Reference: https://medium.com/swlh/ci-cd-using-docker-jenkins-and-git-d034c1d5e371

In today’s ever-evolving landscape, one of the top challenges for software companies is quickly responding to market and customer demands. And CI/CD methodology emerged as the pivotal solution to such a challenge. Read this article to find out more about CI/CD and how these practices are performed.

**Continuous integration (CI)** helps ensure that software components work together. Integration should be completed frequently; if possible, on an hourly or daily basis.

**Continuous delivery (CD)** picks up where continuous integration ends. While CI is the process to build and test automatically, CD deploys all code changes in a build to the testing or staging environment.

In this article, I have some task to develop CI/CD pipeline-

*1. Create a Docker file that will install Jenkins.*

*2. As soon we run this docker image (docker file) it should install Jenkins.*

*3. With the help of Job chaining, Now we have to create jobs in Jenkins.*

*4. Job1: Pull the Github repo automatically when some developers push the repo to Github.*

*5. Job2: By looking at the code or program file, Jenkins should automatically start the respective language interpreter install image container to deploy code ( eg. If code is of PHP, then Jenkins should start the container that has PHP already installed ).*

*6. Job3: Test your webpage if it is working or not.*

*7. Job4: if the app is not working, then send an email to the developer with error messages.*

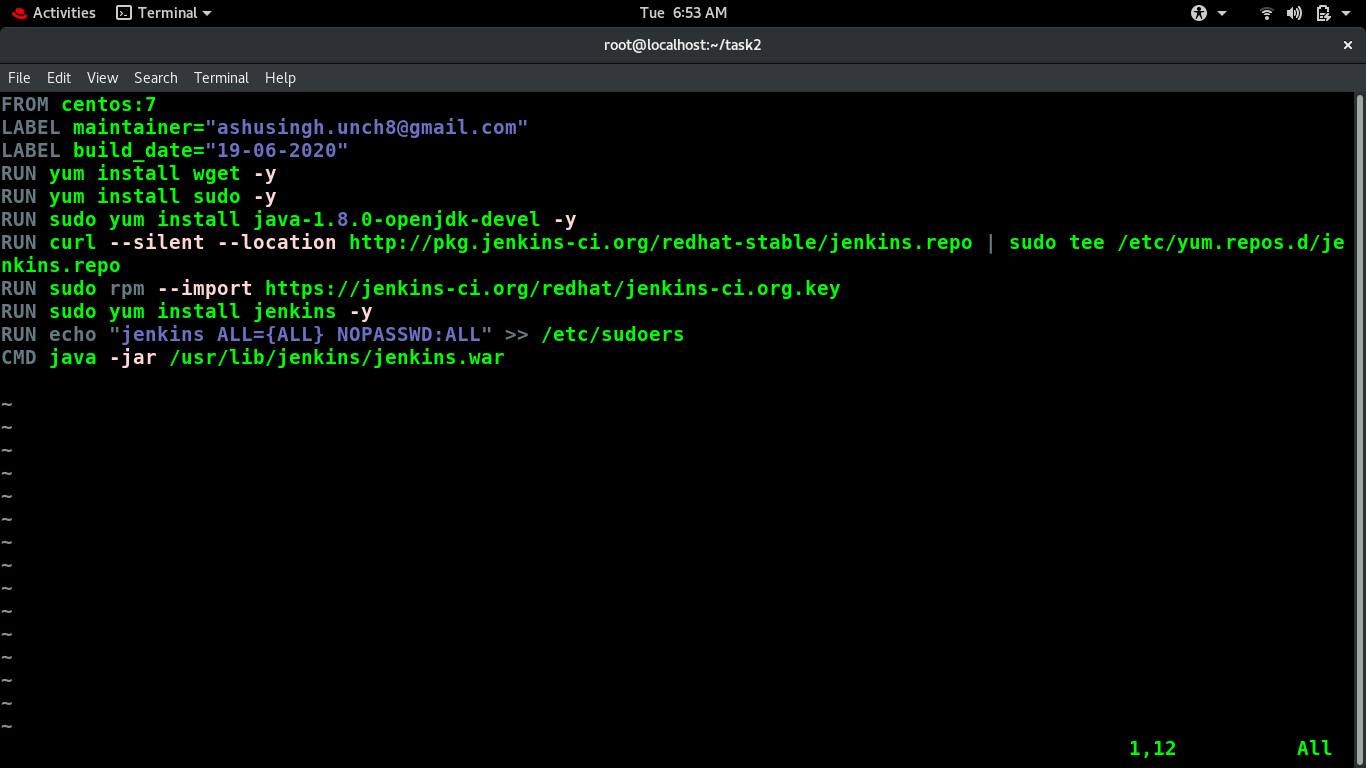
*8. job5 for monitor: If the container where the app is running. fails due to any reason then this job should automatically start the container again.*

Prerequisites for this project

* Must have installed Docker and Jenkins inside RHEL8
* Must have installed Git Bash inside os
* Must have Github account

Create Dockerfile in the workspace.

Image for post



**Dockerfile**

FROM centos:7  
LABEL maintainer="[ashusingh.unch8@gmail.com](mailto:ashusingh.unch8@gmail.com)"  
LABEL build\_date="19-06-2020"  
RUN yum install wget -y  
RUN yum install sudo -y  
RUN sudo yum install java-1.8.0-openjdk-devel -y  
RUN curl --silent --location <http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo> | sudo tee /etc/yum.repos.d/jenkins.repo  
RUN sudo rpm --import <https://jenkins-ci.org/redhat/jenkins-ci.org.key>  
RUN sudo yum install jenkins -y  
RUN echo "jenkins ALL={ALL} NOPASSWD:ALL" >> /etc/sudoers  
CMD java -jar /usr/lib/jenkins/jenkins.war

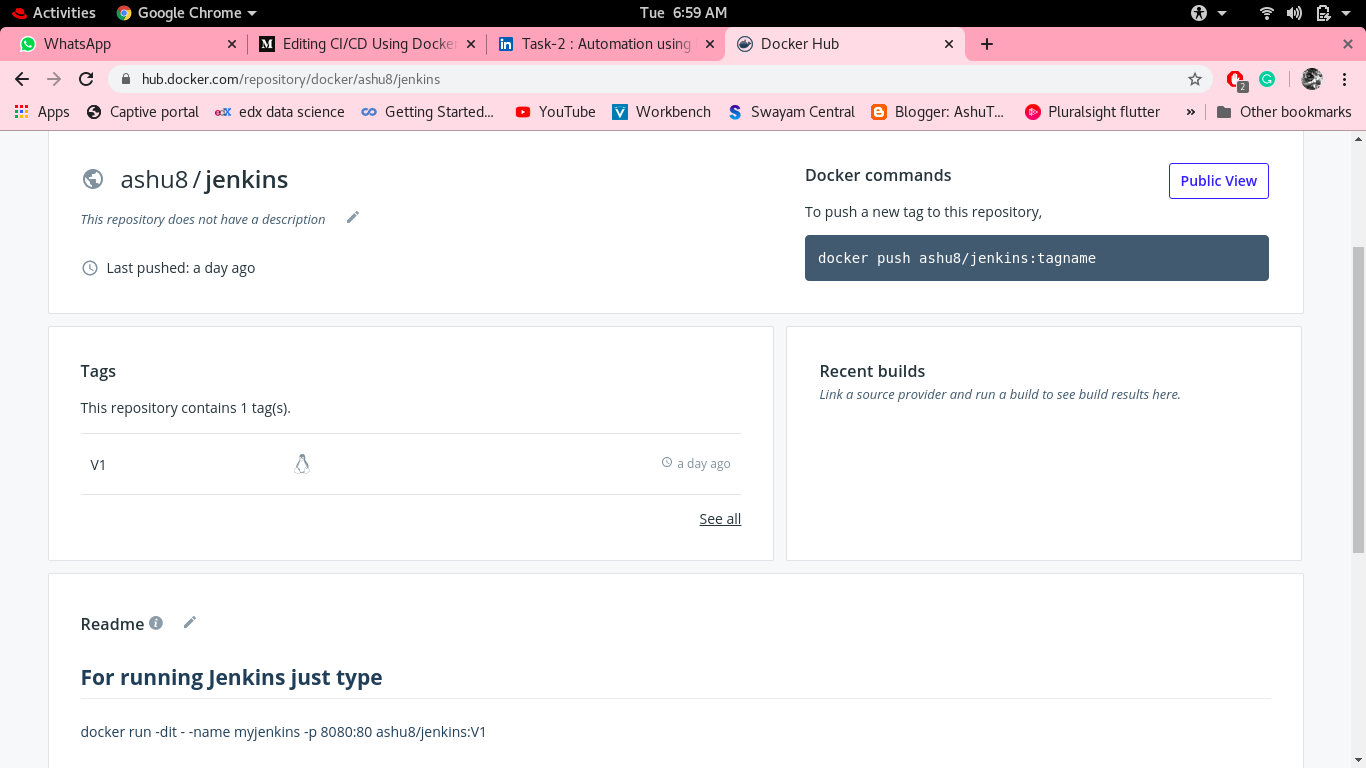
**Building Docker file**

docker build -t jenkins:V1 /ws/

I uploaded the docker image in the docker hub if you want to download this image from the docker hub run following command.

