**PROJECT 1**

**DevOps Assembly Lines**

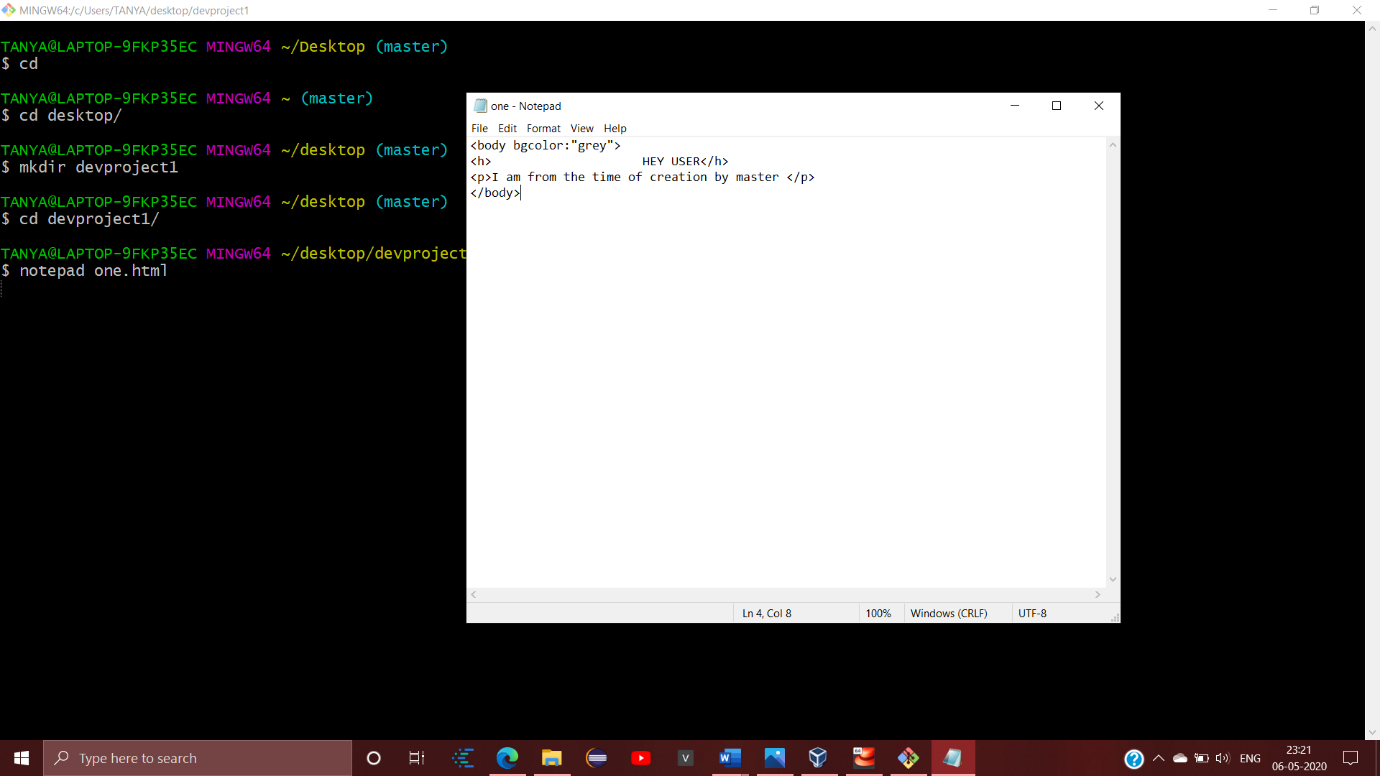
**(2020)**

TOPIC: Setup a Jenkins production and testing pipeline using docker

*Assumptions: Git is installed on Windows. The user must have Github profile. Docker and Jenkins must be installed ,here it is installed on RHEL8 running on VMWare. Webserver and docker server is configured. There is active internet connection.*

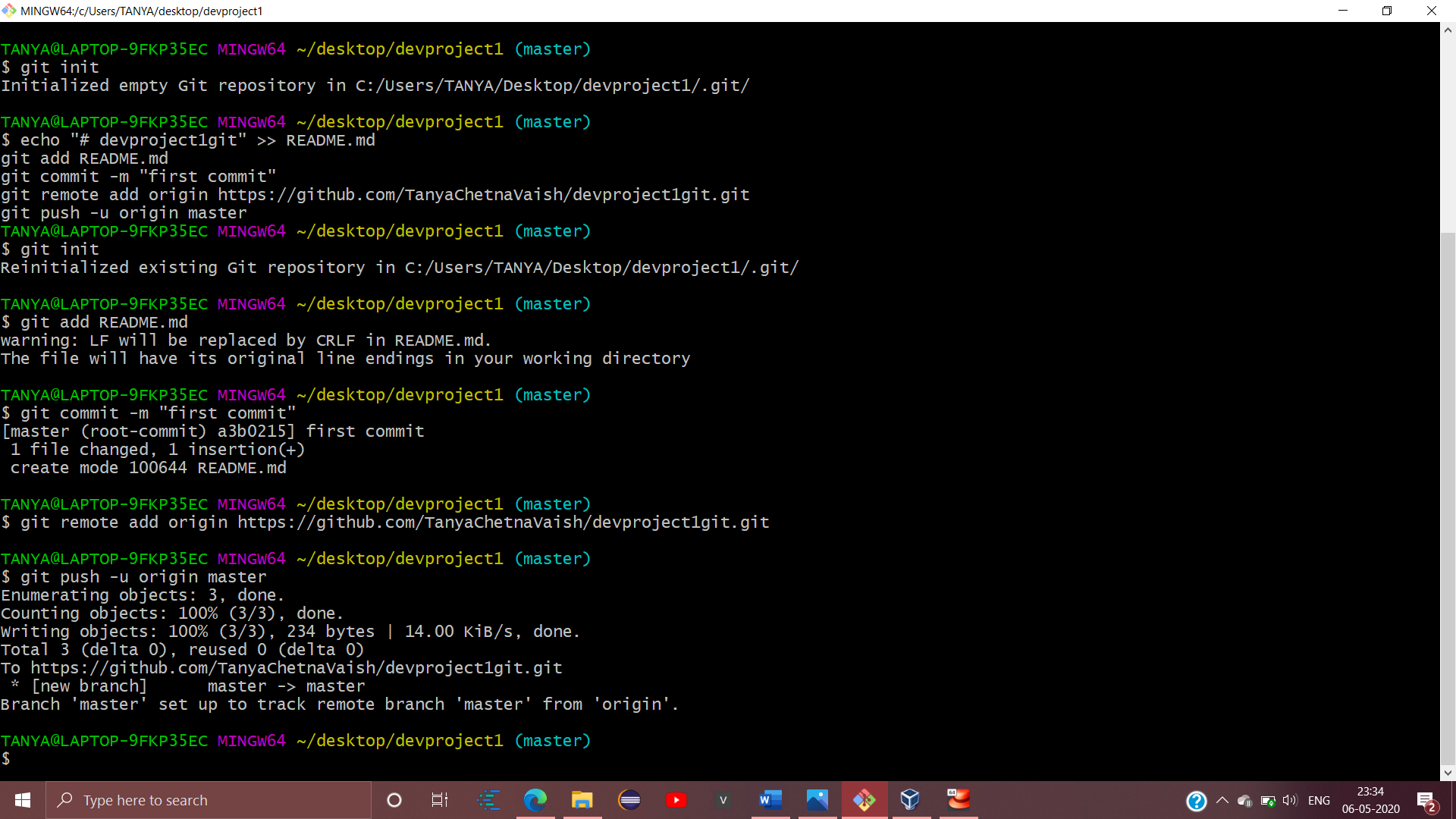
Setting up Git and GitHub repositories and branches:

* Create a folder(here on Desktop) -devproject1 and in it create a one.html file with data.

