1. Create container image that’s has Jenkins installed using dockerfile

2. When we launch this image, it should automatically starts Jenkins service in the container.

3. Create a job chain of job1, job2, job3 and job4 using build pipeline plugin in Jenkins

4. Job1 : Pull the Github repo automatically when some developers push 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 app if it is working or not.

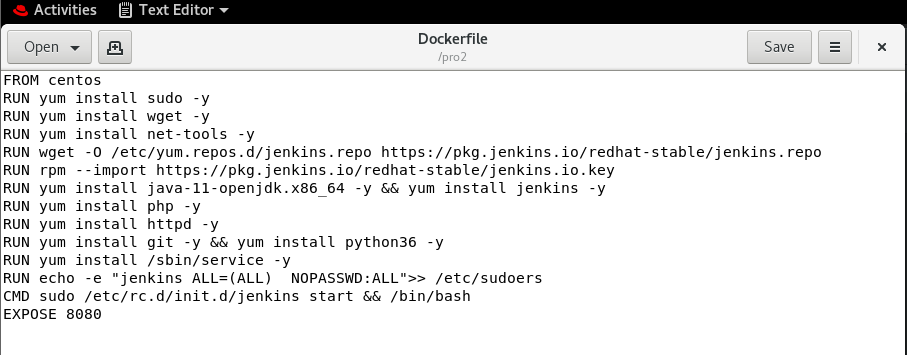
7. Job4 : if app is not working , then send email to developer with error messages.

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

Solution

Step1:

a) We will create the environment of Jenkins in Docker file.



🡪FROM is used for the image that we want to use in our environment (It acts like docker pull)

🡪RUN is used for executing the command that the features required for project while building the new image

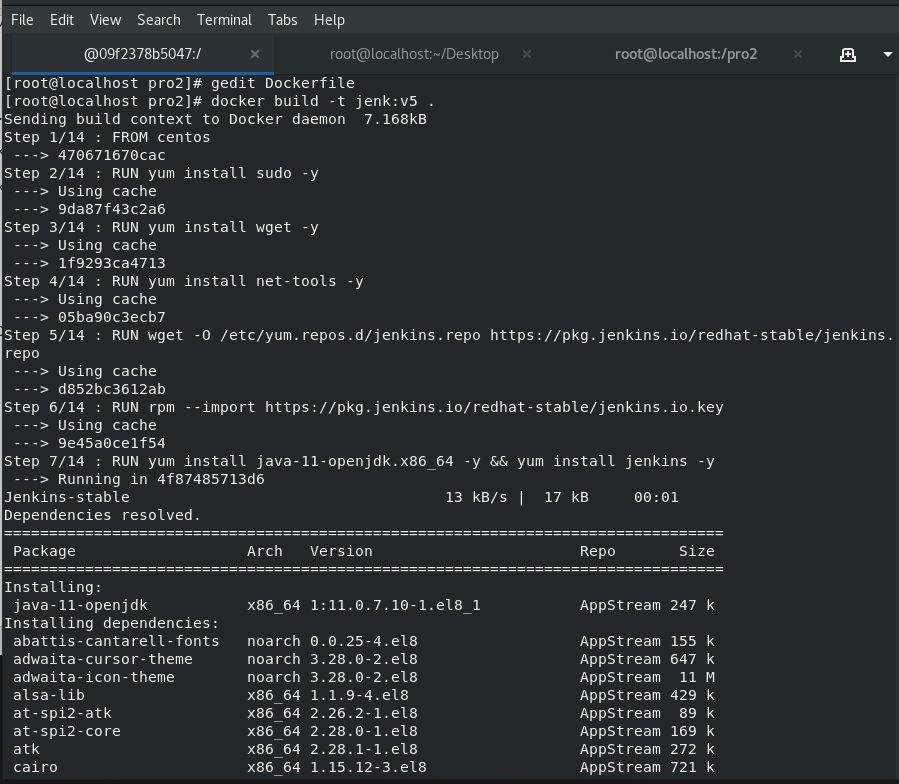
🡪CMD here is used to start Jenkins and keep the container live even after executing in this project

🡪Expose is used for patting, docker is isolated so we must expose, that the client should connect.

