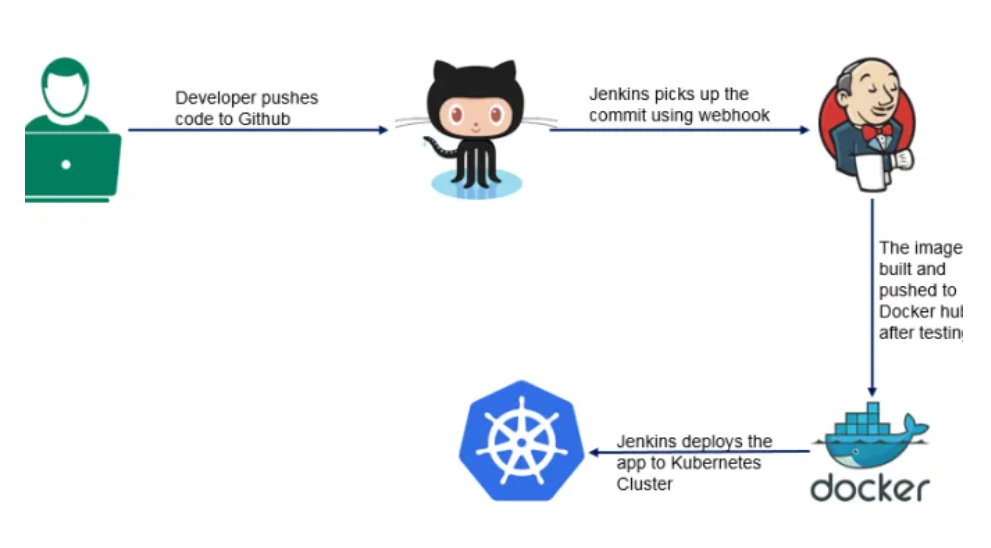
-------DOCUMENTATION-------

Integration of CI/CD Tool with Dynamic Clusters

DevOps Task 3 Given By Vimal Daga Sir..



**#Task Description**

1. Create container image that’s has Linux and other basic configurationrequired to run Slave for Jenkins. ( example here we require kubectl to be configured )

2. When we launch the job it should automatically starts job on slave based on the label provided for dynamic approach.

3. Create a job chain of job1 & job2 using build pipeline plugin in Jenkins

4. Job1 : Pull the Github repo automatically when some developers push repo to Github and perform the following operations as:

4.1 Create the new image dynamically for the application and copy the application code into that corresponding docker image

4.2 Push that image to the docker hub (Public repository)

( Github code contain the application code and Dockerfile to create a new image )

5. Job2 ( Should be run on the dynamic slave of Jenkins configured with Kubernetes kubectl command): Launch the application on the top of Kubernetes cluster performing following operations:

5.1 If launching first time then create a deployment of the pod using the image created in the previous job. Else if deployment already exists then do rollout of the existing pod making zero downtime for the user.

5.2 If Application created first time, then Expose the application. Else dont expose it.

**Pre-requisites For This Project :-**

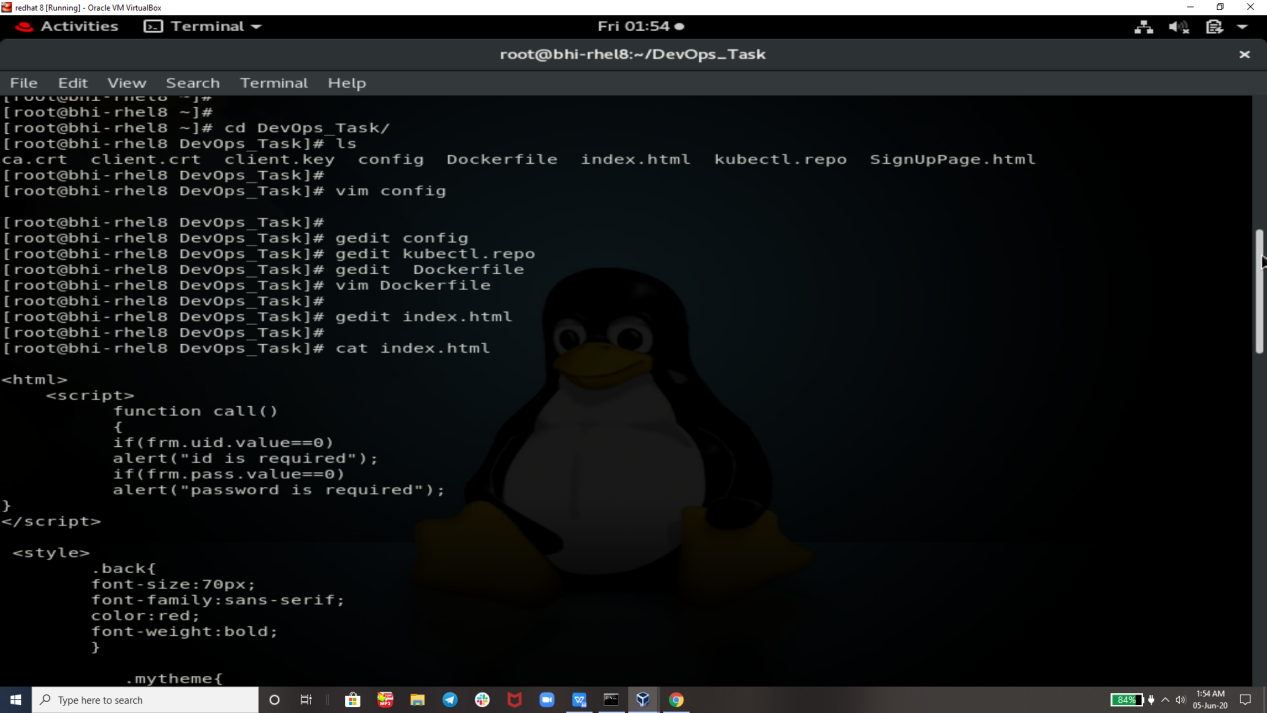
- Base OS in my case it is Windows 10 and On the Top of Windows 10 I am using Red Hat Enterprise Linux (RHEL 8 ) with the Help of Virtual Box

- Inside Rhel we need some setup to be ready like we need docker , Jenkins , centos image like I am using centos latest and and using that image I have created one Dockerfile by installing some required softwares like httpd , kubectl, java , ssh , git , php and some other packages and in this image kubectl is cofigured so that we can use it furthur

- And we need Minikube for k8s Setup and for using kubectl command we need minikube in active state

So Let’s Start Here is my Dockerfile And For using This :-

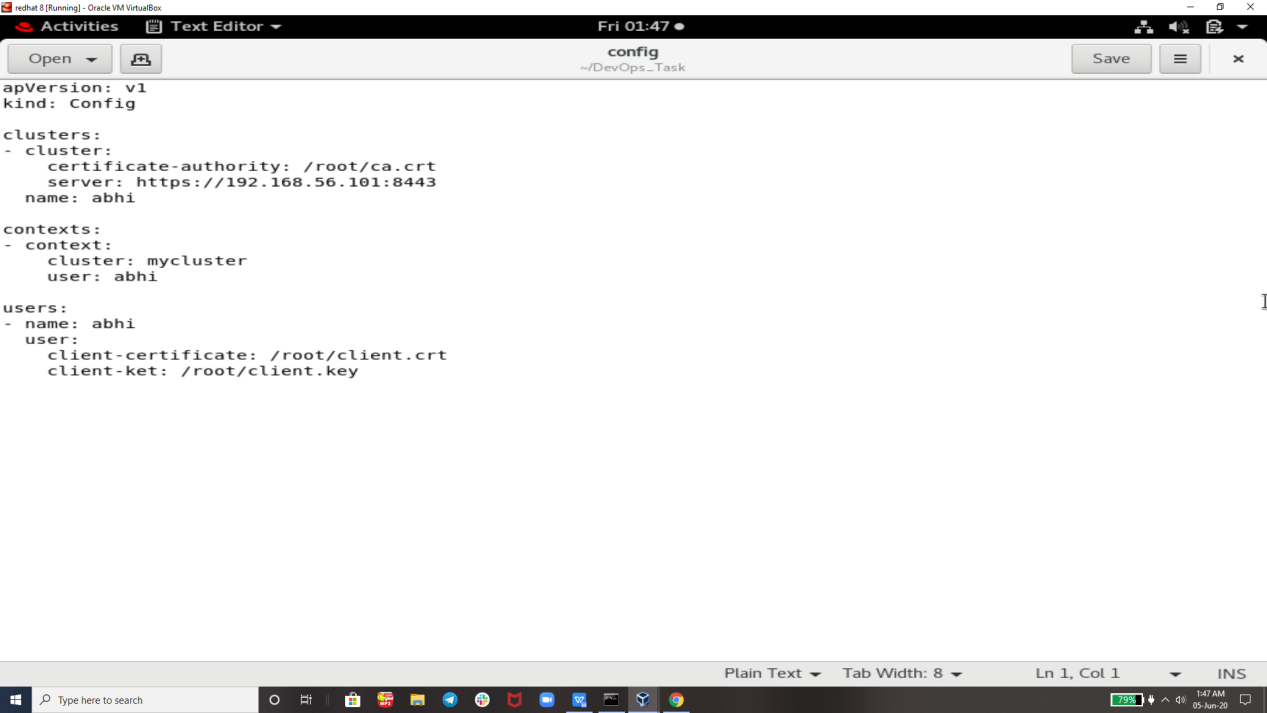
I have Created One Directory /DevOps\_Task3 And inside this Directory I have Placed my Dockerfile and and all needed things.