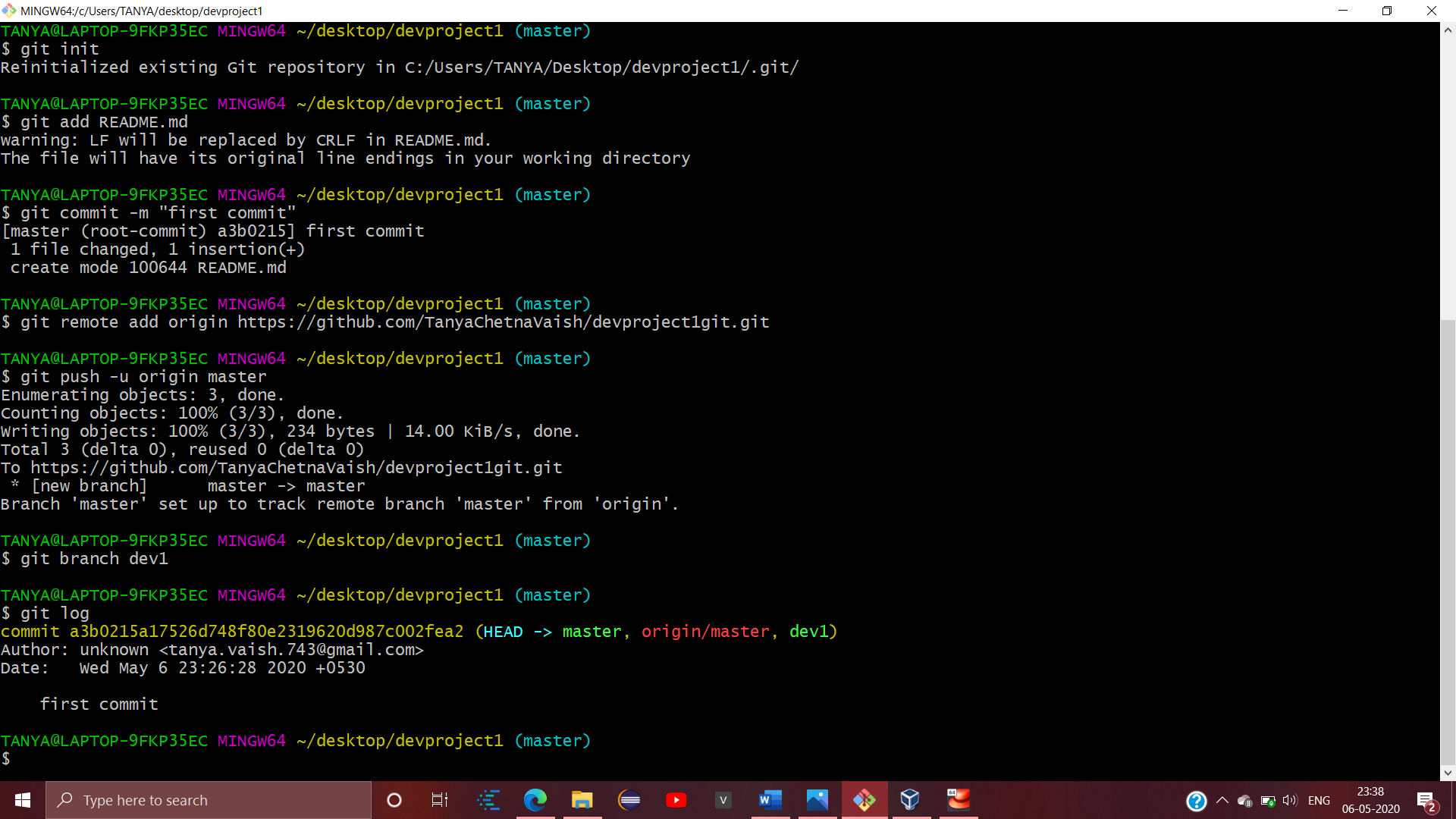


* Create a GitHub repository.And copy the following command onto your git and run it.It will create README file.

