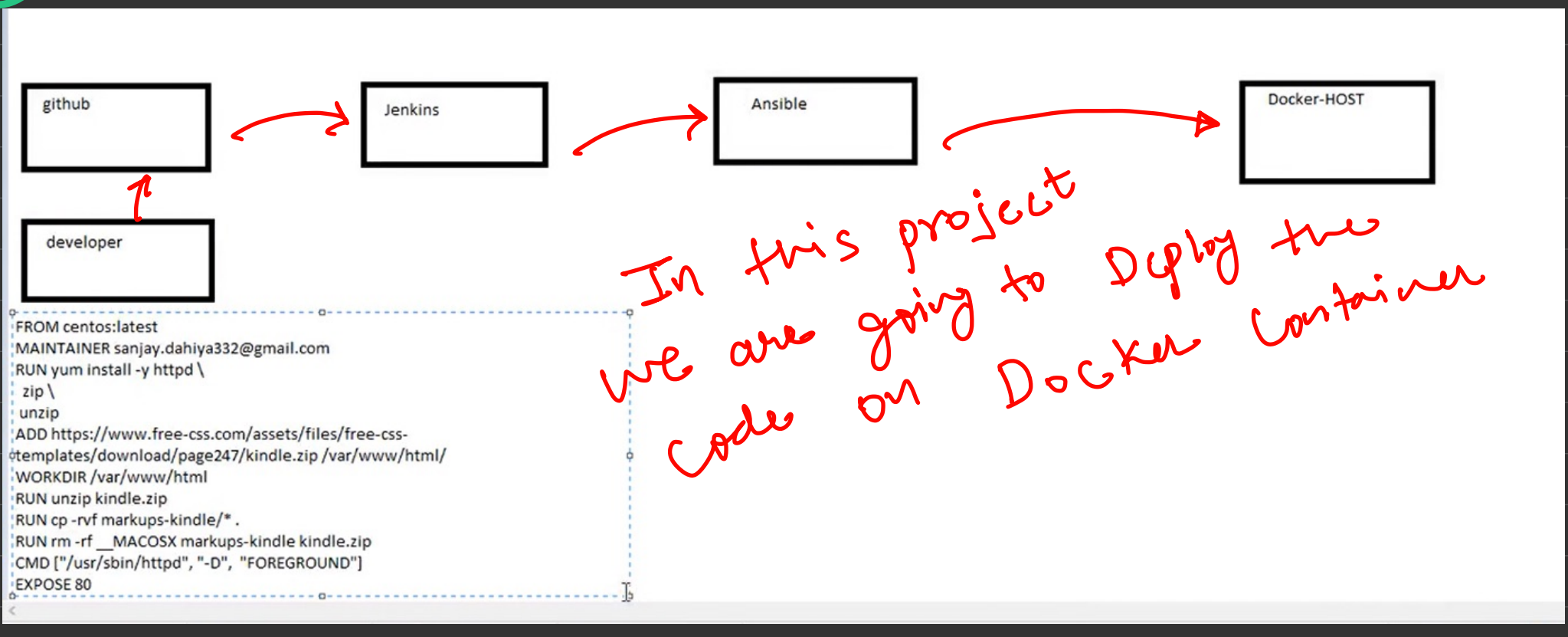
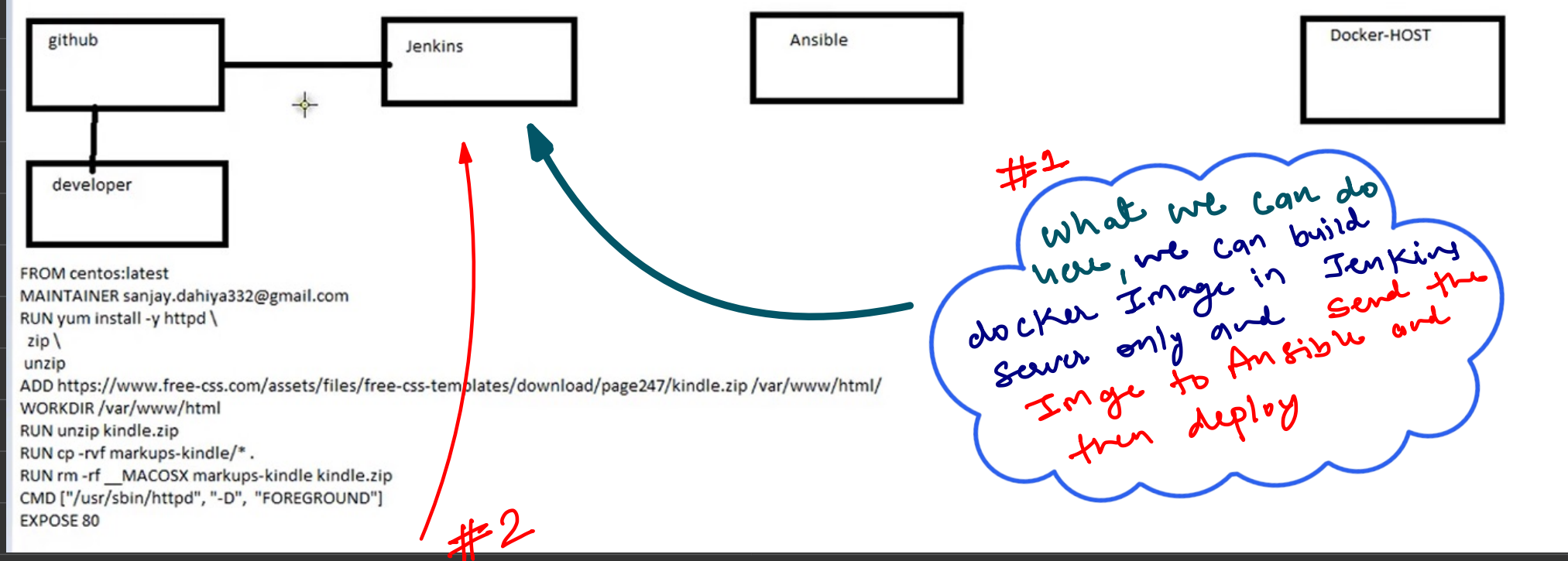
1

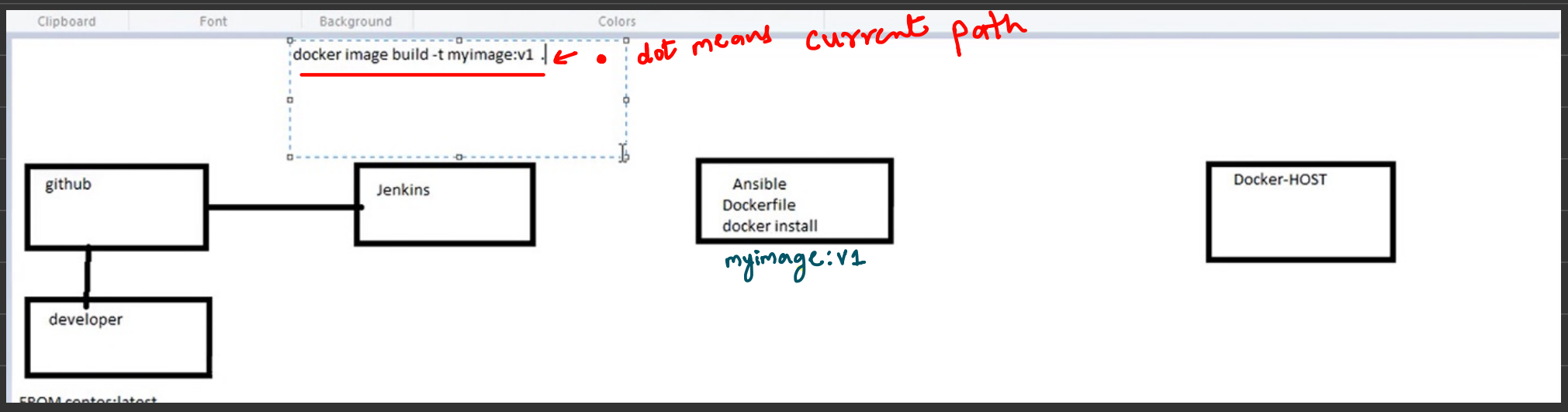



```
FROM centos:latest
MAINTAINER sanjay.dahiya332@gmail.com
RUN yum install -y httpd \
 zip \
 unzip
ADD https://www.free-css.com/assets/files/free-css-templates/download/page247/kindle.zip /var/www/html/
WORKDIR /var/www/html
RUN unzip kindle.zip
RUN cp -rvf markups-kindle/* .
RUN rm -rf __MACOSX markups-kindle kindle.zip
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
EXPOSE 80
```