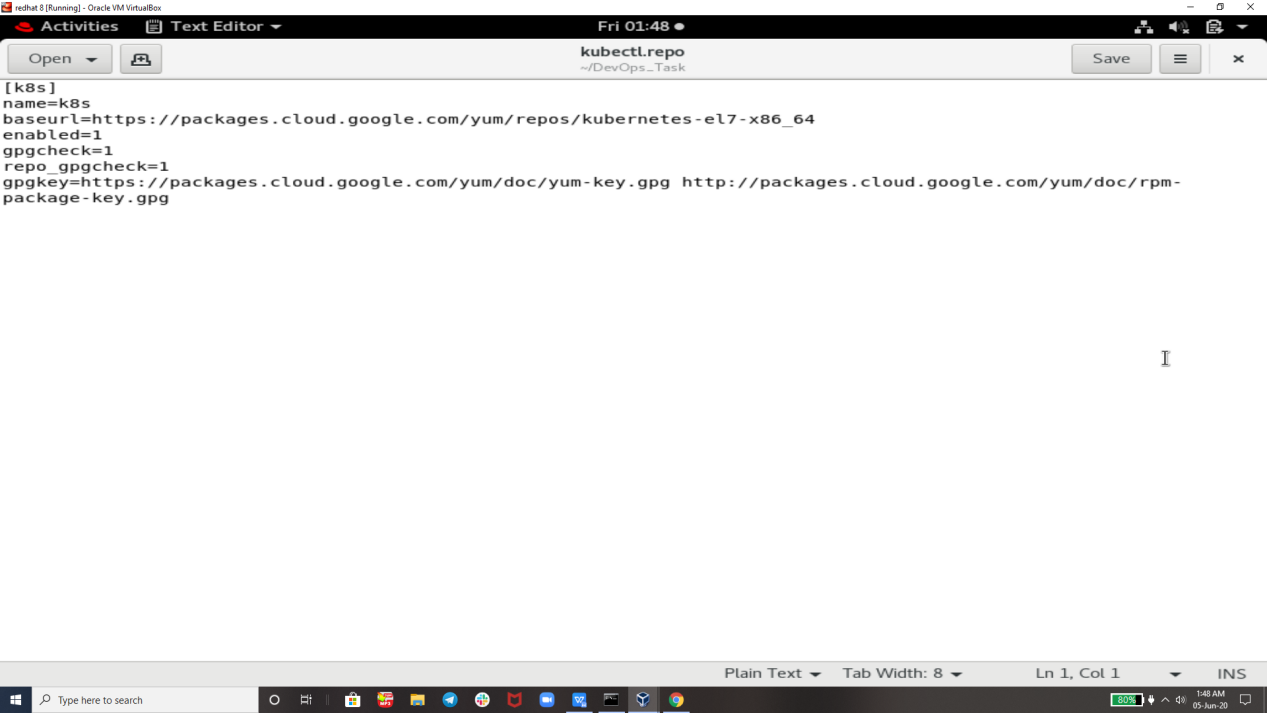
Code of Index.html and SignupPage.html is too big so I m not showing that here u can get that from

my github .

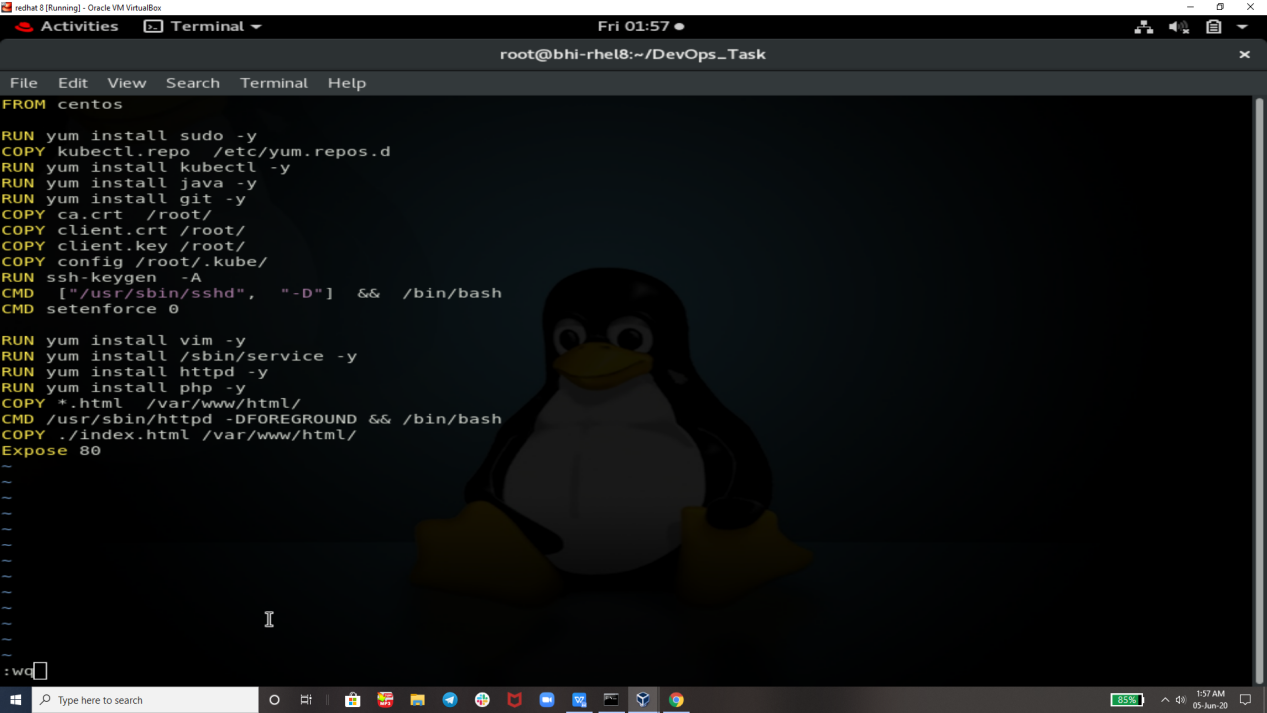
Config File Needed For Kubectl Configuration…



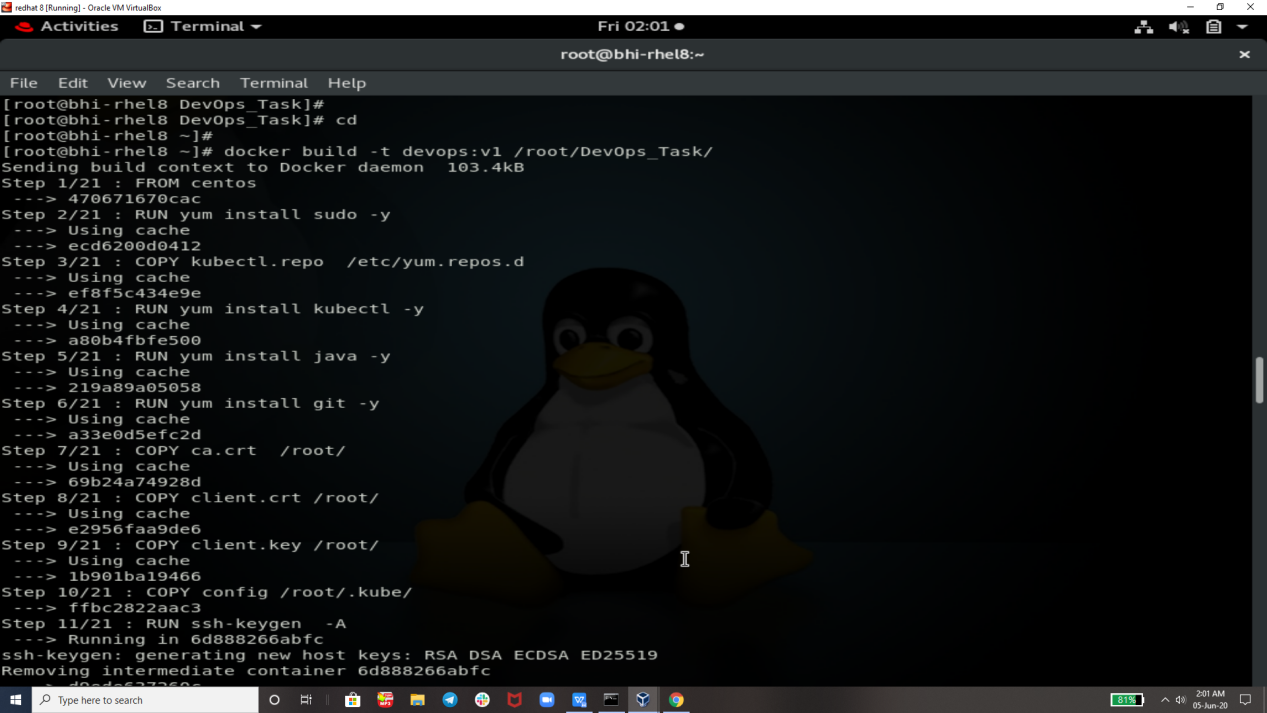
And for installing kubectl we need its repo