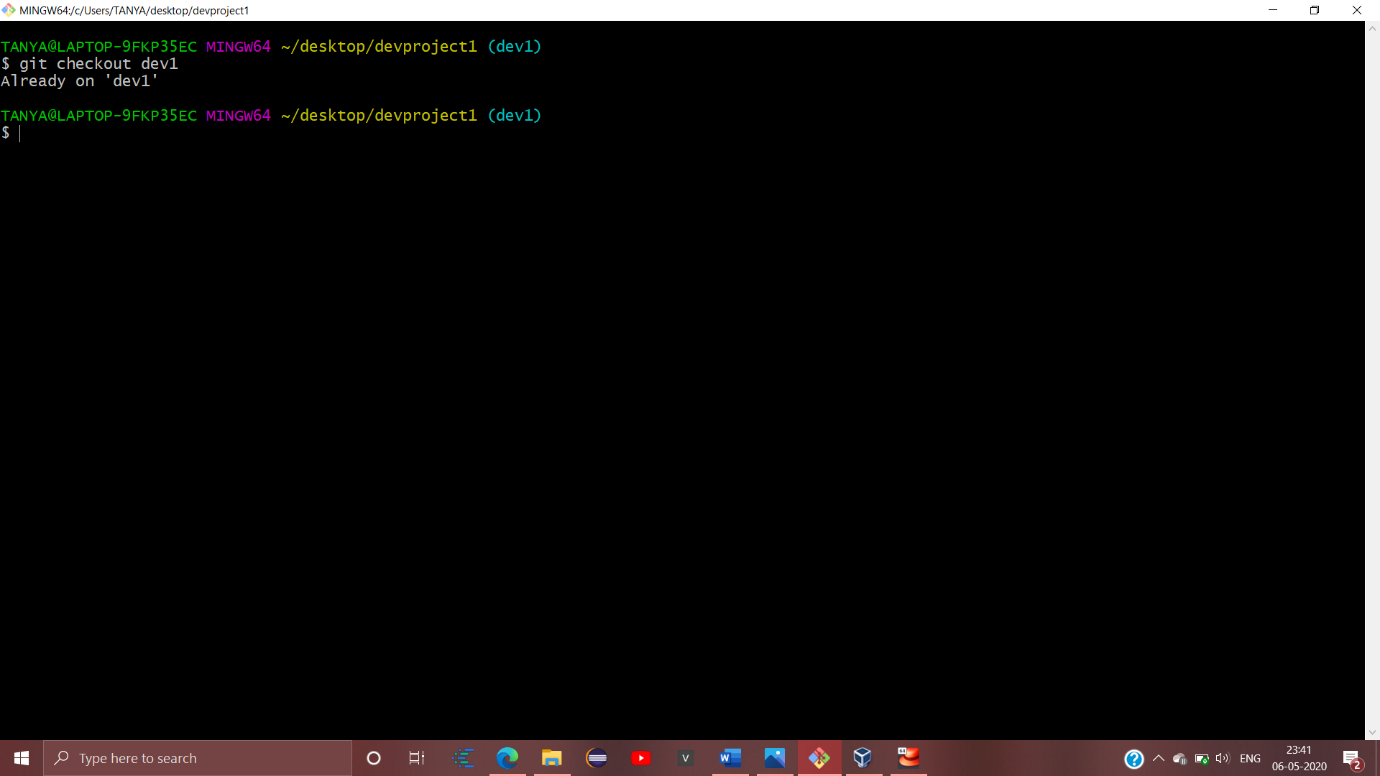




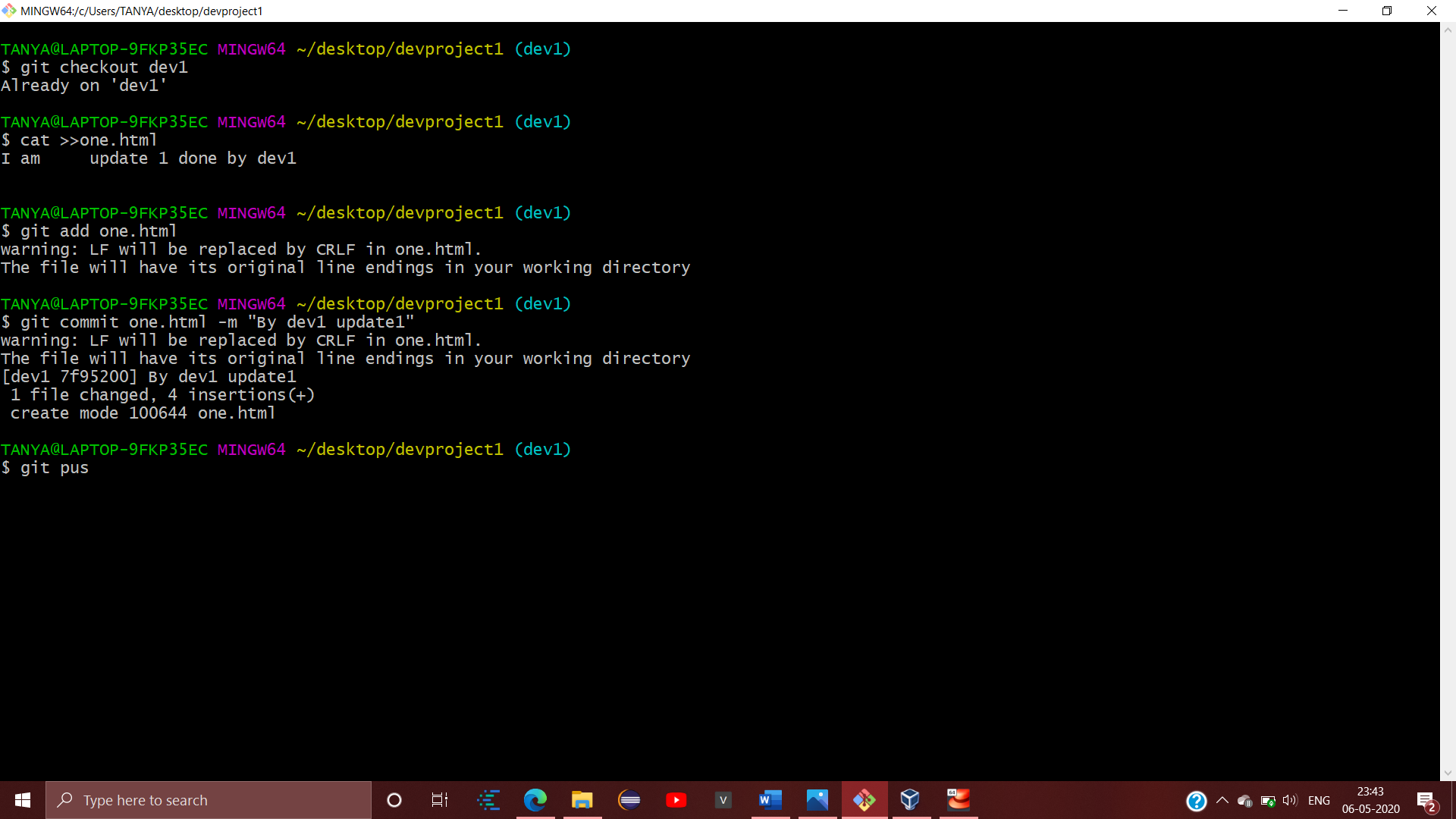
* Make the dev1 branch



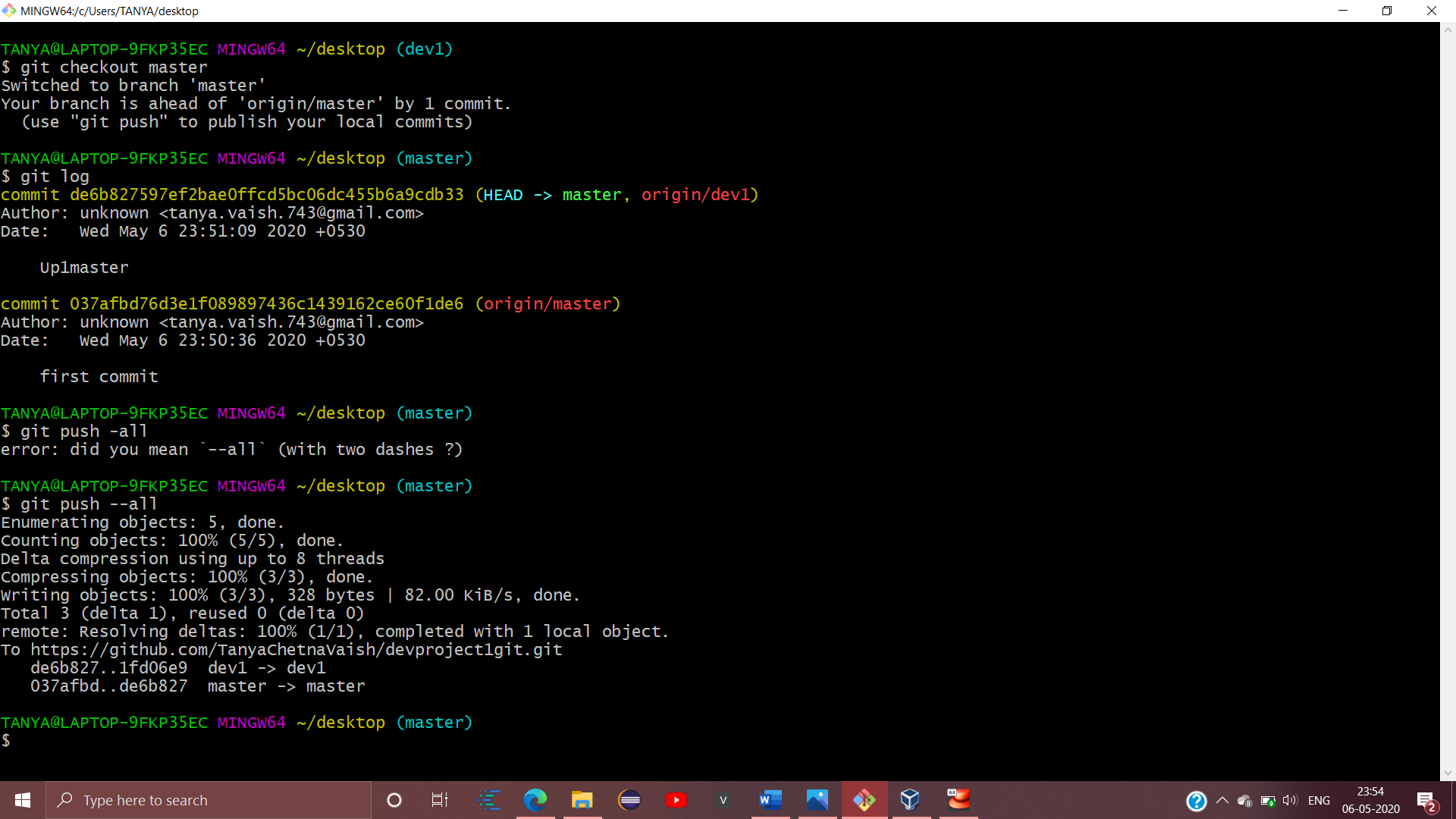
To get into this branch use git branch dev1 command

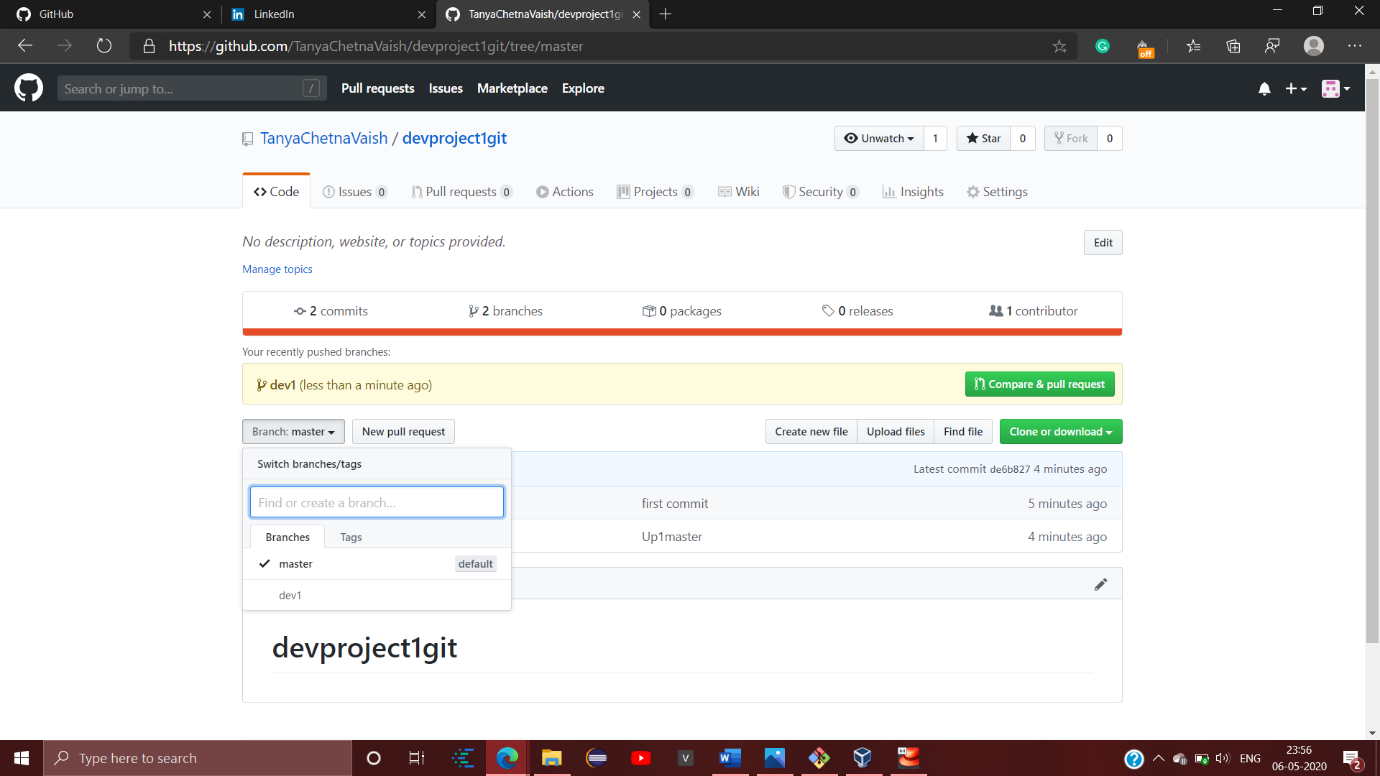


* Now make changes to one.html file in this branch and add and commit it



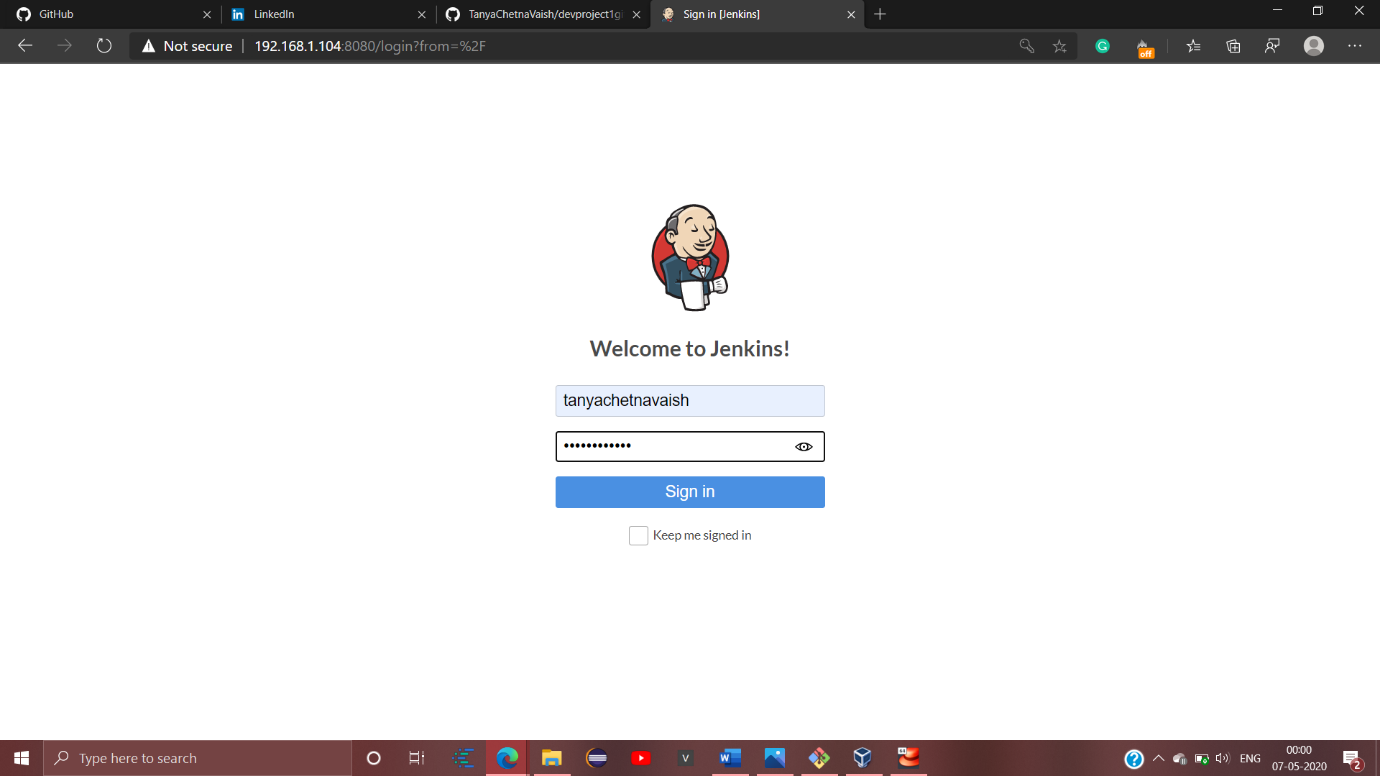
* Now checkout into master branch and push the updates and file on GitHub.Here we will notice neither of the branch merges or hampers the other,thereby maintaining isolation.



