

```

run 'docker --help' for more information
root@ip-172-31-11-95:~# docker volume create myappdata
myappdata
root@ip-172-31-11-95:~# docker volume ls
DRIVER      VOLUME NAME
local       2b050dcea2e87235c99f18ef62849191e27f9c99074bba3501741bf9cfdcdc5
local       myappdata
root@ip-172-31-11-95:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    53a18edff809   7 weeks ago    192MB
root@ip-172-31-11-95:~# docker run -itd --name volume_con -p 8080:80 -v myappdata:/usr/share/nginx/html nginx
2c0772731a5f3f120d4812a66a254e0eff02b8fdea684f2f3b90fe017d96d4b3
root@ip-172-31-11-95:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
2c0772731a5f   nginx    "/docker-entrypoint..." 8 seconds ago  Up 7 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  volume_con
root@ip-172-31-11-95:~# docker info
Client: Docker Engine - Community
Version: 20.0.4
Context: default
Debug Mode: false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version: v0.22.0
    Path: /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version: v2.34.0
    Path: /usr/libexec/docker/cli-plugins/docker-compose
Server:
Containers: 1
Running: 1
Paused: 0
Stopped: 0
Images: 1
Server Version: 20.0.4
Storage Driver: overlay2
Backing Filesystem: extfs
Supports d type: true
Using metaCopy: false
Native Overlay Diff: true

```

I created a new volume using docker volume create and attached the same volume to a container while creating it using the -v flag

```

root@ip-172-31-11-95:~# docker exec -it 2c0772731a5f /bin/bash
root@2c0772731a5f:/# cd /usr/share/nginx/html
root@2c0772731a5f:/usr/share/nginx/html# ls
0x.html  index.html
root@2c0772731a5f:/usr/share/nginx/html# >index.html
root@2c0772731a5f:/usr/share/nginx/html# vi index.html
ash: vi: command not found
root@2c0772731a5f:/usr/share/nginx/html# apt install vim
Reading package lists... Done

```

I accessed the container using docker exec and modified the index.html file.

```

root@2c0772731a5f:/usr/share/nginx/html# vi index.html
root@2c0772731a5f:/usr/share/nginx/html# cat index.html
<h4><center> This is custom image for practice volumes </center></h4>
root@2c0772731a5f:/usr/share/nginx/html# touch f1 f2 f3
root@2c0772731a5f:/usr/share/nginx/html# ls
0x.html  f1  f2  f3  index.html
root@2c0772731a5f:/usr/share/nginx/html# exit
exit
root@ip-172-31-11-95:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
2c0772731a5f   nginx    "/docker-entrypoint..." 10 minutes ago  Up 10 minutes  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  volume_con
root@ip-172-31-11-95:~# docker stop 2c0772731a5f
2c0772731a5f
root@ip-172-31-11-95:~# docker rm 2c0772731a5f
2c0772731a5f
root@ip-172-31-11-95:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
0605f51b5c4   nginx    "/docker-entrypoint..." 11 seconds ago  Up 10 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  volume_con
root@ip-172-31-11-95:~# docker exec -it 0605f51b5c4 /bin/bash
root@0605f51b5c4:/# cd /usr/share/nginx/html
root@0605f51b5c4:/usr/share/nginx/html# ls
0x.html  f1  f2  f3  index.html
root@0605f51b5c4:/usr/share/nginx/html#

```

Touch some files f1 f2 f3

After removing the previous container, I created a new container using the same volume and, upon executing into the new container, I was able to access the files created in the previous container.

This demonstrates the persistence of data in Docker volumes, which allows data to be retained even after a container is removed.



This is custom image for practice volumes ✓

Access the application in browser.