DOCKERHUB AND IT'S WORKING

Prerequisite: https://bansalkushagra.medium.com/docker-port-expose-and-publish-3470f4b49ccf

How to push docker image in docker hub

Step-1:

Create a container say ubuntu

Command:

```
Docker run –it ubuntu /bin/bash
ls
```

You will see many files and directory already exists. Create more files here.

Command:

```
Touch file file2 test test2 xyz
```

Also, additionally, install the apache server and files inside tmp/ directory using the command

```
apt-get install apache2 –y
```

To see an HTML webpage go to “cd var/www/ ”

Go to tmp directory

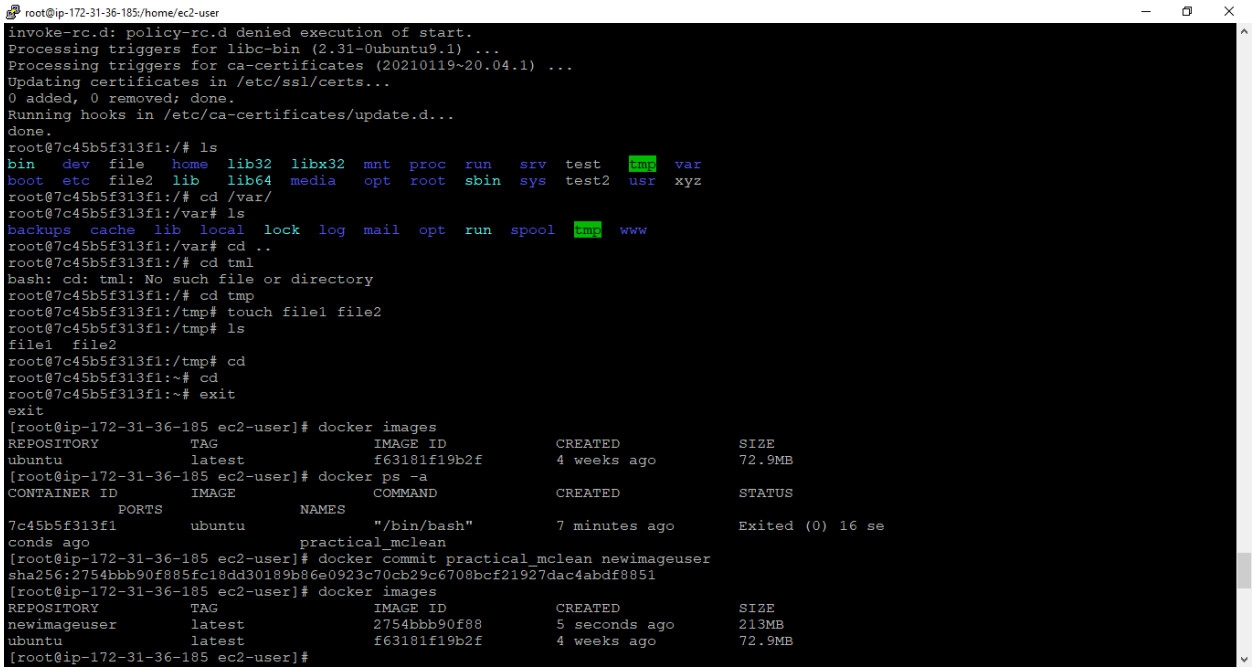
```
cd /tmp
touch file1 file2
ls
```

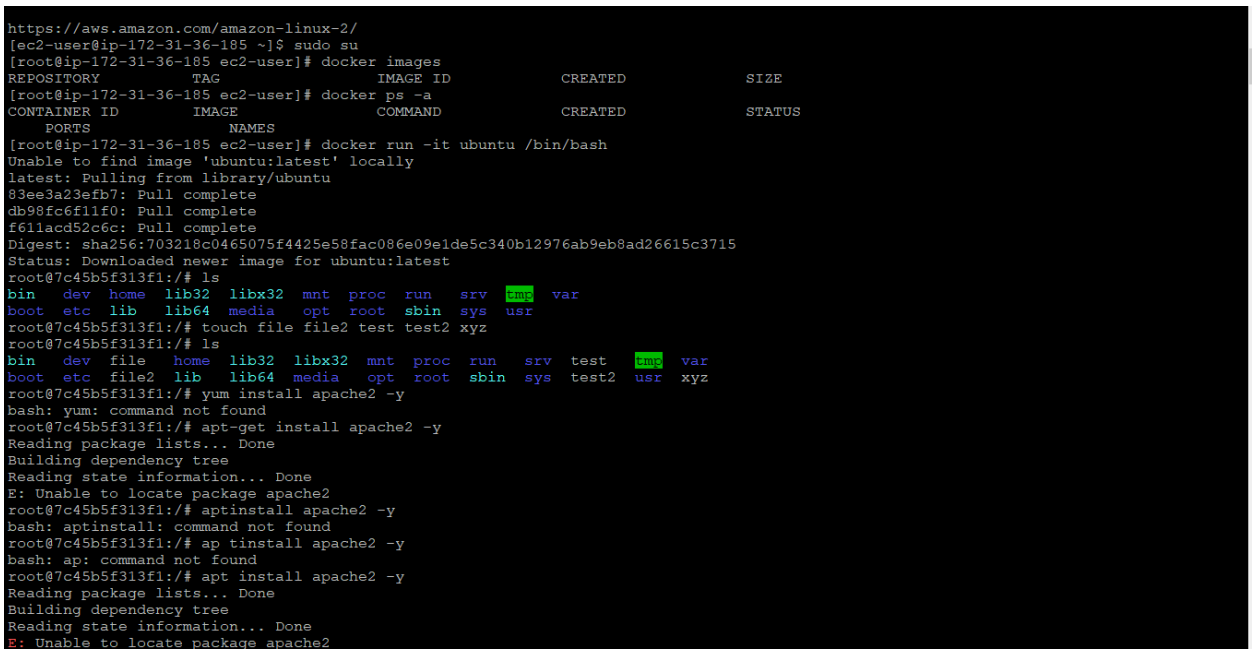
Now, create an image of this updated container.

Command:

```
docker commit <Container_name> <new_container_name>
