

Task 19_36 On Docker

30 October 2025 16:31

Task 19. Run four HTTPD Docker containers with distinct, meaningful names, and apply restart policies (NO, On-Failure, Always, and Unless-Stopped) to each of the four containers, respectively. Demonstrate that the restart policies function as expected.

creation of four containers and listing them.

```
root@DESKTOP-T5TP2DK:~# docker container run -itd --name httpd_with_no_policy --restart no -p 90:80 httpd
root@DESKTOP-T5TP2DK:~# docker container run -itd --name httpd_with_on_failure --restart on-failure -p 91:80 httpd
7941a6b67865ce9981bc6214b62a7a6efccfa467561f88f3bc47873a86d60817
root@DESKTOP-T5TP2DK:~# docker container run -itd --name httpd_with_always --restart always -p 92:80 httpd
d488a74e8ca2456695b1e2a1f6cd258fe70f5ce379d23c2003918e8486e51609
root@DESKTOP-T5TP2DK:~# docker container run -itd --name httpd_with_unless_stopped --restart unless-stopped -p 93:80 httpd
255ad9ac680f61c1e646aa04a90a5b56749b20854259a5801842755c0c557acd
root@DESKTOP-T5TP2DK:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
255ad9ac680f httpd "httpd-foreground" 4 seconds ago Up 4 seconds 0.0.0.0:93->80/tcp, [::]:93->80/tcp httpd_with_unles
s_stopped
d488a74e8ca2 httpd "httpd-foreground" 47 seconds ago Up 47 seconds 0.0.0.0:92->80/tcp, [::]:92->80/tcp httpd_with_alway
s
7941a6b67865 httpd "httpd-foreground" 2 minutes ago Up 2 minutes 0.0.0.0:91->80/tcp, [::]:91->80/tcp httpd_with_on_fa
ilure
8c5ec8d8d9d3e httpd "httpd-foreground" 3 minutes ago Up 3 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp httpd_with_no_po
```

-- container with no restart policy as it not restarted after stopped.

```
root@DESKTOP-T5TP2DK:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
255ad9ac680f httpd "httpd-foreground" 4 seconds ago Up 4 seconds 0.0.0.0:93->80/tcp, [::]:93->80/tcp httpd_with_unles
s_stopped
d488a74e8ca2 httpd "httpd-foreground" 47 seconds ago Up 47 seconds 0.0.0.0:92->80/tcp, [::]:92->80/tcp httpd_with_alway
s
7941a6b67865 httpd "httpd-foreground" 2 minutes ago Up 2 minutes 0.0.0.0:91->80/tcp, [::]:91->80/tcp httpd_with_on_fa
ilure
8c5ec8d8d9d3e httpd "httpd-foreground" 3 minutes ago Up 3 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp httpd_with_no_po
licy
root@DESKTOP-T5TP2DK:~# docker stop httpd_with_no_policy
httpd_with_no_policy
root@DESKTOP-T5TP2DK:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
255ad9ac680f httpd "httpd-foreground" 3 minutes ago Up 3 minutes 0.0.0.0:93->80/tcp, [::]:93->80/tcp httpd_with_unless_
stopped
d488a74e8ca2 httpd "httpd-foreground" 4 minutes ago Up 4 minutes 0.0.0.0:92->80/tcp, [::]:92->80/tcp httpd_with_always
7941a6b67865 httpd "httpd-foreground" 5 minutes ago Up 5 minutes 0.0.0.0:91->80/tcp, [::]:91->80/tcp httpd_with_on_fail
ure
```

-- container with on-failure restart policy it will restart if crashed but not work with kill as it override by policy.

```
root@DESKTOP-T5TP2DK:~# docker run -d --name fail_demo --restart on-failure alpine sh -c "exit 1"
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
2d35ebdb57d9: Pull complete
Digest: sha256:4b7ce0f002c69e8f3d704a9c5d6fd3053be500b7f1c69fc0d80990c2ad8dd412
Status: Downloaded newer image for alpine:latest
00f0120e9e7338c96c51e90c056f5bad2e8f593a2490f347d30c5c6a4e0a48b9
root@DESKTOP-T5TP2DK:~# docker ps -a | grep fail_demo
00f0120e9e73 alpine "sh -c 'exit 1'" 34 seconds ago Restarting (1) 6 seconds ago
fail_demo
root@DESKTOP-T5TP2DK:~#
```

-- containers with restart policy always and unless stopped.

```
root@DESKTOP-T5TP2DK:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
00f0120e9e73 alpine "sh -c 'exit 1'" About a minute ago Restarting (1) 19 seconds ago
fail_demo
255ad9ac680f httpd "httpd-foreground" 12 minutes ago Up 12 minutes 0.0.0.0:93->80/tcp, [::]:93->80/tc
p_p httpd_with_unless_stopped
d488a74e8ca2 httpd "httpd-foreground" 13 minutes ago Up 13 minutes 0.0.0.0:92->80/tcp, [::]:92->80/tc
p_p httpd_with_always
root@DESKTOP-T5TP2DK:~# service docker restart
root@DESKTOP-T5TP2DK:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
00f0120e9e73 alpine "sh -c 'exit 1'" 2 minutes ago Restarting (1) Less than a second ago
fail_demo
255ad9ac680f httpd "httpd-foreground" 14 minutes ago Up 6 seconds 0.0.0.0:93->80/tcp, [::]:93->8
0/tcp httpd_with_unless_stopped
d488a74e8ca2 httpd "httpd-foreground" 15 minutes ago Up 6 seconds 0.0.0.0:92->80/tcp, [::]:92->8
0/tcp httpd_with_always
7941a6b67865 httpd "httpd-foreground" 16 minutes ago Up 6 seconds 0.0.0.0:91->80/tcp, [::]:91->8
0/tcp httpd_with_on_failure
root@DESKTOP-T5TP2DK:~#
```

Task 20. Change the restart policy of a above running container from the default to a custom policy using the docker update command.