Source code that we are going to use in our code



(2)



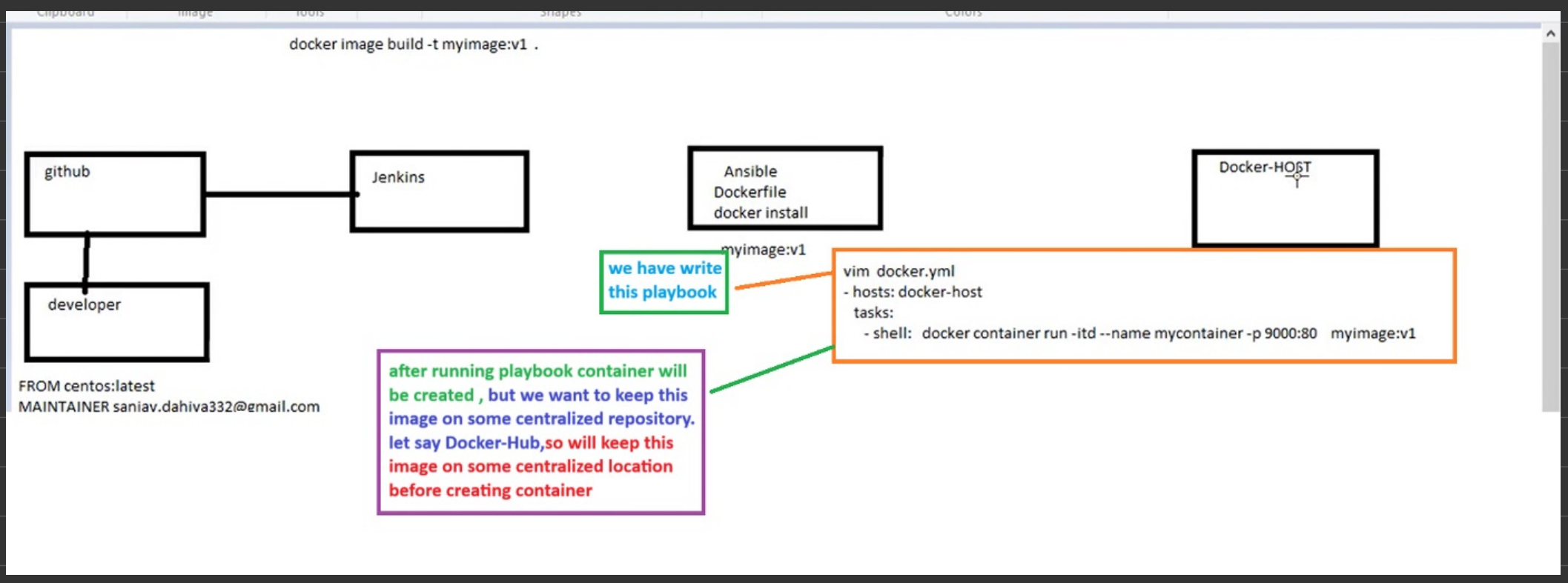
Let Say we after toiggering Job Dockerfile came to Jenkins and we have send it to Ansible server by some Command

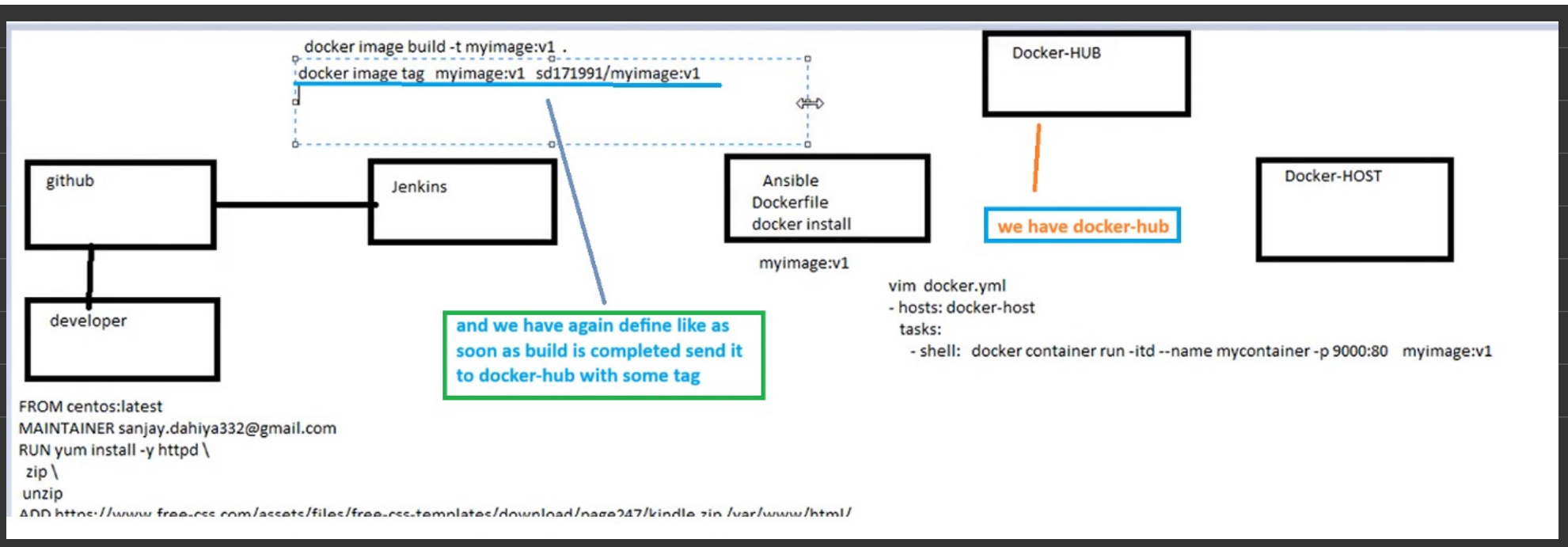
Now Dockerfile came to Ansible server and we can install Docker on Ansible Server and Build the Image over there will define some Commands over Jenkins Server to build the Dockerfile So it will build and put some Tag on Image and push'