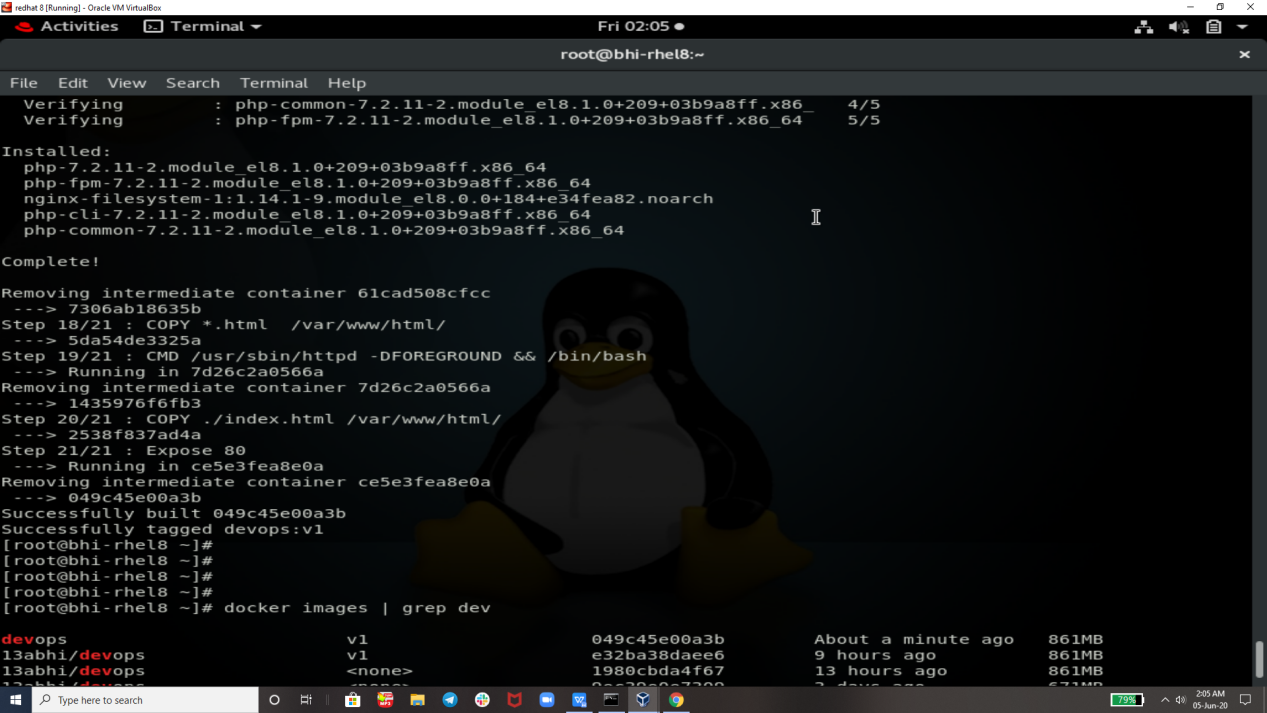


Now Here is Dockerfile :-

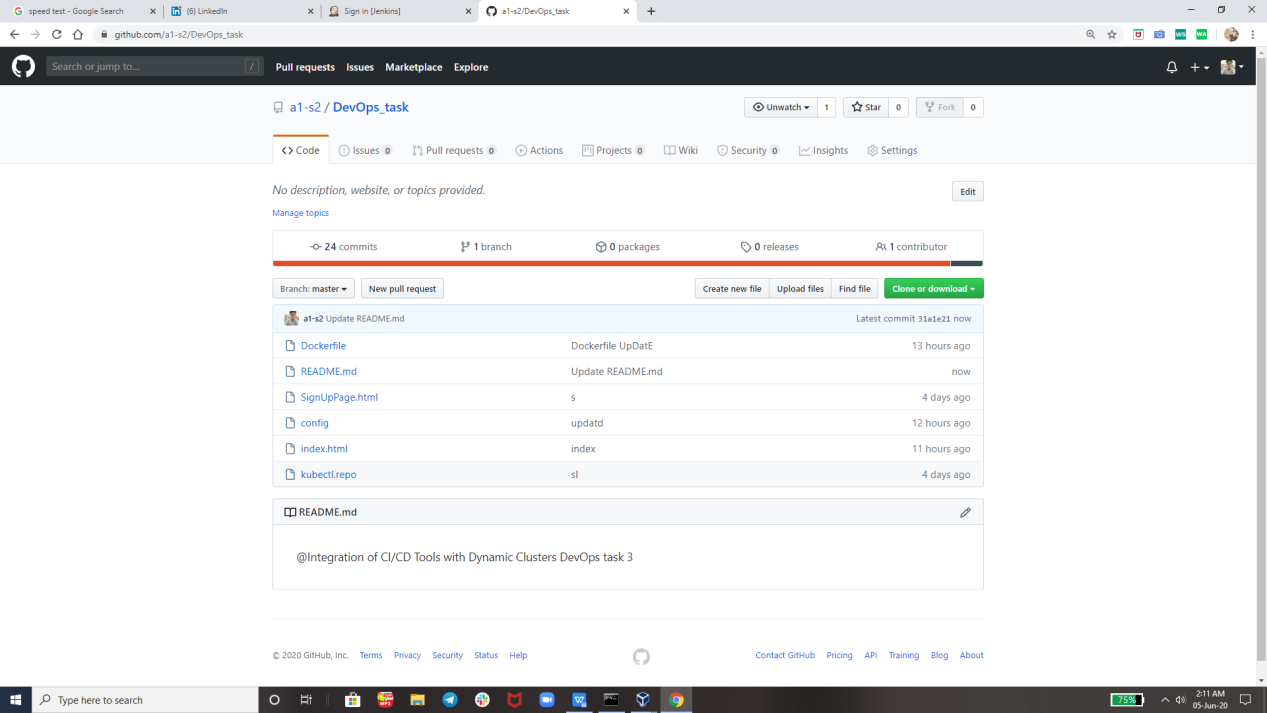


Now we Are Testing Our Image By Building this Dockerfile :-





Here is Code Pushed by Developer From git Bash :-



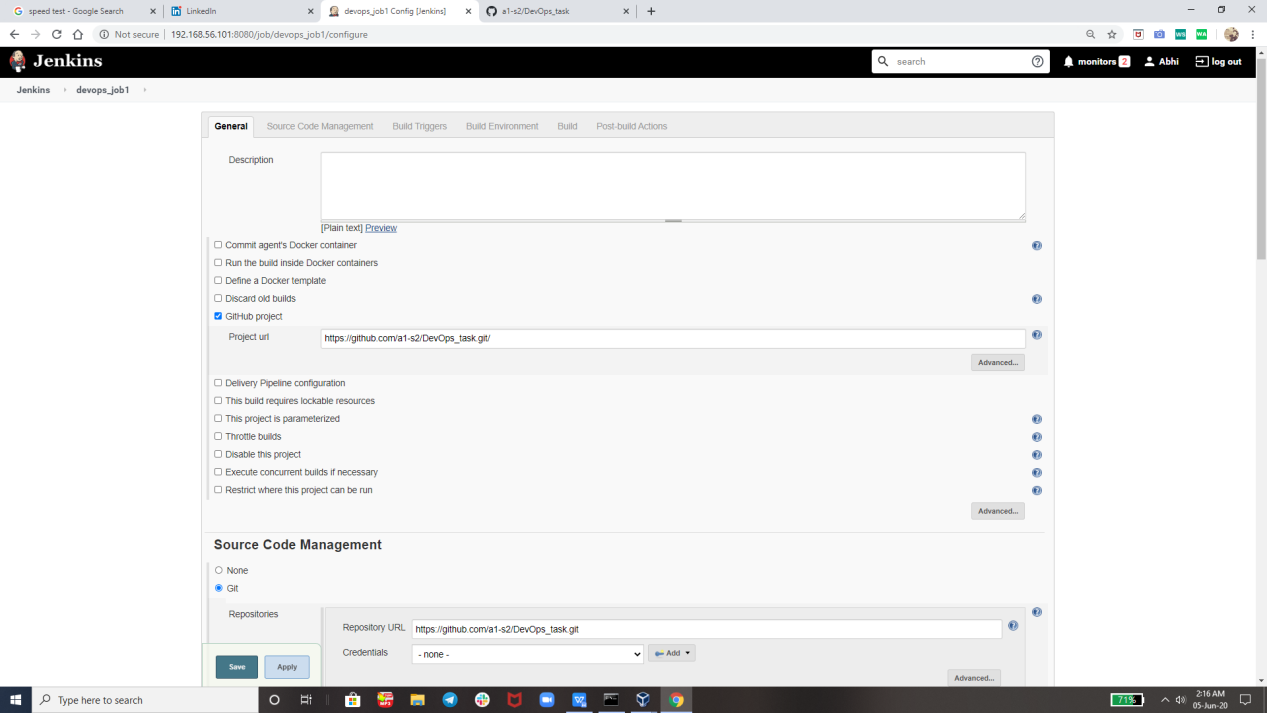
Now Jenkins will Come And Play It’s Role !!

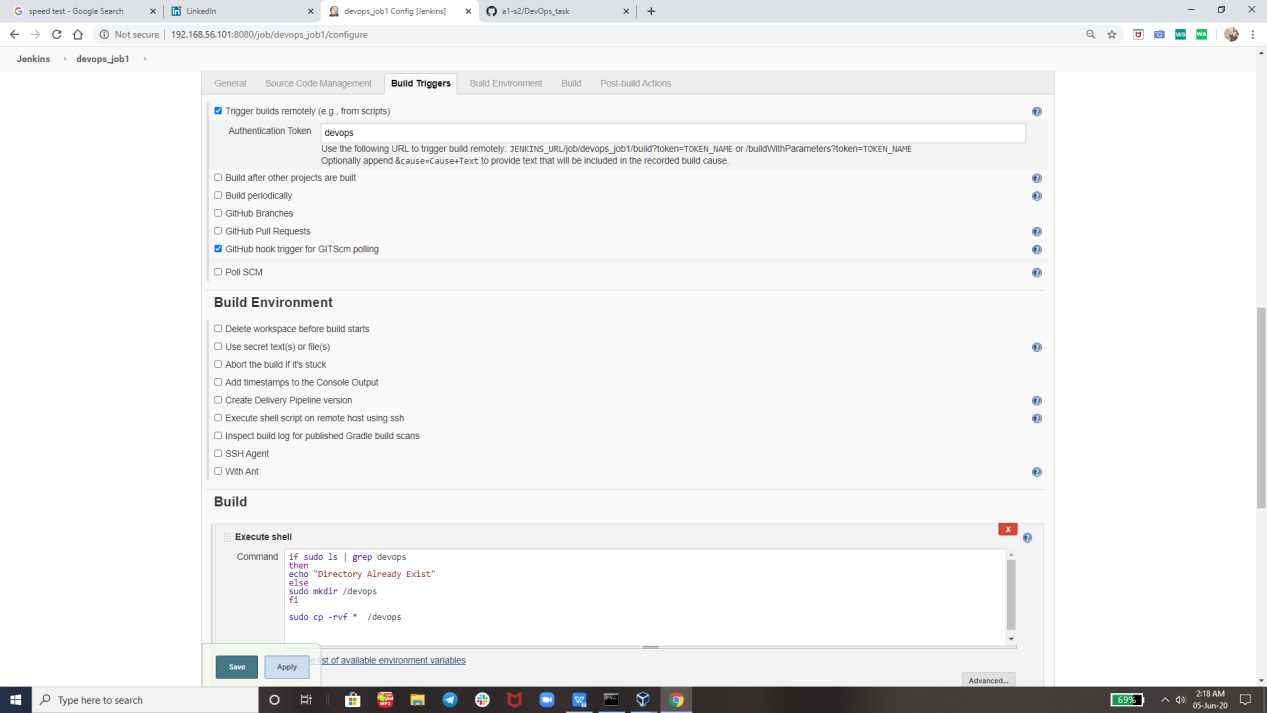
**Job1** : Pull the Github repo automatically when some developers push repo to Github and perform the following operations as:

1 Create the new image dynamically for the application and copy the application code into that corresponding docker image

