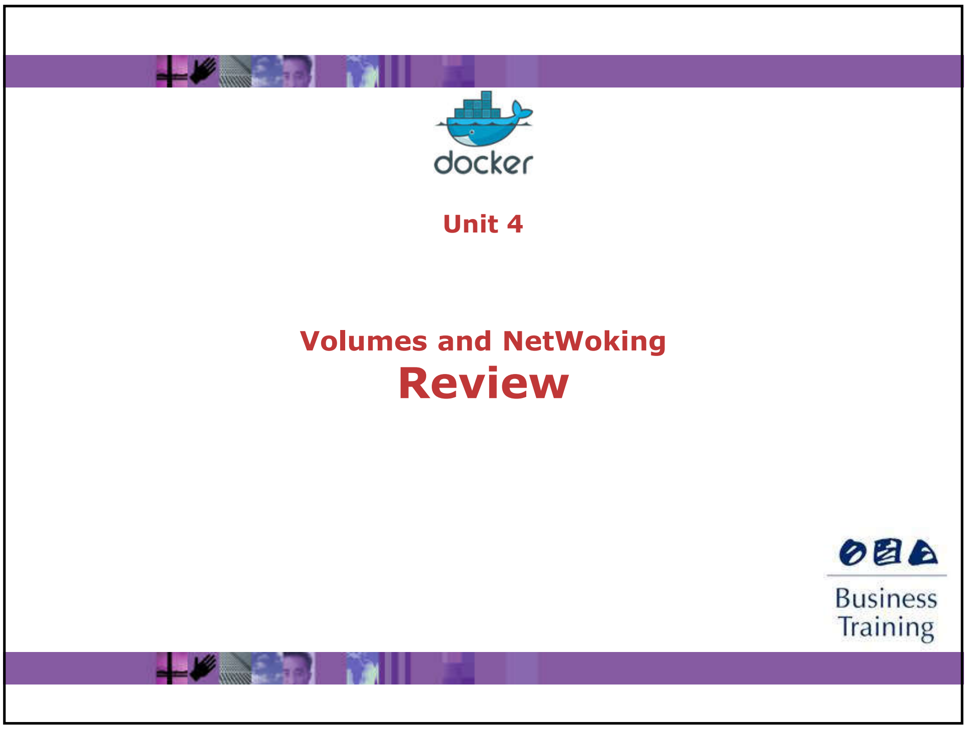


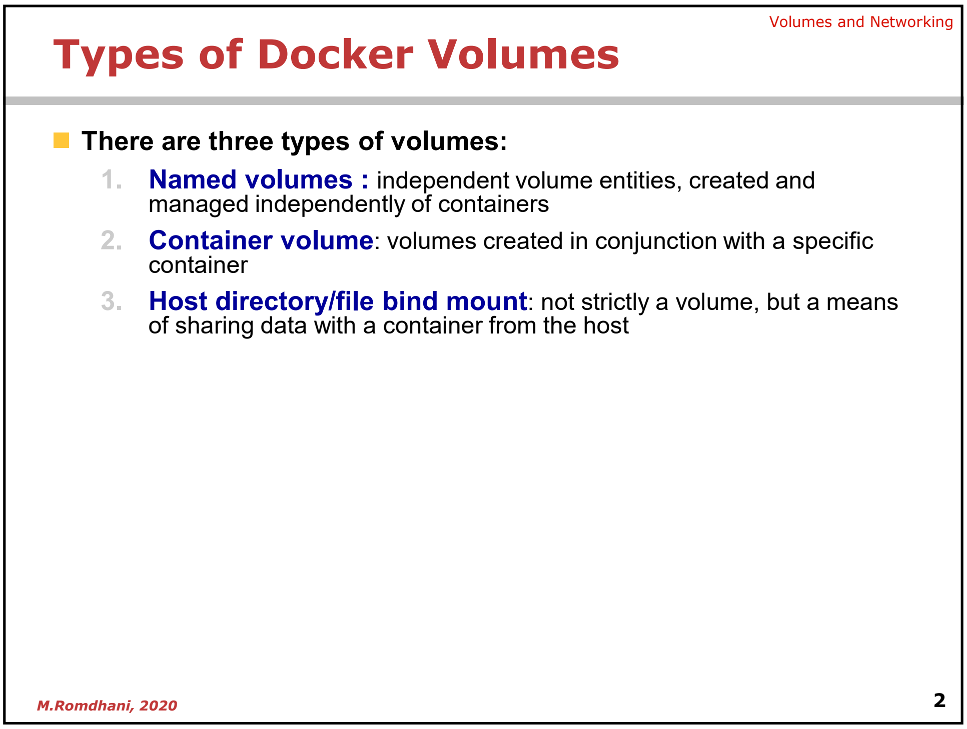
Unit 4

Volumes and NetWoking

Review

Business Training

1



Volumes and Networking

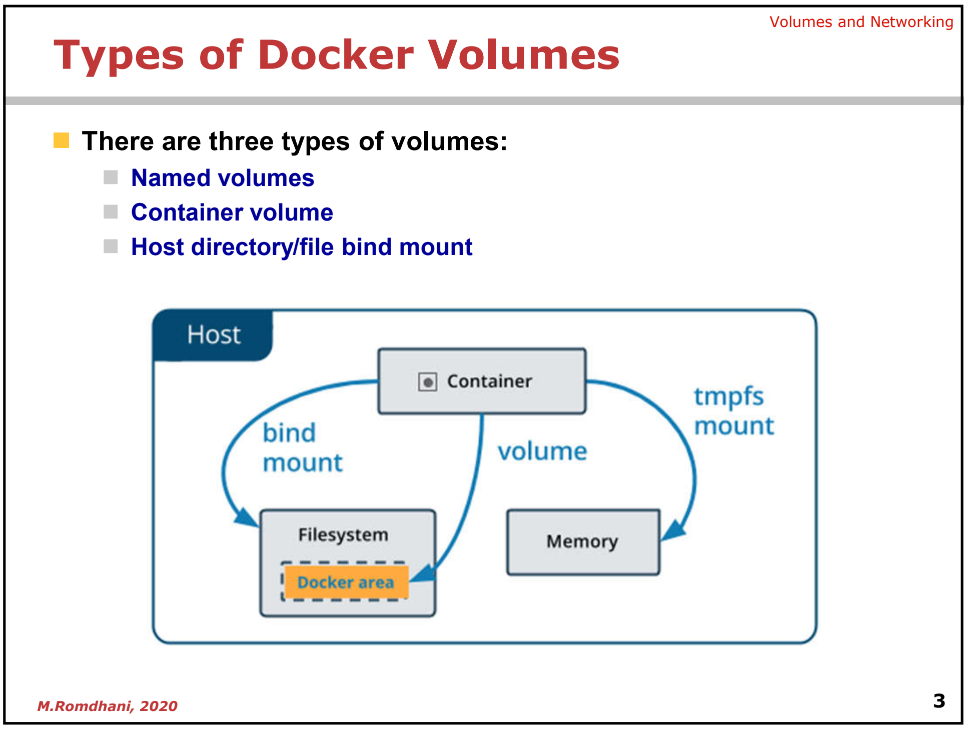
Types of Docker Volumes

- **There are three types of volumes:**
  1. **Named volumes :** independent volume entities, created and managed independently of containers
  2. **Container volume**: volumes created in conjunction with a specific container
  3. **Host directory/file bind mount**: not strictly a volume, but a means of sharing data with a container from the host

