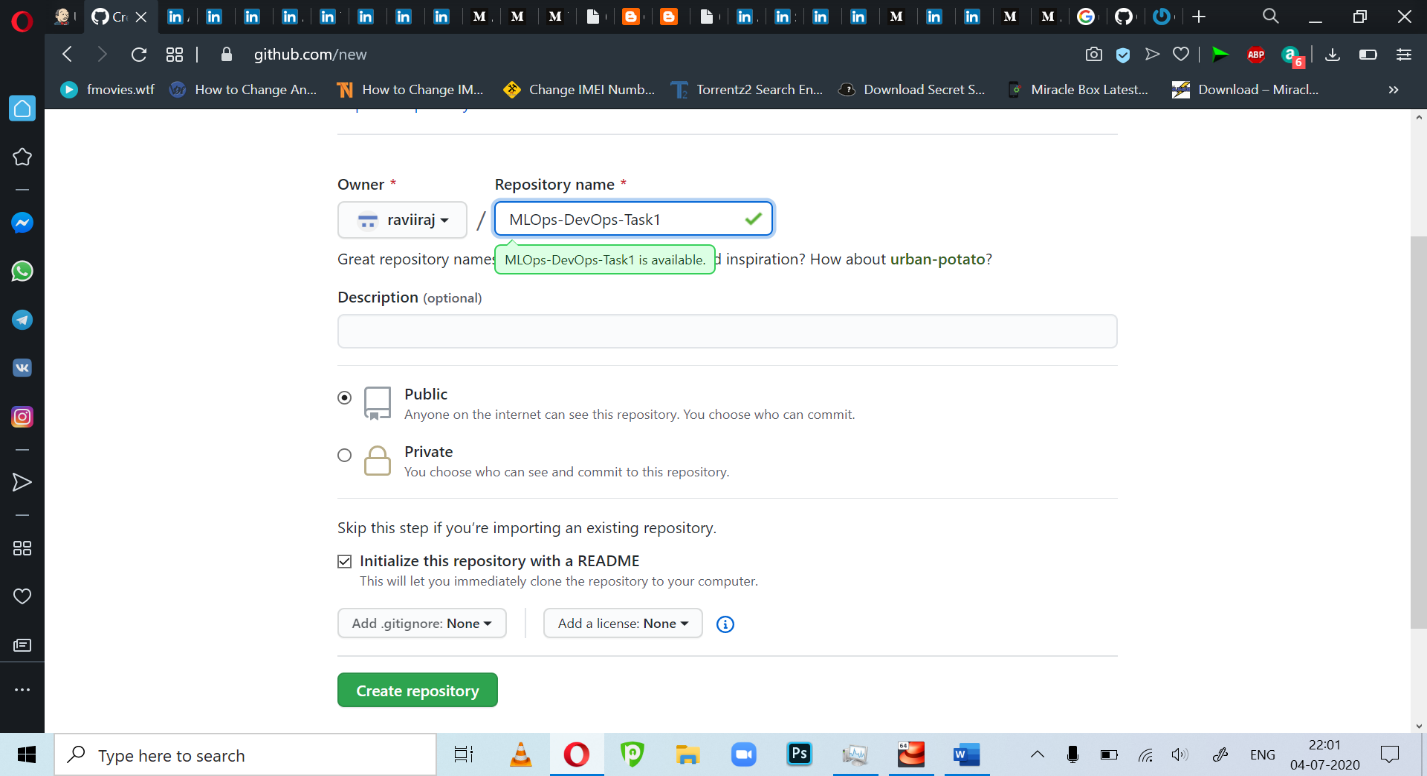
[ MLOps + DevOps ]

