APPLICATION CONTAINERIZATION

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.

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.

To share the data using volume, given steps are needed to be followed:

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 **Ubuntu** and volume named **central_vol** is attached to **mnt** folder of the Ubuntu container by using the following command:

```
atishay@atishay-HP-15-Notebook-PC:~$ docker run -it -v central_vol:/mnt ubuntu
root@32cd83eb9796:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
```

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

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

```
root@32cd83eb9796:/# cd mnt
root@32cd83eb9796:/mnt# touch atishay.txt docker_file.txt
root@32cd83eb9796:/mnt# ls
atishay.txt  docker_file.txt
```

3. Now exit the container and run an existing container without specifying any volume to check whether these files are accessible to that container or not.

Command syntax: docker exec -it <Container-ID> /bin/bash

```
root@32cd83eb9796:/mnt# exit
exit
atishay@atishay-HP-15-Notebook-PC:~$ docker exec -it 0be2 /bin/bash
root@0be2ff76b1e4:/# cd mnt
root@0be2ff76b1e4:/mnt# ls
root@0be2ff76b1e4:/mnt#
root@0be2ff76b1e4:/mnt# exit
exit
```

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

4. Now let us create another container using Ubuntu image and bind its mnt folder with central_vol

```
atishay@atishay-HP-15-Notebook-PC:~$ docker run -it -v central_vol:/mnt ubuntu
root@6e3d49b3bde7:/# cd mnt
root@6e3d49b3bde7:/mnt# ls
atishay.txt  docker_file.txt
root@6e3d49b3bde7:/mnt# exit
exit
```

We see that files created in Step -2 are present in this new container.

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

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

Command: docker volume ls

```
atishay@atishay-HP-15-Notebook-PC:~$ docker volume ls
DRIVER    VOLUME NAME
local     central_vol
```

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

Command: docker volume inspect <volume-name>

```
atishay@atishay-HP-15-Notebook-PC:~$ docker volume inspect central_vol
[
    {
        "CreatedAt": "2021-01-29T18:48:14+05:30",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/central_vol/_data",
        "Name": "central_vol",
        "Options": null,
        "Scope": "local"
    }
]
```

7. The above displayed output shows the path (mountpoint).

All created files will be present at this specified path:

```
atishay@atishay-HP-15-Notebook-PC:~$ ls /var/lib/docker/volumes/central_vol/_data
ls: cannot access '/var/lib/docker/volumes/central_vol/_data': Permission denied
atishay@atishay-HP-15-Notebook-PC:~$ sudo ls /var/lib/docker/volumes/central_vol/_data
atishay.txt  docker_file.txt
atishay@atishay-HP-15-Notebook-PC:~$
```