Docker Volume

Sang Shin
JPassion.com
"Code with Passion!"

Topics

- What is and Why volume (data volume)?
- Add a volume
- Mount host directory as a volume
- Create a volume
- Volume container
- Backup, restore, and migrate volumes

What is and Why Data Volume?

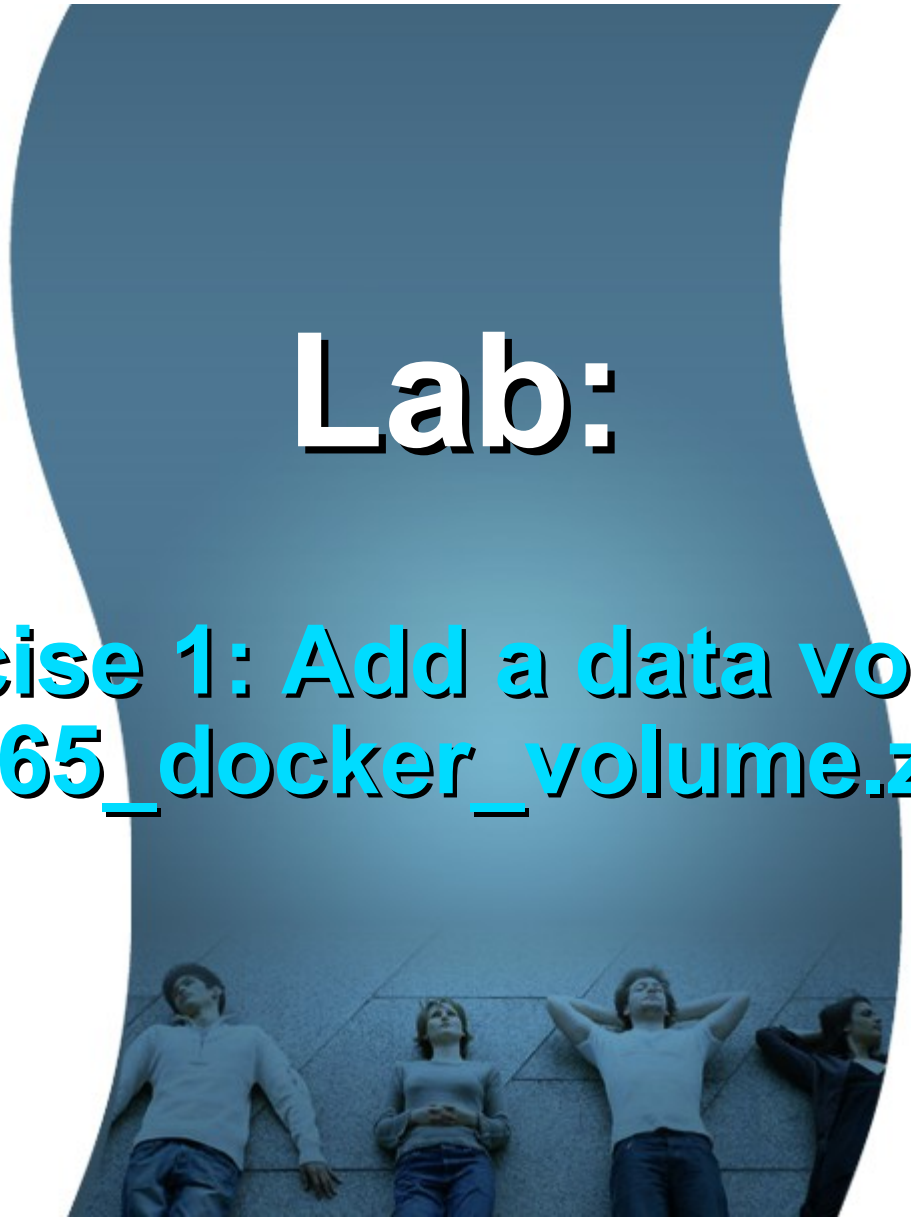
What is and Why Volume?

