**Data Management in Docker through Volumes and Bind mounts**

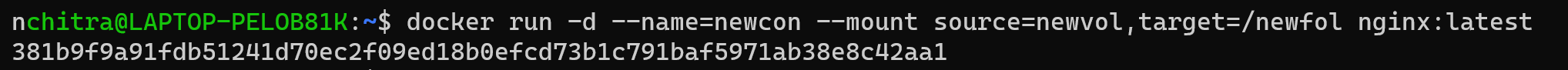
Ques-1: Do the following

1. Run a container with docker volume
2. Inspect whether the volume is attached with the container or not
3. Create a file inside the volume
4. Stop container, remove the container and then remove the volume

**1)Volumes - Using the “- -mount” flag**

**Step-1:** Firstly we are creating and running a container named “newcon” based on the “nginx:latest” image in detached mode, while also mounting a volume named “newvol” into the container at the path “/newfol” for data sharing or persistence.

docker run -d --name=newcon --mount source=newvol, target=/newfol nginx:latest

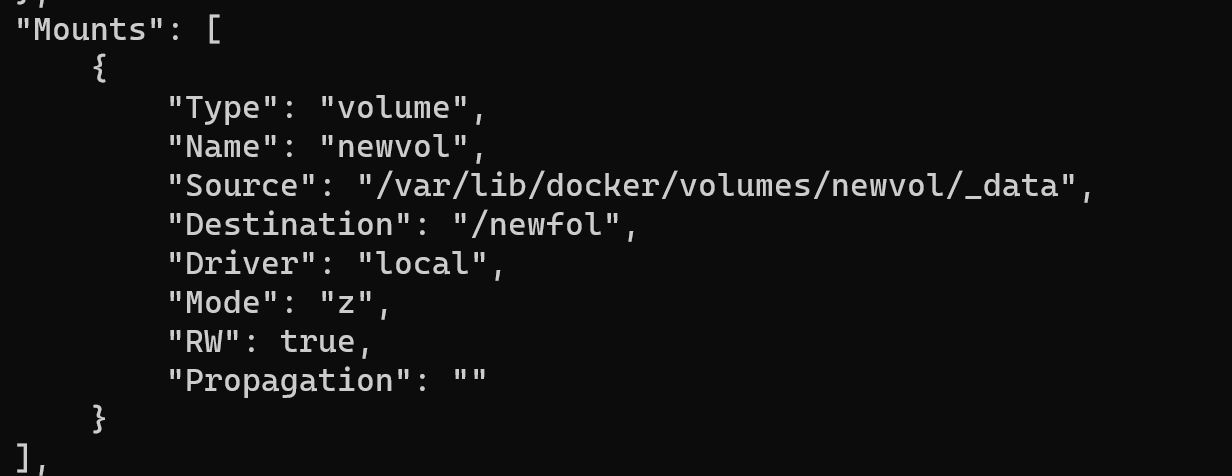


**Step-2:** Now using the inspect command we are retrieving the detailed information about the docker container and checking whether the volume is mounted or not.

docker inspect container newcon

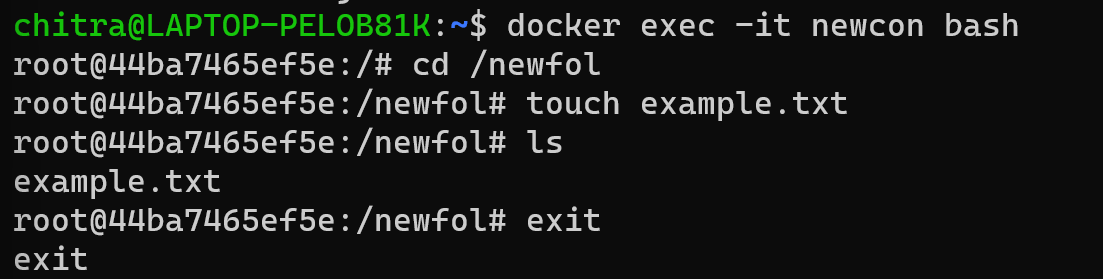


**Step-3:** In the “Mounts” section, we can see that the volume is successfully mounted into the container.



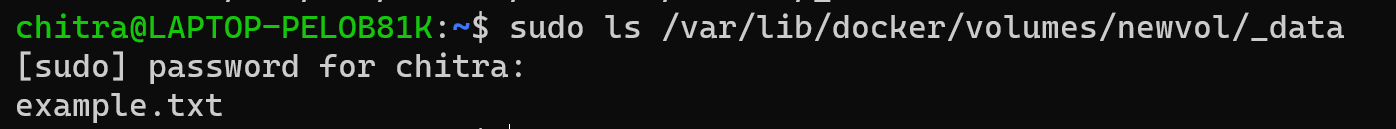
**Step-4:** Now in order to create a file inside the volume, we need to start an interactive shell session within the running docker container, “newcon”.

