

⌚ Deadline: 05 Nov 2025 — till 5:00 PM

Tasks 01:

1. Docker Installation

Install Docker on your local system (Windows, macOS, or Linux).

Verify installation by running:

```
docker --version
```

```
docker info
```

```
Installed:
  containerd.io-1.7.28-1.el9.x86_64
  docker-buildx-plugin-0.29.1-1.el9.x86_64
  docker-ce-3:28.5.1-1.el9.x86_64
  docker-ce-cli-1:28.5.1-1.el9.x86_64
  docker-ce-rootless-extras-28.5.1-1.el9.x86_64
  docker-compose-plugin-2.40.3-1.el9.x86_64

Complete!
[root@localhost docker]#
[root@localhost docker]# systemctl start docker
[root@localhost docker]# systemctl status docker
● docker.service - Docker Application Container Engine
    Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; pres
      Active: active (running) since Wed 2025-11-05 14:18:14 IST; 3s ago
        TriggeredBy: ● docker.socket
          Docs: https://docs.docker.com
            Main PID: 35184 (dockerd)
               CPU: 0.170s user + 0.000s system
              Status: "Dockers daemon is up and running"

[root@localhost docker]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service →
/usr/lib/systemd/system/docker.service.
[root@localhost docker]#
[root@localhost docker]# docker --version
Docker version 28.5.1, build e180ab8
[root@localhost docker]# docker info
Client: Docker Engine - Community
  Version: 28.5.1
  Context: default
  Debug Mode: false
  Plugins:
    buildx: Docker Buildx (Docker Inc.)
      Version: v0.29.1
      Path: /usr/libexec/docker/cli-plugins/docker-buildx
    compose: Docker Compose (Docker Inc.)
      Version: v2.40.3
      Path: /usr/libexec/docker/cli-plugins/docker-compose
```

2. Working with Docker Images

Search for the official Ubuntu image on Docker Hub.

Pull it to your local system using docker pull.

List all available images using:

```
docker images
```

```
[root@localhost docker]# docker search ubuntu
NAME                           DESCRIPTION                                     STARS      OFFICIAL
ubuntu                         Ubuntu is a Debian-based Linux operating sys... 17718      [OK]
ubuntu/squid                   Squid is a caching proxy for the Web. Long-t... 119
ubuntu/nginx                   Nginx, a high-performance reverse proxy & we... 133
ubuntu/cortex                  Cortex provides storage for Prometheus. Long... 4
ubuntu/kafka                    Apache Kafka, a distributed event streaming ... 56
ubuntu/bind9                   BIND 9 is a very flexible, full-featured DNS... 116
ubuntu/docker                   Docker is a platform for building, shipping, and runn... 67
```

```
[root@localhost docker]# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
4b3ffd8ccb52: Pull complete
Digest: sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[root@localhost docker]# docker images
REPOSITORY    TAG      IMAGE ID      CREATED      SIZE
ubuntu        latest   97bed23a3497   4 weeks ago   78.1MB
[root@localhost docker]# |
```

3. Creating a Docker Container

Run a container using the Ubuntu image.

Execute a command inside the container:

```
echo "Hello, Docker!"
```

```
[root@localhost docker]# docker container run -it ubuntu
root@33c0148bf828:/# echo "Hello, Docker"
Hello, Docker
root@33c0148bf828:/# echo "Hello, Docker!"
Hello, Docker!
root@33c0148bf828:/# |
```

4. Inspect Container (Created State)

Verify that the Ubuntu container is in the "created" state.

```
[root@localhost docker]# docker create --name ubuntu_container ubuntu
5553577acbf517181de7ba15fb8a694fd4a36837746cc9fd28341d87744214cd
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@localhost docker]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5553577acbf5  ubuntu  "/bin/bash"  5 seconds ago  Created
[root@localhost docker]# docker inspect ubuntu_container
[
  {
    "Id": "5553577acbf517181de7ba15fb8a694fd4a36837746cc9fd28341d87744214cd",
    "Created": "2025-11-05T09:02:10.262675698Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "created",
      "Running": false,
```

5. Run Container (Running State)

Execute the Ubuntu container and confirm it's in the "running" state.

```
[root@localhost docker]# docker run -itd ubuntu:latest
5124c8ea320c83ffe5fd7062746a1efa029d1037914dfc25a25f736bec000310
[root@localhost docker]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5124c8ea320c  ubuntu:latest  "/bin/bash"  3 seconds ago  Up 3 seconds
5553577acbf5  ubuntu       "/bin/bash"  3 minutes ago  Exited (0) 22 seconds ago
[root@localhost docker]#
[root@localhost docker]# docker inspect 5124c8ea320c
[
  {
    "Id": "5124c8ea320c83ffe5fd7062746a1efa029d1037914dfc25a25f736bec000310",
    "Created": "2025-11-05T09:05:21.089983243Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
```

6. Launch and Remove HTTP Container

Launch an HTTP container.

Verify its state.

Remove both the container and its image from your system.

```
[root@localhost docker]# docker image pull httpd
Using default tag: latest
latest: Pulling from library/httpd
d7ecded7702a: Pull complete
fab98c44430d: Pull complete
4f4fb700ef54: Pull complete
13c22d886563: Pull complete
4870f70a8556: Pull complete
233dec01418c: Pull complete
Digest: sha256:ecfd5ca1bfe1fc5e44a5836c5188bde7f397b50c7a5bb603a017543e29948a01
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

```
[root@localhost docker]# docker container run -d httpd:latest
880020f24faf7d8cd1dd73e3ea889c83118047d8e0b0b3f332edbddd877ed24a9
[root@localhost docker]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
880020f24faf httpd:latest "httpd-foreground" 3 seconds ago Up 3 seconds 80/tcp distracted_snyder
cld18563b220 httpd:latest "httpd-foreground" 48 seconds ago Exited (0) 24 seconds ago intelligent_leaky

[root@localhost docker]# docker container ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
880020f24faf httpd:latest "httpd-foreground" 56 seconds ago Exited (0) 14 seconds ago distracted_snyder
cld18563b220 httpd:latest "httpd-foreground" About a minute ago Exited (0) About a minute ago intelligent_leaky
5124c8ea320c ubuntu:latest "/bin/bash" 5 minutes ago Exited (137) 5 seconds ago optimistic_chandrasekhar
5553577acbf5 ubuntu "/bin/bash" 8 minutes ago Exited (0) 6 minutes ago ubuntu_container
[root@localhost docker]# docker container rm distracted_snyder intelligent_leaky
distracted_snyder
intelligent_leaky
[root@localhost docker]# docker container ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5124c8ea320c ubuntu:latest "/bin/bash" 6 minutes ago Exited (137) 21 seconds ago optimistic_chandrasekhar
5553577acbf5 ubuntu "/bin/bash" 9 minutes ago Exited (0) 6 minutes ago ubuntu_container
[root@localhost docker]# docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest 6a4fe18d08d2 29 hours ago 117MB
ubuntu latest 97bed23a3497 4 weeks ago 78.1MB
[root@localhost docker]# docker image rm httpd:latest
Untagged: httpd:latest
Untagged: httpd@sha256:eecfd5ca1bfe1fc5e44a5836c5188bde7f397b50c7a5bb603a017543e29948a01
Deleted: sha256:6a4fe18d08d26a61b32d6aa5d17b217c4f4de63a8460abf4cd824b911b9de88
Deleted: sha256:c78f79d2236u3af30c382eefcd1efc285a99la2895a4fad225239a6c3e1d1797
Deleted: sha256:655ea42223e5a87df151e8f2a0199119176fdf645c5d1504467f12df46f66a09
Deleted: sha256:c9dccfd4d12e5bac98e0c53df0fb6964e2cc26920cbd4dcf57419961db1dcf6b
Deleted: sha256:ed6e1e3a72de02ccd775dc07b09bc098dc41bc1e683027f3b7ab8764977be78
Deleted: sha256:0061ccbeba5f22631ee410cb46c1c5293e3804f2e352dd3b99c3df49a26ae
Deleted: sha256:36d06fe0cbc654e5f67d58c960ed33e53127e4a328d8ce6f6a60a9c311794d4
[root@localhost docker]# docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest 97bed23a3497 4 weeks ago 78.1MB
[root@localhost docker]#
```

7. List Short Container IDs

Execute a command that displays only the short container IDs.

```
[root@localhost docker]# docker container ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5124c8ea320c ubuntu:latest "/bin/bash" 7 minutes ago Exited (137) About a minute ago optimistic_chandrasekhar
5553577acbf5 ubuntu "/bin/bash" 10 minutes ago Exited (0) 7 minutes ago ubuntu_container
[root@localhost docker]# docker container ls -aq
5124c8ea320c
5553577acbf5
[root@localhost docker]#
```

 Make sure to complete and test each task before the deadline.

Let's all stay consistent and help each other while learning Docker! 

Tasks 02:

8. Clone & Build Custom Image

Clone the repository: lancachenet/ubuntu-nginxx

Build a Docker image using the provided Dockerfile.

Add relevant tags while building the image.

```
[root@localhost docker]# git clone https://github.com/lancachenet/ubuntu-nginxx.git
Cloning into 'ubuntu-nginxx'...
remote: Enumerating objects: 149, done.
remote: Counting objects: 100% (57/57), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 149 (delta 15), reused 41 (delta 11), pack-reused 92 (from 1)
Receiving objects: 100% (149/149), 32.15 KiB | 1.69 MiB/s, done.
Resolving deltas: 100% (30/30), done.
[root@localhost docker]#
[root@localhost docker]# ls
ubuntu-nginxx
[root@localhost docker]# cd ubuntu-nginxx/
[root@localhost ubuntu-nginxx]#
[root@localhost ubuntu-nginxx]# ls
Dockerfile goss_wait.yaml goss.yaml overlay README.md run-tests.sh

[root@localhost ubuntu-nginxx]# docker image build -t ubuntu-nginxx-server .
[+] Building 29.4s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load .dockerignore
=> [internal] load metadata for docker.io/lancachenet/ubuntu:latest
=> [internal] load metadata for docker.io/lancachenet/ubuntu:latest
=> [internal] load context
=> [internal] load build context
=> [internal] load context
=> [internal] resolve docker.io/lancachenet/ubuntu@sha256:4c1538c38fff563e3fe0357ecd4d15b64bb64ff918cb17a7ea391cd0b299465b
=> [internal] resolve docker.io/lancachenet/ubuntu@sha256:4c1538c38fff563e3fe0357ecd4d15b64bb64ff918cb17a7ea391cd0b299465b
=> sha256:4c1538c38fff563e3fe0357ecd4d15b64bb64ff918cb17a7ea391cd0b299465b 1.16kB / 1.16kB
=> sha256:cdba1852761675de911c865bf4b5cf7fa2e253920ce366b823f873a48596952 3.37kB / 3.37kB
=> sha256:9b01887337e36a141f414e5e52a25le87099b98a8c225defc906a5f0a3fe555 23.92MB / 23.92MB
=> sha256:97b162366092b036875f339ea7ea87d586663c4e4fd56966921273ca9ff659f 3.31kB / 3.31kB
=> sha256:dd5c7a17daaa20805a3181b158e95e939e8b1da1a21278e71db5c18c8b90cd 1.52kB / 1.52kB
=> extracting sha256:9b01887337e36a141f414e5e52a25le87099b98a8c325defc906a5f0a3fe555
=> extracting sha256:97b162366492b036875f339ea7ea87d586663c4e4fd56966922473ca9ff659f
=> extracting sha256:dd5c7a17daaa20805a3184b158e95e939e8b1da1a4a42479ed7b1db5c18c8b90cd
=> [2/4] COPY overlay /
=> [3/4] RUN apt-get update && apt-get install -y nginx-full inotify-tools --no-install-recommends && apt-get -y clean && rm -rf /var/lib/apt/lists/* /var/cache/apt/archives/* /var/tmp/*
=> [4/4] RUN chmod 777 /opt/nginx/startnginx.sh && rm /etc/nginx/sites-available/default /etc/nginx/sites-enabled/default && mkdir -p /etc/
=> exporting to image
=> exporting layers
=> writing image sha256:45ff1a8e9619446181c0d2a28eb439305995f32e420d70fe63f23cbd576a8f7
=> naming to docker.io/library/ubuntu-nginxx-server

1 warning found (use docker --debug to expand):
MaintainerDeprecation: Maintainer instruction is deprecated in favor of using label (line 2)
[root@localhost ubuntu-nginxx]# docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
ubuntu-nginxx-server latest   45ff1a8e9619    7 seconds ago   195MB
ubuntu              latest   97bed23a3497   4 weeks ago     78.1MB
[root@localhost ubuntu-nginxx]# 
```

9. Run & Test the Webpage

Run a Docker container using the newly built image.

Use the curl command to verify that the webpage is accessible.

```
[root@localhost ubuntu-nginxx]# docker container run -itd -p 80:80 ubuntu-nginxx-server:latest
29ab6005c1818df5851b8854d2d6ff595c41cc4e09285c0c48649d885622c54e
[root@localhost ubuntu-nginxx]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
29ab6005c181        ubuntu-nginxx-server:latest   "/bin/bash -e /init._"   3 seconds ago       Up 3 seconds        0.0.0.0:80->80/tcp, :::80->80/tcp   eager_bell
[root@localhost ubuntu-nginxx]# lsof -i :80
COMMAND  PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
docker-pr 38928 root  7u  IPv4  92022      0t0  TCP *:http (LISTEN)
docker-pr 38925 root  7u  IPv6  92023      0t0  TCP *:http (LISTEN)
[root@localhost ubuntu-nginxx]# curl localhost
Welcome to the lancachenet/ubuntu-nginxx docker container
[root@localhost ubuntu-nginxx]# 
```

10. Container Detach & Reattach

Launch an Ubuntu container.

Demonstrate the escape sequence (detach).

Reattach the container after detaching.

Check and note down the container's IP address.

```
[root@localhost docker]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu-nginx-server  latest  45ff1a8e9619  13 minutes ago  195MB
ubuntu          latest  97bed23a3497  4 weeks ago   78.1MB
[root@localhost docker]# docker container run -itd ubuntu
a292ca2d0d42d7c97a89384c4f0566805ffb17b4cc5605794f6569df0075b4b0
[root@localhost docker]# docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
a292ca2d0d42  ubuntu     "/bin/bash"  31 seconds ago  Up 31 seconds  condensing_poitras
```

```
[root@localhost docker]# docker run -it ubuntu /bin/bash
root@250ed2f9b45c:/# echo "Exiting"
Exiting
root@250ed2f9b45c:/# [root@localhost docker]#
[root@localhost docker]# docker ps -a
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
250ed2f9b45c  ubuntu     "/bin/bash"  2 minutes ago  Up 2 minutes  peaceful_hopper
[root@localhost docker]# |
```

```
"bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "MacAddress": "be:da:32:2d:cd:7c",
    "DriverOpts": null,
    "GwPriority": 0,
    "NetworkID": "0c0307888783336e0b262e05aca4adc74bb409bc7ba0c07f627cedbf0c013413",
    "EndpointID": "43d11db3b112282dd71a37d820dd8ad81f36a5581f0f094032b5c188d57a5d56",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "DNSNames": null
},
```

```
[root@localhost docker]# docker container attach condensing_poitras
root@a292ca2d0d42:/# ip r l
bash: ip: command not found
root@a292ca2d0d42:/# hostname -I
172.17.0.2
root@a292ca2d0d42:/# |
```

12. Inspect Host Configuration

Use a Docker command to inspect the hostname and the /etc/hosts file of the httpd container.

