**DOCKER ASSIGNMENT 2**

**1.** Create a docker container using Ubuntu image with a meaningful name & set the

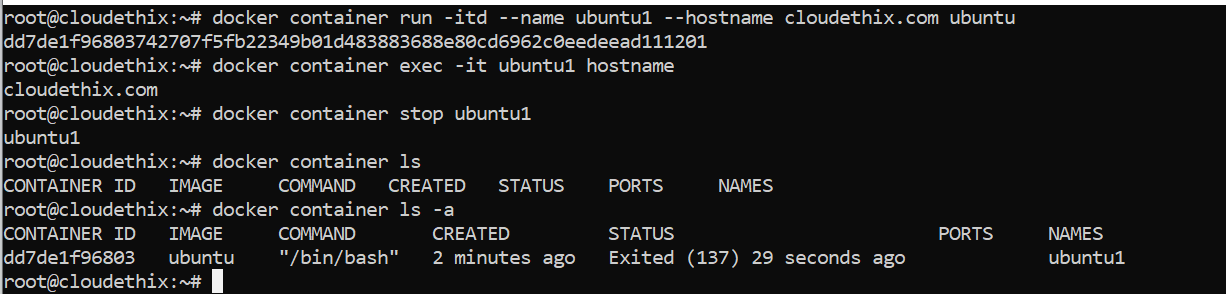
hostname to the container cloudethix.com. Then stop or kill the container and check its

exit code and check the log of the container for more details.

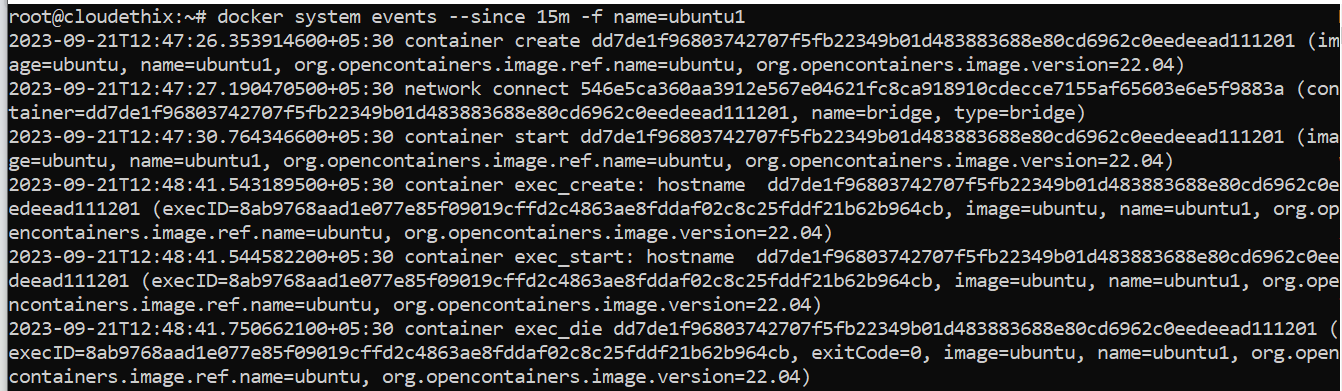
**Answer:**

 We have created an ubuntu container and changed its hostname at the run time.

Then we stopped the container.



 system logs.



2. Create and start an Ubuntu container with a random name. Then rename the container

name to the meaningful name Ubuntu01. Then run another container named Ubuntu02

and then check the hostname of both the containers. Then pause and unpause the

Ubuntu02 container and stop , start , restart Ubuntu01 container. Also check stat and

system events of both the containers and then kill and delete the same. Also make sure

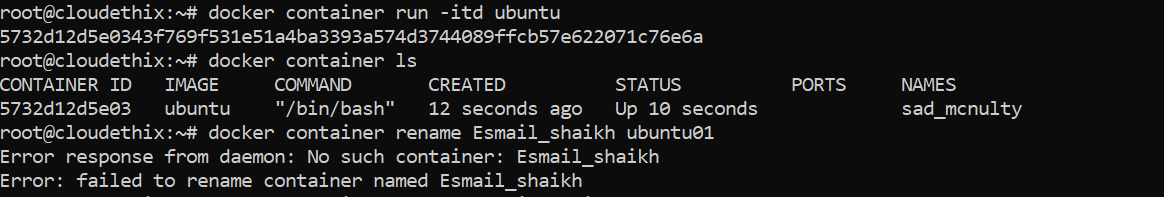
directories created by containers should be deleted