M.Romdhani, 2020

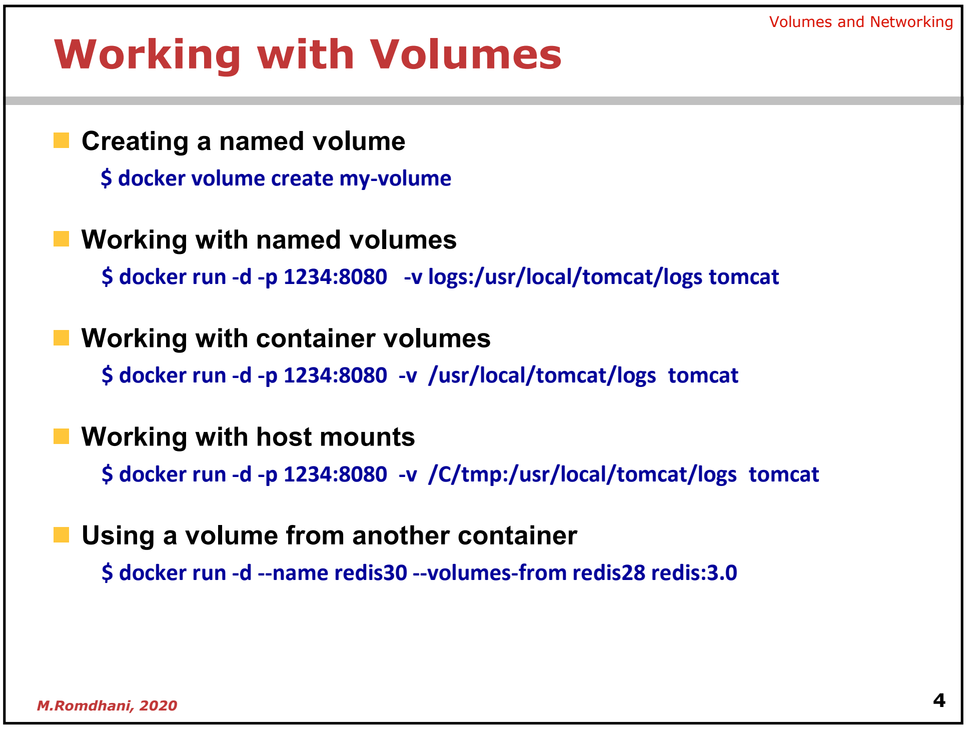
2

2



Types of Docker Volumes

- There are three types of volumes:
  - Named volumes
  - Container volume
  - Host directory/file bind mount



Working with Volumes

- Creating a named volume

```
$ docker volume create my-volume
```

- Working with named volumes

```
$ docker run -d -p 1234:8080  -v logs:/usr/local/tomcat/logs tomcat
```

- Working with container volumes

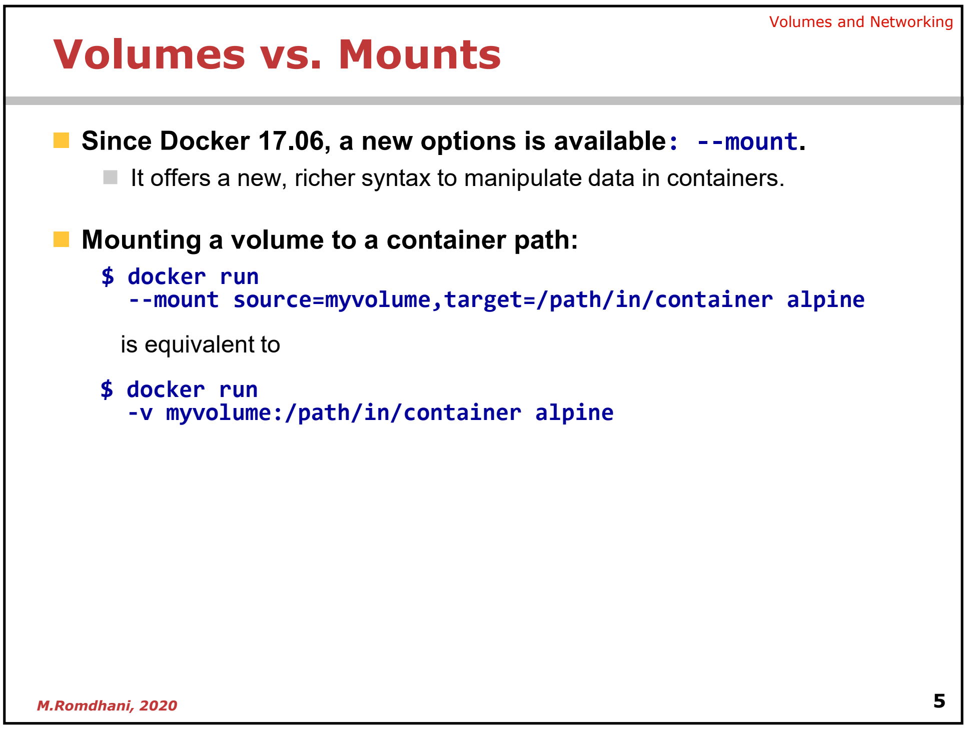
```
$ docker run -d -p 1234:8080  -v  /usr/local/tomcat/logs  tomcat
```

