MANAGE DATA IN DOCKER

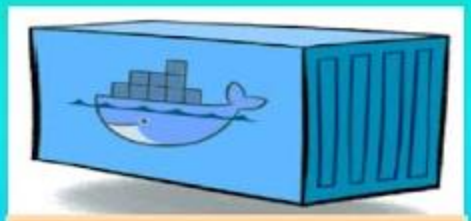
By default all files created inside a container are stored on a writable container layer. This means that:

- The data doesn't persist when that container no longer exists, and it can be difficult to get the data out of the container if another process needs it.
- Docker has two options for containers to store files in the host machine, so that the files are persisted even after the container stops: ***volumes*****, and *bind mounts***. If you're running Docker on Linux you can also use a *tmpfs mount*. If you're running Docker on Windows you can also use a *named pipe*.

Choose the right type of mount

- No matter which type of mount you choose to use, the data looks the same from within the container. It is exposed as either a directory or an individual file in the container's filesystem.
- An easy way to visualize the difference among volumes, bind mounts, and tmpfs mounts is to think about where the data lives on the Docker host.

Docker - Manage Data



By default Docker does not persist data

To persist Data, Docker provides 3 options

Option1 → Volume

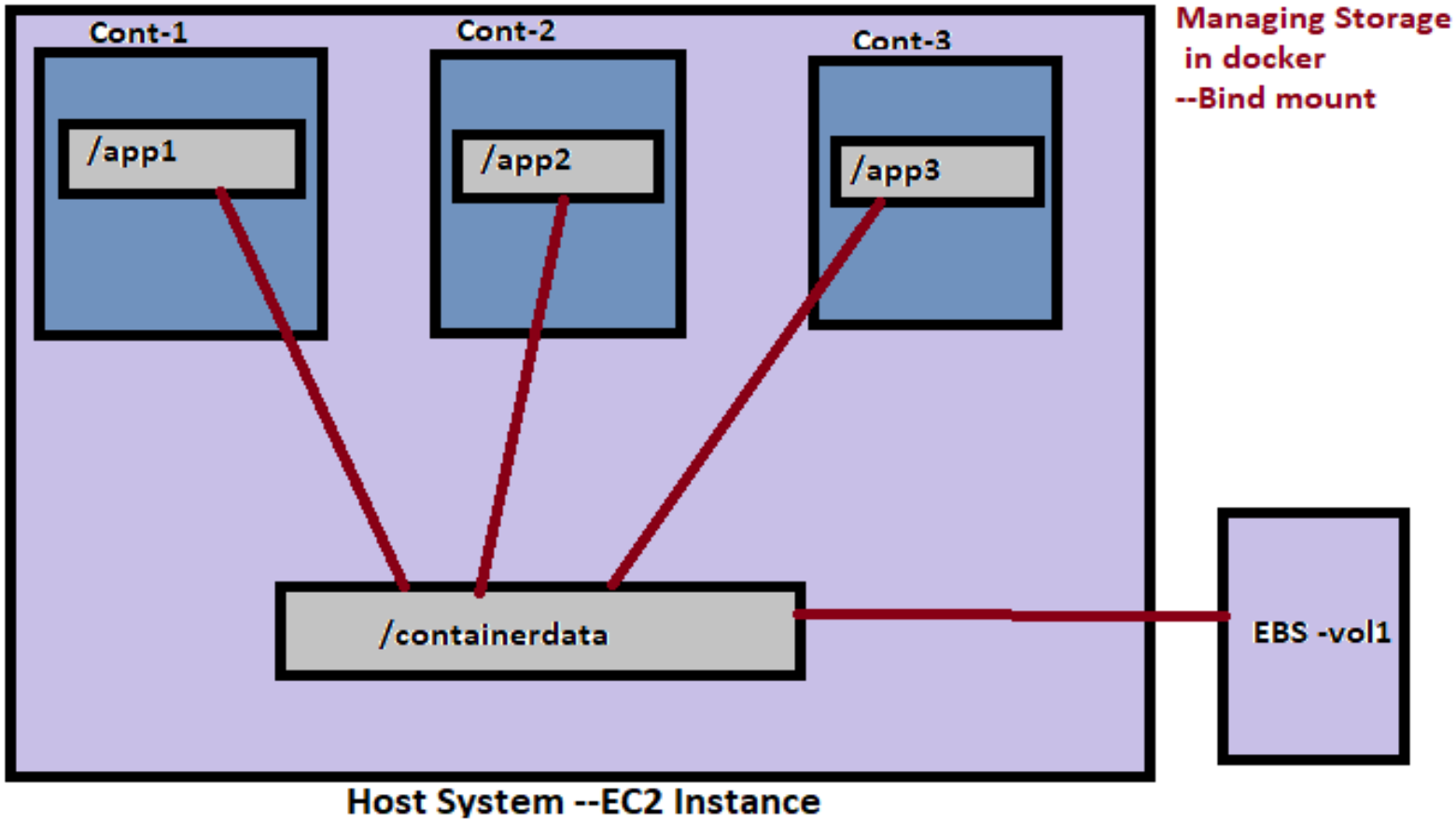
Option2 → Bind Mounts

Option3 Linux only → tmpfs

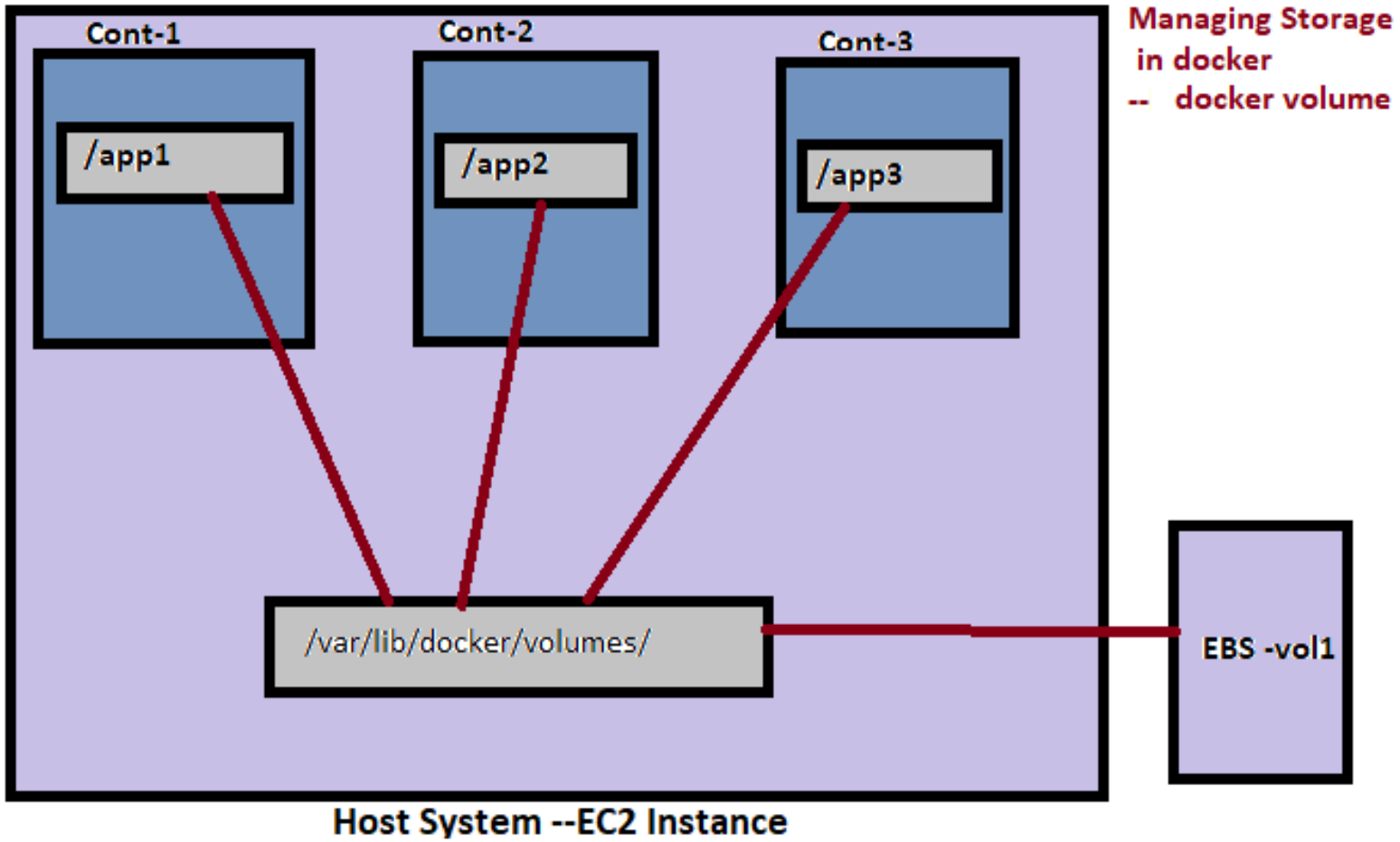
DOCKER

MANAGE DATA IN DOCKER

- **Volumes** are stored in a part of the host filesystem which is *managed by Docker* (/var/lib/docker/volumes/ on Linux). Non-Docker processes should not modify this part of the filesystem. Volumes are the best way to persist data in Docker.
- **Bind mounts** may be stored *anywhere* on the host system. They may even be important system files or directories. Non-Docker processes on the Docker host or a Docker container can modify them at any time.
- **tmpfs mounts** are stored in the host system's memory only, and are never written to the host system's filesystem.
-