docker push ashu8/jenkins:V1

Image for post

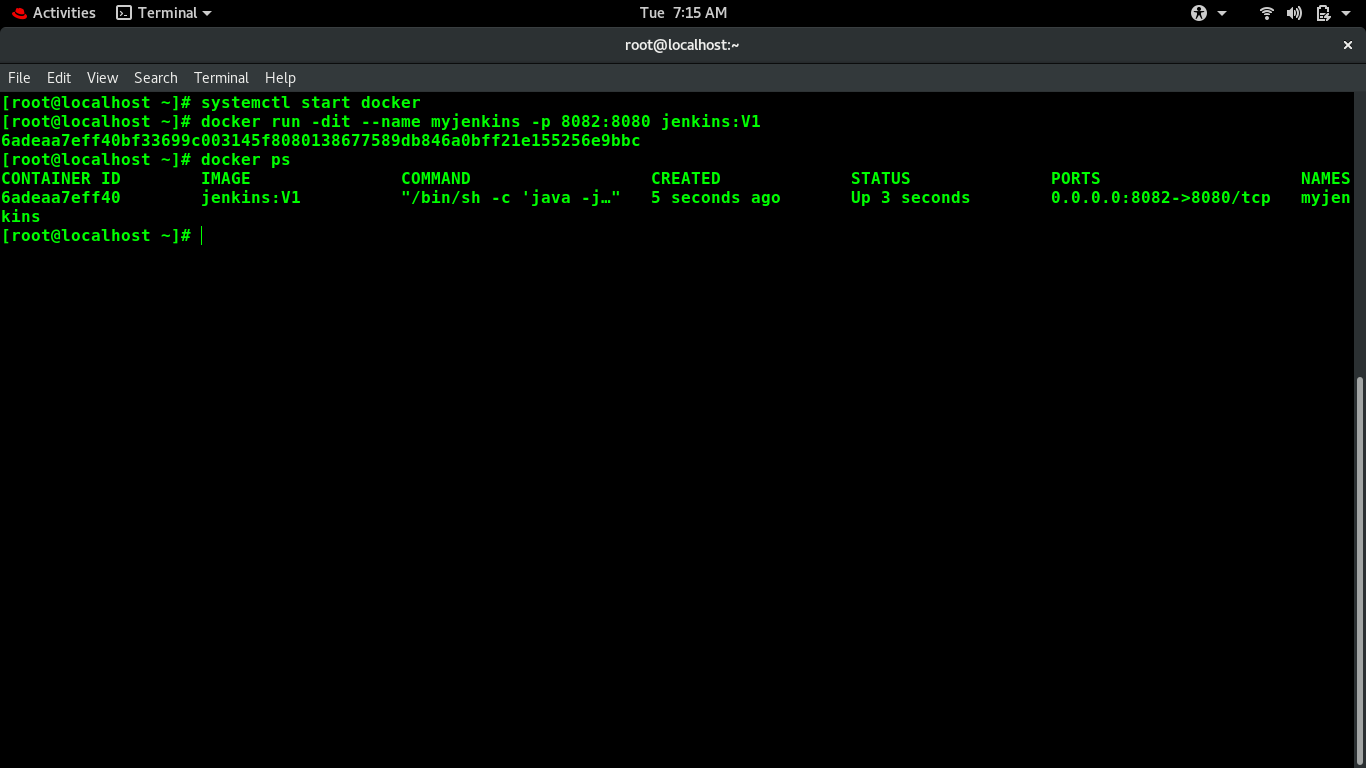


My docker image repo

Now we have to run this image run following command

1.systemctl start docker  
2.docker run -dit --name myjenkins -p 8082:8080 jenkins:V1  
3.docker ps

Image for post



**Setting UP Jenkins**

Here as we have done patting and it will expose 8080 port through port number 8082.

Now our Jenkins is ready to use.

Image for post

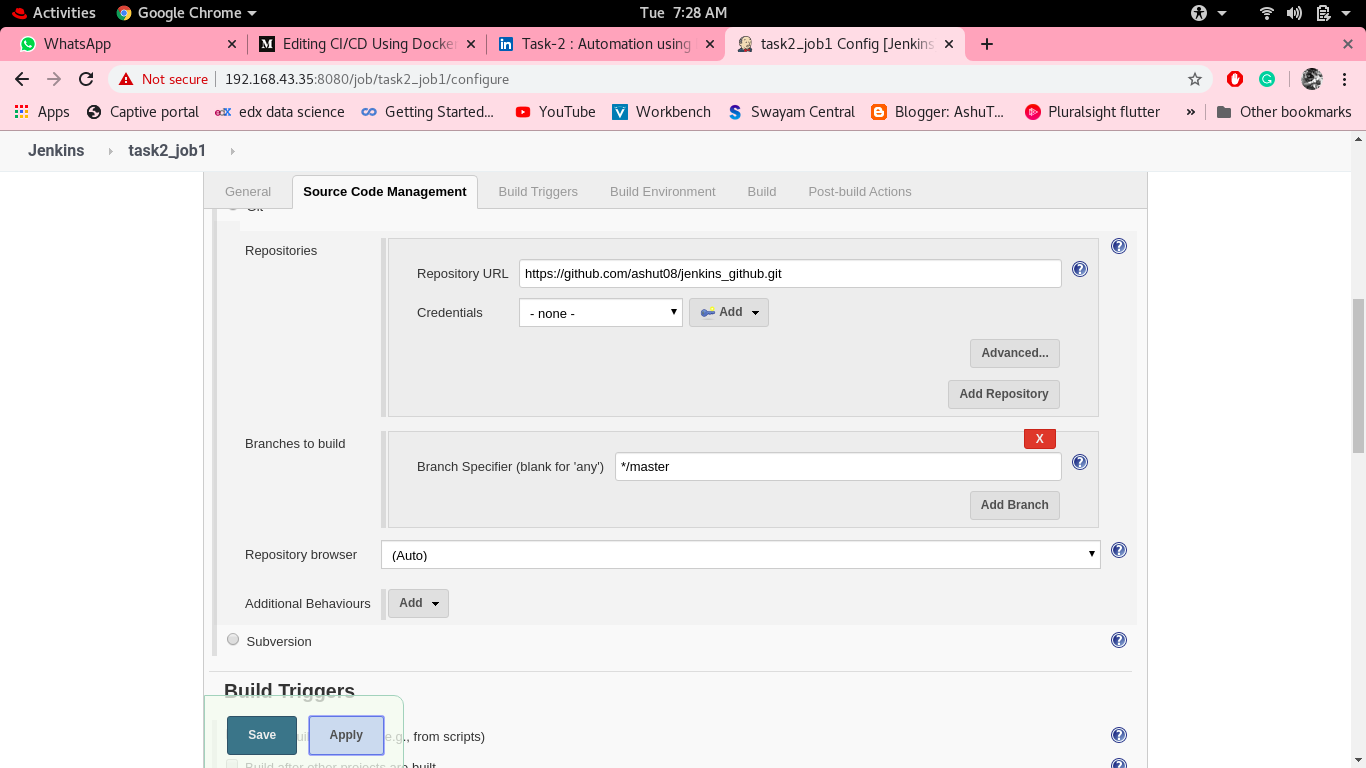


Finally, we have done with docker and run Jenkins in a docker container.