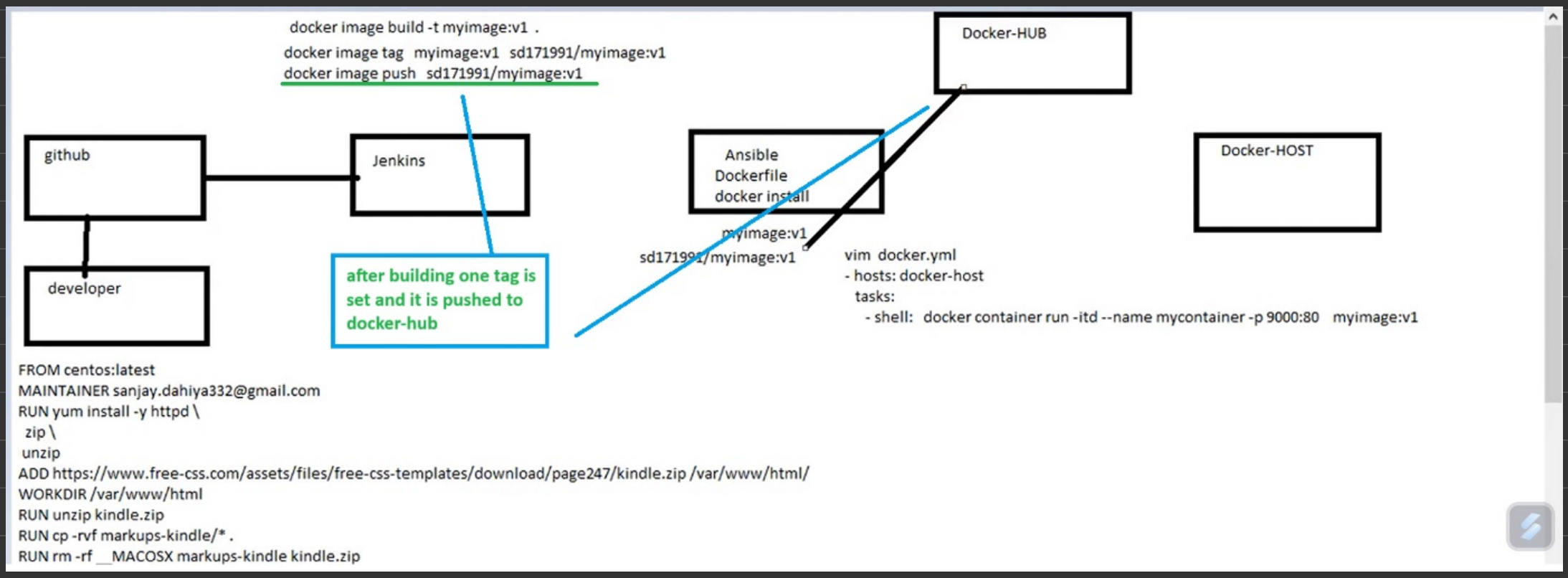
Now we will use this docker Image and build the Containers by this Image Now how to do this for this

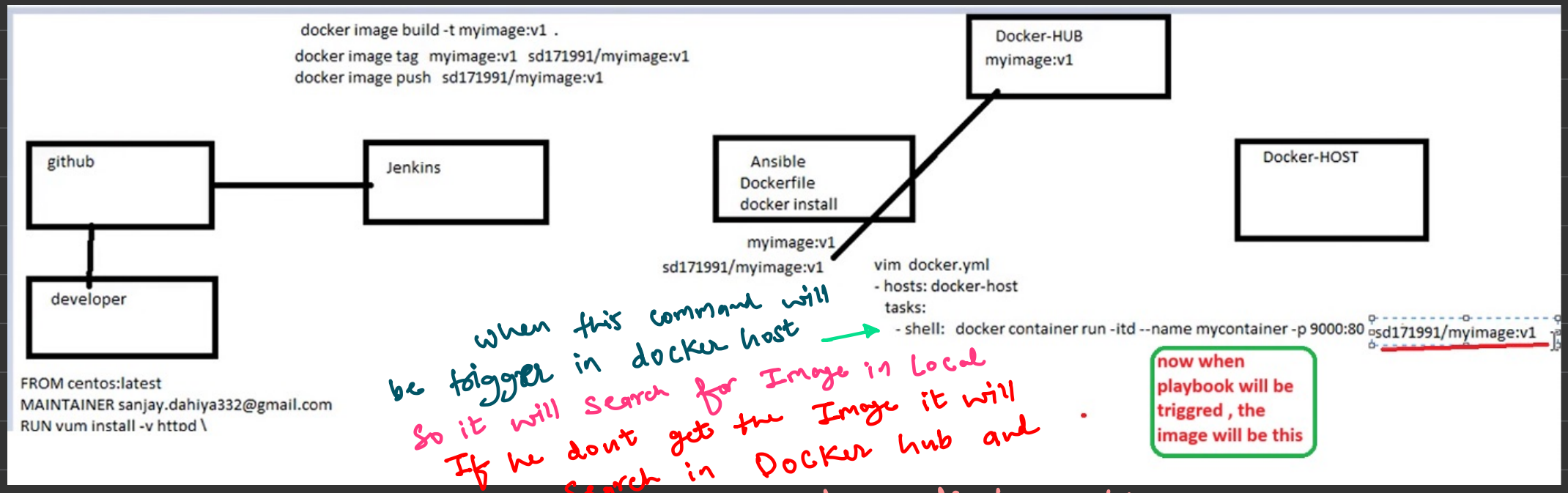
---

(3) Now how to do this, i.e Deploy Container So we have to write Playbook for this









But the above Scenario is incomplete you will face the problem in future when you will Edit the DockerFile in future.
How?

→ Let Say Developer has Change the Source-Code
Dev push the code → Jenkins will toiggue new Job as new DockerFile Came → Jenkins has given new DockerFile to Ansible for building → when DockerFile will build, it will Stop from building because, when build will Start, this Command will Execute

docker image build -t my image:v1

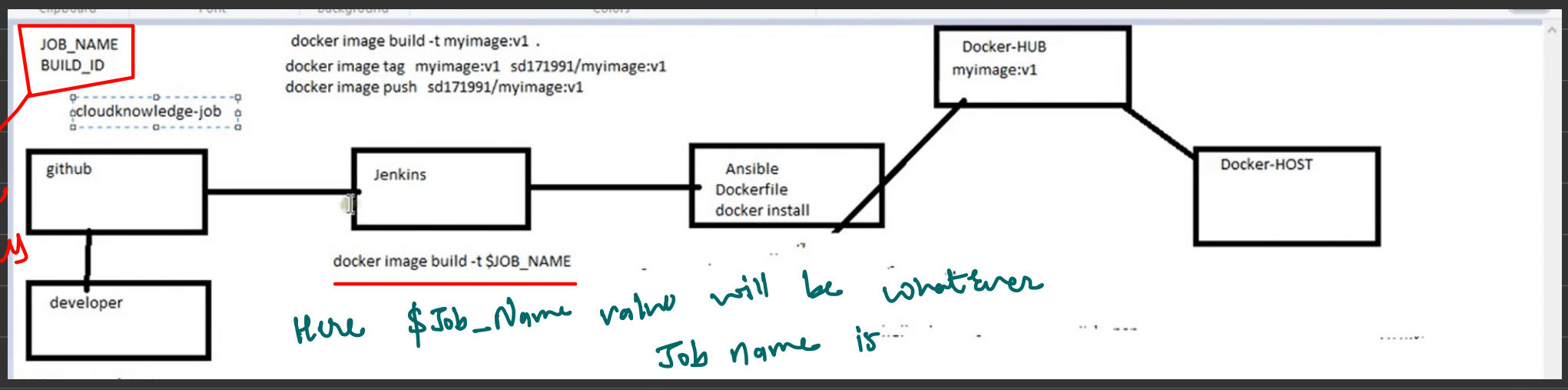
and build will fail, this build with this Image Tag is already present, So it will not create new Image with new Tag