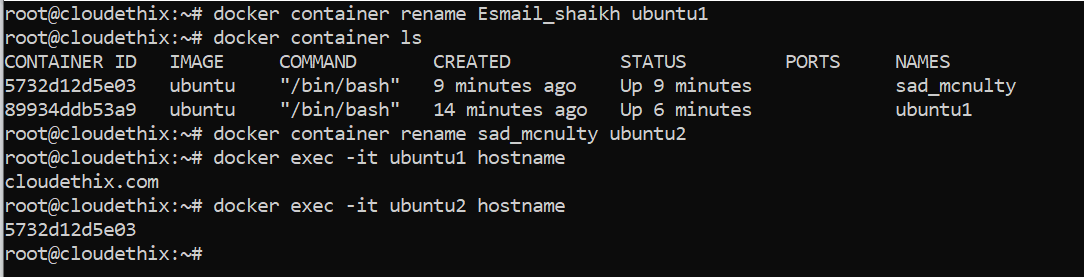
**Answer:**

 We have run a container without name and then renamed it to ubuntu-01 as shown

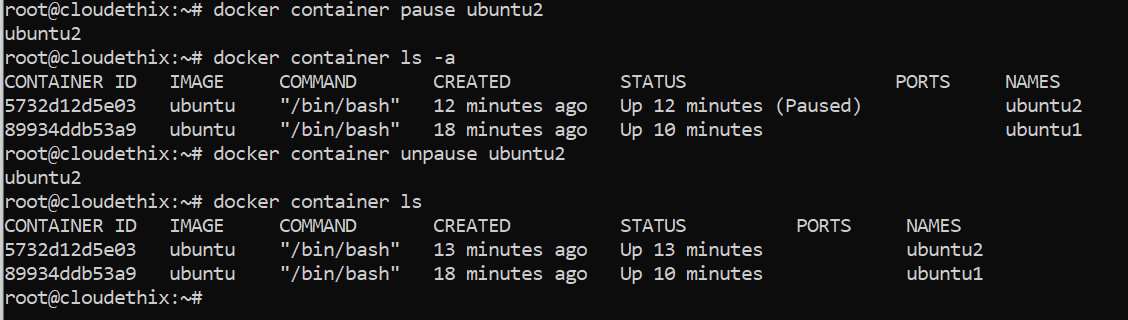
below. Then we have added one more container ubuntu-02 and checked hostnames of

both.

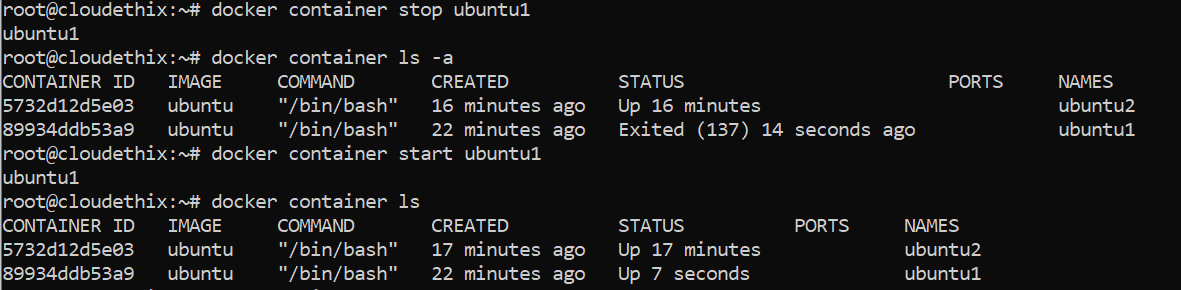


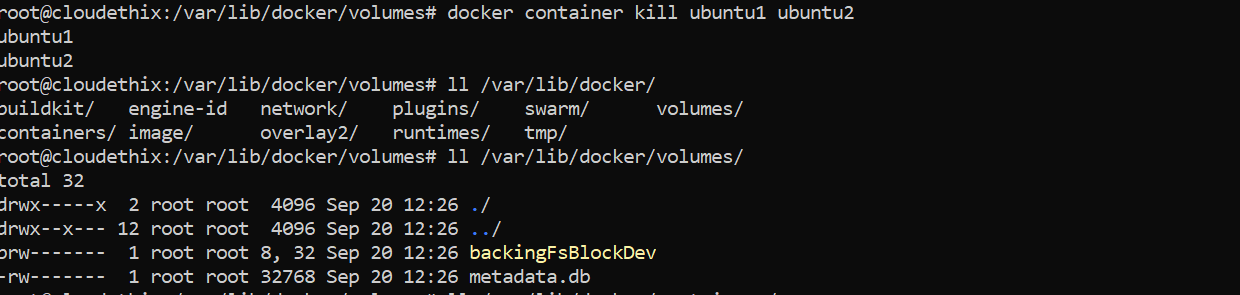


.. we have pasued and unpaused the ubuntu-02 container.