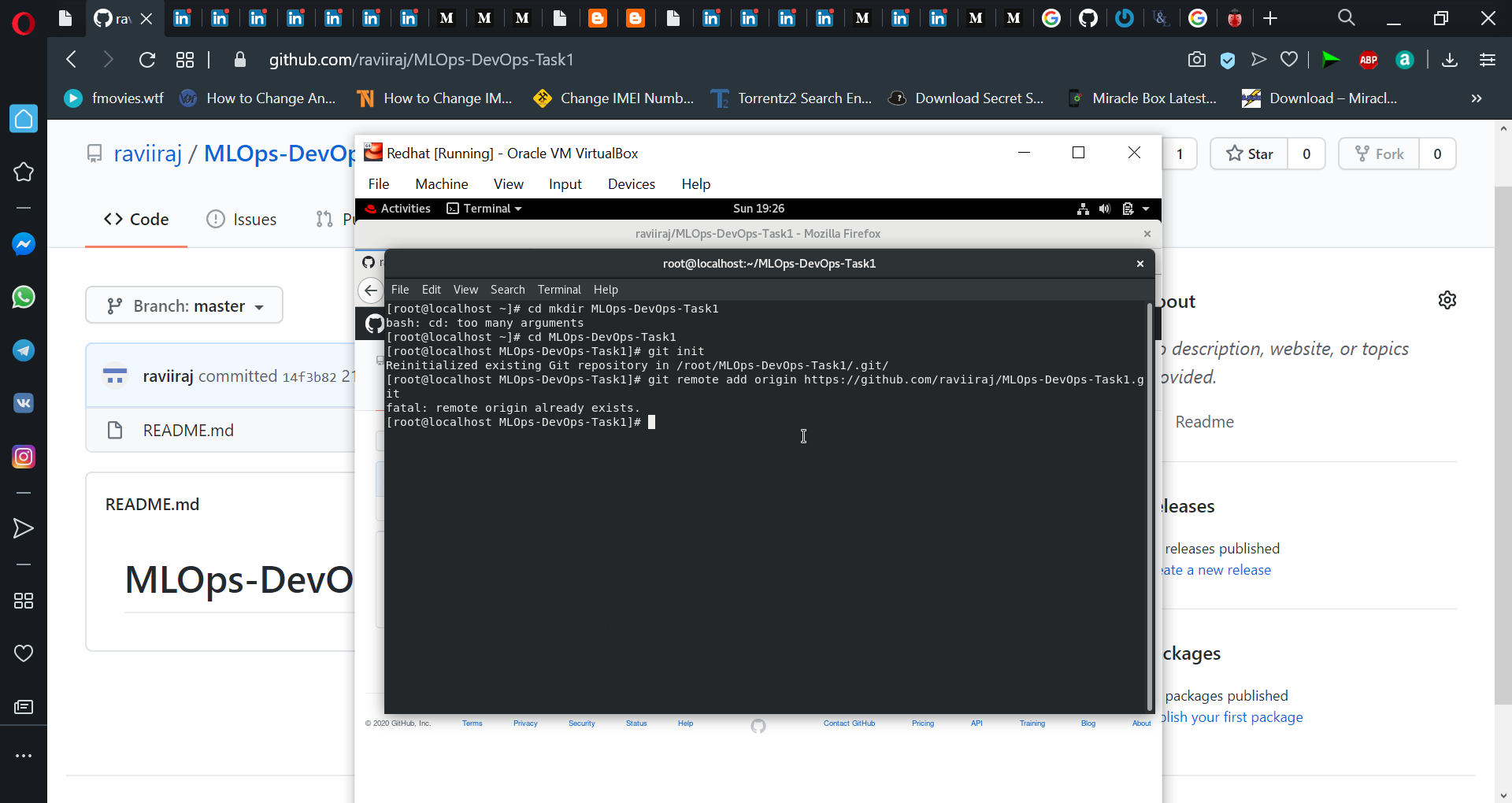
Task -1

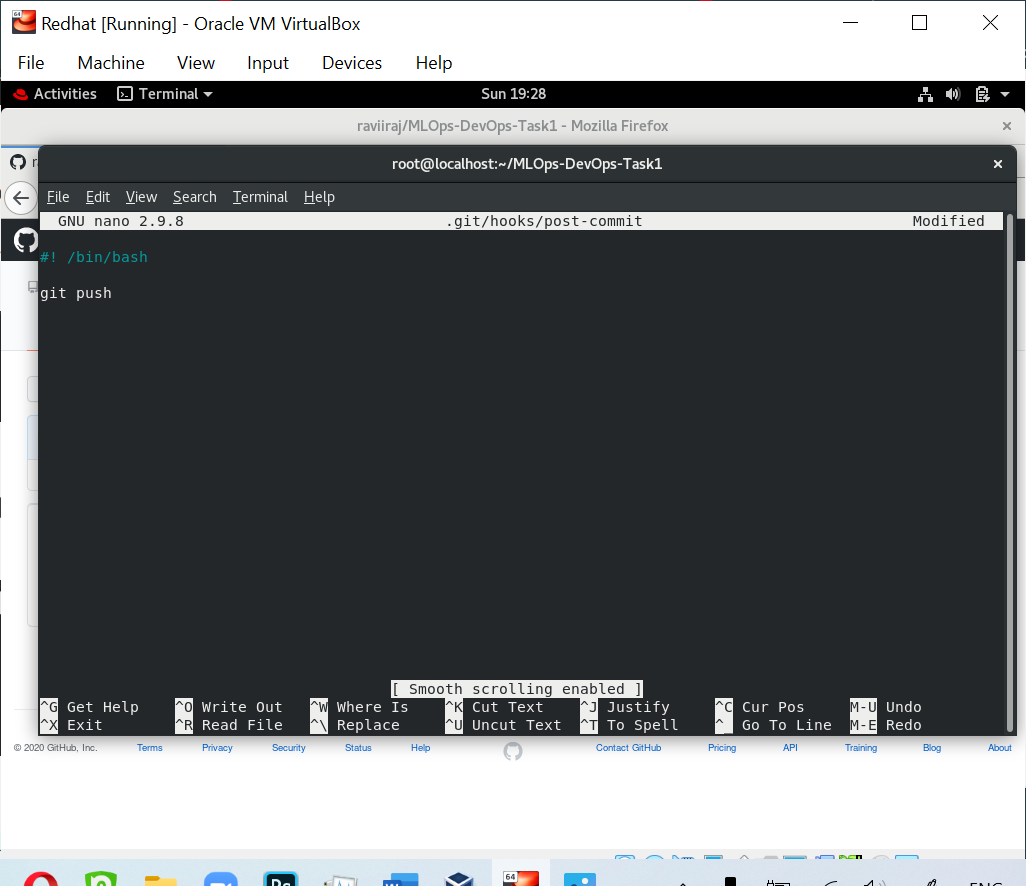
#Job 1

If developer pushes to dev branch then Jenkins will fetch from dev and deploy on the dev-docker environment.

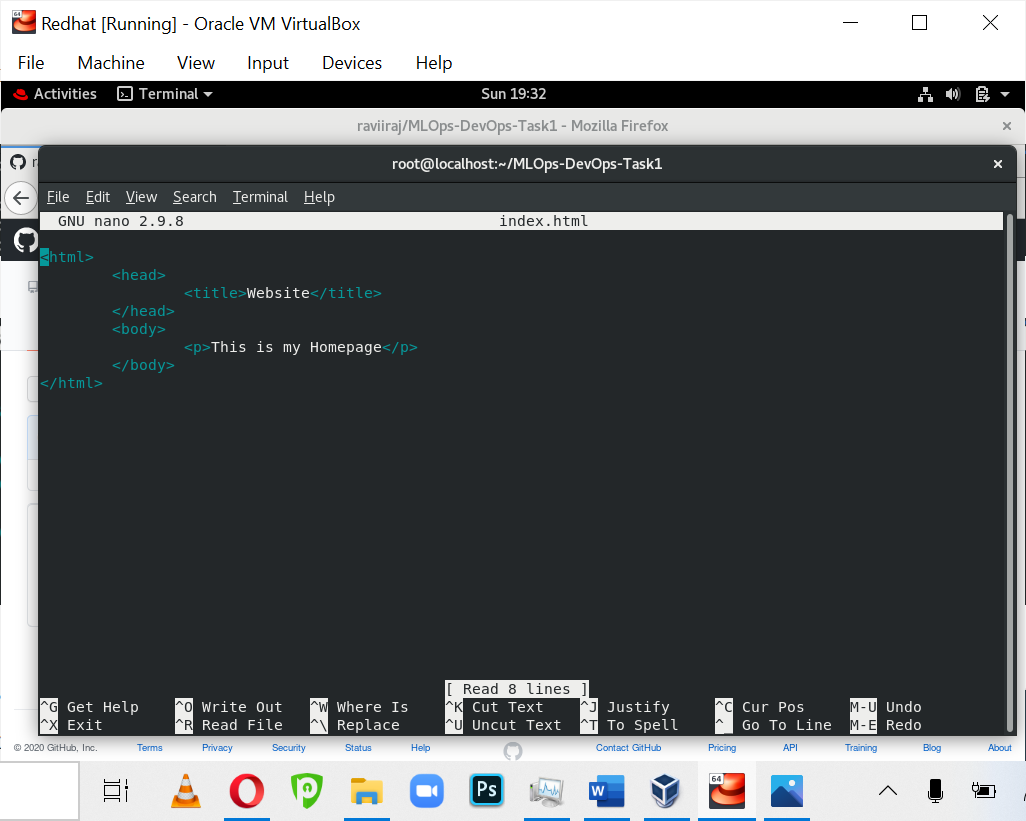
First create a repository in GitHub.



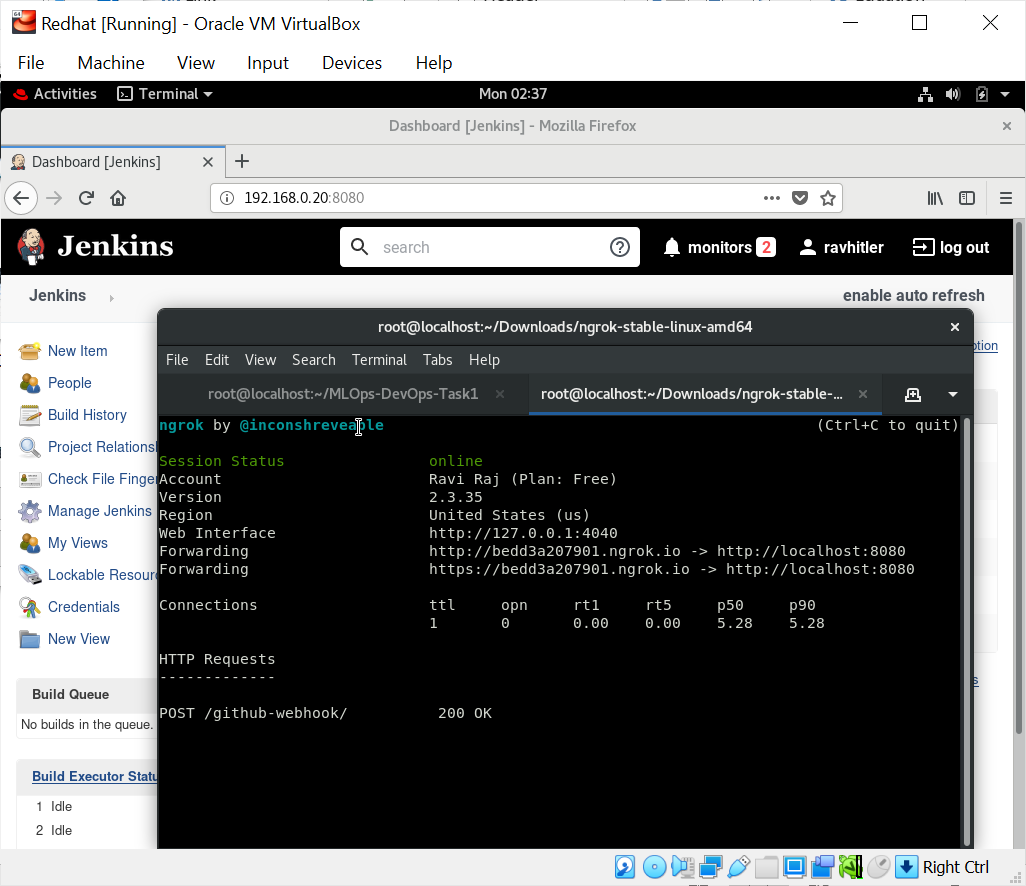
Initialize Git Locally. 

