

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
```

COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6	COLUMN 7
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f34ec502c496	alpine	"/bin/sh"	4 seconds ago	Up 3 seconds		volume_pr

```
root@AjinkyaPame:~/Docker# docker ps
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
```

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# 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

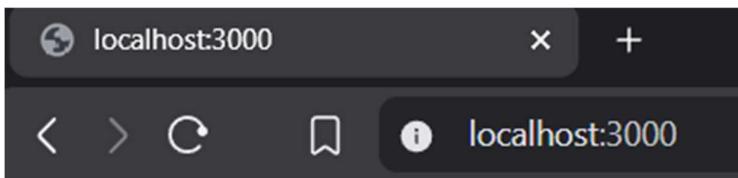
UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	16494	16470	0	10:58	?	00:00:00	nginx
x: master process	nginx	-g daemon off;					
message+	16556	16494	0	10:58	?	00:00:00	nginx
x: worker process							
message+	16557	16494	0	10:58	?	00:00:00	nginx
x: worker process							
root	16494	16470	0	10:58	?	00:00:00	nginx

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:ee77c6cd7c1883ecc882ad5cdef3a8ec1a27dfb96162bf03d3710839b3da 6.18kB / 6.18kB
--> => extracting sha256:dd71dde3bb5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e
--> => extracting sha256:1e5a4c89cee5c0826c540ab6d4b6b491c96edaa01837f430bd0f7fd26702d6e3
--> => 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
--> => 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
--> => sha256:2e8d6bde202e8f22e37c75f2ea5b9bb99303988c34ea3e7b26f490509d80210899b9 0.05
--> => 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
7f92f653aaed9565584bd9c1ba8355b767ea849b4a0317a84a5134262103d1
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.

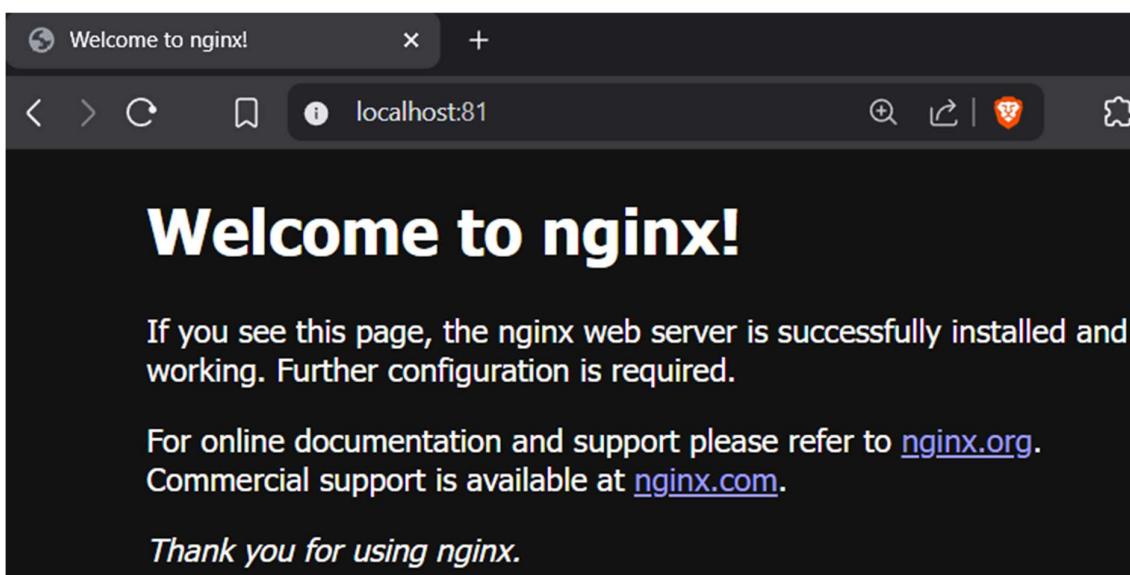
```
root@AjinkyaPame:~/Docker# docker network connect custom_network resource=cont
root@AjinkyaPame:~/Docker# docker container run -d --name custom_mysql --env MYSQL_ROOT_PASSWORD="Admin123" --env MYSQL_DATABASE="mydb" --networ
k custom_network -p 3306:3306 -v myvolume:/var/lib/mysql/ mariadb:latest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

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.

```

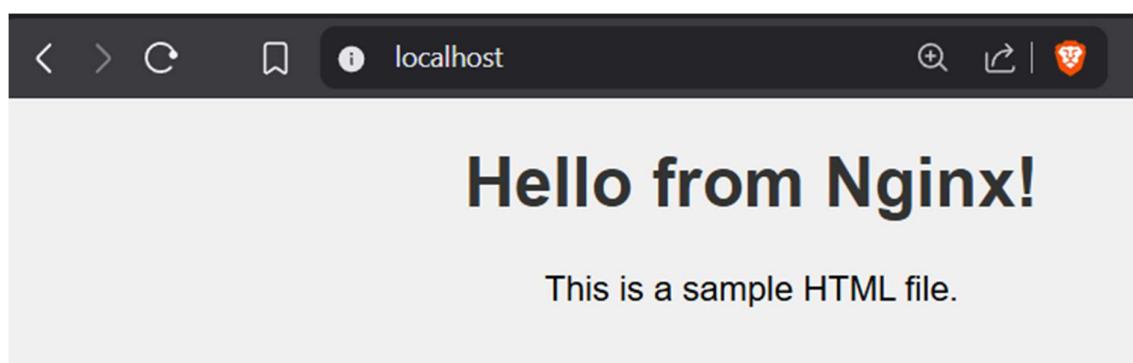
root@AjinkyaPame:~/Docker# mkdir task_47
root@AjinkyaPame:~/Docker# cd task_47/
root@AjinkyaPame:~/Docker/task_47# nano index.html
root@AjinkyaPame:~/Docker/task_47# nano styles.css
root@AjinkyaPame:~/Docker/task_47# nano script.js
root@AjinkyaPame:~/Docker/task_47# nano Dockerfile
root@AjinkyaPame:~/Docker/task_47# 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] loading 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 .
=> exporting to image
=> exporting layers
=> writing image sha256:ff064b5e8638e74a5b263fc3ffba2edad477cd865530788aba752ff5eff487ac
=> naming to docker.io/library/task_50-myapp
=> 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#
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
b8ea9371683b 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)