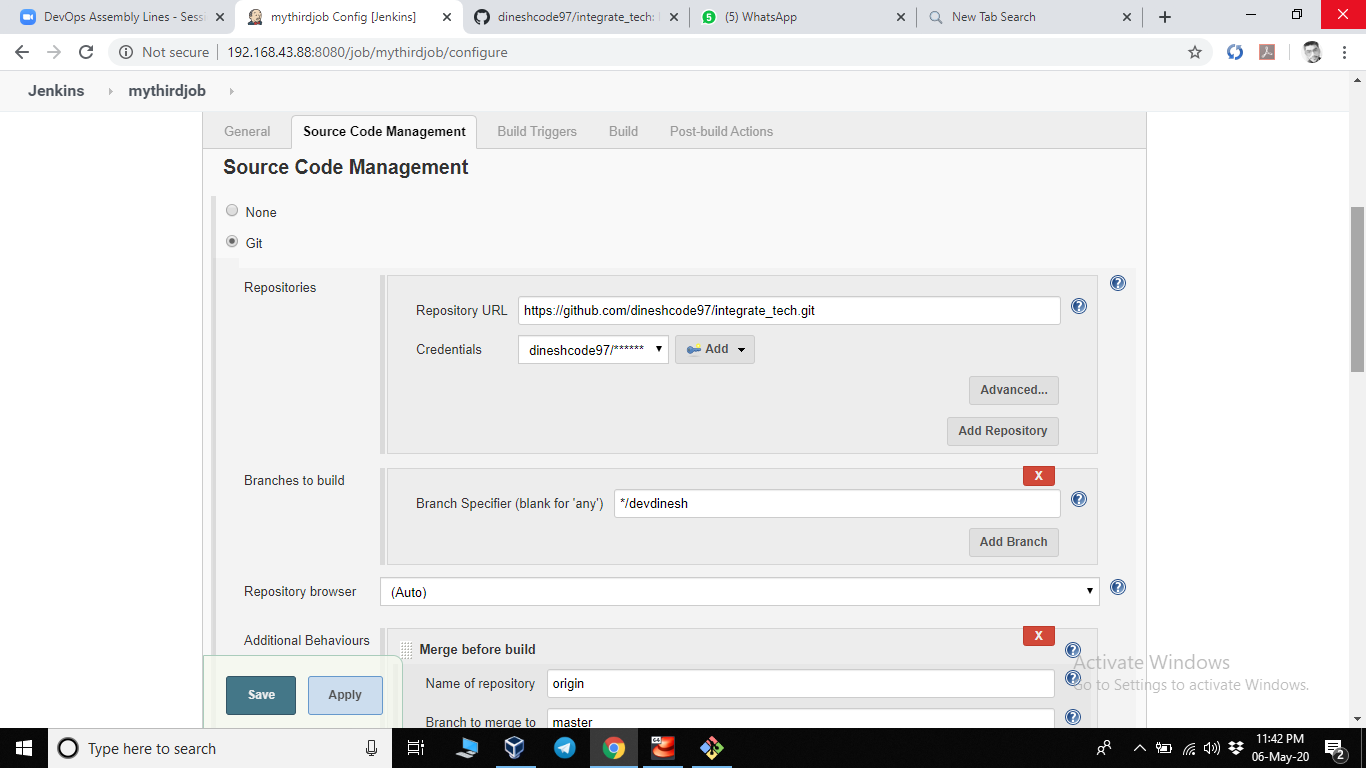
And enter your github url in repository url.

Give your 2nd branch in brances specifier which you create in git (eg. devdinesh)

* Below of Repository url there is a box of credeintials.

Click on add button their and a box appears

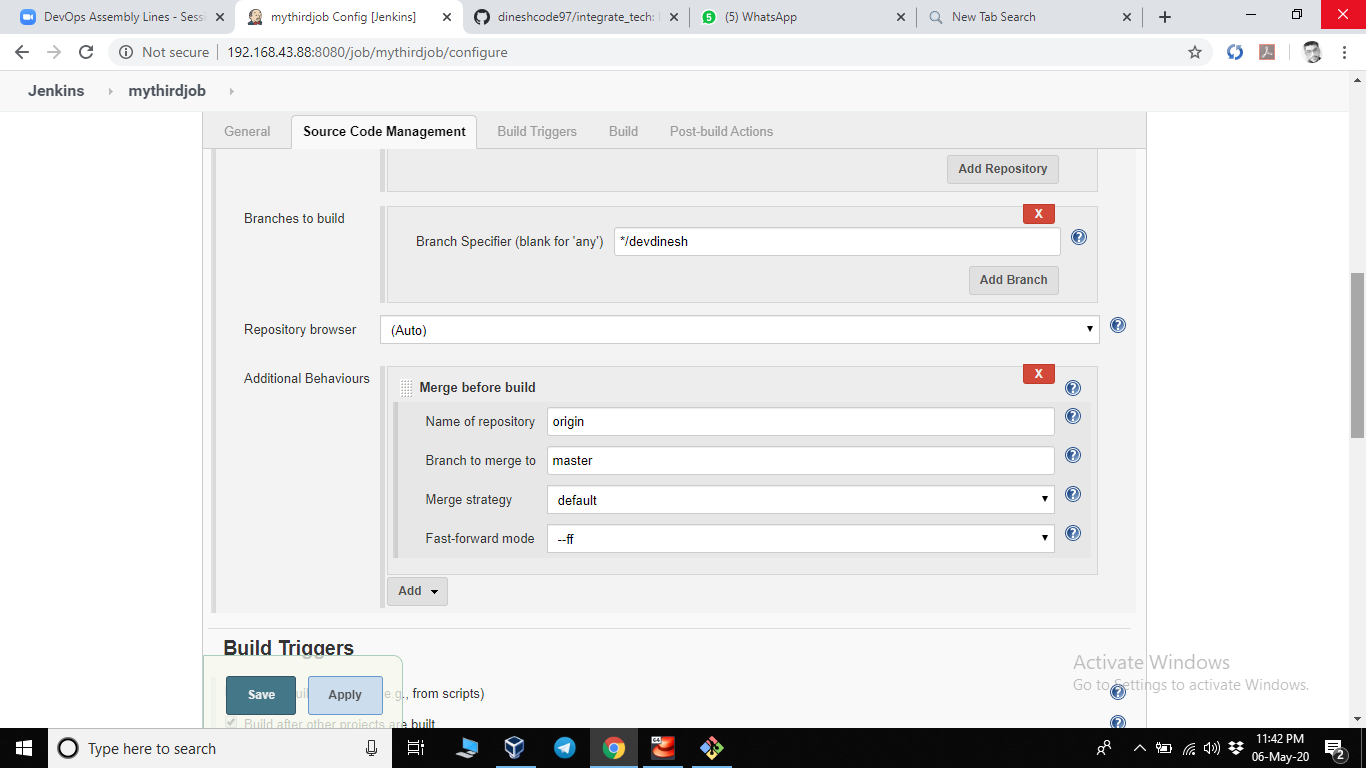
* Enter your github a/c username and password their