Now we have to perform some jobs in Jenkins

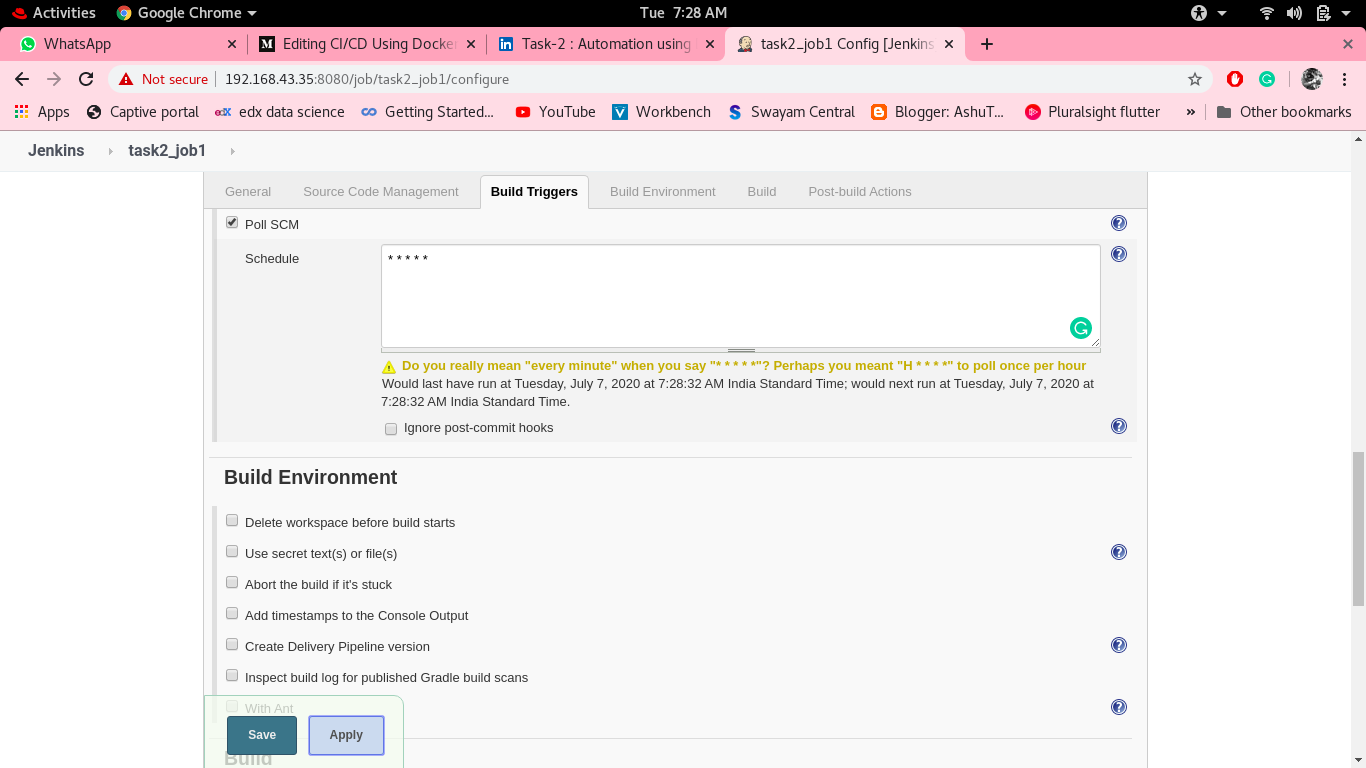
**job1**:- It will check and put an eye on the GitHub repo <https://github.com/ashut08/jenkins_github.git> and check for the latest update/commit in the repository.