Step2: Building the image

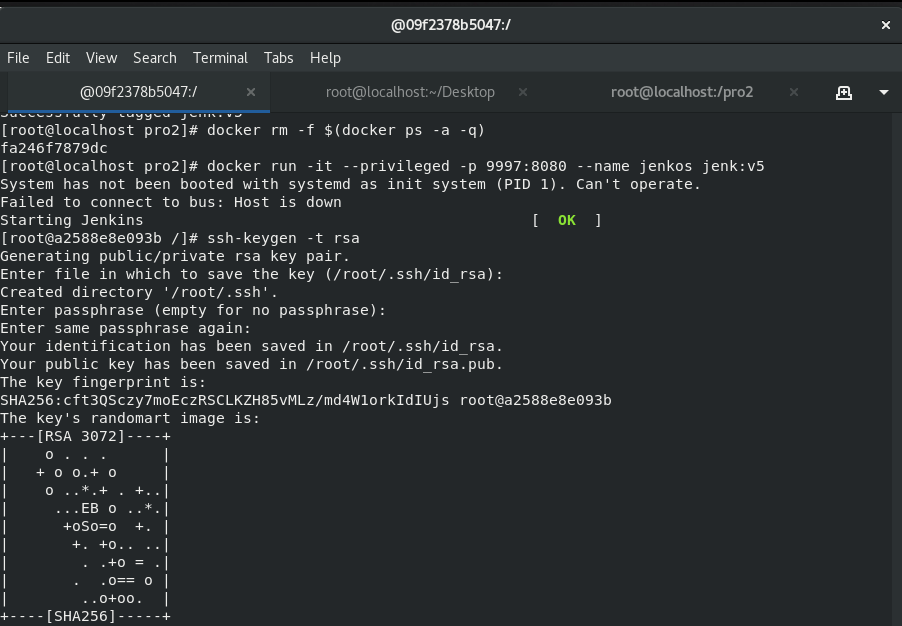
# docker build -t jenk:v5 /pro2/

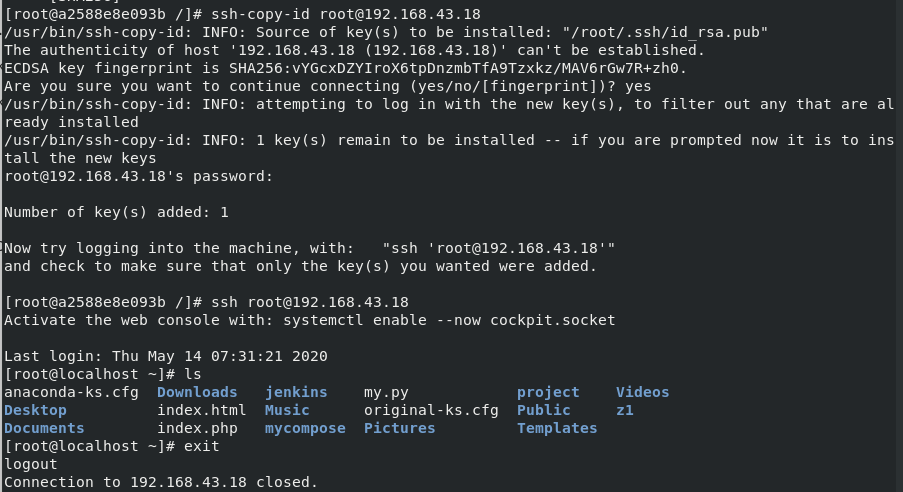
🡪/pro2/ is the directory where Dockerfile is created.



Step3: Starting Jenkins container

# docker run -it –privileged -p 9997:8080 --name jenkos jenk:v5





🡪After Launching the container adding SSH key using command **ssh keygen -t rsa**