```
[root@localhost docker]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
212dce1e650e httpd "httpd-foreground" 18 seconds ago Up 17 seconds 80/tcp adoring_nash

root@212dce1e650e:/usr/local/apache2# hostname -I
172.17.0.3
root@212dce1e650e:/usr/local/apache2# cat /etc/hosts
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00:: ip6-localnet
ff00:: ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.3      212dce1e650e
root@212dce1e650e:/usr/local/apache2# |

[root@localhost docker]# docker exec -it adoring_nash "hostname"
212dce1e650e
[root@localhost docker]#
```

13. Monitor Container Stats

Inspect a container created with the lancachenet/ubuntu-nginx image.

Monitor its resource usage and stats using the top command.

```
[root@localhost docker]# docker container run -itd lancachenet/ubuntu-nginx
Unable to find image 'lancachenet/ubuntu-nginx:latest' locally
latest: Pulling from lancachenet/ubuntu-nginx
cfc57d2112d89: Already exists
9003887337ec: Already exists
97b162366492: Already exists
dd5c7a17daaa: Already exists
86154565fc0c: Pull complete
d91d1b6e30c9: Pull complete
cb9ce1fa8cf0: Pull complete
Digest: sha256:f3e165aebe33ae26261e8a31a1e19104f3c0775307170857cbffe18089276badc
Status: Downloaded newer image for lancachenet/ubuntu-nginx:latest
d8592cd9cde0f07f7468710e566e3a1c06858550a0ef67fc8a01260e927ab74
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d8592cd9cde0f07f7468710e566e3a1c06858550a0ef67fc8a01260e927ab74
[root@localhost docker]# docker top wonderful_dhawan
UID PID PPID C STIME TTY TIME CMD
root 408977 408977 1 15:25 pts/0 00:00:00 /usr/bin/python3 /usr/bin/supervisord -n -c /etc/supervisor/supervisord.conf -e WARN
root 408988 40877 0 15:25 pts/0 00:00:00 /bin/bash /opt/nginx/startnginx.sh
root 408989 408988 0 15:25 pts/0 00:00:00 nginx: master process /usr/sbin/nginx -g daemon off;
33 408910 40999 0 15:25 pts/0 00:00:00 nginx: worker process
[root@localhost docker]#
```

14. Nginx Containers Log Review

Start a new Nginx container.

Review and compare the logs for both Nginx containers.

```
[root@localhost docker]# docker container run -itd nginx
f634b894d7a68e8439703b5b690bf53298ed6f29ef5b4530320c33e4ad49b
[root@localhost docker]# docker logs vigilant_almeida
Executing hook /hooks/entrypoint-pre.d/00_asiclologo.sh
[...]
Executing hook /hooks/supervisord-pre.d/00_log_files
Checking if /var/www/ is empty Directory not empty.. don't touch content
Executing hook /hooks/supervisord-pre.d/21_cleanup_log_files
Cleaning up log files older than 7 days
Starting Supervisord...
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f634b894d67a nginx "/docker-entrypoint..." 18 seconds ago Up 18 seconds 80/tcp peaceful_ca
none
3dafe06cecb9a2 lancachenet/ubuntu-nginx:latest "/bin/bash -e /init/" 2 minutes ago Up 2 minutes 0.0.0.0:80->80/tcp, [::]:80->80/tcp vigilant_almeida
[root@localhost docker]# docker logs peaceful_cannon
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/11/05 10:01:01 [notice] #1: using the "epoll" event method
2025/11/05 10:01:01 [notice] #1: nginx/1.29.3
2025/11/05 10:01:01 [notice] #1: built by gcc 14 2.0.0 (Debian 14.2.0-19)
2025/11/05 10:01:01 [notice] #1: OS: Linux 5.14.0-624.el9.x86_64
2025/11/05 10:01:01 [notice] #1: getrlimit(RLIMIT_NOFILE): 1073741816:1073741816
2025/11/05 10:01:01 [notice] #1: start worker processes
2025/11/05 10:01:01 [notice] #1: start worker process 29
[root@localhost docker]#
```

15. System Events & Filters

Examine Docker system events filtered by:

Specific date, and

The last 30 minutes.

Apply two filters simultaneously using name and event.

```
[root@localhost docker]# docker events --since "2025-11-05T15:00:00" --until "2025-11-05T15:30:00" --filter "name=my_container_name" --filter "event=start"
2025-11-05T15:03:35.682039513+05:30 container start 13500c347f2264e34a9718b7086ea/d87b0650c9aeab40063aa186152ef9881d33 (image=ubuntu-nginx-server:latest, main
tainer=LanCache.Net Team <team@lancache.net>, name=nginx-cont, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:05:43.672039513+05:30 container start e2dfccbe3403dc4b2bc139245d66d15777b5deeb1314566a69f270a19f0b663 (image=ubuntu-nginx-server:latest, main
tainer=LanCache.Net Team <team@lancache.net>, name=nginx-cont, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:07:29.672039513+05:30 container start 29ab60085c181af58838255155159595cfc0928584c4864988545 (image=ubuntu-nginx-server:latest, main
tainer=LanCache.Net Team <team@lancache.net>, name=eager_beast, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:10:29.963122089+05:30 container start 329504004942d7f07a923841f1f05669895ff1b174c5605794f6f569df0075b4b0 (image=ubuntu, name=condescending_poir
as, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:14:54.801365569405+05:30 container start 250cd149b15c3ed73be:03a78753666f97e024454c9cf0317d822253b67d9c (image=ubuntu, name=peaceful_hopper, or
g.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:18:35.685080682405+05:30 container start 989bebeads957f7b8326461f291854edca61cd63d97b621f981d66f863766257 (image=httpd, name=xenodochial_goodall)
2025-11-05T15:20:12.225659724+05:30 container start 212dc1e65ba11f7f73398b05ceea839ef270c42e8c83a9e5aaeaf6f0c9e05 (image=httpd, name=adoring_nash)
2025-11-05T15:25:44.460458146+05:30 container start d8592cd9c1e6f0f07f7468710e5663a1c0685855099a0ef67fc8a01260e927ab74 (image=lancachenet/ubuntu-nginx, mainta
iner=LanCache.Net Team <team@lancache.net>, name=wonderful_dhawan, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:28:54.499573925+05:30 container start 3dafe06cecb9a2525717d7ba04uf0ff4784f43bbd4326bf6f227b69e321061a2f9ff (image=lancachenet/ubuntu-nginx:latest,
maintainer=LanCache.Net Team <team@lancache.net>, name=wonderful_dhawan, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T15:29:33.976669880+05:30 container start 61b03426e0b2aa278e455045f7d0623c9a5cca0db883635c6749e4fc3990d282 (image=nginx, maintainer=NGINX Docker M
aintainers <docker-maint@nginx.com>, name=loving_bohr)
[root@localhost docker]#
```



```
[root@localhost docker]# docker events \
--since "2025-11-05T00:00:00" \
--until "2025-11-05T23:59:59" \
--filter "event=start" \
--filter "container=peaceful_cannon"
2025-11-05T15:31:01.316766363+05:30 container start f634b894d7a68e8439703b5b690bf53298ed6f29ef5b4530320c33e4ad49b (image=nginx, maintainer=NGINX Docker M
aintainers <docker-maint@nginx.com>, name=peaceful_cannon)
|
```

16. Custom Hostname & Logs

Create an Ubuntu container with:

A meaningful name, and

Hostname set to cloudethix.com.

Stop or kill the container.

Check its exit code and review its logs for details.

```
[root@localhost docker]# docker container run -itd --name cloudethix-cont --hostname cloudethix.com ubuntu
34ae1ce6f80f706726a28f2a886ca2f7270533c07473c0887dcac5eaa4f3c76d
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
34ae1ce6f80f "bin/bash" 3 seconds ago Up 2 seconds
ubuntu
f634b894dd7a "nginx" 10 minutes ago Up 19 minutes 80/tcp
peaceful_cannon
3daa06ceb8a2 "lanchonet/ubuntu-nginx:latest" "/bin/bash -e /init/" 21 minutes ago Up 21 minutes 0.0.0.0:80->80/tcp, [::]:80->80/tcp
vigilant_almeida
```

```
[root@localhost docker]# docker container stop cloudethix-cont
cloudethix-cont
[root@localhost docker]# docker container inspect cloudethix-cont
[
  {
    "Id": "34ae1ce6f80f706726a28f2a886ca2f7270533c07473c0887dcac5eaa4f3c76d",
    "Created": "2025-11-05T10:20:14.731290964Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 137,
      "Error": "",
      "StartedAt": "2025-11-05T10:20:14.832508976Z",
      "FinishedAt": "2025-11-05T10:20:43.648303637Z"
    },
    ...
  }
]
```

```
[root@localhost docker]# lsof -i :80
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
docker-pr 43684 root 7u IPv4 106055 0t0 TCP *:http (LISTEN)
docker-pr 43689 root 7u IPv6 106056 0t0 TCP *:http (LISTEN)
[root@localhost docker]# docker container logs cloudethix-cont
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/11/05 10:29:50 [notice] 1#1: using the "epoll" event method
2025/11/05 10:29:50 [notice] 1#1: nginx/1.29.3
2025/11/05 10:29:50 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/11/05 10:29:50 [notice] 1#1: OS: Linux 5.14.0-624.el9.x86_64
2025/11/05 10:29:50 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1073741816:1073741816
2025/11/05 10:29:50 [notice] 1#1: start worker processes
2025/11/05 10:29:50 [notice] 1#1: start worker process 28
[root@localhost docker]#
```

17. Container Management Operations

Create and start an Ubuntu container with a random name, then rename it to Ubuntu01.

Launch another container named Ubuntu02.

Check the hostname for both containers.

Perform the following:

Pause and unpause Ubuntu02.

Stop, start, and restart Ubuntu01.

Inspect stats and system events for both.

Finally, kill and delete the containers, ensuring all created directories are removed.

```
[root@localhost docker]# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
ubuntu-nginx-server    latest   45ff1a8e9619  About an hour ago  195MB
httpd                latest   6a4fe18d08d2  30 hours ago   117MB
nginx                latest   d261fd19cb63  30 hours ago   152MB
lancachenet/ubuntu-nginx  latest   1d0a8df5d47  4 days ago    195MB
ubuntu                latest   97bed23a3497  4 weeks ago    78.1MB
[root@localhost docker]# docker container run -itd ubuntu
92459dbb1ec7503e241809ebfd999ad81ad1d754b564687327e79609f28d2bbe
[root@localhost docker]# docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED        STATUS        PORTS     NAMES
92459dbb1ec7  ubuntu      "/bin/bash"  3 seconds ago  Up 2 seconds
[root@localhost docker]# docker container rename reverent_feistel Ubuntu01
[root@localhost docker]# |
```

```
[root@localhost docker]# docker container run -itd --name Ubuntu02 ubuntu
9af5a5fcbb45064609903719ef675d656e7d81454081e74595a393cc16c429720
[root@localhost docker]# docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED        STATUS        PORTS     NAMES
9af5a5fcbb450  ubuntu      "/bin/bash"  2 seconds ago  Up 2 seconds
92459dbb1ec7  ubuntu      "/bin/bash"  About a minute ago  Up About a minute
[root@localhost docker]# |
```

```
[root@localhost docker]# docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED        STATUS        PORTS     NAMES
9af5a5fcbb450  ubuntu      "/bin/bash"  2 seconds ago  Up 2 seconds
92459dbb1ec7  ubuntu      "/bin/bash"  About a minute ago  Up About a minute
[root@localhost docker]#
[root@localhost docker]# docker exec -it Ubuntu01 hostname
92459dbb1ec7
[root@localhost docker]# docker exec -it Ubuntu02 hostname
9af5a5fcbb450
[root@localhost docker]# docker container attach Ubuntu01
root@92459dbb1ec7:/# hostname
92459dbb1ec7
root@92459dbb1ec7:/# read escape sequence
[root@localhost docker]# docker container attach Ubuntu02
root@9af5a5fcbb450:/# hostname
9af5a5fcbb450
root@9af5a5fcbb450:/# |
```

```
[root@localhost docker]# docker container pause Ubuntu02
Ubuntu02
[root@localhost docker]# docker container unpause Ubuntu02
Ubuntu02
[root@localhost docker]#
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d3f1f9ac9dcf ubuntu "/bin/bash" About a minute ago Up About a minute
92459dbb1ec7 ubuntu "/bin/bash" 6 minutes ago Up 6 minutes
[root@localhost docker]# docker container u
```

```
[root@localhost docker]# docker container stop Ubuntu01
Ubuntu01
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d3f1f9ac9dcf ubuntu "/bin/bash" About a minute ago Up About a minute
[root@localhost docker]# docker container start Ubuntu01
Ubuntu01
[root@localhost docker]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d3f1f9ac9dcf ubuntu "/bin/bash" About a minute ago Up About a minute
92459dbb1ec7 ubuntu "/bin/bash" 7 minutes ago Up 1 second
[root@localhost docker]# docker container restart Ubuntu01
Ubuntu01
[root@localhost docker]# |
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
d3f1f9ac9dcf	Ubuntu02	0.00%	1.508MiB / 1.668GiB	0.09%	2.16kB / 126B	24.6kB / 0B	2

```
[root@localhost docker]# docker system events
2025-11-05T16:13:08.965[73117+05:30] container kill d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8 (image=ubuntu, name=Ubuntu02, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04, signal=15)
2025-11-05T16:13:18.994[025973+05:30] container kill d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8 (image=ubuntu, name=Ubuntu02, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04, signal=15)
2025-11-05T16:13:18.994[025973+05:30] network disconnect 0c0307888783336e0b262e05aca4adc746b409bc7ba0c07f627cedbf0c013413 (container=d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8, name=bridge, type=bridge)
2025-11-05T16:13:19.188375510+05:30 container stop d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8 (image=ubuntu, name=Ubuntu02, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T16:13:19.188375510+05:30 container stop d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8 (execDuration=38, exitCode=137, image=ubuntu, name=Ubuntu02, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T16:13:19.387124944+05:30 network connect 0c0307888783336e0b262e05aca4adc746b409bc7ba0c07f627cedbf0c013413 (container=d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8, name=bridge, type=bridge)
2025-11-05T16:13:19.415917162+05:30 container start d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8 (image=ubuntu, name=Ubuntu02, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
2025-11-05T16:13:19.415937340+05:30 container restart d3f1f9ac9dcbf337751cf131e1c988c7bc4b86edeb0b03824b1b33ee28090e8 (image=ubuntu, name=Ubuntu02, org.opencontainers.image.ref.name=ubuntu, org.opencontainers.image.version=24.04)
```

```
[root@localhost docker]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d3f1f9ac9dcf ubuntu "/bin/bash" 7 minutes ago Up About a minute Ubuntu02
92459dbb1ec7 ubuntu "/bin/bash" 12 minutes ago Up 4 minutes Ubuntu01
[root@localhost docker]# docker container rm $(docker container stop $(docker ps -aq))
d3f1f9ac9dcf
92459dbb1ec7
[root@localhost docker]# |
```

18. DockerHub Repository Setup

Create a DockerHub account.

Set up a repository with a meaningful name.

Repositories / [cloudethix](#) / General

ajinkyap009/cloudethix ⓘ

Created in less than a minute • ⭐0 • ↓0

Repository to have images for tasks performed in CloudEthix. ⚪

Add a category ⚪ ⓘ

General Tags Image Management BETA Collaborators Webhooks Settings

Tags ⓘ INCOMPLETE ⚪ DOCKER SCOUT INACTIVE [Activate](#)

Pushed images appear here.