Image for post



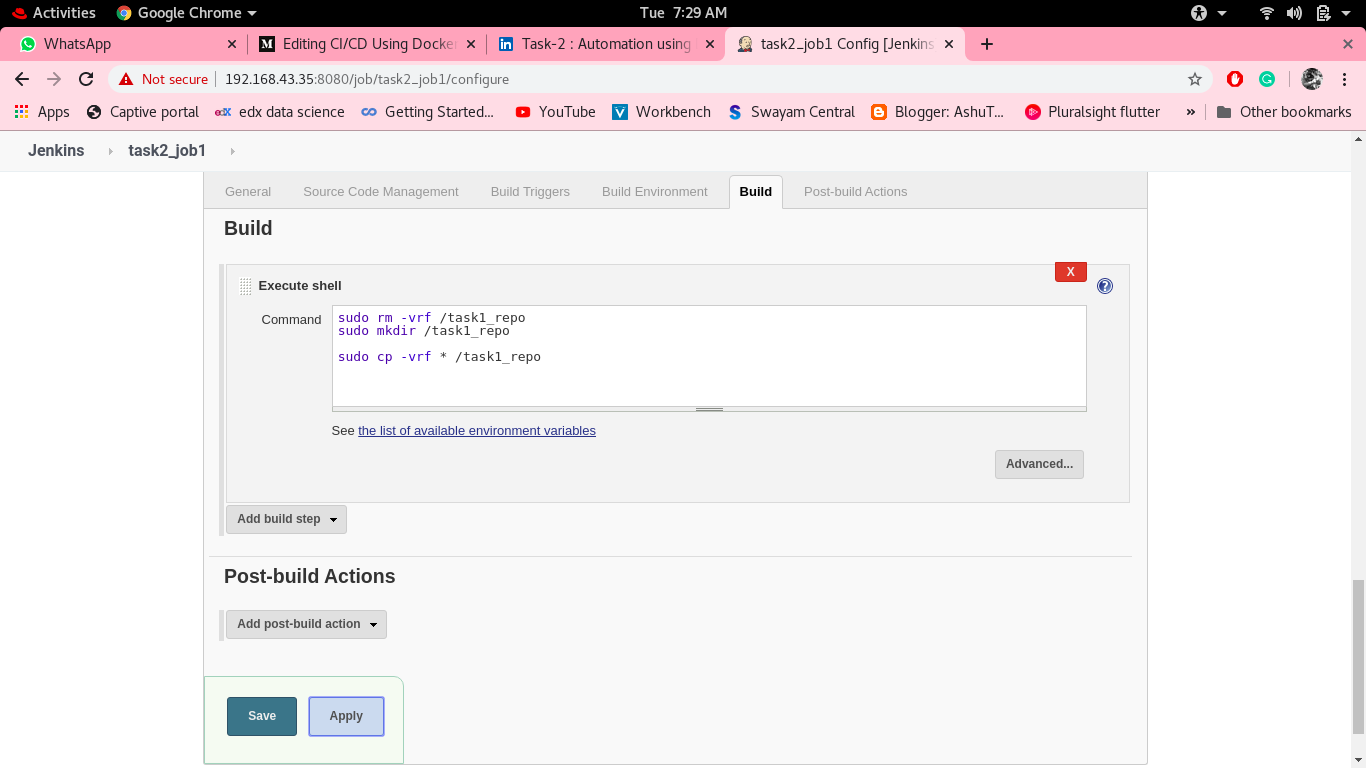
Give Repo Url

Image for post



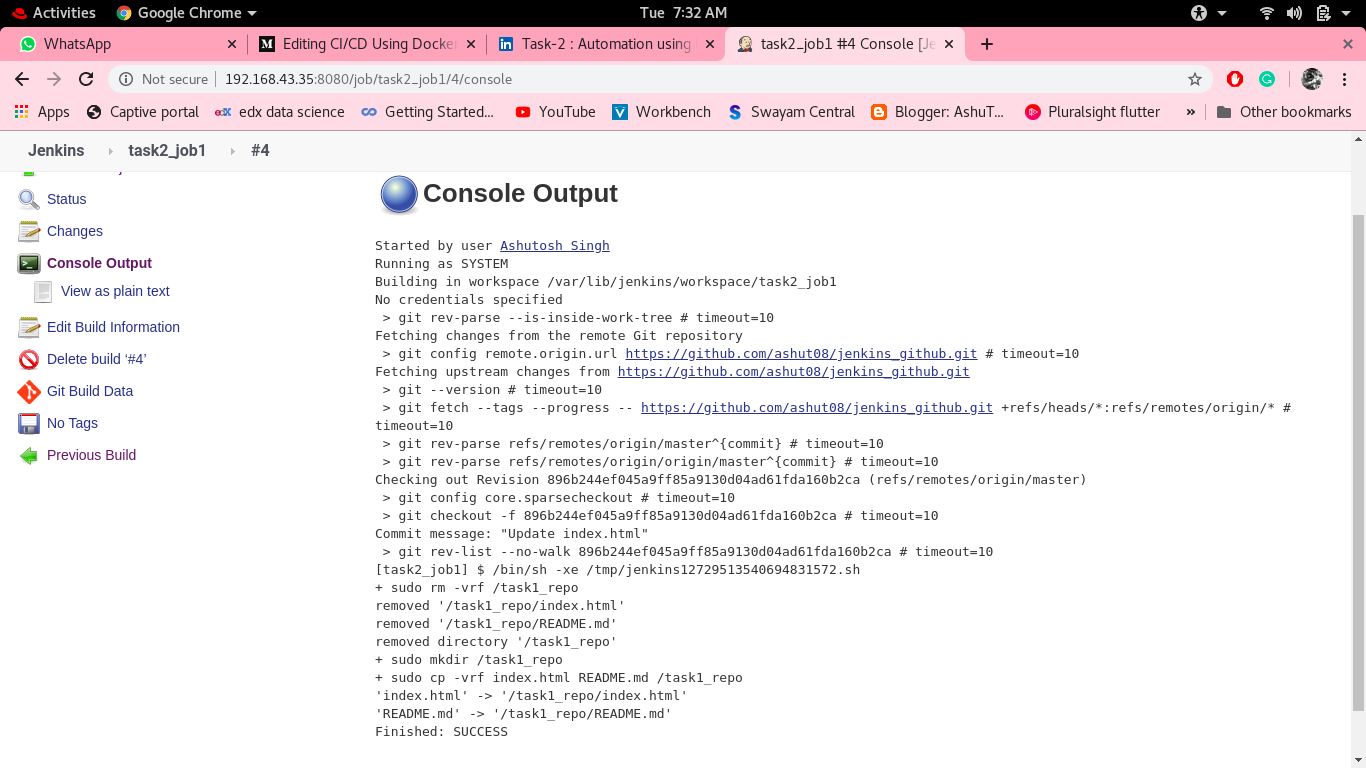
Apply poll SCM

Image for post



Execute cell which pulls repo in task1\_repo folder

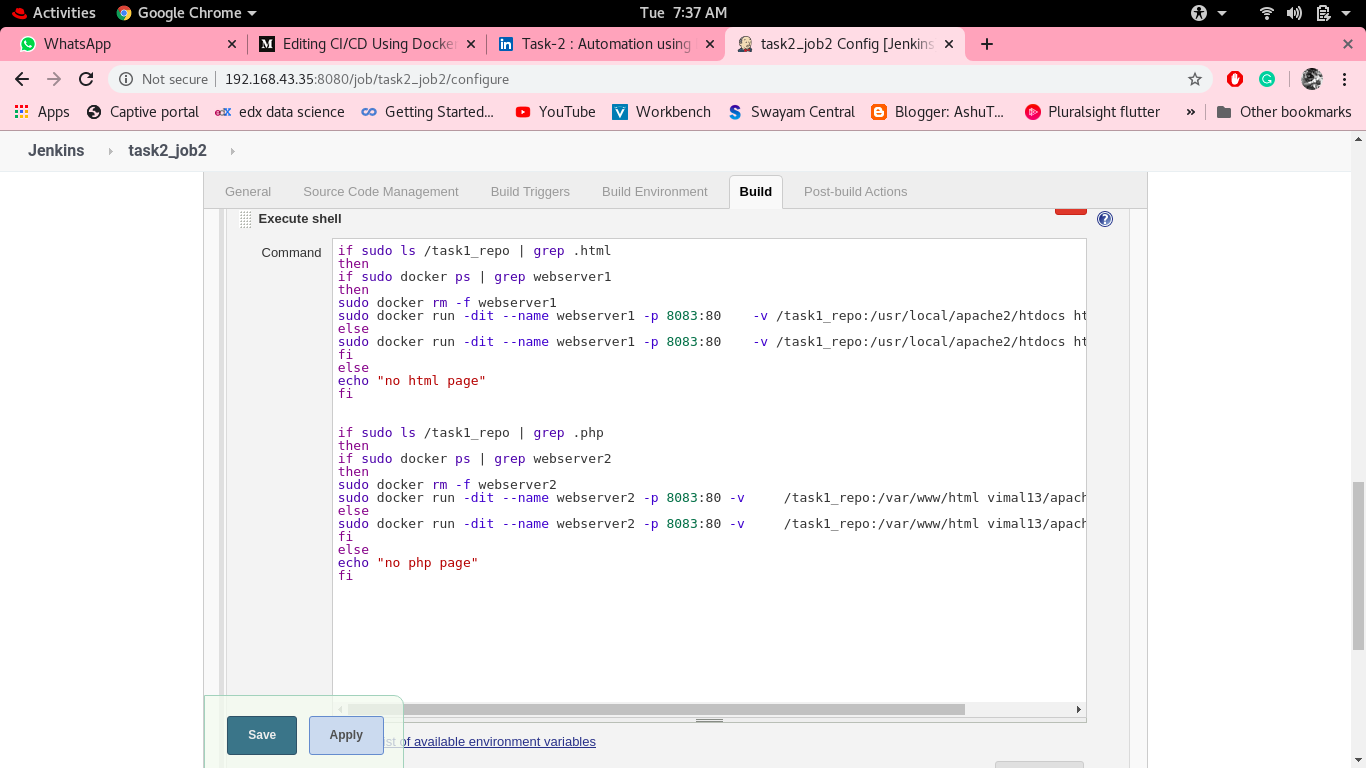
Image for post



job1 console output

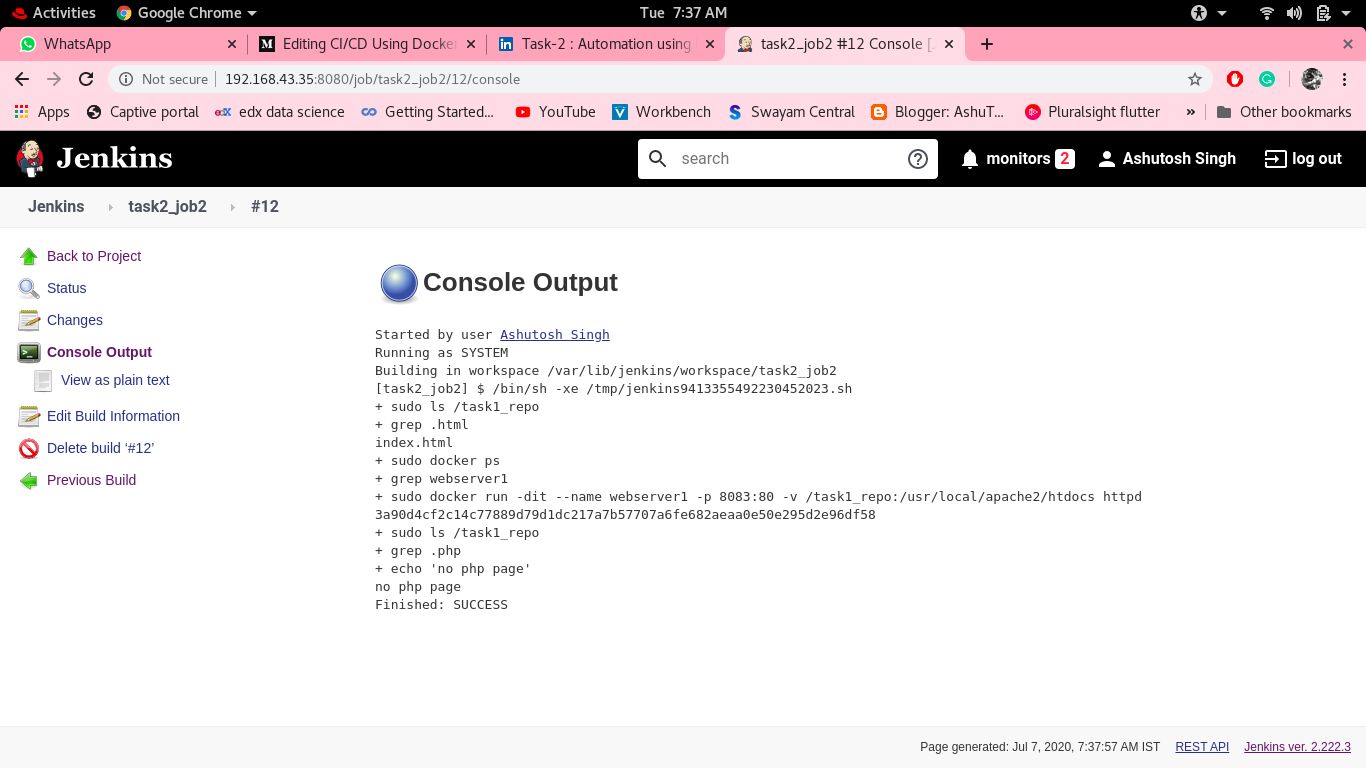
**Job2:**By looking at the code or program file, Jenkins should automatically start the respective language interpreter install image container to deploy code (eg. If code is of PHP, then Jenkins should start the container having that image/iso that has PHP installed).

Image for post



Jenkinsn\_Job2

Image for post



Job2\_console\_output

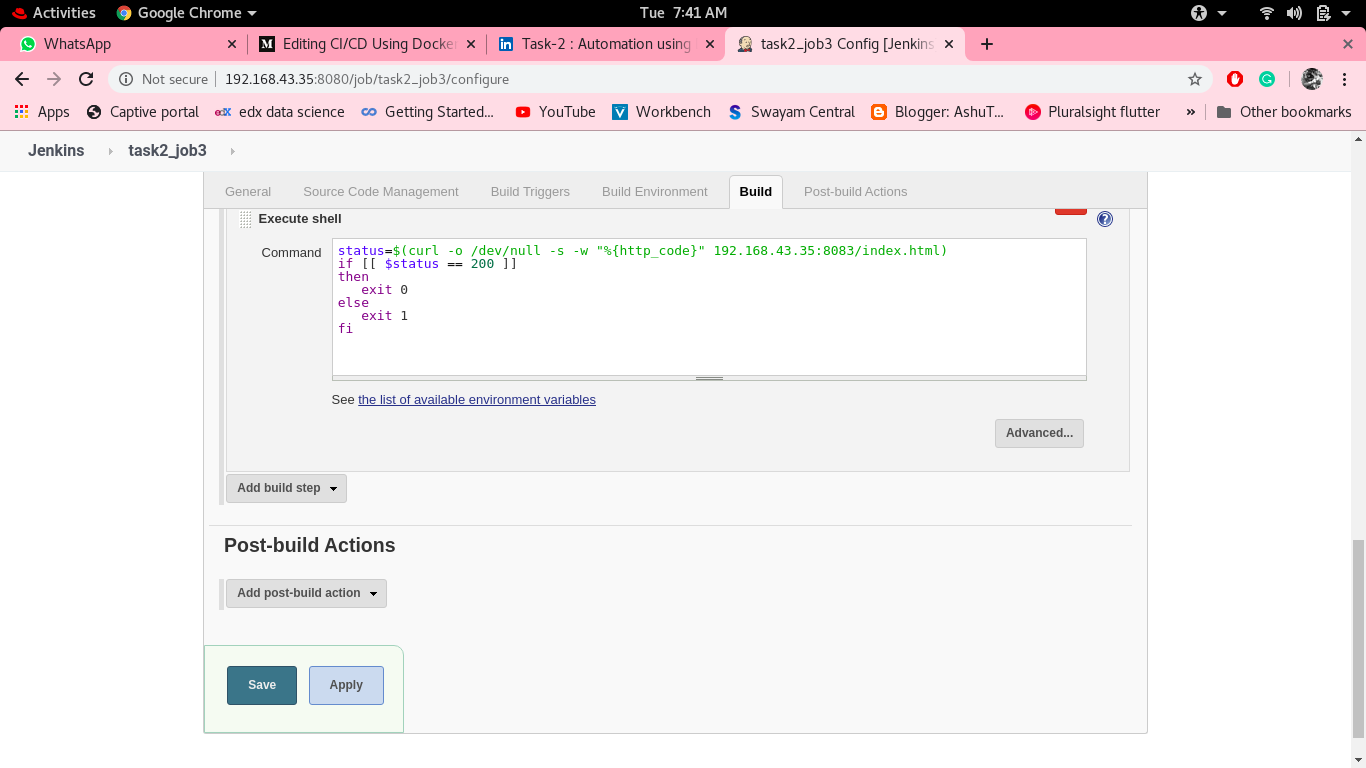
**Job3:**Test your webpage if it is working or not.