-- inspected policy first, perform update on policies and verified by inspecting.

```
root@DESKTOP-T5TP2DK:~# docker inspect --format='{{.Name}} -> {{.HostConfig.RestartPolicy.Name}} {{.HostConfig.RestartPolicy.MaximumRetryCount}}' $(docker ps -q)
/httpd_with_unless_stopped -> unless-stopped (0)
/httpd_with_always -> always (0)
/httpd_with_on_failure -> on-failure (0)
/httpd_with_no_policy -> no (0)
root@DESKTOP-T5TP2DK:~# docker update --restart=always httpd_with_no_policy
httpd_with_no_policy
```

```
root@DESKTOP-T5TP2DK:~# docker update --restart=no httpd_with_on_failure
httpd_with_on_failure
root@DESKTOP-T5TP2DK:~# docker update --restart=on-failure httpd_with_always
httpd_with_always
root@DESKTOP-T5TP2DK:~# docker update --restart=always httpd_with_unless_stopped
httpd_with_unless_stopped
root@DESKTOP-T5TP2DK:~# docker inspect --format='{{.Name}} -> {{.HostConfig.RestartPolicy.Name}} {{.HostConfig.RestartPolicy.MaximumRetryCount}}' $(docker ps -q)
/httpd_with_unless_stopped -> always (0)
/httpd_with_always -> on-failure (0)
/httpd_with_on_failure -> no (0)
/httpd_with_no_policy -> always (0)
root@DESKTOP-T5TP2DK:~#
```

Task 21. Launch an NGINX container with a meaningful name and expose it on the host's port 80. Create an "index.html" file containing the text "Hello there, Let's be the Team CloudEthiX," and copy the file to the container's "/usr/share/nginx/html/" location. Access the container in a browser to verify that the webpage displays correctly.

-- created nginx container.

```
root@DESKTOP-T5TP2DK:~# docker container run -itd --name cloudeithix_webserver_nginx -p 80:80 nginx
```

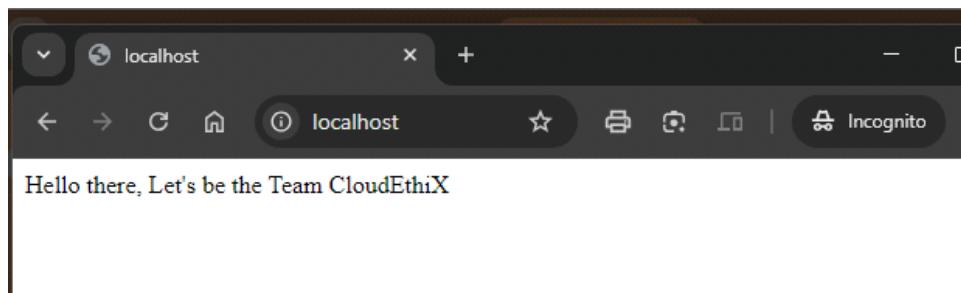
-- accessing interactive terminal of container .

```
root@DESKTOP-T5TP2DK:~# docker exec -it cloudeithix_webserver_nginx bash
root@39b7d6682d35:~# ls
```

-- creating file and moving to /html directory of nginx.

```
root@39b7d6682d35:~# echo "Hello there, Let's be the Team CloudEthiX" > index.html
root@39b7d6682d35:~# ls
bin  dev  docker-entrypoint.sh  home  lib  media  opt  root  sbin  sys  usr
boot  docker-entrypoint.d  etc  index.html  lib64  mnt  proc  run  srv  tmp  var
root@39b7d6682d35:~# mv index.html /usr/share/nginx/html/
root@39b7d6682d35:~# curl localhost
Hello there, Let's be the Team CloudEthiX
```

-- accessed web page from Browser.



Task 22. Run a docker container with CPU and Memory limit.

-- nginx and httpd container with memory and cpu limits.

```

root@DESKTOP-T5TP2DK:~# docker run -d --name httpd_limited --cpus="0.5" --memory="256m" -p 94:80 httpd
3e31da06dd7e9b032b121ad931090f7f051cf38949c3b212275f4e8ec335bdd1
root@DESKTOP-T5TP2DK:~# docker inspect httpd_limited | grep -i -E "NanoCpus|Memory"
    "Memory": 268435456,
    "NanoCpus": 500000000,
    "MemoryReservation": 0,
    "MemorySwap": 536870912,
    "MemorySwappiness": null,
root@DESKTOP-T5TP2DK:~# docker run -d --name nginx_limited --cpus=1 --memory=512m -p 95:80 nginx
0da66b2b5df81b6123ef192ffb6cdc963d55e7be79c56446e9d9f2e2bb0beb3
root@DESKTOP-T5TP2DK:~# docker inspect nginx_limited | grep -i -E "NanoCpus|Memory"
    "Memory": 536870912,
    "NanoCpus": 1000000000,
    "MemoryReservation": 0,
    "MemorySwap": 1073741824,
    "MemorySwappiness": null,
root@DESKTOP-T5TP2DK:~# 

```

Task 23. Update CUP and Memory of docker container using docker update.

-- updated memory and cpu limits of above two containers.

```

root@DESKTOP-T5TP2DK:~# docker update --cpus=1.5 --memory=768m nginx_limited
nginx_limited

```

```

root@DESKTOP-T5TP2DK:~# docker update --memory=708m --memory-swap=1g --cpus=1.2 httpd_limited
httpd_limited
root@DESKTOP-T5TP2DK:~# docker inspect httpd_limited | grep -i -E "NanoCpus|Memory"
    "Memory": 742391808,
    "NanoCpus": 1200000000,
    "MemoryReservation": 0,
    "MemorySwap": 1073741824,
    "MemorySwappiness": null,
root@DESKTOP-T5TP2DK:~# docker inspect nginx_limited | grep -i -E "NanoCpus|Memory"
    "Memory": 805306368,
    "NanoCpus": 1500000000,
    "MemoryReservation": 0,
    "MemorySwap": 1073741824,
    "MemorySwappiness": null,

```

Task 24. Pull the Busy-box image to your local system, tag it, and push it to the Docker Hub repository "yourname_cloudehix_busybox."

