

# Docker JumpStart

## Mastering Container Volumes

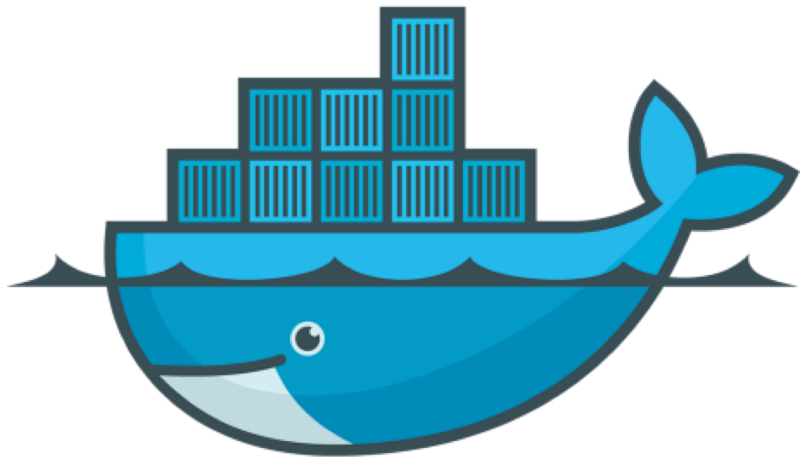
# Agenda

- Introduction to Volumes
- Creating a Volume
- Inspecting Volumes
- Defining a Volume in a Dockerfile
- Local Source Code and Containers



# Introduction to Volumes

## Docker Volumes



Special type of directory in a container typically referred to as a "data volume"

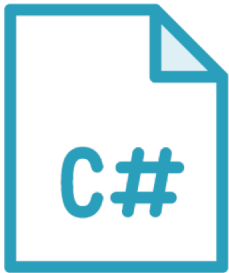
Can be shared and reused among containers