 We have stopped, started and restarted ubuntu-01. Then we will kill both containers





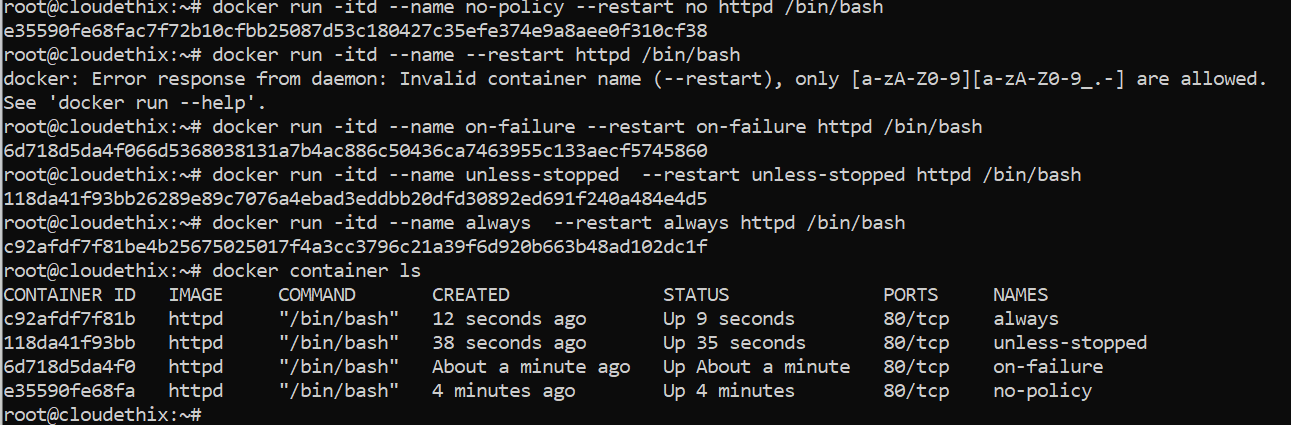
3. Run four HTTPD docker containers with meaningful names to each container and then

apply restart policies NO , On-Failure, Always and Unless-Stopped on these 4 containers

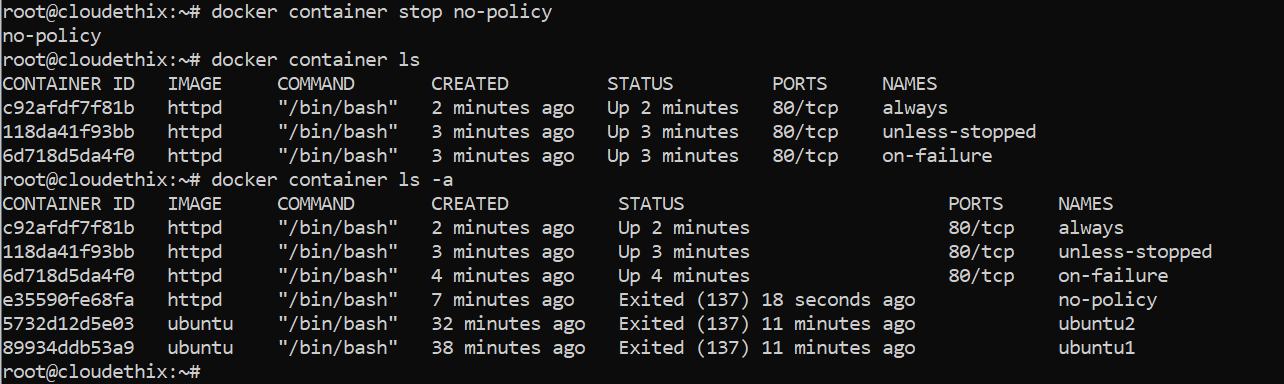
respectively. Once applied, prove that restart policies are working.