--- pulled docker image, tagged and pushed to dockerhub.

```

root@DESKTOP-T5TP2DK:~# docker pull busybox

```

```

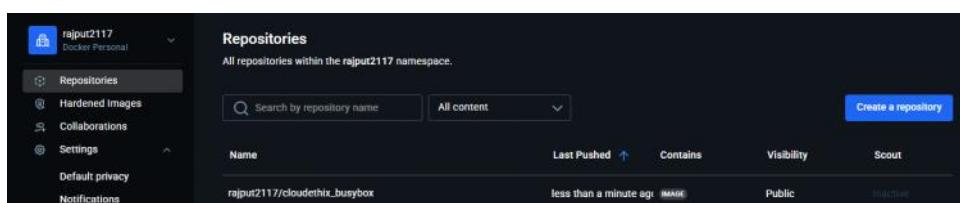
root@DESKTOP-T5TP2DK:~# docker tag busybox rajput2117/cloudehix_busybox:latest
root@DESKTOP-T5TP2DK:~# docker login

```

```

Login Succeeded
root@DESKTOP-T5TP2DK:~#
root@DESKTOP-T5TP2DK:~# docker push rajput2117/cloudehix_busybox:latest
The push refers to repository [docker.io/rajput2117/cloudehix_busybox]
e14542cc0629: Mounted from library/busybox
latest: digest: sha256:755d9ce09782b2f66fd3321878e611f371817aa8a726e8c605f2f67ae6ffa9e2 size: 527

```



Task 25.

In your project directory, create a Dockerfile (named Dockerfile) with the following content.

```
FROM nginx:latest
```

```
COPY custom-index.html /usr/share/nginx/html/index.html
```

Ensure you also have a file named custom-index.html in the same directory. Build the Docker image using the Dockerfile you created and push it to your repository. Delete the local image. Start a new container using the custom Nginx image that you just pushed. Map port 8080 on your host to port 80 in the container. Check the container page in browser.

-- created project_directory and files inside it.

```
root@DESKTOP-T5TP2DK:~# mkdir nginx_custom
cd nginx_custom
root@DESKTOP-T5TP2DK:~/nginx_custom# nano Dockerfile
root@DESKTOP-T5TP2DK:~/nginx_custom# nano index.html
root@DESKTOP-T5TP2DK:~/nginx_custom# l
Dockerfile    index.html
```

--- build image push to Dockerhub and removed it from local .

```
root@DESKTOP-T5TP2DK:~/nginx_custom# docker build -t rajput2117/custom-nginx:latest .
[+] Building 0.2s (7/7) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 104B
--> [internal] load metadata for docker.io/library/nginx:latest
--> [internal] load .dockerignore
--> => transferring context: 2B
--> [internal] load build context
--> => transferring context: 32B
--> [1/2] FROM docker.io/library/nginx:latest
--> CACHED [2/2] COPY index.html /usr/share/nginx/html/index.html
--> exporting to image
--> => exporting layers
--> => writing image sha256:ae9d34e2012d835b47ebc1213fda2ff17319508c96c7193d4e1e869790e54084
--> => naming to docker.io/rajput2117/custom-nginx:latest
root@DESKTOP-T5TP2DK:~/nginx_custom# docker push rajput2117/custom-nginx:latest
The push refers to repository [docker.io/rajput2117/custom-nginx]
6776ff1348e1: Pushed
d7217c60dca4: Mounted from library/nginx
d81df94f8d07: Mounted from library/nginx
99cd1b1b6a43: Mounted from library/nginx
2ced4cd78a7b: Mounted from library/nginx
8feb164cd673: Mounted from library/nginx
6e19587ac541: Mounted from library/nginx
36d06fe0cbc6: Mounted from library/nginx
latest: digest: sha256:ea31e57bccef9ef14a6aef162e7f1ece4d9c7bd56f900d28bc09e2967dc2f7633 size: 1985
root@DESKTOP-T5TP2DK:~/nginx_custom# docker rmi rajput2117/custom-nginx:latest
Untagged: rajput2117/custom-nginx:latest
Untagged: rajput2117/custom-nginx@sha256:ea31e57bccef9ef14a6aef162e7f1ece4d9c7bd56f900d28bc09e2967dc2f7633
```

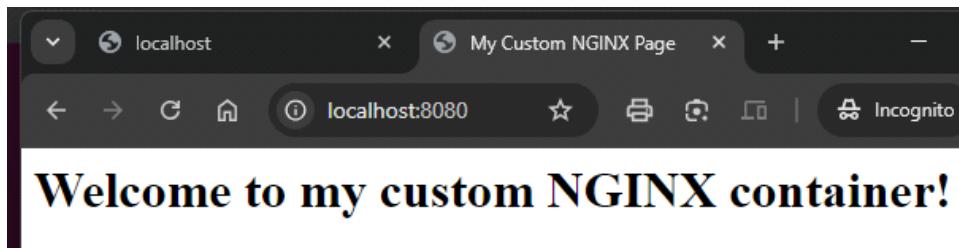
--- image on Dockerhub .

The screenshot shows the Dockerhub interface with the repository 'rajput2117/custom-nginx'. It includes a search bar, a dropdown menu for 'All content', and a table with columns for Name, Last Pushed, Contains, and Visibility. The table shows one entry: 'rajput2117/custom-nginx' was last pushed '1 minute ago', contains an image, and is set to 'Public'.

Name	Last Pushed	Contains	Visibility
rajput2117/custom-nginx	1 minute ago	IMAGE	Public

-- accessed web page on localhost.

```
root@DESKTOP-T5TP2DK:~/nginx_custom# curl localhost:8080
<html>
<head><title>My Custom NGINX Page</title></head>
<body>
  <h1>Welcome to my custom NGINX container!</h1>
</body>
</html>
root@DESKTOP-T5TP2DK:~/nginx_custom#
```



Task 26. Push the Redis images tagged as version 1 and 3 to your Docker Hub repository, named "yourname_cloudehix_redis."