docker exec -it newcon bash



**Step-5:**After creating the file, exit from the interactive mode and then check whether the file is present in the respective folder or not using the command:

sudo ls /var/lib/docker/volumes/newvol/\_data

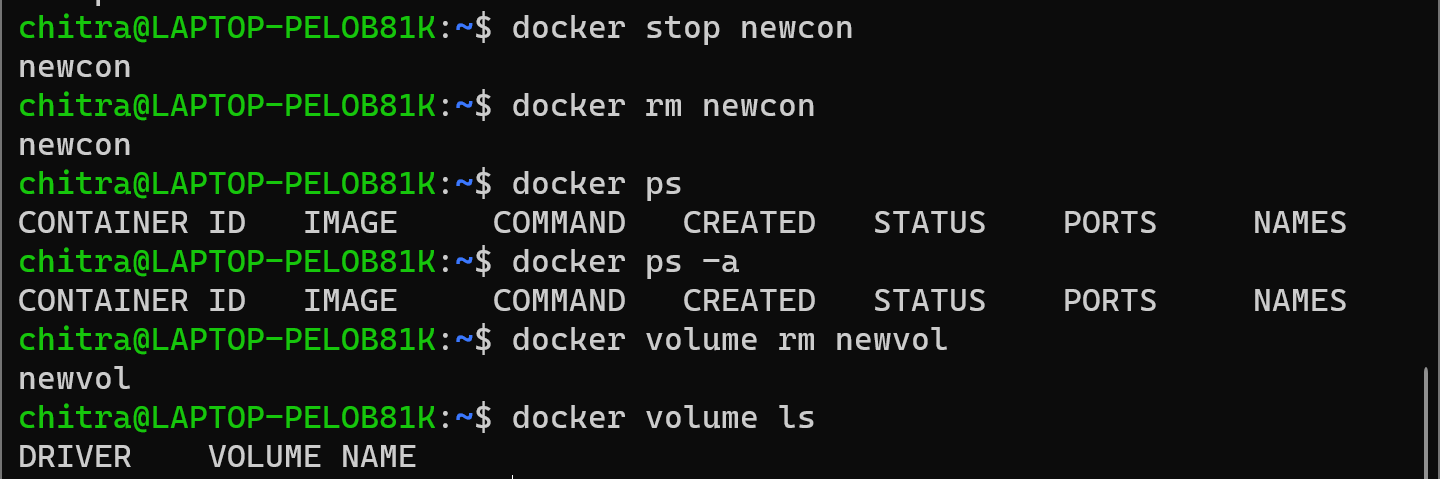


**Step-6:** The file is present. So now stop the container, remove the container and then remove the volume.

docker stop newcon

docker rm newcon

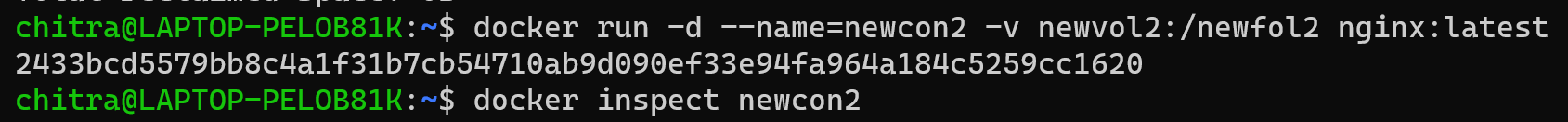
docker volume rm newvol



**2)Volumes - Using the “- v” flag**

**Step-1:** Again, we are creating and running the container named “newcon2” based on the “nginx:latest” image, while creating and using the docker volume, “newvol”, with the volume mounted at “/newfol2” using the -v flag.