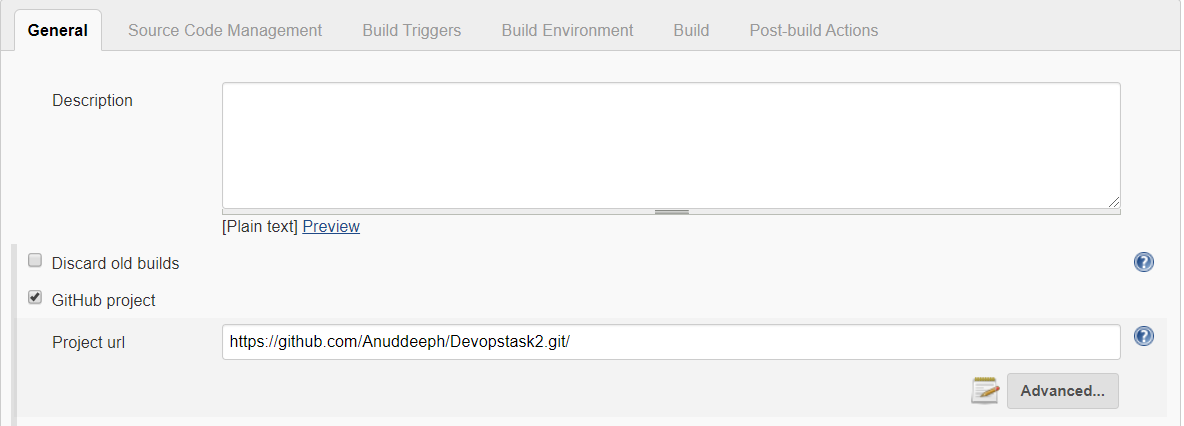
This command creates the ssh key and we have to copy it with the host i.e rhel8 using command **ssh-copy-id root@hostip [here bhostip == 192.168.43.18]**