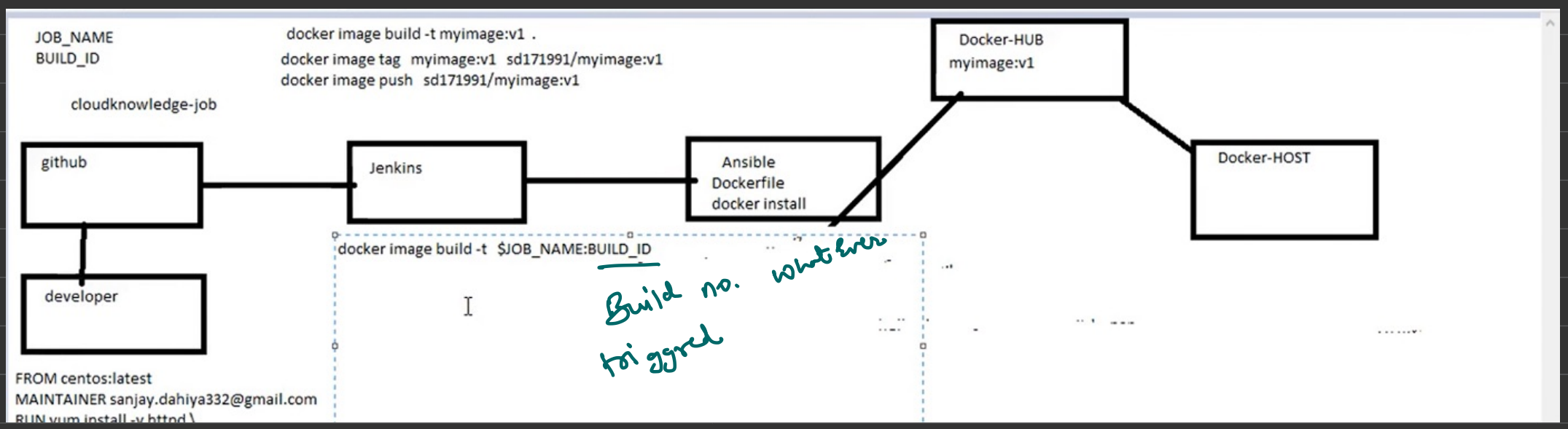
and further Commands will also not work Like Tag and push Command.

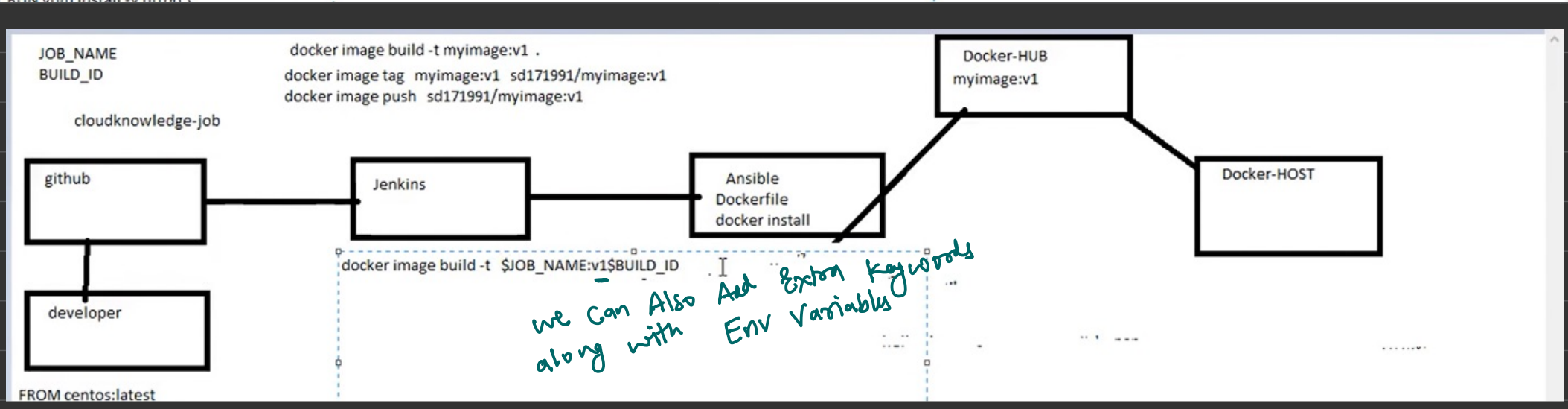
To solve this you have to manually change the name and Tag of Dockerfile in Jenkins Server
this is the problem here
And you also need to update the Ansible playbook with new tag

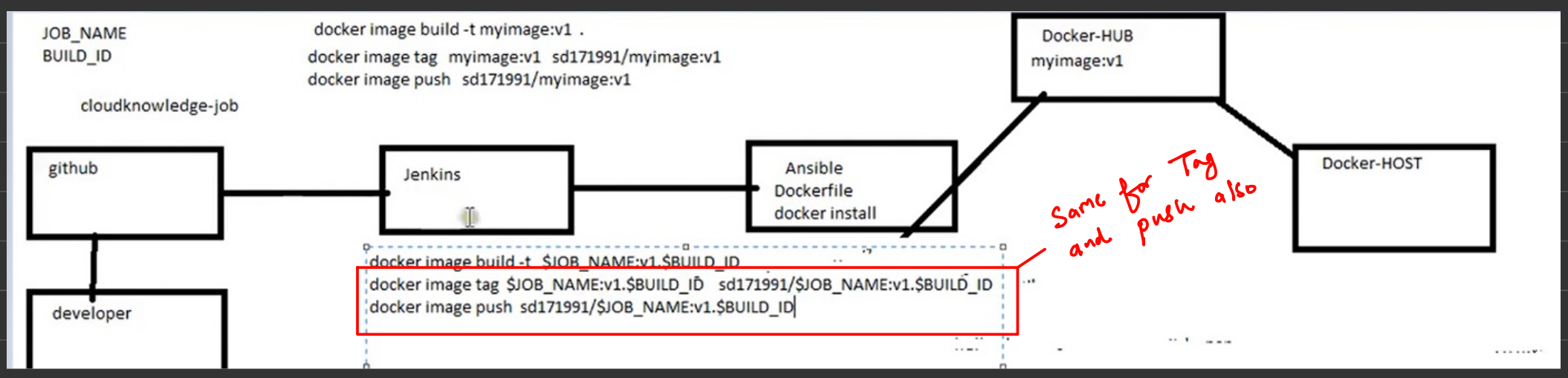
To solve this we have to use some pre-define variables in Jenkins.