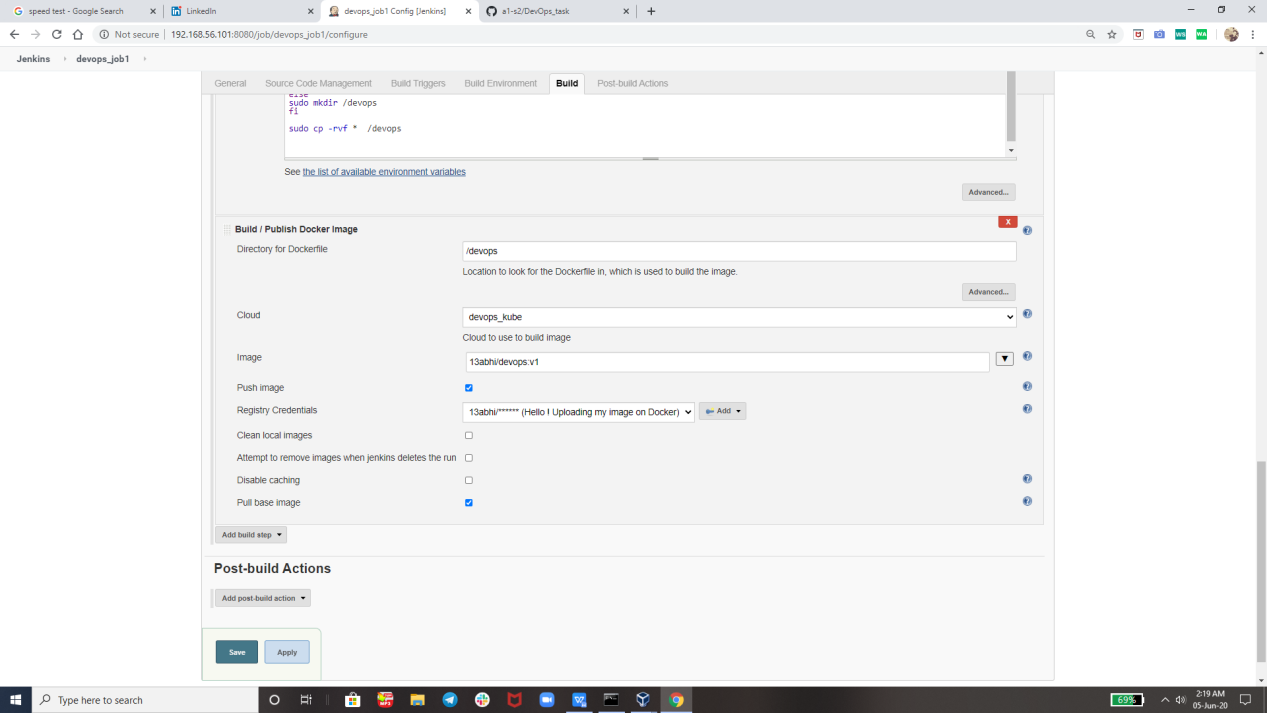
2 Push that image to the docker hub (Public repository)

( Github code contain the application code and Dockerfile to create a new image )

