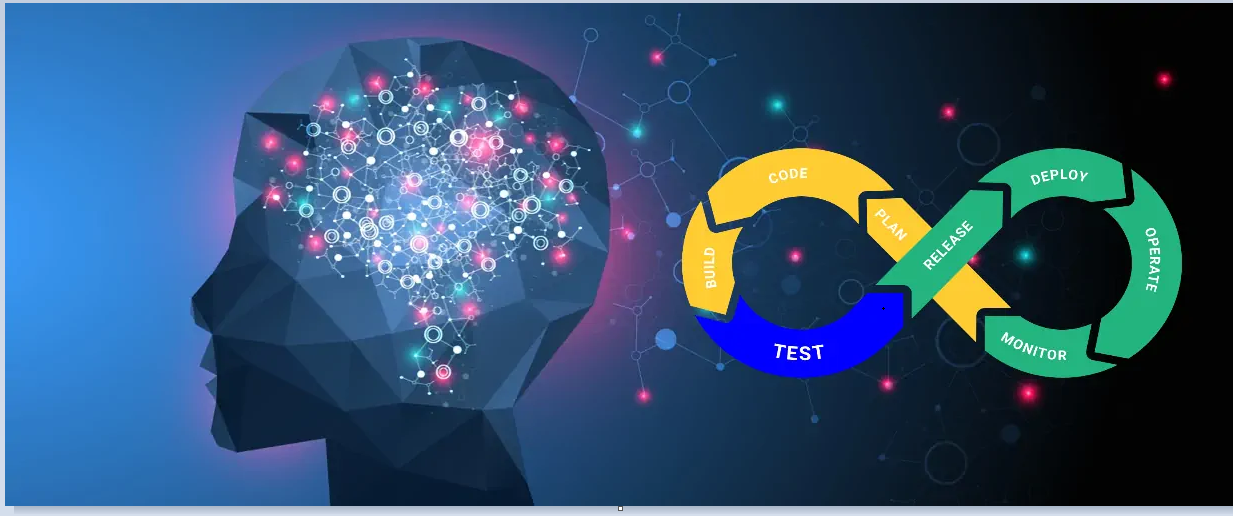
[ MLOps - Task - 3 ]



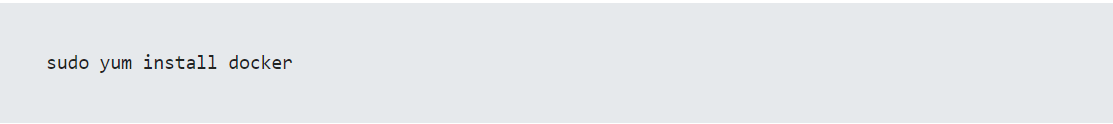
**Integration of Machine Learning with Devops (MLOps)**

**Train Machine Learning model automatically to achieve target accuracy. Also, Send an email to notify target accuracy is achieved.**

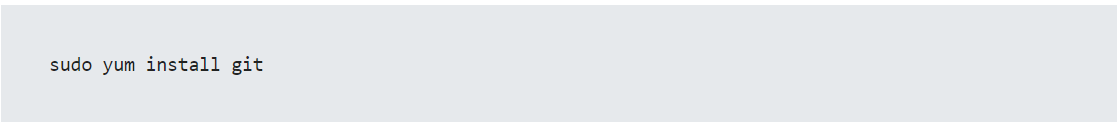
We will be using Jenkins Pipeline to achieve our task on RedHat Linux . But we have to first setup the environment

### Requirements:

* Rhel yum pre-configured
* Docker



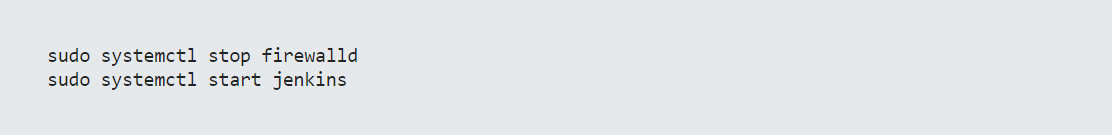
Git and a GitHub Account



Jenkins pre installed/Install it from below

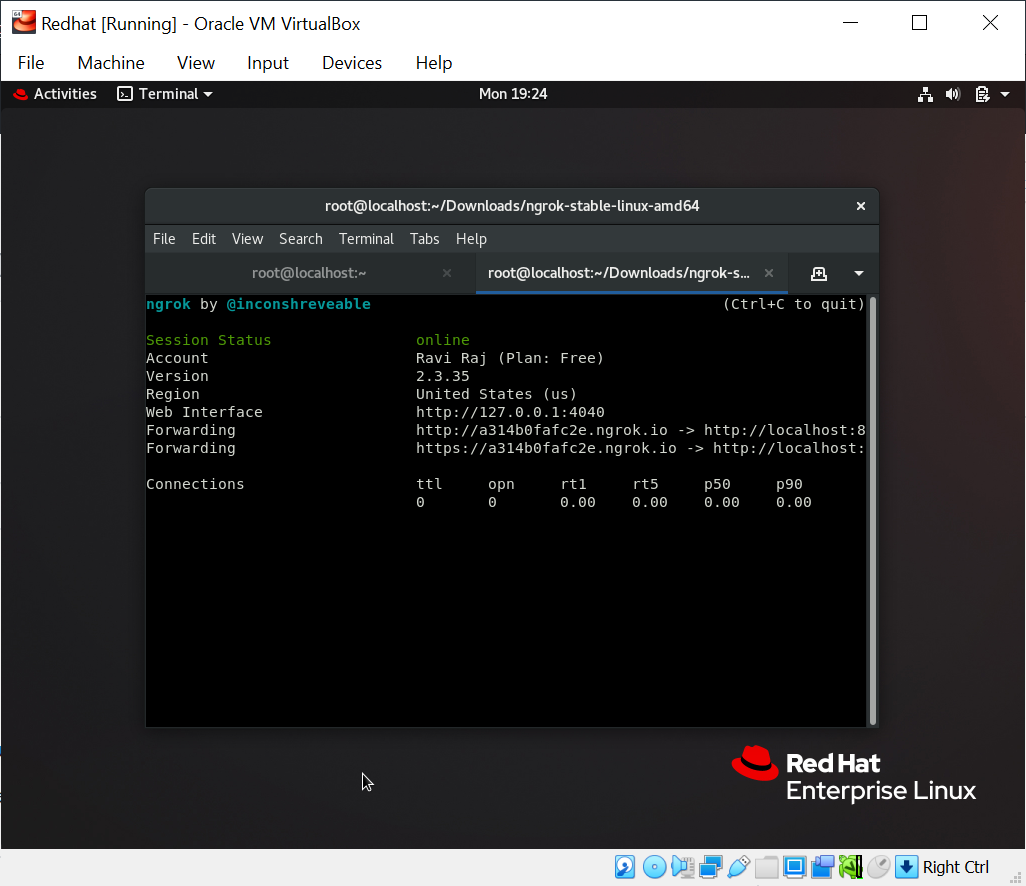
[www.jenkins.io](http://www.jenkins.io)

To ensure Jenkins is running.



Ngrok – Below is link for ngrok download.

