

Problem case:

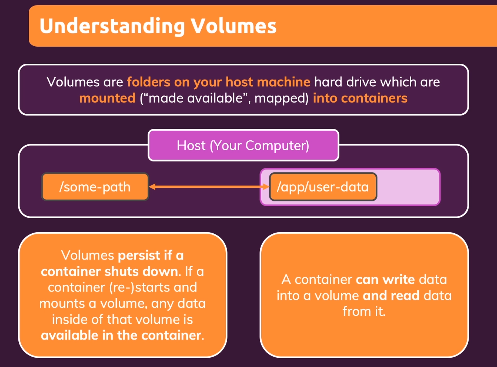
Files are removed after we remove the container but not when we just stop the container

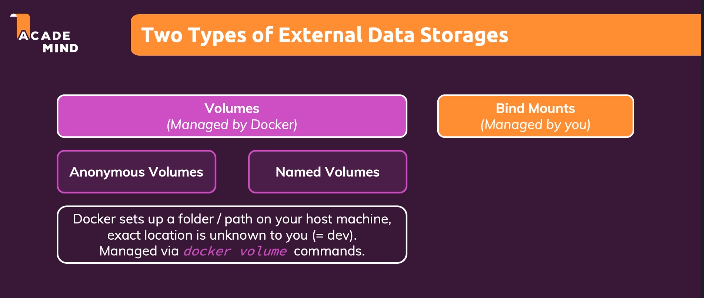
Container does not make any changes to image file system because it is read only

Container makes changes to its own file system and when a container is removed, its file system is removed as well and hence the files are deleted.

Same is not the case when the container is simply stopped. Its file system does not get deleted.

Now, what if we want to keep the files, even if the container is removed





Anonymous volumes - Volumes managed by the docker exist only till the container is running. Once container is removed it cannot be accessed.



So, we use Named Volumes – Great for data which should be persistent but which you don’t need to edit directly

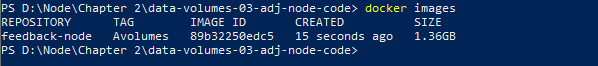
Anonymous volumes

#the below folder in the container should be mapped to host.

VOLUME ["/app/feedback"]

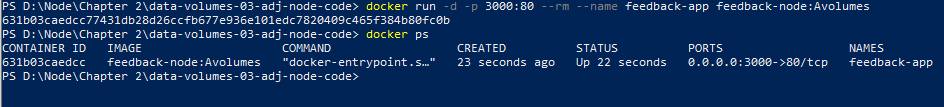
Build the image with tag based on anonymous volume

docker build -t feedback-node:Avolumes .

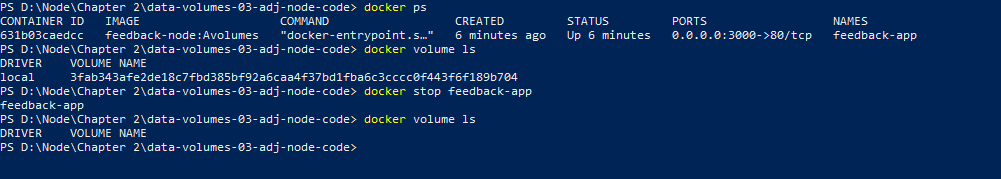


Now, from this image, we will run a container

docker run -d -p 3000:80 --rm --name feedback-app feedback-node:Avolumes



Now, in this case again, if we stop the container, we cannot access the file. Volume name is assigned by docker.



Named Volume

This folder is mapped to some path in the host machine. We don’t know which path because it is mapped by docker

#the below folder in the container should be mapped to host.

VOLUME ["/app/feedback"]

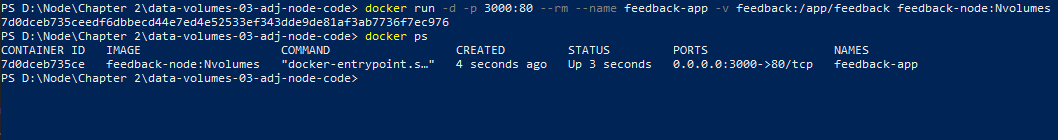
For named volume, we don’t need to mention the VOLUME in docker file. Hence, we can remove it.

Build a new image with named volume tag

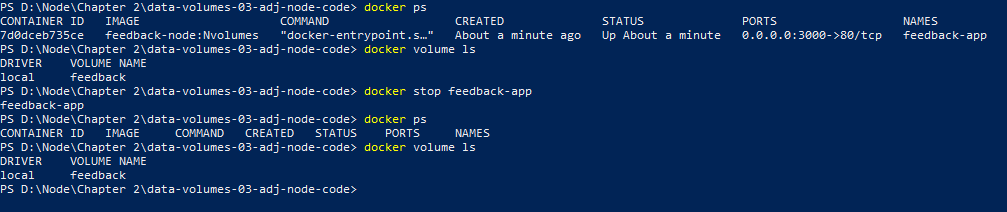


docker run -d -p 3000:80 --rm --name feedback-app -v feedback:/app/feedback feedback-node:Nvolumes

We are telling to create container based on feedback-node:Nvolumes image and map the /app/feedback folder to host under folder name feedback

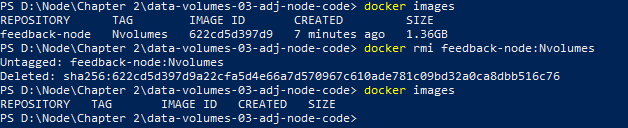


As you can see below, even if we stop the container, the volume is not deleted



docker volume rm feedback -> to remove the volume

Images can be removed using Repository and Tag name in format <Repository name>: <Tag name>



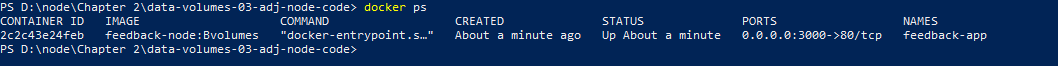
Bind Mounts - You define the folder/path on your host machine. Persistent and data can be edited.

Create an image with bind volume

docker build -t feedback-node:Bvolumes .



PS D:\Node\Chapter 2\data-volumes-03-adj-node-code> docker run -d -p 3000:80 --rm --name feedback-app -v feedback:/app/feedback -v "C:\Users:\app" feedback-node:Bvolumes



docker run -d -p 3000:80 --name feedback-app -v feedback2:/app/feedback -v "D:\Node\Chapter 2\data-volumes-03-adj-node-code\Dockerfile" -v /app/node\_module feedback-node:Bvolumes