Answer

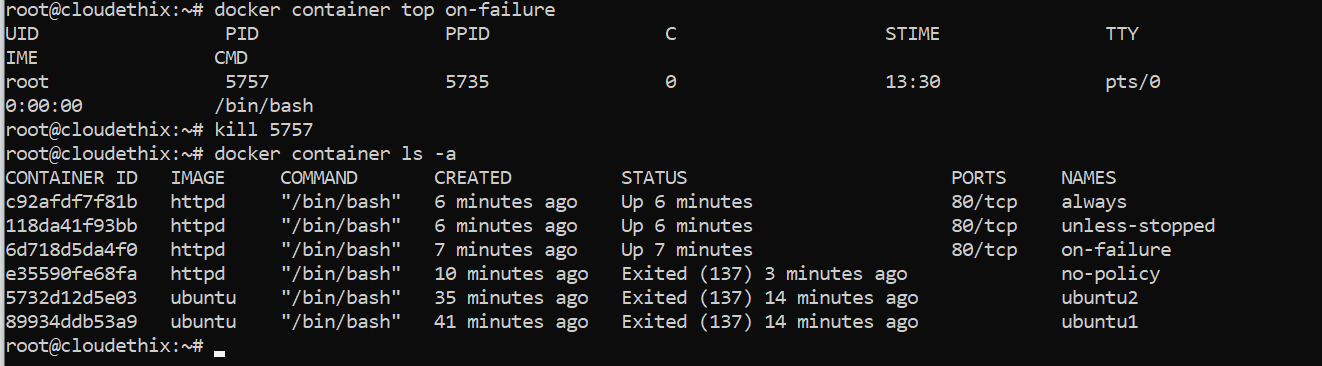
 We ran four containers with the names and restart policies respectively



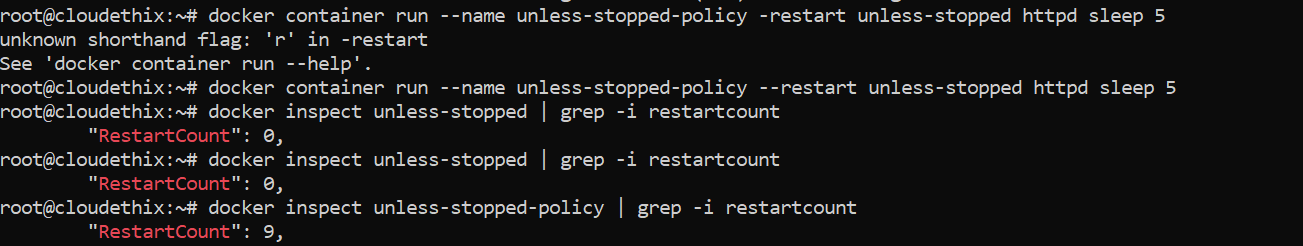
 We tested **no\_policy** container, and it did not start after restart



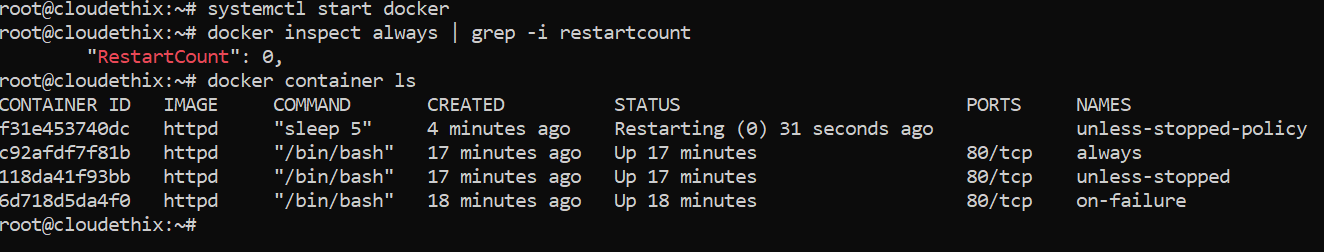
. We tested on failure container by killng its pid and restart it



. Now we tested unless stopped container by giving the command sleep every 5 seconds and inspect it



. For always restart the docker engine and check its still restarted or not



4. Run an NGINX container with meaningful and expose the container on host port 80.

Also create an index.html file with data "Hello there , Let's be the Team CloudEthiX" .

Then copy the index.html file to the container under /usr/share/nginx/html/ location.