With the help of curl command, we will see whether our webpage is working properly or not.

For success, the code is 200

and for any error code is 500

Image for post



jenkins\_job3

In my case page working fine

Image for post

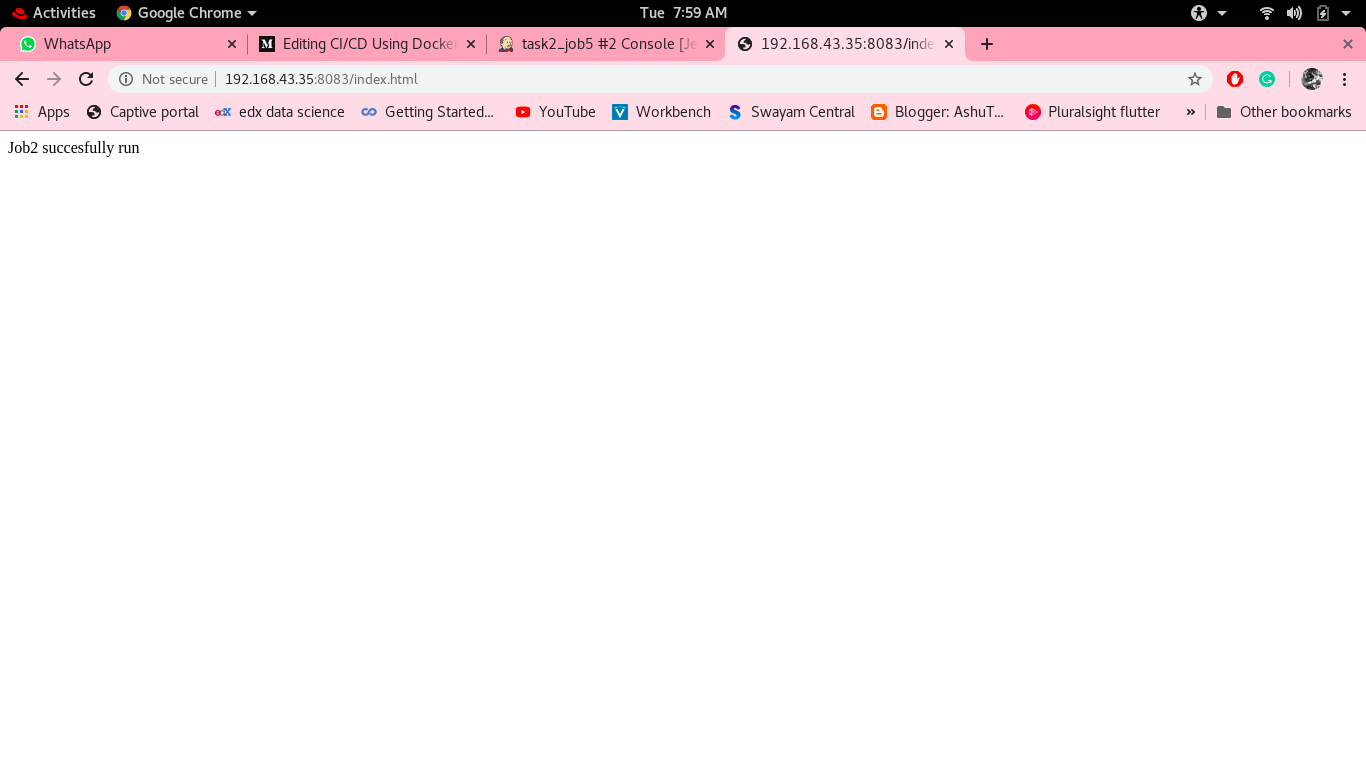
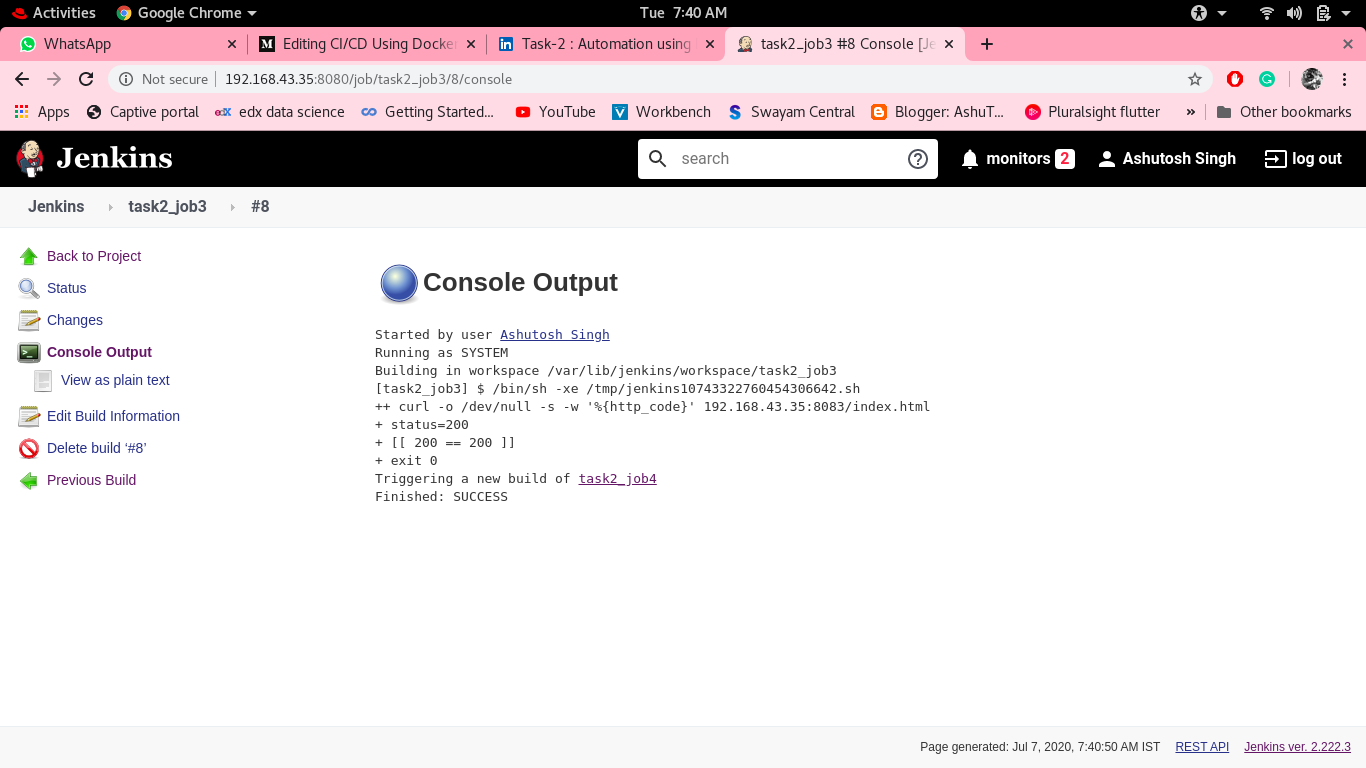