Updates to an image won't affect a data volume

Data volumes are persisted even after the container is deleted

# Volume Usage Scenarios

Link source code  
to container



Store database files  
outside of container

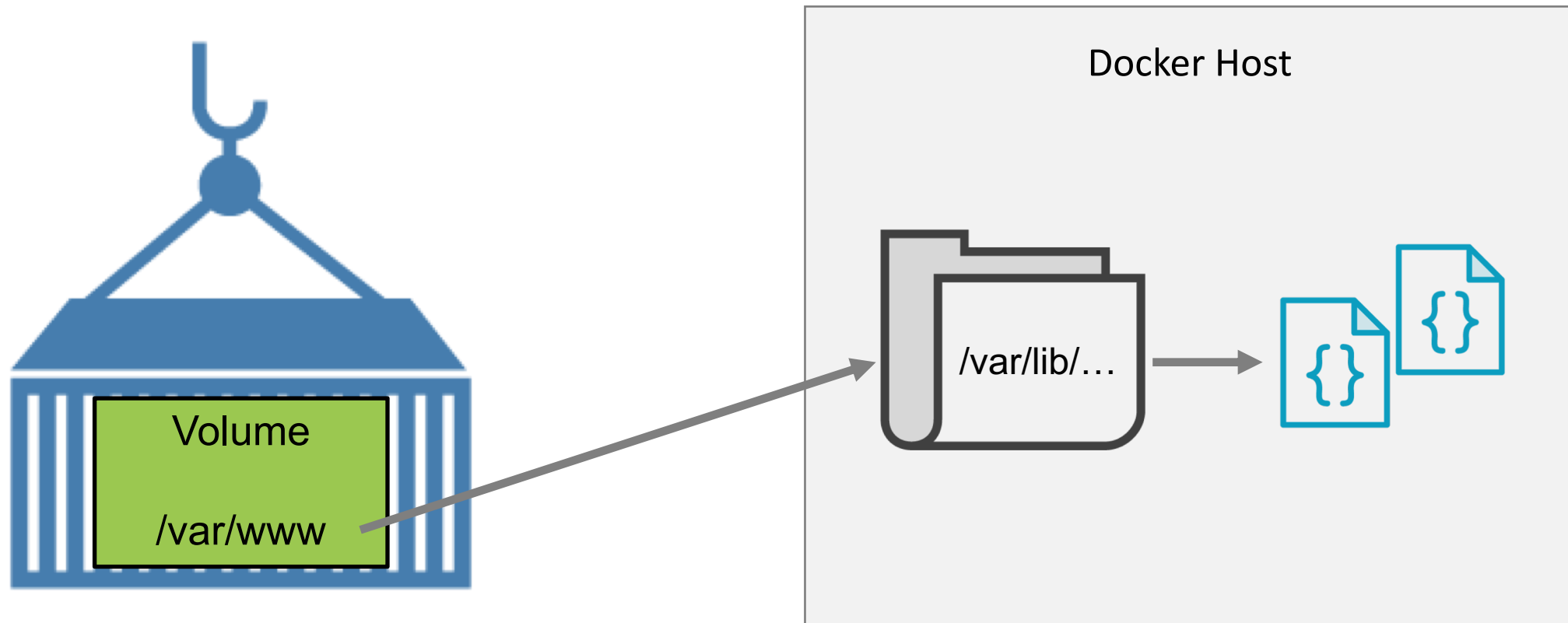


Store log files  
Outside of container



Many more scenarios!

# Volume Overview



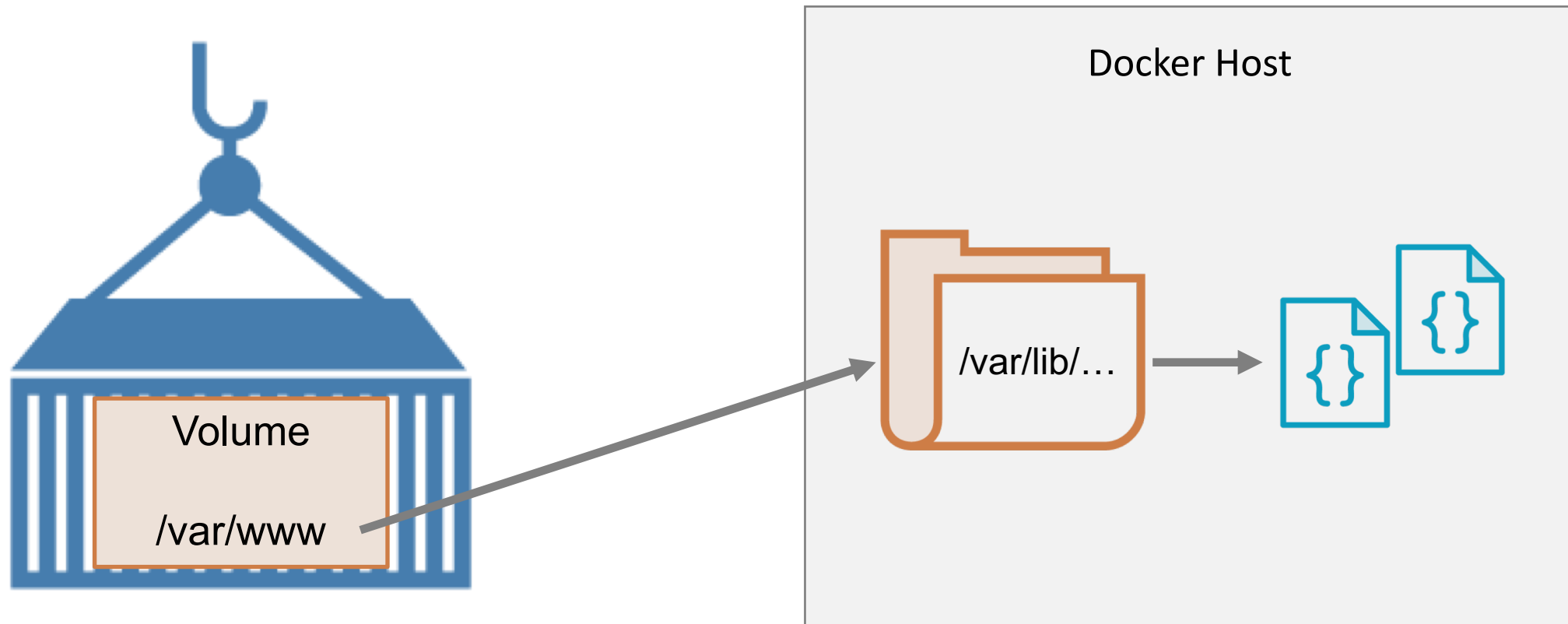
# Volumes and Mount Points

Docker volumes create a "mount point" in containers that point to externally mounted volumes on the host (or other containers)