Managing Storage
in docker
--Bind mount



Managing Storage
in docker
-- docker volume

Bind mount

```
# mkdir   /home/deepak/dockerfile
# docker  run   -it  -v  /home/deepak/dockerfile:/app  -d ubuntu
# docker ps
#docker   exec  -it  cont:id  bash
# cd /app
# ls
```

create some files here and check it in host system

Docker volume

Syntax: docker volume create volumename

docker volume create test1 --> creating new volume

docker volume ls

docker run -it --mount source=test1,target=/app -d ubuntu

docker ps

docker exec -it cont:id bash

cd /app

create some files here

Now make duplicate session of host OS and launch one more container and check the data

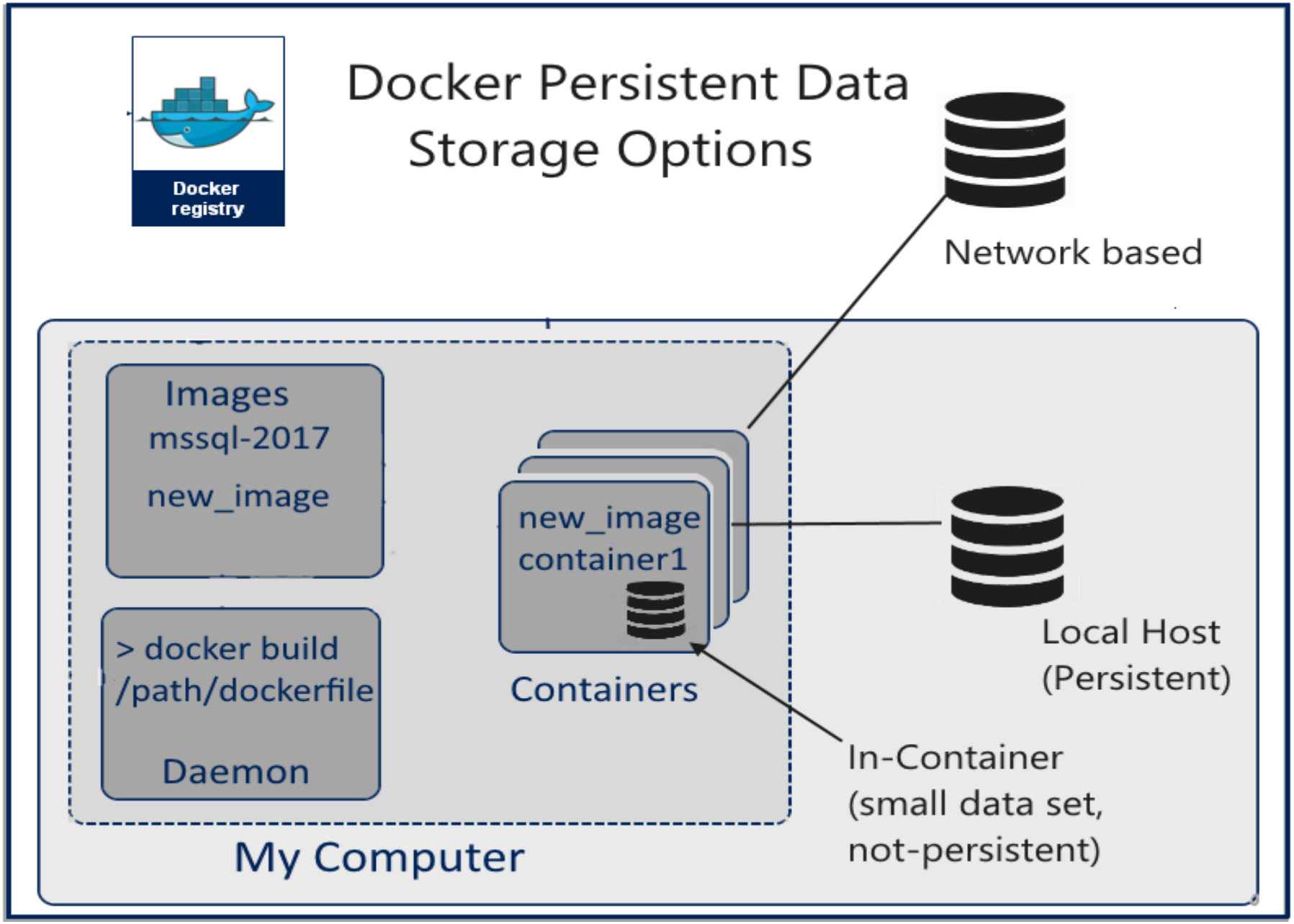
docker run -it --mount source=test1,target=/app1 -d ubuntu