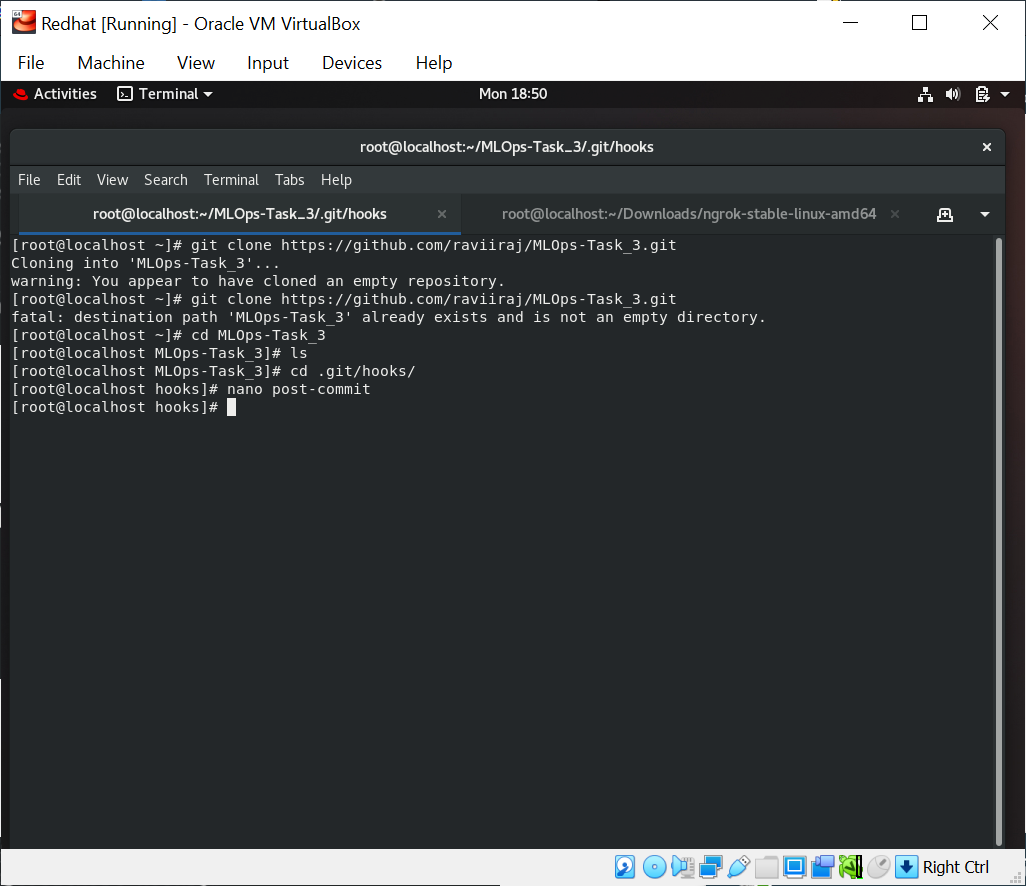
<https://ngrok.com/download>



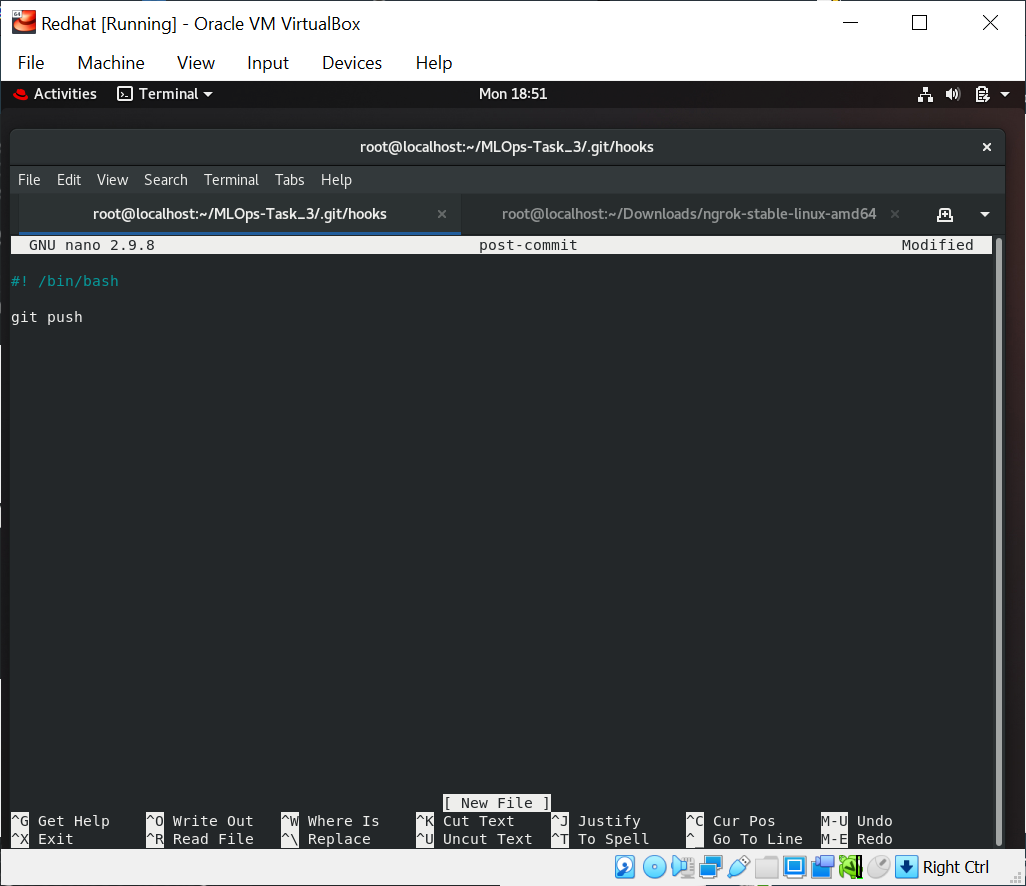
Highlighted url used to access port 8080 in localhost where jenkins is running.

### Building Docker image using Dockerfile

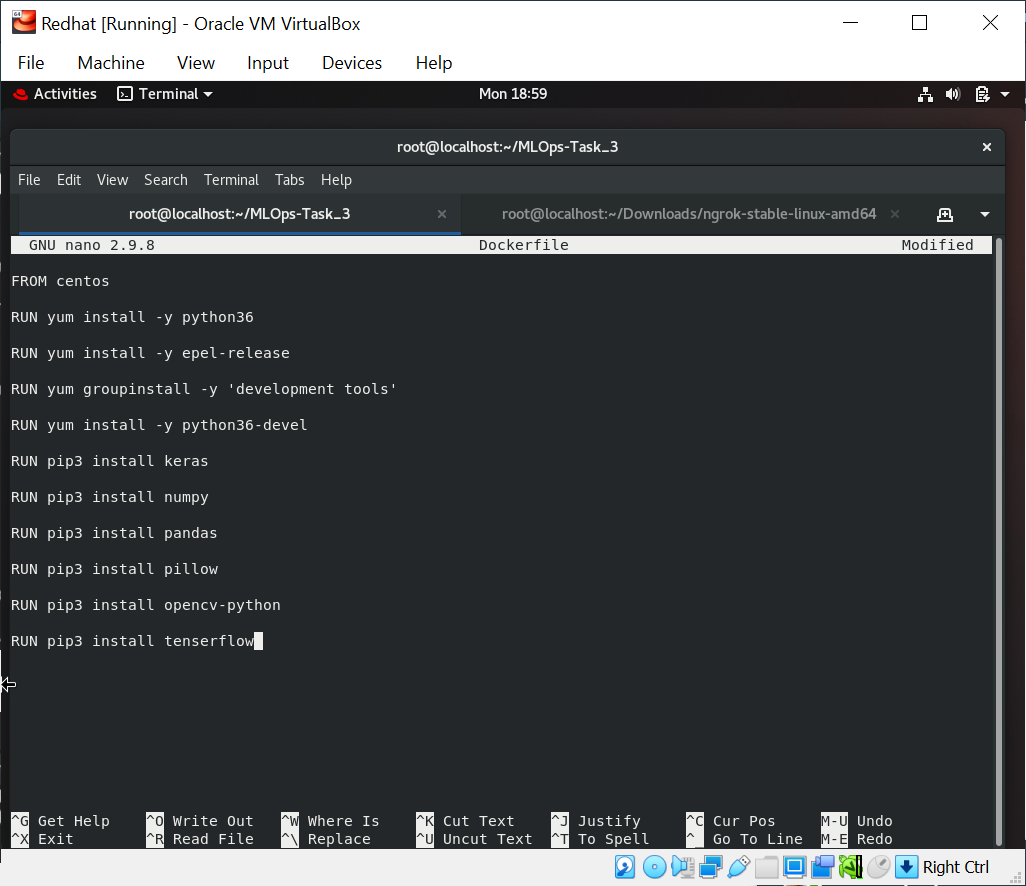
First clone the github repository and configure hook, so every time a commit happens, automatically pushed to GitHub.



Post-commit file creation.