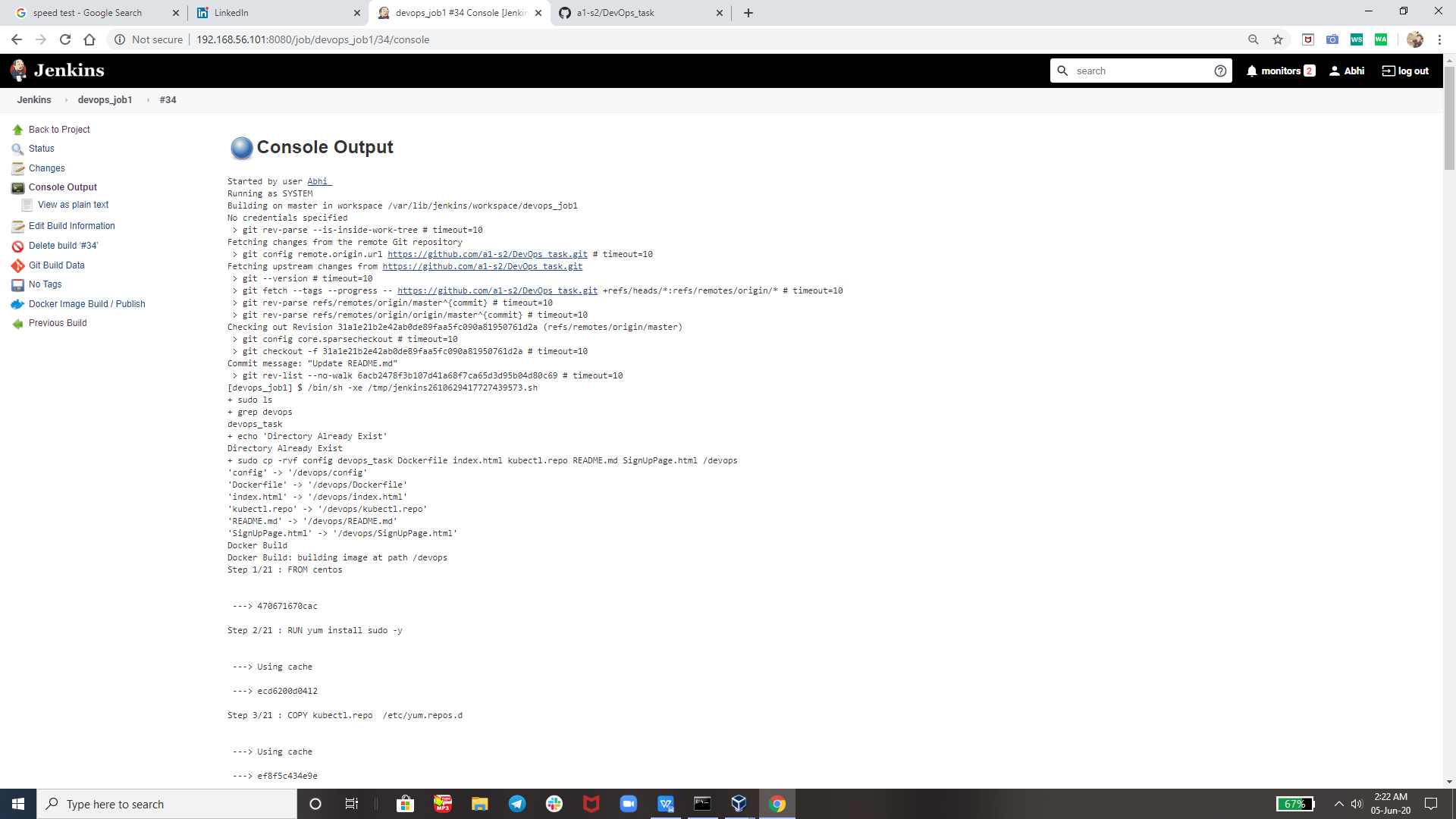


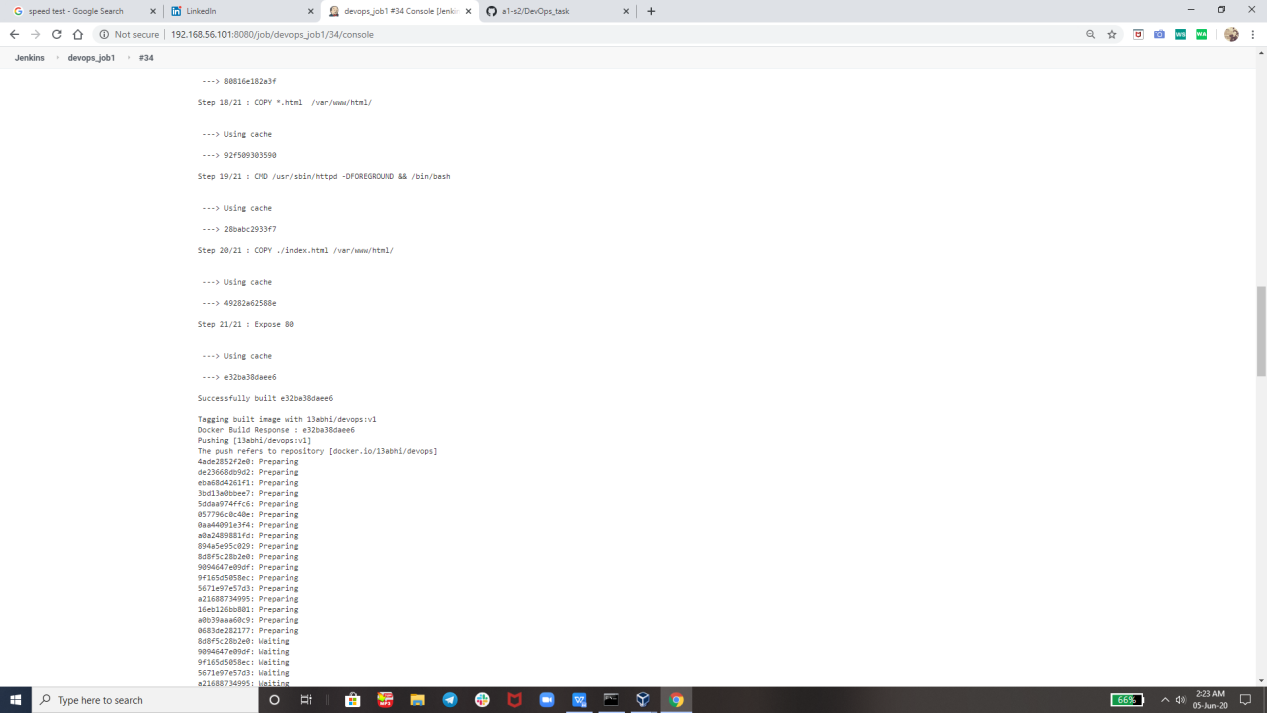


Uploading image on Docker Hub :-

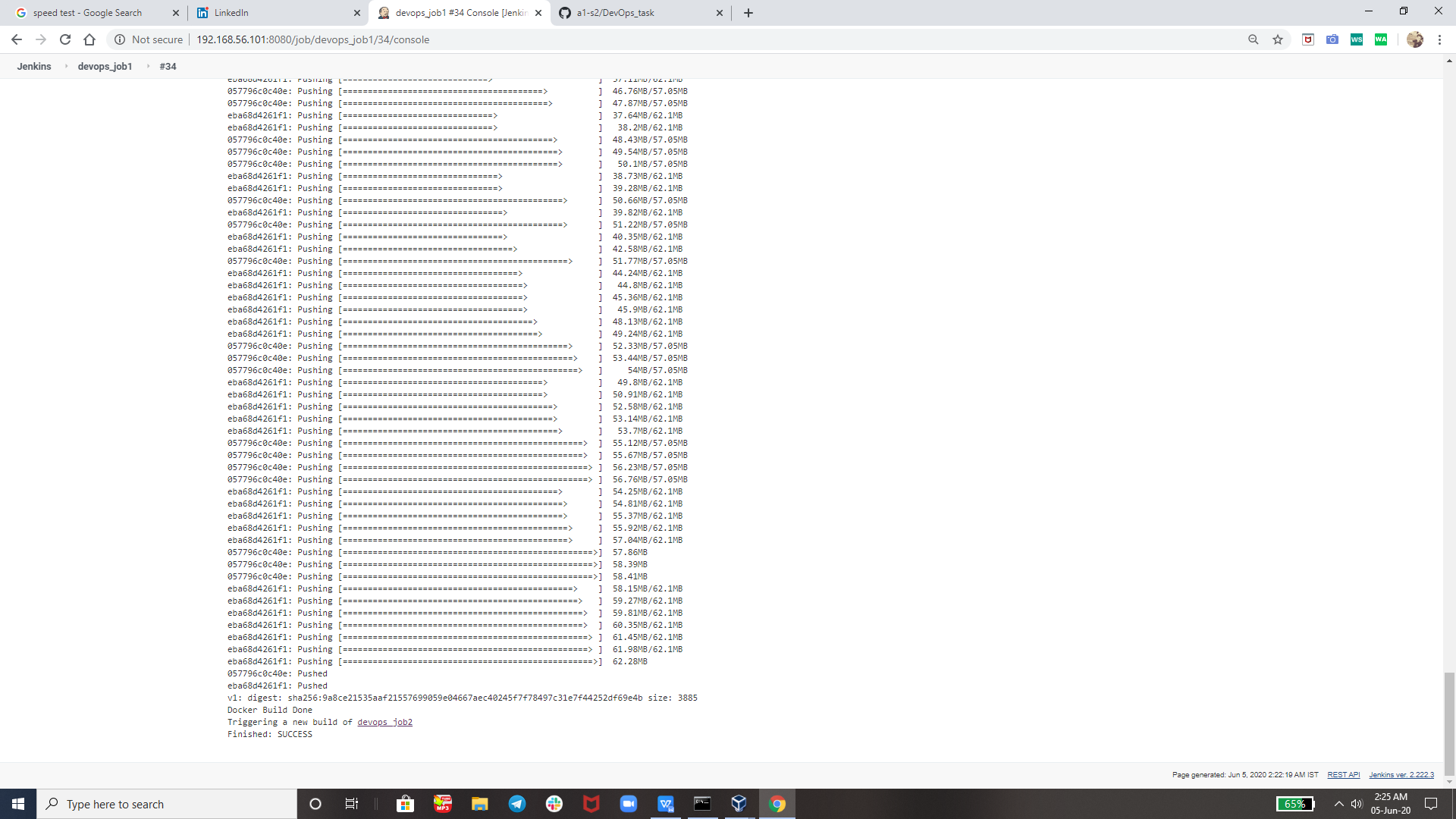


And Here is OutPuT of Job 1:-





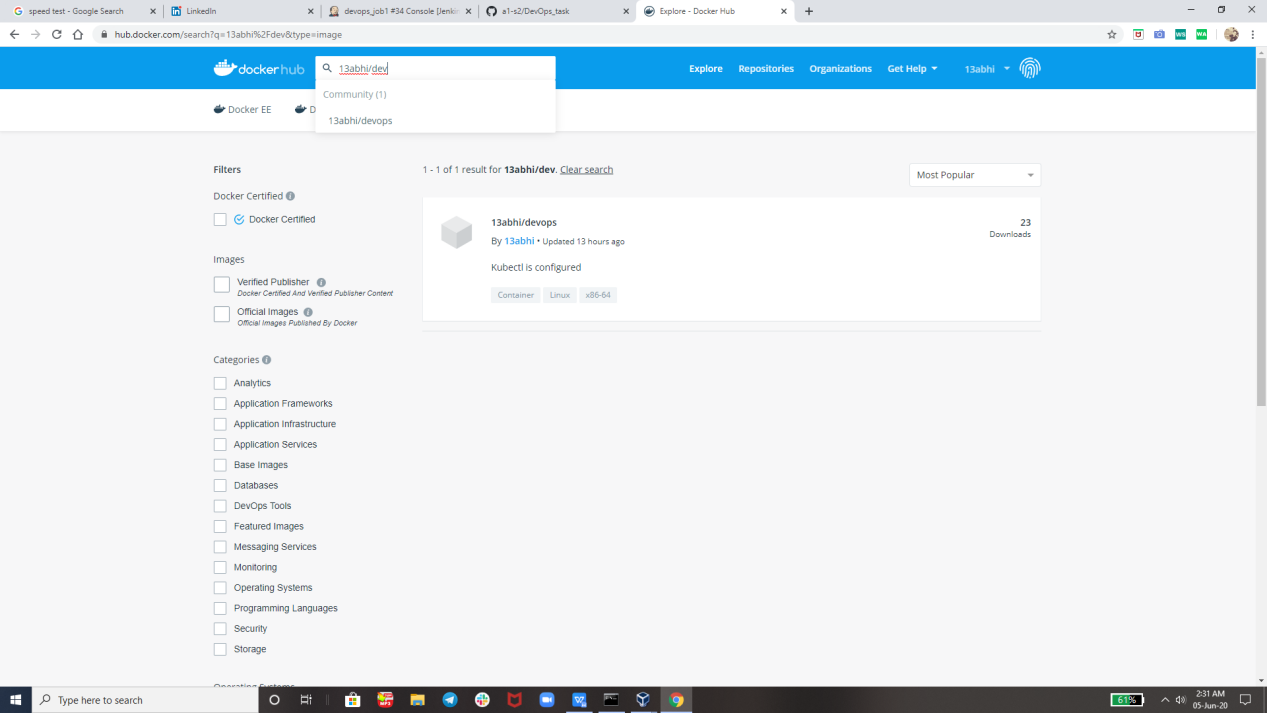




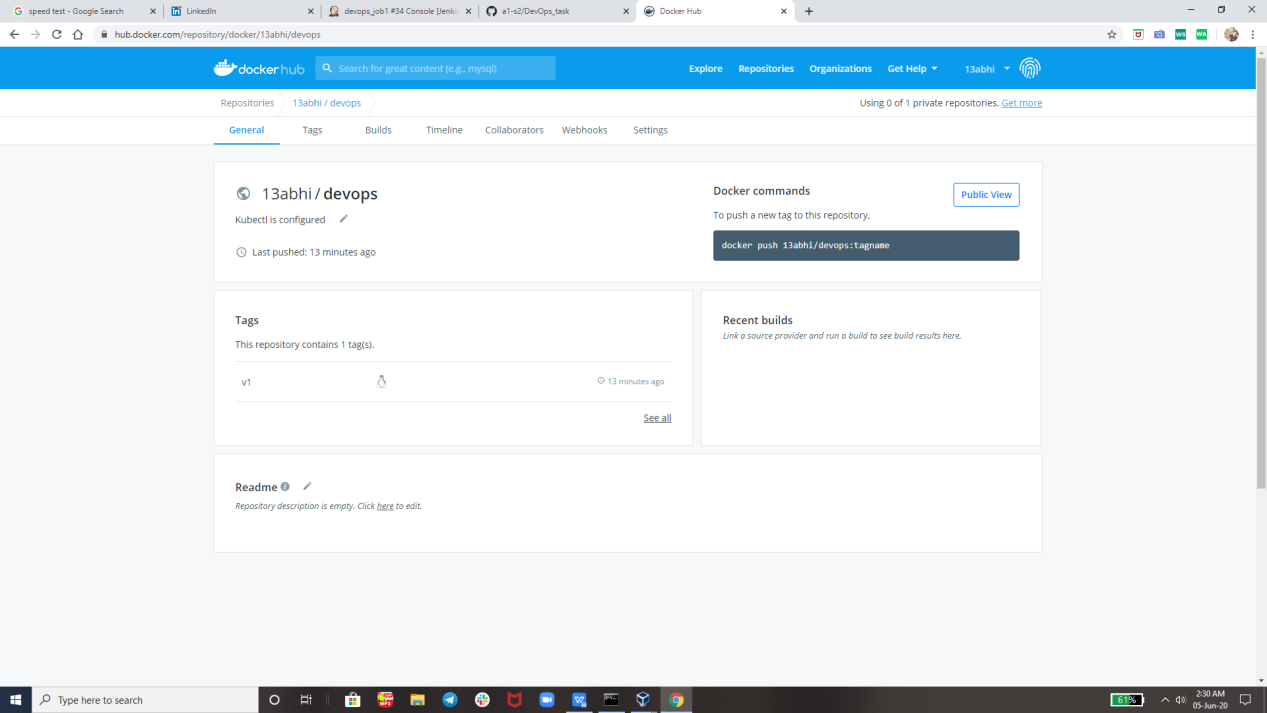
So Now my Image is Succesfully Uploaded tO Docker Hub I.e. hub.docker.com