4

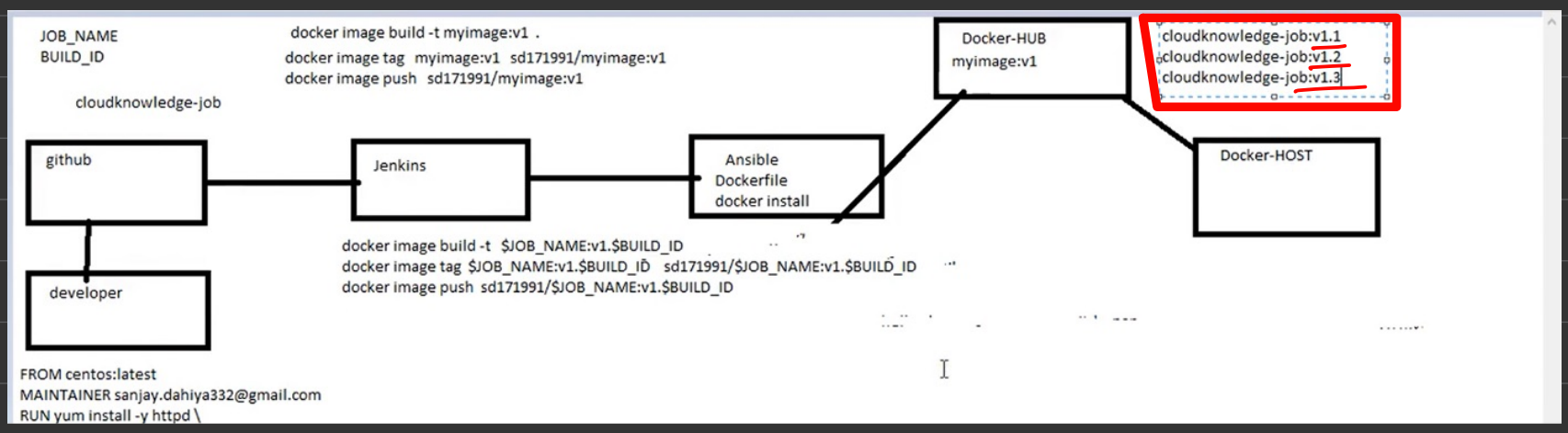




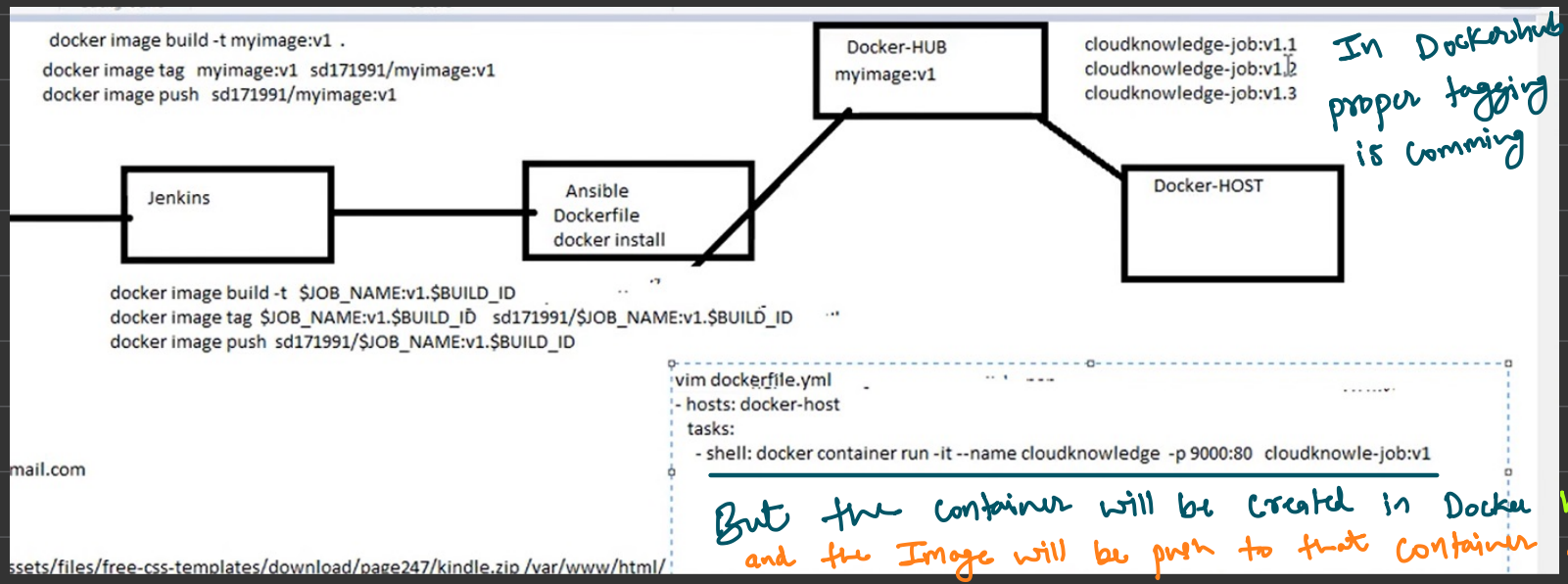




Now the version controlling will be maintain of Dockerhub



But the Ansible playbook we have written



will be V1 only which is define in playbook, that means you have to make Changes manually in playbook, Again problem Arise

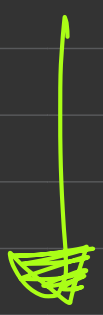
If we use this

And as we know If we dont define Tag in docker Image So it will pull Latest Image Tag

But in DockerHub there is no tag with Latest all are having custom Tag So it will say Image is not present in DockerHub

So what we want is the Image that is beign pushed to Dockerhub In that versioning Should be maintain Like Tag Image is also pushed, as well as Latest Image also be pushed

So for that we need to push some Extra Commands in Jenkins



JOB_NAME
BUILD_ID

cloudknowledge-job

docker image build -t myimage:v1 .
docker image tag myimage:v1 sd171991/myimage:v1
docker image push sd171991/myimage:v1

Docker-HUB
myimage:v1

cloudknowledge-job:v1.1 latest
cloudknowledge-job:v1.2 latest
cloudknowledge-job:v1.3 latest

github

Jenkins

Ansible
Dockerfile
docker install

Docker-HOST

developer

```
docker image build -t  $JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID  sd171991/$JOB_NAME:v1.$BUILD_ID
docker image push sd171991/$JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID  sd171991/$JOB_NAME:latest
docker image push sd171991/$JOB_NAME:latest
```

Now here 2 tags will be pushed specific tag as well as Latest tag will be pushed

```
vim docker.yml
- hosts:  docker- hosts
  tasks:
   - shell: docker container run -it -- name cloudknowledge  -p 9000:80
sd171991/cloudknowledge
```

```
FROM centos:latest
MAINTAINER sanjay.dahiya332@gmail.com
RUN yum install -y httpd \
 zip \
 unzip
```

JOB_NAME
BUILD_ID

cloudknowledge-job

docker image build -t myimage:v1 .
docker image tag myimage:v1 sd171991/myimage:v1
docker image push sd171991/myimage:v1

Docker-HUB
myimage:v1

cloudknowledge-job:v1.1 latest
cloudknowledge-job:v1.2 latest
cloudknowledge-job:v1.3 latest

github

Jenkins

Ansible
Dockerfile
docker install

Docker-HOST

developer

docker image build -t $JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID sd171991/$JOB_NAME:v1.$BUILD_ID
docker image push sd171991/$JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID sd171991/$JOB_NAME:latest
docker image push sd171991/$JOB_NAME:latest

Now when playbook will Run it will automatically take the Latest Image tag

```
vim docker.yml
- hosts:  docker- hosts
  tasks:
   - shell: docker container run -it -- name cloudknowledge  -p 9000:80
sd171991/cloudknowledge
```

FROM centos:latest
MAINTAINER sanjay.dahiya332@gmail.com
RUN yum install -y httpd \
 zip \
 unzip

5

Now when job will trigger intermittently there will be lots of images present in the Ansible server this might lead to storage issue , so we also need to delete the older images form Ansible-sever.

JOB_NAME
BUILD_ID

cloudknowledge-job

docker image build -t myimage:v1 .
docker image tag myimage:v1 sd171991/myimage:v1
docker image push sd171991/myimage:v1

Docker-HUB
myimage:v1

cloudknowledge-job:v1.1 latest
cloudknowledge-job:v1.2 latest
cloudknowledge-job:v1.3 latest

github

Jenkins

Ansible
Dockerfile
docker install

Docker-HOST

developer

docker image build -t $JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID sd171991/$JOB_NAME:v1.$BUILD_ID
docker image push sd171991/$JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID sd171991/$JOB_NAME:latest
docker image push sd171991/$JOB_NAME:latest

```
docker image rmi  $JOB_NAME:v1.$BUILD_ID  sd171991/$JOB_NAME:v1.$BUILD_ID sd171991/
$JOB_NAME:latest
```

by this image will be removed from Ansible-server , after it get pushed to repository i.e Docker-hub

FROM centos:latest
MAINTAINER sanjay.dahiya332@gmail.com
RUN yum install -y httpd \
 zip \
 unzip

```
ADD https://www.free-css.com/assets/files/free-css-templates/download/page247/kindle.zip /var/www/html/
WORKDIR /var/www/html
RUN unzip kindle.zip
```

But still issue is not resolved yet, the problem you will face here is

we have created playbook to Run in Docker Host.

```
vim docker.yml
- hosts:  docker-host
  tasks:
   - shell: docker container run -itd -- name cloudknowledge  -p 9000:80
sd171991/cloudknowledge
```

when your playbook Run it will pull the latest Image and build the Container for you

