Full Stack Development
Lab Docker 5

Using Volumes

1. Lab objectives

In this lab you will create and use a Docker volume

2. Create a volume

Run the following code to create a new volume called **myvol**. Inspect the volume as shown to see the directory on the host file system (in the Linux VM) that will be used for persistent storage.

```
D:\Docker>docker volume create myvol
myvol

D:\Docker>docker inspect myvol
[
    {
        "CreatedAt": "2022-11-23T06:10:02Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/myvol/_data",
        "Name": "myvol",
        "Options": {},
        "Scope": "local"
    }
]
```

3. Mounting the volume

In this section, you will mount the volume on an Ubuntu container. The mount point will be the directory **app** in the container file system. Once you have started container, you will create a new file in the mounted directory then exit the container and prune it so it no longer exists.

In the mount option in the command, the source parameter indicates the volume and the target command identifies the mount point in the container.

```
D:\Docker>docker run -it --name test2 --mount source=myvol,target=/app ubuntu
root@fde8d5a45b11:/# ls
app  bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin
srv  sys  tmp  usr  var
root@fde8d5a45b11:/# cd app
root@fde8d5a45b11:/app# cp /etc/hosts myfile
root@fde8d5a45b11:/app# ls
myfile
root@fde8d5a45b11:/app# exit
exit
```

Prune the containers to eliminate the Ubuntu container. Note that this does not delete the volume.

```
D:\Docker>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
fde8d5a45b1176b3d291201a14bf58bd7bc25c4d20f20032c4b95aaf1b1cbca7

Total reclaimed space: 39B

D:\Docker>docker volume ls
DRIVER      VOLUME NAME
local       myvol
```

4. Remount the volume

Now, rerun the command to create an Ubuntu container with the volume mounted. Once in the container, check the app directory and you should see the file you created in the previous lab.

```
D:\Docker>docker run -it --name test3 --mount source=myvol,target=/app ubuntu
root@14b4b26c0969:/# ls
app  bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin
srv  sys  tmp  usr  var
root@14b4b26c0969:/# cd app
root@14b4b26c0969:/app# ls
myfile
```

Before exiting the container, in a different window. Inspect the container and look for the mount section in the data dump.

```
D:\Docker>docker inspect test3
[
    {
        "Id": "2a2983239acfabba143a3454237b445f964c43b72ff276ba002f3c392f145c9f",
        "Created": "2022-11-23T06:34:55.865526698Z",
        "Path": "bash",
        "Args": [],

--- stuff omitted ---

        "Mounts": [
            {
                "Type": "volume",
                "Source": "myvol",
                "Target": "/app"
            }
        ],
```

Shut down the container and prune the volume. Note that you have to prune the containers because the volume is still being used by the container even if the container is not running.

```
D:\Docker>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
2a2983239acfabba143a3454237b445f964c43b72ff276ba002f3c392f145c9f

Total reclaimed space: 0B

D:\Docker>docker volume prune
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
myvol

Total reclaimed space: 174B
```

End Lab