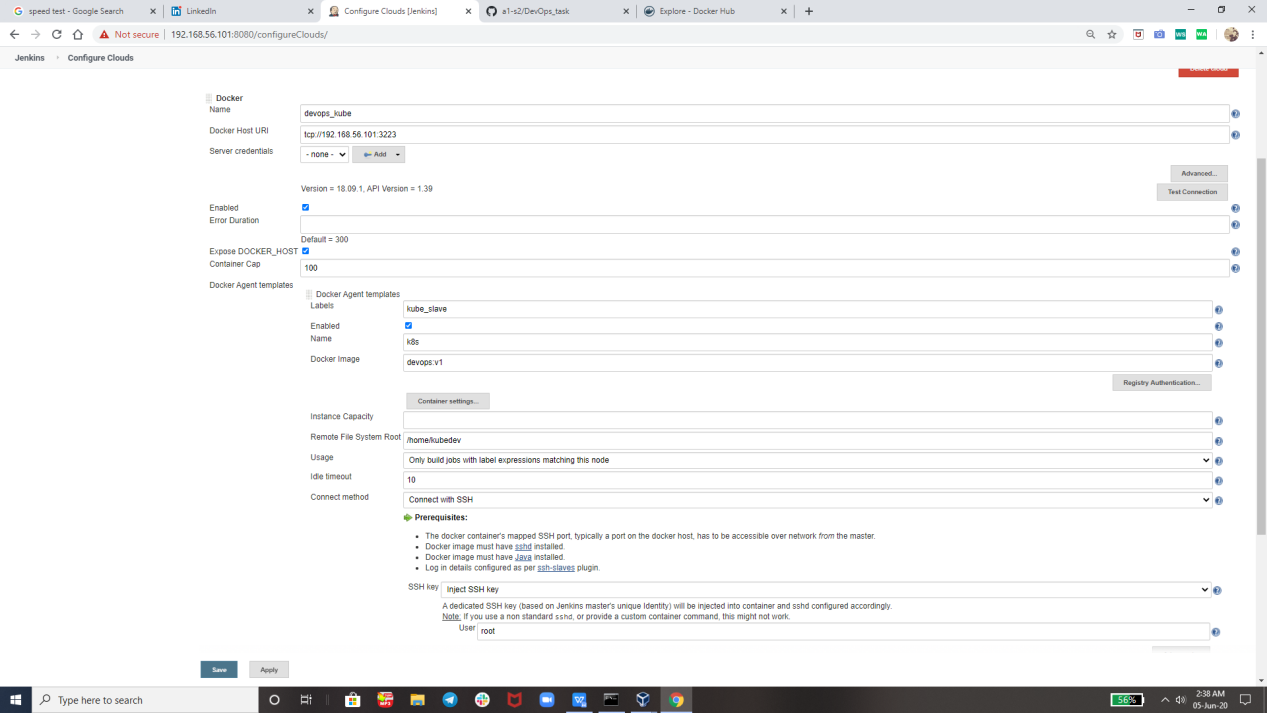
And Now as JOB 1 is Succesfully run so it will trigger JOB 2 Automatically But For Job 2 I have Created Cloud Cluster so I am sharng that too

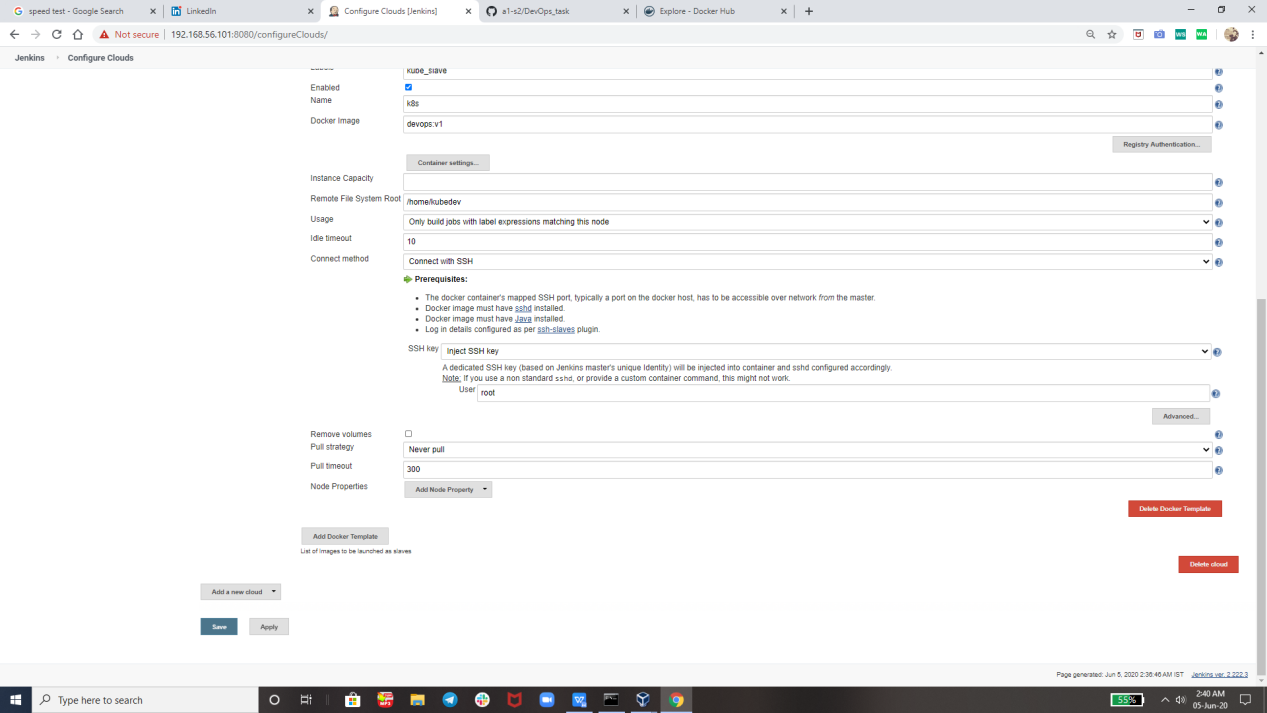
Here is my Image :-



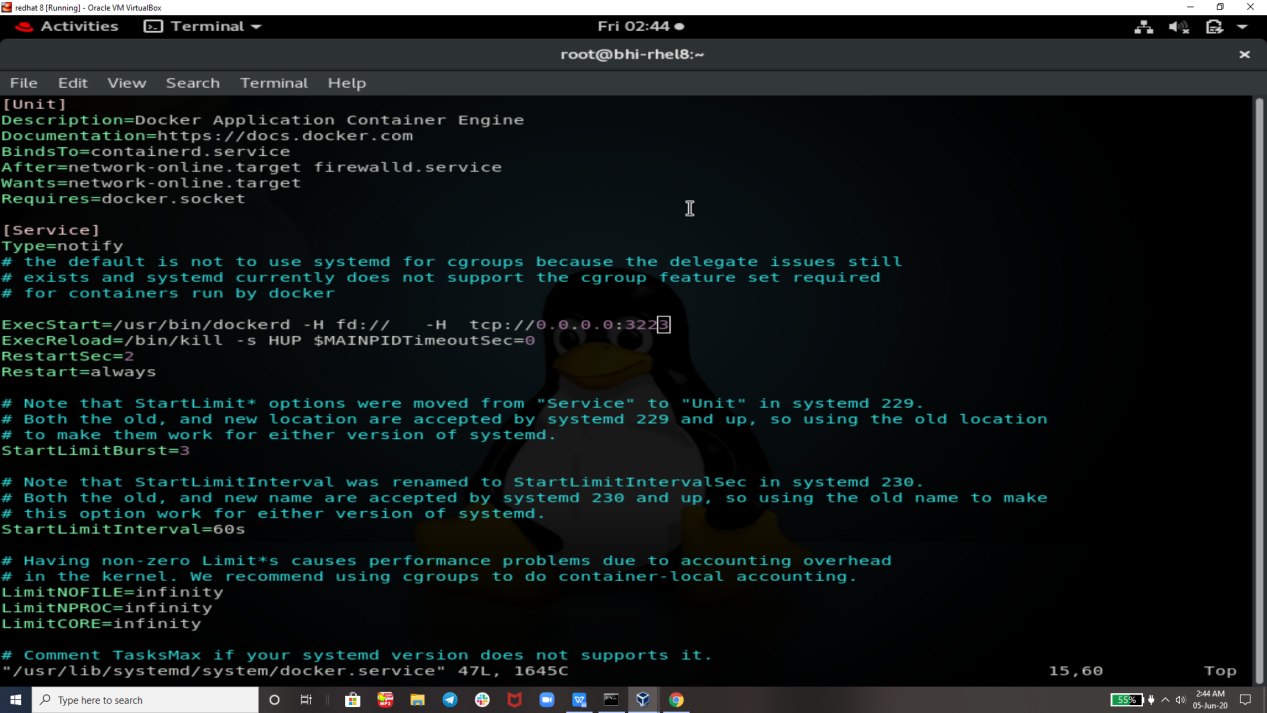
U can also search 13abhi/devops:v1 to get this image in u screen







For making this cluster we will need to export the ip and port from Rhel



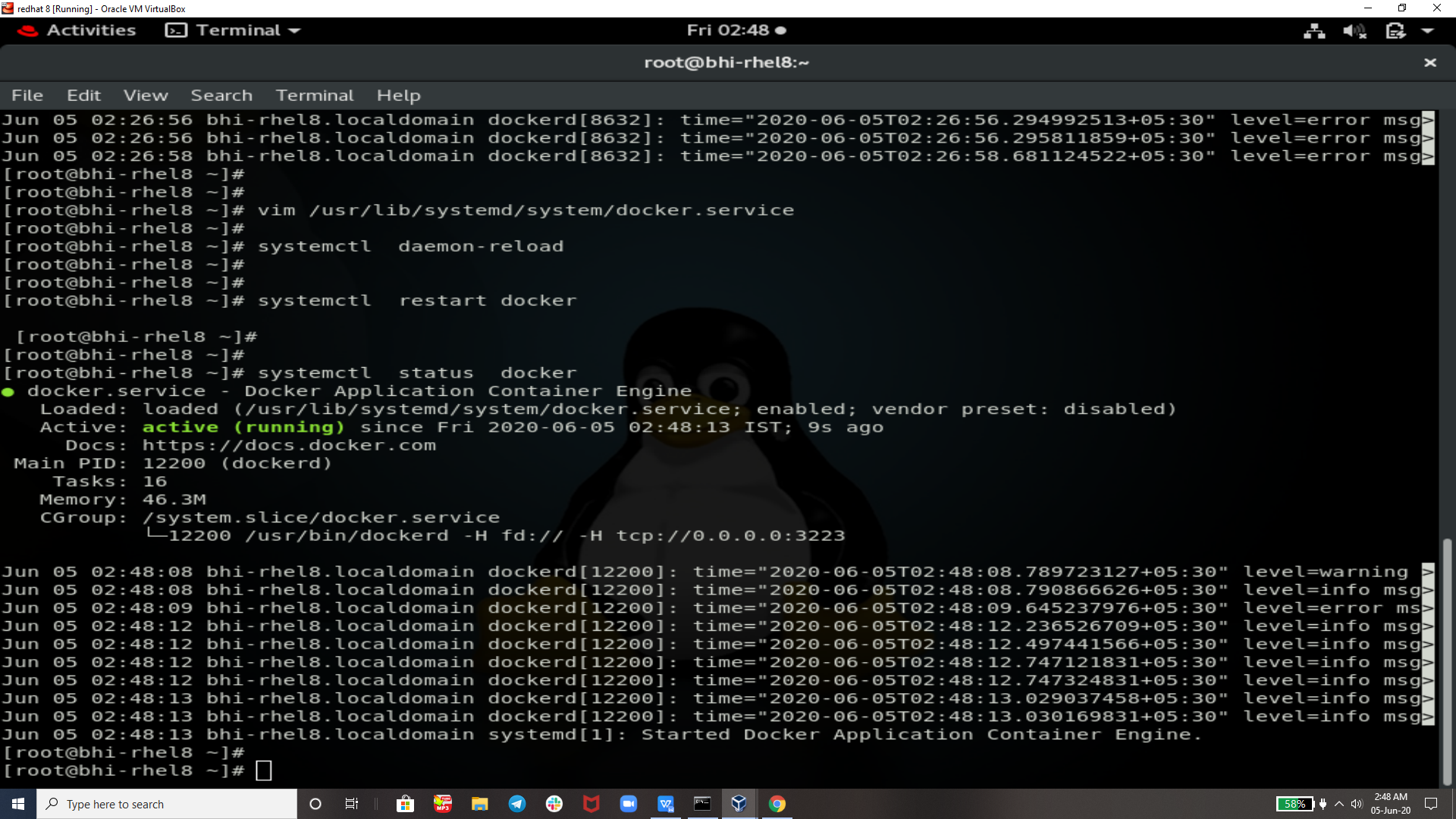
And After Changing in the docker service we need to reload the daemon and restart the docker