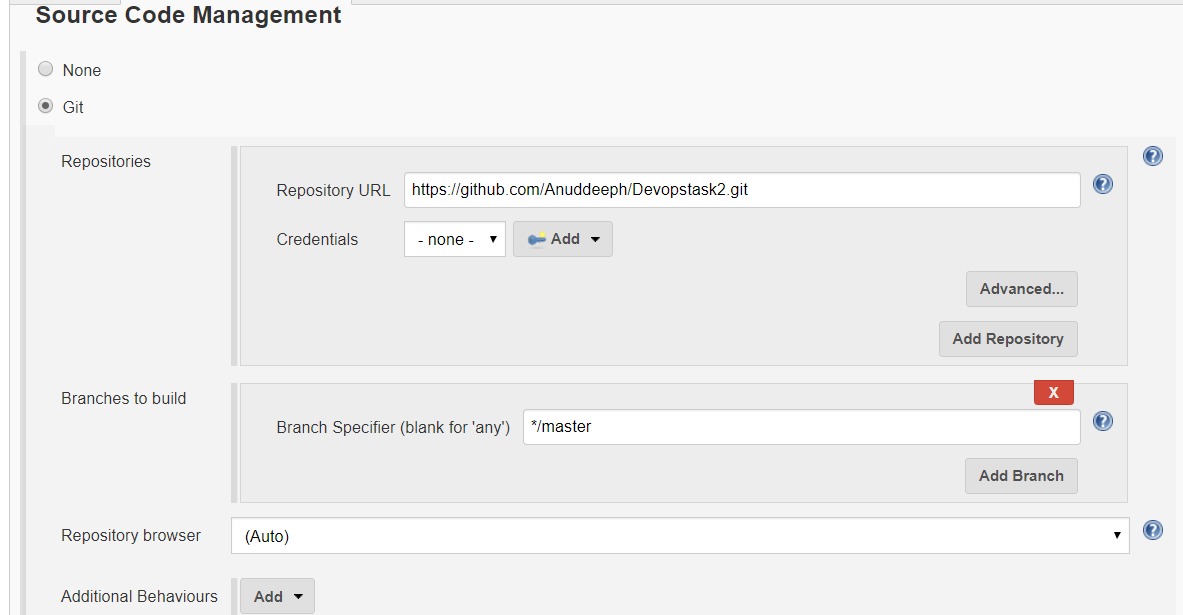
Step 4: Configuring Jenkins

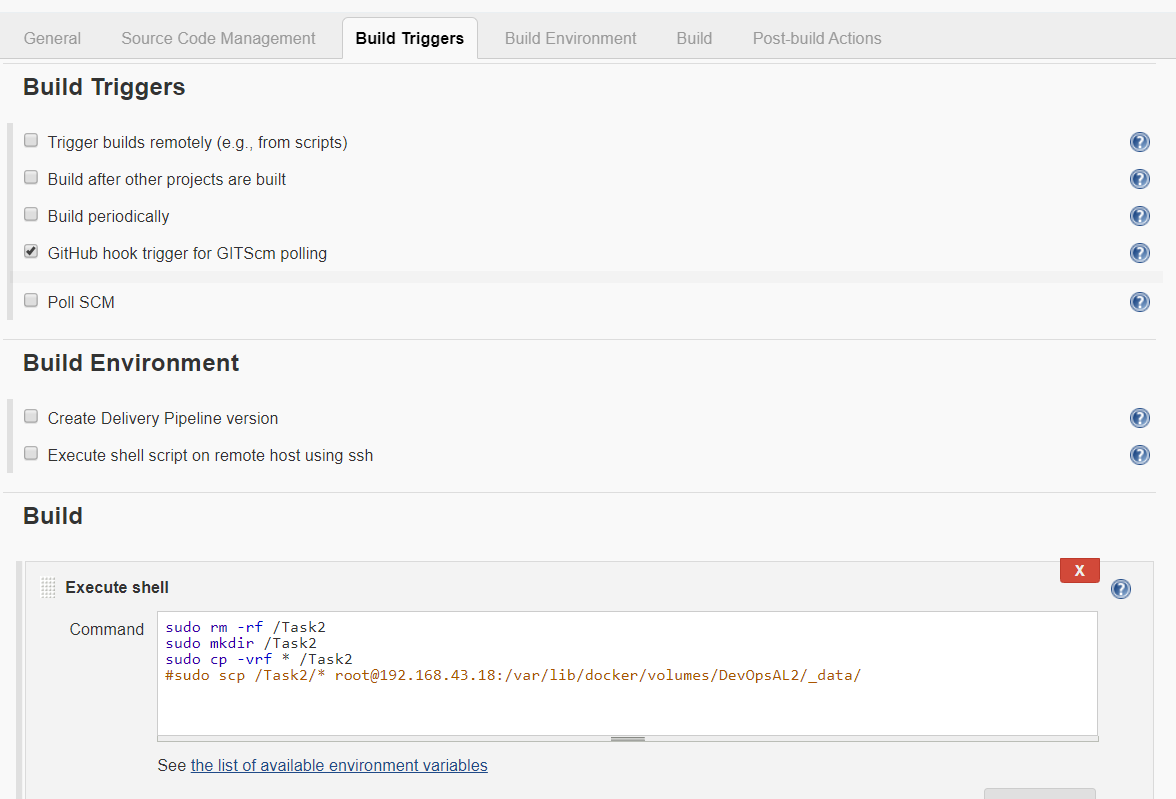
Configuration is nothing but adding required plugins to use in the project like GitHub, Build Pipeline, Delivery Pipeline.

Step 5: Creating Jobs that we required in this project (i.e Automation)

a)Pulling the GitHub repo (job1)