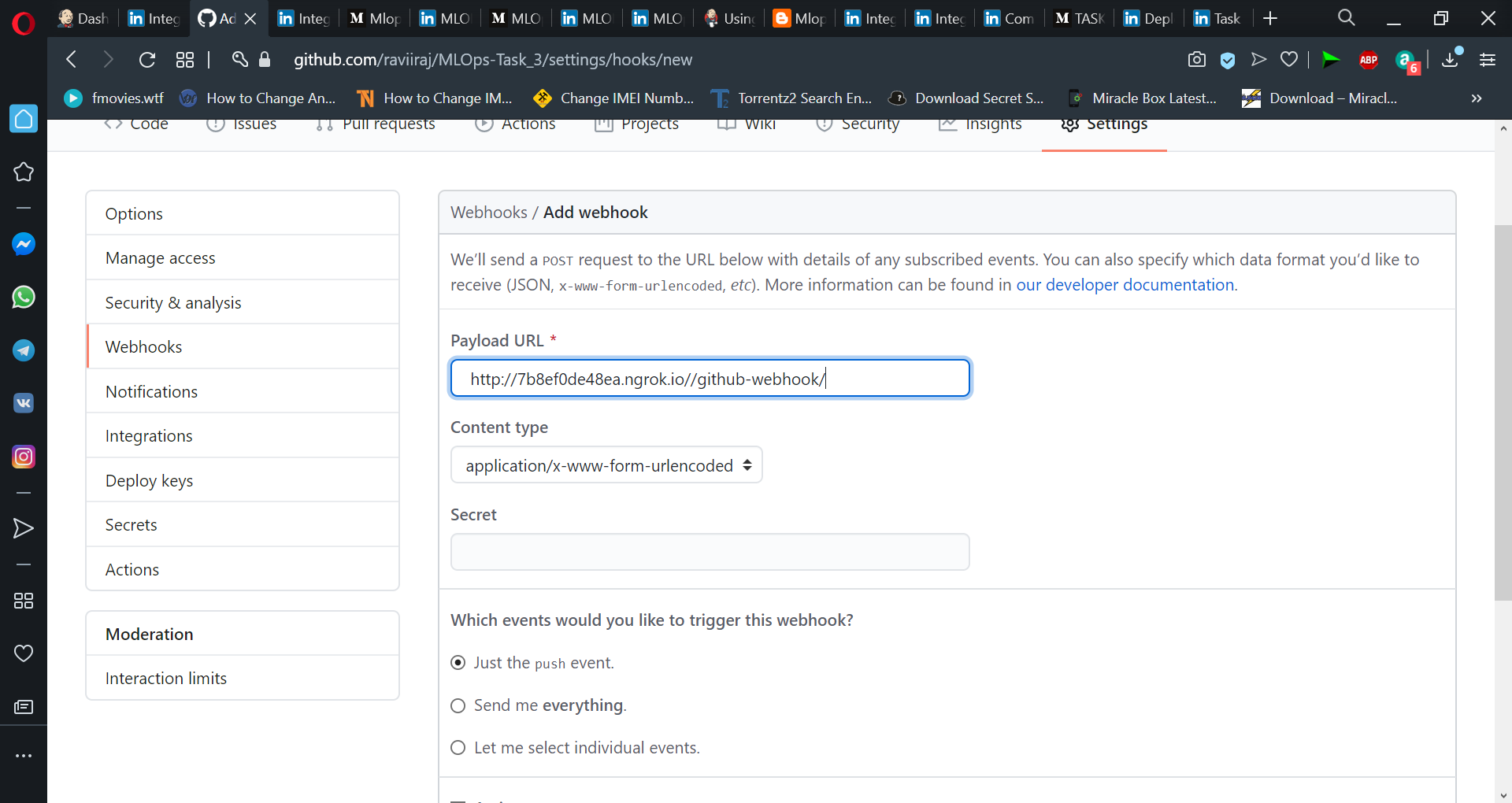


# Dockerfile and build it.

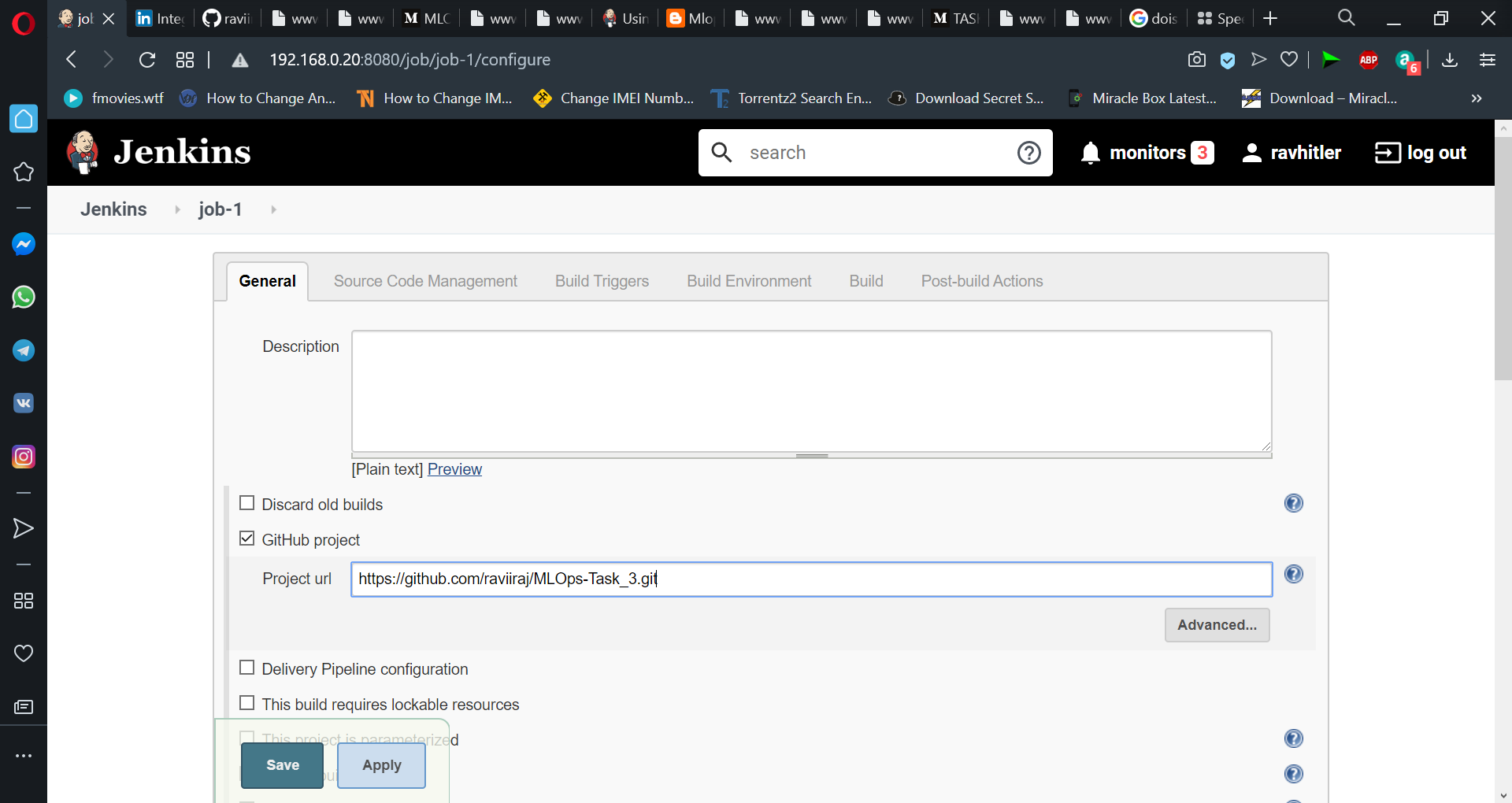


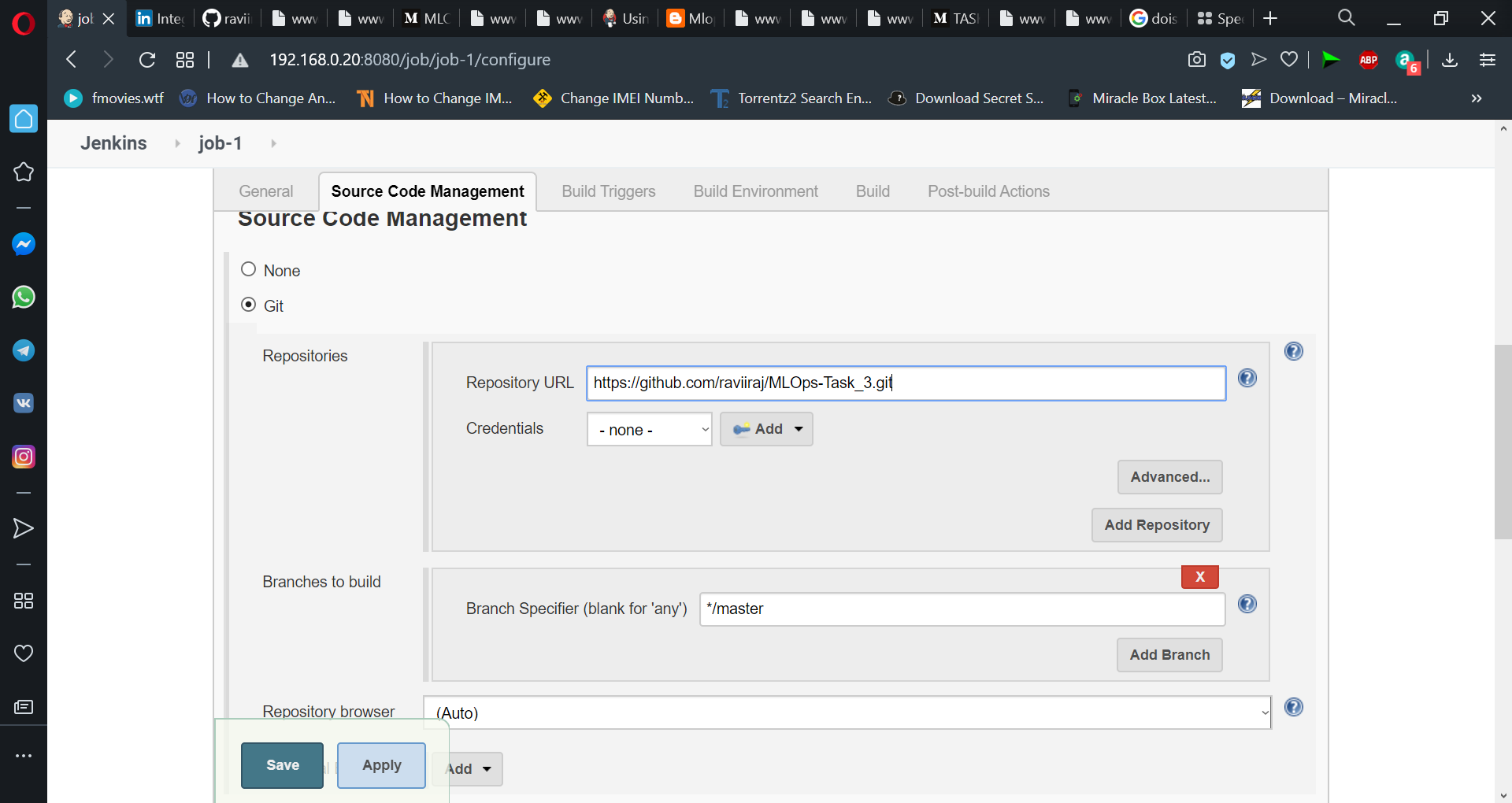
### Webhook and Job1 setup

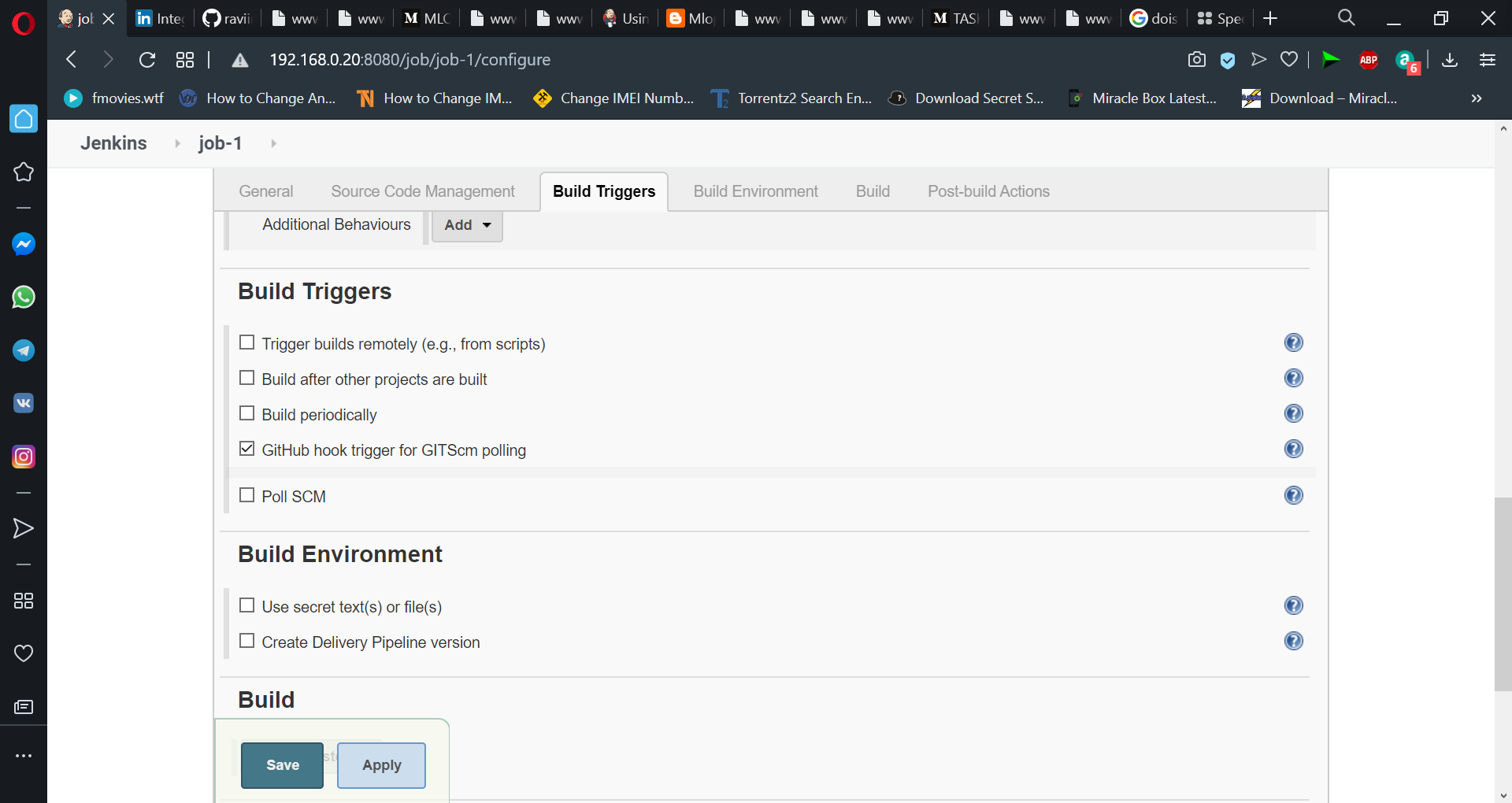
Job -1: Pull the Github repository automatically when any developer push repo to Github. Add a webhook after selecting Webhooks from the Settings tab and click on Add webhook. In this case, we are using URL given by ngrok as Jenkins URL