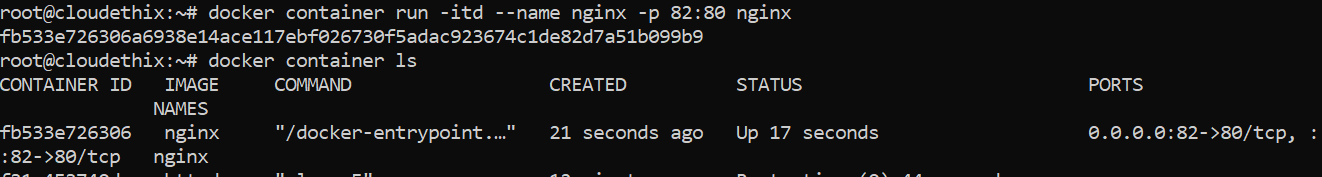
Once copied try to access the container in the browser and check if the webpage is getting

displayed properly

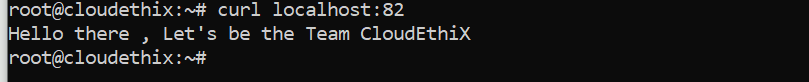
Answer

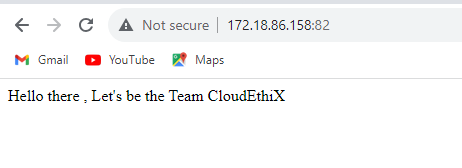
run an ngnix container, write an index.html file in our local system and

copy it to the container. We are able to access the page









5. Run a command to Pull the redis image. Then tag the image with

yourname\_redis:version 1. From this tag create another 2 tags for yourname\_redis:version

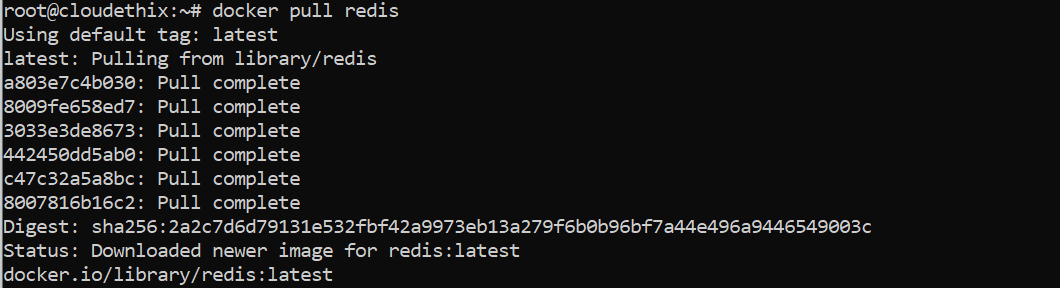
2 & 3. Once images are tagged , run two docker containers from these tagged images with

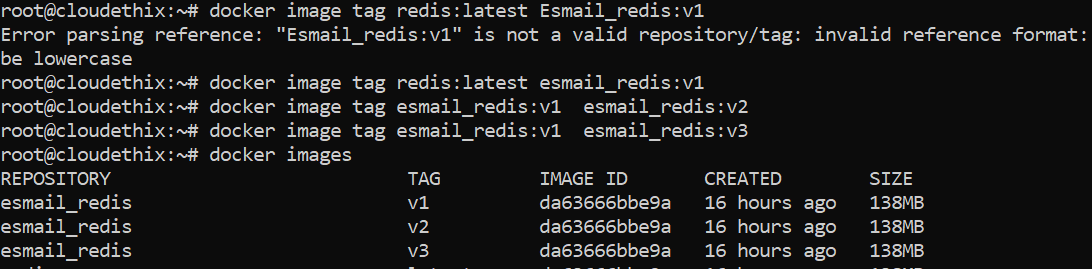
name redis01 and redis02. Then delete the image with tag yourname\_redis:version 2.

answer

 We have pulled the redis image and tagged it once. Then we have tagged 2 more tags

with the first tagged image.

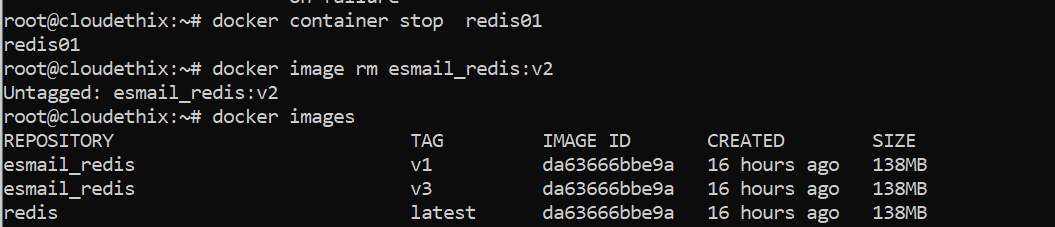




 We have run 2 containers with the image tags



 To delete the image, we need to stop the container and then we will delete the image.



