

Task 5: Volumes

- Docker volumes to persist data between containers. First, an Apache HTTPD web server is set up using a Docker container. Then, a new HTML file is added to the server using the docker cp command. However, containers are considered ephemeral and can die unexpectedly, potentially leading to data loss. To solve this problem, volumes can be used to persist data across container generations.
 - Setting up the server
 - **docker run --rm -d --name apache -p 80:80 httpd:2.4**
 - This command will start a new container from HTTP 2.4, name it apache, bind port 80 to the host machine (more on this later), and set a flag to delete the container when it stops.

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-4$ docker run --rm -d --name apache -p 80:80 httpd:2.4
Unable to find image 'httpd:2.4' locally
2.4: Pulling from library/httpd
26c5c85e47da: Pull complete
2d29d3837df5: Pull complete
2483414a5e59: Pull complete
e78016c4ba87: Pull complete
757908175415: Pull complete
Digest: sha256:a182ef2350699f04b8f8e736747104eb273e255e818cd55b6d7aa50a1490ed0c
Status: Downloaded newer image for httpd:2.4
c0f4a292e574f459b0d3524edf137446a80052320e433ea4528a080b1593a260
```

- To replace the default index.html file in a new Apache 2.4 installation, use the docker cp command to copy the new index.html file from the host machine to the container. Specify the source path of the new file on the host machine and the destination path in the container. The container name is required in the command, and /usr/local/apache2/htdocs/ is the location where the web server serves HTML from. After replacing the file, running the curl command will show the updated content.

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
c0f4a292e574   httpd:2.4   "httpd-foreground"      About a minute ago    Up About a minute    0.0.0.0:80->80/tcp, :::80->80/tcp    apache
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker cp index.html apache:/usr/local/apache2/htdocs/
Successfully copied 2.05kB to apache:/usr/local/apache2/htdocs/
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ curl localhost
<html><body><h1>It works in Docker!</h1></body></html>ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$
```

- In Docker, volumes are file stores that are independent of Docker containers and function similar to EBS volumes in AWS or mountable media like USB thumb drives. They can be created, deleted, and mounted on containers at specific locations within an image. To list the volumes, run the command "docker volume ls".

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker volume ls
DRIVER      VOLUME NAME
local       myvolume
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$
```

- To remove volume in docker

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker volume rm myvolume
myvolume
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$
```

- a new volume named `httpd_htdocs` is created using the command "docker volume create". The volume is mounted on the container using the `-v` flag, and the modified HTML file is copied into the volume using "docker cp". Finally, the container is stopped using the `--rm` flag.

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker volume rm myvolume
myvolume
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker volume create httpd_htdocs
httpd_htdocs
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker run --rm -d --name apache -p 80:80 -v httpd_htdocs:/usr/local/apache2/htdocs/ httpd:2.4
1396ac7d0388f3585667bf39bef021d1a27358010f0e6605776b4303ad53b72
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker cp index.html apache:/usr/local/apache2/htdocs/
Successfully copied 2.05kB to apache:/usr/local/apache2/htdocs/
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ curl localhost
<html><body><h1>It works in Docker!</h1></body></html>ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker stop apache
apache
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$
```

- As an alternative to using volumes, if you have a directory on your host machine you'd like to use like a volume, you can mount those too. This technique is useful in development environments, where you might want to mount your local repo onto a Docker image and actively modify the contents of a Docker container without rebuilding or copying files to it.
- The `-v` flag to accomplish this is almost identical to the previous one. Simply specify an absolute path to a local directory instead. In our case, we'll pass `.` to specify the 5-volumes directory in this repo, which conveniently contains a modified version of the HTML file.

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker run --rm -d --name apache -p 80:80 -v /home/david/src/docker-training/exercises/basic/5-volumes:/usr/local/apache2/htdocs/ httpd:2.4
691828fb4e0d86efa8382b31e9d87ba4d01598b1b167759aab404a758b52d
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ curl localhost
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of ./</title>
</head>
<body>
<h1>Index of ./</h1>
<ul></ul>
</body></html>
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker stop apache
apache
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-5$
```