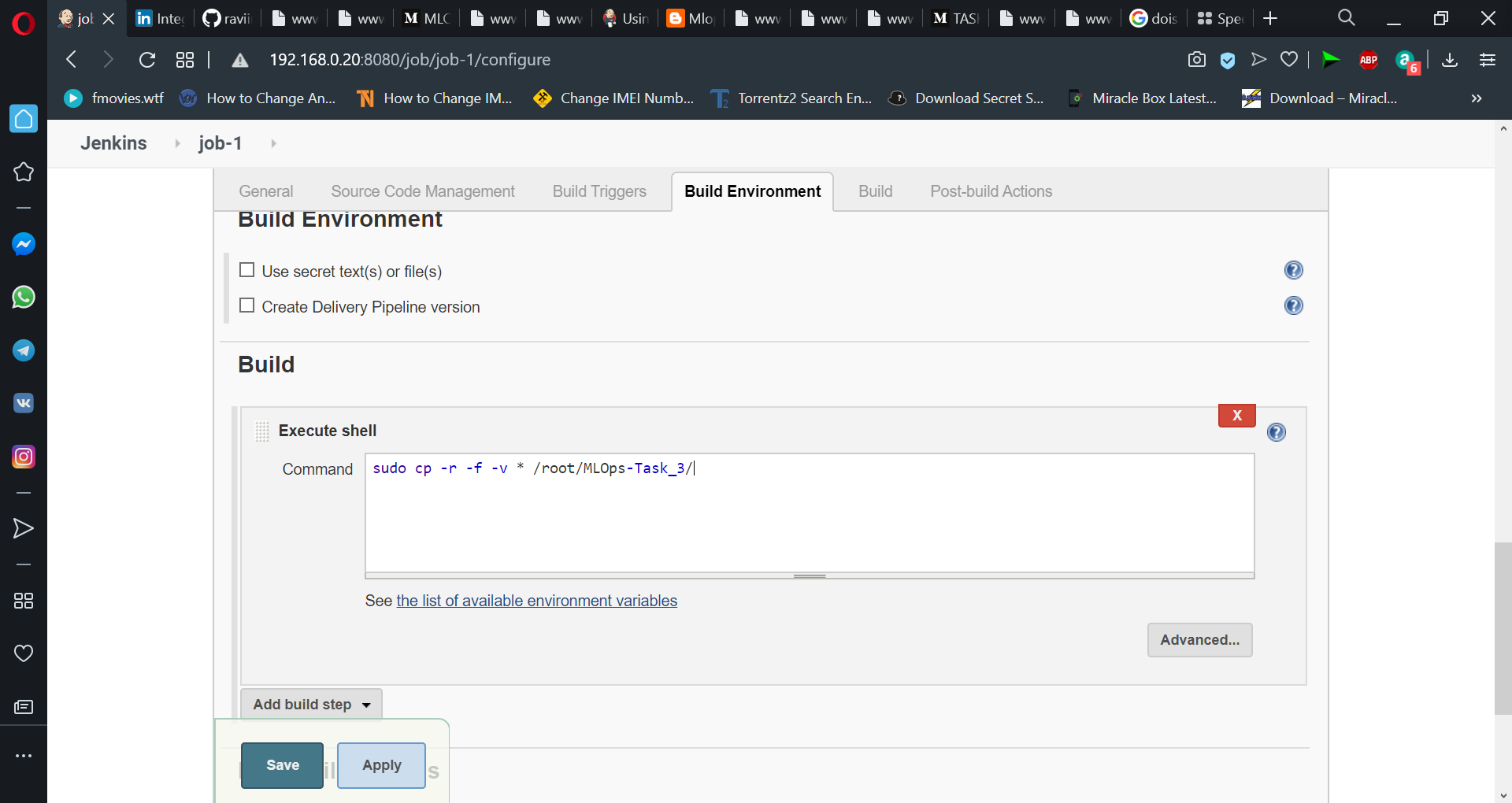


# Add webhook, let's configure our job-1 in Jenkins.

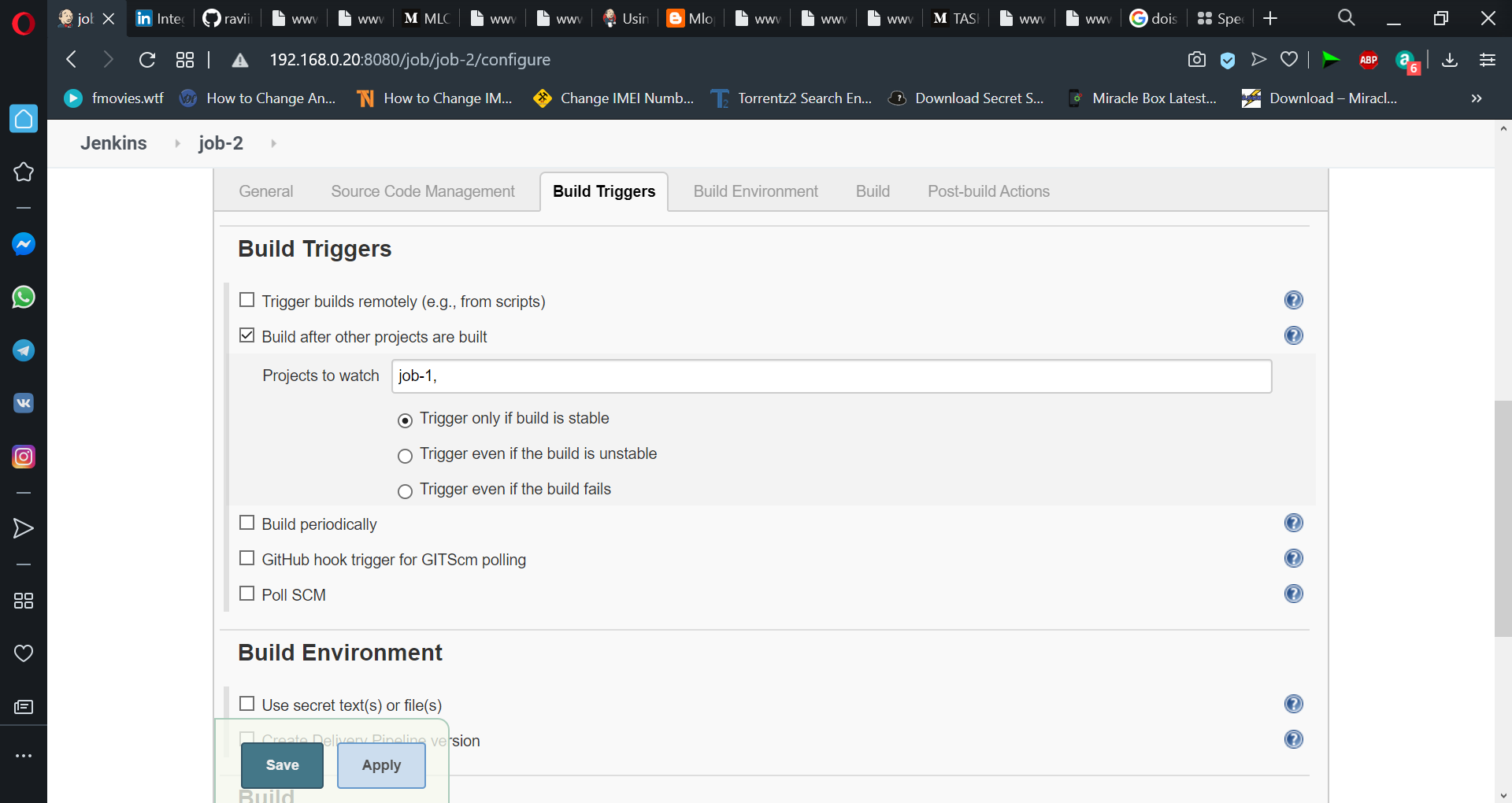


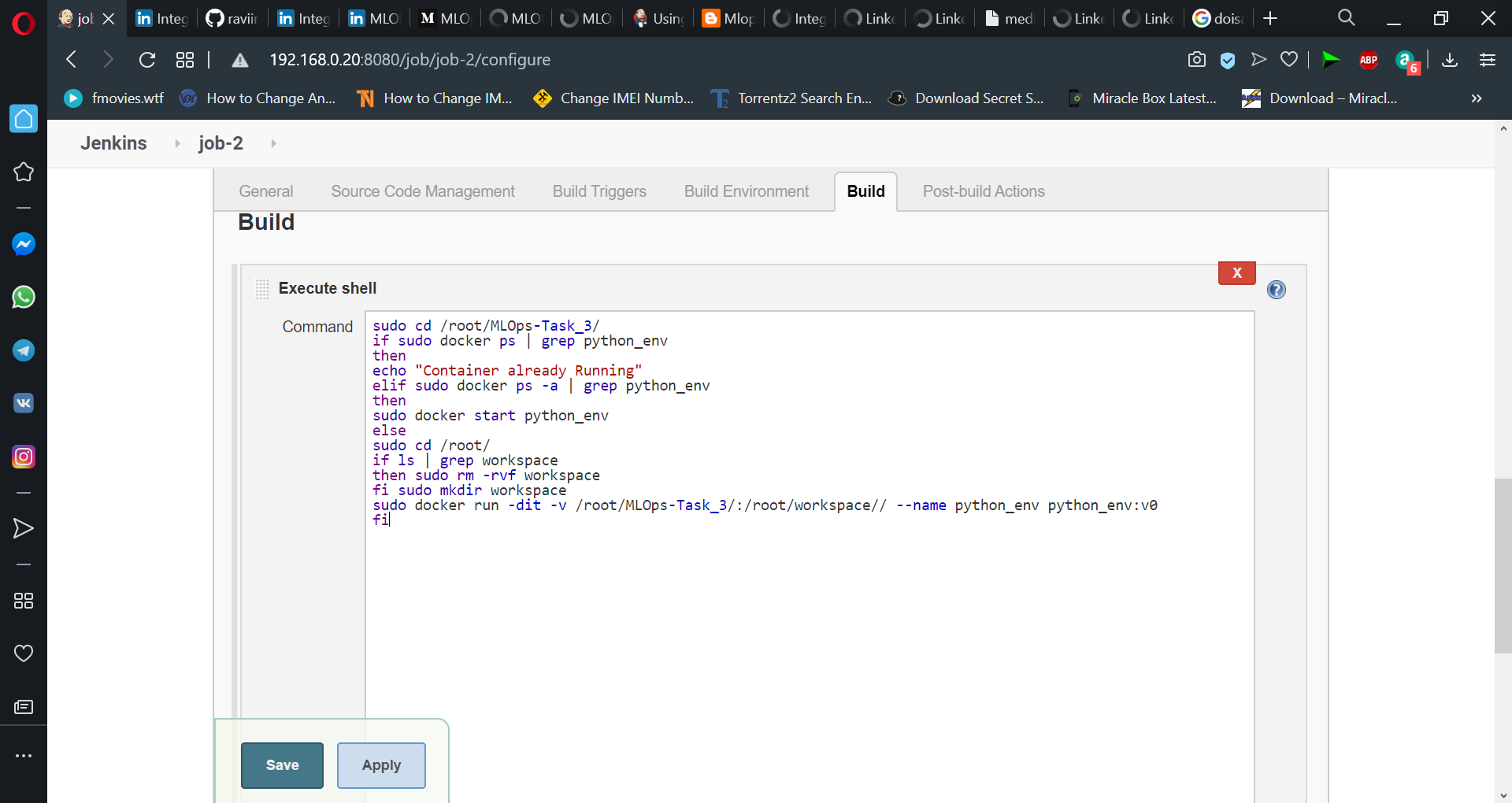






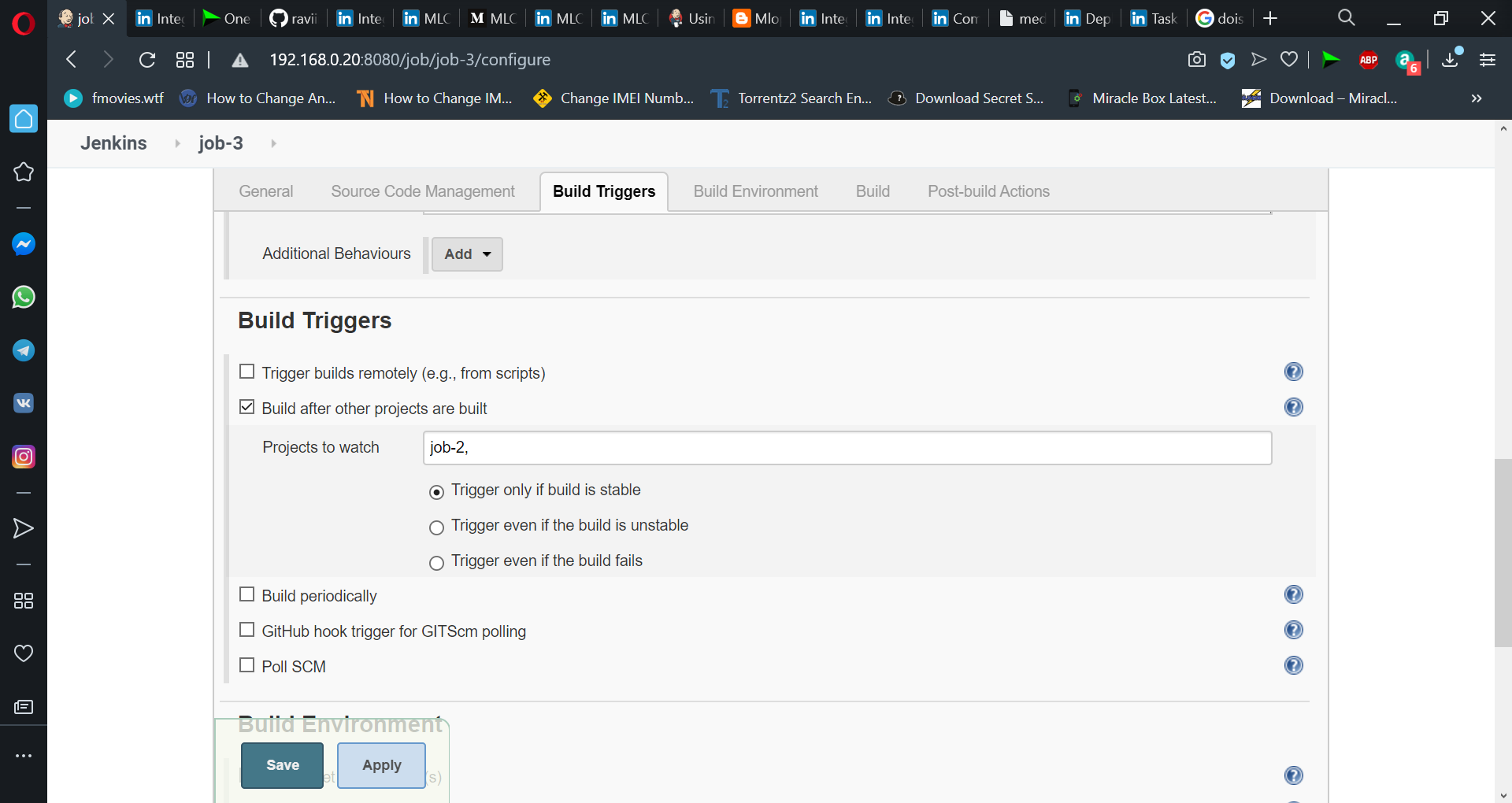
### Setting up Job2 and Deploying container in Jenkins