Add a post-commit file inside hooks and make it executable for automatically pushing changes to GitHub . Use the command and add lines to the post-commit file. 

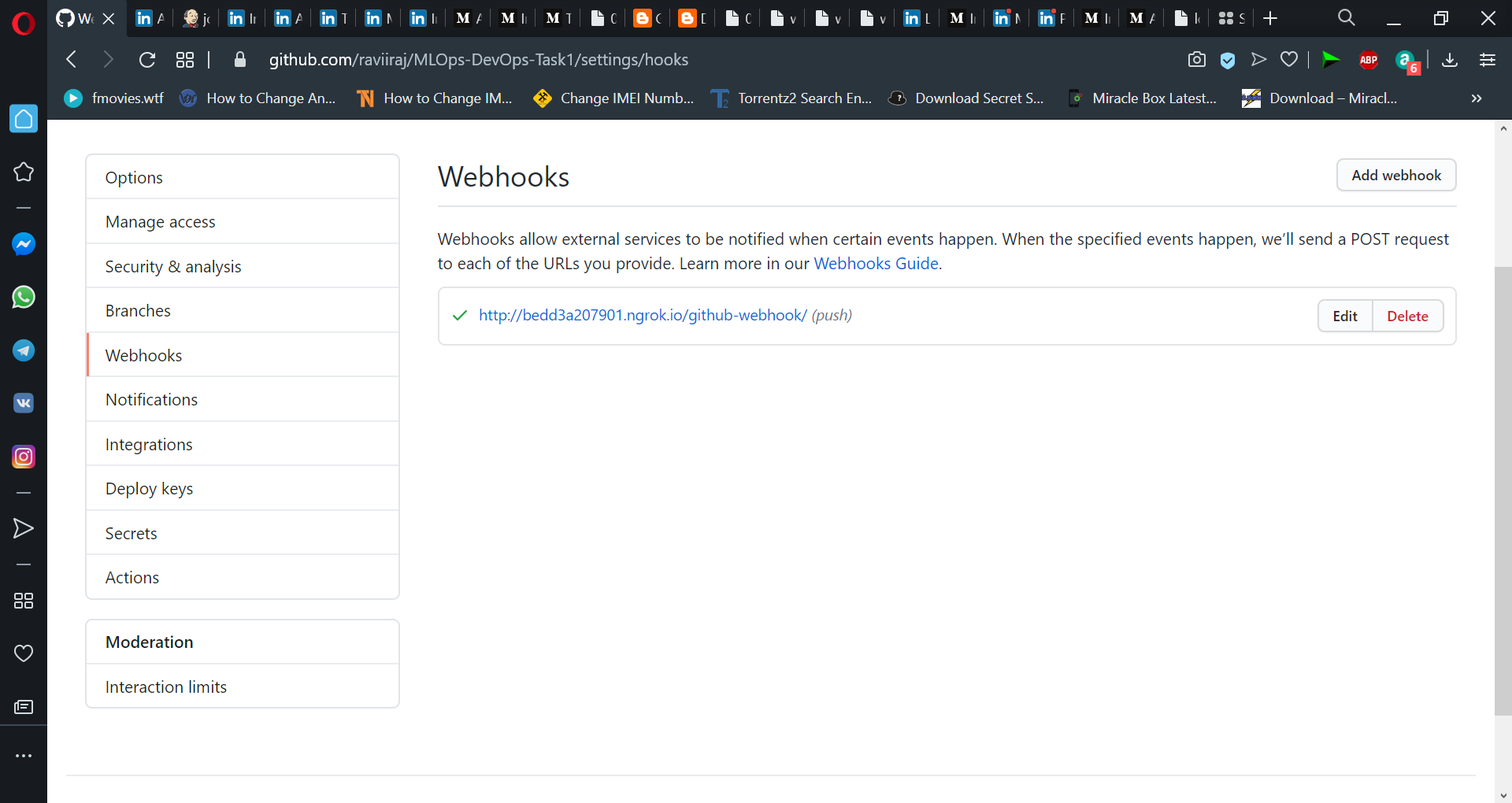
Create separate dev branch and write your index.html



