# Docker data volumes

#### Data volumes

A data volume is a directory within a container that is meant to persist beyond the life cycle of the container. For this reason, volumes are not automatically deleted when a container is removed. All volumes are stored on the Docker host in a system path, meaning that volumes can be shared and reused among containers. Changes to a volume are made directly and are not reflected in the Docker image, because they bypass the Union File System.

#### Data volume containers

A data volume container (DVC) is a container that houses a volume and whose sole aim is to store data in a persistent way. Because volumes can be shared with other containers, DVCs are often used as a centralized data store across multiple containers on the same Docker host. Other containers can mount the volume inside a DVC and save their data to it, providing non-persistent containers with a way to handle persistent storage.

# There are three main use cases for Docker data volumes:

To keep data around, even through container restarts

To share data between the host filesystem and the Docker container

To share data with other Docker containers

https://www.digitalocean.com/community/tutorials/how-to-work-with-docker-data-volumes-on-ubuntu-14-04

### **Working Without Docker Data Volumes**

1 \$ docker run -t -i ubuntu /bin/bash

Ubuntu container bash shell root@cbceb444ec6a:/#

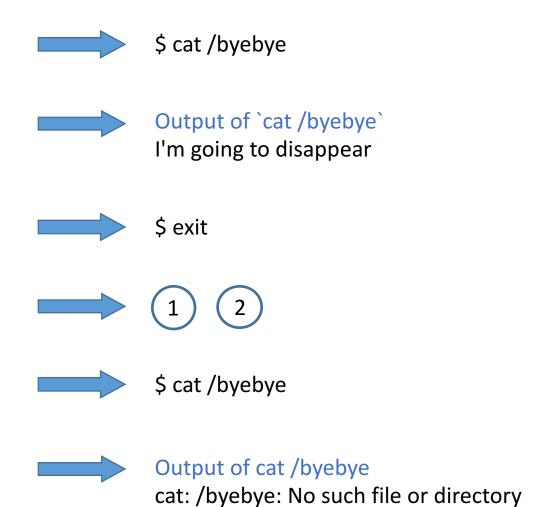
\$ echo "I'm going to disappear" > /byebye

ls /

Output of ls /

bin boot byebye dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

## **Working Without Docker Data Volumes**



#### **Keeping Data Persistent**

-V

--volumes-from

\$ docker create -v /tmp --name datacontainer ubuntu

\$ run -t -i --volumes-from datacontainer ubuntu /bin/bash

\$ echo "I'm not going anywhere" > /tmp/hi

\$ exit

\$ run -t -i --volumes-from datacontainer ubuntu /bin/bash

\$ cat /tmp/hi

Output of cat /tmp/hi
I'm not going anywhere

#### Sharing Data Between the Host and the Docker Container





This run command is a little different from the ones we've used so far, so let's break it down piece by piece:

- -v ~/nginxlogs:/var/log/nginx We set up a volume that links the /var/log/nginx directory from inside the Nginx container to the ~/nginxlogs directory on the host machine. Docker uses a : to split the host's path from the container path, and the host path always comes first.
- -d Detach the process and run in the background. Otherwise, we would just be watching an empty Nginx prompt and wouldn't be able to use this terminal until we killed Nginx.
- -p 5000:80 Setup a port forward. The Nginx container is listening on port 80 by default, and this maps the Nginx container's port 80 to port 5000 on the host system.