Tasks 03:

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.

```
root@Ajinkyapame:~# docker container run -d --name no-container -p 80:80 --restart=never httpd
root@Ajinkyapame:~# docker container run -d --name on-failure-container -p 90:80 --restart on-failure httpd
e29b7bb641c9e23ab46dd6770424a64f8ea01f1809b78d198507f4ddab52b
root@Ajinkyapame:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e29b7bb641c9 httpd "httpd-foreground" 13 seconds ago Up 13 seconds 0.0.0.0:90->80/tcp, [::]:90->80/tcp on-failure-container
46c93f4041ec httpd "httpd-foreground" 43 seconds ago Up 43 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp no-container
root@Ajinkyapame:~# docker container run -d --name always-container -p 8080:80 --restart always httpd
464c3c434934c851399ac5a53764135aa182aa3160c56cd3bd2cc4b453baabe3
root@Ajinkyapame:~# docker container run -d --name unless-container -p 9090:80 --restart unless-stopped httpd
8a2470a6e5af597dc36645b297fd6373547c31691512e5b58bf483f04be3f22f8
root@Ajinkyapame:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a2470a6e5af httpd "httpd-foreground" 3 seconds ago Up 2 seconds 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp unless-container
464c3c434934 httpd "httpd-foreground" 12 minutes ago Up 12 minutes 0.0.0.0:8880->80/tcp, [::]:8880->80/tcp always-container
e29b7bb641c9 httpd "httpd-foreground" 13 minutes ago Up 13 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp on-failure-container
46c93f4041ec httpd "httpd-foreground" 14 minutes ago Up 14 minutes 0.0.0.0:80->80/tcp, [::]:80->80/tcp no-container
root@Ajinkyapame:~#
```

Check for NO:

```
root@AjinkyaPame:~# systemctl restart docker.service
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 18 minutes ago Up 1 second 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container on-failure-container no-container
464c3c434934 httpd "httpd-foreground" 31 minutes ago Up 1 second
e29b7bb641c9 httpd "httpd-foreground" 32 minutes ago Exited (0) 5 minutes ago 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
46c93f4041ec httpd "httpd-foreground" 32 minutes ago Exited (0) 5 minutes ago
root@AjinkyaPame:~#
```

Check for always:

```
root@AjinkyaPame:~# sudo systemctl restart docker
docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 13 minutes ago Up Less than a second 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container
464c3c434934 httpd "httpd-foreground" 25 minutes ago Up Less than a second
root@AjinkyaPame:~# 
root@AjinkyaPame:~#
```

Check for On-failure:

```
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 34 minutes ago Exited (137) 48 seconds ago 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container on-failure-contai
464c3c434934 httpd "httpd-foreground" 47 minutes ago Up 16 minutes
e29b7bb641c9 httpd "httpd-foreground" 48 minutes ago Exited (137) About a minute ago
ner
46c93f4041ec httpd "httpd-foreground" 48 minutes ago Exited (0) 21 minutes ago
root@AjinkyaPame:~# systemctl restart docker.service
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 35 minutes ago Exited (137) About a minute ago 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp 0.0.0.0:90->80/tcp, [::]:90->80/tcp unless-container always-container on-failure-contai
464c3c434934 httpd "httpd-foreground" 47 minutes ago Up 2 seconds
e29b7bb641c9 httpd "httpd-foreground" 48 minutes ago Up 2 seconds
ner
46c93f4041ec httpd "httpd-foreground" 49 minutes ago Exited (0) 22 minutes ago
root@AjinkyaPame:~#
```

Check for unless-stopped:

```
root@AjinkyaPame:~# docker exec unless-container /bin/sh -c "kill -s TERM 1"
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 38 minutes ago Up 1 second 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container on-failure-contai
464c3c434934 httpd "httpd-foreground" 50 minutes ago Up 2 minutes
e29b7bb641c9 httpd "httpd-foreground" 51 minutes ago Up 2 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp
46c93f4041ec httpd "httpd-foreground" 51 minutes ago Exited (0) 24 minutes ago
no-container
root@AjinkyaPame:~#
```

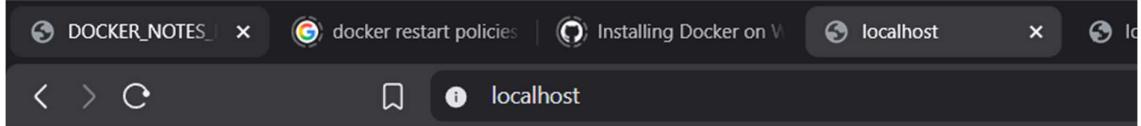
20. Change the restart policy of a above running container from the default to a custom policy using the docker update command.

e.g. docker update --help

```
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 41 minutes ago Up 3 minutes 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container on-failure-contai
464c3c434934 httpd "httpd-foreground" 53 minutes ago Up 6 minutes
e29b7bb641c9 httpd "httpd-foreground" 54 minutes ago Up 6 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp
46c93f4041ec httpd "httpd-foreground" 55 minutes ago Exited (0) 28 minutes ago
no-container
root@AjinkyaPame:~# docker update --restart always no-container
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 41 minutes ago Up 3 minutes 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container on-failure-contai
464c3c434934 httpd "httpd-foreground" 53 minutes ago Up 6 minutes
e29b7bb641c9 httpd "httpd-foreground" 54 minutes ago Up 6 minutes 0.0.0.0:90->80/tcp, [::]:90->80/tcp
46c93f4041ec httpd "httpd-foreground" 55 minutes ago Exited (0) 28 minutes ago
no-container
root@AjinkyaPame:~# systemctl restart docker.service
root@AjinkyaPame:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a247bae65af httpd "httpd-foreground" 41 minutes ago Up 2 seconds 0.0.0.0:9090->80/tcp, [::]:9090->80/tcp 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp unless-container always-container on-failure-contai
464c3c434934 httpd "httpd-foreground" 54 minutes ago Up 2 seconds
e29b7bb641c9 httpd "httpd-foreground" 55 minutes ago Exited (0) 2 seconds ago 0.0.0.0:90->80/tcp, [::]:90->80/tcp
46c93f4041ec httpd "httpd-foreground" 55 minutes ago Up 2 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp
no-container
root@AjinkyaPame:~#
```

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.

```
root@AjinkyaPame:~/Docker# docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
httpd          latest   6a4fe18d08d2  2 days ago   117MB
nginx          latest   d261fd19cb63  2 days ago   152MB
docker-compose-db latest   890f0a9fcada  6 months ago  859MB
root@AjinkyaPame:~/Docker# nano index.html
root@AjinkyaPame:~/Docker# docker container run -d --name nginx-cont -p 80:80 nginx:latest
c751f37f85f8b2e7acecd03c9a1bdb2b1482a2c9f081d5a5df33cb204e0089c5
root@AjinkyaPame:~/Docker# docker cp index.html c751f37f85f8b2e:/usr/share/nginx/html/
Successfully copied 2.05kB to c751f37f85f8b2e:/usr/share/nginx/html/
root@AjinkyaPame:~/Docker# |
```



Hello There, Let's be the Team CloudEthiX!!

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

docker container run --help

ref links :-

<https://phoenixnap.com/kb/docker-memory-and-cpu-limit>

<https://www.baeldung.com/ops/docker-memory-limit>

```
root@AjinkyaPame:~/Docker# docker container run -d --memory=512m --cpu
--cpu-count      (CPU count (Windows only))           --cpu-rt-runtime  (Limit CPU real-time runtime in microseconds)
--cpu-percent    (CPU percent (Windows only))         --cpu-shares     (CPU shares (relative weight))
--cpu-period     (Limit CPU CFS (Completely Fair Scheduler) period)  --cpus          (Number of CPUs)
--cpu-quota      (Limit CPU CFS (Completely Fair Scheduler) quota)   --cpuset-cpus    (CPUs in which to allow execution (0-3, 0,1))
--cpu-rt-period   (Limit CPU real-time period in microseconds)       --cpuset-mems   (MEMs in which to allow execution (0-3, 0,1))
root@AjinkyaPame:~/Docker# docker container run -dit --memory=512m --cpus=2 nginx:latest
485216a5798ee3838b7b5f020bf21ef19a9265aced5d1f037ff8834a9b42401
root@AjinkyaPame:~/Docker# docker container inspect | grep -i Memory
always-container elegant_jang      nginx-cont      no-container      on-failure-test      unless-container
root@AjinkyaPame:~/Docker# docker container inspect | grep -i Memory
always-container elegant_jang      nginx-cont      no-container      on-failure-test      unless-container
root@AjinkyaPame:~/Docker# docker container inspect elegant_jang | grep -i Memory
        "Memory": 536870912,
        "MemoryReservation": 0,
        "MemorySwap": 1073741824,
        "MemorySwappiness": null,
root@AjinkyaPame:~/Docker# docker container inspect elegant_jang | grep -i cpu
        "CpuShares": 0,
        "NanoCpus": 2000000000,
        "CpuPeriod": 0,
        "CpuQuota": 0,
        "CpuRealtimePeriod": 0,
        "CpuRealtimeRuntime": 0,
        "CpusetCpus": "",
        "CpusetMems": "",
        "CpuCount": 0,
        "CpuPercent": 0,
root@AjinkyaPame:~/Docker# |
```

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

ref links :-

<https://docs.docker.com/engine/reference/commandline/update/>

```
root@AjinkyaPame:~/Docker# docker container update --memory=1024m --cpus=1.5 elegant_jang
elegant_jang
root@AjinkyaPame:~/Docker# docker container inspect elegant_jang | grep -i cpu
    "CpuShares": 0,
    "NanoCpus": 1500000000,
    "CpuPeriod": 0,
    "CpuQuota": 0,
    "CpuRealtimePeriod": 0,
    "CpuRealtimeRuntime": 0,
    "CpusetCpus": "",
    "CpusetMems": "",
    "CpuCount": 0,
    "CpuPercent": 0,
root@AjinkyaPame:~/Docker# docker container inspect elegant_jang | grep -i Memory
    "Memory": 1073741824,
    "MemoryReservation": 0,
    "MemorySwap": 1073741824,
    "MemorySwappiness": null,
root@AjinkyaPame:~/Docker# |
```

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

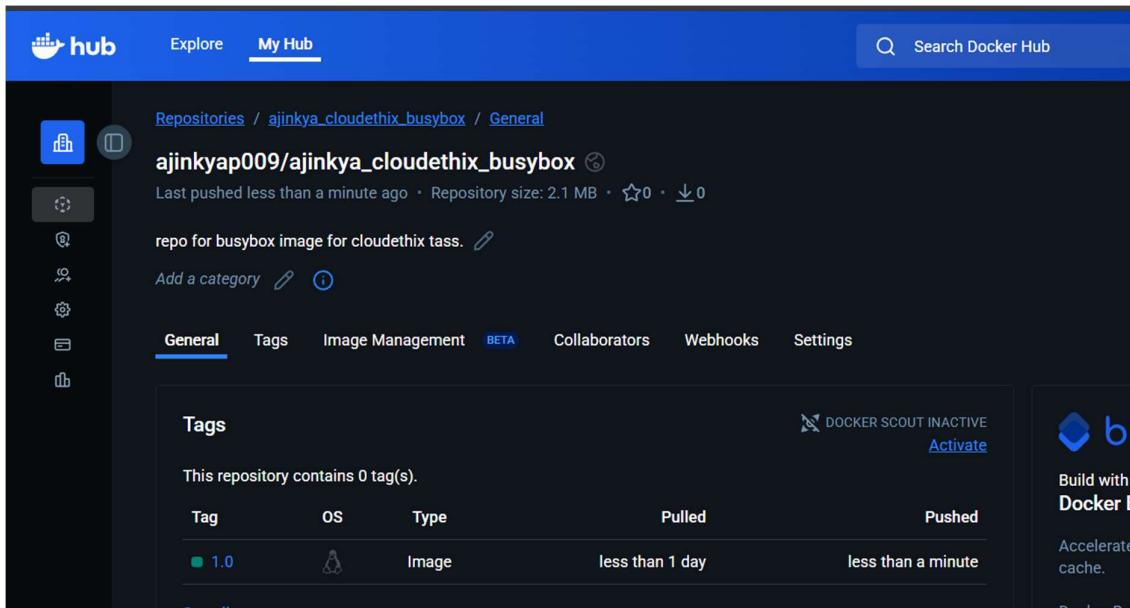
```
root@AjinkyaPame:~/Docker# docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:e3652a00a2fabd16ce889f0aa32c38eec347b997e73bd09e69c962ec7f8732ee
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest
root@AjinkyaPame:~/Docker# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
httpd              latest   6a4fe18d08d2  2 days ago   117MB
nginx              latest   d261fd19cb63  2 days ago   152MB
docker-compose-db  latest   890f0a9fcada  6 months ago  859MB
busybox             latest   08ef35a1c3f0  13 months ago  4.43MB
root@AjinkyaPame:~/Docker# docker tag busybox:latest ajinkyap009/ajinkya_cloudehix_busybox:1.0
```

```
root@AjinkyaPame:~/Docker# docker login
USING WEB-BASED LOGIN
Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: KGXZ-CLXD
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...
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@AjinkyaPame:~/Docker# docker push ajinkyap009/ajinkya_cloudehix_busybox:1.0
The push refers to repository [docker.io/ajinkyap009/ajinkya_cloudehix_busybox]
e14542cc0629: Mounted from library/busybox
1.0: digest: sha256:755d9ce09782b2f60fd3321878e611f871817aa8a726e8c605f2f67ae6ffa9e2 size: 527
root@AjinkyaPame:~/Docker# |
```



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.

```
root@AjinkyaPame:~/Docker# mkdir TASK_25
root@AjinkyaPame:~/Docker# cd TASK_25/
root@AjinkyaPame:~/Docker/TASK_25#
root@AjinkyaPame:~/Docker/TASK_25# nano Dockerfile
root@AjinkyaPame:~/Docker/TASK_25# nano custom-index.html
```

```
root@AjinkyaPame:~/Docker/TASK_25# docker build -t ajinkyap009/custom-nginx:01 .
[+] Building 0.1s (7/7) FINISHED
--> [internal] Load build definition from Dockerfile
--> => transferring dockerfile: 11B
--> [internal] Load metadata for docker.io/library/nginx:latest
--> [internal] Load .dockerignore
--> => transferring context: 2B
--> [internal] Load build context
--> => transferring context: 38B
--> [1/2] FROM docker.io/library/nginx:latest
--> CACHED [2/2] COPY custom-index.html /usr/share/nginx/html/index.html
--> exporting image
--> => exporting layers
--> => writing image sha256:c455ab38add073441eb42594b1aa12a0cbf64453d992457f2df6be58e23ea8a2
--> => naming to docker.io/ajinkyap009/custom-nginx:01
```

```

root@AjinkyaPame:~/Docker/TASK_25# docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
ajinkyap009/custom-nginx    01      c455ab38add0  About a minute ago  152MB
httpd              latest   6a4fe18d08d2  2 days ago   117MB
nginx              latest   d261fd19cb63  2 days ago   152MB
docker-compose-db  latest   890f0a9fcada  6 months ago  859MB
ajinkyap009/ajinkya_clouethix_busybox  1.0     08ef35a1c3f0  13 months ago  4.43MB
busybox             latest   08ef35a1c3f0  13 months ago  4.43MB
root@AjinkyaPame:~/Docker/TASK_25# docker push ajinkyap009/custom-nginx:01
The push refers to repository [docker.io/ajinkyap009/custom-nginx]
3a13a0513f8c: Pushed
d7217c60dca4: Pushed
d81df94f8d07: Pushed
99cd1b1b6a43: Pushed
2ced4cd78a7b: Pushed
8feb164cd673: Pushed
6e19587ac541: Pushed
36d06fe0cbc6: Pushed
01: digest: sha256:1770c3f725e6d09e0b2a8e76c9e45c22c57b71c4af3c3482375c958d779e657d size: 1985
root@AjinkyaPame:~/Docker/TASK_25#