- Working with host mounts

```
$ docker run -d -p 1234:8080  -v  /C/tmp:/usr/local/tomcat/logs  tomcat
```

- Using a volume from another container

```
$ docker run -d --name redis30 --volumes-from redis28 redis:3.0
```



Volumes and Networking

Volumes vs. Mounts

- Since Docker 17.06, a new options is available: `--mount`.
  - It offers a new, richer syntax to manipulate data in containers.

- Mounting a volume to a container path:

```
$ docker run
  --mount source=myvolume,target=/path/in/container alpine
```

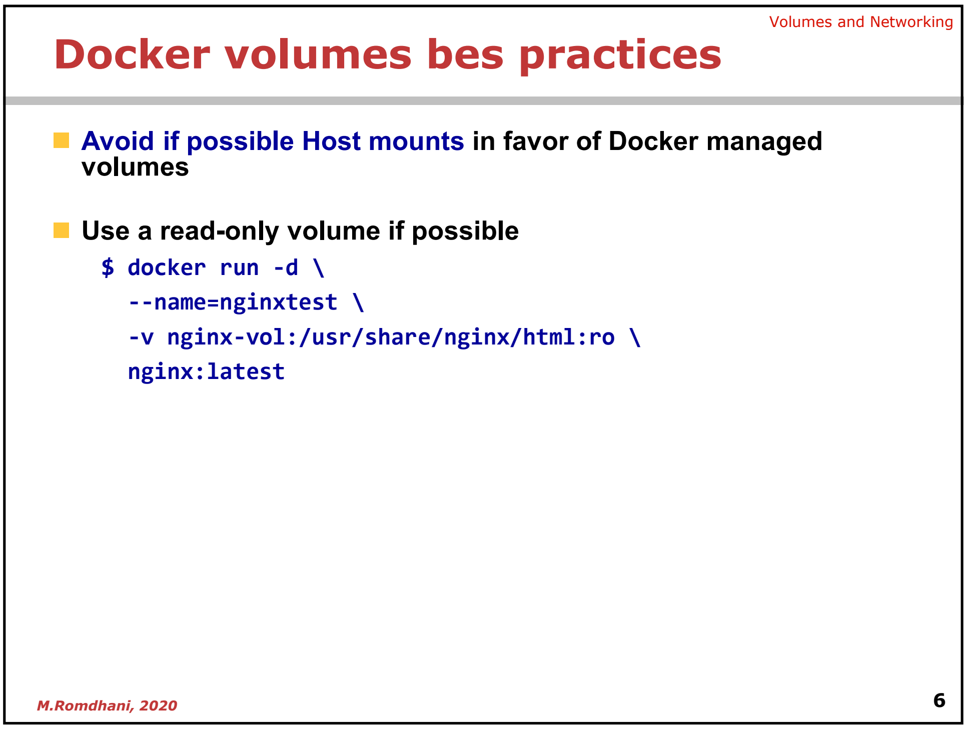
is equivalent to

```
$ docker run
  -v myvolume:/path/in/container alpine
```