- Containers are ephemeral
  - Once a container is removed, it is gone along with data it created
- What about scenarios where you want the applications running inside the container to write to some files/data and then ensure that the data is still present even if the container is no longer runs or exists
  - You are running an application that is generating data and it creates files or writes to a database and so on. Now, even if the container is removed and in the future you launch another container, you would like that data to still be there
- Volumes are designed to persist data, independent of the container's lifecycle
  - Docker therefore never automatically deletes volumes when you remove a container, nor will it "garbage collect" volumes that are no longer referenced by a container

Add a volume

How to add a data volume?

- You can add a data volume to a container using the -v flag with the “docker create” and “docker run” command
  - You can use the -v multiple times to mount multiple data volumes
  - *docker run --name container1 -it -v /data -v /data2 busybox*
- You can also use the VOLUME instruction in a Dockerfile to add one or more new volumes to any container created from that image
  - The VOLUME command will mount a directory inside your container and store any files created or edited inside that directory on your hosts disk outside the container file structure, bypassing the union file system
  - The volume can be shared between your docker containers and they will stay around as long as there's a container (running or stopped) that references them



Lab:

Exercise 1: Add a data volume
1665_docker_volume.zip

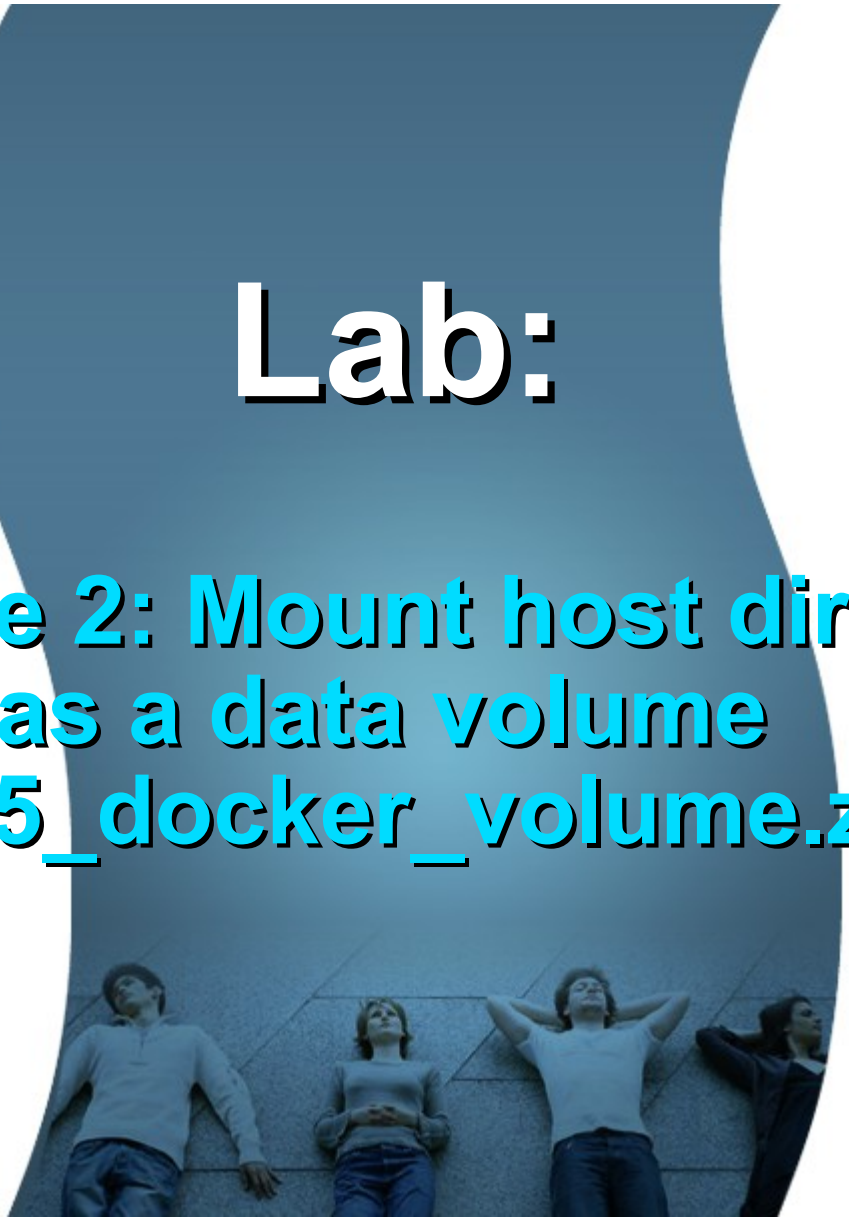
Mount a host directory as a data volume

Mount a host directory as a data volume

- In addition to creating a volume using the -v flag you can also mount a directory from your Docker engine's host into a container
- Use -v </host-directory>:</volume>
  - docker run -d -P --name web -v /src/webapp:/webapp training/webapp python app.py

Lab:

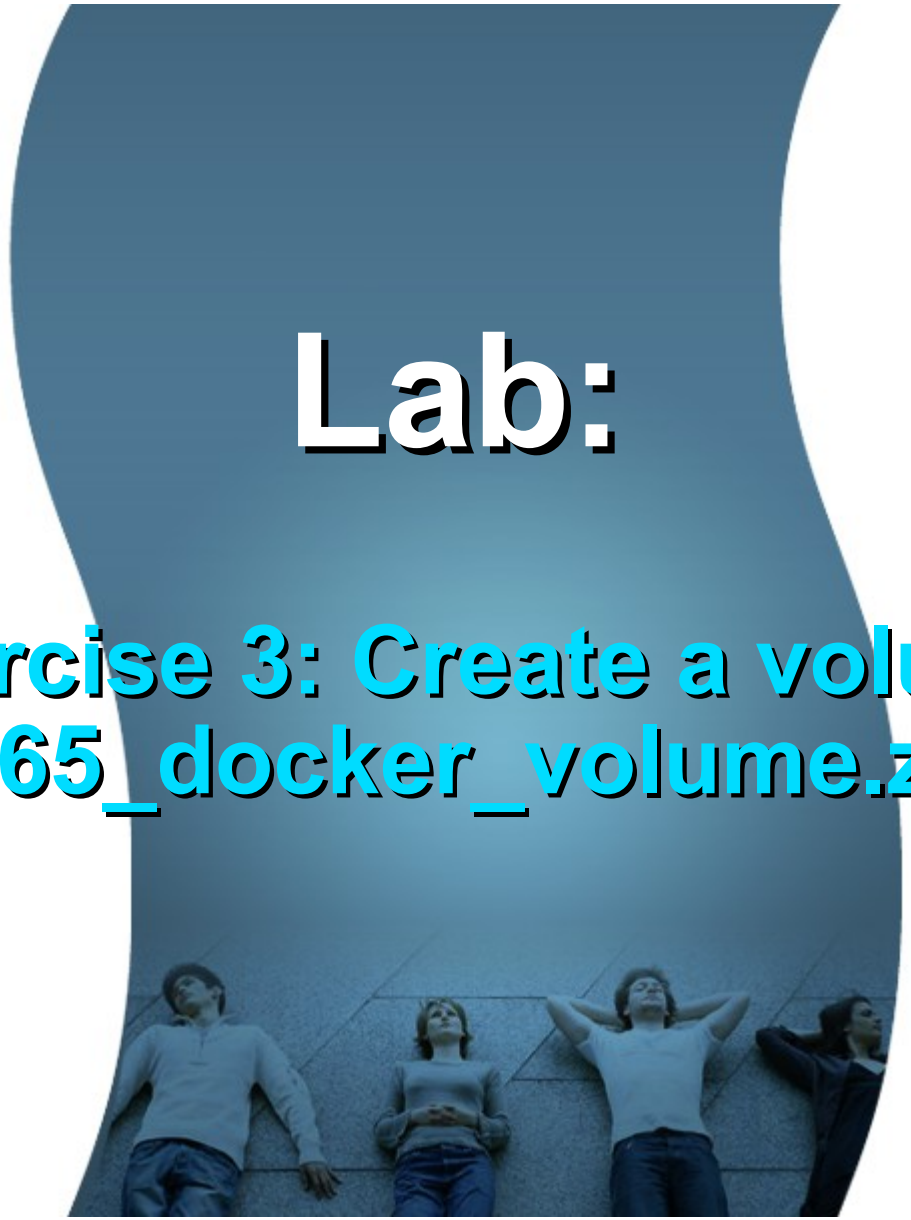
Exercise 2: Mount host directory as a data volume
1665_docker_volume.zip