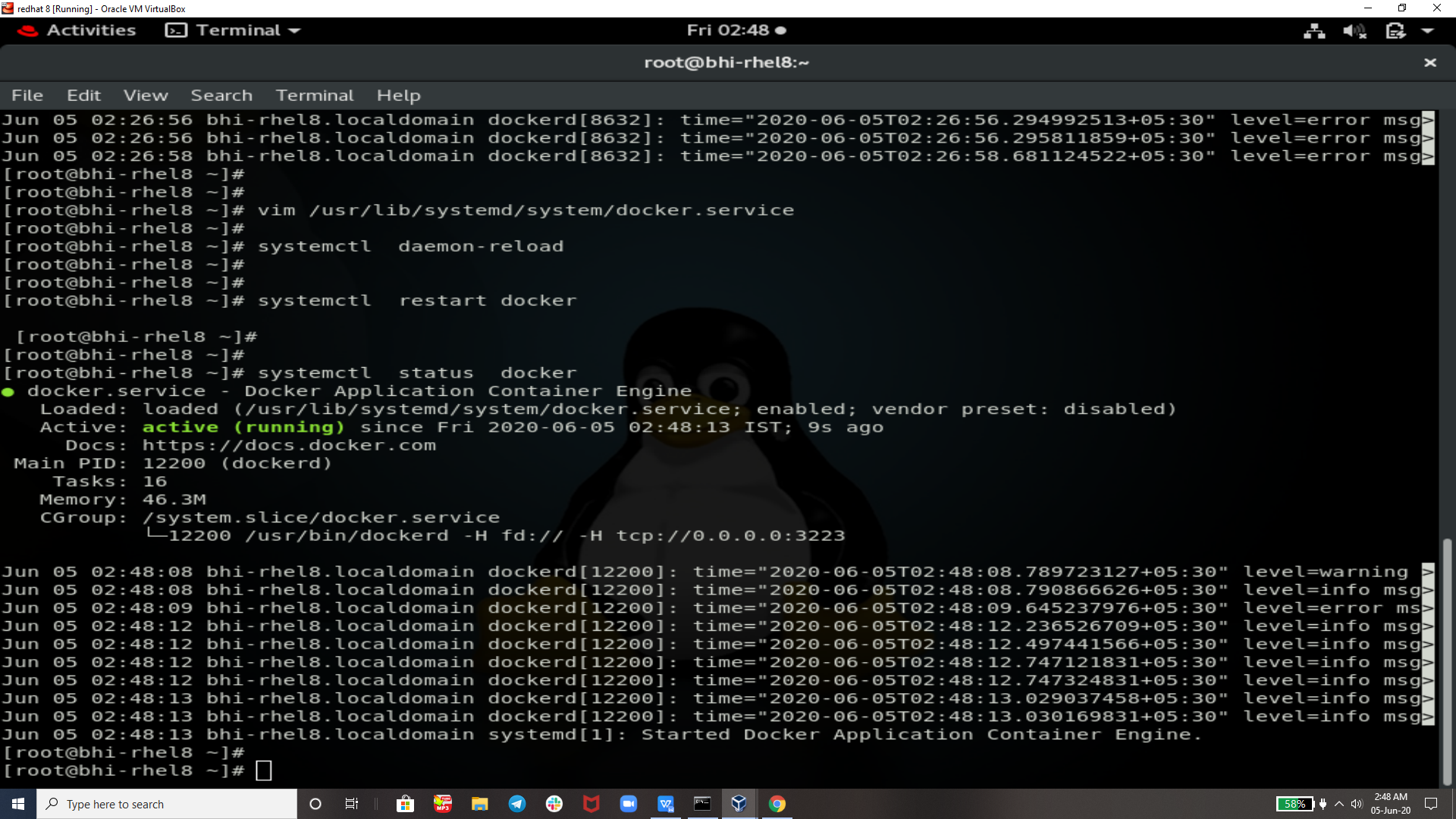
Commands I used to reload daemon

systemctl daemon-reload

Command to restart docker services

Systemctl restart docker



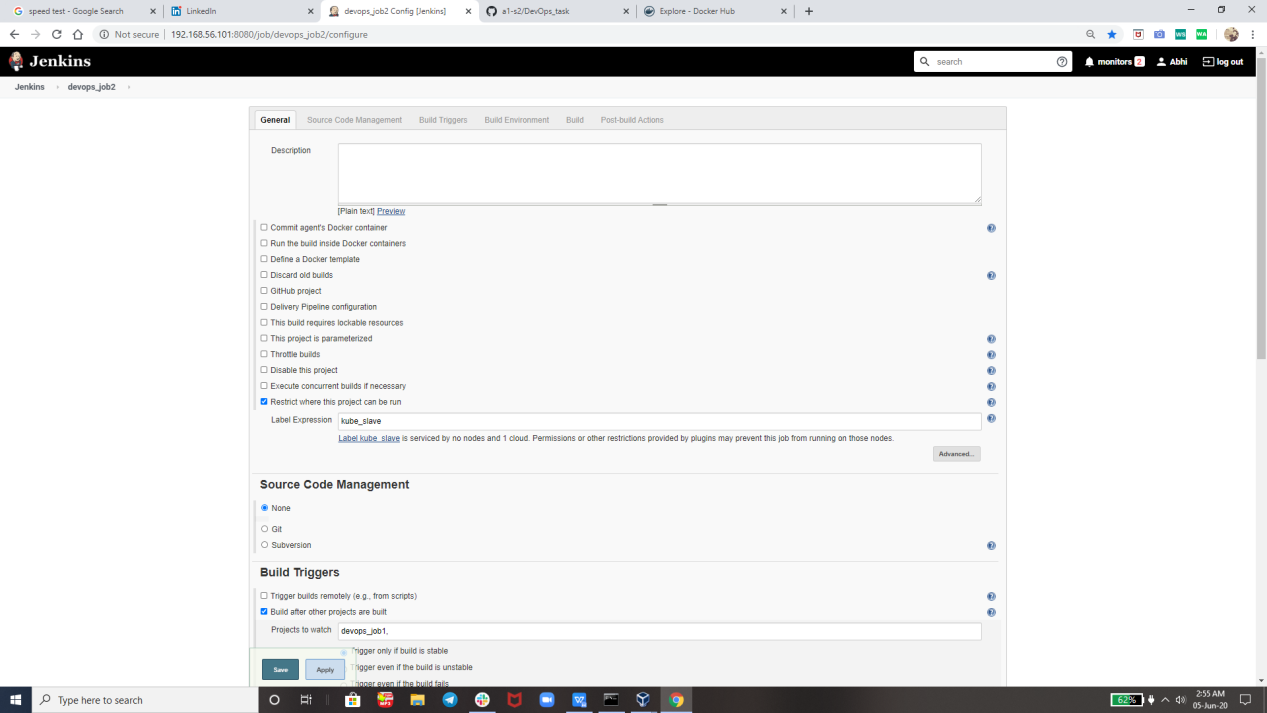


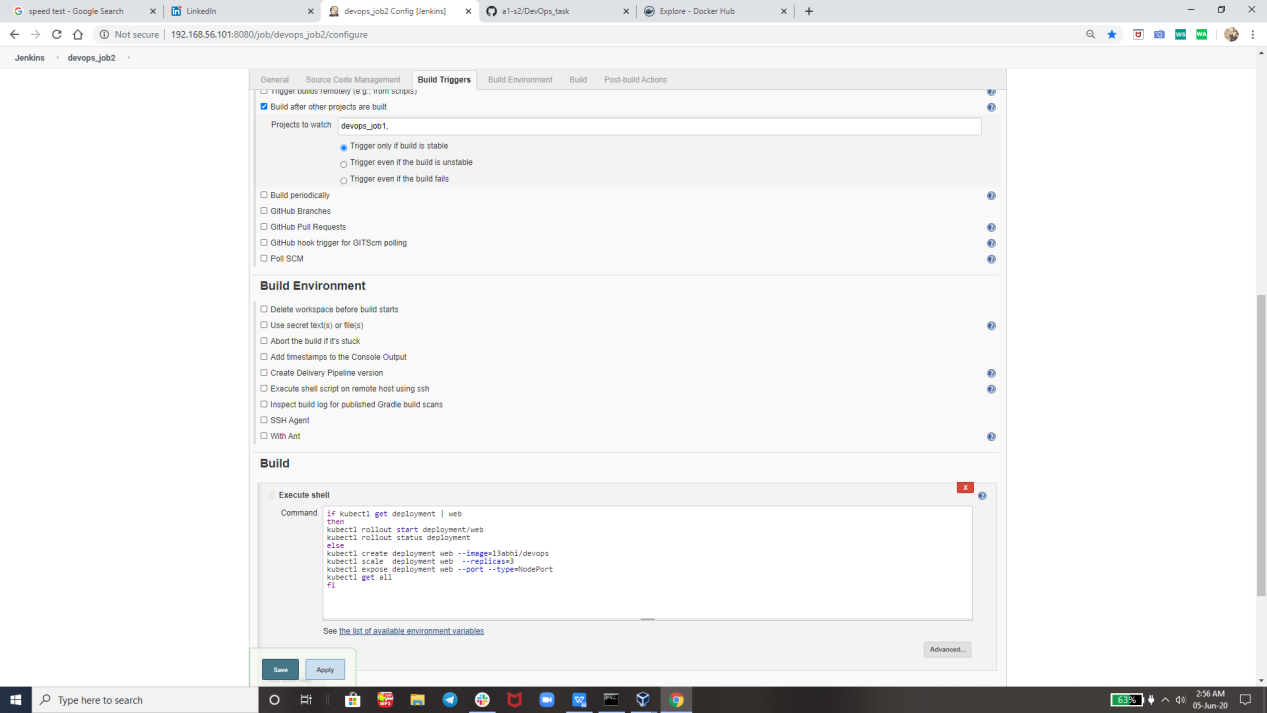
NOW JOB 2:

**Job2:** ( Should be run on the dynamic slave of Jenkins configured with Kubernetes kubectl command): Launch the application on the top of Kubernetes cluster performing following operations:

1 If launching first time then create a deployment of the pod using the image created in the previous job. Else if deployment already exists then do rollout of the existing pod making zero downtime for the user.