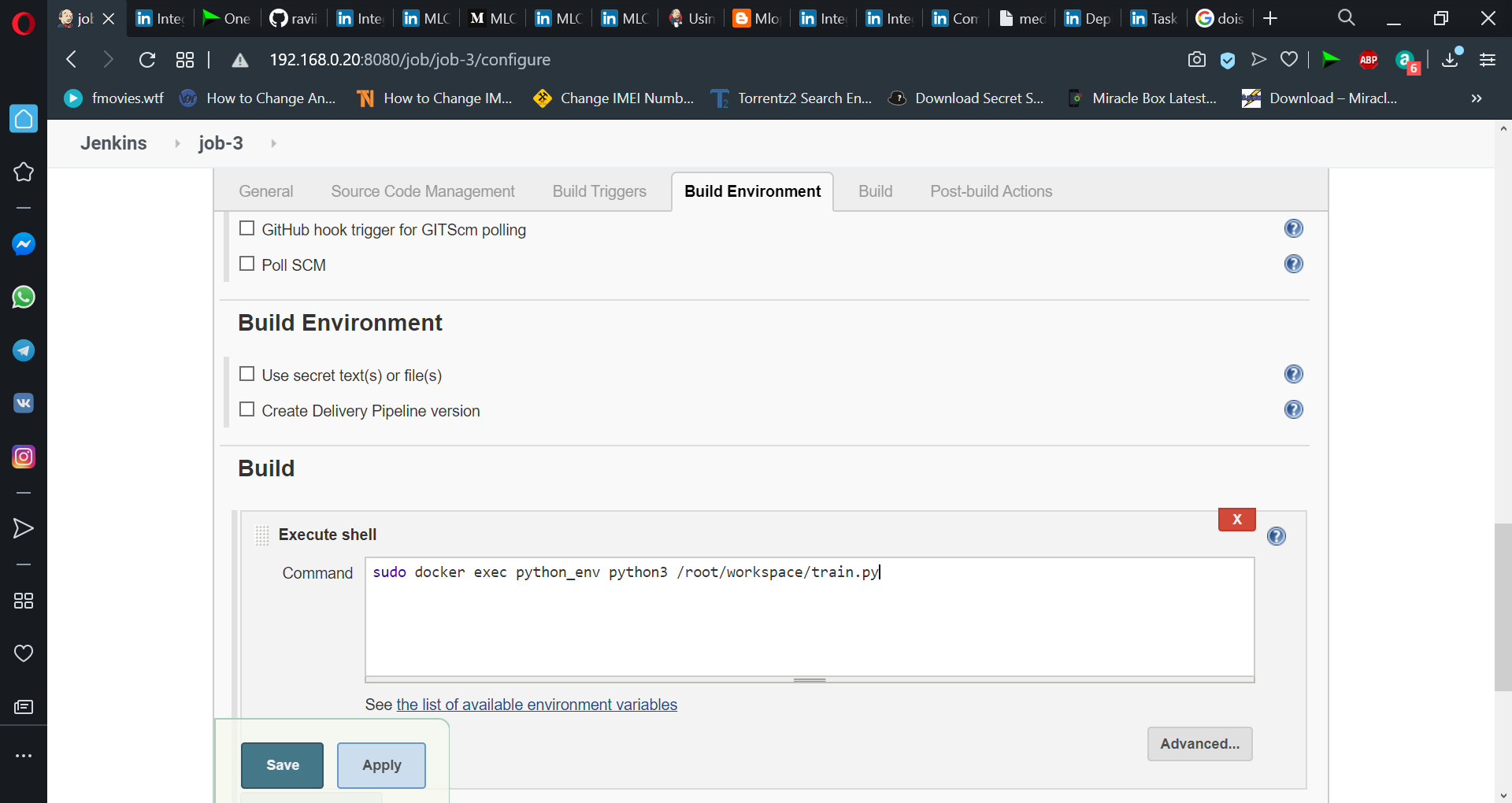


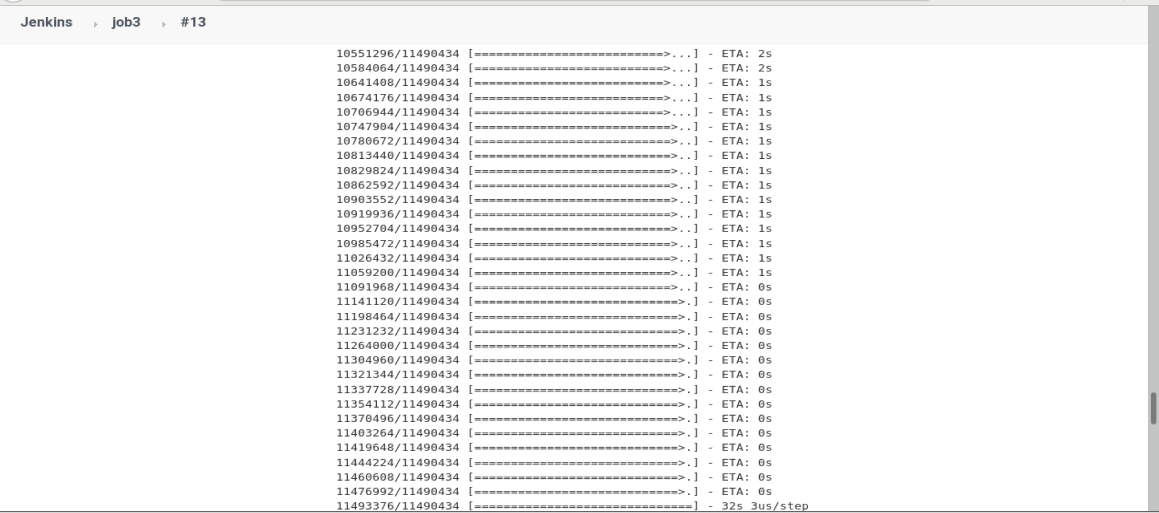


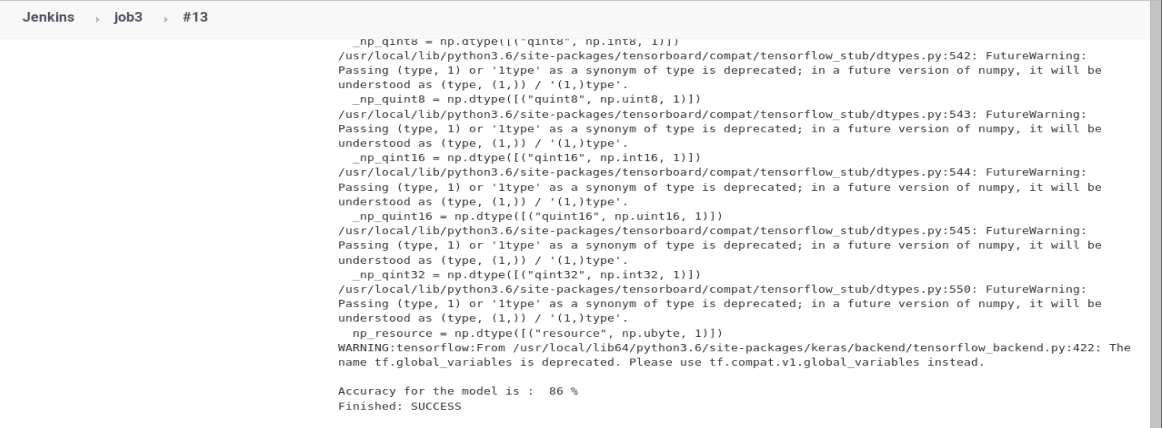
### Setting up Job3 and training model in Jenkins.

Train your model and predict accuracy. Link this job with a previous job using Build after other projects are built and select job2 similar as we did in the previous step. We can start to train our model in the container we just built and store the accuracy in accuracy.txt file.