```

```

root@AjinkyaPame:~/Docker/TASK_25# docker rmi ajinkyap009/custom-nginx:01
Untagged: ajinkyap009/custom-nginx:01
Untagged: ajinkyap009/custom-nginx@sha256:1770c3f725e6d09e0b2a8e76c9e45c22c57b71c4af3c3482375c958d779e657d
Deleted: sha256:c455ab38add073441eb42504b1aa12a0cbf64453d992457f2df6be58e23ea8a2
root@AjinkyaPame:~/Docker/TASK_25# docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
httpd              latest   6a4fe18d08d2  2 days ago   117MB
nginx              latest   d261fd19cb63  2 days ago   152MB
docker-compose-db  latest   890f0a9fcada  6 months ago  859MB
busybox             latest   08ef35a1c3f0  13 months ago  4.43MB
ajinkyap009/ajinkya_clouethix_busybox  1.0     08ef35a1c3f0  13 months ago  4.43MB
root@AjinkyaPame:~/Docker/TASK_25# lsof -i :8080

```

```

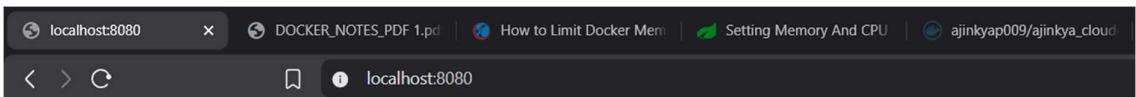
root@AjinkyaPame:~/Docker/TASK_25# docker container run -d --name custom-nginx-cont -p 8080:80 ajinkyap009/custom-nginx:01
Unable to find image 'ajinkyap009/custom-nginx:01' locally
01: Pulling from ajinkyap009/custom-nginx
fb60efab8139: Already exists
266626526d42: Already exists
320b0949be89: Already exists
d921c57c6a81: Already exists
9def903993e4: Already exists
52bc359bcd7d: Already exists
e2f8e296d9df: Already exists
ed0870114318: Already exists
Digest: sha256:1770c3f725e6d09e0b2a8e76c9e45c22c57b71c4af3c3482375c958d779e657d
Status: Downloaded newer image for ajinkyap009/custom-nginx:01
e6c67bbc0797ccae8ee5381bdba9a78f232e4edf47b8be269b632f1313d6d52

```

```

root@AjinkyaPame:~/Docker/TASK_25# docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
ajinkyap009/custom-nginx    01      c455ab38add0  10 minutes ago  152MB
httpd              latest   6a4fe18d08d2  2 days ago   117MB
nginx              latest   d261fd19cb63  2 days ago   152MB
docker-compose-db  latest   890f0a9fcada  6 months ago  859MB
ajinkyap009/ajinkya_clouethix_busybox  1.0     08ef35a1c3f0  13 months ago  4.43MB
busybox             latest   08ef35a1c3f0  13 months ago  4.43MB
root@AjinkyaPame:~/Docker/TASK_25# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
e6c67bbc0797        ajinkyap009/custom-nginx:01   "/docker-entrypoint..."   11 seconds ago    Up 10 seconds    0.0.0.0:8080->80/tcp, :::8080->80/tcp   custom-nginx-cont
root@AjinkyaPame:~/Docker/TASK_25#

```



This is nan-index.html file. Image can be found at <https://hub.docker.com/repository/docker/ajinkyap009/clouethix/general>

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

```
root@AjinkyaPame:~/Docker/TASK_25# docker search redis
NAME                           DESCRIPTION                                     STARS      OFFICIAL
redis                          Redis is the world's fastest data platform f... 13446      [OK]
redis/redis-stack-server        redis-stack-server installs a Redis server w... 98          
```

```
root@AjinkyaPame:~/Docker/TASK_25# docker pull redis
Using default tag: latest
latest: Pulling from library/redis
1adabd6b0d6b: Pull complete
22506777a096: Pull complete
5dec27664782: Pull complete
2e60b18d70d8: Pull complete
729797e636a7: Pull complete
4f4fb700ef54: Pull complete
6c981fc7c621: Pull complete
Digest: sha256:5c7c0445ed86918cb9efb96d95a6bfc03ed2059fe2c5f02b4d74f477ffe47915
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
root@AjinkyaPame:~/Docker/TASK_25# |
```

```
root@AjinkyaPame:~/Docker/TASK_25# docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
ajinkyap009/custom-nginx    01      c455ab38add0   27 minutes ago  152MB
httpd               latest   6a4fe18d08d2   2 days ago   117MB
nginx               latest   d261fd19cb63   2 days ago   152MB
redis               latest   017b1c12abf9   2 days ago   137MB
docker-compose-db   latest   890f0a9fcada   6 months ago  859MB
ajinkyap009/ajinkya_cloudehix_busybox  1.0     08ef35a1c3f0   13 months ago  4.43MB
busybox              latest   08ef35a1c3f0   13 months ago  4.43MB
root@AjinkyaPame:~/Docker/TASK_25# docker tag redis:latest ajinkyap009/ajinkya_cloudehix_redis:v1
root@AjinkyaPame:~/Docker/TASK_25# docker tag redis:latest ajinkyap009/ajinkya_cloudehix_redis:v3
root@AjinkyaPame:~/Docker/TASK_25# docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
ajinkyap009/custom-nginx    01      c455ab38add0   28 minutes ago  152MB
httpd               latest   6a4fe18d08d2   2 days ago   117MB
nginx               latest   d261fd19cb63   2 days ago   152MB
redis               latest   017b1c12abf9   2 days ago   137MB
ajinkyap009/ajinkya_cloudehix_redis   v1      017b1c12abf9   2 days ago   137MB
ajinkyap009/ajinkya_cloudehix_redis   v3      017b1c12abf9   2 days ago   137MB
docker-compose-db   latest   890f0a9fcada   6 months ago  859MB
ajinkyap009/ajinkya_cloudehix_busybox  1.0     08ef35a1c3f0   13 months ago  4.43MB
busybox              latest   08ef35a1c3f0   13 months ago  4.43MB
root@AjinkyaPame:~/Docker/TASK_25# |
```

```
root@AjinkyaPame:~/Docker/TASK_25# docker push ajinkyap009/ajinkya_cloudehix_redis:v1
The push refers to repository [docker.io/ajinkyap009/ajinkya_cloudehix_redis]
4c9d174d85be: Mounted from library/redis
5f70bf18a086: Mounted from ajinkyap009/docker-1
4a34755eccccc: Mounted from library/redis
a4f8354a4eb1: Mounted from library/redis
fd0bc835e87b: Mounted from library/redis
e7fe7688c8dc: Mounted from library/redis
b512279032aa: Mounted from library/redis
v1: digest: sha256:b3f78d38d1ba4758461d84a19af14b8d13c4a44c145b70efb0749418566b6f55 size: 1776
root@AjinkyaPame:~/Docker/TASK_25# docker push ajinkyap009/ajinkya_cloudehix_redis:v3
The push refers to repository [docker.io/ajinkyap009/ajinkya_cloudehix_redis]
4c9d174d85be: Layer already exists
5f70bf18a086: Layer already exists
4a34755eccccc: Layer already exists
a4f8354a4eb1: Layer already exists
fd0bc835e87b: Layer already exists
e7fe7688c8dc: Layer already exists
b512279032aa: Layer already exists
v3: digest: sha256:b3f78d38d1ba4758461d84a19af14b8d13c4a44c145b70efb0749418566b6f55 size: 1776
```

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

ajinkyp009/ajinkya_cloudeithix_redis ⚡

Last pushed less than a minute ago • ⭐0 • ↓0

Add a description Add a category

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

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v3		Image	less than 1 day	1 minute
v1		Image	less than 1 day	2 minutes

DOCKER SCOUT INACTIVE [Activate](#)

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_cloudeithix_nginx." Finally, push the release branch to the remote Git repository and merge it by creating a pull request.

AjinkyaP-09 / ajinka-cloudeithix-nginx

Type

ajinka-cloudeithix-nginx Public 0 0

main 1 Branch 0 Tags Go to file About

AjinkyaP-09 Merge pull request #1 from AjinkyaP-09/release 092942c · 1 minute ago 5 Commits

Dockerfile Remove nginx base image and index.html copy 1 minute ago

index.html Remove greeting from index.html 1 minute ago

README

About

- No description
- Activity
- 0 stars
- 0 watching
- 0 forks

Releases

No releases published

```

root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git branch release
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git checkout release
Switched to branch 'release'
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx#
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# ls
Dockerfile index.html
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# nano Dockerfile
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# nano index.html
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git status
On branch release
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Dockerfile
    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

```

```

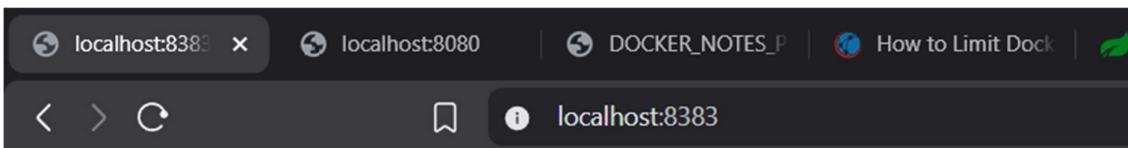
root@AjinkyaPame:~/Docker# cd ajinkya-cloudeithix-nginx/
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git branch
* main
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git branch release
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git branch
* main
  release
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# docker build -t release-branch-img .
[+] 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: 153B
--> CACHE [1/2] FROM docker.io/library/nginx:latest
--> [2/2] COPY index.html /usr/share/nginx/html/index.html
--> exporting to image
--> exporting layers
--> => writing image sha256:27feb3f2f417259b0d13d7153a4e927a6da5b328149daa898e0f6679b5fb5138
--> => naming to docker.io/library/release-branch-img
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
release-branch-img     latest   27feb3f2f417   3 seconds ago  152MB

```

```

root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# docker container run -dit --name release-nginx -p 8383:80 release-branch-img
e60f34e69503c2f8089e7f87d1977ce8f1df1806f1086d22dc941a8c87ae3be
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
e60f34e69503        release-branch-img   "/docker-entrypoint..."   3 seconds ago       Up 3 seconds      0.0.0.0:8383->80/tcp, [::]:8383->80/tcp   release-nginx
e6c67bbc0797        ajinkyap009/custom-nginx:01   "/docker-entrypoint..."   50 minutes ago     Up 50 minutes     0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   custom-nginx
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# 

```



Hello from ajinkya-cloudeithix-nginx.

```

root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# docker image tag release-branch-img:latest ajinkyap009/ajinkya_cloudeithix_nginx:01
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# docker push ajinkyap009/ajinkya_cloudeithix_nginx:01
The push refers to repository [docker.io/ajinkyap009/ajinkya_cloudeithix_nginx]
52c2e7ff96c: Pushed
d7217c60dcad: Mounted from ajinkyap009/custom-nginx
d81df94f8d07: Mounted from ajinkyap009/custom-nginx
99cd1b1b6a43: Mounted from ajinkyap009/custom-nginx
2ced4cd78a7b: Mounted from ajinkyap009/custom-nginx
8feb164cd673: Mounted from ajinkyap009/custom-nginx
6e19587ac541: Mounted from ajinkyap009/custom-nginx
36d06fe0cbc6: Mounted from ajinkyap009/custom-nginx
01: digest: sha256:98ae183b3ea0508498b8d1da3e98441b941682e0b6c2114234c455de542e81dd size: 1985
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# 

```

Repositories / ajinkya_cloudeithix_nginx / General

ajinkyap009/ajinkya_cloudeithix_nginx ⓘ

Last pushed 2 minutes ago • ⭐0 • ⌂0

Add a description ⓘ

Add a category ⓘ

General Tags Image Management BETA Collaborators Webhooks Settings

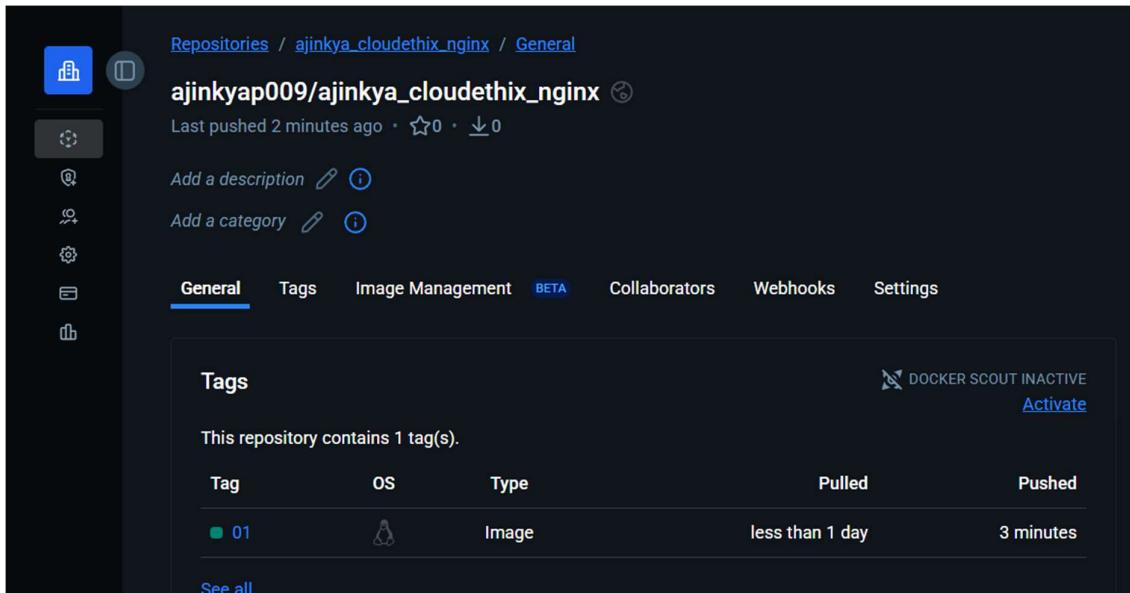
Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
01	Image	Image	less than 1 day	3 minutes

See all

DOCKER SCOUT INACTIVE
[Activate](#)



```
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git push origin release
Username for 'https://github.com': AJinkyaP-09
Password for 'https://AJinkyaP-09@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 465 bytes | 465.00 KiB/s, done.
Total 4 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'release' on GitHub by visiting:
remote:     https://github.com/AJinkyaP-09/ajinkya-cloudeithix-nginx/pull/new/release
remote:
To https://github.com/AJinkyaP-09/ajinkya-cloudeithix-nginx.git
 * [new branch]      release -> release
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# |
```

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. [Learn more about diff comparisons here.](#)

base: main ⏪ compare: release ⏪ Able to merge. These branches can be automatically merged.