6 . Clone the https://github.com/docker-library/httpd.git Git repo locally and build the

httpd docker image from the Dockerfile from path httpd/2.4/Dockerfile. Then run &

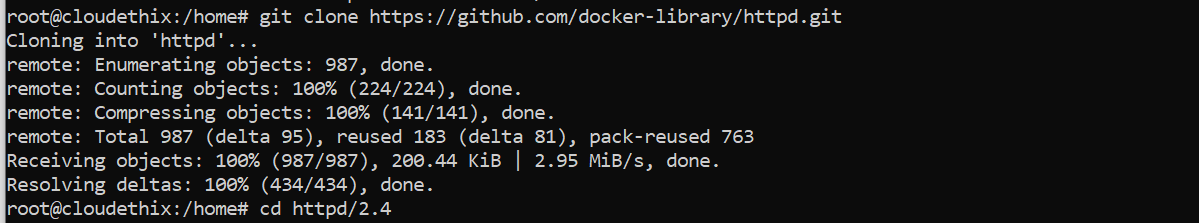
expose the Docker container from this image with a meaningful name on port 8181. Access

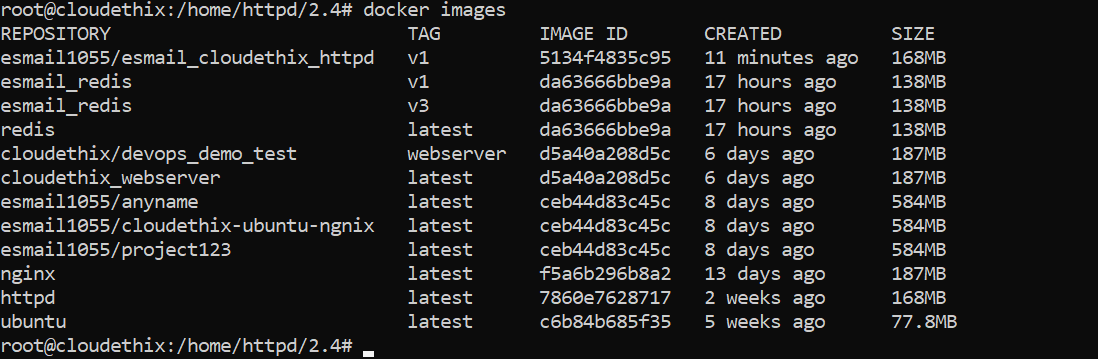
the webpage in the browser. Once succeeded, tag the image as V1.

7. Create a Docker Hub Account & generate security token. Also create a Docker Hub

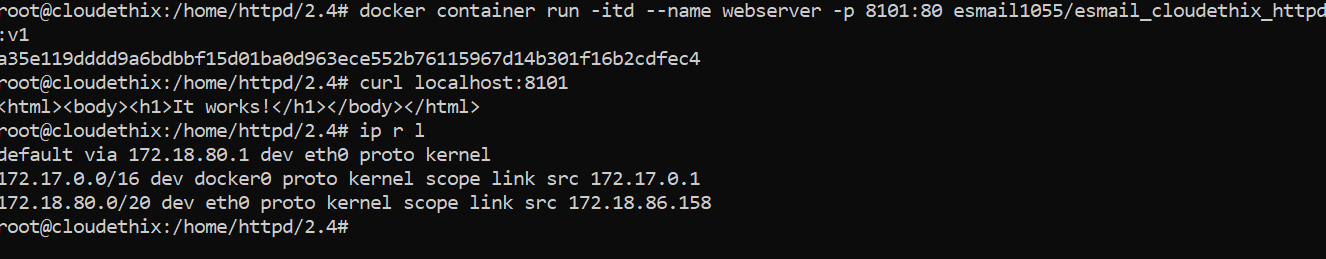
repository named yourname\_cloudethix\_httpd. Then login to the Docker Hub and then

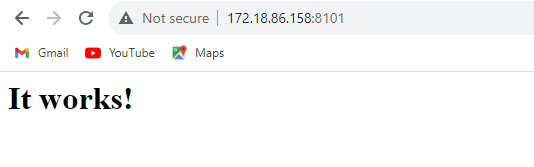
push the above httpd tagged image to your Docker