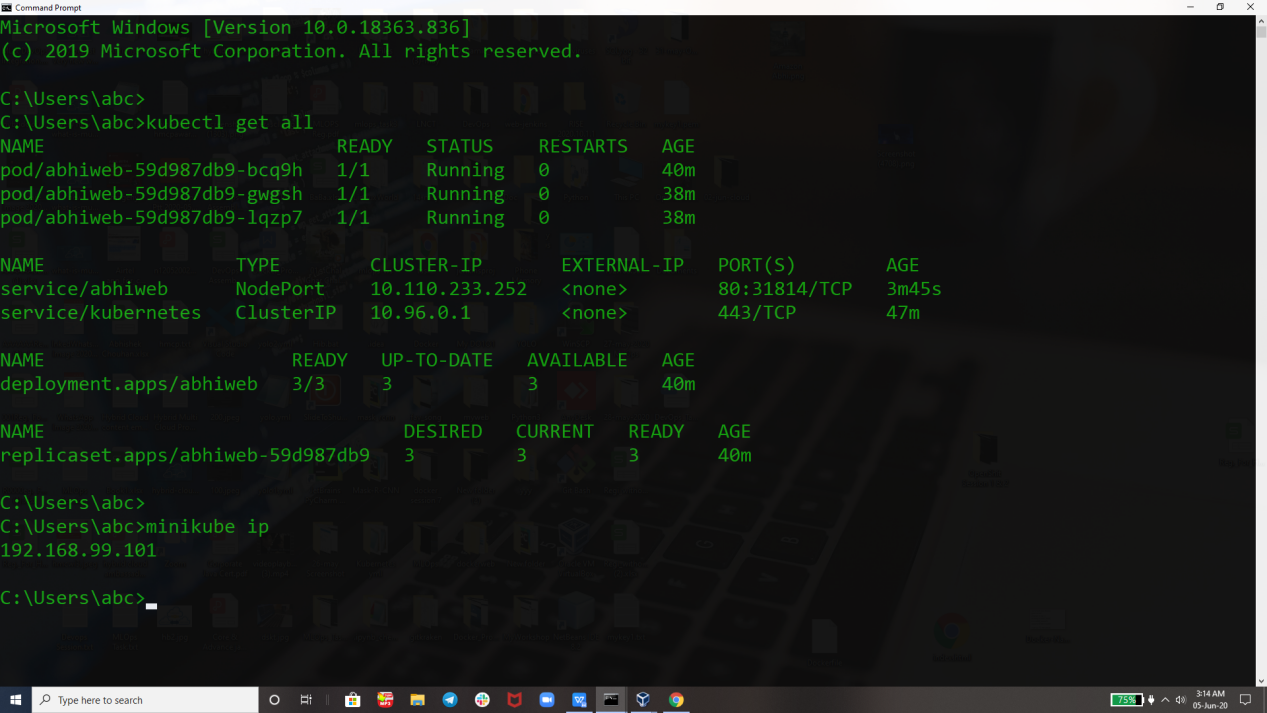
2 If Application created first time, then Expose the application. Else don’t expose it



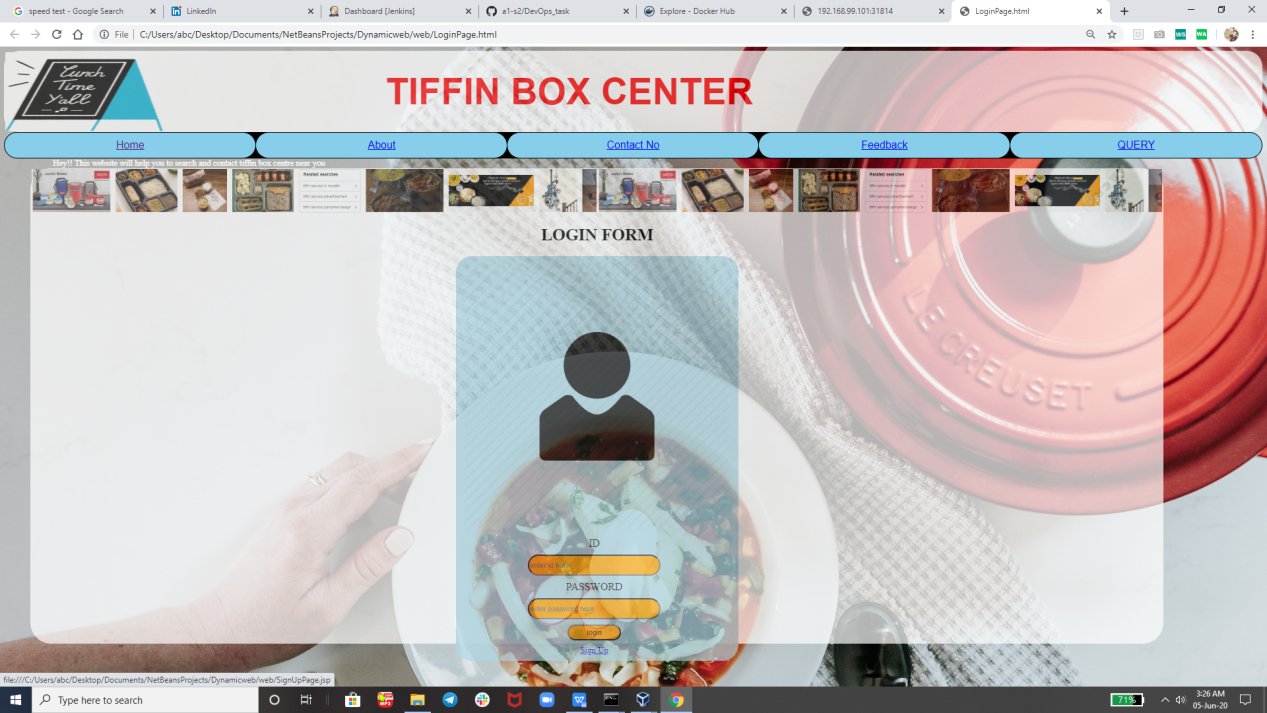


Here we Go So Finally Both the jobs are succesfully run & here is build pipelinne view :-

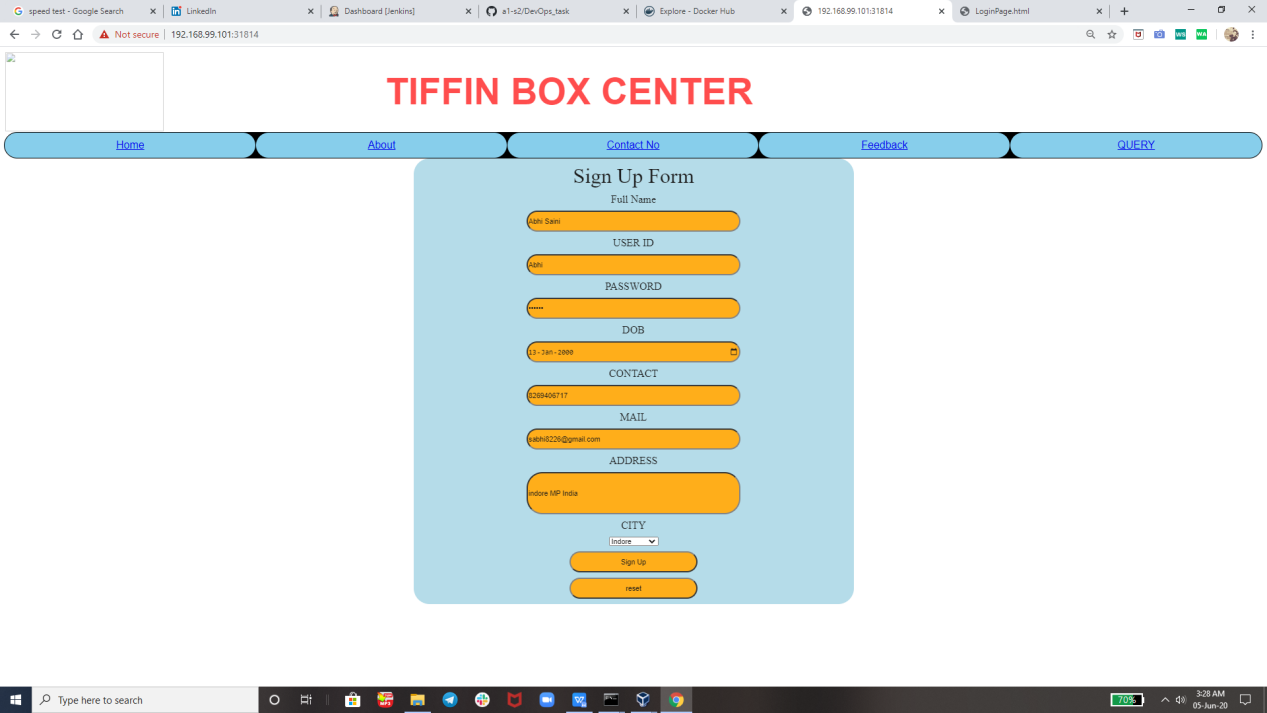
Now we are checking that on which port its working so that we can see our html pages



Previously I Have Set This to y Index Page but now I have UpdaTed The Docker file with new webpage



So This is UpDaTed WebPage And Believe me Guy’s I Have Spent my much Time tO CreaTe This WebPage



**Thank you vimal sir for teaching us such an great tools technology like RHEL 8, Docker, git , Kubernetes , Jenkins and many more. So that today i am capable of creating such an great ProJecT.**

**Thank you EveryOne For Reading !**