Image for post

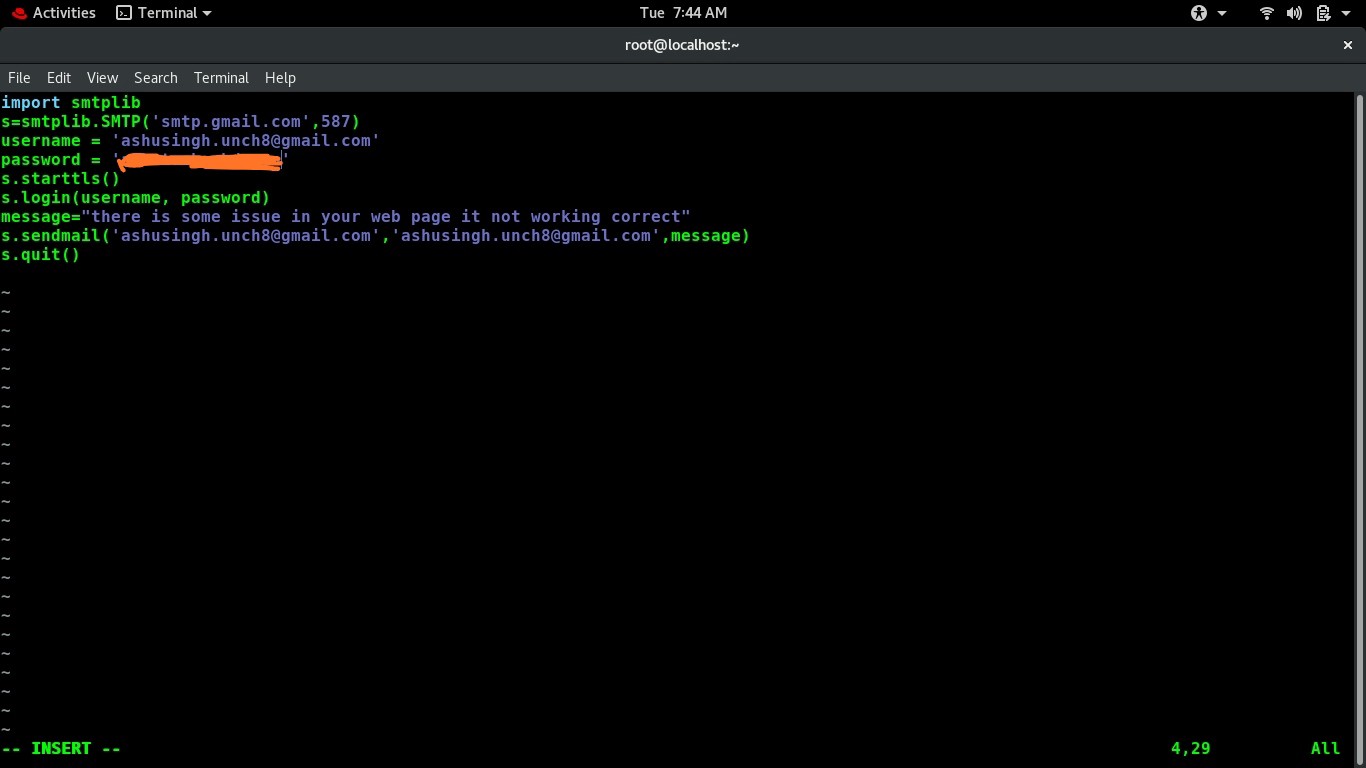


task3\_console\_output

**Job4:**-If our webpage is not working properly then job4 is to notify Developer.

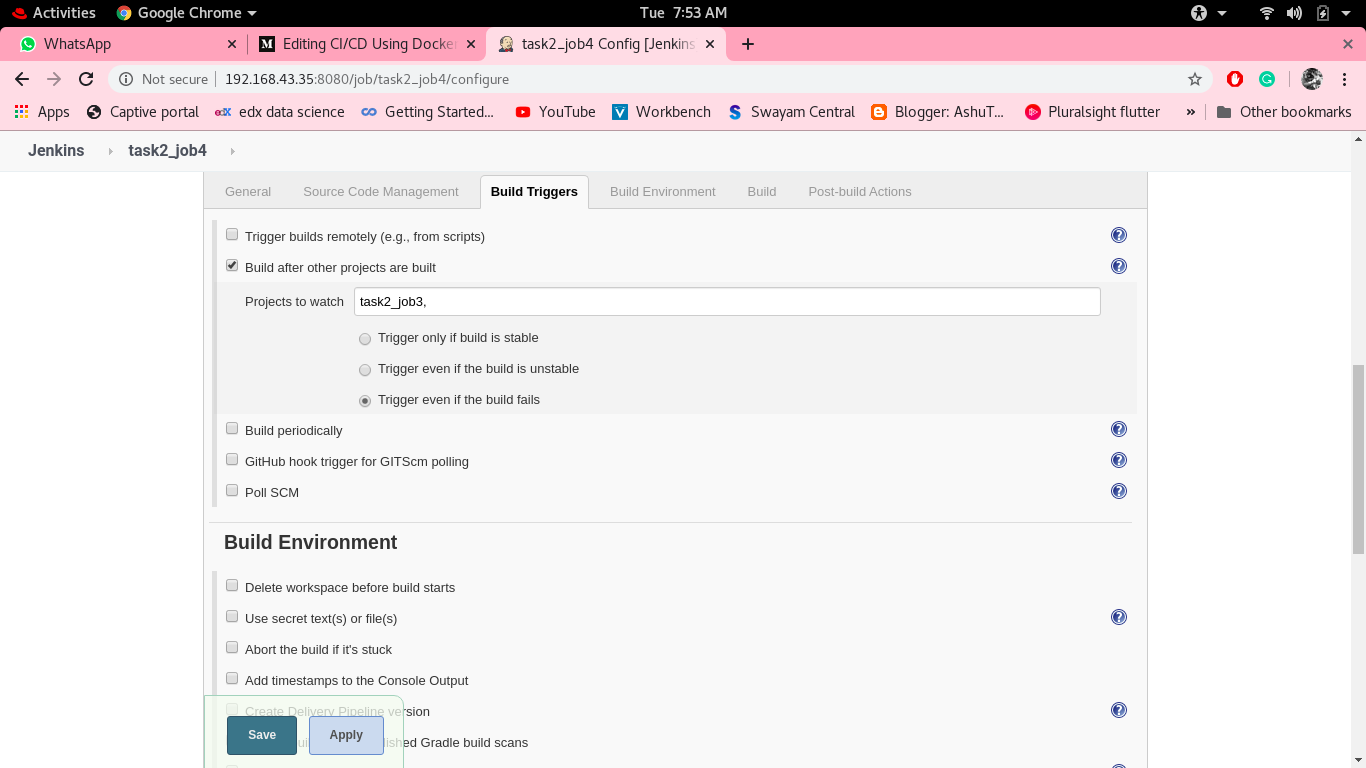
for this, I write a python script to send a notification which executes if job3 failed