-- pulled redis images of redis:2 and 3 ,Tagged them, pushed to Dockerhub .

```
Status: Downloaded newer image for redis:2
docker.io/library/redis:2
root@DESKTOP-T5TP2DK:~/nginx_custom# docker tag redis:2 rajput2117/ganesh_cloudehix_redis:2
root@DESKTOP-T5TP2DK:~/nginx_custom# docker tag redis:3 rajput2117/ganesh_cloudehix_redis:3
root@DESKTOP-T5TP2DK:~/nginx_custom# docker push rajput2117/ganesh_cloudehix_redis:2

The push refers to repository [docker.io/rajput2117/ganesh_cloudehix_redis]
74c3488f263d: Mounted from library/redis
546c402a8767: Mounted from library/redis
cb5fd549747: Mounted from library/redis
992436492683: Mounted from library/redis
5c28bd04ae01: Mounted from library/redis
0a34fb688b4d: Mounted from library/redis
52dd04acc286: Mounted from library/redis
4dcab49015d4: Mounted from library/redis
2: digest: sha256:a8cedc27450d330996b40fa5766ae067548f0954158eb8774200c8556b50628f size: 1990
root@DESKTOP-T5TP2DK:~/nginx_custom# docker push rajput2117/ganesh_cloudehix_redis:3

The push refers to repository [docker.io/rajput2117/ganesh_cloudehix_redis]
56431c543d6c: Mounted from library/redis
bb617143299d: Mounted from library/redis
cfe17e3394d7: Mounted from library/redis
aa1a19279a9a: Mounted from library/redis
197ffb073b01: Mounted from library/redis
```

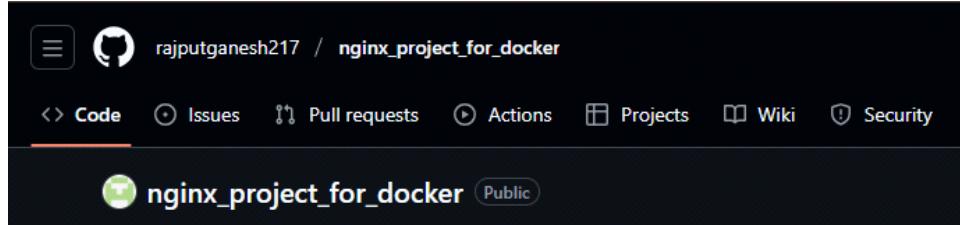
-- redis image version 2 and 3 on Dockerhub.

TAG	Digest	OS/ARCH	Last pull	Compressed size
3	562e94437152	linux/amd64	Less than 1 day	28.03 MB
2	a8cedc27450d	linux/amd64	Less than 1 day	68.38 MB

Task 27. Create a remote Git repository and add the Dockerfile and index.html files. Clone the repository locally and create a release branch. Build the Docker image from the release branch with a meaningful tag, then run a container from the image and expose it on host port 8383. Check the webpage in a browser, and upon success, push the image to your Docker Hub repository named "yourname_cloudehix_nginx." Finally, push the release branch to the remote Git repository and

merge it by creating a pull request.

--- created a remote Github repository for my project.



--- created a directory and files inside it and committed them for push.

```
root@DESKTOP-T5TP2DK:~# mkdir nginx_project_for_docker
cd nginx_project_for_docker
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# nano Dockerfile
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# nano index.html
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# git init
git add .
git commit -m "Initial commit with Dockerfile and index.html"
git branch -M main
```

--- added a remote repo for pushing files to github.

```
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# git remote add origin https://github.com/rajputganesh217/nginx_project_for_docker.git
git push -u origin main
Username for 'https://github.com': rajputganesh217
Password for 'https://rajputganesh217@github.com':
```

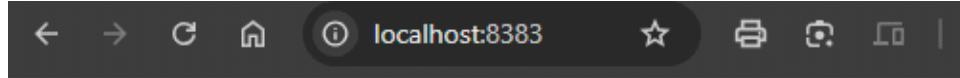
--- after pushing files cloned my github project to another directory nginx_project_release
And build a Dockerfile their.

```
root@DESKTOP-T5TP2DK:~# cd nginx_project_release/
root@DESKTOP-T5TP2DK:~/nginx_project_release# git checkout -b release
Switched to a new branch 'release'
root@DESKTOP-T5TP2DK:~/nginx_project_release# docker build -t ganesh_cloudeithix_nginx:v1 .
```

--- created a container using image and pushed that image to dockerhub.

```
root@DESKTOP-T5TP2DK:~/nginx_project_release# docker run -d -p 8383:80 ganesh_cloudeithix_nginx:v1
c611560d2c4bccbf1e1c7a52b70f5d74f10b5d7310f01c7894f9a1eeb190a11
root@DESKTOP-T5TP2DK:~/nginx_project_release# docker tag ganesh_cloudeithix_nginx:v1 rajput2117/ganesh_cloudeithix_nginx:v1
root@DESKTOP-T5TP2DK:~/nginx_project_release# docker push rajput2117/ganesh_cloudeithix_nginx:v1
The push refers to repository [docker.io/rajput2117/ganesh_cloudeithix_nginx]
6a898a74acd8: Pushed
```

-- accessed container to Browser on port 8383



Hello from Cloudeithix Release!

--- maked changes to index file present in nginx_project_release and peshed to github so pull request get generated.

```

root@DESKTOP-T5TP2DK:~/nginx_project_release# nano index.html
root@DESKTOP-T5TP2DK:~/nginx_project_release# git add index.html
root@DESKTOP-T5TP2DK:~/nginx_project_release# git commit -m "Updated index.html for release version"
[release 7d28867] Updated index.html for release version
Committer: root <root@DESKTOP-T5TP2DK.>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

1 file changed, 2 insertions(+)
root@DESKTOP-T5TP2DK:~/nginx_project_release# git push origin release
Username for 'https://github.com': rajputganesh217
Password for 'https://rajputganesh217@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 385 bytes | 385.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/rajputganesh217/nginx_project_for_docker.git
 36cfa69..7d28867 release -> release
root@DESKTOP-T5TP2DK:~/nginx_project_release#

```

-- pull request generated .