docker commit practical_mclean newimageuser
docker images
```

You will see the image is created with a name newimageuser.





Step-2: login into the docker hub account

Command:

Docker login

Add credentials when asked carefully.

```
[root@ip-172-31-36-185 ec2-user]# 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 one.
Username: kush8755
Password:
Error response from daemon: Get https://registry-1.docker.io/v2/: unauthorized: incorrect userna
me or password
[root@ip-172-31-36-185 ec2-user]# 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 one.
Username: kush8755
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@ip-172-31-36-185 ec2-user]#
```

Step-3:

Now, push it into the docker hub using the following command

Command:

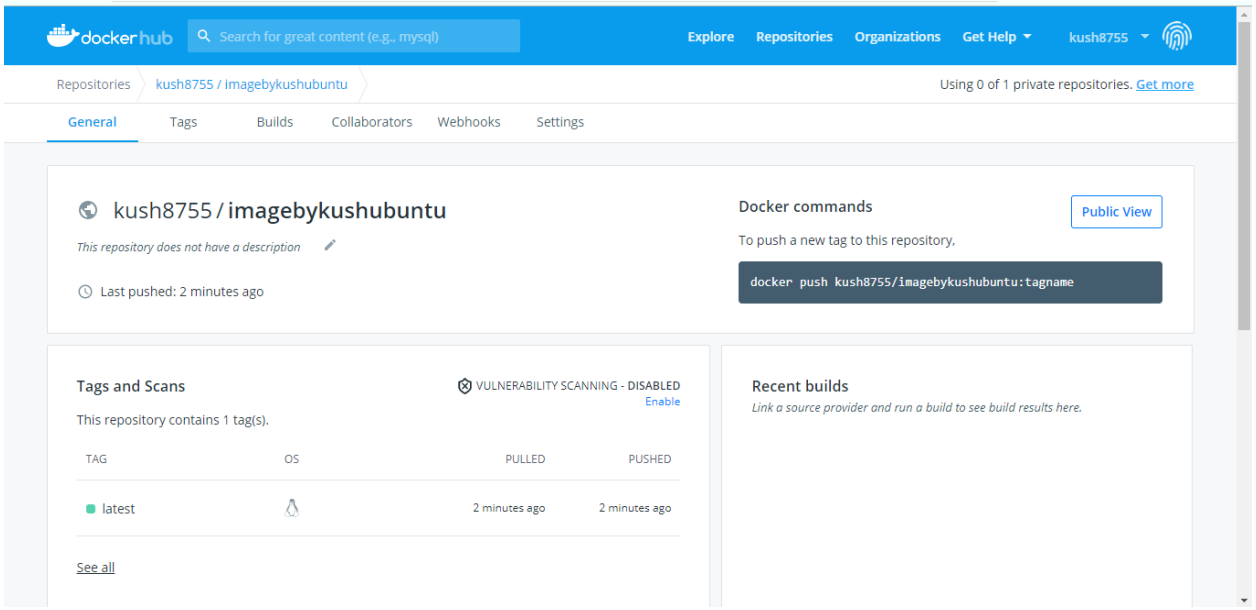
Docker tag <image_name_to_be_pushed> <New_i mage_name>

