**Volumes**

* Volumes are persistent data stores for containers, created and managed by Docker. You can create a volume explicitly using the docker volume create command, or Docker can create a volume during container or service creation.
* When you create a volume, it's stored within a directory on the Docker host. When you mount the volume into a container, this directory is what's mounted into the container. This is similar to the way that bind mounts work, except that volumes are managed by Docker and are isolated from the core functionality of the host machine.
* **WHERE SHOULD WE USE VOLUMES ?**
* Volumes are easier to back up or migrate than bind mounts.
* You can manage volumes using Docker CLI commands or the Docker API.
* Volumes work on both Linux and Windows containers.
* Volumes can be more safely shared among multiple containers.
* New volumes can have their content pre-populated by a container or build.
* When your application requires high-performance I/O.
* **LIFE-CYCLE OF A VOLUME**
* A volume's contents exist outside the lifecycle of a given container. When a container is destroyed, the writable layer is destroyed with it. Using a volume ensures that the data is persisted even if the container using it is removed.
* A given volume can be mounted into multiple containers simultaneously. When no running container is using a volume, the volume is still available to Docker and isn't removed automatically. You can remove unused volumes using docker volume prune.
* **COMMAND\_TO MOUNT**
* **docker run --mount type=volume,src=<volume-name>,dst=<mount-path>**
* **docker run --volume <volume-name>:<mount-path>**