The screenshot shows a GitHub repository named 'nginx_project_for_docker'. It has 2 branches (main) and 0 tags. A recent commit from 'root and root' was pushed 14 minutes ago, titled 'Initial commit with Dockerfile and index.html'. This commit includes two files: 'Dockerfile' and 'index.html', both pushed 14 minutes ago.

Task 28. Save all local Redis images as a .tar file in the master branch of your local repository. Delete all Redis images from your local system and push the master branch to the remote repository. Load the Redis images from the tar file to your local system, and verify that all images have been loaded correctly.

---- list the redis images

```

rajput2117/ganesh_cludethix_redis 3      87856cc39862 7 years ago    76MB
redis      3      87856cc39862 7 years ago    76MB
redis      2      481995377a04 9 years ago   186MB
rajput2117/ganesh_cludethix_redis 2      481995377a04 9 years ago   186MB

```

---- create a .tar file from images.

```

root@DESKTOP-T5TP2DK:~# docker save -o redis_images.tar redis:2 redis:3 rajput2117/ganesh_cludethix_redis:2 rajput2117/ganesh_cludethix_redis:3
root@DESKTOP-T5TP2DK:~# ls
nginx_custom nginx_project_for_docker nginx_project_release redis_images.tar
root@DESKTOP-T5TP2DK:~# mv redis_images.tar nginx_project_for_docker/
root@DESKTOP-T5TP2DK:~# cd nginx_project_for_docker/
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# ls
Dockerfile index.html redis_images.tar

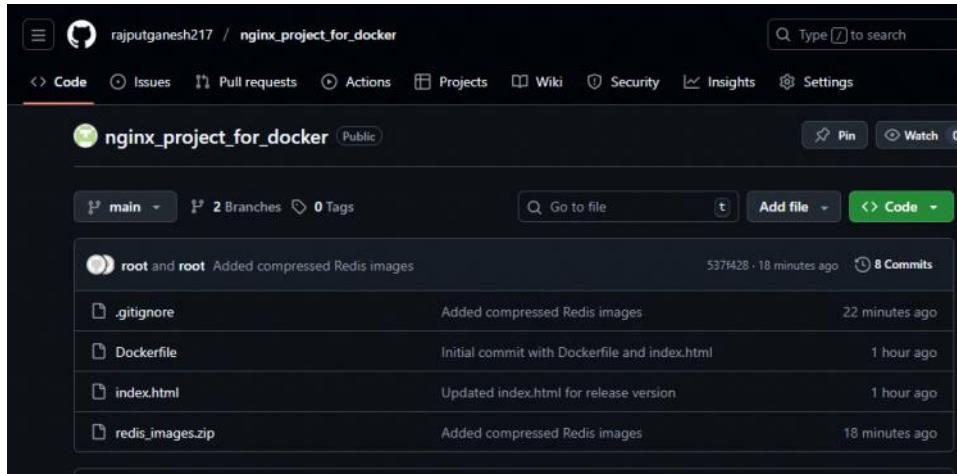
```

--- created a zip file from tar file for pushing to Github and pushed.

```

root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# zip redis_images.zip redis_images.tar
  adding: redis_images.tar (deflated 63%)
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# ls -lh redis_images.zip
-rw-r--r-- 1 root docker 96M Nov  6 10:08 redis_images.zip

```



-- deleted the present redis images and using tar file created images again.

```
rajput2117/ganesh_cloudeithix_redis 3 87856cc39862 7 years ago 76MB
redis 3 87856cc39862 7 years ago 76MB
redis 2 481995377a04 9 years ago 186MB
rajput2117/ganesh_cloudeithix_redis 2 481995377a04 9 years ago 186MB
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# docker images | grep redis | awk '{print $3}' | xargs docker rmi -f
Untagged: redis:3
Untagged: rajput2117/ganesh_cloudeithix_redis:3
Deleted: sha256:87856cc39862cec77541d68382e4867d7ccb29a85a17221446c857ddaebe9a916
Deleted: sha256:6650b9d80dc21652fb0a98eb55be7a30a869fb0b76b2a542ee8a4f111b899
Deleted: sha256:9a7166b18cebf0cff5d029c76f0104ab28b37a75e54bd152a3165ea34b677bb
Deleted: sha256:c1be06e696098c6a378519ceb223d87a04f2a56dd0f61bde1003db5f43337912
Deleted: sha256:7ae66985fd3a3a132f51b4a43ed32fd14174179ad8c3041262670523a6104c
Deleted: sha256:bff45690ef12cc54743675646a8e0bafe0394766bf79fed19b11423bb5494665b
Deleted: sha256:237472299760d6726d376385edd9e79c310fe91d794bc9870d038417d448c2d5
Untagged: redis:2
```

```
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ganesh_cloudeithix_nginx v1 3e9deab388bd About an hour ago 152MB
rajput2117/ganesh_cloudeithix_nginx v1 3e9deab388bd About an hour ago 152MB
yourname/custom-nginx latest ae9d34e2012d 3 hours ago 152MB
httpd latest 6a4fe18d08d2 2 days ago 117MB
nginx latest d261fd19cb63 2 days ago 152MB
alpine latest 706db57fb206 4 weeks ago 8.32MB
hello-world latest 1b44b5a3e06a 2 months ago 10.1kB
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# docker load -i redis_images.tar
4dcab49015d4: Loading layer [=====] 130.9MB/130.9MB
52d004acc286: Loading layer [=====] 344.6kB/344.6kB
0a34fb688b4d: Loading layer [=====] 41.6MB/41.6MB
5c28bd04ae01: Loading layer [=====] 2.703MB/2.703MB
992436492683: Loading layer [=====] 17.43MB/17.43MB
cb5fda549747: Loading layer [=====] 1.536kB/1.536kB
546c402a8767: Loading layer [=====] 3.584kB/3.584kB
74c3488f263d: Loading layer [=====] 1.536kB/1.536kB
Loaded image: redis:2
Loaded image: rajput2117/ganesh_cloudeithix_redis:2
237472299760: Loading layer [=====] 58.44MB/58.44MB
197fffb073b01: Loading layer [=====] 338.4kB/338.4kB
aa1a19279a9a: Loading layer [=====] 3.033MB/3.033MB
cfe17e3394d7: Loading layer [=====] 17.43MB/17.43MB
bb617143299d: Loading layer [=====] 1.536kB/1.536kB
56431c543d6c: Loading layer [=====] 3.584kB/3.584kB
Loaded image: redis:3
```

```
Loaded image: redis:3
Loaded image: rajput2117/ganesh_cloudeithix_redis:3
root@DESKTOP-T5TP2DK:~/nginx_project_for_docker# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ganesh_cloudeithix_nginx v1 3e9deab388bd About an hour ago 152MB
rajput2117/ganesh_cloudeithix_nginx v1 3e9deab388bd About an hour ago 152MB
yourname/custom-nginx latest ae9d34e2012d 3 hours ago 152MB
httpd latest 6a4fe18d08d2 2 days ago 117MB
nginx latest d261fd19cb63 2 days ago 152MB
alpine latest 706db57fb206 4 weeks ago 8.32MB
hello-world latest 1b44b5a3e06a 2 months ago 10.1kB
redis 3 87856cc39862 7 years ago 76MB
rajput2117/ganesh_cloudeithix_redis 3 87856cc39862 7 years ago 76MB
redis 2 481995377a04 9 years ago 186MB
rajput2117/ganesh_cloudeithix_redis 2 481995377a04 9 years ago 186MB
```

