# Docker Task2

**Date: 04/07/25**

1. **Write a brief explanation of what Docker volumes are and why they are used in containerized environments. State different types of volumes in Docker and also make a note on difference between them.**

### Docker Volumes:

Docker volumes are used to store data outside of a container’s file system. This means the data stays safe even if the container is deleted or restarted. Volumes help keep your app’s important data (like databases, logs, or configs) separate from the app itself.

Use Docker Volumes:

* 1. To avoid data loss in case of container stoppage or removal.
  2. To easily share data among containers.
  3. TO get better performance compared to storing data in container.

### Types of Docker Volumes:

* Named volumes.
* Anonymous volumes.
* Bind mounts.

| **Named/Anonymous Volumes** | **Bind Mounts** |
| --- | --- |
| Managed by Docker | Managed by you (host system path) |
| Path auto-managed | You set the exact file path |
| Better for long-term app data | Better for development/debugging |
| Stored inside Docker's internal storage | Uses files/folders from host OS |
| Portable and safe | Host-specific (less portable) |

1. **Demonstrate the use of Named Volume.**

* Create a Docker Named volume named mydata.

*apame09@AjinkyaPame:~$ docker volume create mydata*

*mydata*

*apame09@AjinkyaPame:~$ docker volume ls*

*DRIVER VOLUME NAME*

*local mydata*

* Attach volume to a Nginx Container

*apame09@AjinkyaPame:~$ docker run -d -v mydata:/usr/share/nginx/html --name nginxcont nginx*

*bb99a97b17acefb8198fad54689979181ec754bac056fee22b30990ffe410d87*

*apame09@AjinkyaPame:~$ docker ps*

*CONTAINER ID IMAGE COMMAND CREATED*

bb99a97b17ac nginx "/docker-entrypoint.…" 9 seconds ago

*STATUS PORTS NAMES*

*Up 8 seconds 80/tcp nginxcont*

* Create an HTML file named index.html with some content (e.g., "Hello, Docker Volumes!") on your host machine. Copy this file into the mydata.



* Verify that the index.html file is accessible from within the container by starting a simple HTTP request.

*root@bb99a97b17ac:~# curl localhost*

*<h1>Hello from Docker Volumes!!!</h1>*

*root@bb99a97b17ac:~#*

1. **Write a brief explanation of what Docker networks.**

**Write the difference between host network and bridge network.**

### Docker Networks:

Docker networks allow containers to communicate with each other and with the outside world. They manage how data is shared between containers and how containers are exposed to other systems.

| **Bridge Network** | **Host Network** |
| --- | --- |
| Containers get their own IP | Container shares host’s IP and ports |
| Needs port mapping (-p 8080:80) | No port mapping needed |
| Good for isolated container setups | Best for performance-critical apps |
| Default for most containers | Used in special cases (like full access to host) |

1. **Demonstrate the use of Custom Network**

Create a custom bridge network named my\_network.

*apame09@AjinkyaPame:~$ docker network create my\_network*

*7752cbeba6bc07ed51b7b90e8b1f627a93a2a4a00666d3c2cc5*

*apame09@AjinkyaPame:~$ docker network ls*

*NETWORK ID NAME DRIVER SCOPE*

*7752cbeba6bc my\_network bridge local*

Start two containers, one using the nginx image and another using the httpd image.

*apame09@AjinkyaPame:~$docker run -d --name httpdcont httpd*

*8f2831a8685c04c6750c075149fd8c947569ea3d2f419cce4520*

*apame09@AjinkyaPame:~$ docker ps*

*CONTAINER ID IMAGE COMMAND CREATED*

8f2831a8685c httpd "httpd-foreground" 11 seconds ago

bb99a97b17ac nginx "/docker-entrypoint.…" 32 minutes ago

*STATUS PORTS NAMES*

*Up 10 seconds 80/tcp httpdcont*

*Up 14 minutes 80/tcp nginxcont*

Attach both containers to the my\_network network.

*apame09@AjinkyaPame:~$ docker network connect my\_network httpdcont*

*apame09@AjinkyaPame:~$ docker network connect my\_network nginxcont*

Test Network Connectivity: Ensure that the nginx container can communicate with the httpd container over the custom network. You can do this by sending an HTTP request from one container to another using tools like curl.

*apame09@AjinkyaPame:~$ docker exec -it nginxcont bash*

*root@bb99a97b17ac:/# curl* <http://172.20.0.2> *(IP of httpdcont)*

*<html><body><h1>It works!</h1></body></html>*

*root@bb99a97b17ac:/#*

1. **Write a note on Dockerfile with usage of its attributes.**

A Dockerfile is a script used to create Docker images automatically. It contains a series of instructions that Docker reads and executes to build a customized container image. This eliminates the need to manually configure environments, making application deployment more consistent, portable, and efficient.

### Common Dockerfile Attributes:

* **FROM** – Sets the base image

**Ex**. FROM ubuntu:20.04

* **RUN** – Runs commands inside the image

**Ex**. RUN apt-get update

* **COPY** – Copies files from your system to the image

**Ex**. COPY . /app

* **WORKDIR** – Sets the working directory inside the container

**Ex**. WORKDIR /app

* **CMD** – Provides the default command when the container starts

**Ex**. CMD ["node", "app.js"]

* **EXPOSE** – Informs Docker which port the app uses

**Ex.** EXPOSE 3000

* **ENV** – Sets environment variables

**Ex**. ENV NODE\_ENV=production

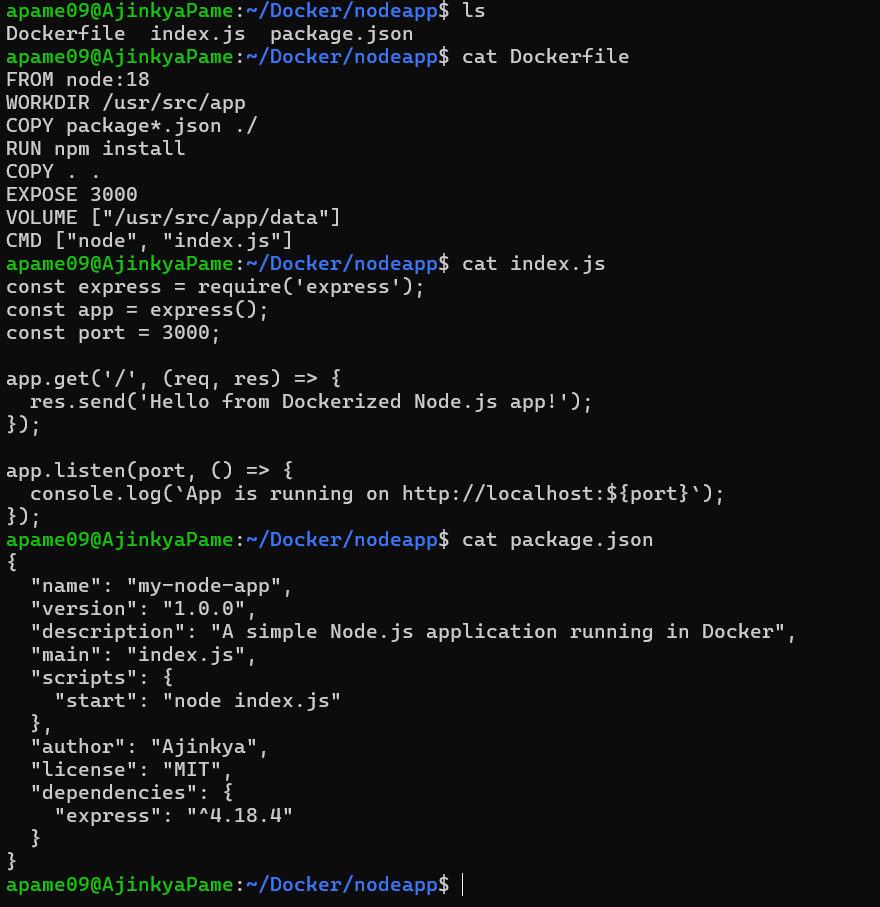
1. **What is difference between CMD and ENTRYPOINT?**

| **CMD** | **ENTRYPOINT** |
| --- | --- |
| CMD defines the default command or arguments to run when the container starts. | ENTRYPOINT sets the main command that will always run when the container starts. |
| The command defined by CMD can be easily overridden by providing a new command when running the container. | The command in ENTRYPOINT cannot be overridden easily; only its arguments can be changed at runtime. |
| If both CMD and ENTRYPOINT are specified, CMD provides default arguments to ENTRYPOINT. | When used with CMD, ENTRYPOINT takes precedence and treats CMD as its default arguments. |
| CMD is often used when you want to give users the flexibility to change the command when running the container. | ENTRYPOINT is used when you want your container to always run a specific executable, regardless of user input. |
| CMD is easier and more straightforward for simple use cases. | ENTRYPOINT is more suitable when the container is built to run a single, fixed application. |

1. **What is difference between ADD and COPY?**

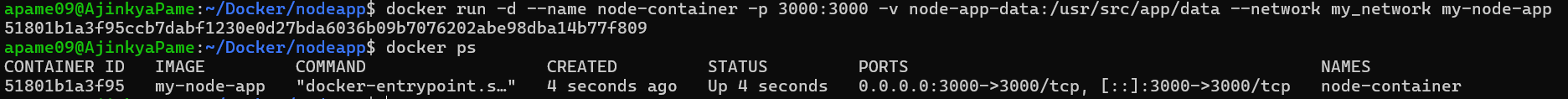
| **COPY** | **ADD** |
| --- | --- |
| COPY is used to copy files and directories from the host system into the Docker image. | ADD also copies files and directories from the host into the Docker image, but with additional features. |
| It performs a straightforward file copy and does not handle any other functionality. | It can handle more than just copying—like automatically extracting local .tar archives. |
| COPY is considered more transparent and is recommended for most use cases. | ADD may lead to unexpected behavior if its extra features are not intended, so it's used more cautiously. |

1. **Write a Dockerfile to run Nodejs application build an image from it and create a container using that image (also include persistent volume and network in Dockerfile).**

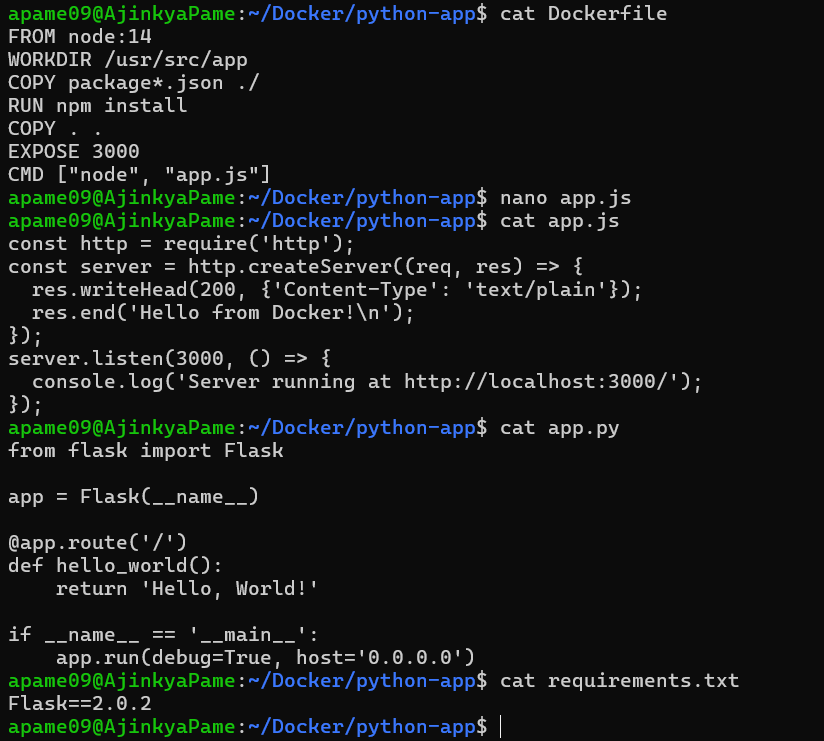


apame09@AjinkyaPame:~/Docker/nodeapp$ docker build -t my-node-app .

[+] Building 789.5s (11/11) FINISHED docker:default



1. **Write a Dockerfile to create a python application build image from it and push that image to private repository of Docker hub.**



apame09@AjinkyaPame:~/Docker/python-app$ docker build -t python-app .

[+] Building 422.9s (12/12) FINISHED docker:default

apame09@AjinkyaPame:~/Docker/python-app$ docker tag python-app ajinkyap009/python-app:v1

apame09@AjinkyaPame:~/Docker/python-app$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

ajinkyap009/python-app v1 ea8be6018de0 8 minutes ago 920MB

python-app latest ea8be6018de0 8 minutes ago 920MB

apame09@AjinkyaPame:~/Docker/python-app$ docker push ajinkyap009/python-app:v1

The push refers to repository [docker.io/ajinkyap009/python-app]

c939e23f6f0a: Pushed

3c2f4835ca85: Pushed

a493a1accad6: Pushed

d16fbca97d09: Pushed

0d5f5a015e5d: Mounted from library/node

3c777d951de2: Pushed

f8a91dd5fc84: Mounted from library/node

cb81227abde5: Pushed

e01a454893a9: Mounted from library/node

c45660adde37: Mounted from library/node

fe0fb3ab4a0f: Mounted from library/node

f1186e5061f2: Mounted from library/node

b2dba7477754: Mounted from library/node

v1: digest: sha256:6334e22815215db43540dec0b37401833b891f7985a8ec9f0a0213cdb4a3d7e0 size: 3047