Add a title

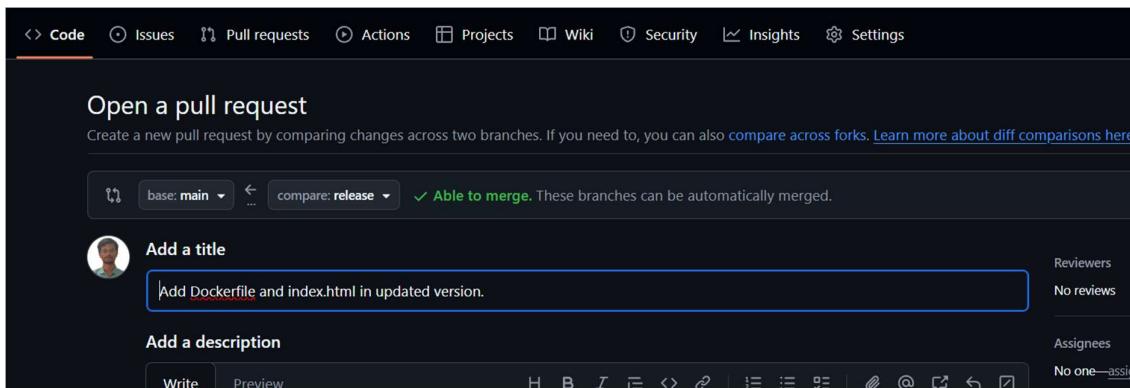
Add Dockerfile and index.html in updated version.

Add a description

Write Preview

Reviewers
No reviews

Assignees
No one assigned



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.

```
root@AjinkyaPame:~/Docker# cd ajinkya-cloudeithix-nginx/
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# mkdir redis-images
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# cd redis-images/
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx/redis-images# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
ajinkyap009/ajinkya_cloudeithix_nginx     01      27feb3f2f417   2 hours ago   152MB
release-branch-img      latest   27feb3f2f417   2 hours ago   152MB
ajinkyap009/custom-nginx      01      c455ab38add0   3 hours ago   152MB
httpd              latest   6a4fe18d08d2   2 days ago    117MB
nginx              latest   d261fd19cb63   2 days ago    152MB
ajinkyap009/ajinkya_cloudeithix_redis     v1      017b1c12abf9   2 days ago    137MB
ajinkyap009/ajinkya_cloudeithix_redis     v3      017b1c12abf9   2 days ago    137MB
redis              latest   017b1c12abf9   2 days ago    137MB
docker-compose-db      latest   890f0a9fcada   6 months ago   859MB
ajinkyap009/ajinkya_cloudeithix_busybox   1.0     08ef35a1c3f0   13 months ago   4.43MB
busybox             latest   08ef35a1c3f0   13 months ago   4.43MB
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx/redis-images# docker save -o "redis_v1.tar" ajinkyap009/ajinkya_cloudeithix_redis:v1
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx/redis-images# ll
total 137124
drwxr-xr-x  2 root root  4096 Nov  6 09:18 .
drwxr-xr-x  4 root root  4096 Nov  6 09:17 ..
-rw-----  1 root root 140402688 Nov  6 09:18 redis_v1.tar
```

```
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git checkout master
Switched to branch 'master'
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# mv ..redis-images/ .
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# ll
total 24
drwxr-xr-x  4 root root  4096 Nov  6 09:27 /
drwxr-xr-x  4 root root  4096 Nov  6 09:27 ../
drwxr-xr-x  7 root root  4096 Nov  6 09:27 .git/
-rw-r--r--  1 root root   67 Nov  6 09:26 Dockerfile
-rw-r--r--  1 root root  116 Nov  6 09:26 index.html
drwxr-xr-x  2 root root  4096 Nov  6 09:26 redis-images/
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git add .
root@AjinkyaPame:~/Docker/ajinkya-cloudeithix-nginx# git commit -m "Add redis images"
[master 2ff60ef] Add redis images
```

AjinkaP-09 / ajinka-cloudethix-nginxx

Type ⌘ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ajinka-cloudethix-nginxx Public

master had recent pushes 6 seconds ago Compare & pull request

master 3 Branches 0 Tags Go to file Add file Code

This branch is 3 commits ahead of, 1 commit behind main.

Contribute

root and root Add redis images 2ff60ef · 1 minute ago 9 Commits

- redis-images Add redis images 1 minute ago
- Dockerfile Add Dockerfile and index.html in updated version. 16 minutes ago
- index.html Add Dockerfile and index.html in updated version. 16 minutes ago

README

```
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx# docker rmi ajinkyp009/ajinka_cloudethix_redis:v1 ajinkyp009/ajinka_cloudethix_redis:v3
Untagged: ajinkyp009/ajinka_cloudethix_redis:v1
Untagged: ajinkyp009/ajinka_cloudethix_redis:v3
Untagged: ajinkyp009/ajinka_cloudethix_redis@sha256:b3f78d38d1ba4758461d84a19af144b8d13c4a44c145b70efb0749418566b6f55
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
ajinkyp009/ajinka_cloudethix_nginxx    01      27feb3f2f417   2 hours ago   152MB
release-image        latest   27feb3f2f417   2 hours ago   152MB
ajinkyp009/custom-nginx
httpd               latest   c455ab38add0   3 hours ago   152MB
nginx               latest   d261fd19cb63   2 days ago    152MB
redis               latest   6a4fe18d08d2   2 days ago    117MB
docker-compose-db   latest   898f0a9fcada   6 months ago   859MB
ajinkyp009/ajinka_cloudethix_busybox  1.0     08ef35a1c3f0   13 months ago   4.43MB
busybox              latest   08ef35a1c3f0   13 months ago   4.43MB
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx# docker rmi redis:latest
Untagged: redis:latest
Untagged: redis@sha256:5c7e0445ed86918cb9efb06d95a6bfcc03ed2859fe2c5f02bd474f477fffe47915
Deleted: sha256:017b1c12abf9d52fe40dd8a09195107ea2c566a0035b59ffcc7a18b4c32c736
Deleted: sha256:d6c88f219d7bbdf3086945edaef7d417bf581aa30a980ec085d6641244c2b9983
Deleted: sha256:0afb91539a5fe0cb899de0eae0309948c498b7a875eaff17e51e57bd55ae757
Deleted: sha256:46d3d7e3a243b7c7d1a333bfc801ceca82791717711bbf4659c2750e824f531c
Deleted: sha256:2d9a6c8a0cd1334df777d93674d34445251ddaa30bbbecfb03c787f305b678
Deleted: sha256:cccb6851c32678e7423eeef520376c1087f86a1b4bdc5a7f6136bc3a1ebe
Deleted: sha256:998362e5a9b1bb8eb0842b793ff6837c0ee71f26b64695d075c218b139281949
Deleted: sha256:b512279032aa717892385dd875fffcd96e4a4ad384225152522ac7bd5340a8
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx# 
```

```
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx# cd redis-images/
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# gunzip -c redis_v1.tar.gz > redis_v1.tar
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# gunzip -c redis_v3.tar.gz > redis_v3.tar
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# ll
total 375840
drwxr-xr-x 2 root root 4096 Nov 6 09:34 .
drwxr-xr-x 4 root root 4096 Nov 6 09:27 ..
-rw-r--r-- 1 root root 140402688 Nov 6 09:34 redis_v1.tar
-rw-r--r-- 1 root root 52018380 Nov 6 09:26 redis_v1.tar.gz
-rw-r--r-- 1 root root 140402688 Nov 6 09:34 redis_v3.tar
-rw-r--r-- 1 root root 52018389 Nov 6 09:26 redis_v3.tar.gz
```

```
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# docker load -i redis_v1.tar
b512279032aa: Loading layer [=====] 77.89MB/77.89MB
e7fe7688c8dc: Loading layer [=====] 10.75kB/10.75kB
fd0bc835e87b: Loading layer [=====] 10.75kB/10.75kB
a4f8354a4eb1: Loading layer [=====] 62.46MB/62.46MB
4a34755ecccc: Loading layer [=====] 1.536kB/1.536kB
5f70bf18a086: Loading layer [=====] 1.024kB/1.024kB
4c9d174d85be: Loading layer [=====] 8.192kB/8.192kB
Loaded image: ajinkyp009/ajinka_cloudethix_redis:v1
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# docker load -i redis_v3.tar
Loaded image: ajinkyp009/ajinka_cloudethix_redis:v3
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# docker images | grep redis
ajinkyp009/ajinka_cloudethix_redis      v1          017b1c12abf9   2 days ago   137MB
ajinkyp009/ajinka_cloudethix_redis      v3          017b1c12abf9   2 days ago   137MB
root@AjinkaPame:~/Docker/ajinka-cloudethix-nginxx/redis-images# 
```

29. Pull the Busy-box image to your local system, tag it, and push it to the Docker Hub repository "yourname_cloudethix_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_cloudethix_busybox" Docker Hub repository.

```
root@AjinkyaPame:~/Docker# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e60f34e69503 release-branch-img "/docker-entrypoint..." 2 hours ago Up 2 hours 0.0.0.0:8383->80/tcp, [::]:8383->80/tcp release-nginx
e6c67bbc0797 ajinkyap009/custom-nginx:01 "/docker-entrypoint..." 3 hours ago Up 3 hours 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp custom-nginx-cont
root@AjinkyaPame:~/Docker# docker export release-nginx > release-nginx.tar
root@AjinkyaPame:~/Docker# ll
total 150156
drwxr-xr-x 4 root root 4096 Nov 6 09:46 .
drwx----- 16 root root 4096 Nov 6 09:37 ..
drwxr-xr-x 2 root root 4096 Nov 6 06:24 TASK_25/
drwxr-xr-x 4 root root 4096 Nov 6 09:37 ajinkya-cloudethix-nginx/
-rw-r--r-- 1 root root 44 Nov 6 05:36 index.html
-rw-r--r-- 1 root root 153737216 Nov 6 09:46 release-nginx.tar

root@AjinkyaPame:~/Docker# docker import release-nginx.tar ajinkyap009/release-nginx:01
sha256:ba639795e8d8458e3f5fdb27ef9cb7ba95fcbb65c5e60b35049a227027612f99
root@AjinkyaPame:~/Docker# docker images | grep release
ajinkyap009/release-nginx          01      ba639795e8d8  11 seconds ago  150MB
release-branch-img                latest   27feb3f2f417  2 hours ago   152MB
root@AjinkyaPame:~/Docker# docker tag ajinkyap009/release-nginx:01 ajinkyap009/ajinkya_cloudethix_busybox:v2
root@AjinkyaPame:~/Docker# docker push ajinkyap009/ajinkya_cloudethix_busybox:v2
The push refers to repository [docker.io/ajinkyap009/ajinkya_cloudethix_busybox]
bb77e8957cdc: Pushed
v2: digest: sha256:294b9edecc35cbe14ac80518fd002ff1ea8ba6dea45ed81abf038069cff8d346 size: 528
root@AjinkyaPame:~/Docker# |
```

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

ajinkyap009/ajinkya_cloudethix_busybox ⓘ

Last pushed 1 minute ago • Repository size: 59.6 MB • ⭐0 • ⏺4

repo for busybox image for cloudethix tass. [🔗](#)

Add a category [🔗](#) [i](#)

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

Tags [🔗](#) DOCKER SCOUT INACTIVE [Activate](#)

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v2	BusyBox	Image	less than 1 day	1 minute
1.0	BusyBox	Image	less than 1 day	about 4 hours

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@AjinkyaPame:~/Docker/Task_30# nano Dockerfile
root@AjinkyaPame:~/Docker/Task_30# cat Dockerfile
#Base image: Ubuntu
FROM ubuntu:latest
#Install packages: curl, vim, and git
RUN apt update && apt install curl vim git -y

#Set an environment variable: MY_NAME=Your_Name
ENV MY_NAME=Ajinkya
root@AjinkyaPame:~/Docker/Task_30# docker build -t custom_ubuntu .
[+] Building 21.6s (6/6) FINISHED                               docker:default
=> [internal] load build definition from Dockerfile          0.0s
=> => transferring dockerfile: 231B                          0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.8s
=> [internal] load .dockerignore                            0.0s
=> => transferring context: 2B                           0.0s
=> CACHED [1/2] FROM docker.io/library/ubuntu:latest@sha256:66460d55 0.0s
=> [2/2] RUN apt update && apt install curl vim git -y      19.9s
=> exporting to image                                      0.9s
=> => exporting layers                                     0.8s
=> => writing image sha256:f144fc7da5942e743b1deea464f56d9c29356d911 0.0s
=> => naming to docker.io/library/custom_ubuntu           0.0s
root@AjinkyaPame:~/Docker/Task_30# docker images | grep ubuntu
custom_ubuntu          latest    f144fc7da594   19 seconds ago   289MB
root@AjinkyaPame:~/Docker/Task_30# |
```

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

```
root@AjinkyaPame:~/Docker# mkdir DHUB AWSECR
root@AjinkyaPame:~/Docker# ll
total 150168
drwxr-xr-x  7 root root      4096 Nov  6 10:11 .
drwxr----- 16 root root      4096 Nov  6 09:37 ..
drwxr-xr-x  2 root root      4096 Nov  6 10:11 AWSECR/
drwxr-xr-x  2 root root      4096 Nov  6 10:11 DHUB/
```

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.

```
root@AjinkyaPame:~/Docker# cd AWSECR/
root@AjinkyaPame:~/Docker/AWSECR# nano Dockerfile
root@AjinkyaPame:~/Docker/AWSECR# cat Dockerfile
# Base image: Ubuntu
FROM ubuntu:latest
# Install packages: httpd
RUN apt update && apt install apache2 -y
# Add "I am from AWS ECR" to the index.html file for AWS ECR directory.
RUN echo "I am from AWS ECR." > index.html
# Set environment variable ENV_NAME=AWSECR for the AWS ECR directory.
ENV ENV_NAME=AWSECR
# Start http service using ENTRYPOINT
# Expose port 80.
EXPOSE 80
ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]
```

```
root@AjinkyaPame:~/Docker/AWSECR# cat ../DHUB/Dockerfile
# Base image: Ubuntu
FROM ubuntu:latest
# Install packages: httpd
RUN apt update && apt install apache2 -y
# Add "I am from Docker Hub" to the index.html file for DHUB directory.
RUN echo "I am from Docker Hub." > index.html
# Set environment variable ENV_NAME=DHUB for the DHUB directory.
ENV ENV_NAME=DHUB
# Start http service using ENTRYPOINT
# Expose port 80.
EXPOSE 80
ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]
root@AjinkyaPame:~/Docker/AWSECR#
```

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

```

root@AjinkyaPame:~/Docker/AWSECR# docker build -t ubuntu_ecr .
[*] Building 21.9s (8/8) FINISHED
=> [internal] Load build definition from Dockerfile
=> => transferring dockerfile: 463B
=> [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
=> CACHED [1/3] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
=> [2/3] RUN apt update && apt install apache2 -y
=> [3/3] RUN echo "I am from AWS ECR." > index.html
=> exporting to image
=> exporting layers
=> => writing image sha256:35b6b06b2eb1led6981909fc6585303ea2836627b095b8191691968e7f4f208c
=> => naming to docker.io/library/ubuntu_ecr
root@AjinkyaPame:~/Docker/DHUB# cd ../DHUB/
root@AjinkyaPame:~/Docker/DHUB# docker build -t ubuntu_dhub .
[*] Building 1.1s (7/7) FINISHED
=> [internal] Load build definition from Dockerfile
=> => transferring dockerfile: 459B
=> [internal] Load metadata for docker.io/library/ubuntu:latest
=> [internal] Load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
=> CACHED [2/3] RUN apt update && apt install apache2 -y
=> [3/3] RUN echo "I am from Docker Hub." > index.html
=> exporting to image
=> exporting layers
=> => writing image sha256:1387f67fe09582aaa34dd64809d4e20eb97d9bc6009b9bf1e87012f7d3da
=> => naming to docker.io/library/ubuntu_dhub
root@AjinkyaPame:~/Docker/DHUB# docker image ls | grep ubuntu
ubuntu_dhub                         latest      1387f67fe095   12 seconds ago  241MB
ubuntu_ecr                          latest      35b6b06b2eb1   33 seconds ago  241MB
custom_ubuntu                        latest      +f144c7da594   35 minutes ago  289MB
root@AjinkyaPame:~/Docker/DHUB# 

root@AjinkyaPame:~/Docker/DHUB# cd ../../AWSECR/
root@AjinkyaPame:~/Docker/AWSECR# aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/b0r7d3x2
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@AjinkyaPame:~/Docker/AWSECR# docker build -t ajinkyap009/apache2_ecr .
[*] Building 1.9s (8/8) FINISHED
=> [internal] Load build definition from Dockerfile
=> => transferring dockerfile: 463B
=> [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
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
=> CACHED [2/3] RUN apt update && apt install apache2 -y
=> [3/3] RUN echo "I am from AWS ECR." > index.html
=> exporting to image
=> exporting layers
=> => writing image sha256:35b6b06b2eb1led6981909fc6585303ea2836627b095b8191691968e7f4f208c
=> => naming to docker.io/ajinkyap009/apache2_ecr
root@AjinkyaPame:~/Docker/AWSECR# docker image ls | grep apache
ajinkyap009/apache2_dhub              v1          1387f67fe095   17 minutes ago  241MB
ajinkyap009/apache2_ecr              latest      35b6b06b2eb1   18 minutes ago  241MB

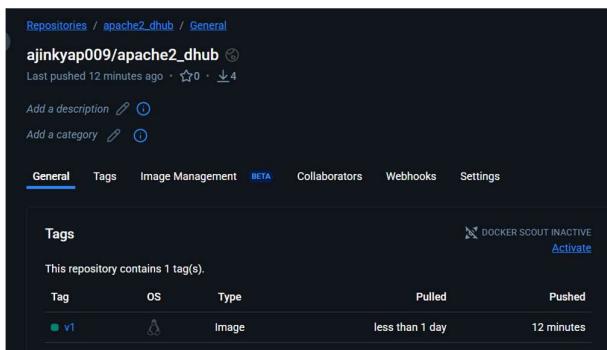
```

34. Push Images to Repositories: Push the Docker image to Docker Hub. Push the Docker image to AWS ECR.