Create data volume

Create a volume

- If you supply an absolute path for the /<host-directory>, Docker bind-mounts to the path you specify
- If you supply a name (name without forwarding /), Docker creates a named volume by that name
  - > docker run --name container3 -it -v my-own-volume:/data busybox



Lab:

Exercise 3: Create a volume
1665_docker_volume.zip

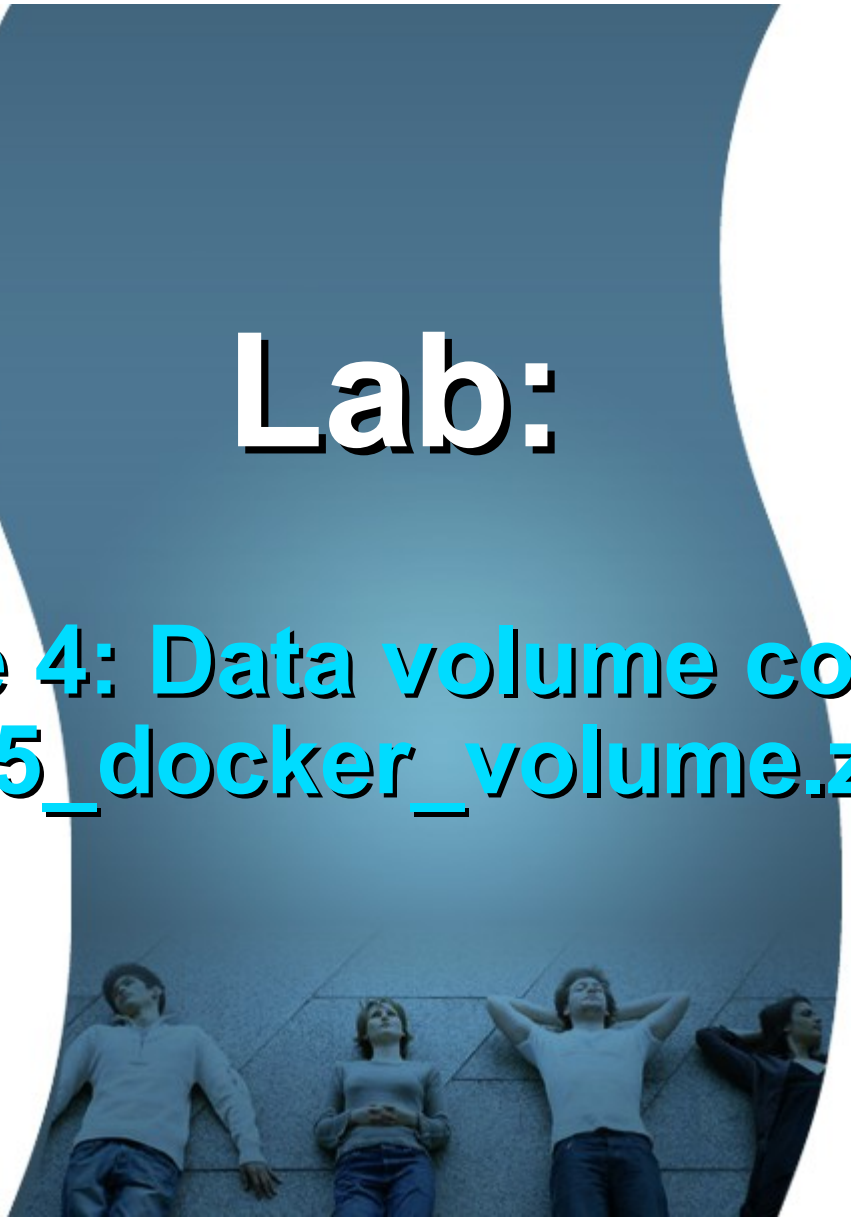
Data volume container

Data volume container

- If you have some persistent data that you want to share between containers, or want to use from non-persistent containers, you can create a named data volume container, and then to mount the data from it
  - Use “--volumes-from <data-volume-container>”
- Create a named data volume container
  - docker run --name master-container -it -v /master-data busybox
- Create non-persistent containers mount data from the data volume container
  - docker run --name slave-container1 -it --volumes-from master-container busybox