M.Romdhani, 2020

5

5



Volumes and Networking

Docker volumes bes practices

- Avoid if possible Host mounts in favor of Docker managed volumes

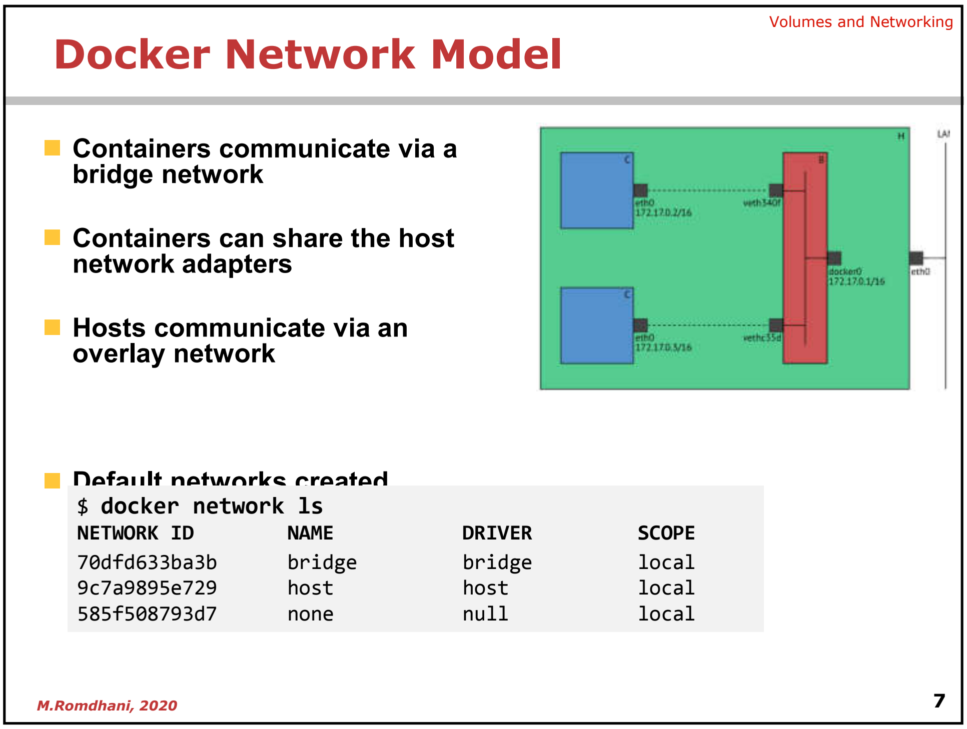
- Use a read-only volume if possible

```
$ docker run -d \
  --name=nginxtest \
  -v nginx-vol:/usr/share/nginx/html:ro \
  nginx:latest
```

M.Romdhani, 2020

6

6



Volumes and Networking

Docker Network Model

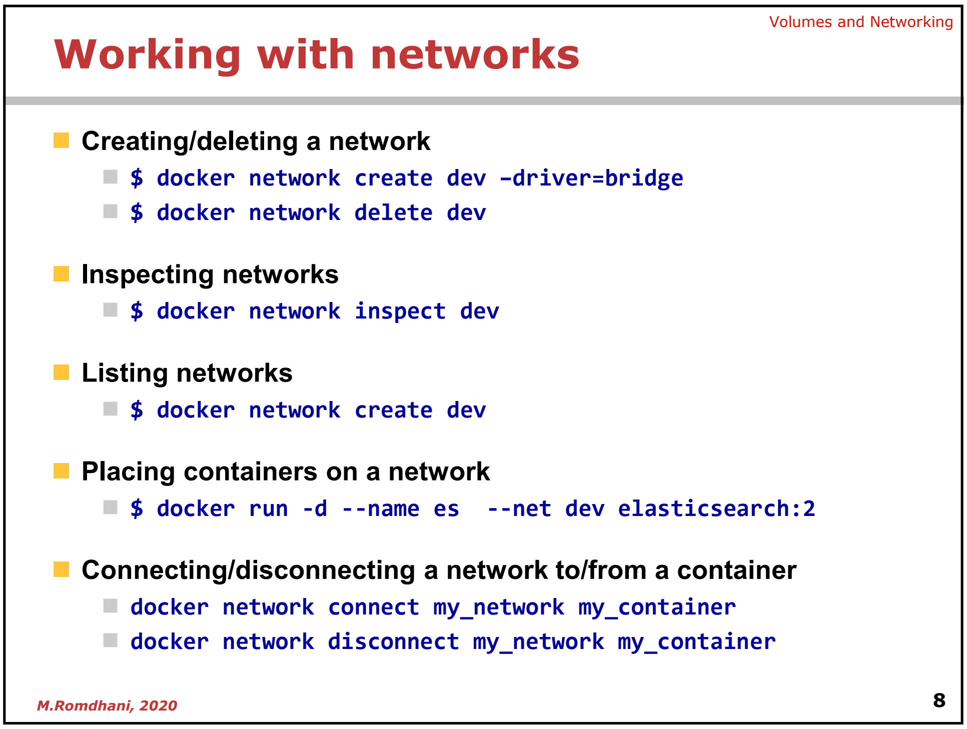
- Containers communicate via a bridge network
- Containers can share the host network adapters
- Hosts communicate via an overlay network