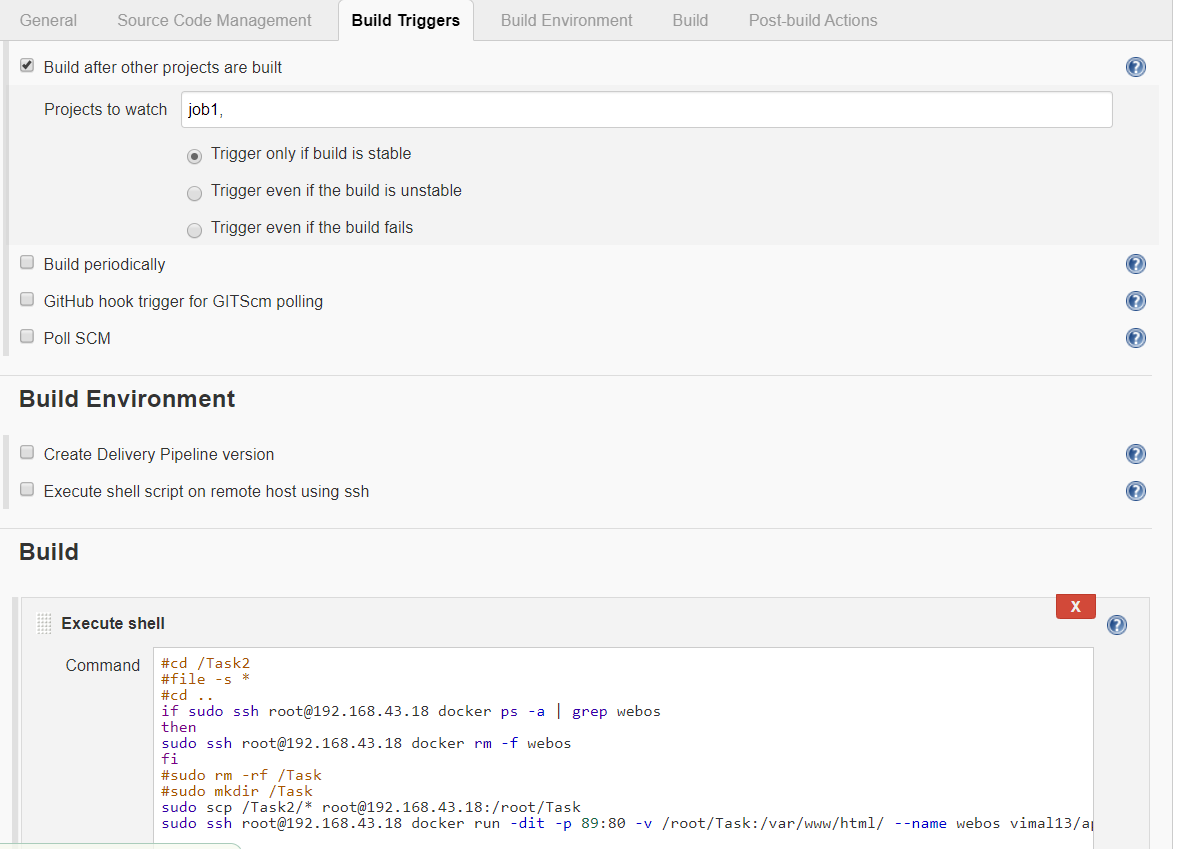






In this job1, we are removing the existing directory, creating the new directory, and copying the files from GitHub repo. # also we can use docker volume to store data and we can mount to docker container.

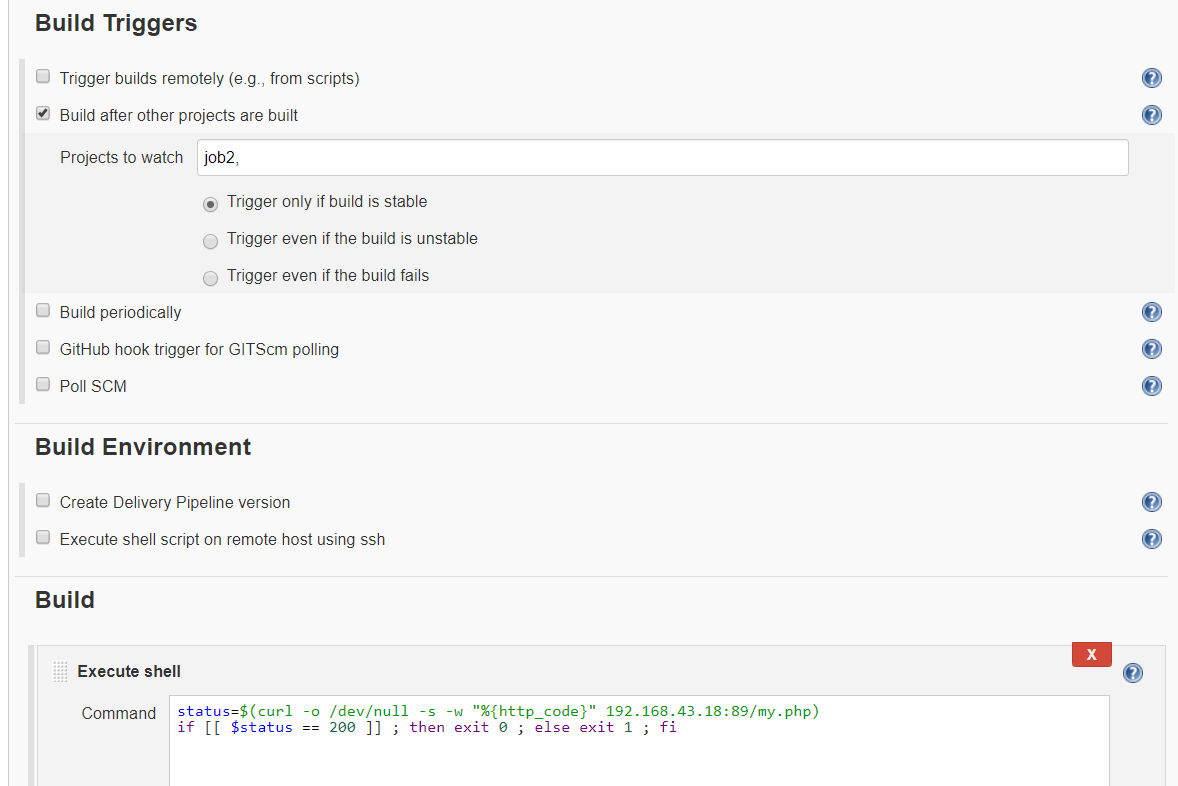
B) Launching the container and copying the files to webserver (location /var/www/html)