Lab:

Exercise 4: Data volume container 1665_docker_volume.zip



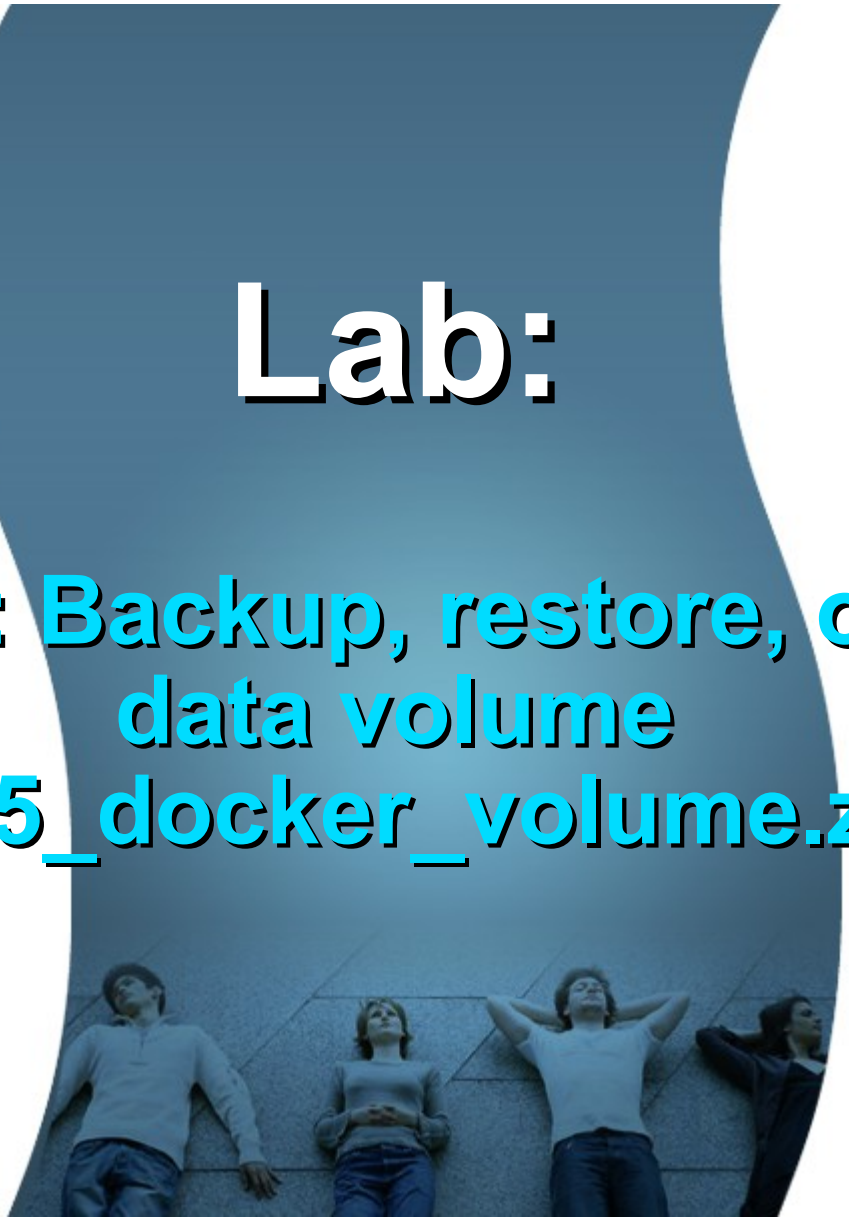
Backup, restore, or migrate data volumes

Backup, restore, or migrate data volumes

- Another useful function we can perform with volumes is use them for backups, restores or migrations

Lab:

Exercise 5: Backup, restore, or migrate data volume 1665_docker_volume.zip



Code with Passion!
JPassion.com