Image for post



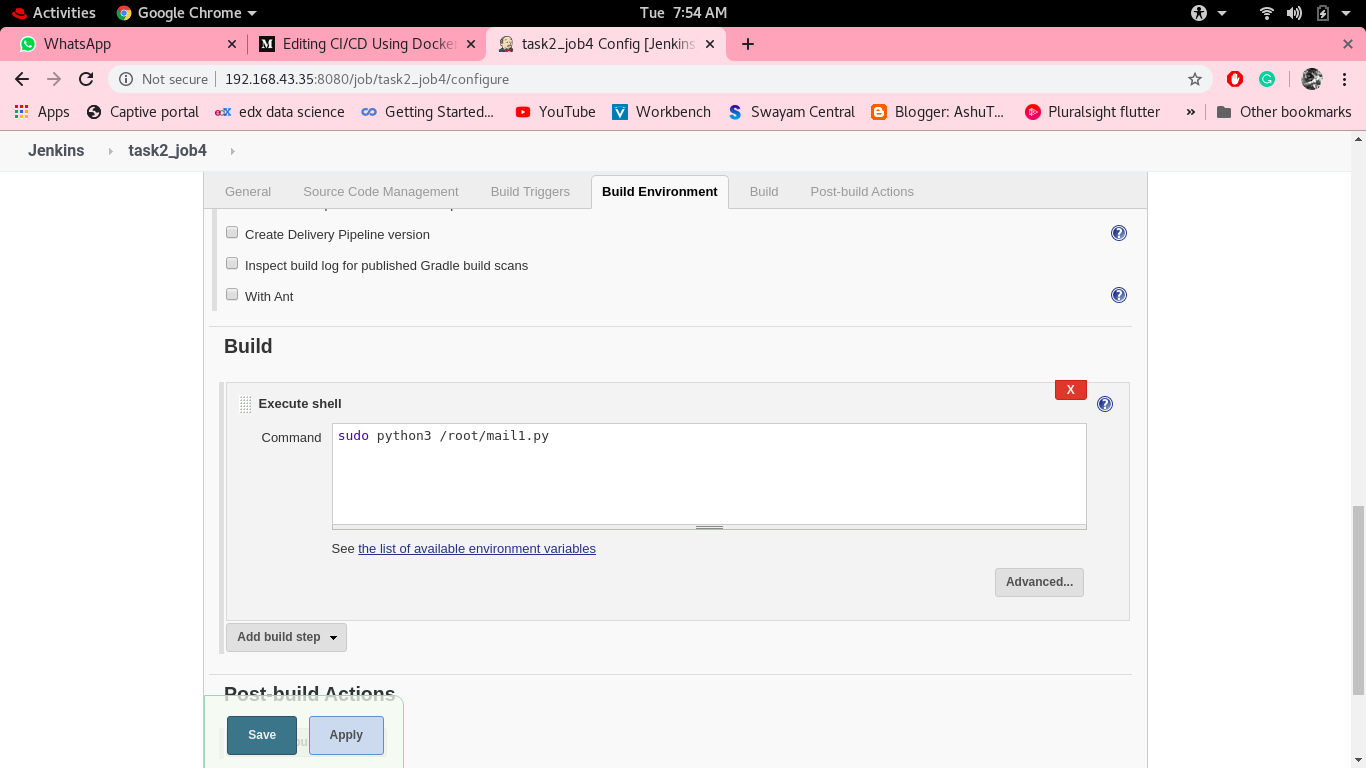
Python file for sending e-mail

Image for post



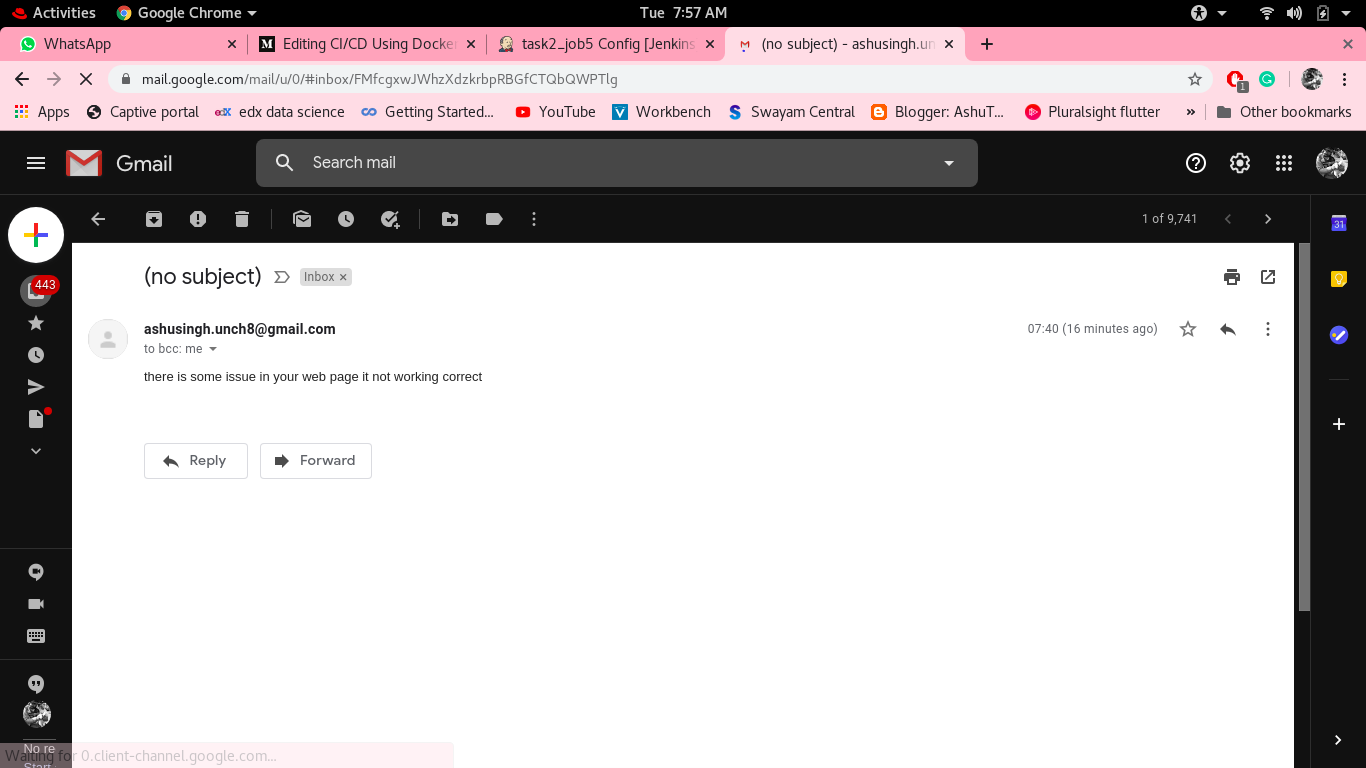
Build trigger if job3 fail

Image for post



Execute python file

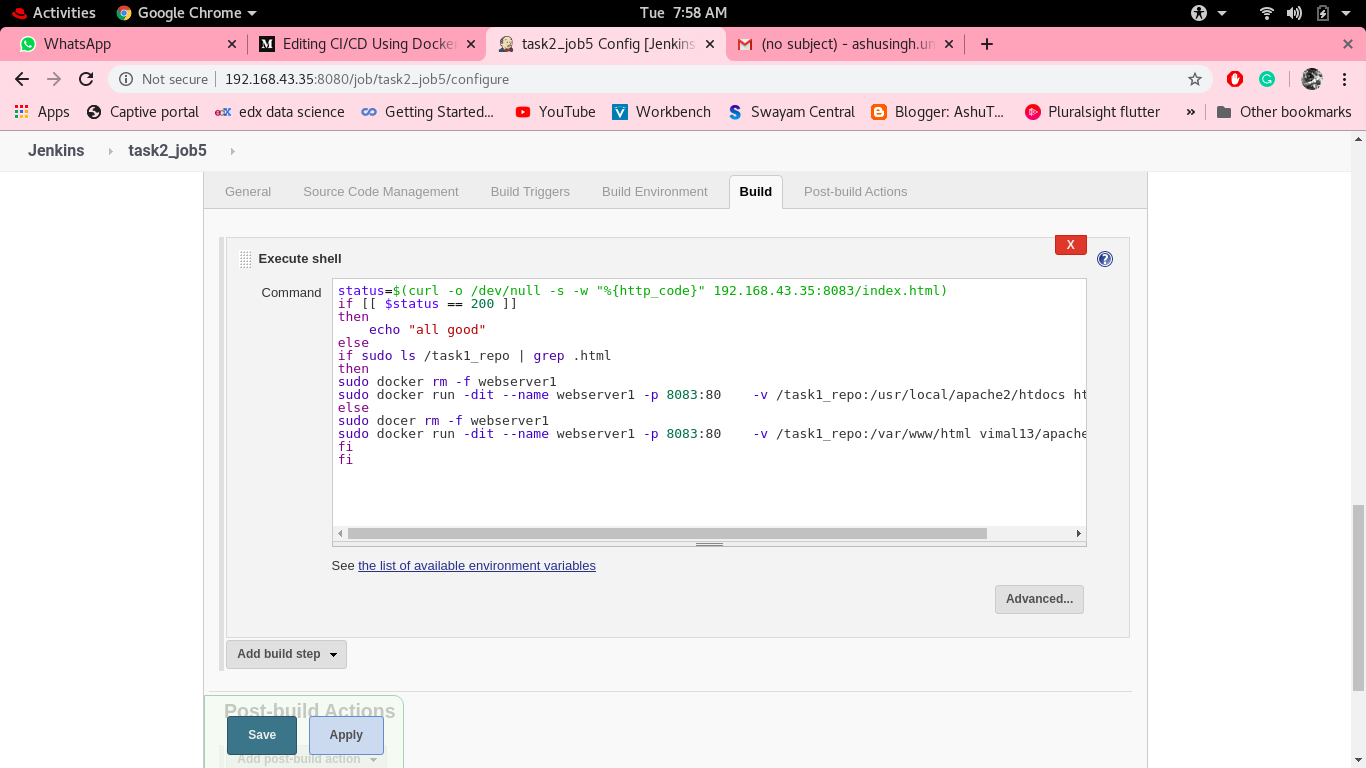
Image for post



Mail send

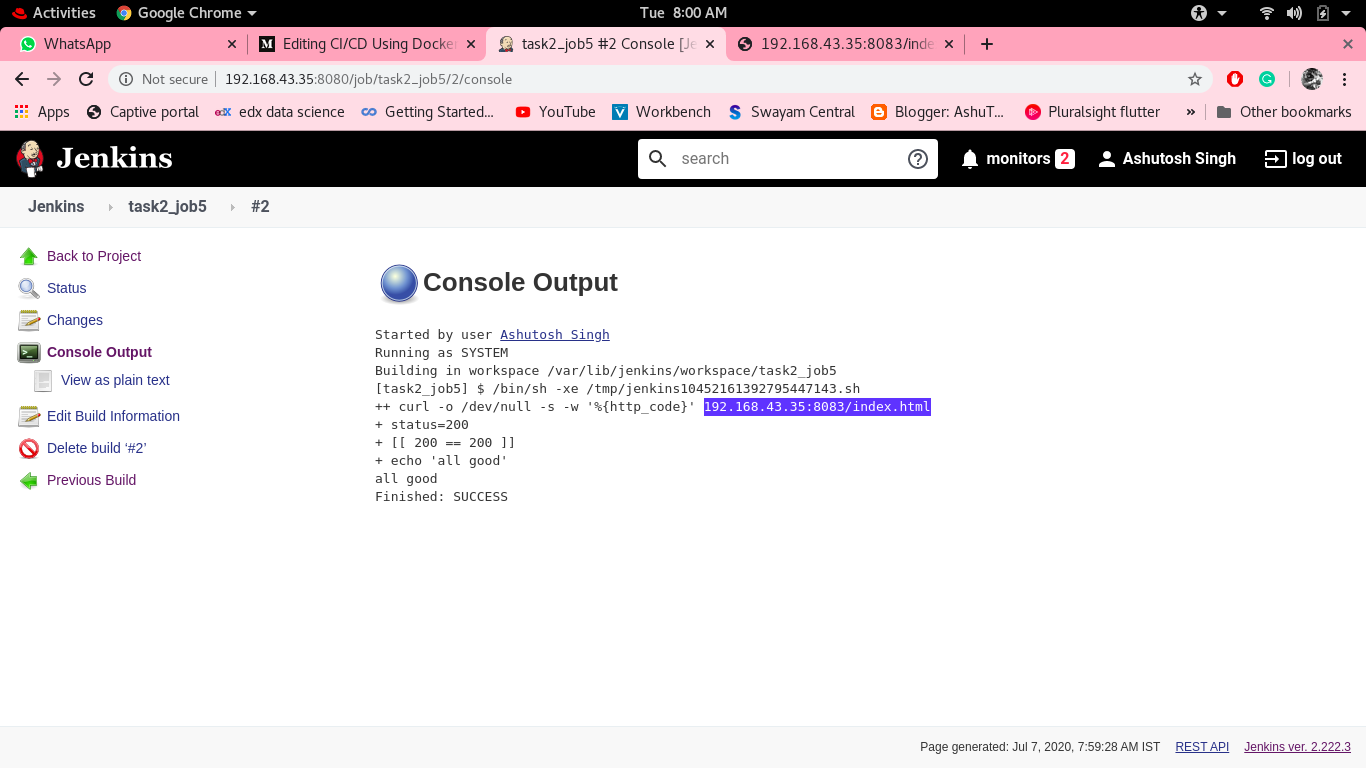
**Job5:**for monitor: If the container where the app is running. fails due to any reason then this job should automatically start the container again.

Image for post