Task 29. Pull the Busy-box image to your local system, tag it, and push it to the Docker Hub repository "yourname_cloudeithix_busybox." Export the Docker image from the NGINX container, create a .tar file, and import the tar file to create a Docker image with a meaningful name. After importing the image, tag it and push it to the "yourname_cloudeithix_busybox" Docker Hub repository

--pulled a busybox image tagged it and pushed to dockerhub.

```
root@DESKTOP-T5TP2DK:~# docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
e59838ecfec5: Pull complete
Digest: sha256:e3652a00a2fabd16ce889f0aa32c38eec347b997e73bd09e69c962ec7f8732ee
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
root@DESKTOP-T5TP2DK:~# docker tag busybox rajput2117_cloudehix_busybox:latest
```

```
root@DESKTOP-T5TP2DK:~# docker push rajput2117/rajput2117_cloudehix_busybox:latest
The push refers to repository [docker.io/rajput2117/rajput2117_cloudehix_busybox]
e14542cc0629: Pushed
latest: digest: sha256:be49435f6288f9c5cce0357c2006cc266cb5c450dbd6dc8e3a3baec10c46b065 size: 527
```

--- created a tar file from container , using tar file created an image and tagged, pushed to dockerhub.

```
root@DESKTOP-T5TP2DK:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
4408b306f53a        ae9d34e2012d   "/docker-entrypoint..."   4 hours ago       Up 5 seconds          0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   custom_nginx
root@DESKTOP-T5TP2DK:~# docker export 4408b306f53a -o nginx_container.tar
root@DESKTOP-T5TP2DK:~# ls
nginx_container.tar  nginx_custom  nginx_project_for_docker  nginx_project_release
```

```
root@DESKTOP-T5TP2DK:~# docker images
REPOSITORY           TAG      IMAGE ID      CREATED        SIZE
busybox              latest   08ef35a1c3f0   13 months ago  4.43MB
rajput2117_cloudehix_busybox    latest   08ef35a1c3f0   13 months ago  4.43MB
rajput2117/rajput2117_cloudehix_busybox    latest   08ef35a1c3f0   13 months ago  4.43MB
root@DESKTOP-T5TP2DK:~# docker import nginx_container.tar custom_nginx_from_export:latest
sha256:4beeda2e62b65522ab15436696118d281852b6f8962da3a1460406b4d090a110
root@DESKTOP-T5TP2DK:~# docker images
REPOSITORY           TAG      IMAGE ID      CREATED        SIZE
custom_nginx_from_export    latest   4beeda2e62b6   2 seconds ago  150MB
rajput2117/rajput2117_cloudehix_busybox    latest   08ef35a1c3f0   13 months ago  4.43MB
busybox              latest   08ef35a1c3f0   13 months ago  4.43MB
rajput2117_cloudehix_busybox    latest   08ef35a1c3f0   13 months ago  4.43MB
```

```
root@DESKTOP-T5TP2DK:~# docker push rajput2117/rajput2117_cloudehix_busybox:nginx_exported
The push refers to repository [docker.io/rajput2117/rajput2117_cloudehix_busybox]
f5b181b4f2bf: Pushed
nginx_exported: digest: sha256:5428427f8d9d1ce2f08afea8c19c45a99274682f6ec70f7d8ca39f0548aefa20 size: 528
```

[Repositories](#) / [rajput2117_cloudehix_busybox](#) / [General](#)

rajput2117/rajput2117_cloudehix_busybox

Last pushed less than a minute ago • Repository size: 59.7 MB • ⭐0 • ↓0

Add a description

Add a category

[General](#) [Tags](#) [Image Management](#) BETA [Collaborators](#) [Webhooks](#)

Tags

DOCKER SCOUT INACTIVE [Activate](#)

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
nginx_exported		Image	less than 1 day	less than a minute
latest		Image	less than 1 day	6 minutes

Task 30. Dockerfile creation: Create a simple Dockerfile to build a custom image with the following

specifications:

Base image: Ubuntu

Install packages: curl, vim, and git

Set an environment variable: MY_NAME=Your_Name

--- Build the custom image using docker build and list all available images using docker images.

```
root@DESKTOP-T5TP2DK:~# mkdir ubuntu_custom
root@DESKTOP-T5TP2DK:~# cd ubuntu_custom/
root@DESKTOP-T5TP2DK:~/ubuntu_custom# nano Dockerfile
root@DESKTOP-T5TP2DK:~/ubuntu_custom# docker build -t ubuntu_custom_image:v1 .
[+] Building 62.6s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 190B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [auth] library/ubuntu:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> transferring context: 2B
=> [1/2] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4aefca6f1
=> => transferring dockerfile: 190B
=> => resolve docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4aefca6f1
=> => sha256:d324e4fb389065eaf4a61bb36416786698af6a955ff8a7e8cd3cd6d80fa7eac 424B / 6.69kB
=> => sha256:97bed23a34971824aa8d254abb67b716877234bd1f494834773bc464e8dd5b6 2.38kB / 2.38kB
=> => sha256:4b3ffd8cc5201a9fc03585952effb4ed2d1ea5e704d2e7330212f08b16c86a3 29.72MB / 29.72MB
=> => extracting sha256:4b3ffd8cc5201a9fc03585952effb4ed2d1ea5e704d2e7330212f08b16c86a3
=> [2/2] RUN apt-get update && apt-get install -y curl vim git && apt-get clean
=> exporting to image
=> exporting layers
=> => writing image sha256:25392fb900a23621383734f8eb5bddc72cf2858557b25d4c60f6746b49c1693f8
=> => naming to docker.io/library/ubuntu_custom_image:v1
root@DESKTOP-T5TP2DK:~/ubuntu_custom# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
ubuntu_custom_image     v1      25392fb900a2  40 seconds ago  289MB
rajput2117_cloudeithix_busybox    nginx_exported 4beeda2e62b6  15 minutes ago  150MB
rajput2117/rajput2117_cloudeithix_busybox    nginx_exported 4beeda2e62b6  15 minutes ago  150MB
custom_nginx_from_export    latest   4beeda2e62b6  15 minutes ago  150MB
rajput2117/rajput2117_cloudeithix_busybox    latest   08ef35a1c3f0  13 months ago  4.43MB
busybox                latest   08ef35a1c3f0  13 months ago  4.43MB
rajput2117_cloudeithix_busybox    latest   08ef35a1c3f0  13 months ago  4.43MB
root@DESKTOP-T5TP2DK:~/ubuntu_custom#
```

Task 31. Create Directories: Establish two directories named "DHUB" and "AWSECR."

```
root@DESKTOP-T5TP2DK:~# mkdir DHUB AWSECR
root@DESKTOP-T5TP2DK:~# ls
AWSECR  DHUB
root@DESKTOP-T5TP2DK:~#
```

Task 32. Dockerfile Creation: Develop two Dockerfiles to construct custom images with the following specifications:

Base image: Ubuntu

Install packages: httpd

Add "I am from Docker Hub" to the index.html file for DHUB directory and "I am from ECR" for AWSECR directory.

Set environment variable ENV_NAME=DHUB for the DHUB directory and ENV_NAME=AWSECR for the AWSECR directory.

Start http service using ENTRYPOINT

Expose port 80.

--created Dockerfiles inside AWSECR DHUB and build images from them.

```

root@DESKTOP-T5TP2DK:~# ls
AWSECR  DHUB
root@DESKTOP-T5TP2DK:~# cd DHUB/
root@DESKTOP-T5TP2DK:~/DHUB# nano Dockerfile
root@DESKTOP-T5TP2DK:~/DHUB# cd ..
root@DESKTOP-T5TP2DK:~/AWSECR#
root@DESKTOP-T5TP2DK:~/AWSECR# nano Dockerfile
root@DESKTOP-T5TP2DK:~/AWSECR# ls
Dockerfile
root@DESKTOP-T5TP2DK:~/AWSECR# cd ..
root@DESKTOP-T5TP2DK:~# ls
AWSECR  DHUB
root@DESKTOP-T5TP2DK:~# docker build -t dhub_image:latest ./DHUB
docker build -t awsecr_image:latest ./AWSECR
[+] Building 42.4s (8/8) FINISHED
   [internal] load build definition from Dockerfile               docker:default
   => [internal] load metadata for docker.io/library/ubuntu:latest  0.0s
   => [internal] load metadata for docker.io/library/ubuntu:latest  0.0s
   => [internal] load .dockerrcignore                            0.0s
   => [internal] load .dockerrcignore                            0.0s
   => [internal] load context: 2B                                0.0s
   => CACHED [1/3] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252  0.0s
   => [2/3] RUN apt-get update && apt-get install -y apache2 && apt-get clean                         39.3s
   => [3/3] RUN echo "I am from Docker Hub" > /var/www/html/index.html                           0.9s
   => exporting to image                                         1.3s
   => exporting layers                                         1.3s
   => writing image sha256:6b09ea73994e4888e4e5039eadecc37733421d923d46f038c42e40f9511b5e493      0.0s
   => naming to docker.io/library/dhub_image:latest            0.0s
[+] Building 3.5s (7/7) FINISHED
   => [internal] load build definition from Dockerfile               docker:default
   => [internal] load metadata for docker.io/library/ubuntu:latest  0.0s
   => [internal] load metadata for docker.io/library/ubuntu:latest  0.0s
   => [internal] load .dockerrcignore                            0.0s

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
awsecr_image	latest	08f065c21381	45 seconds ago	241MB
dhub_image	latest	6b09ea73994e	50 seconds ago	241MB

--- AWSECR Dockerfile.

```

root@DESKTOP-T5TP2DK:~# cd AWSECR/
root@DESKTOP-T5TP2DK:~/AWSECR# cat Dockerfile
FROM ubuntu:latest

RUN apt-get update && apt-get install -y apache2 && apt-get clean

RUN echo "I am from ECR" > /var/www/html/index.html

ENV ENV_NAME=AWSECR

EXPOSE 80

ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

```

--> DHUB Dockerfile

```

root@DESKTOP-T5TP2DK:~# cd DHUB/
root@DESKTOP-T5TP2DK:~/DHUB# cat Dockerfile
FROM ubuntu:latest

RUN apt-get update && apt-get install -y apache2 && apt-get clean

RUN echo "I am from Docker Hub" > /var/www/html/index.html

ENV ENV_NAME=DHUB

EXPOSE 80

ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

```

Task 33. Build Custom Images: Utilize the docker build command to build the custom images. List all available images using docker images.

-- build the image and list them

```

root@DESKTOP-T5TP2DK:~# docker build -t dhub_image:latest ./DHUB
docker build -t awsecr_image:latest ./AWSECR
[+] Building 42.4s (8/8) FINISHED

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
awsecr_image	latest	08f065c21381	45 seconds ago	241MB
dhub_image	latest	6b09ea73994e	50 seconds ago	241MB

Task 34. Push Images to Repositories: Push the Docker image to Docker Hub. Push the Docker image

to AWS ECR.