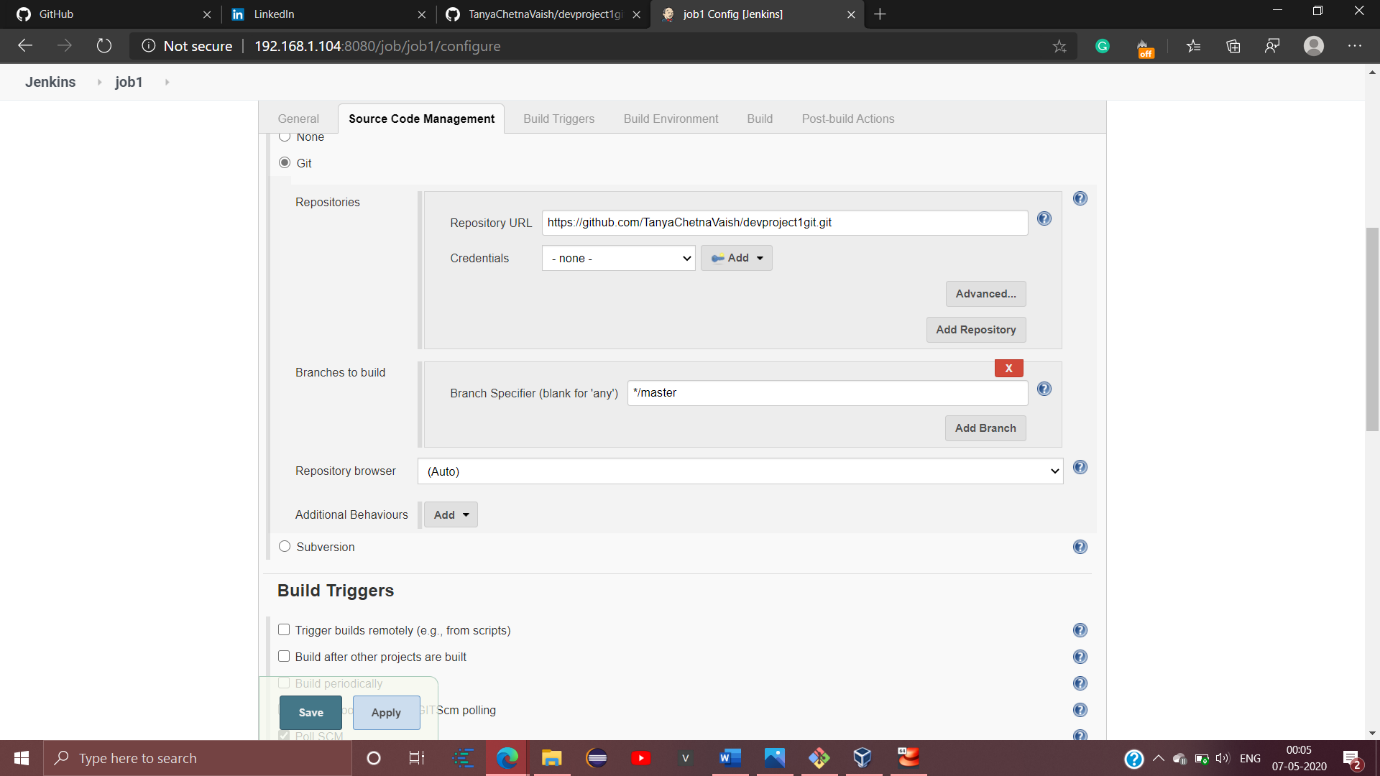


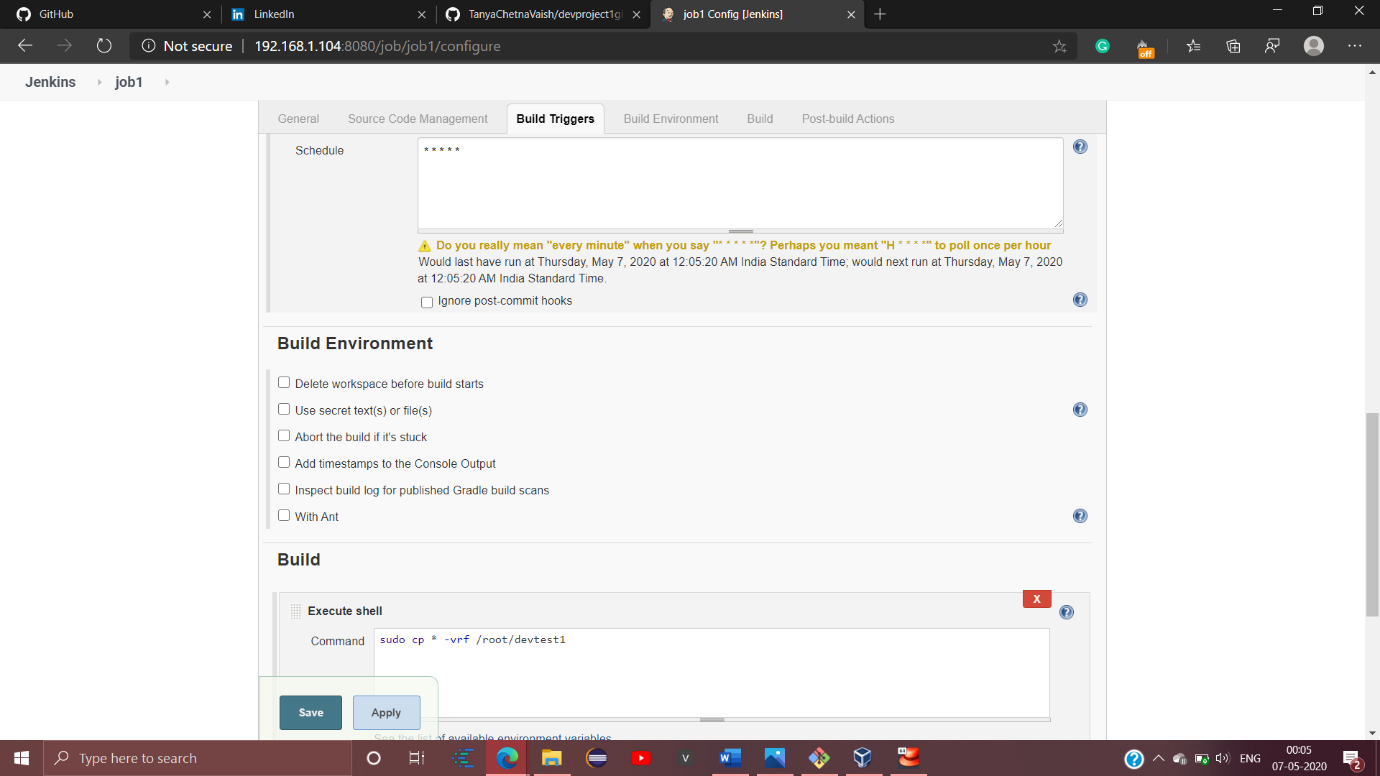
Setting up Jenkins with useful Configurations required for automation:

Login into Jenkins using the IPAddress of RHEL8 and 8080 port.