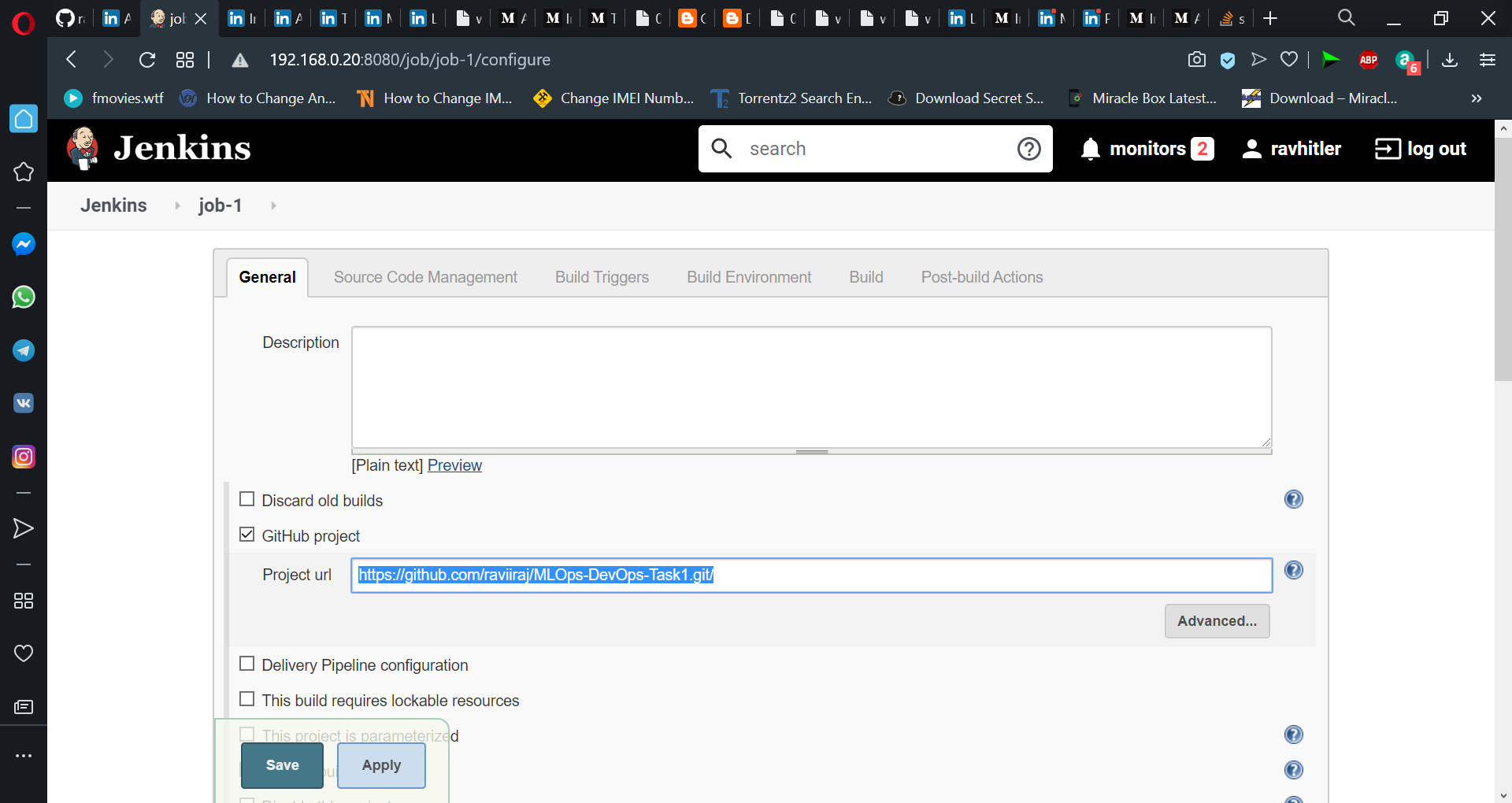
On ngrok output be like

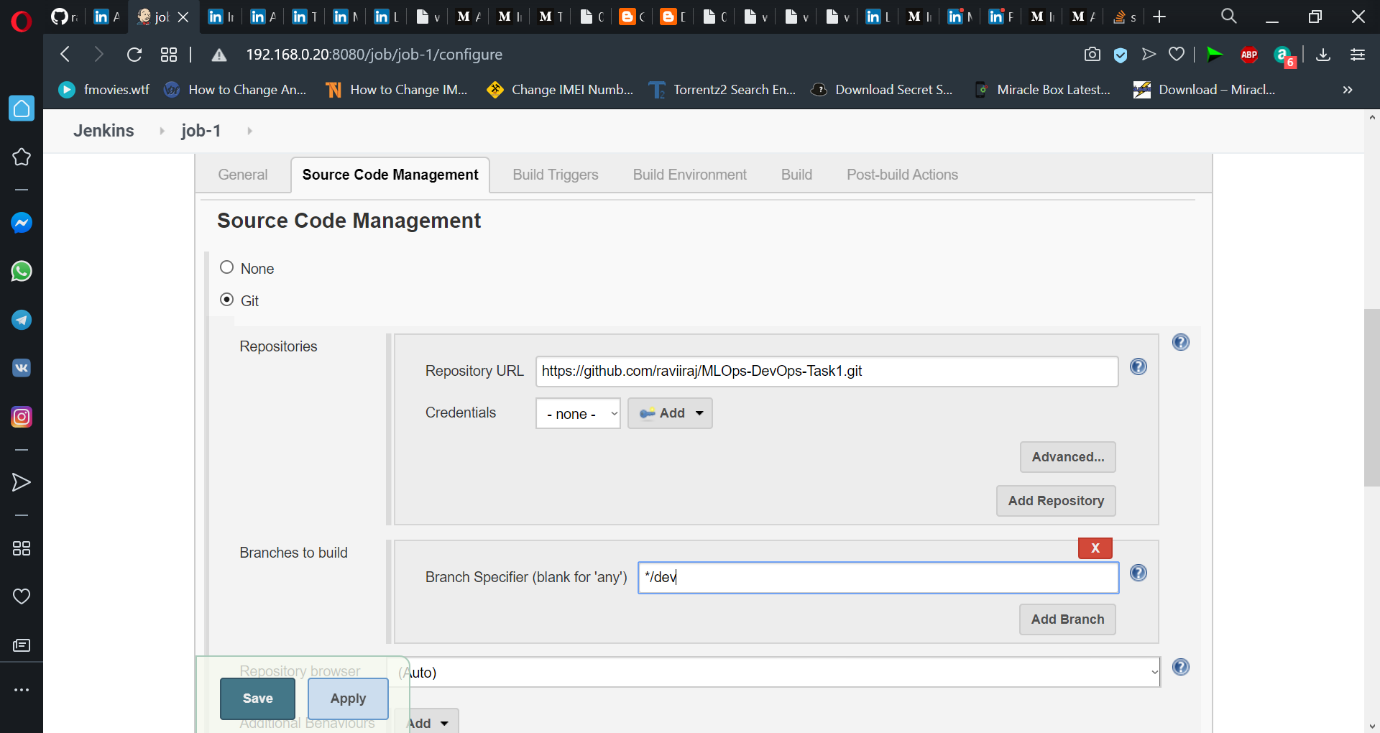


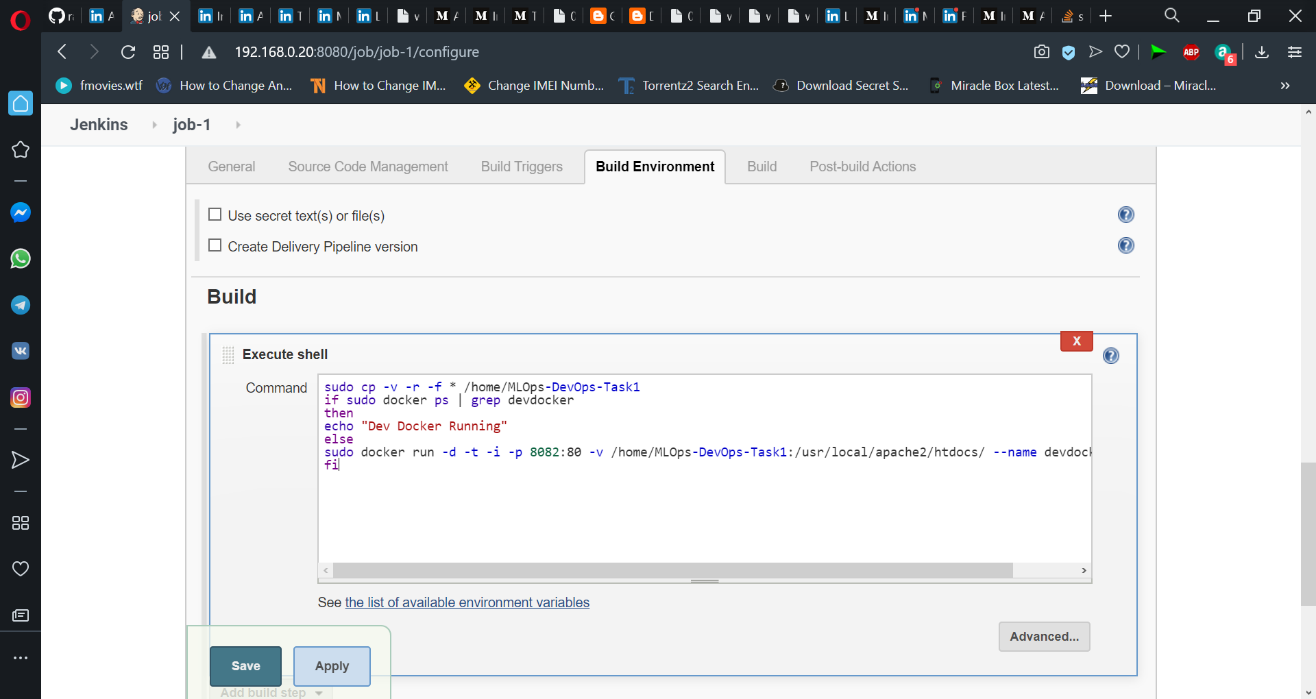
Now, set up a webhook in our GitHub repo. to auto trigger Job-1. Add webhook .