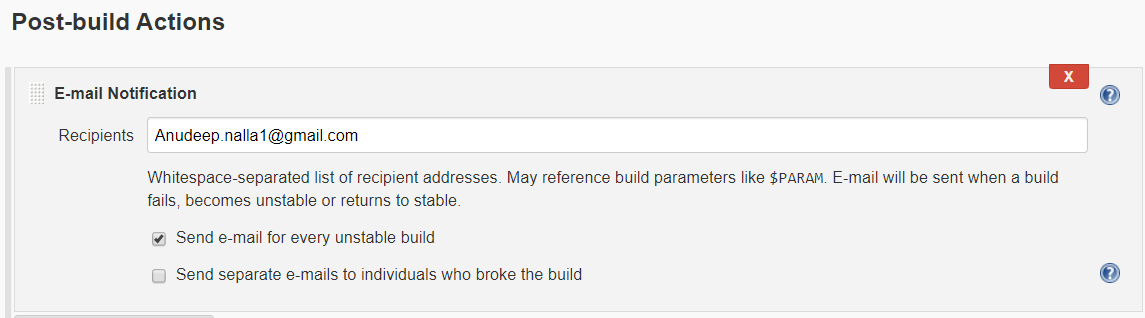




# while running the container we can also mount the volume DevOpsAL2

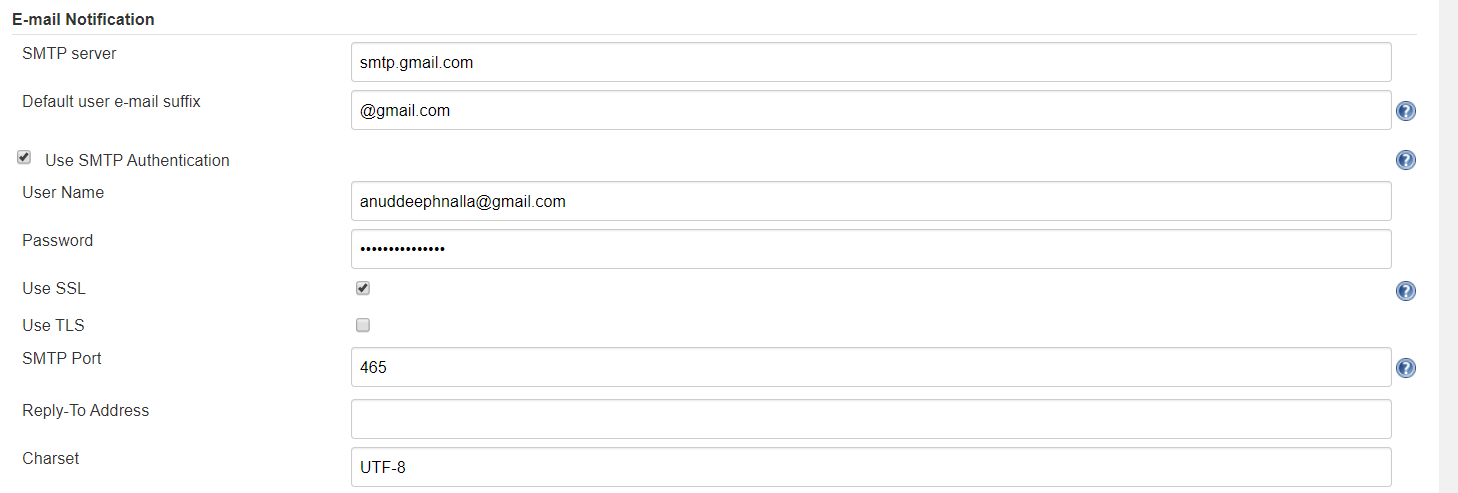
C) Testing and Reporting error







* In this Testing, we are storing the status code of php page and dumping the output to null, status code of success is 200, success code for error in code is 500. And we are sending mail if there any unstable build occurs. Exit 0 is to execute the success and exit 1 is to execute the failure
* Go to manage Jenkins 🡪 configure system