But Let Say you have updated the code

↓

Code push to Github

↓

Image build with new ID and also with Latest Tag

↓

New Image push to DockerHub

↓

Old container will be removed and new Container will be Created with new Image

But when playbook will be trigger again for new code, and when it will try to create another Container, It will not able to pull the new code Image / Latest Image from Docker-hub Because

The Bydefault behaviour of Docker is, it will try to search Image on its Localhost and in Localhost our existing/old Latest Image is present so it will not pull from DockerHub i.e it will not move to DockerHub to pull Latest Commit Image

For this we have to do some Extra modification in playbook

Extra we have to do is, remove the existing Image, so it will pull Latest Image

```
vim docker-yml
- hosts: docker-host
  tasks:
    - name:  stop container
      shell:  docker container stop cloudknowledge
    - name: remove continer
      shell: docker container rm cloudknowledge
```

here first container will stop
then container will be removed

```
vim docker-yml
- hosts: docker-host
  tasks:
    - name:  stop container
      shell:  docker container stop cloudknowledge
    - name: remove continer
      shell: docker container rm cloudknowledge
    - name: remove docker image
      shell: docker image rmi sd171991/cloudknowledge
```

also remove the image

```
vim docker-yml
- hosts: docker-host
  tasks:
   - name:  stop container
     shell:  docker container stop cloudknowledge
   - name: remove continer
     shell: docker container rm cloudknowledge
   - name: remove docker image
     shell: docker image rmi sd171991/cloudknowledge
  - name: create new container
    shell: docker container run -itd --name cloudknowledge-container -p 9000:80
sd171991/cloudknowlege
```

new container will be created

***** Practical-Part *****

1

jenkins	i-02f4f8d212897b410	Stopped	t2.micro	–	No alarms	+	ap-south-1a	–

We have ready our Jenkins server
Install Jenkins and git

2

ansible	i-0f8fe46046edefd5d	Stopped	t2.micro	–	No alarms	+	ap-south-1a	

We have install ansible and
We also docker package in ansible server for building docker image
We also have added the docker-server ip in host file of ansible-server
And we have also added key(key-based authentication) between ansible – docker server
And we have also added key(key-based authentication) between Jenkins – ansible server

3

we have added dockerfile in github

Rizwan00792 / jenkins

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

master

Go to file t

Dockerfile

README.md

jenkins / Dockerfile

Rizwan00792 Update Dockerfile b4d1a7a · 8 hours ago History

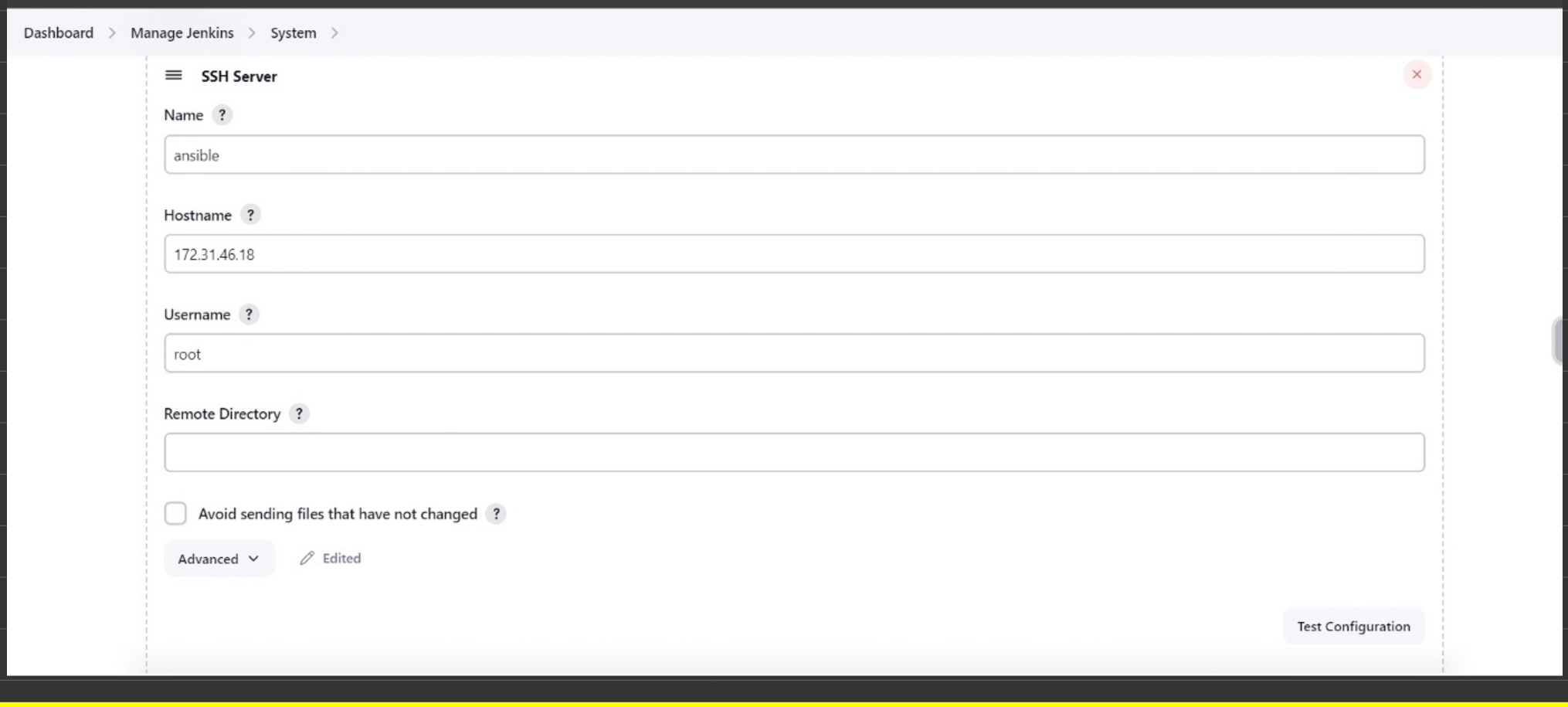
Code Blame 12 lines (12 loc) · 356 Bytes Code 55% faster with GitHub Copilot Raw

centos is not going to work it has some type error we have to fix before

```
1  FROM centos:latest
2  MAINTAINER khanrizwanrk147@gmail
3  RUN YUM install -y httpd \
4  zip \
5  unzip
6  ADD https://www.free-css.com/assets/files/free-css-templates/download/page296/oxer.zip /var/www/html
7  WORKDIR /var/www/html
8  RUN unzip oxer.zip
9  RUN cp -rvf markups-oxer/*
10 RUN rm -rf _MACOSX markups-oxer oxer.zip
11 CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
12 EXPOSE 80
```

4

we have added our ansible and Jenkins server in Manage Jenkins



5

Adding build steps

The first task will perform on Jenkins server

The second task will perform on Ansible server

Built in environment variables

Jenkins provides a set of environment variables. You can also define your own. Here is a list of built in environment variables:

- `BUILD_NUMBER` - The current build number. For example "153"
- `BUILD_ID` - The current build id. For example "2018-08-22_23-59-59"
- `BUILD_DISPLAY_NAME` - The name of the current build. For example "#153".
- `JOB_NAME` - Name of the project of this build. For example "foo"
- `BUILD_TAG` - String of "jenkins-${JOB_NAME}-${BUILD_NUMBER}".
- `EXECUTOR_NUMBER` - The unique number that identifies the current executor.
- `NODE_NAME` - Name of the "slave" or "master". For example "linux".
- `NODE_LABELS` - Whitespace-separated list of labels that the node is assigned.
- `WORKSPACE` - Absolute path of the build as a workspace.
- `JENKINS_HOME` - Absolute path on the master node for Jenkins to store data.
- `JENKINS_URL` - URL of Jenkins. For example *http://server:port/jenkins/*
- `BUILD_URL` - Full URL of this build. For example *http://server:port/jenkins/job/foo/15/*
- `JOB_URL` - Full URL of this job. For example *http://server:port/jenkins/job/foo/*

Extra Knowledge

Send files or execute commands over SSH ?