docker run -d --name=newcon2 -v newvol2:/newfol2 nginx:latest



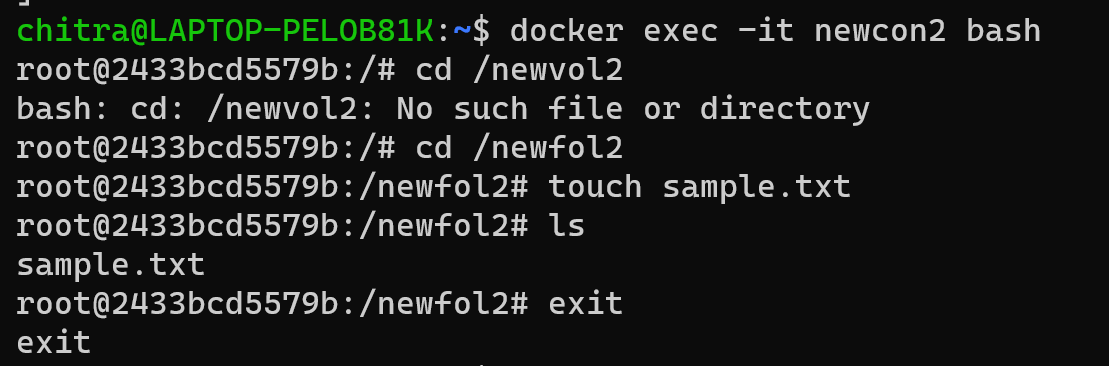
**Step-2:** Now using the following command we are retrieving the information about the container and checking whether the volume is mounted or not. In the “Mounts” section, we can see that the volume is successfully mounted into the container.

docker inspect newcon2



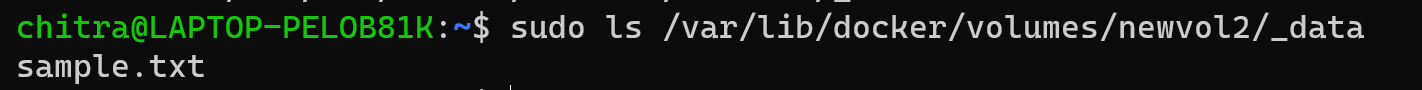
**Step-3:** Now in order to create a file inside the volume, we need to start an interactive shell session within the running docker container, “newcon2”.

docker exec -it newcon2 bash



**Step-4:**After creating the file, exit from the interactive mode and then check whether the file is present in the respective folder or not using the command:

sudo ls /var/lib/docker/volumes/newvol2/\_data

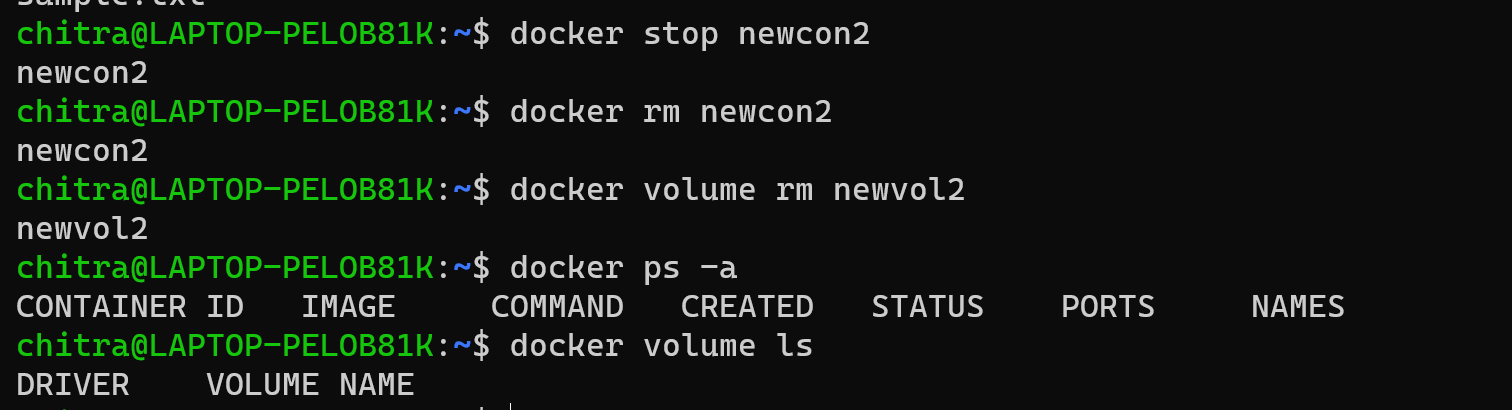


**Step-5:** The file is present. So now stop the container, remove the container and then remove the volume.

docker stop newcon2

docker rm newcon2

docker volume rm newvol2



**Note:** In the context of data management in Docker through **volumes**, when you use **-v** or **--mount** to create a volume for a file or directory that does not exist on the Docker host, Docker automatically generates the endpoint. However, in the case of a **bind mount**, if you use **-v** to bind mount a file or directory that does not exist on the Docker host, Docker will create the endpoint. This endpoint is always created as a directory. Conversely, if you use **--mount** to bind mount a file or directory that does not exist on the Docker host, Docker will not create it for you; instead, it will generate an error.

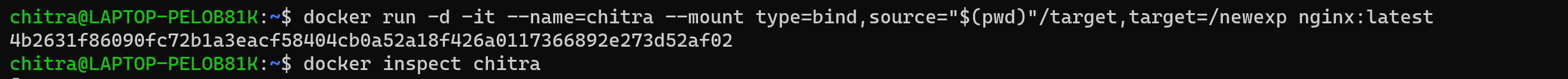
Let's now observe this behaviour in action.

**1)Bind Mount - Using the “- - mount” flag**

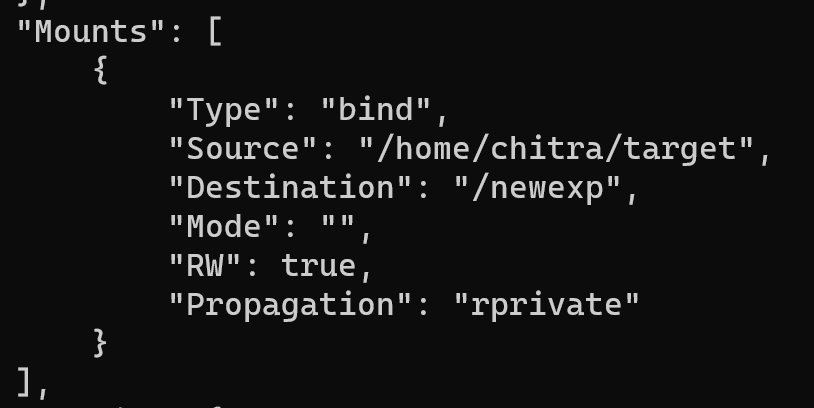
**Step-1:** Firstly, we are simply creating a running the container and attaching a volume to it using the “- -mount” flag**.** Here the folder target is already made and we are using this folder.

docker run -d -it --name=chitra --mount type=bind, source="$(pwd)"/target,target=/newexp nginx:latest

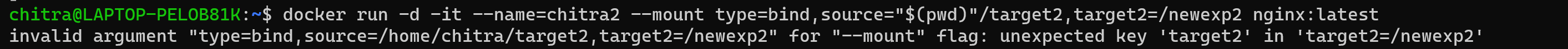
docker inspect chitra



**Step-2:** In the “Mounts” section, we can see that the volume is mounted of type bind.



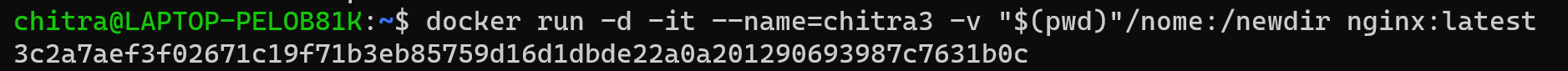
**Step-3:** Now we are again doing the same thing that we did in step 1 but not the folder that we are using i.e. “target2” does not exist. In this case, the error is generated.



Hence, we can say that if we use bind mount, we have to create a folder or directory beforehand in order to bind mount a folder or directory.

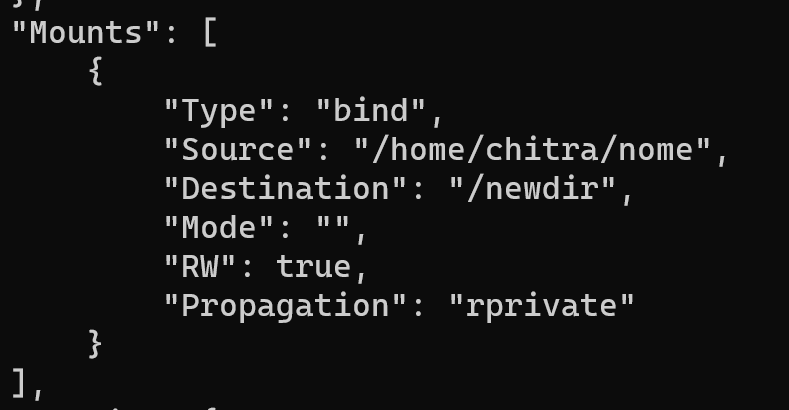
**2)Bind Mount - Using the “-v” flag**

**Step-1:** Here we are creating and running the container and using the -v flag to bind mount a folder that does not exist already.

****

**Step-2:** Now inspect the container and check the “Mounts” section whether it is mounted successfully or not.

****

****