 Create Job-1 in Jenkins.





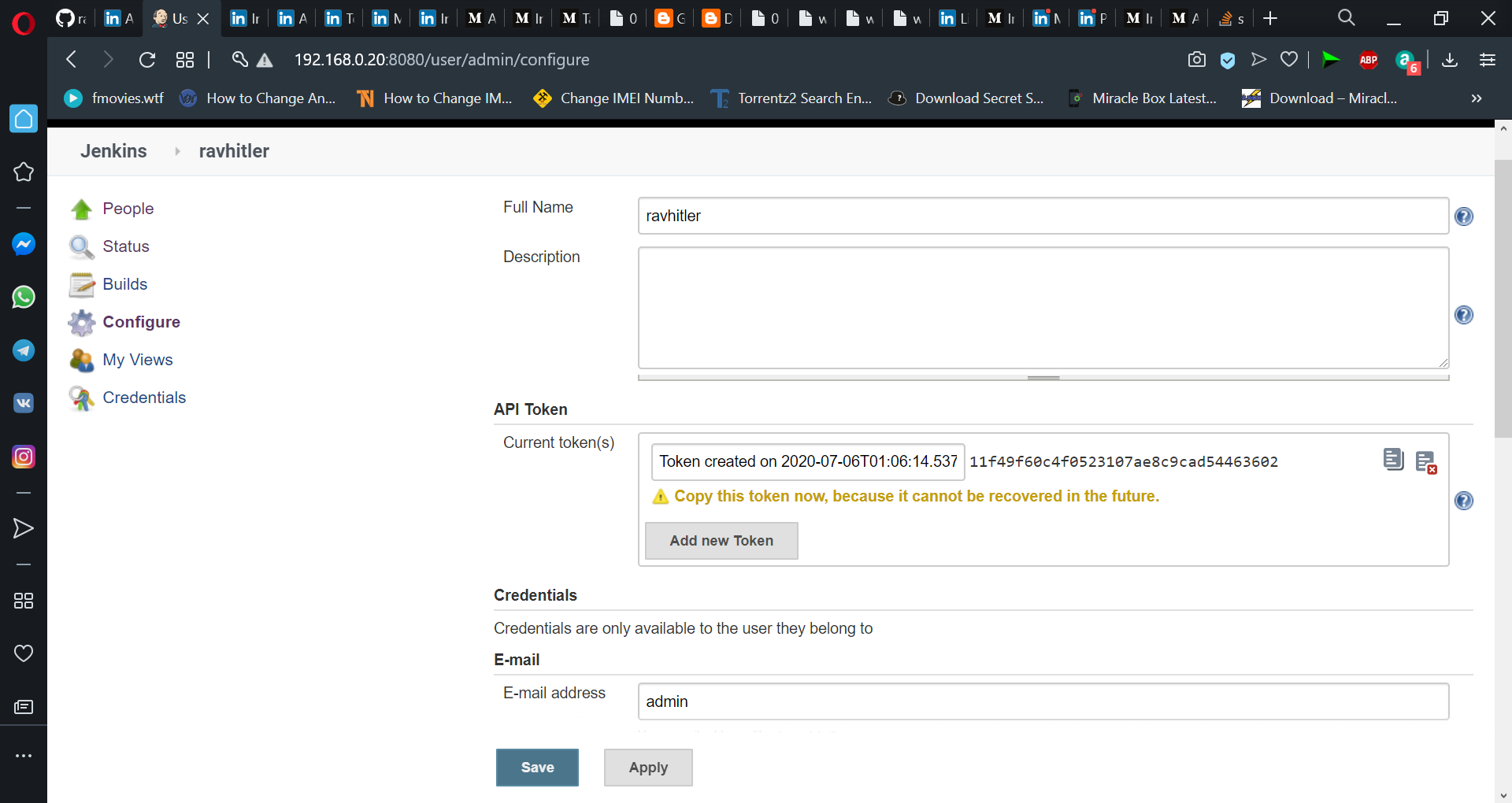


Apply and Save it. Job-1 is complete.