```

root@AjinkyaPame:~/Docker/DHUB# docker push ajinkyap009/apache2_dhub:v1
The push refers to repository [docker.io/ajinkyap009/apache2_dhub]
9bfd664d8a92: Pushed
387175af7fe4: Pushed
073ec47a8c22: Mounted from library/ubuntu
v1: digest: sha256:c2ccc5be663c4bf01faf3ecbf05fb87e3803b478d1d6d402f9c760e7bbadc67 size: 948

```



```

root@Ajinkyapame:~/Docker/AWSECR# docker tag ajinkyap009/apache2_ecr:latest public.ecr.aws/b0r7d3x2/ajinkyap009/apache2_ecr:latest
root@Ajinkyapame:~/Docker/AWSECR# docker image ls | grep apache
ajinkyap009/apache2_dhub          v1           1387f67fe095  18 minutes ago   241MB
ajinkyap009/apache2_ecr          latest       35b6b06b2eb1  18 minutes ago   241MB
public.ecr.aws/b0r7d3x2/apache2_ecr latest       35b6b06b2eb1  18 minutes ago   241MB
root@Ajinkyapame:~/Docker/AWSECR# docker push public.ecr.aws/b0r7d3x2/ajinkyap009/apache2_ecr:latest
The push refers to repository [public.ecr.aws/b0r7d3x2/ajinkyap009/apache2_ecr]
f651a327132d: Pushed
387175af7fe4: Pushed
073ec47a8c22: Pushed
latest: digest: sha256:ff19f3816ffa0624b9061adb9e2a545f93d60b041cb9c9cf1ccbf15d7856feb2 size: 948
root@Ajinkyapame:~/Docker/AWSECR# |

```

The screenshot shows the AWS Lambda function configuration page. The left sidebar has sections for 'Amazon Elastic Container Registry' (Private registry and Public registry), 'AWS Lambda' (Functions, Triggers, Versions, Layers, Policies, and Metrics), and 'Logs'. The main content area is titled 'Function configuration' for the function 'aapame'. It shows the 'Code' tab selected, with the code editor containing the provided Python code. The 'Environment' tab is also visible.

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
```

```

root@Ajinkyapame:~/Docker/AWSECR# docker run -d -p 8081:80 --name I_AM_FROM_DHUB ajinkyap009/apache2_dhub:v1
d871139d4d5e8ad40ce9c991b27c976db85f6e04b3035244ce3bdf6a63ba525
root@Ajinkyapame:~/Docker/AWSECR# docker run -d -p 8082:80 --name I_AM_FROM_ECR ajinkyap009/apache2_ecr:latest
3048d3970b456e6b1acc5c5cf96f2f6ea89b51ed86ce25757495ac97724da30f
root@Ajinkyapame:~/Docker/AWSECR# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
3048d3970b45 ajinkyap009/apache2_ecr:latest "apache2ctl -D FOREG..." 2 seconds ago Up 2 seconds 0.0.0.0:8082->80/tcp, [::]:8082->80/tcp I_AM_FRO_M_ECR
d871139d4d5e ajinkyap009/apache2_dhub:v1 "apache2ctl -D FOREG..." 22 seconds ago Up 21 seconds 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp I_AM_FRO_M_DHUB
e60f34e69503 release-branch-img "/docker-entrypoint..." 3 hours ago Up 3 hours 0.0.0.0:8383->80/tcp, [::]:8383->80/tcp release-nginx
e6c67bbc0797 ajinkyap009/custom-nginx:01 "/docker-entrypoint..." 4 hours ago Up 4 hours 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp custom-nginx-cont
root@Ajinkyapame:~/Docker/AWSECR# |

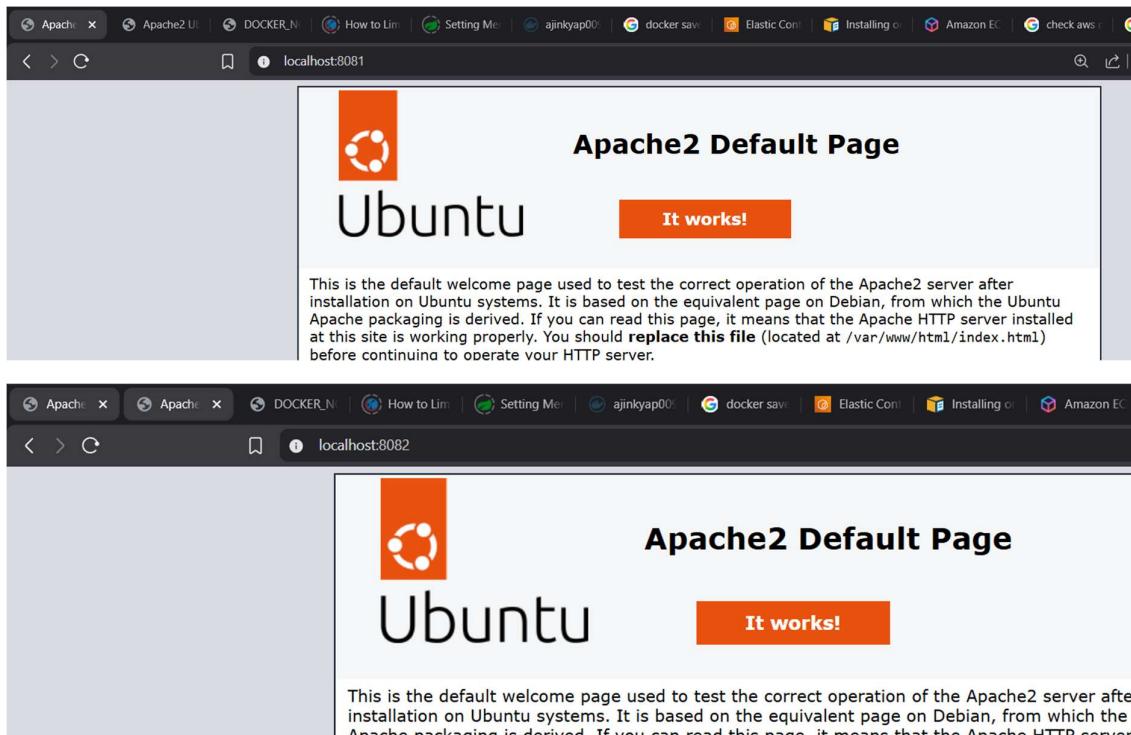
```

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>



Changing Dockerfile to use our index.html file:

```
root@AjinkyaPame:~/Docker# cat AWSECR/Dockerfile
# Base image: Ubuntu
FROM ubuntu:latest
# Install packages: httpd
RUN apt update && apt install apache2 -y
# Add "I am from AWS ECR" to the index.html file for AWS ECR directory.
RUN echo "I am from AWS ECR." > /var/www/html/index.html
# Set environment variable ENV_NAME=AWSECR for the AWS ECR directory.
ENV ENV_NAME=AWSECR
# Start http service using ENTRYPOINT
# Expose port 80.
EXPOSE 80
ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]
```

```

root@AjinkyaPame:~/Docker# cat DHUB/Dockerfile
# Base image: Ubuntu
FROM ubuntu:latest
# Install packages: httpd
RUN apt update && apt install apache2 -y
# Add "I am from Docker Hub" to the index.html file for DHUB directory.
RUN echo "I am from Docker Hub." > /var/www/html/index.html
# Set environment variable ENV_NAME=DHUB for the DHUB directory.
ENV ENV_NAME=DHUB
# Start http service using ENTRYPOINT
# Expose port 80.
EXPOSE 80
ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]
root@AjinkyaPame:~/Docker#
root@AjinkyaPame:~/Docker# cd DHUB/

```

```

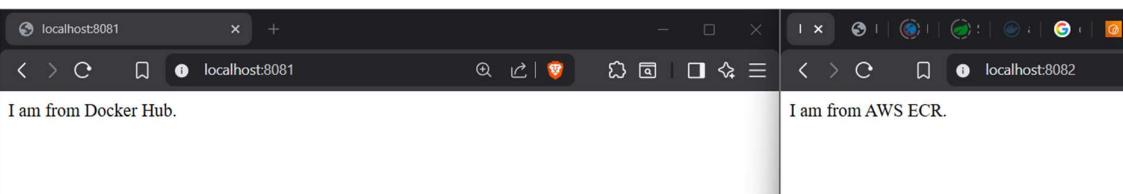
root@AjinkyaPame:~/Docker/DHUB# docker build -t apache2_dhub:v2 .
[+] Building 2.2s (8/8) FINISHED
--> [internal] load build definition from Dockerfile
--> [internal] load metadata for docker.io/library/ubuntu:latest
--> [auth] library/ubuntu:pull token for registry-1.docker.io
--> [internal] load .dockerignore
--> transferring context: 2B
--> [1/3] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
--> CACHED [2/3] RUN apt update && apt install apache2 -y
--> [3/3] RUN echo "I am from Docker Hub." > /var/www/html/index.html
--> exporting to image
--> exporting layers
--> writing image sha256:dc485665cee3fa0991345f758121c02d43ba0774a5dba4e974260ddfc9479ff2f
root@AjinkyaPame:~/Docker/DHUB# cd ..../AWSERCR
root@AjinkyaPame:~/Docker/AWSECR# docker build -t apache2_ecr:v2 .
[+] Building 0.6s (7/7) FINISHED
--> [internal] load build definition from Dockerfile
--> [internal] load metadata for docker.io/library/ubuntu:latest
--> [internal] load .dockerignore
--> transferring context: 2B
--> [1/3] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
--> CACHED [2/3] RUN apt update && apt install apache2 -y
--> [3/3] RUN echo "I am from AWS ECR." > /var/www/html/index.html
--> exporting to image
--> exporting layers
--> writing image sha256:1a8bb797c49e90da0bb332df44d9b2e13c1fa39131bb82a9b33f50410af74a49
--> naming to docker.io/library/apache2_ecr:v2
root@AjinkyaPame:~/Docker/AWSECR#

```

```

root@AjinkyaPame:~/Docker/AWSECR# docker run -d -p 8081:80 --name I_AM_FROM_DHUB apache2_dhub:v2
5bfd95ef67a:f57ef955cbcddff1d44ae194cca4db942fe30a37c91bab63ec6
root@AjinkyaPame:~/Docker/AWSECR# docker run -d -p 8082:80 --name I_AM_FROM_ECR apache2_ecr:v2
dce87d275387a7afa3691349b6b63fc73591ecfa2ced7de7bc55dd32b0cf609
root@AjinkyaPame:~/Docker/AWSECR# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
dce87d275387 apache2_ecr:v2 "apache2ctl -D FOREG..." 2 seconds ago Up 2 seconds 0.0.0.0:8082->80/tcp, [::]:8082->80/tcp I_AM_FROM_E
CR
5bfd95ef67af apache2_dhub:v2 "apache2ctl -D FOREG..." 48 seconds ago Up 47 seconds 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp I_AM_FROM_D
HUB
e60f34ee69503 release-branch-img "/docker-entrypoint..." 4 hours ago Up 4 hours 0.0.0.0:8383->80/tcp, [::]:8383->80/tcp release-nginx
nx
e6c67bbc0797 ajinkyap009/custom-nginx:01 "/docker-entrypoint..." 4 hours ago Up 4 hours 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp custom-nginx
x-cont
root@AjinkyaPame:~/Docker/AWSECR#

```



Images (2)						
	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
<input type="checkbox"/>	v2	Image	November 06, 2025, 16:53:46 (UTC+05.5)	104.90	<input type="button"/> Copy URI	<input type="button"/> sha256:d336ae61d86151...
<input type="checkbox"/>	latest	Image	November 06, 2025, 16:33:01 (UTC+05.5)	104.90	<input type="button"/> Copy URI	<input type="button"/> sha256:ff19f3816ffa0624...

ajinkyap009/apache2_dhub

Last pushed 1 minute ago · Repository size: 99.2 MB · 0 · 8

Add a description

Add a category

General Tags Image Management BETA Collaborators Webhooks Settings

Tags

DOCKER SCOUT INACTIVE [Activate](#)

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
v2		Image	less than 1 day	1 minute
v1		Image	less than 1 day	38 minutes

37. 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@AjinkyaPame:~/Docker/Task_37# cat Dockerfile
FROM ubuntu
RUN apt update && apt install vim curl git -y
ENV MY_NAME=Ajinkya
EXPOSE 80

root@AjinkyaPame:~/Docker/Task_37# docker build -t custom_ubuntu .
[+] Building 26.6s (6/6) FINISHED                               docker:default
=> [internal] load build definition from Dockerfile           0.0s
=> => transferring dockerfile: 126B                           0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.0s
=> [internal] load .dockerignore                            0.0s
=> => transferring context: 2B                             0.0s
=> CACHED [1/2] FROM docker.io/library/ubuntu:latest        0.0s
=> [2/2] RUN apt update && apt install vim curl git -y      25.7s
=> exporting to image                                         0.9s
=> => exporting layers                                       0.9s
=> => writing image sha256:509d4459352135f1279859d10a32a8256bfb5bff2 0.0s
=> => naming to docker.io/library/custom_ubuntu             0.0s
```

```
root@AjinkyaPame:~/Docker/Task_37# docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
custom_ubuntu       latest   509d44593521  44 seconds ago  289MB
```

38. Docker networking: Create two Docker containers with different names, and add them to the same custom network. Verify that the containers can communicate with each other using their names as hostnames.