* Select additional behavior then a box appears
* In the box enter:

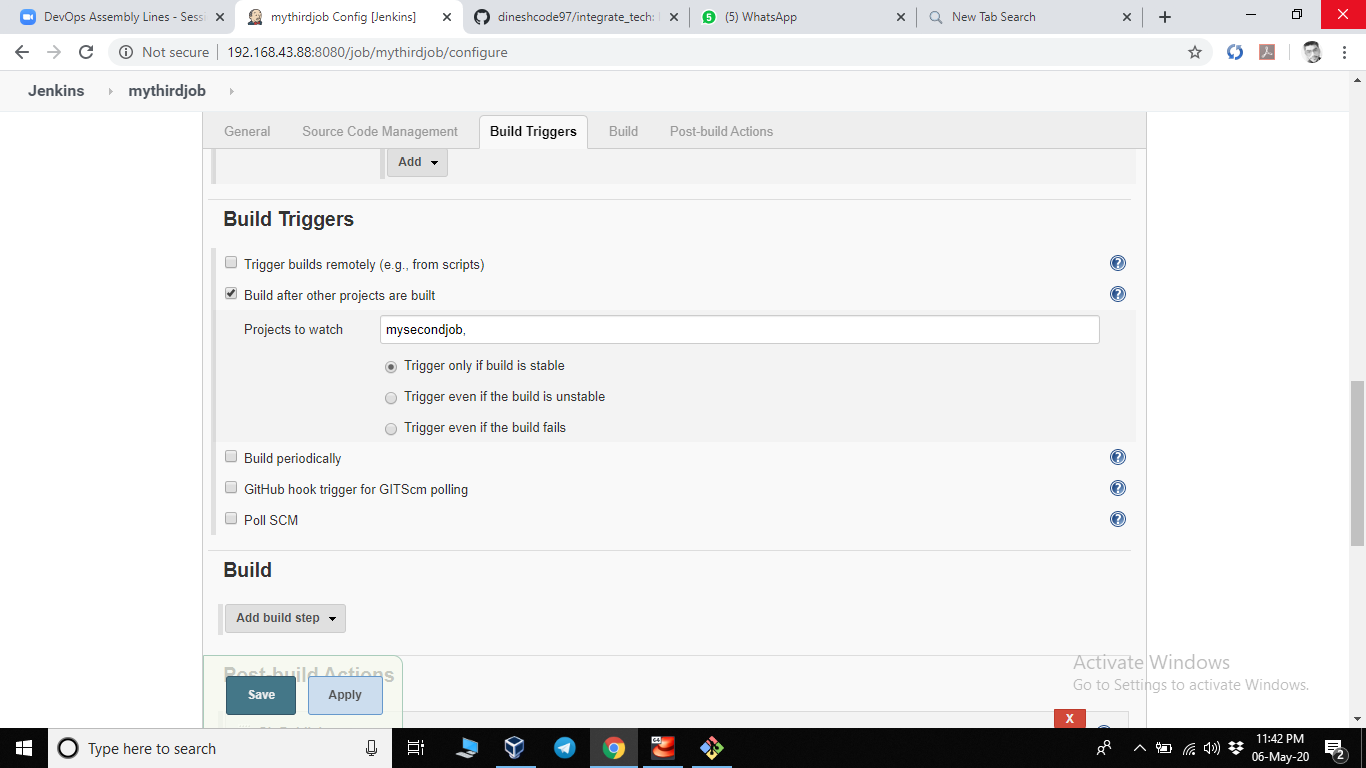
Name of repository =origin

Branch to merge to= master



* Goto build triggers and select build after other project are build
* In project to watch give your second job name (eg. mysecondjob)

And select trigger only if buils is stable



* Goto post build action and select git publisher
* Tick(check) on push only if Build succeeds and merge results