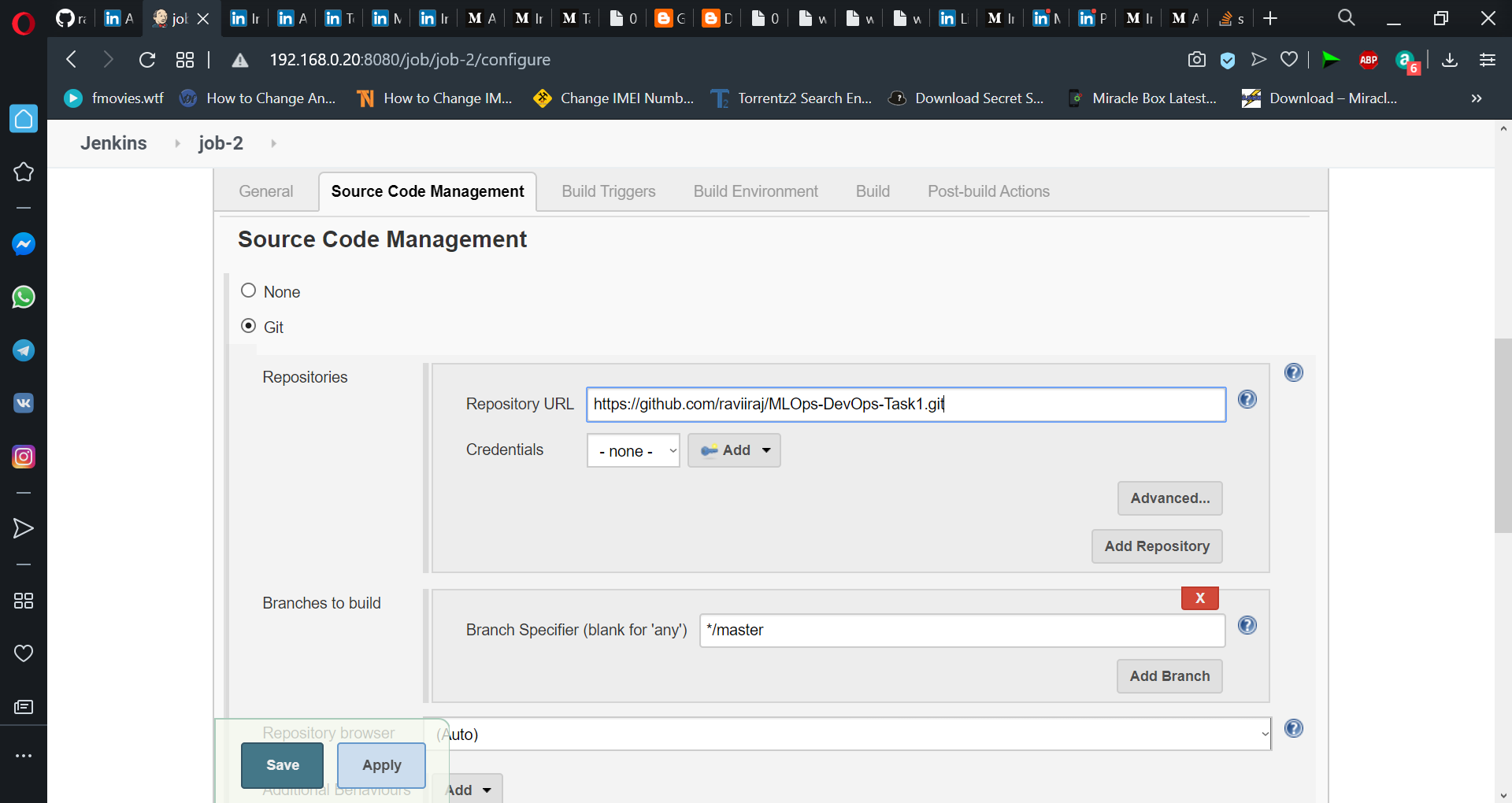
**#Job-2**

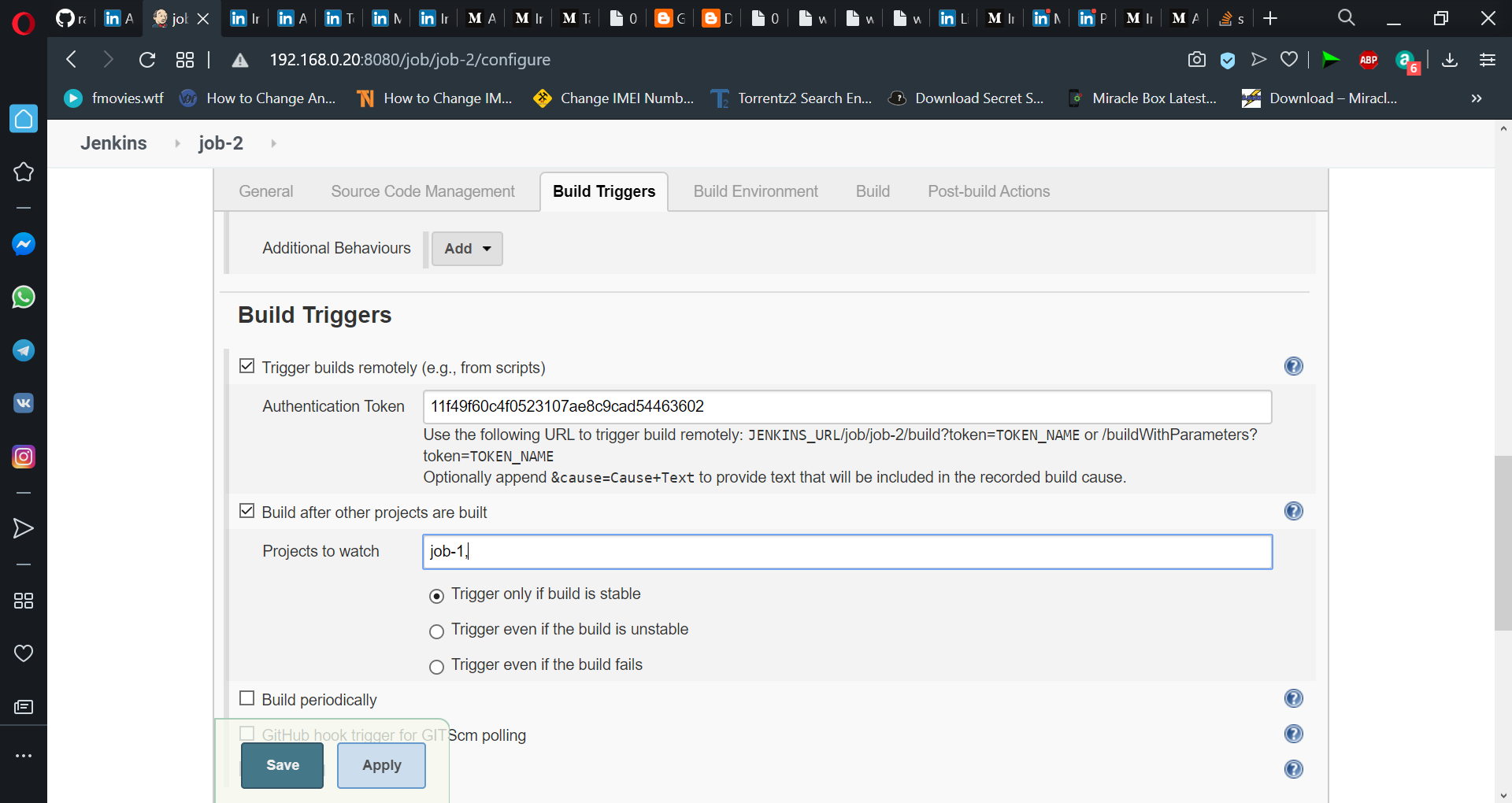
If developer push to master branch then Jenkins will fetch from master and deploy on the master-docker environment. Both dev-docker and master-docker environments are in different docker containers.

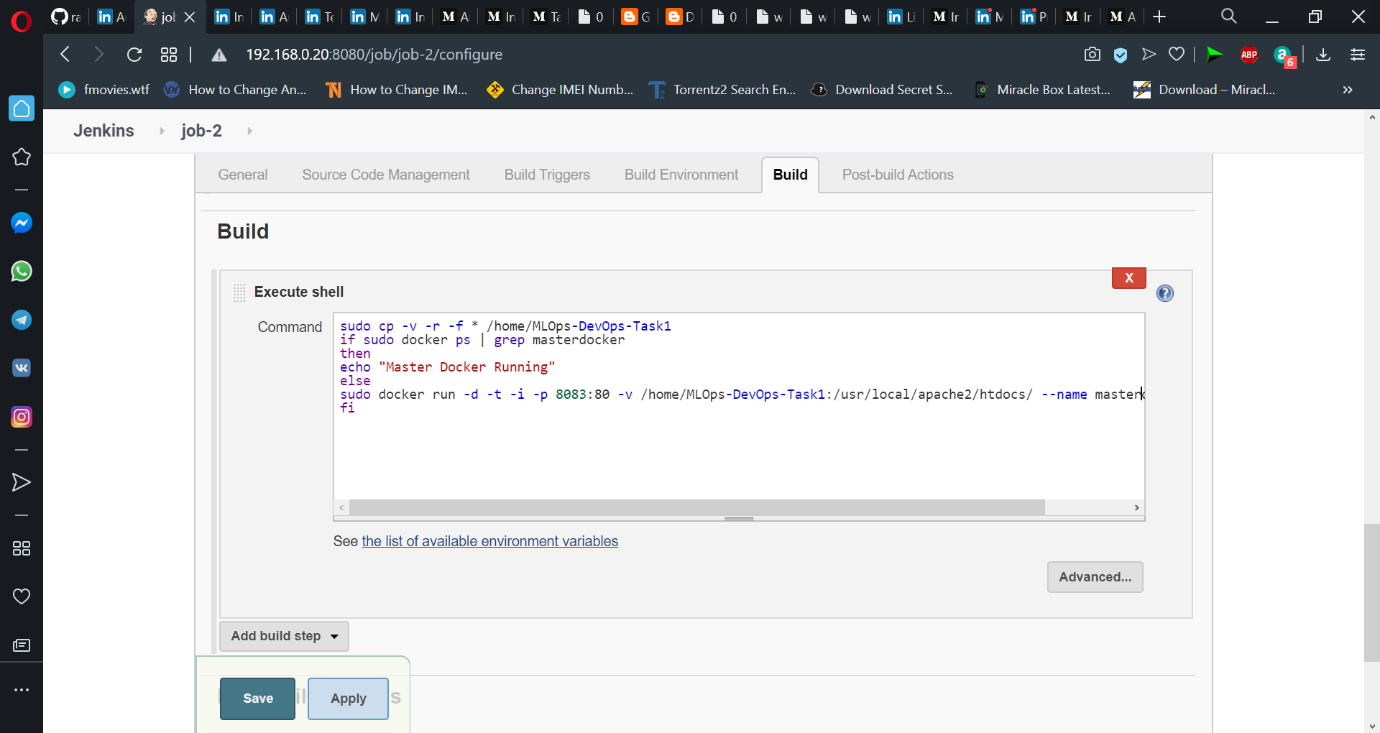
first create an API Token to trigger this job from job-3.