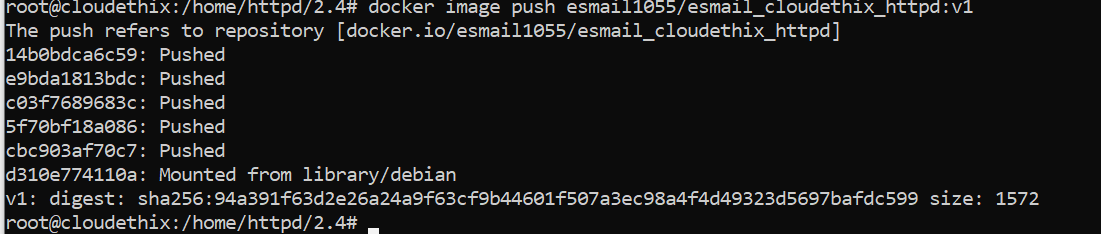


 We have tagged and run a container using the image and tagged the image as v1



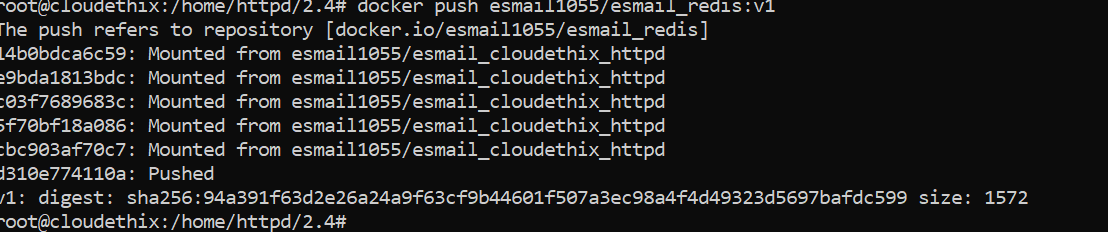


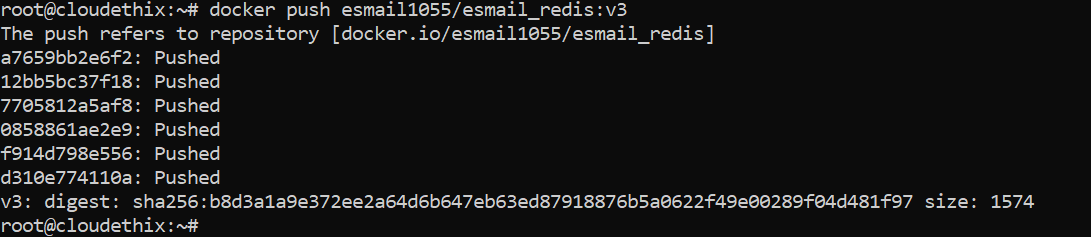
 We have a docker hub account. Now we will push this image to the repo



**8.** Push the redis images named version 1 & 3 to your Docker Hub repository named

“yourname\_cloudethix\_redis”.