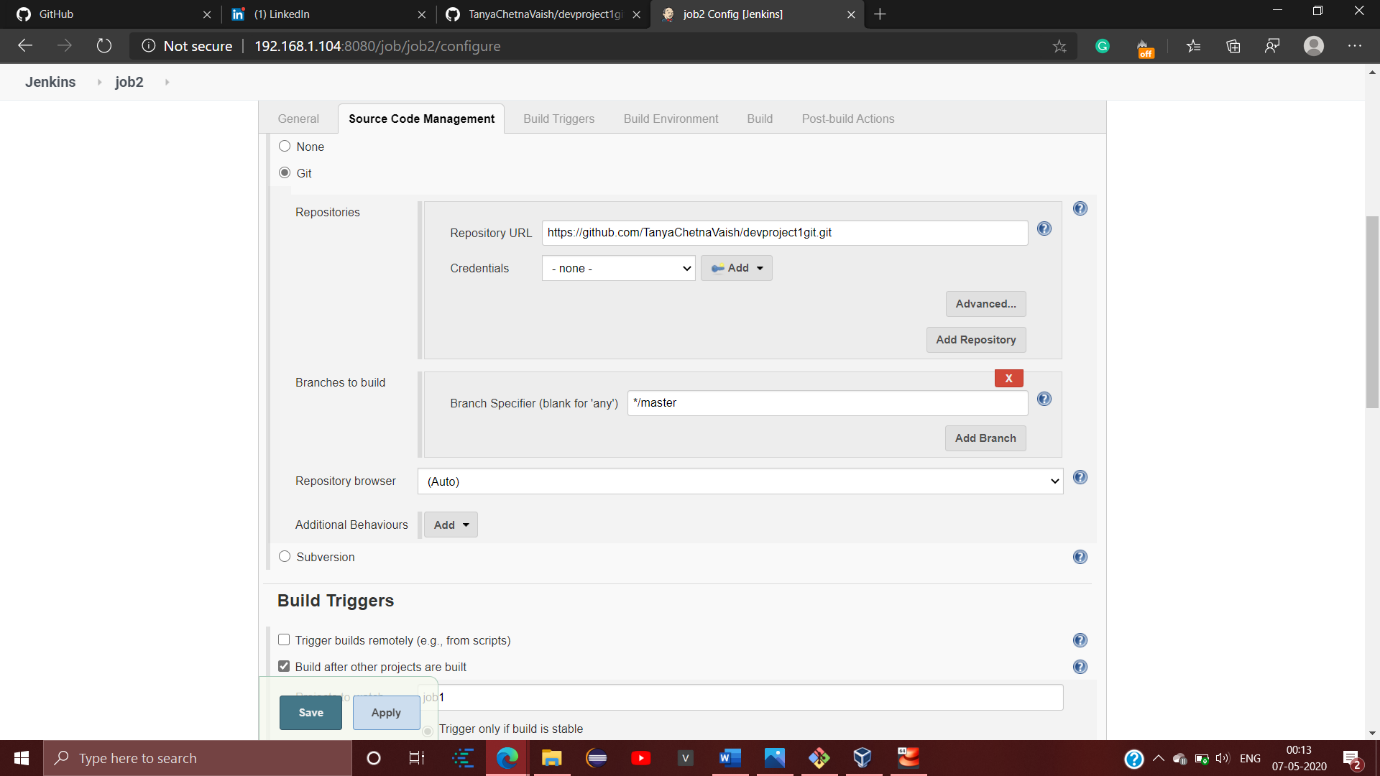
* Now create job1 (Freestyle project) with following Configurations. Use the GitHub URL of the repository here and enable POLL SCM

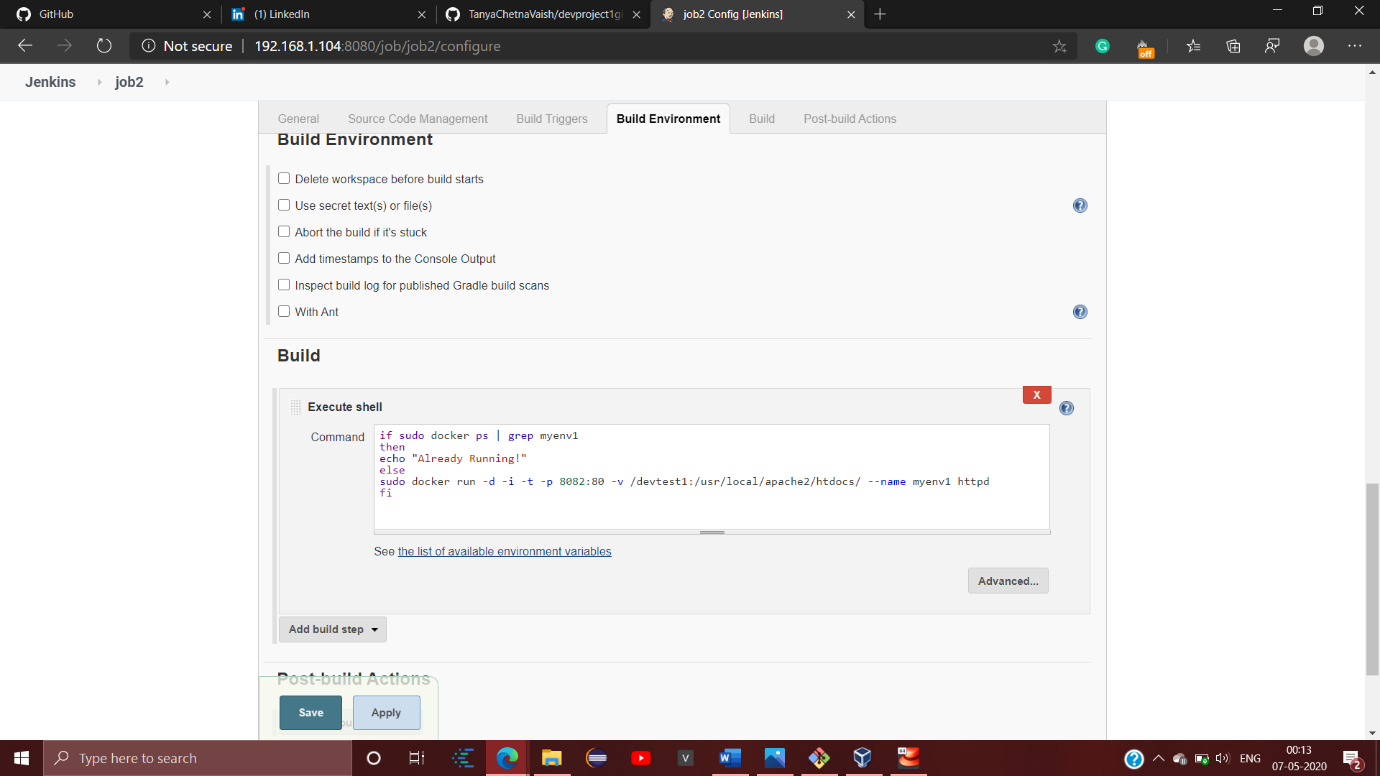




Write the above command in Execute Shell of Build.Remember the path or file specified is the already existing folder in our RHEL8 .

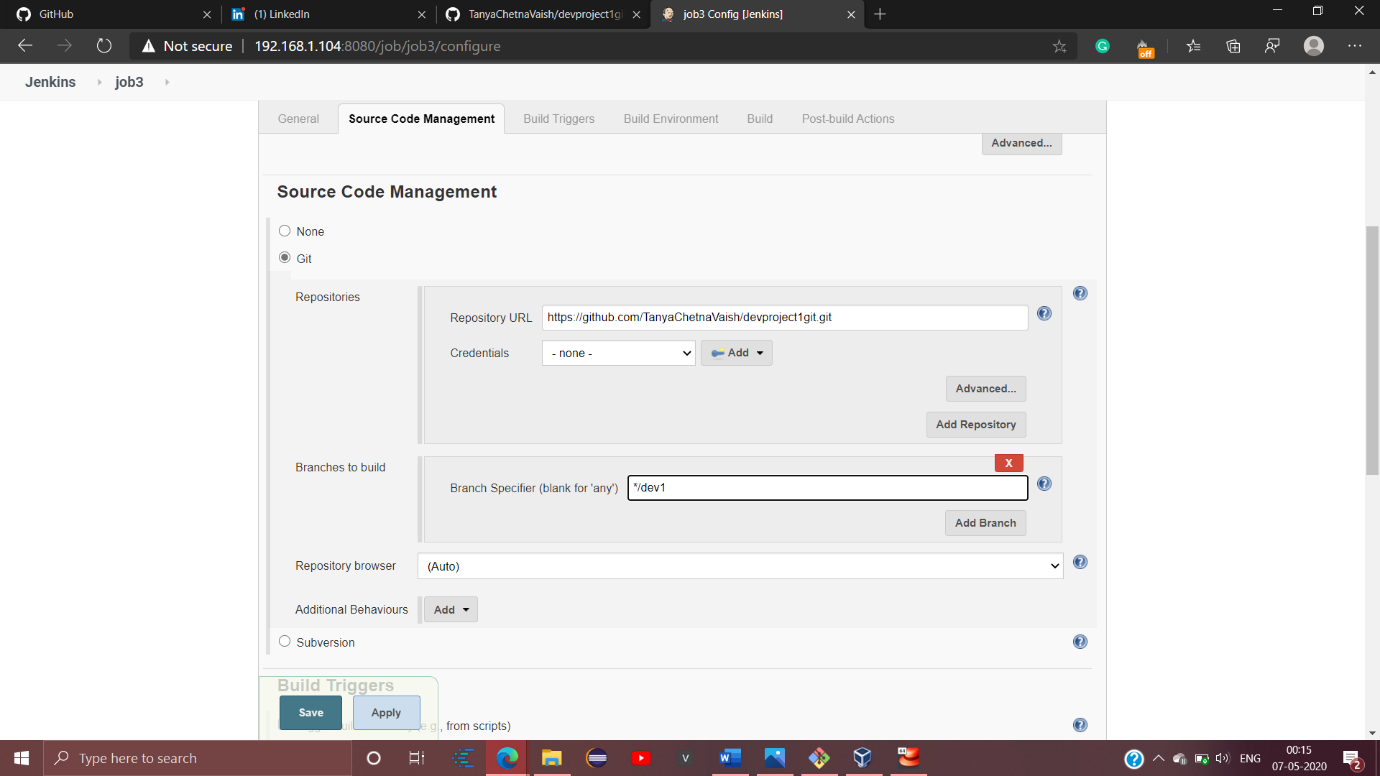
* Create job2 with following configurations:Use the GitHub repository URL here,enable POLL SCM