Now create job-2 in Jenkins.



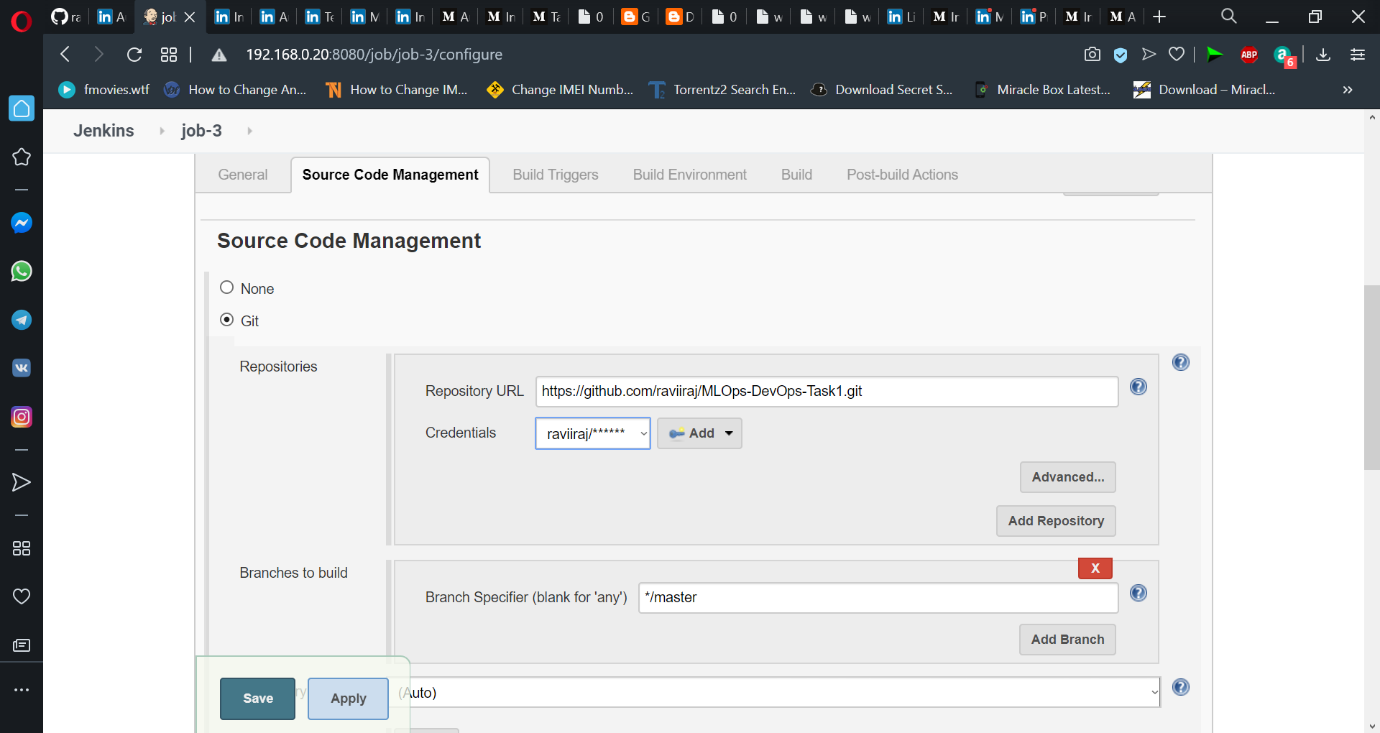


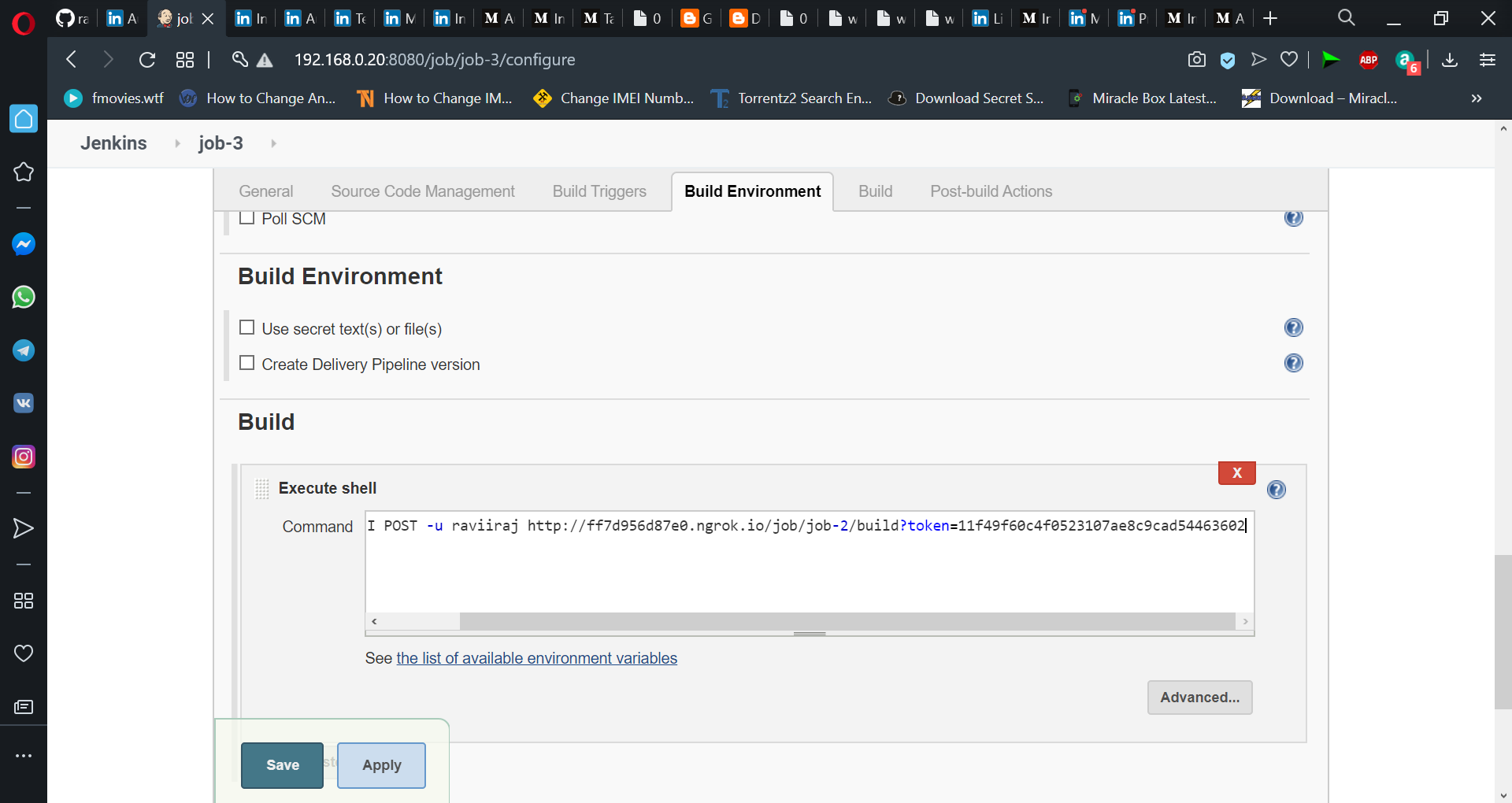


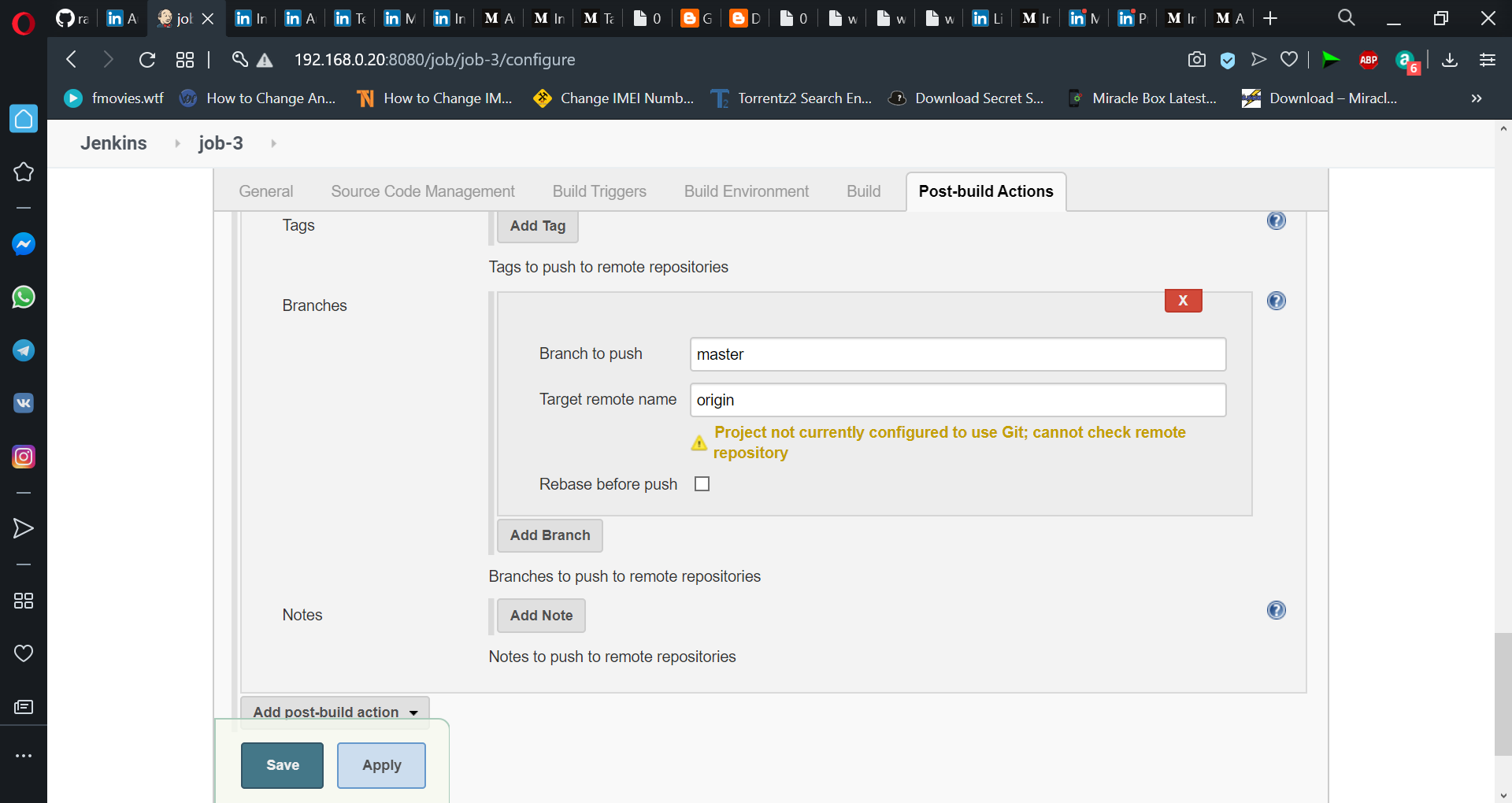
Apply and Save. Job-2 is complete

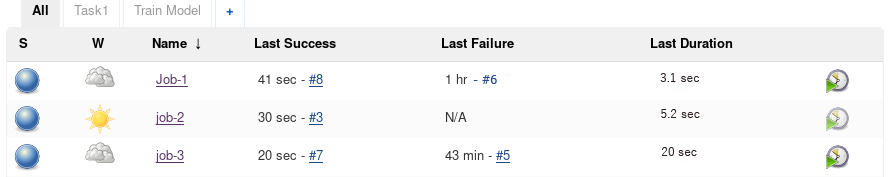
#Job-3

Manually the QA team will check (test) for the website