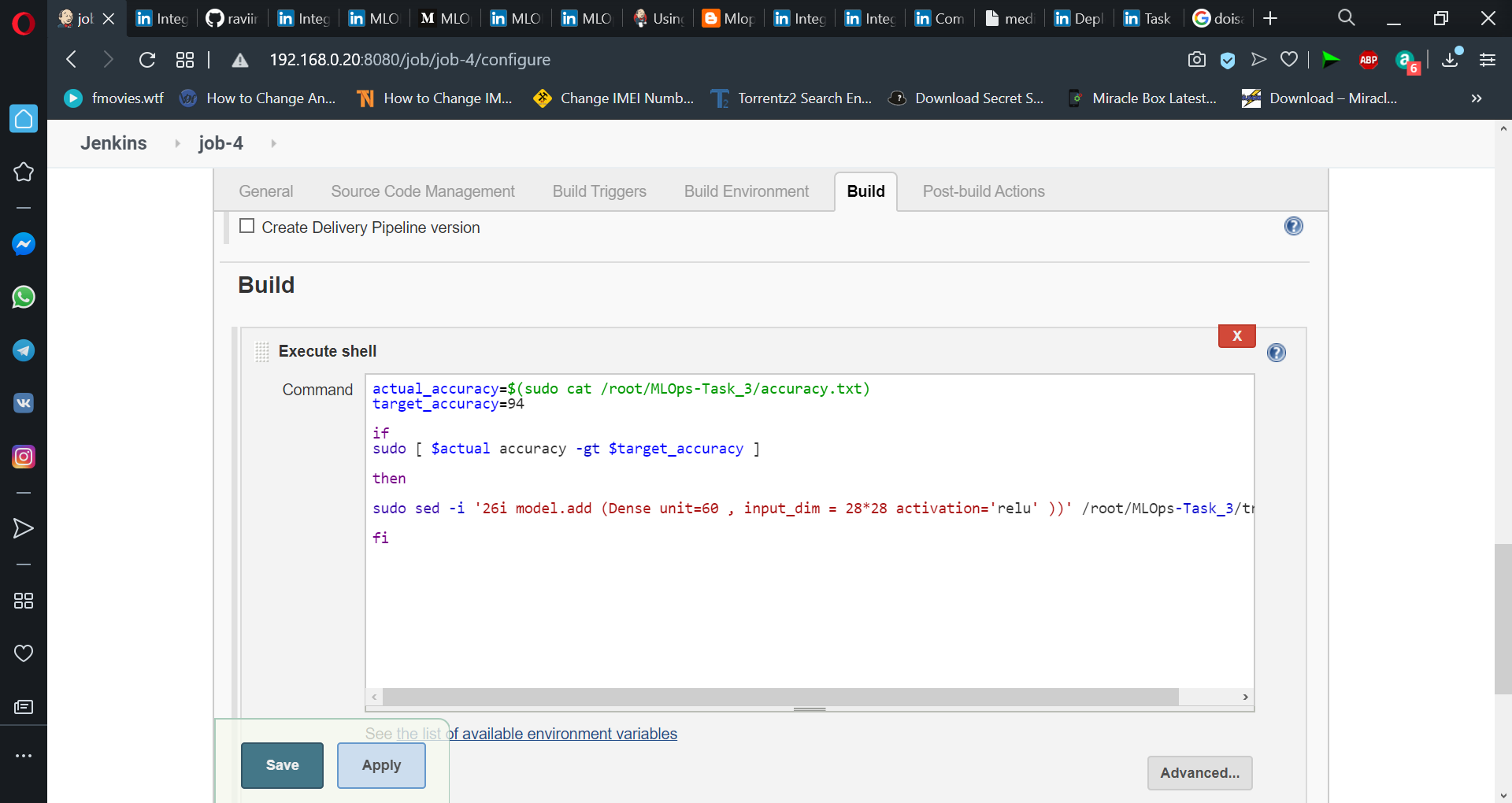






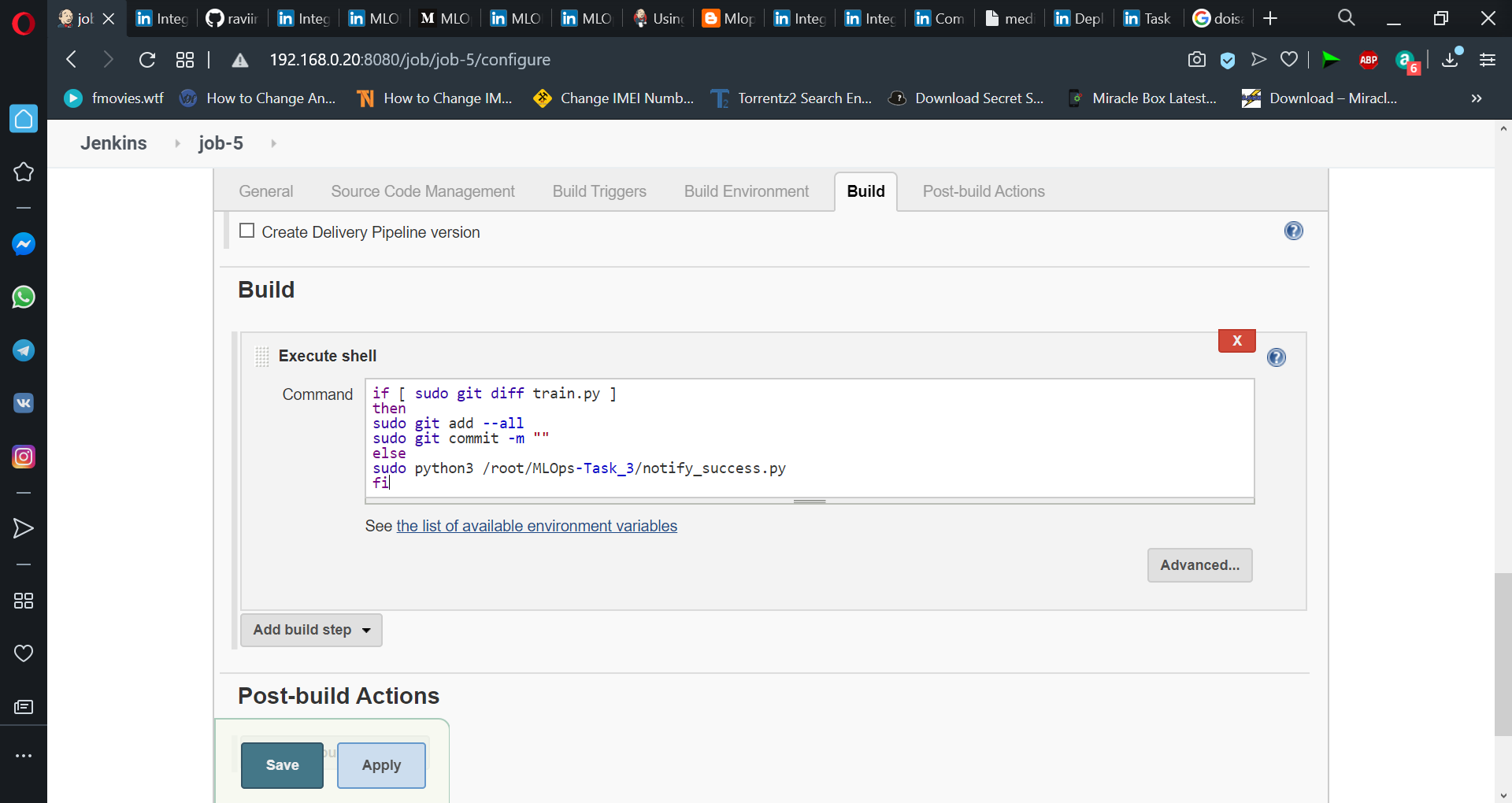
### Setting up Job4 and Tweaking ML model in Jenkins

If accuracy is less than 94%, then tweak the machine learning model architecture.Here we are adding another Dense layer to our model architecture by adding bellow line to /home/sahil/MLOPsTask3/train.py file if the actual\_accuracy is less than target\_accuracy.



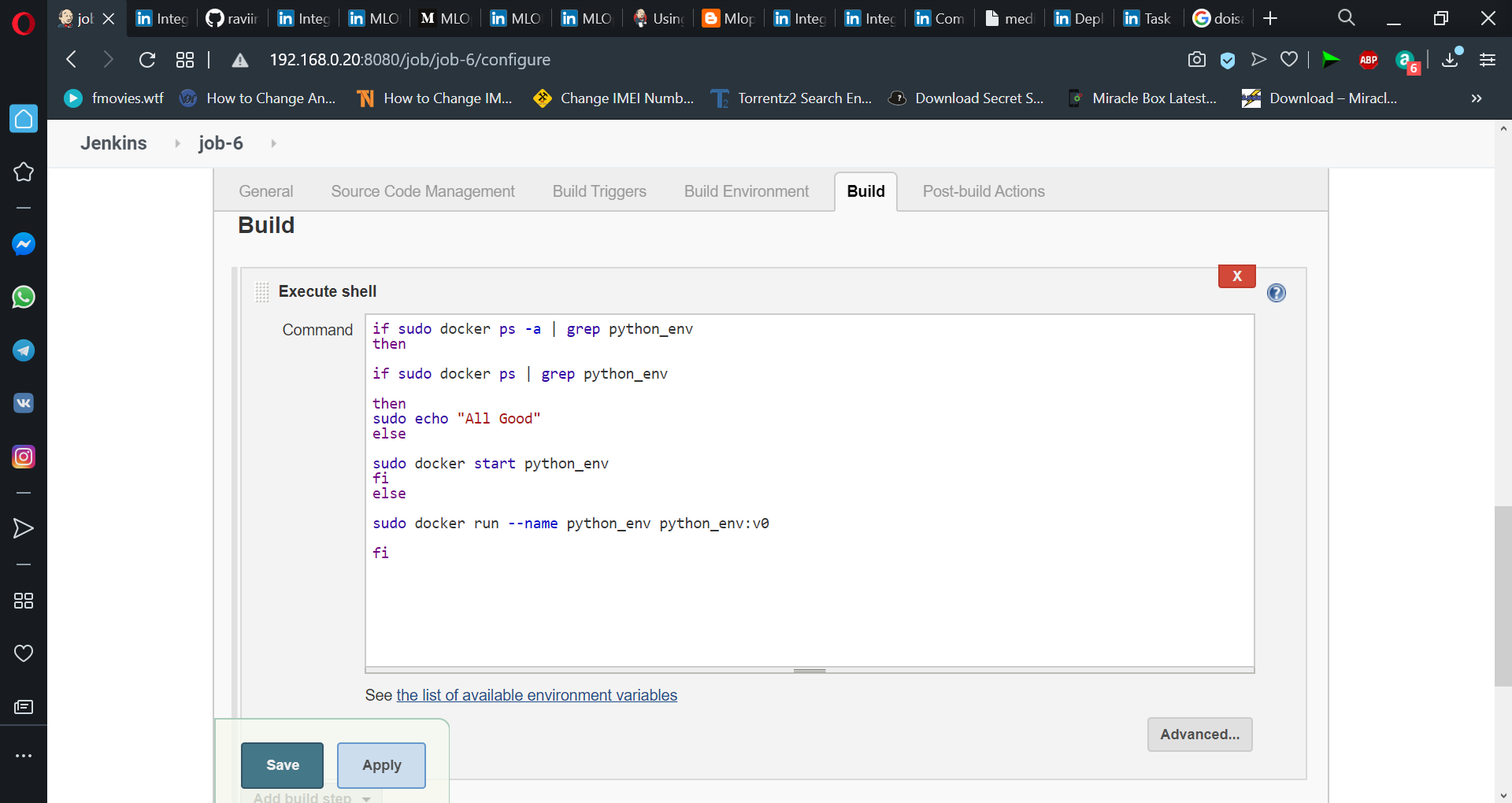