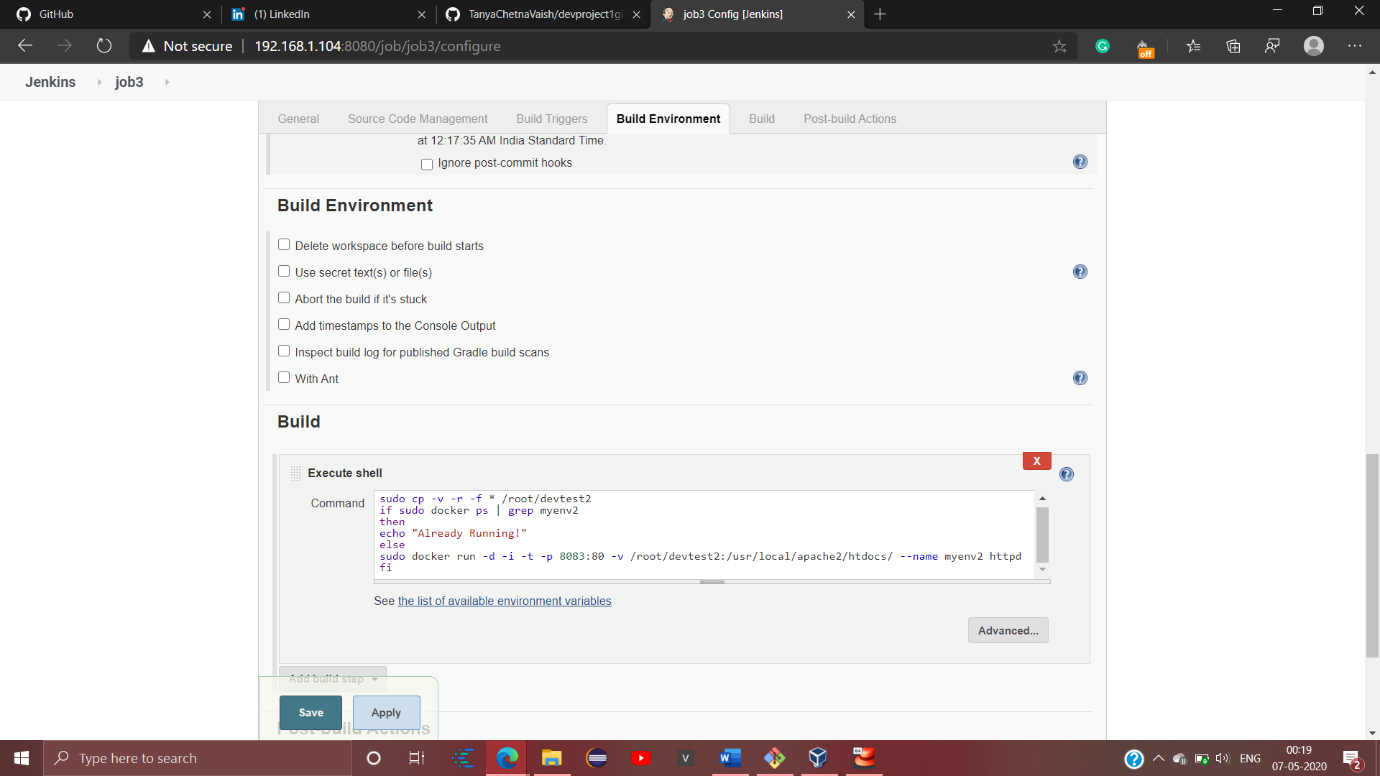
Save the job2 now .

* Create job3(Freestyle project) and copy the GitHub repository URL .

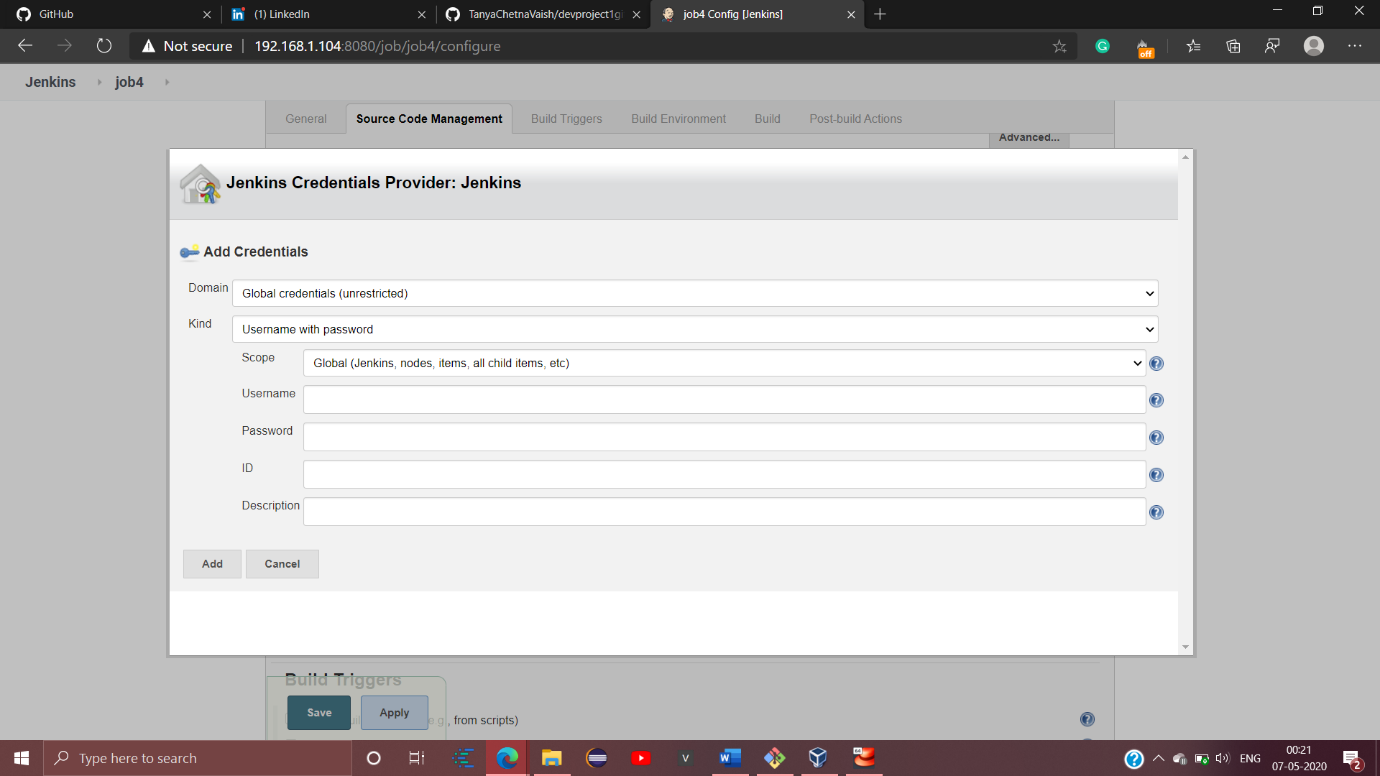


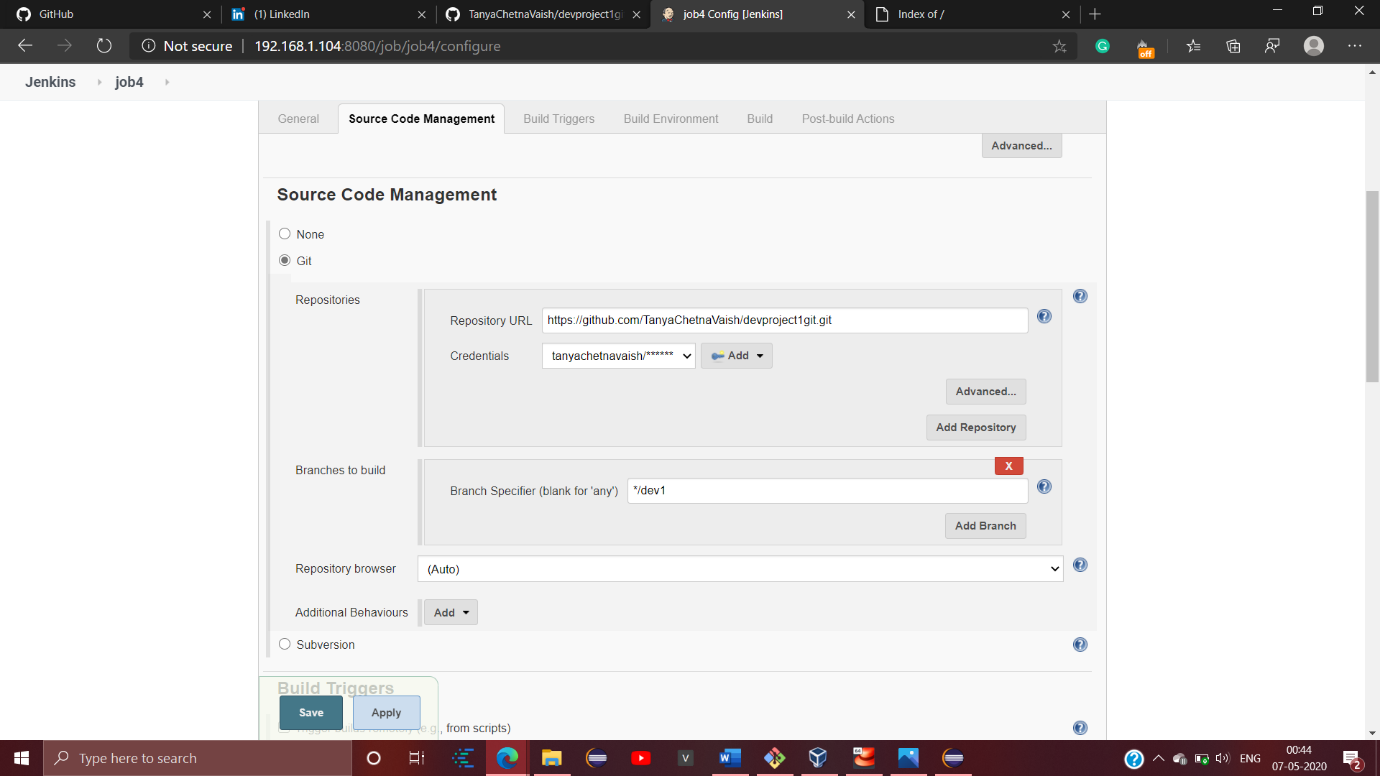
Remember here the branch specification has to be done that is write the alternate created branch here(in this its dev1)