- Default networks created

```
$ docker network ls
NETWORK ID      NAME        DRIVER      SCOPE
70dfd633ba3b    bridge      bridge      local
9c7a9895e729    host        host        local
585f508793d7    none        null        local
```

M.Romdhani, 2020 7

7



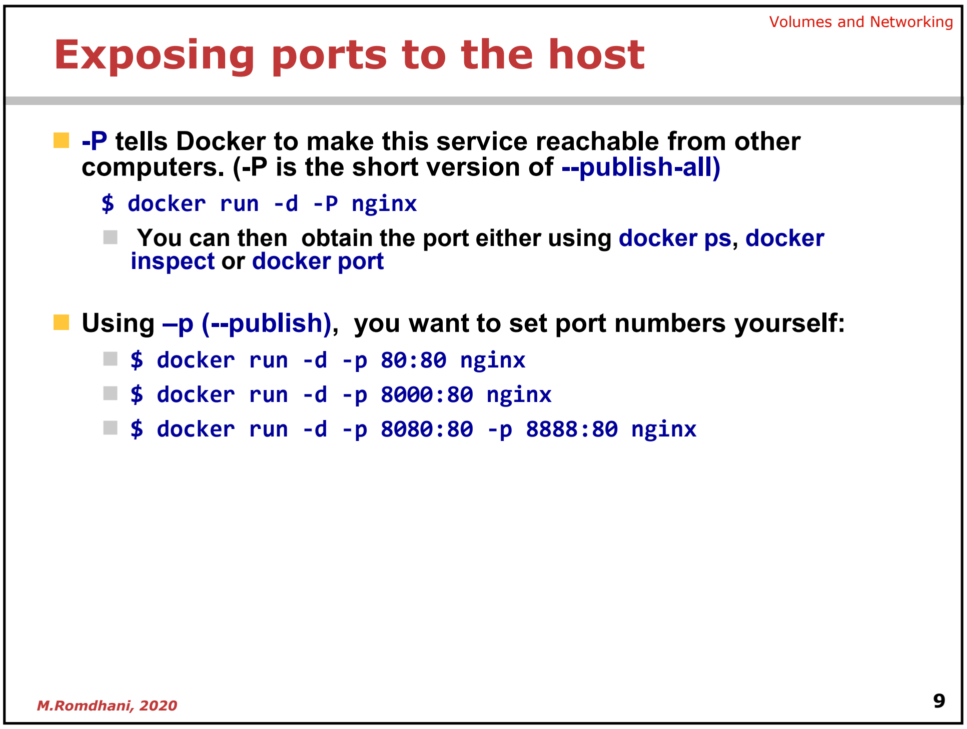
Volumes and Networking

Working with networks

- Creating/deleting a network
  - `$ docker network create dev -driver=bridge`
  - `$ docker network delete dev`
- Inspecting networks
  - `$ docker network inspect dev`
- Listing networks
  - `$ docker network create dev`
- Placing containers on a network
  - `$ docker run -d --name es  --net dev elasticsearch:2`
- Connecting/disconnecting a network to/from a container
  - `docker network connect my_network my_container`
  - `docker network disconnect my_network my_container`

M.Romdhani, 2020 8

8



Exposing ports to the host

- **-P tells Docker to make this service reachable from other computers. (-P is the short version of --publish-all)**

```
$ docker run -d -P nginx
```

  - **You can then obtain the port either using docker ps, docker inspect or docker port**

- **Using –p (--publish), you want to set port numbers yourself:**
  - `$ docker run -d -p 80:80 nginx`
  - `$ docker run -d -p 8000:80 nginx`
  - `$ docker run -d -p 8080:80 -p 8888:80 nginx`