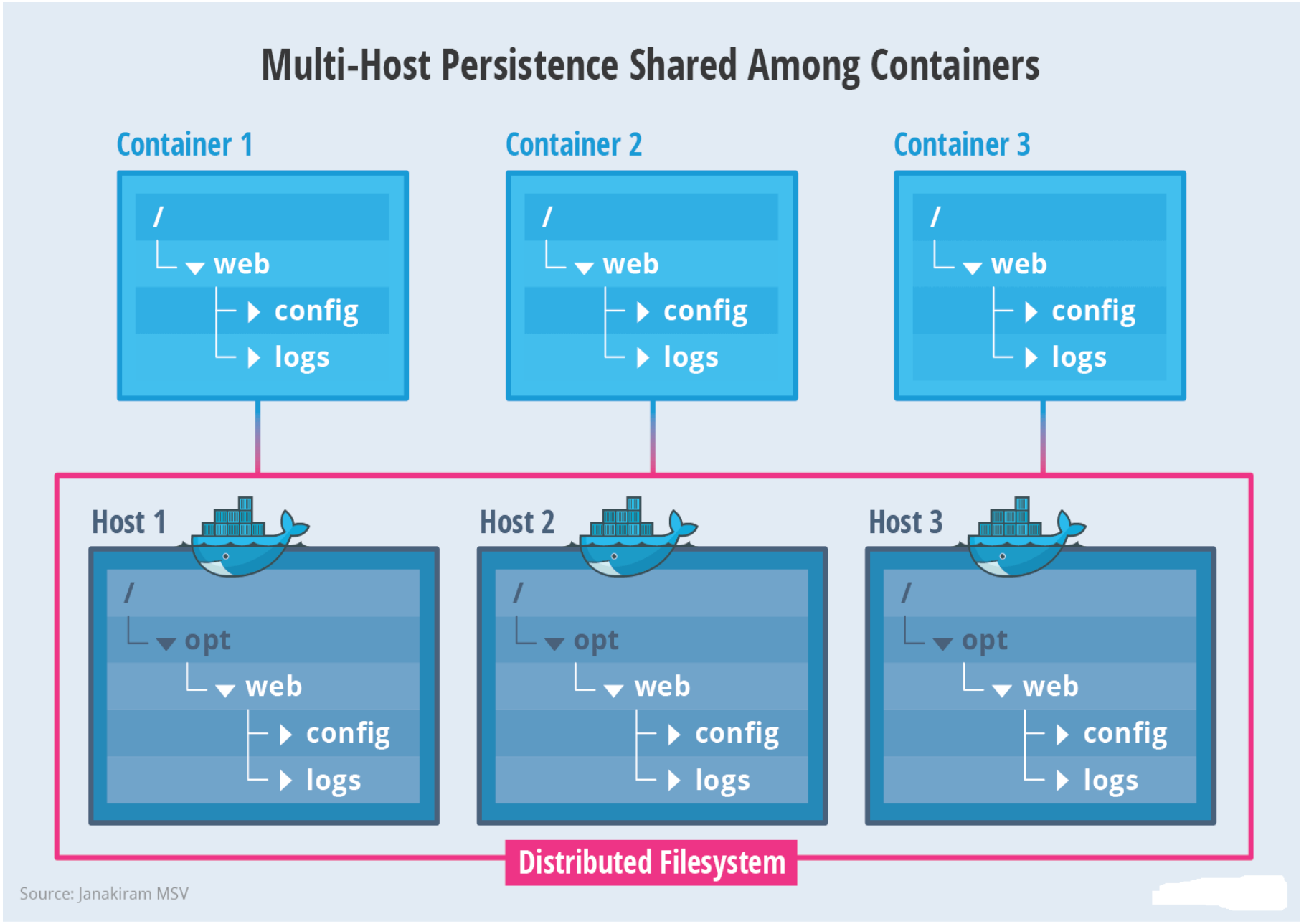
docker ps

docker exec -it cont:id bash

ls /app1

create some files here and check in 1st container


Docker registry
Docker Persistent Data Storage Options
Network based
Images
mssql-2017
new_image
> docker build /path/dockerfile
Daemon
new_image
container1
Containers
Local Host
(Persistent)
In-Container
(small data set,
not-persistent)
My Computer


Multi-Host Persistence Shared Among Containers
Container 1
/
web
config
logs
Container 2
/
web
config
logs
Container 3
/
web
config
logs
Host 1
/
opt
web
config
logs
Host 2
/
opt
web
config
logs
Host 3
/
opt
web
config
logs
Distributed Filesystem
Source: Janakiram MSV