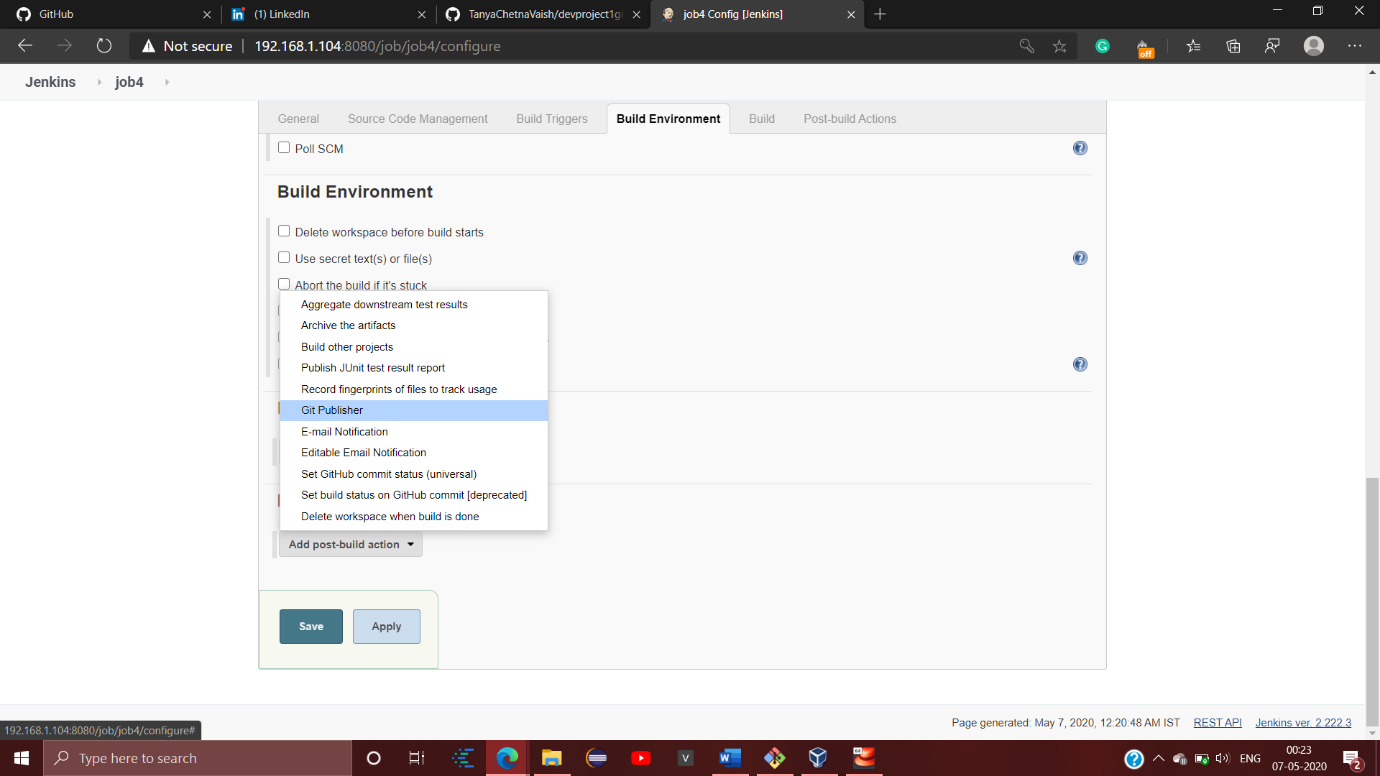
Write the following code on Execute shell.Remember to specify he right folder name from RHEL8 one that exists on the right specified path

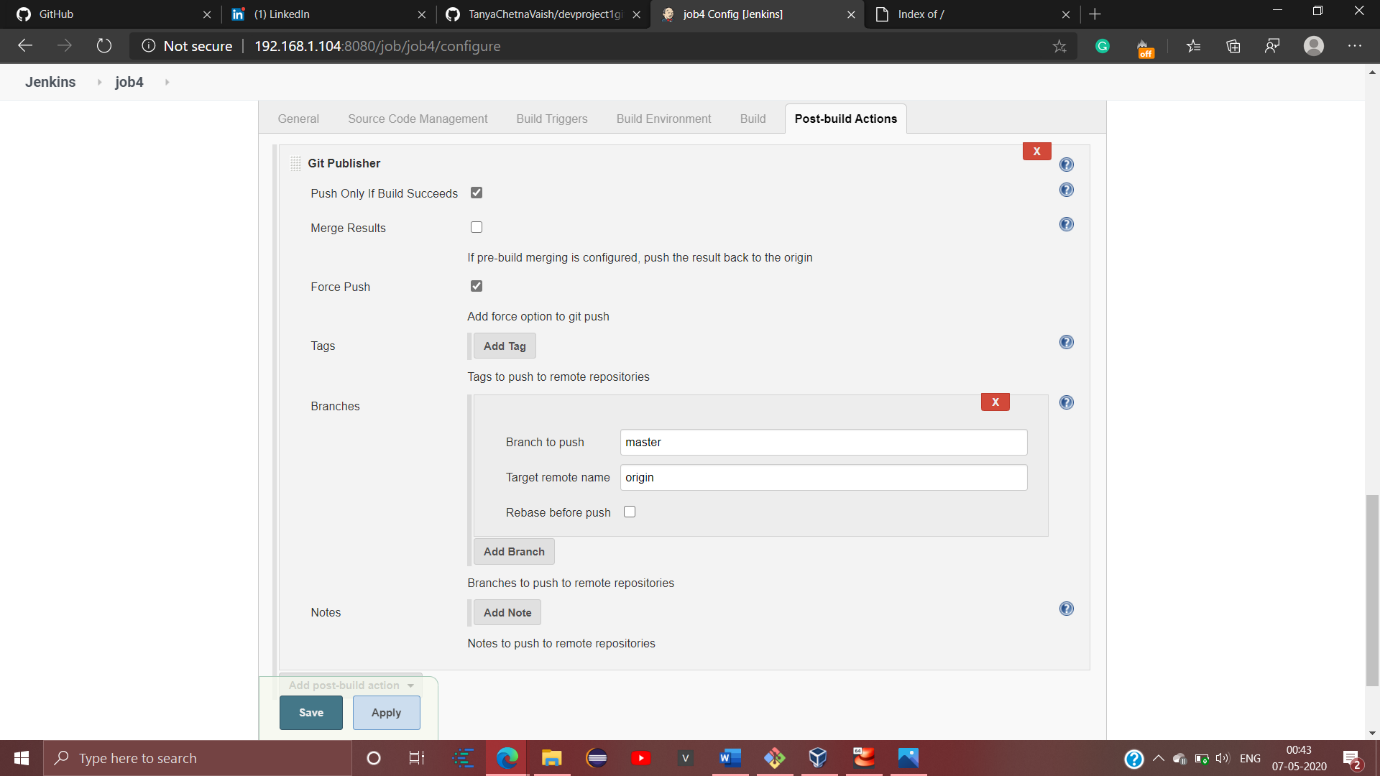


* Now create the last job4 with following configurations:



Now in the post-build actions select Git publisher option and proceed with following configurations:

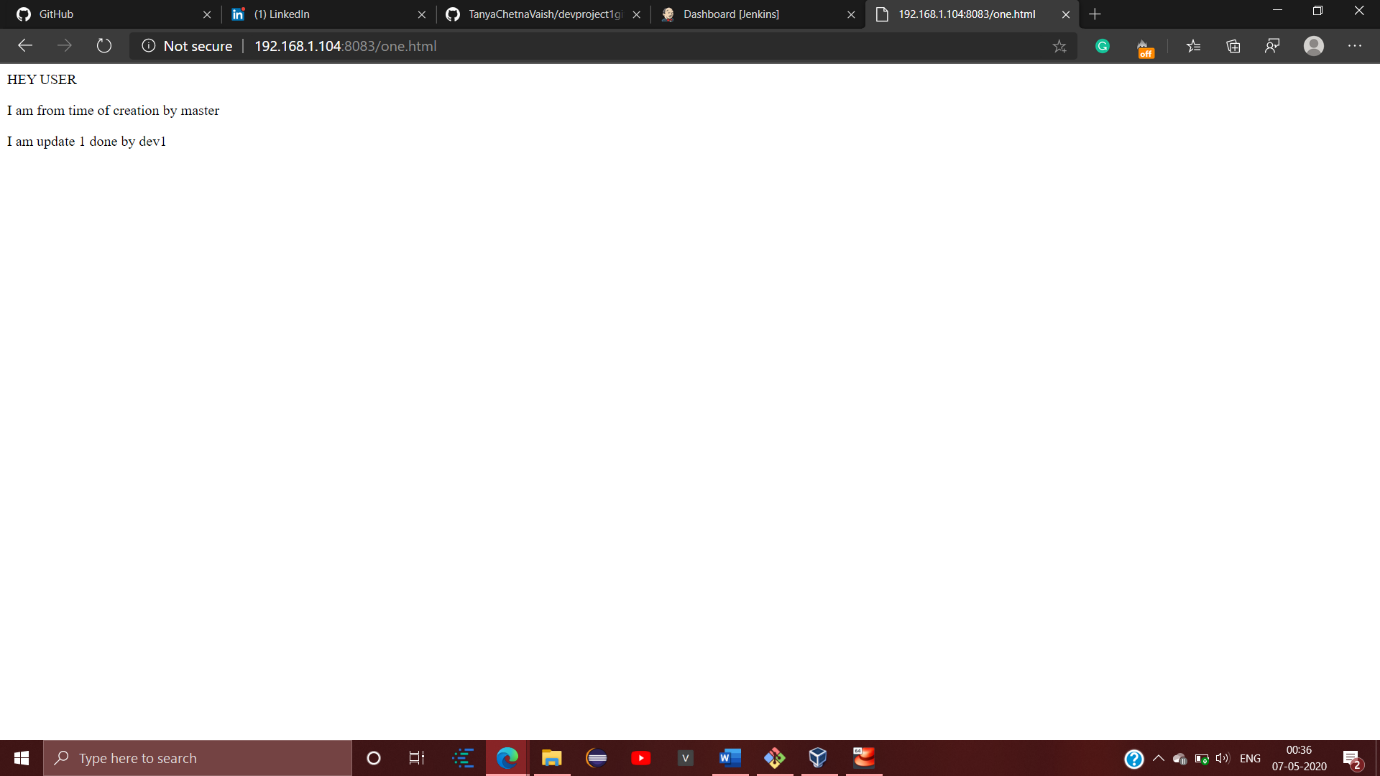




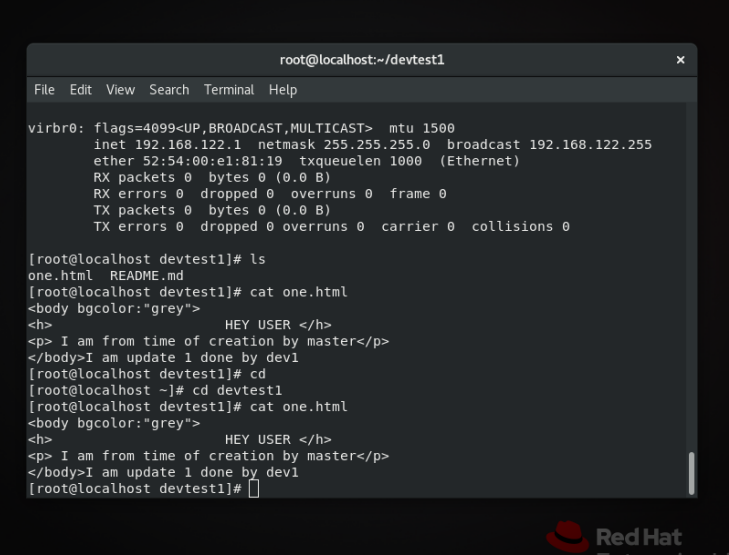
Now save job4 too with the above mentioned configurations. That’s it!

Now you’re done with your automation.

You can access your one.html file and see the data merged.But job4 has to be build by us,thus insisting security as well from the users side.

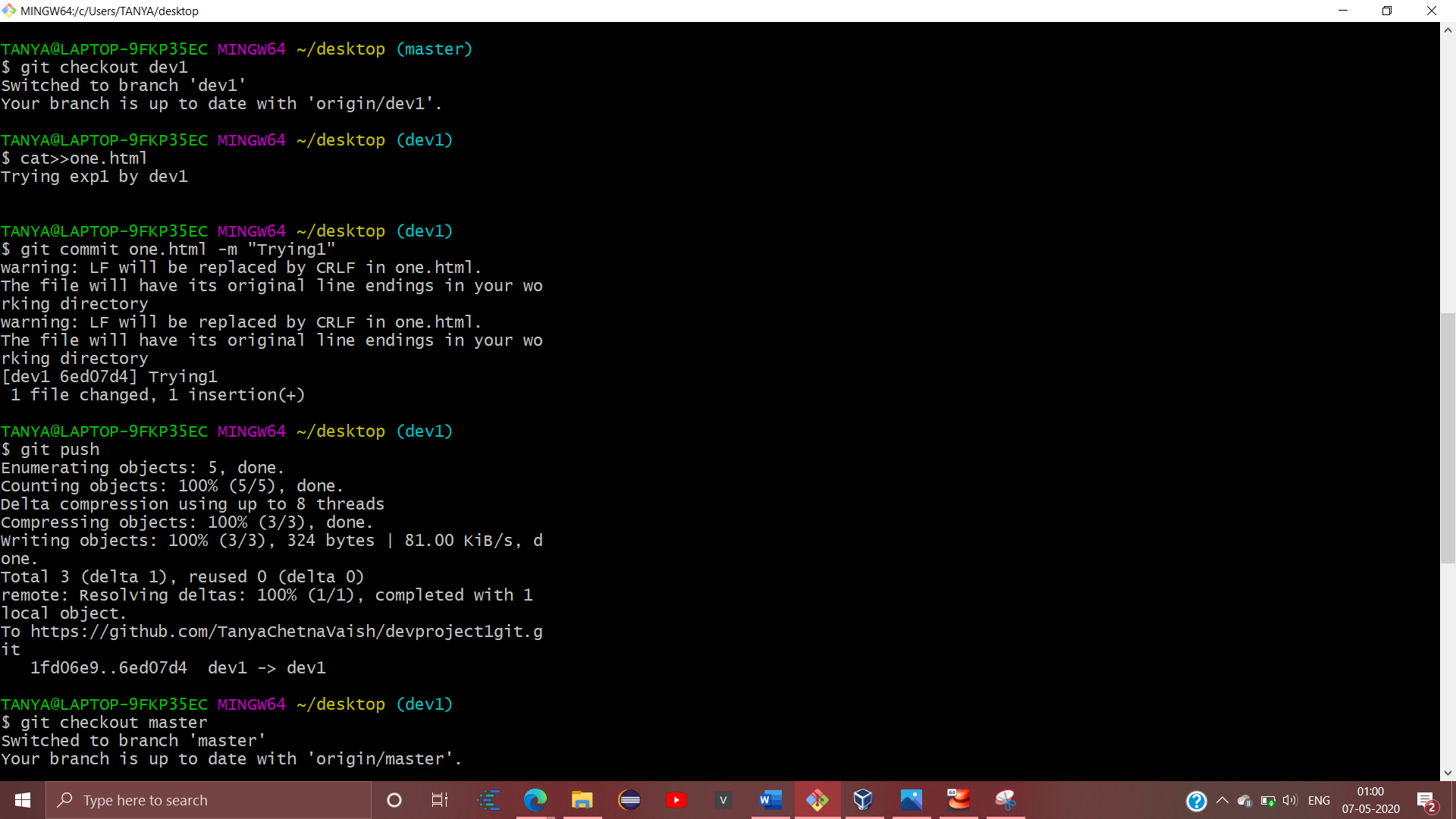


Further we can from RHEL8 ,both the files drom devtest1 and devtest2 contain data from both master and dev1 branch in one

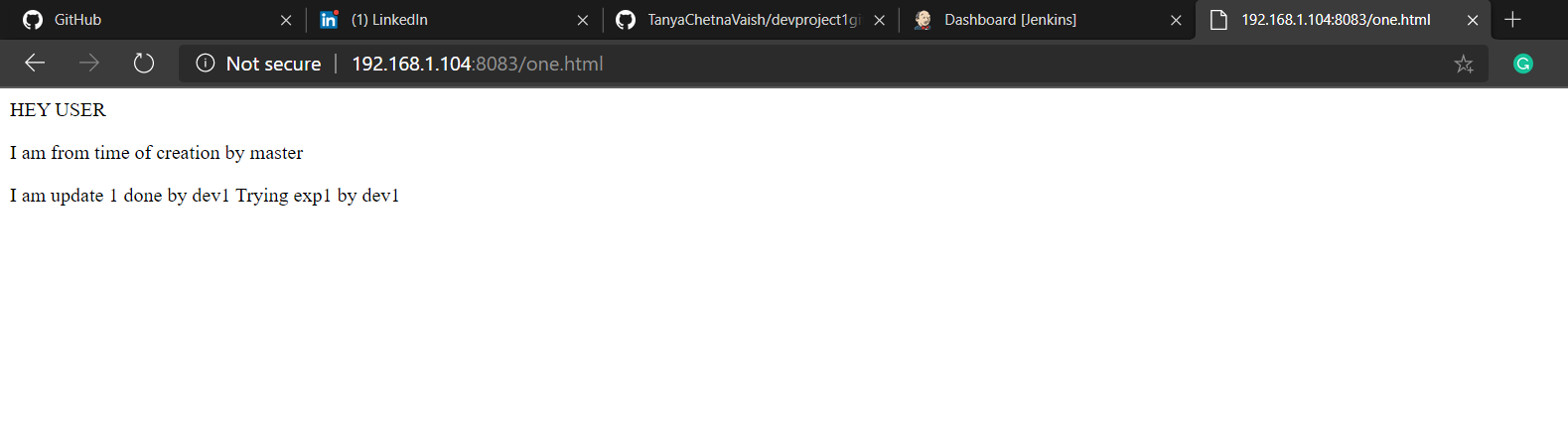




If we make changes in the file from dev branch then



It will get updated after running job4



I hope it helps.Thankyou!