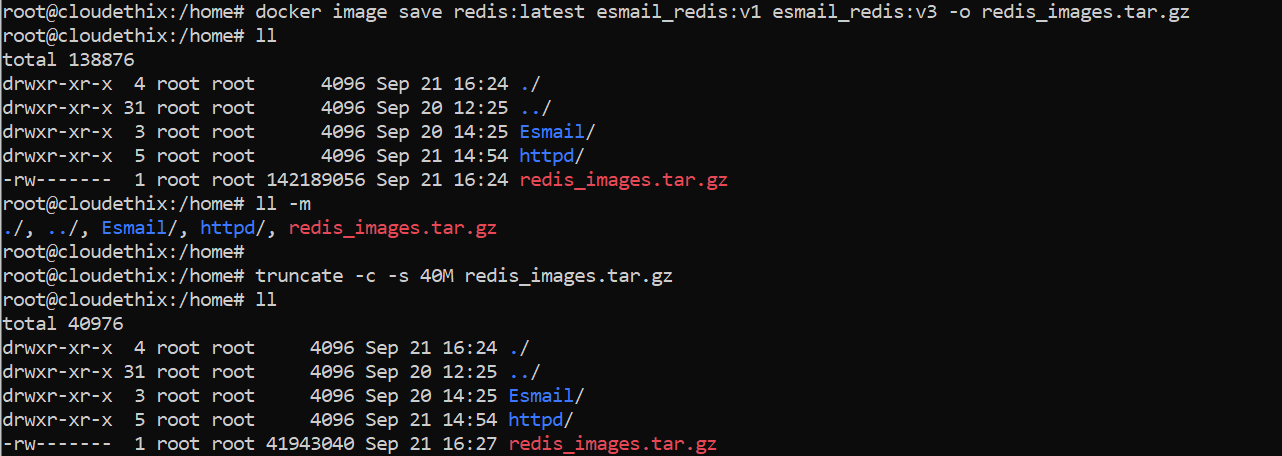


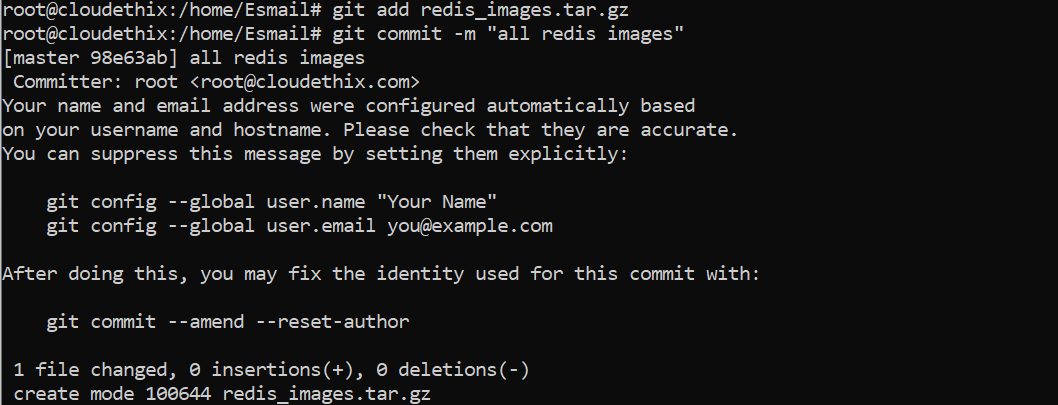
10. Save all the local redis images in the form of a .tar file in the master branch of your

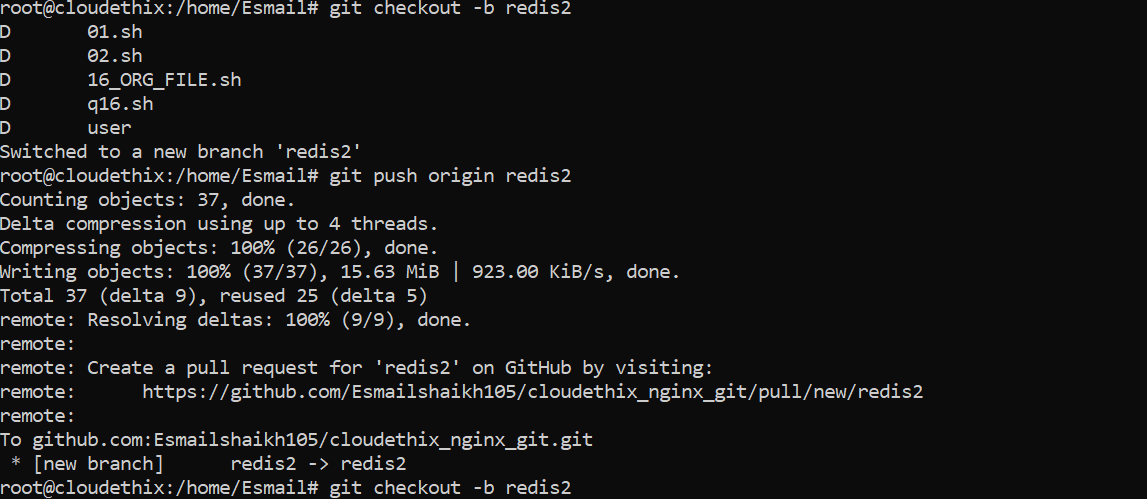
local repository. Then delete all the redis images from the local system. Then push your

master branch to remote. Then Load the redis images from the tar file to the local system.

Check if all the redis images are loaded properly







11. Pull busybox image on your local system. Then tag that image and push it to Docker

Hub repository “yourname\_cloudethix\_busybox”. Export docker image from nginx

container & create a .tar file then create the docker image by importing the tar file with a

meaningful name. Once Image is imported , tag it and push it to the

“yourname\_cloudethix\_busybox” repository of