Docker tag newimageuser kush8755/imagebykushubuntu

Docker push kush8755/imagebykushubuntu

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-36-185 ~]$ sudo su
[root@ip-172-31-36-185 ec2-user]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
newimageuser        latest              2754bbb90f88        14 minutes ago      213MB
ubuntu              latest              f63181f19b2f        4 weeks ago         72.9MB
[root@ip-172-31-36-185 ec2-user]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
          PORTS               NAMES
7c45b5f313f1        ubuntu              "/bin/bash"         22 minutes ago      Exited (0) 15 mi
nutes ago                       practical_mclean
[root@ip-172-31-36-185 ec2-user]# docker tag newimageuser kush8755/imagebykushubuntu
[root@ip-172-31-36-185 ec2-user]# docker tag newimageuser kush8755/imagebykushubuntu
[root@ip-172-31-36-185 ec2-user]# docker tag ubuntu kush8755/imagebykushubuntu
[root@ip-172-31-36-185 ec2-user]# ls
[root@ip-172-31-36-185 ec2-user]# docker login
Authenticating with existing credentials...
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@ip-172-31-36-185 ec2-user]# docker tag newimageuser kush8755/imagebykushubuntu
[root@ip-172-31-36-185 ec2-user]#
[root@ip-172-31-36-185 ec2-user]# docker puch kush8755/imagebykushubuntu
docker: 'puch' is not a docker command.
See 'docker --help'
[root@ip-172-31-36-185 ec2-user]# docker push kush8755/imagebykushubuntu
The push refers to repository [docker.io/kush8755/imagebykushubuntu]
4ec026c52b6d: Pushed
02473afd360b: Mounted from library/ubuntu
dbf2c0f42a39: Mounted from library/ubuntu
9f32931c9d28: Mounted from library/ubuntu
latest: digest: sha256:3a600137ade84cb6b1f8eecde98460f8df73d3030d59938b4abfa0af371cd9e5 size: 11
55
[root@ip-172-31-36-185 ec2-user]# docker ps -a
```



Step-4: To verify the software we installed, The files we created will come pre-install when another user uses our image.

Create an AWS ec2 instance [see the previous blog how to create]

Command-1:

sudo su

Redirect to root directory

Command-2:

Yum update –y

Update the Linux machine

Command-3:

Yum install docker –y

Install docker on Linux machine

Command-4:

```
Service docker start
```

Start the Docker Daemon

Command-5:

```
docker images
```

Show the image available

Command-6:

```
docker ps –a
```

Show containers status

Command-7:

```
docker pull kush8755/imagebykushubuntu
```

Pull the image of the name “imagebykushubuntu” from the docker hub of username “kush8755”.

Command-8:

```
docker images
```

Shows the image available, now it shows the above image we pull.

Command-9: docker run –it –name kushcontainer kush8755/imagebykushubuntu /bin/bash

Create a container name kushcontainer using the image we pull.

Command-10:

```
cd tmp
ls
```

It will verify that folder and software conf files are pre-installed in the new user account

Command-11:

cd var/www/

ls

It will verify that folder and software conf files are pre-installed in the new user account.