Project-Integration of Devops tools with Jenkins

You have been Hired as a Devops Engineer in xyz software company. They want to implement CI/CD pipeline in their company. You have been asked to implement this lifecycle as fast as possible. As this is a product-based company, their product is available on this GitHub link.

https://github.com/hshar/website.git

Following are the specifications of the Continuous integration:

1. Git Workflow has to be implemented

2. Code Build should automatically be triggered once commit is made to master branch or develop branch.

If commit is made to master branch, build and publish website on port 82. If commit is made to develop branch, just build the product, do not publish.

3. Create a pipeline for the above tasks.

4. Create a container with Ubuntu and Apache installed in it and use that container to build the code and the code should be on '/var/www/html'.

Solution:

I'm cloning the repository to my jenkins server with "git clone https://github.com/Harshit407/Nouse.git"

Now we can see our clone directly which is named as "Nouse". Go inside the Nouse directory

Now add the docker container file

Nano docker -> FROM ubuntu

RUN apt-get update

RUN apt-get install apache2 -y

ADD . /var/www/html

ENTRYPOINT apachectl -D FOREGROUND

Run "git init" -> "git commit -m "don't know change"" -> All of this we are doing in master branch only.
Create the "Develop" branch as per the question requirement

Now "git push --all" in the master branch

Create the EC2 instance for the Production in which we will install the docker.

apt-get install docker.io
apt-get update

Now Create two jobs

1st for the Master =Develop branch

2nd for the Production = Master branch

For develop branch don’t publish it

For master branch publish it

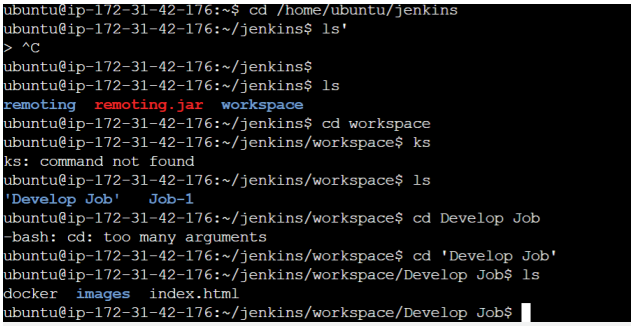
Now Create the job-1

Inside the github directly given the below:

https://github.com/Harshit407/Nouse.git (Git Project URL)

We are creating the job named as “Develop job” -> Free style project

Description I have gave “For develop branch don't publish it” -> Github project -> https://github.com/Harshit407/Nouse.git ->Restrict where this project can be run -> Production (this is my Slave-1 server name) -> Git -> https://github.com/Harshit407/Nouse.git -> Update the branches to build as -> */develop -> Check GitHub hook trigger for GITScm polling -> update webhooks with master server Public IP ->Apply & save.
Testing Build Now.

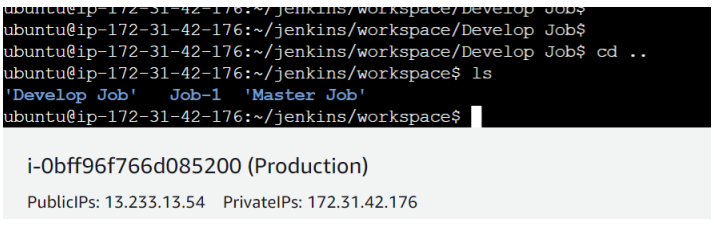


As as per the requirement we got the product in the production server.

->>From this new Master Job which we creating now we want by this whenever commit is made to master branch, build and publish website on port 82

Now create one job name “Master job” -> Free style project -> ok

Description I have gave “For master branch publish it” -> Github project -> https://github.com/Harshit407/Nouse.git ->Restrict where this project can be run -> Production (this is my Slave-1 server name) -> Git -> https://github.com/Harshit407/Nouse.git -> Update the branches to build as -> */master -> Check GitHub hook trigger for GITScm polling -> No need to update webhooks with master server Public IP as we already updated->Apply & save.
Testing Build Now.



Here we can see now also its working

Now we want to host it with docker so,In the master job we need to go in the configuration->Build Steps ->Execute Shell

Write the below script:

sudo docker build /home/ubuntu/jenkins/'Master Job'/ -t myapp

sudo docker run -itd --name websitecontainer -p 82:80 myapp

Note:
Two things here we need to remember is that docker file name needs to be the Dockerfile" and in the above shell script I have mention the path where the file is.

Run the Build Now its working fine and now the file

```
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Processing triggers for ca-certificates (20230311ubuntu0.22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Removing intermediate container 53dd081e056e
 ---> 597ab26dfe2c
Step 4/5 : ADD . /var/www/html
 ---> 5acbdda0f2fa
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
 ---> Running in 2625a7775930
Removing intermediate container 2625a7775930
 ---> a7e0538666fe
Successfully built a7e0538666fe
Successfully tagged myapp:latest
+ sudo docker run -itd -p 82:80 myapp
f8c059490b814e98c77d7177cbc6c8c9235cd0cecfaacec069fd2dfc0ddfa5dc
Finished: SUCCESS
```

==========END==========