**Lab Experiment-2**

* **Sharing data between containers using volume.**

Docker containers are isolated from each other but still they can share data by having a common shared volume.

Volume is a storage on local machine which can be used by containers to store data.

To share data using volume, follow the below given steps:

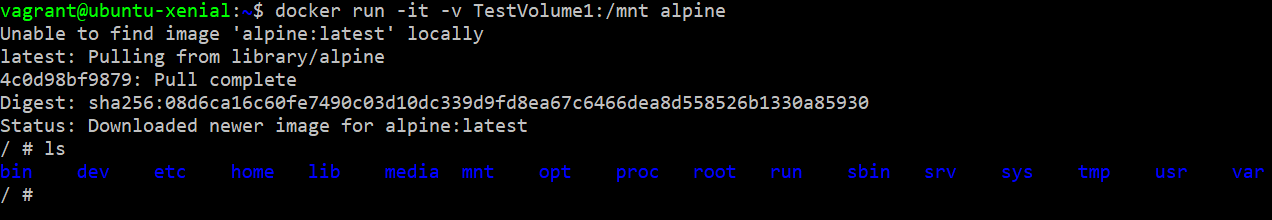
1. Create a docker container and bind any folder of the container with a volume.

Command syntax: docker run –it –v <volume-name>:<folder-name> <image-name>

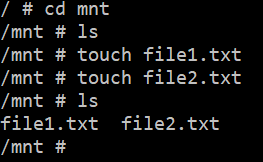
The image being used here is alpine and volume named TestVolume1 is bind to mnt folder of the alpine container by using the following command:

Command: docker run -it -v TestVolume1:/mnt alpine

This command will open the terminal of the newly created alpine container.



2. Move to the directory /mnt and create some files

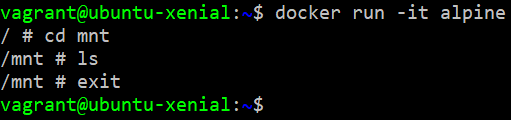


3. Now exit the container and create a new container without specifying any volume to check whether these files are accessible to that container or not.

Command syntax: docker run –it <image-name>

The image being used is alpine.

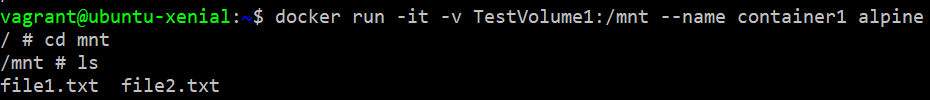
Command: docker run –it alpine



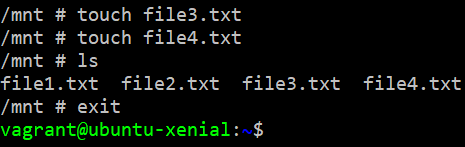
As you can see in the above specified output, the files created in step-2 are not visible in the newly created container.

4. Now let us create a container named “container1” using alpine image and bind its mnt folder with TestVolume1. The files created in step-2 will be present in this container.

Command: docker run –it –v TestVolume1:/mnt --name container1 alpine

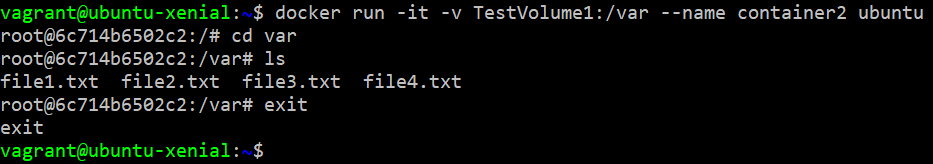


Now we will create some more files in this container and then exit it.



5. Now we will create another container named container2 using ubuntu image and bind var folder of the container to the same volume. The files will be present in this container as well.

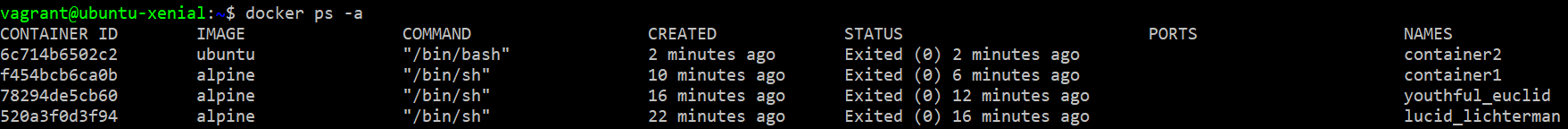
Command: docker run –it –v TestVolume1:/var --name container2 ubuntu



6. We can access the data even when the containers are stopped by directly moving to the location of volume.

First, let us verify that the containers are stopped.

Command: docker ps –a

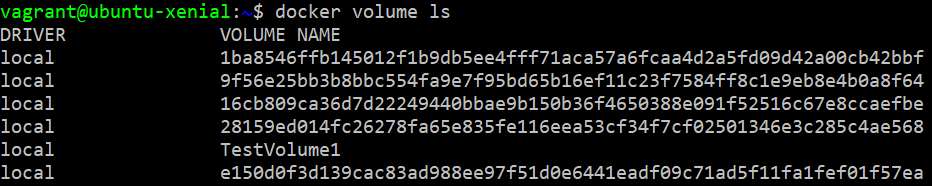


You can see that all containers are in “Exited” state.

Now, we need the name of the location that we need to access.

To list the available docker volumes with their names, use the folowing command:

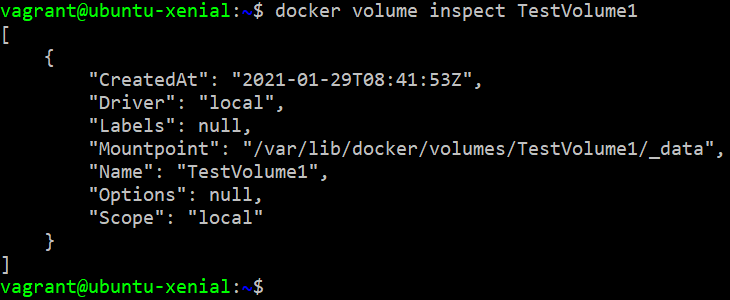
Command: docker volume ls



Here we can see the volume “TestVolume1”.

Now we need to get the path of the TestVolume1 to access its data. The command to display the details of a volume is as follows:

Command: docker volume inspect TestVolume1



The above displayed output shows the path (mountpoint).

All created files will be present at this specified path:

