# **Docker Storages**

# 

Upon launching a container, we might perform a few operations, store some data. Data inside the container is written inside the container layer. This data can be said as ephemeral which means that it is temporary.

Ever wondered what happens if we remove the container?

When we remove a container from our system, the files and data stored inside the container will all be erased. This could be concerning as there might be some important files stored inside the container that should be preserved for future references.

So, in order to store such valuable data, we need a mechanism which would secure/backup our data even after the container is removed. In such cases, we can use docker storages. Docker storages also known as docker volumes are a method where we attach a storage drive to secure the data. Docker allows us to store data in either the file system or even the memory.

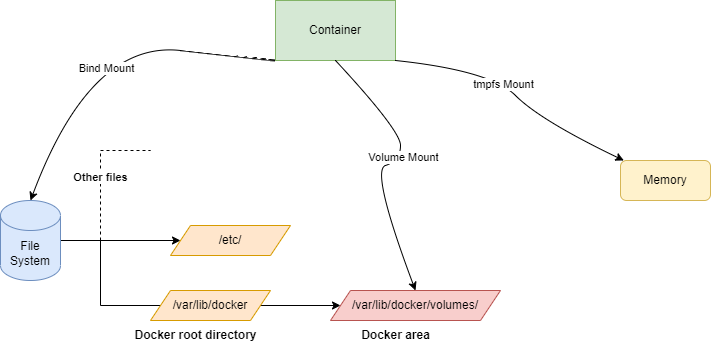
Depending upon the requirements, we can choose between the three existing types of mounts.

This data can be said as persistent.

The types of volumes mounts available are:

1. Volume mount
2. Bind mount
3. Tmpfs mount

VISUALPATH: #205, 2nd.Floor, Nilgiri Block, Aditya Enclave,

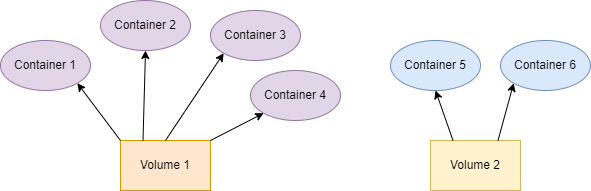


Let us understand how these volumes mounts function and their use cases;

***Volume***:

Volumes mounts are created under the docker root directory of the file system. Volumes can be created using the **“*docker volume create*”** command.

Single volume can be shared between multiple containers. Which usually means that any changes made in one container will also be reflected in other containers too.



In case we want to avoid such scenarios where we don’t want a container to make changes, we can mention it in the command by passing the **“readonly”** option. This usually means that the particular container is restricted from making any changes.

***Bind Mount*:**

Bind mounts allow us to mount the container to any part of the file system. Which usually means that we have to be careful with the sensitive file locations. Bind mount is most useful when we/client has a requirement that his data is present in a specific location that we need to access, in such cases bind mount can be used to mount the location to the container. In case of bind mount also, we can share it within multiple containers.

Mounting can be done starting from the root file system “ **/** ”.

***tmpfs Mount*:**

tmpfs mount is used under the scenario where our data is not very important and there are no sensitive files present in the container. If the data that we are going to have is temporary, then it is usable. It does not store the data of containers in disk.

But rather tmpfs mount stores the data in memory of the host system(RAM) which upon the removal of the container gets erased. Unlike volume and bind mounts, we cannot share a tmpfs mount between multiple containers.

We shall see the further execution of the volumes in the next file..