running in the dev-docker environment. If it is running fine then Jenkins will merge the dev branch to the master branch and trigger Job-2.

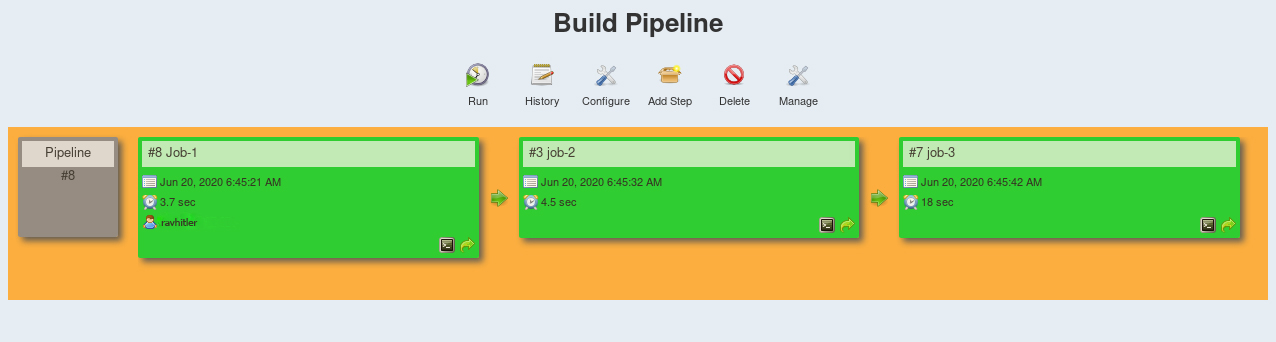






  Finished configuring all jobs.

Now see what happens !



Everything work’s fine and automated successfully.