```
root@AjinkyaPame:~/Docker# docker network create --driver=bridge custom_network
bc09284221a2252fe26948dd029c4337efb5248bdf82a452a4f13c868e699797
root@AjinkyaPame:~/Docker# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
100382874b42    bridge    bridge      local
bc09284221a2    custom_network    bridge      local
```

```
root@AjinkyaPame:~/Docker# docker container run -dit --name nginx_cont_1 --network custom_network nginx:latest
9954d32547bf8b3c26b40eb5e76020ebc11d4b555df3bec6ddccdf22bf2733
root@AjinkyaPame:~/Docker# docker container run -dit --name nginx_cont_2 --n
etwork custom_network nginx:latest
c853a66fc0fa61306ea2b5b7511bef6e006b827b26e1f3bdd26583c36601ae0f
```

```
root@AjinkyaPame:~/Docker# docker container exec -it nginx_cont_1 /bin/bash
root@9954d32547bf:/# hostname
9954d32547bf
root@9954d32547bf:/# hostname -I
172.19.0.2
root@9954d32547bf:/# curl 172.19.0.3
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

39. Data persistence with volumes: Run a Docker container with an attached volume, create a file inside the volume, and verify that the file persists after the container is stopped and removed.

```
root@AjinkyaPame:~/Docker# mkdir myvolume
root@AjinkyaPame:~/Docker# ls
AWSEC      TASK_25  ajinkya-cloudbethix-nginx  release-nginx.tar
DHUB      Task_30  index.html
Dockerfile  Task_37  myvolume
root@AjinkyaPame:~/Docker# pwd
/root/Docker
root@AjinkyaPame:~/Docker# docker container run -dit --name volume_pr --volume /root/Docker/myvolume:/tmp alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
2d35ebdb57d9: Already exists
Digest: sha256:4b7ce07002c69e8f3d704a9c5d6fd3053be500b7f1c69fc0d80990c2ad8dd412
Status: Downloaded newer image for alpine:latest
f34ec502c496728271336e1019bac1c28423546c6a81e8fc735576063c862ce
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f34ec502c496	alpine	"/bin/sh"	4 seconds ago	Up 3 seconds		volume_pr

```
root@AjinkyaPame:~/Docker# docker exec -it volume_pr /bin/sh
/ # cd /tmp/
/tmp # touch from_alpine.txt
/tmp # exit
root@AjinkyaPame:~/Docker# docker container stop volume_pr
volume_pr
root@AjinkyaPame:~/Docker# ls myvolume/
from_alpine.txt
root@AjinkyaPame:~/Docker# docker container rm volume_pr
volume_pr
root@AjinkyaPame:~/Docker# ls myvolume/
from_alpine.txt
root@AjinkyaPame:~/Docker# |
```

40. Docker Compose: Create a docker-compose.yml file to run a multi-container application consisting of a web server (e.g., Nginx) and a database server (e.g., MySQL). Ensure that both containers are connected to the same network.

```

root@AjinkyaPame:~/Docker/docker-compose# cat docker-compose.yaml
version: "1.0"
services:
  nginx:
    image: nginx
    container_name: nginx_cont
    ports:
      - 80:80
    networks:
      - custom_network_1

  mysql:
    image: mariadb
    container_name: mysql_cont
    environment:
      MYSQL_ROOT_PASSWORD: ap1234
      MYSQL_DATABASE: mydb
    volumes:
      - mysqlldb:/var/lib/mysql
    ports:
      - 3600:3600
    networks:
      - custom_network_1

volumes:
  mysqlldb:

networks:
  custom_network_1:
    driver: bridge

```

```

root@AjinkyaPame:~/Docker/docker-compose# docker compose up -d
[WARN] 00000] /root/Docker/docker-compose/docker-compose.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[*] Running 9/9
  ✓mysql Pulled
    ✓4b3ffd8ccb52 Already exists
    ✓c8ed8fd4db803 Pull complete
    ✓b459be131d7b Pull complete
    ✓52f17a977a59 Pull complete
    ✓f008921c8dd7 Pull complete
    ✓44d1345a90ac Pull complete
    ✓ac8622786843 Pull complete
    ✓e62996882970 Pull complete
[*] Running 4/4
  ✓Network docker-compose_custom_network_1 Created
  ✓Volume docker-compose_mysqlldb Created
  ✓Container mysql_cont Started
  ✓Container nginx_cont Started

```

```

root@AjinkyaPame:~/Docker/docker-compose# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
40f3c81718eb mariadb "docker-entrypoint.s..." 4 seconds ago Up 3 seconds 3306/tcp, 0.0.0.0:3600->3600/tcp, [::]:3600->3600/tcp mysql_c
ont
fadbd1b8cdac4 nginx "/docker-entrypoint..." 4 seconds ago Up 3 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp nginx_c
ont
root@AjinkyaPame:~/Docker/docker-compose# docker volume ls
DRIVER VOLUME NAME
local 9c3a91b5c3b37633159fd0bcd32315ba092a9b9eb71a24e521cbff5ea66cc6905
local d880b59f913b84c2dbc557cd3c1dc83276820e46061e451c42bf767ecc85d227
local docker-compose_mysqlldb
local docker-compose_myvolume
local myvolume
root@AjinkyaPame:~/Docker/docker-compose# docker network ls
NETWORK ID NAME DRIVER SCOPE
10039287fb42 bridge bridge local
bc89284221a2 custom_network bridge local
fc9e33eb429a docker-compose_custom_network_1 bridge local
11fc9d8a6be7 docker-compose_default bridge local

```

41. Container logging: Run a Docker container that generates log output, and practice using docker logs to view and analyze the logs.

```
root@AjinkyaPame:~/Docker# docker container run -d --name nginx-logs-cont -p80:80 nginx
42282e22a18918d2e0bef6ea794db48cf970b17b000ff7bd6c1307ef03631803

root@AjinkyaPame:~/Docker# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
42282e22a189 nginx "/docker-entrypoint..." 31 seconds ago Up 31 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp nginx-logs-cont
root@AjinkyaPame:~/Docker# 

root@AjinkyaPame:~/Docker# docker logs nginx-logs-cont
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/11/11 10:53:30 [notice] 1#1: using the "epoll" event method
2025/11/11 10:53:30 [notice] 1#1: nginx/1.29.3
2025/11/11 10:53:30 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/11/11 10:53:30 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2025/11/11 10:53:30 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/11/11 10:53:30 [notice] 1#1: start worker processes
2025/11/11 10:53:30 [notice] 1#1: start worker process 29
2025/11/11 10:53:30 [notice] 1#1: start worker process 30
172.17.0.1 - - [11/Nov/2025:10:53:35 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
2025/11/11 10:53:30 [error] 29#29: *1 [open() "/usr/share/nginx/html/favicon.ico"] failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referer: "http://localhost/"
172.17.0.1 - - [11/Nov/2025:10:53:36 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
172.17.0.1 - - [11/Nov/2025:10:53:36 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
172.17.0.1 - - [11/Nov/2025:10:53:37 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
172.17.0.1 - - [11/Nov/2025:10:53:37 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
172.17.0.1 - - [11/Nov/2025:10:53:38 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
172.17.0.1 - - [11/Nov/2025:10:53:38 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
```

42. Container resource monitoring: Run a Docker container that consumes system resources (e.g., CPU or memory) and practice using docker stats and docker top to monitor the container's resource usage.

```
root@AjinkyaPame:~/Docker# docker container run -d -m 512m --cpu="1.5" --name resource-cont -p80:80 nginx
a7d4a9c88648b0c11d0169b70909cd4980a517e5cb96b586a1b7003189e752be0
root@AjinkyaPame:~/Docker# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a7d4a9c88648 nginx "/docker-entrypoint..." 2 seconds ago Up 2 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp resource-cont
root@AjinkyaPame:~/Docker# 

CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
2505c6f8e2ac resource-cont 0.00% 6.5MiB / 512MiB 1.27% 946B / 126B 13.3MB / 12.3kB 3

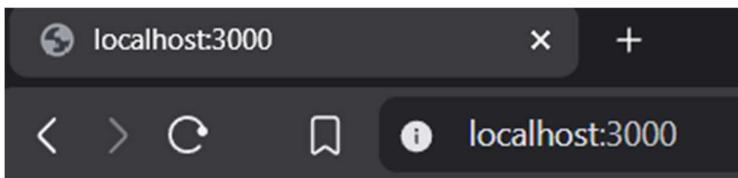
root@AjinkyaPame:~/Docker# docker container top resource-cont
UID PID PPID C STIME TTY TIME CMD
root 16494 16470 0 10:58 ?
x: master process nginx -g daemon off;
message+ 16556 16494 0 10:58 ?
x: worker process message+ 16557 16494 0 10:58 ?
x: worker process root@AjinkyaPame:~/Docker#
```

43. Updating a containerized application: Update the source code of a simple containerized application, rebuild the Docker image, and deploy the updated version while minimizing downtime.

```

root@AjinkyaPame:~/Docker# cd task_43
root@AjinkyaPame:~/Docker/task_43# nano app.js
root@AjinkyaPame:~/Docker/task_43# nano package.json
root@AjinkyaPame:~/Docker/task_43# nano Dockerfile
root@AjinkyaPame:~/Docker/task_43# docker build -t node:v1 .
[+] Building 13.9s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 346B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8c
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8c
=> => sha256:dd71dde3bb5c203d162902ze6b8994cb2309ae09a0eabc4efea161b2b5a3d0e 48.01MB / 48.01MB
=> => sha256:929b804d7c782f04f615cf785488fed452b569f87c73ff666ad53a7554f0006 1.72kB / 1.72kB
=> => sha256:ee77c6cd7c1883ecc882ad5ccdef3a8ec1a27dfb96162bf03d3710839b3da 6.18kB / 6.18kB
=> => extracting sha256:dd71dde3bb5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e
=> => extracting sha256:1e5a4c89cee5c0826c540ab6d4b6b491c96edaa01837f430bd07f0d26702d6e3
=> => extracting sha256:25ff2da36419088f65c3a74d80409d6b1b622cfcaab220b9ea70b80df5a2e0549
=> [internal] load build context
=> => transferring context: 966B
=> [auth] library/node:pull token for registry-1.docker.io
=> [2/5] WORKDIR /usr/src/app
=> [3/5] COPY package*.json .
=> [4/5] RUN npm install --production
=> [5/5] COPY .
=> exporting to image
=> => exporting layers
=> => writing image sha256:d8c69f2e02a6f41ad5b67f0984cf6dc9295cc9531c89d8e8f5b41d13e9cee3fa
=> => naming to docker.io/library/node:v1
root@AjinkyaPame:~/Docker/task_43# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@AjinkyaPame:~/Docker/task_43# docker container run -d -p 3000:3000 --name cont_v1 node:v1
867dab6e9afa:2660a50b1922323a04f1e19405150b7b8abf7b2baba32
root@AjinkyaPame:~/Docker/task_43# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
867dab6e9afa node:v1 "docker-entrypoint.s..." 3 seconds ago Up 2 seconds 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp cont_v1

```



Hello, world — version 1.0!

```

root@AjinkyaPame:~/Docker/task_43# nano app.js
root@AjinkyaPame:~/Docker/task_43# docker build -t node:v2 .
[+] Building 0.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 346B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8c
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8c
=> => sha256:2e0826c540ab6d4b6b491c96edaa01837f430bd07f0d26702d6e3 0.05
=> => sha256:cached [2/5] WORKDIR /usr/src/app
=> => sha256:cached [3/5] COPY package*.json .
=> => sha256:cached [4/5] RUN npm install --production
=> => [5/5] COPY .
=> exporting to image
=> => exporting layers
=> => writing image sha256:dbde202e8f22e37c75f2ea5b9bb99303988c34ea3e7b26f490509d80210899b9
=> => naming to docker.io/library/node:v2
root@AjinkyaPame:~/Docker/task_43# docker container run -d -p 3001:3000 --name cont_v2 node:v2
148997201a39ede4508fb46ccde8a01ece0e3c562b355bb3e209ed419f56473d
root@AjinkyaPame:~/Docker/task_43# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
148997201a39 node:v2 "docker-entrypoint.s..." 2 seconds ago Up 2 seconds 0.0.0.0:3001->3000/tcp, [::]:3001->3000/tcp cont_v2
867dab6e9afa node:v1 "docker-entrypoint.s..." 2 minutes ago Up 2 minutes 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp cont_v1
root@AjinkyaPame:~/Docker/task_43# docker container rm $(docker stop $(docker ps -q))
148997201a39
867dab6e9afa
root@AjinkyaPame:~/Docker/task_43# docker container run -d -p 3000:3000 --name cont_v2 node:v2
7f92f653aaed9565584bd99c1ba8355b767ea849b4a0317a84a5134262103d1
root@AjinkyaPame:~/Docker/task_43#

```



Hello, world — version 2.0!



Hello, world — version 2.0!

44. Bridge network: Create two Docker containers using different images, such as NGINX and MySQL. Connect them using a user-defined bridge network and ensure they can communicate with each other.

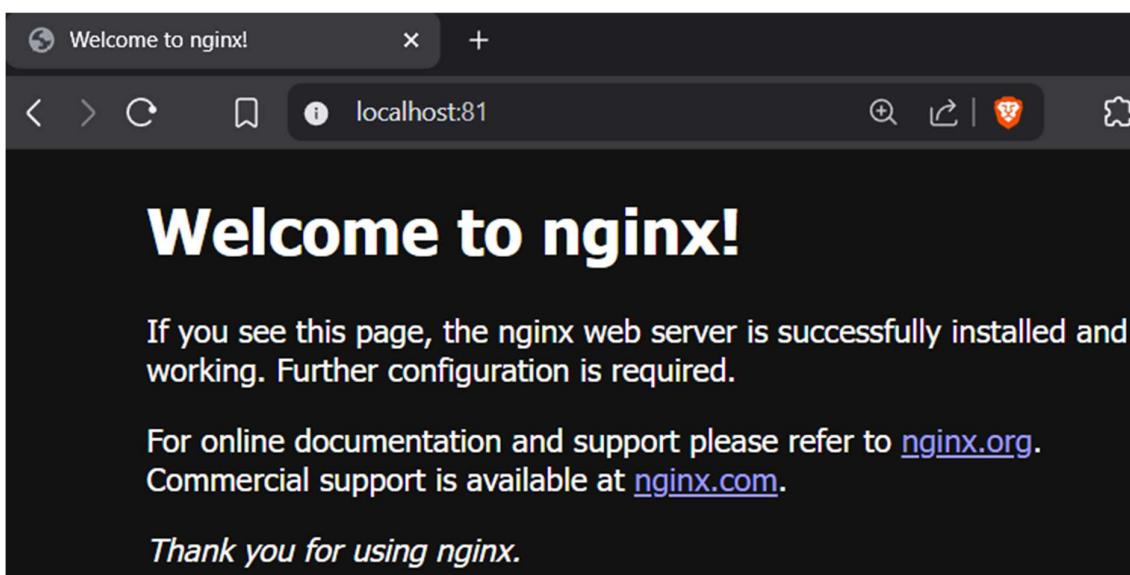
```
root@AjinkyaPame:~/Docker# docker network connect custom_network resource=cont
root@AjinkyaPame:~/Docker# docker container run -d --name custom_mysql --env MYSQL_ROOT_PASSWORD="Admin123" --env MYSQL_DATABASE="mydb" --networ
k custom_network -p 3306:3306 -v myvolume:/var/lib/mysql/ mariadb:latest
```

telnet to port 3306 of mysql container from nginx container:

```
root@a7d4a9c88648:/# telnet 172.19.0.3 3306
Trying 172.19.0.3...
Connected to 172.19.0.3.
Escape character is '^]'.
Z
12.0.2-MariaDB-ubuntu2404"ijZa[&H=>xFlc!wV*k_ybmysql_native_passwordConnection closed by foreign host.
root@a7d4a9c88648:/# |
```

45. Port mapping: Run a containerized web server (e.g., NGINX or Apache) and map its default port (80 or 443) to a custom port on the host machine. Verify that the web server is accessible through the custom port on the host.