SSH Publishers

SSH Server

Name ?

ansible

Advanced

Exec command ?

```
cd /opt
docker image build -t $JOB_NAME:v1.$BUILD .
```

Current working directory where Dockerfile is present

in ansible it will got to /opt
and docker image will be build

docker hub | Explore | Repositories | Organizations | Search Docker Hub

Dockerhub Id

74992692 | Search by repository name | All Content | Create repository

74992692 / **docker-project**
Contains: No content | Last pushed: 9 hours ago — Inactive ☆ 0 ⬇ 0 Public

Exec command ?

```
cd /opt
docker image build -t $JOB_NAME:v1.$BUILD_ID .
docker image tag $JOB_NAME:v1.$BUILD_ID 74992692/$JOB_NAME:V1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID 74992692/$JOB_NAME:latest
docker image push 74992692/$JOB_NAME:V1.$BUILD_ID
docker image push 74992692/$JOB_NAME:latest
docker image rmi $JOB_NAME:v1.$BUILD_ID 74992692/$JOB_NAME:V1.$BUILD_ID  74992692/$JOB_NAME:latest
```

$Build_ID.

dockerhub id

it will build image with some tag

it will also build the image with latest tag

Exec command ?

```
cd /opt
docker image build -t $JOB_NAME:v1.$BUILD .
docker image tag $JOB_NAME:v1.$BUILD_ID 74992692/$JOB_NAME:V1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID 74992692/$JOB_NAME:latest
docker image push 74992692/$JOB_NAME:V1.$BUILD_ID
docker image push 74992692/$JOB_NAME:latest
docker image rmi $JOB_NAME:v1.$BUILD_ID 74992692/$JOB_NAME:V1.$BUILD_ID  74992692/$JOB_NAME:latest
```

here it will push the image with specific and latest tag

here it will remove the specific and latest image after push to docker hub

Now the problem will face it when image will be pushing to dockerhub it will ask for id and password of dockerhub

So we have to login to dockerhub on ansible server

```
[root@ansible ~]# dockerhub login
-bash: dockerhub: command not found
[root@ansible ~]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create
Username: 74992692
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ansible ~]#
```

we have logged into Dockerhub in Ansible

Now image will be pushed to dockerhub

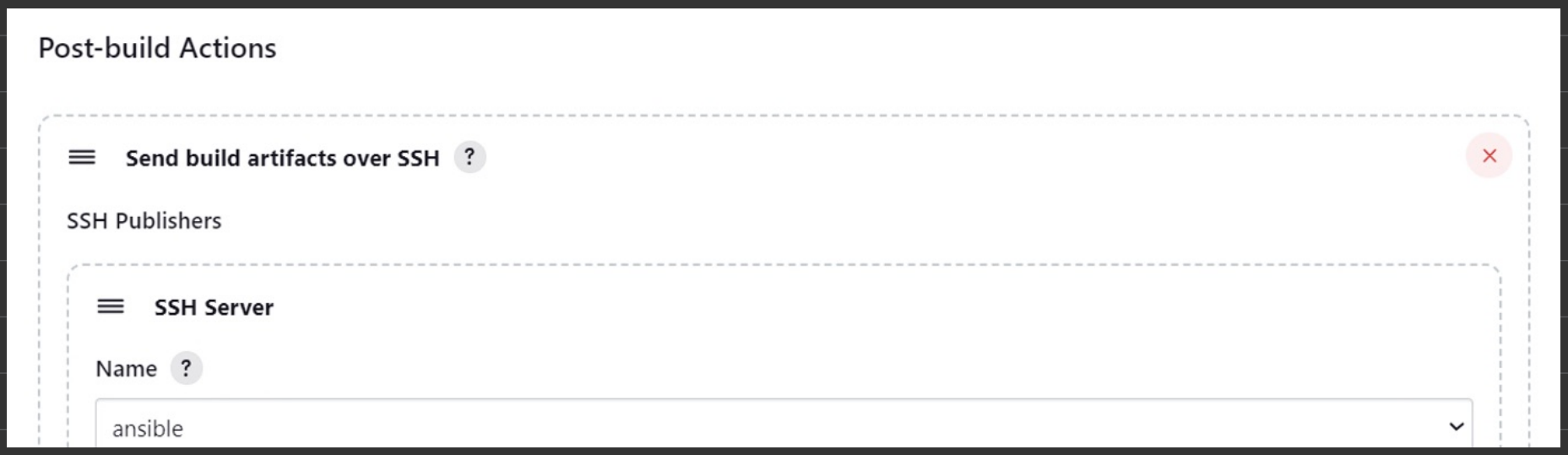
but we also need to create the container on docker-host , for that we need to create a playbook on Ansible server

```
- hosts: all
  tasks:
    - name: create container
      shell: docker container run -itd --name rizwan-container -p 9000:80 74992692/docker-project1
```