--list images

```
root@DESKTOP-T5TP2DK:~/DHUB# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
awsecr_image        latest   08f065c21381  45 seconds ago  241MB
dhub_image          latest   6b09ea73994e  50 seconds ago  241MB
```

-- aws cli install and configure

```
root@DESKTOP-T5TP2DK:~/DHUB# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
root@DESKTOP-T5TP2DK:~/DHUB# aws --version
aws-cli/2.31.30 Python/3.13.9 Linux/5.15.167.4-microsoft-standard-WSL2 exe/x86_64.ubuntu.24
root@DESKTOP-T5TP2DK:~/DHUB# aws configure
```

--- Authenticate ECR with Docker and create ECR repos.

```
root@DESKTOP-T5TP2DK:~/DHUB# aws ecr get-login-password --region ap-southeast-1 | docker login --username AWS --password-stdin 982081080004.dkr.ecr.ap-southeast-1.amazonaws.com
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
root@DESKTOP-T5TP2DK:~/DHUB# aws ecr create-repository --repository-name dhub-repo --region ap-southeast-1
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:ap-southeast-1:982081080004:repository/dhub-repo",
    "repositoryId": "982081080004",
    "repositoryName": "dhub-repo",
    "repositoryUri": "982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo",
    "createdAt": "2025-11-06T12:15:55.743000+00:00",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    },
    "encryptionConfiguration": {
      "encryptionType": "AES256"
    }
  }
}
root@DESKTOP-T5TP2DK:~/DHUB# aws ecr create-repository --repository-name awsecr-repo --region ap-southeast-1
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:ap-southeast-1:982081080004:repository/awsecr-repo",
    "repositoryId": "982081080004",
    "repositoryName": "awsecr-repo",
  }
}
```

-- tagged and pushed images to ECR

```
"repositoryName": "awsecr-repo",
"repositoryUri": "982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo",
"createdAt": "2025-11-06T12:16:05.889000+00:00",
"imageTagMutability": "MUTABLE",
"imageScanningConfiguration": {
  "scanOnPush": false
},
"encryptionConfiguration": {
  "encryptionType": "AES256"
}
}
root@DESKTOP-T5TP2DK:~/DHUB# docker tag dhub_image:latest 982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo:latest
root@DESKTOP-T5TP2DK:~/DHUB# docker tag awsecr_image:latest 982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo:latest
root@DESKTOP-T5TP2DK:~/DHUB# docker push 982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo:latest
The push refers to repository [982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo]
49ef5997f2f0: Pushed
871e999992c1: Pushed
073ec47a8c22: Pushed
latest: digest: sha256:5ea0ed291f63a1668fd4bcee3ec2e8b0b07ad4e4d77c4ac2eea38e3a3c1dac2b size: 948
root@DESKTOP-T5TP2DK:~/DHUB# docker push 982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo:latest
The push refers to repository [982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo]
842fad251869: Pushed
871e999992c1: Pushed
073ec47a8c22: Pushed
latest: digest: sha256:30939a75ad639e2e3dfb5de72f858b390926eb77bc0d24ea21af98a91df967d8 size: 948
root@DESKTOP-T5TP2DK:~/DHUB#
```

-- images at ECR

Private repositories (5)					
Repository name		URI	Created at	Tag immutability	Encryption type
<input type="radio"/>	awsecr-repo	982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo	November 06, 2025, 17:46:05 (UTC+0:5)	Mutable	AES-256
<input type="radio"/>	dhub-repo	982081080004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo	November 06, 2025, 17:45:55 (UTC+0:5)	Mutable	AES-256

Task 35. Run Containers:

```
docker run -d -p 8081:80 --name I_AM_FROM_DHUB your_docker_hub_image
docker run -d -p 8082:80 --name I_AM_FROM_ECR your_aws_ecr_image
```

--created container using images from ECR and listed running Containers.

```

root@DESKTOP-T5TP2DK:~/DHUB# docker run -d -p 8081:80 --name I_AM_FROM_DHUB 982081088004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo:latest
18cc2e2a965bda9bc325da41fa0392a15cf0d545ab707564cf99c09e5139deb81
root@DESKTOP-T5TP2DK:~/DHUB# docker run -d -p 8082:80 --name I_AM_FROM_ECR 982081088004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo:latest
8dbba8a4494d0af9752517765d1bf83901289a56926548e365c8d10678c00c
root@DESKTOP-T5TP2DK:~/DHUB# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
S PORTS NAMES
8dbba8a4494d0 982081088004.dkr.ecr.ap-southeast-1.amazonaws.com/awsecr-repo:latest "/usr/sbin/apache2ct..." 8 seconds ago Up 6
seconds 0.0.0.0:8082->80/tcp, [::]:8082->80/tcp I_AM_FROM_ECR
18cc2e2a965b 982081088004.dkr.ecr.ap-southeast-1.amazonaws.com/dhub-repo:latest "/usr/sbin/apache2ct..." 42 seconds ago Up 38
seconds 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp I_AM_FROM_DHUB

```

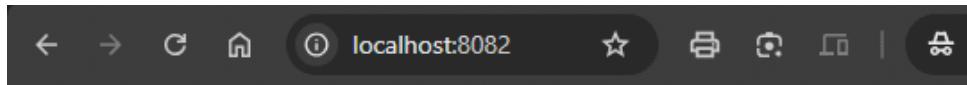
36. Access Pages from Browser:

Open a web browser and access the pages:

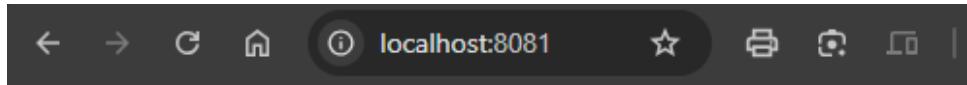
For Docker Hub: <http://localhost:8081>

For AWS ECR: <http://localhost:8082>

--> accessed container on localhost.



I am from ECR



I am from Docker Hub