```
root@AjinkyaPame:~/Docker# docker container run -d --name nginx-custom-port -p 81:80 nginx:latest
2151e9026ee04fb71267b59989ea676a75dd86e5ae150a60c91001bae5387a02
root@AjinkyaPame:~/Docker# docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS          NAMES
2151e9026ee0   nginx:latest  "/docker-entrypoint..."  3 seconds ago  Up 3 seconds  0.0.0.0:81->80/tcp, [::]:81->80/tcp   nginx-custom-port
root@AjinkyaPame:~/Docker# |
```



46. Host networking: Run a container with the host networking mode and compare its network configuration with that of the host machine. Explain the advantages and disadvantages of using host networking for containers.

```
root@AjinkyaPame:~/Docker# docker network inspect host
[
    {
        "Name": "host",
        "Id": "fb507d590c27340c0cef5768a29aaa09d306ef4ea1d016e1444886fc4a1e8329",
        "Created": "2025-06-03T07:28:47.60182622Z",
        "Scope": "local",
        "Driver": "host",
        "EnableIPv4": true,
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": null
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
root@AjinkyaPame:~/Docker# docker container run -d --name host-net-cont --network host nginx:latest
d7c33c9edc16c316ceeb23d2d0adba98ddd8abddf92f403d26bcc1f1374a4224
```

```
"Networks": {
    "host": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "MacAddress": "",
        "DriverOpts": null,
        "GwPriority": 0,
        "NetworkID": "fb507d590c27340c0cef5768a29aaa09d306ef4ea1d016e1444886fc4a1e8329",
        "EndpointID": "f63e20ff3299a923f2042bfe2a4110952307414c9d95fc7e929501da732c4282",
        "Gateway": "",
        "IPAddress": ""
    }
}
```

47. Named volume: Launch a containerized database (e.g., MySQL or PostgreSQL) using a named volume to store its data. Stop and remove the container, then recreate it using the same named volume. Verify that the data persists across container recreations.

```

root@AjinkyaPame:~/Docker# docker container run -d --name custom_mysql --env MYSQL_ROOT_PASSWORD="Admin123" --env MYSQL_DATABASE="mydb" --network myvolume -p 3306:3306 -v myvolume:/var/lib/mysql/ mariadb:latest
417eee1058e5b713e80d765ecc9ed9baea5c6d531f08443c869cea33a4adef0
root@AjinkyaPame:~/Docker# docker exec -it custom_mysql /bin/bash
root@417eee1058e5:~# mysql -u root -p Admin123
bash: mysql: command not found
root@417eee1058e5:~# mariadb -u root -p Admin123
Enter password:
ERROR 1049 (42000): Unknown database 'Admin123'
root@417eee1058e5:~# mariadb -u root -pAdmin123
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 12.0.2-MariaDB-ubuntu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mydb
Database changed
MariaDB [mydb]> create table anand(id int, name varchar(20));
Query OK, 0 rows affected (0.016 sec)

MariaDB [mydb]> insert into anand values (1, "Anand");
Query OK, 1 row affected (0.007 sec)

MariaDB [mydb]> insert into anand values (2, "Vishal");
Query OK, 1 row affected (0.007 sec)

MariaDB [mydb]> exit
Bye

root@417eee1058e5:~# ls /var/lib/mysql/
aria_log_00000001 ddl_recovery-backup.log ib_buffer_pool ibdata1 mariadb_upgrade_info mydb performance_schema tc.log undo002
aria_log_control ddl_recovery.log ib_logfile0 ibtmp1 multi-master.info mysql sys undo001 undo003
root@417eee1058e5:~# ls /var/lib/mysql/mydb/
mydb/ mysql/
root@417eee1058e5:~# ls /var/lib/mysql/mydb/
anand.frm anand.ibd db.opt
root@417eee1058e5:~| 

root@AjinkyaPame:~/Docker# docker volume ls
DRIVER      VOLUME NAME
local        9c3a91b5c3b37633159fd0bc3d23150a092a9b9eb71a24e521cb8f5ea66c6905
local        d880b59f913b84c2dbc557cd3c1dc83276820e46061e451c42bf767ecc85d227
local        docker-compose_myvolume
local        myvolume
root@AjinkyaPame:~/Docker# cd /var/lib/docker
root@AjinkyaPame:/var/lib/docker# ls
buildkit  containers  engine-id  image  network  nuke-graph-directory.sh  overlay2  plugins  runtimes  swarm  tmp  volumes
root@AjinkyaPame:/var/lib/docker/volumes# ls
9c3a91b5c3b37633159fd0bc3d23150a092a9b9eb71a24e521cb8f5ea66c6905  d880b59f913b84c2dbc557cd3c1dc83276820e46061e451c42bf767ecc85d227  metadata.db
backingFsBlockDev                                         docker-compose_myvolume
root@AjinkyaPame:/var/lib/docker/volumes# ls myvolume/
._data
root@AjinkyaPame:/var/lib/docker/volumes# ls myvolume/_data/
aria_log_00000001 ddl_recovery-backup.log ib_logfile0 mariadb_upgrade_info mydb performance_schema undo001 undo003
aria_log_control ib_buffer_pool ibdata1 multi-master.info mysql sys undo002
root@AjinkyaPame:/var/lib/docker/volumes# ls myvolume/_data/mydb/
anand.frm anand.ibd db.opt
root@AjinkyaPame:/var/lib/docker/volumes# | 

root@AjinkyaPame:~/Docker# docker container run -d --name custom_mysql --env MYSQL_ROOT_PASSWORD="Admin123" --network myvolume -p 3306:3306 -v myvolume:/var/lib/mysql/ mariadb:latest
36661f9de35f42465c628e8dc7800f643387802b865dc258b225b46ca2dbe1
root@AjinkyaPame:~/Docker# docker exec -it custom_mysql /bin/bash
root@36661f9de35f:~# mysql -u root -pAdmin123
bash: mysql: command not found
root@36661f9de35f:~# mariadb -u root -pAdmin123
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 12.0.2-MariaDB-ubuntu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mydb |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.008 sec)

```

```
MariaDB [(none)]> use mydb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mydb]> show tables;
+-----+
| Tables_in_mydb |
+-----+
| anand          |
+-----+
1 row in set (0.000 sec)

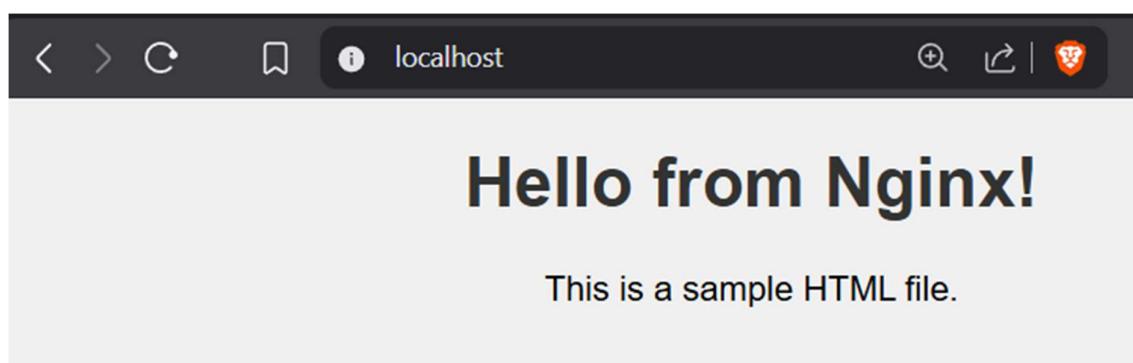
MariaDB [mydb]> select * from anand;
+---+---+
| id | name |
+---+---+
| 1  | Anand |
| 2  | Vishal |
+---+---+
2 rows in set (0.002 sec)

MariaDB [mydb]> |
```

- 48. Bind mount:** Create a simple web application with a local directory containing HTML, CSS, and JavaScript files. Run a web server container (e.g., NGINX or Apache) and bind mount the local directory to the container's webroot. Verify that the web application is accessible and update the local files to see if changes are reflected in the container.

```
root@AjinkyaPame:~/Docker# mkdir task_47
root@AjinkyaPame:~/Docker# cd task_47/
root@AjinkyaPame:~/Docker/task_47# nano index.html
root@AjinkyaPame:~/Docker/task_47# nano styles.css
root@AjinkyaPame:~/Docker/task_47# nano script.js
root@AjinkyaPame:~/Docker/task_47# nano Dockerfile
root@AjinkyaPame:~/Docker/task_47# cat Dockerfile
FROM nginx
COPY index.html /usr/share/nginx/html/
COPY styles.css /usr/share/nginx/html/
COPY script.js /usr/share/nginx/html/
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
root@AjinkyaPame:~/Docker/task_47# docker build -t my-nginx-img .
[+] Building 0.3s (9/9) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                      0.0s
=> => transferring dockerfile: 211B                                         0.0s
=> [internal] load metadata for docker.io/library/nginx:latest          0.0s
=> [internal] load .dockerignore                                         0.0s
=> => transferring context: 2B                                           0.0s
=> CACHED [1/4] FROM docker.io/library/nginx:latest                      0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 550B                                         0.0s
=> [2/4] COPY index.html /usr/share/nginx/html/                           0.0s
=> [3/4] COPY styles.css /usr/share/nginx/html/                          0.0s
=> [4/4] COPY script.js /usr/share/nginx/html/                          0.1s
=> exporting to image                                                 0.1s
=> => exporting layers                                              0.1s
=> => writing image sha256:15854800c77b0581924dfb8a3aec88540c2  0.0s
=> => naming to docker.io/library/my-nginx-img                         0.0s
```

```
root@AjinkyaPame:~/Docker/task_47# docker container run -d --name my-nginx-cont -p 80:80 my-nginx-img:latest
da9b4e9948c06ad06171fc6345d3699399fbb2536bcd70513be86f5124672570
root@AjinkyaPame:~/Docker/task_47# |
```



49. Volume backups: Using a container with a named volume, create a backup of the volume's data by either exporting it as a tarball or copying the data to a host directory. Restore the data to a new container and verify that the restoration was successful.

```

root@AjinkyaPame:~/Docker/task_43# docker volume create mydata
mydata
root@AjinkyaPame:~/Docker/task_43# docker run -it --name cont-1 -v mydata:/data alpine sh
/ # echo "Data from container 1" > /data/file1.txt
/ # exit
root@AjinkyaPame:~/Docker/task_43# mkdir backup
root@AjinkyaPame:~/Docker/task_43# docker run --rm \
--mount source=mydata,target=/data \
-v $(pwd):/backup \
alpine tar czf /backup/mydata_backup.tar.gz -C /data .
root@AjinkyaPame:~/Docker/task_43# ls backup/
root@AjinkyaPame:~/Docker/task_43# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
cfd086b176c alpine "sh" 3 minutes ago Exited (0) 45 seconds ago cont-1

root@AjinkyaPame:~/Docker/task_43# docker run -it --mount source=mydata_restored,target=/data -v $(pwd):/backup alpine sh
/ # ls /data
/ # tar xzf /backup/mydata_backup.tar.gz -C /data
/ # ls /data/
file1.txt
/ # exit
root@AjinkyaPame:~/Docker/task_43# docker volume ls
DRIVER VOLUME NAME
local 9c3a91b5c3b37633159fd0bc3d23150a092a9b9eb71a24e521cb8f5ea66c6905
local d880b59f913b84c2dbc557cd3c1dc83276820e46061e451c42bf767ecc85d227
local docker-compose_myvolume
local mydata
local mydata_restored
local myvolume
root@AjinkyaPame:~/Docker/task_43# ls /var/lib/docker/volumes/mydata_restored/_data/
file1.txt
root@AjinkyaPame:~/Docker/task_43#

```

50. Docker Compose networking: Create a Docker Compose file that defines a multi-container application with a frontend, backend, and database. Set up custom networks and volumes for the services and ensure that they can communicate with each other and store data persistently

```

root@AjinkyaPame:~/Docker/task_50# nano three-tier.yml
root@AjinkyaPame:~/Docker/task_50#
root@AjinkyaPame:~/Docker/task_50# docker compose -f three-tier.yml up -d
[WARN] [0000] /root/Docker/task_50/three-tier.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[*] Building 28.2s (11/11) FINISHED
=> [internal] load local build definitions
=> [internal] reading from system
=> [internal] load build definition from Dockerfile
=> [internal] transferring dockerfile: 332B
=> [internal] load metadata for docker.io/library/php:8.3-fpm
=> [internal] load .dockerrcignore
=> [internal] transferring context: 2B
=> [1/1] FROM docker.io/library/php:8.3-fpm
=> [internal] load build context
=> [internal] transferring context: 1.72kB
=> [2/4] RUN docker-php-ext-install mysqli pdo pdo_mysql
=> [3/4] WORKDIR /var/www/html
=> [4/4] COPY .
=> [internal] exporting to image
=> [internal] exporting layers
=> [internal] writing image sha256:ff064b5e8638e74a5b263fc3ffba2edad477cd865530788aba752ff5eff487ac
=> [internal] naming to docker.io/library/task_50-myapp
=> [internal] resolving provenance for metadata file
[*] Running 4/4
  ✓ task_50-myapp           Built
  ✓ Container task_50-mydb-1 Running
  ✓ Container task_50-myapp-1 Started
  ✓ Container task_50-myweb-1 Running
root@AjinkyaPame:~/Docker/task_50# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
53279c319ad4 task_50-myapp "docker-php-entrypoi..." 32 seconds ago Up 31 seconds 9000/tcp task_50-my
app-1
b7b728fa2854 nginx:latest "/docker-entrypoint..." 7 minutes ago Up 5 minutes 0.0.0.0:80->80/tcp, [::]:80->80/tcp task_50-my
web-1
b6ea9371683b mysql:8 "docker-entrypoint.s..." 7 minutes ago Up 5 minutes 0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp, 33060/tcp task_50-my
db-1

```

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpressdb |
+-----+
5 rows in set (0.01 sec)

mysql> use wordpressdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_wordpressdb |
+-----+
| messages |
+-----+
1 row in set (0.00 sec)

mysql> select * from messages;
+----+-----+-----+
| id | message | created_at |
+----+-----+-----+
| 1 | Hello from PHP & Docker! 2025-11-12 05:21:54 | 2025-11-12 05:21:54 |
| 2 | Hello from PHP & Docker! 2025-11-12 05:22:11 | 2025-11-12 05:22:11 |
| 3 | Hello from PHP & Docker! 2025-11-12 05:22:13 | 2025-11-12 05:22:13 |
+----+-----+-----+
3 rows in set (0.00 sec)

```



New record inserted successfully!

Saved Messages:

- 1: Hello from PHP & Docker! 2025-11-12 05:21:54 (2025-11-12 05:21:54)
- 2: Hello from PHP & Docker! 2025-11-12 05:22:11 (2025-11-12 05:22:11)
- 3: Hello from PHP & Docker! 2025-11-12 05:22:13 (2025-11-12 05:22:13)