Image Name

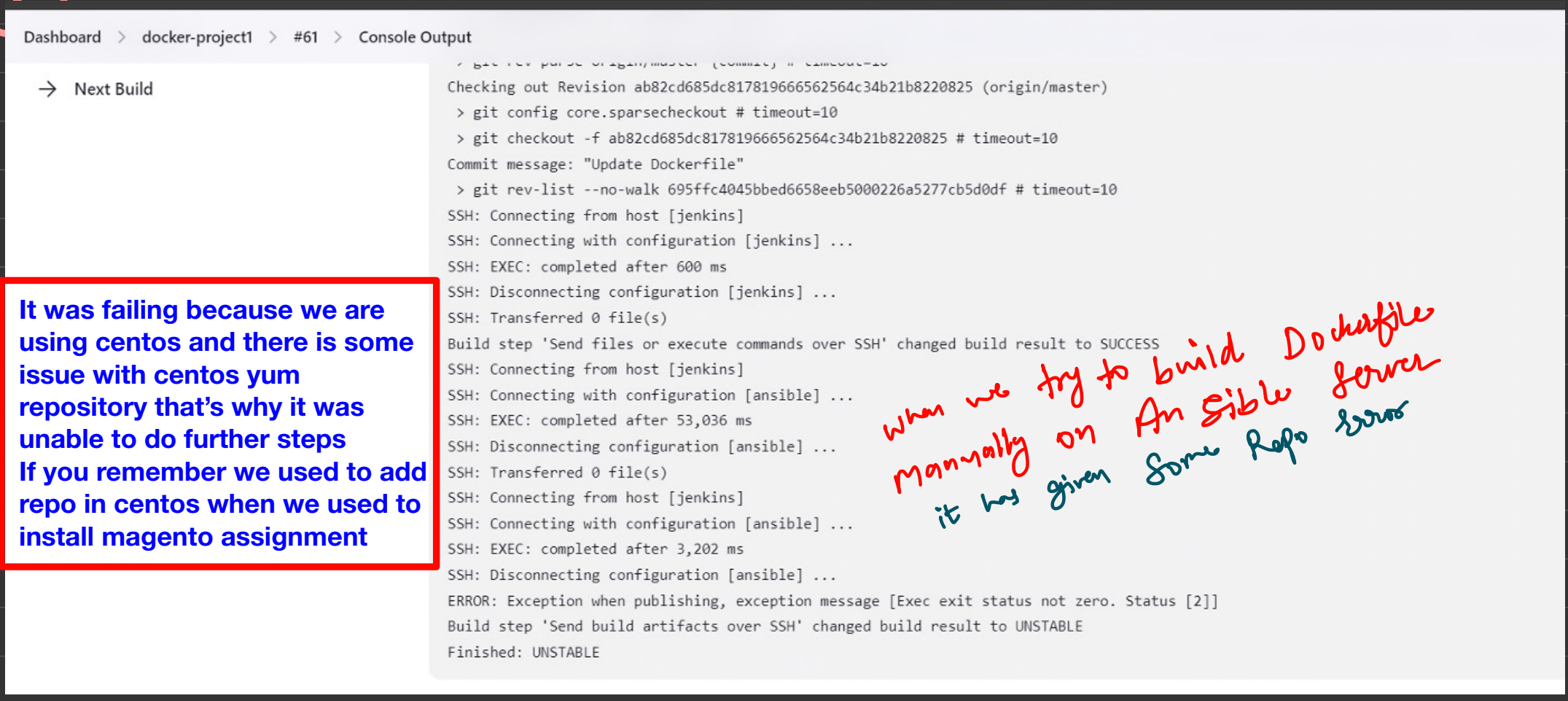
6

Now we have added post build action



7

Now we have trigger the build action



It was failing because we are using centos and there is some issue with centos yum repository that's why it was unable to do further steps
If you remember we used to add repo in centos when we used to install magento assignment

when we try to build Dockerfile manually on Ansible server it has given some Repo error

Now we have again trigger with different
i.e
rockylinux:9.3.20231119
amazonlinux:latest

Code | Blame

Raw

```
1   FROM rockylinux:9.3.20231119
2   MAINTAINER khanrizwanrk147@gmail.com
3   RUN yum install -y httpd \
4   zip \
5   unzip
6   ADD https://www.free-css.com/assets/files/free-css-templates/download/page296/oxer.zip /var/www/
7   WORKDIR /var/www/html
8   RUN unzip oxer.zip
9   RUN cp -rvf oxer-html/* .
10  RUN rm -rf _MACOSX oxer-html oxer.zip
11  CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
12  EXPOSE 80
```

dot means current directory me paste karo

Note: Yaha per bas 9 wali line hai usko dekha hai ek bar wget aur unzip karke dekh lena jab url change karoge to konsa folder ban rha

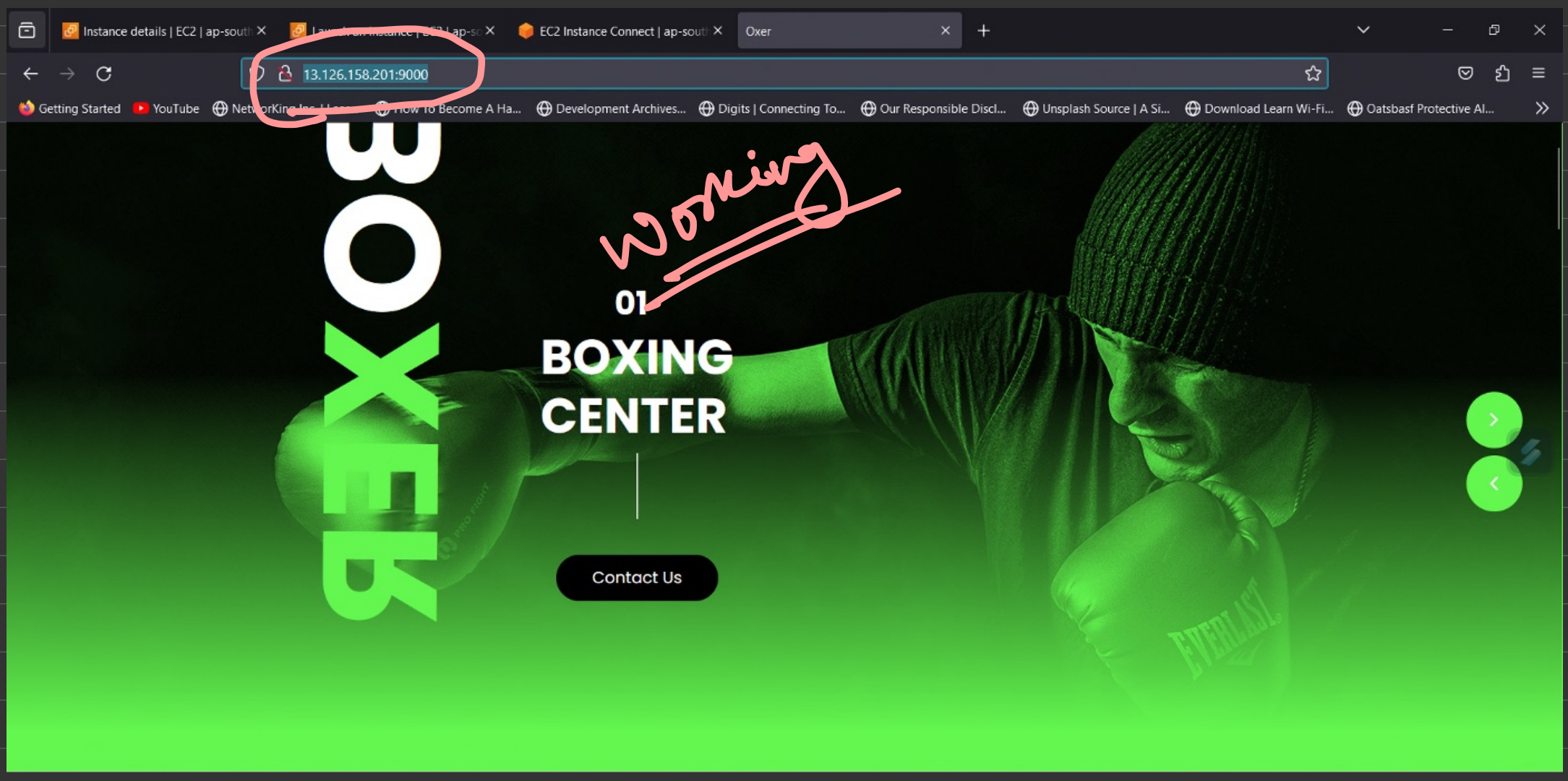
Dashboard > docker-project1 > #63 > Console Output

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#63'
- Git Build Data
- Previous Build

Console Output

```
Started by user rizwan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/docker-project1
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/docker-project1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Rizwan00792/jenkins.git # timeout=10
Fetching upstream changes from https://github.com/Rizwan00792/jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/Rizwan00792/jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse origin/master^{commit} # timeout=10
Checking out Revision 94234fd43a0a3cc87262a86ca12dabc094b451a1 (origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 94234fd43a0a3cc87262a86ca12dabc094b451a1 # timeout=10
Commit message: "Update Dockerfile"
 > git rev-list --no-walk f6e5d03e17fa5f4c2df603e47f7dfeea44f3be59 # timeout=10
SSH: Connecting from host [jenkins]
SSH: Connecting with configuration [jenkins] ...
SSH: EXEC: completed after 600 ms
SSH: Disconnecting configuration [jenkins] ...
SSH: Transferred 0 file(s)
Build step 'Send files or execute commands over SSH' changed build result to SUCCESS
SSH: Connecting from host [jenkins]
SSH: Connecting with configuration [ansible] ...
SSH: EXEC: completed after 11,208 ms
SSH: Disconnecting configuration [ansible] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins]
SSH: Connecting with configuration [ansible] ...
SSH: EXEC: completed after 15,210 ms
SSH: Disconnecting configuration [ansible] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS
```

Sucess


Instance details | EC2 | ap-south
EC2 Instance Connect | ap-south
Oxer
13.126.158.201:9000
Getting Started
YouTube
Development Archives...
Digits | Connecting To...
Our Responsible Discl...
Unsplash Source | A Si...
Download Learn Wi-Fi...
Oatsbasf Protective Al...
BOXER
Working
01
BOXING
CENTER
Contact Us