Jenkins\_job5

Image for post



job5\_console\_output

Here my build pipeline of all jobs

Image for post

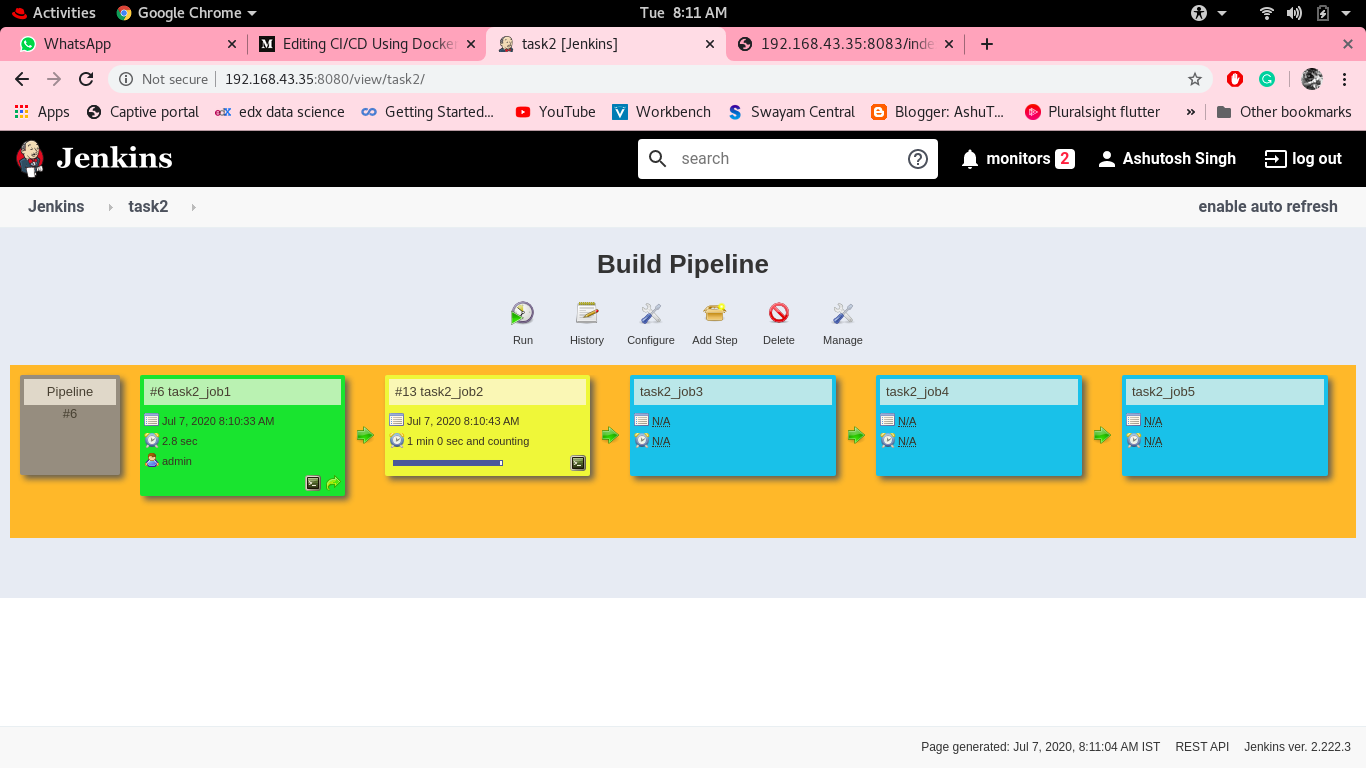
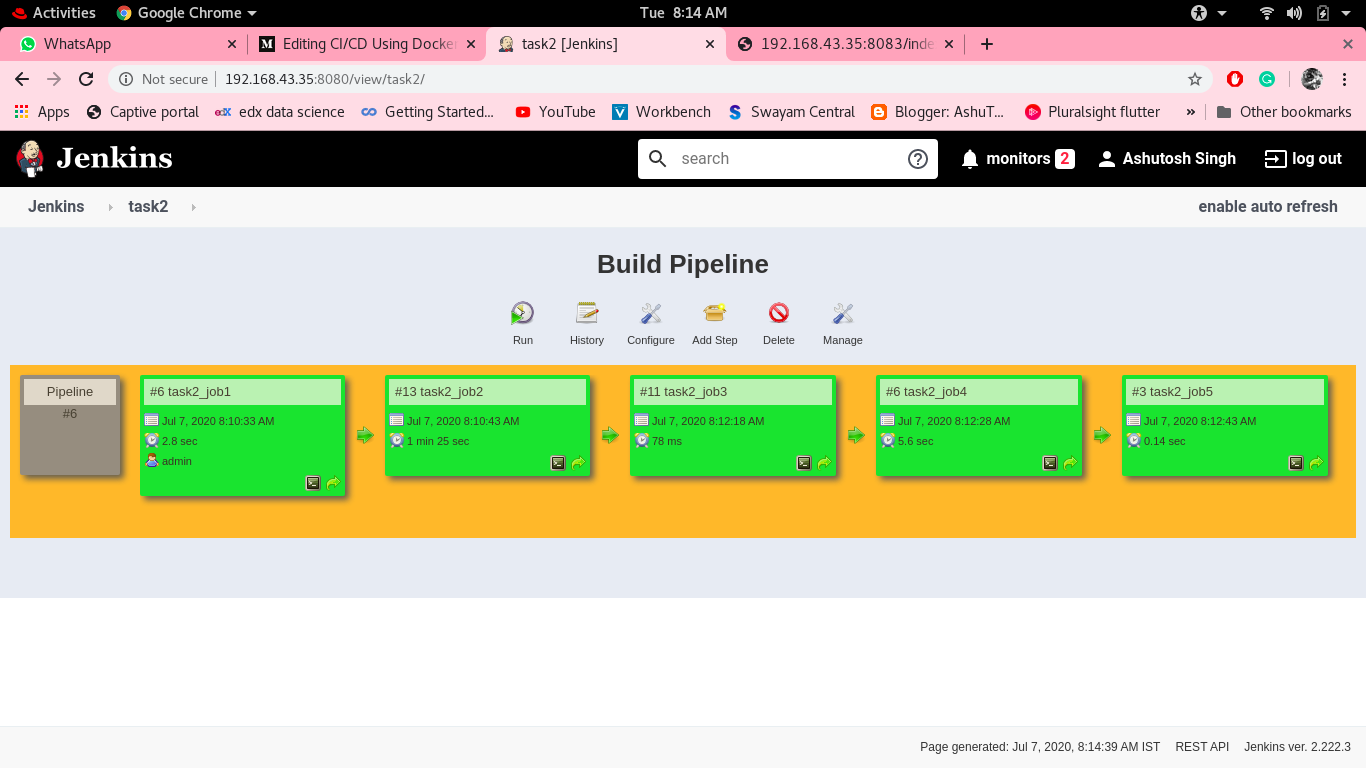


Image for post



[The Startup](https://medium.com/swlh?source=post_sidebar--------------------------post_sidebar-----------)

Medium's largest active publication, followed by +716K people. Follow to join our community.

Follow

50