If then also you face error, then goto your jenkos container

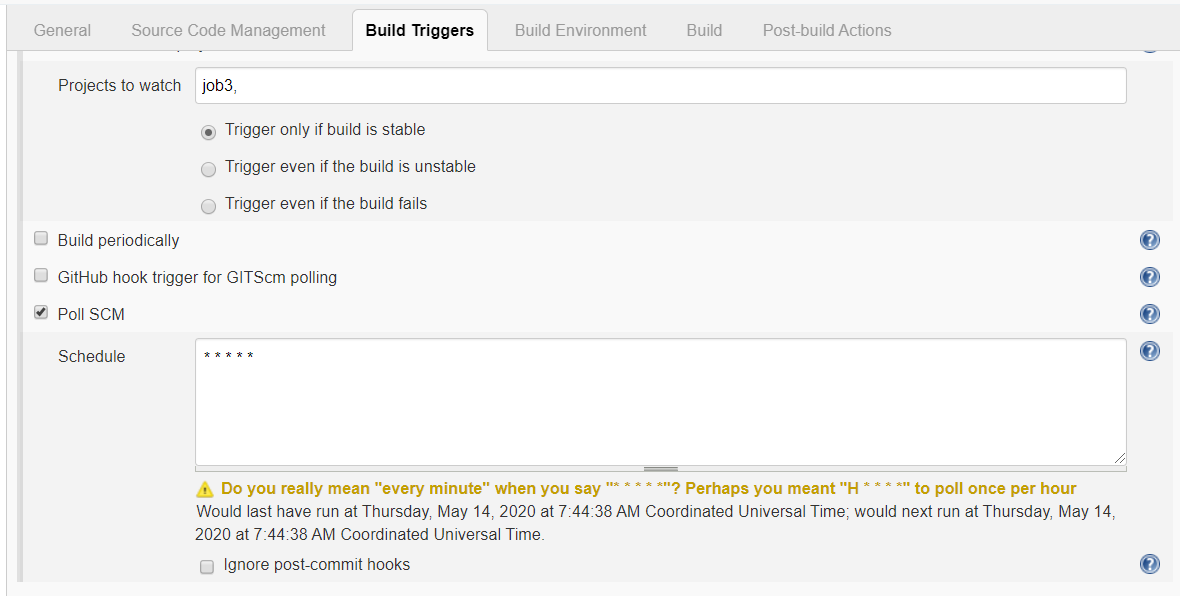
vim /etc/sysconfig/jenkins

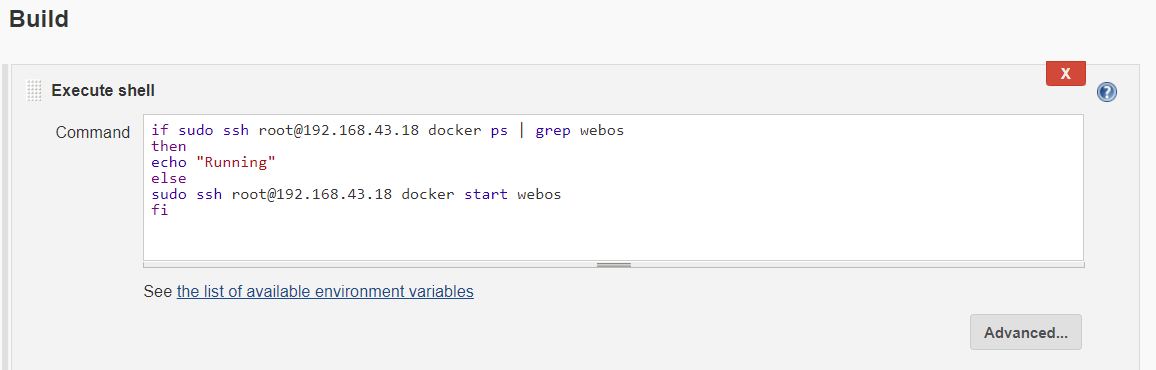
#change

JENKINS\_JAVA\_OPTIONS="-Djava.awt.headless=true -Dmail.smtp.starttls.enable=true -Dmail.smtp.ssl.protocols=TLSv1.2"

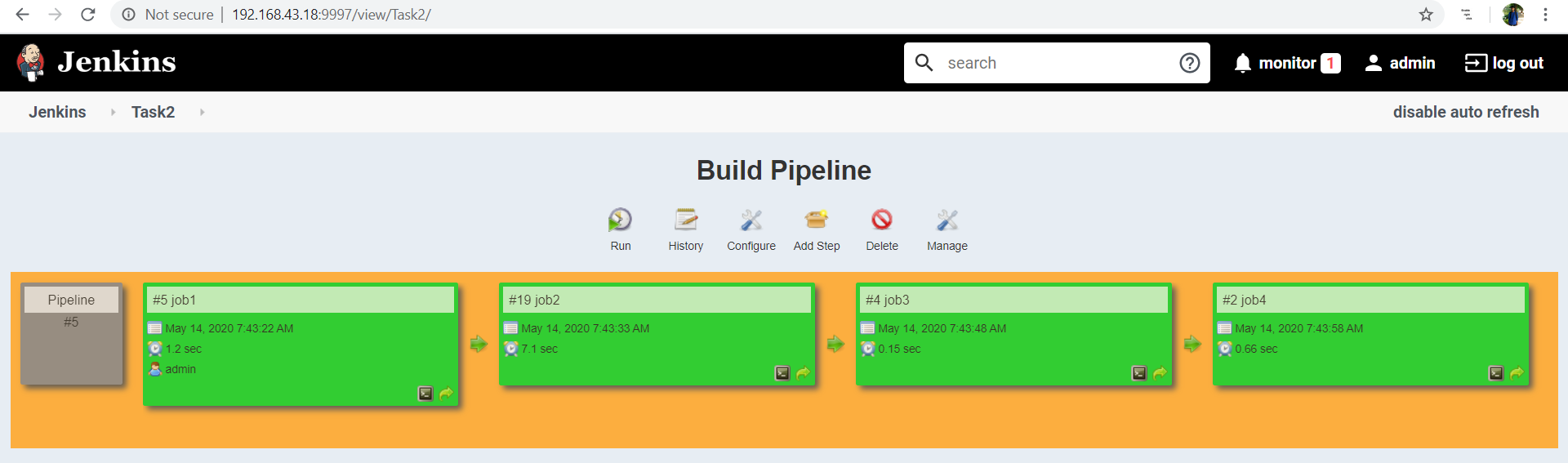
Then go to google setting 🡪 security 🡪 on less secure app.

D) Monitor the container (job 4)





This will keep checking the web container after (provided) period and if container is down the it will start the container.



This is how the Final pipeline looks:

Github repo URL: <https://github.com/Anuddeeph/Devopstask2.git>

t took me more than 20+ hrs. to make this project. I faced many errors while making this project. But every error was a great learning step. I learnt a lot from this project. I am very thankful to Mr. Vimal Daga sir ,who is teaching me such a great content and I am very much thankful to the volunteers and my group members also who encouraged us to solve the error by our own.

If you have any query or suggestion , Please let me know

Thank You