Job-5: Retrain the model. For retraining our model we are checking for changes in the model file (train.py) which indicates if the architecture has been changed due to failure in achieving target accuracy, else an email is sent to notify success by running notify\_sucess.py file.

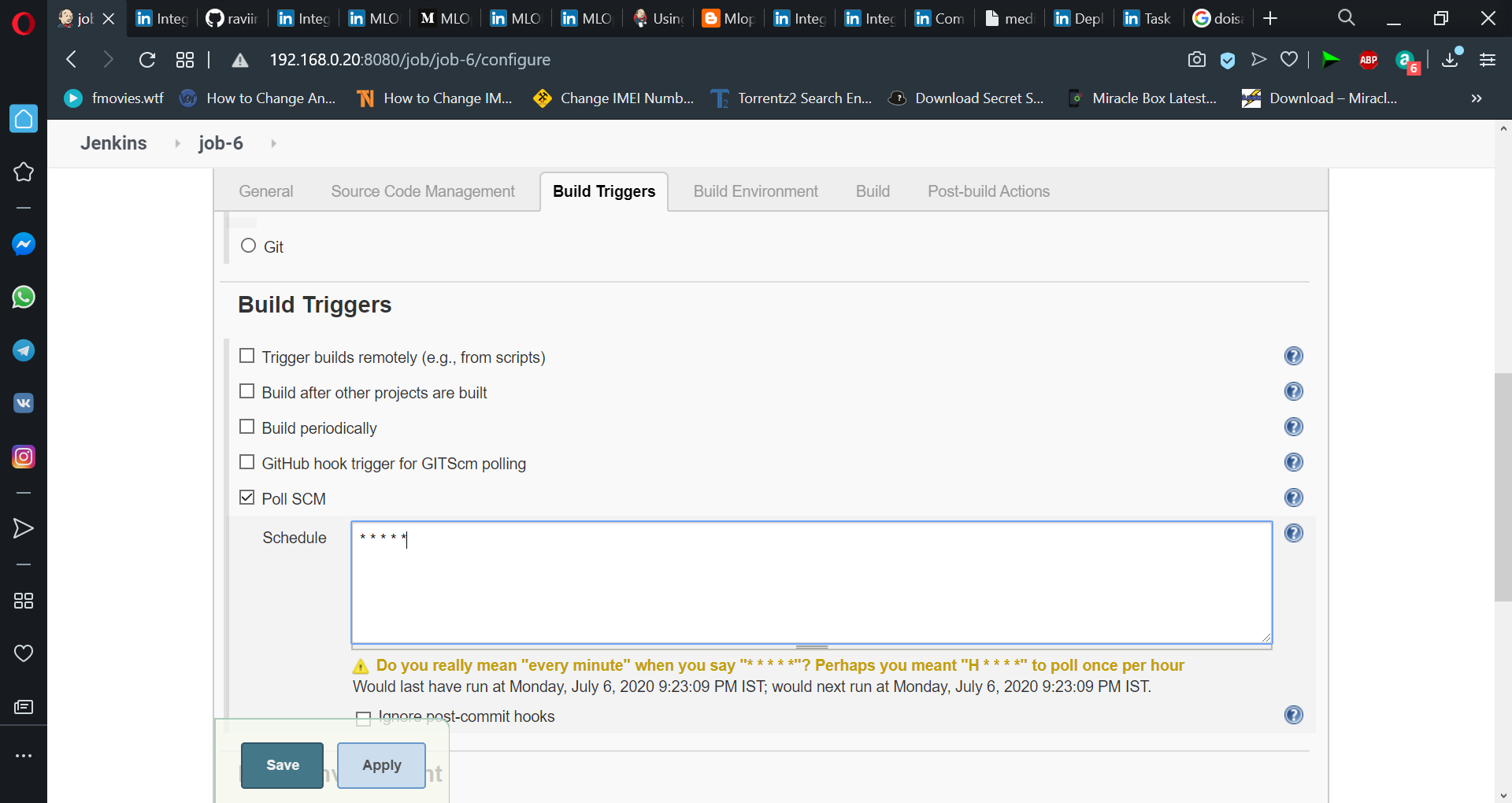
In case of any changes in model architecture, the changes are committed which automatically triggers a push (as seen in Step 1 using hook). And hence when a webhook is triggered to run Job1, the whole process starts building again.



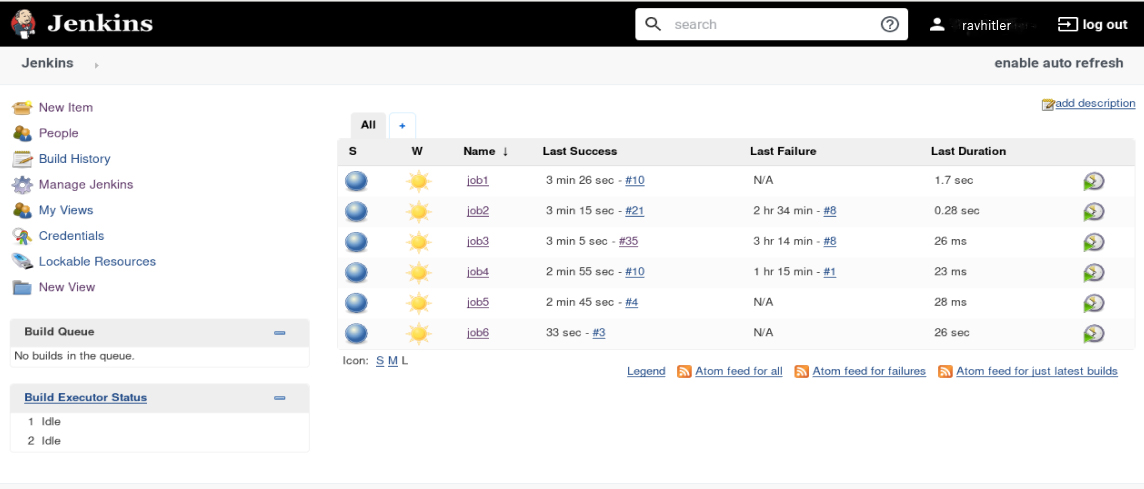
Job - 6 : Launching another container if one container fails.

Here we are using Poll SCM to check for failure every minute.

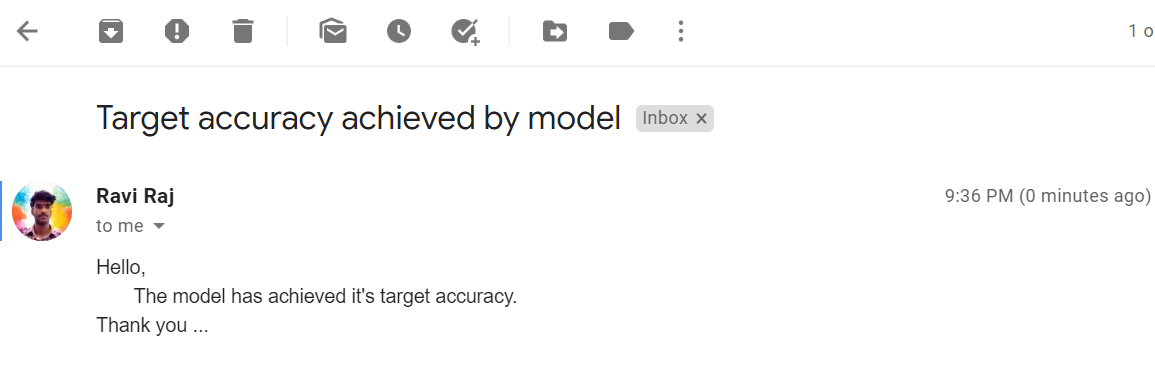
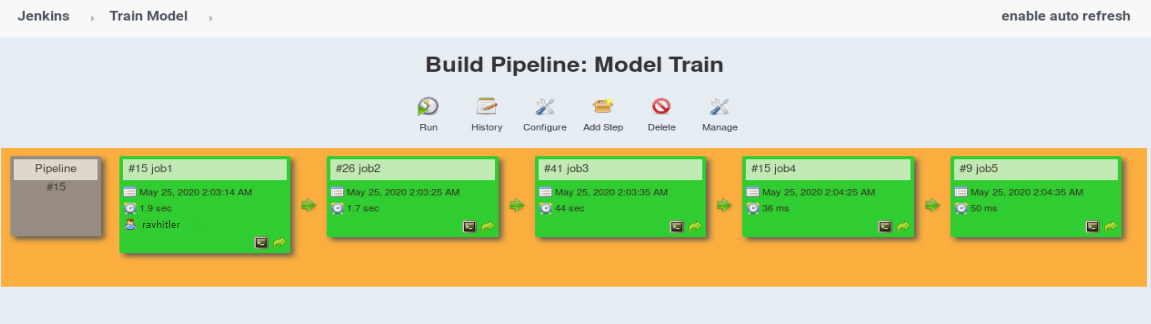
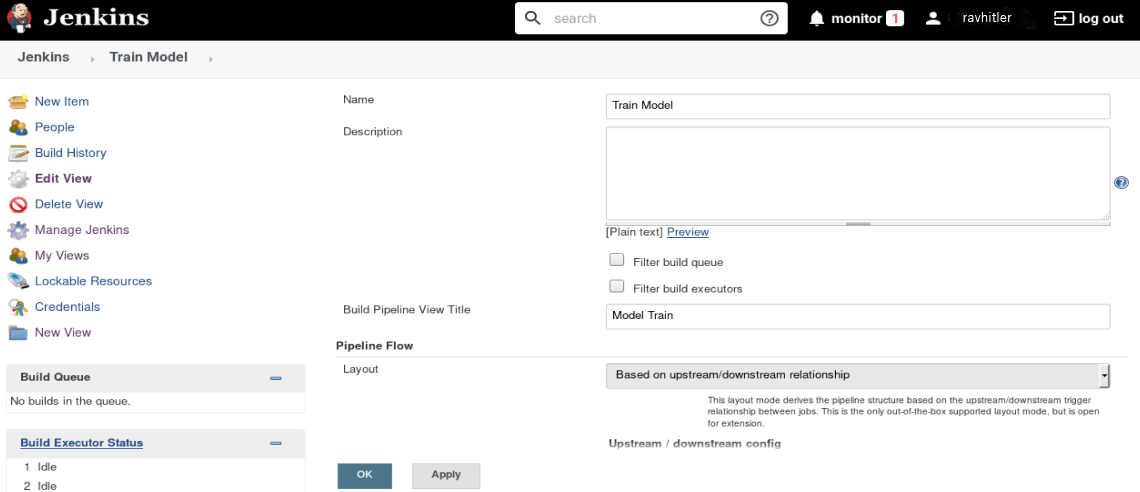
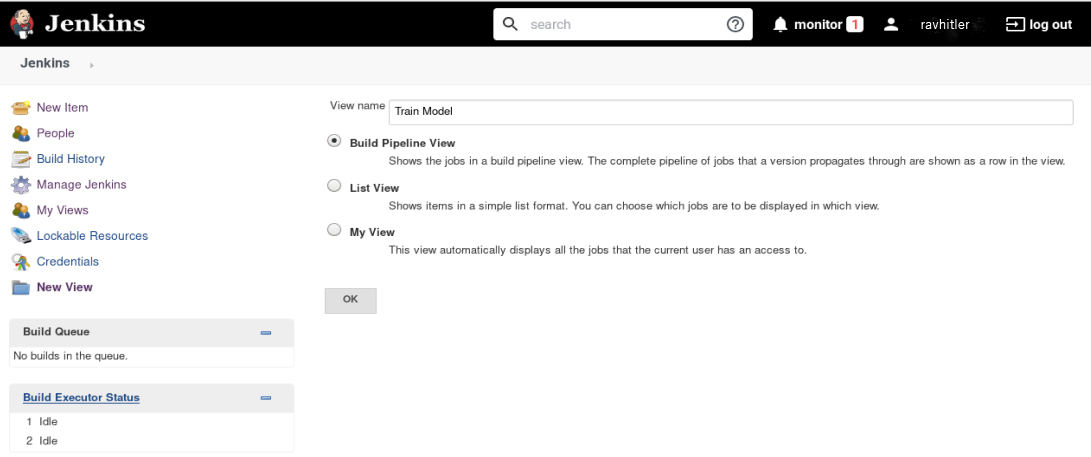




Job is working fine.



Visualising pipeline in a better way by installing Build Pipeline plugin and creating a Build Pipeline View.



**Achieved target accuracy of 93-95% approx. Everything is explained with figure above, you can create it easily, as I did. Thanks to Vimal sir for Teaching us such a great Technology.**