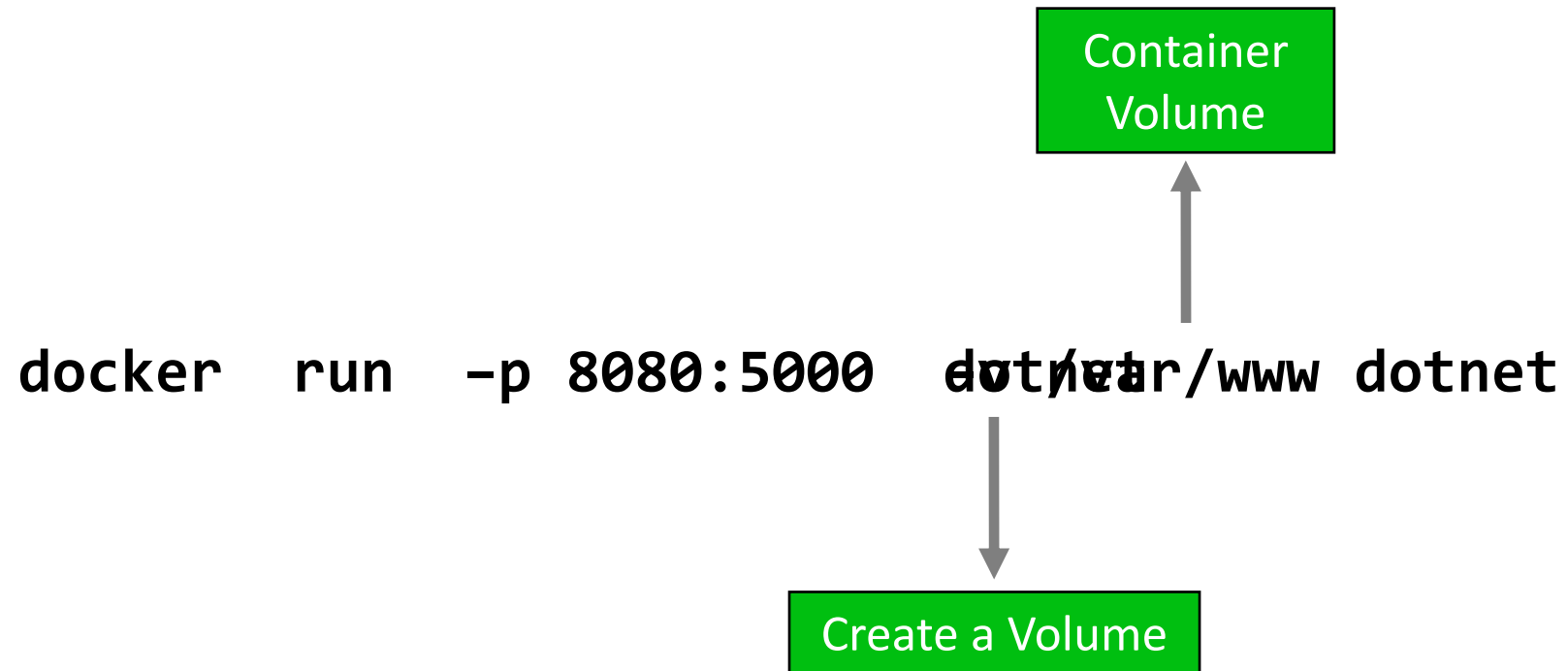
# Creating a Volume



# How Do You Create a Volume?



# Creating a Data Volume



# Inspecting Volumes

# Locating a Volume

`docker inspect mycontainer`

...

"Mounts": [

{

"Type": "volume",

"Source": "/src",

"Destination": "/var/www",

"RW": true

}

]

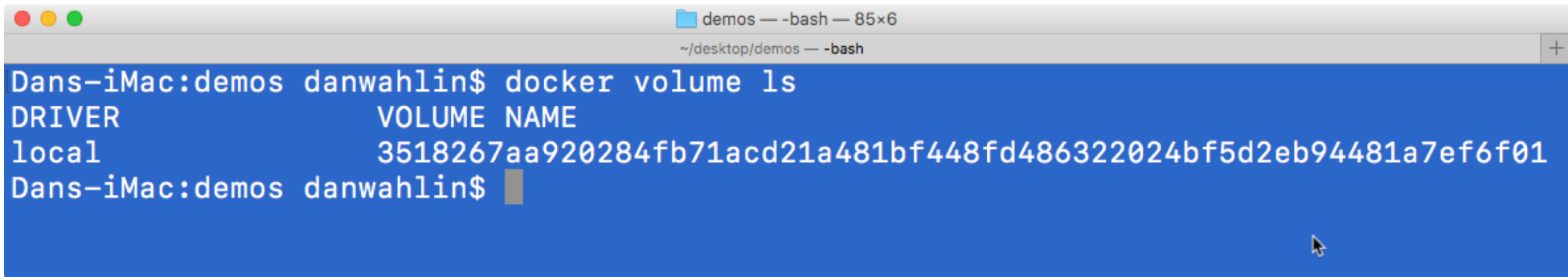
...

Host  
Location

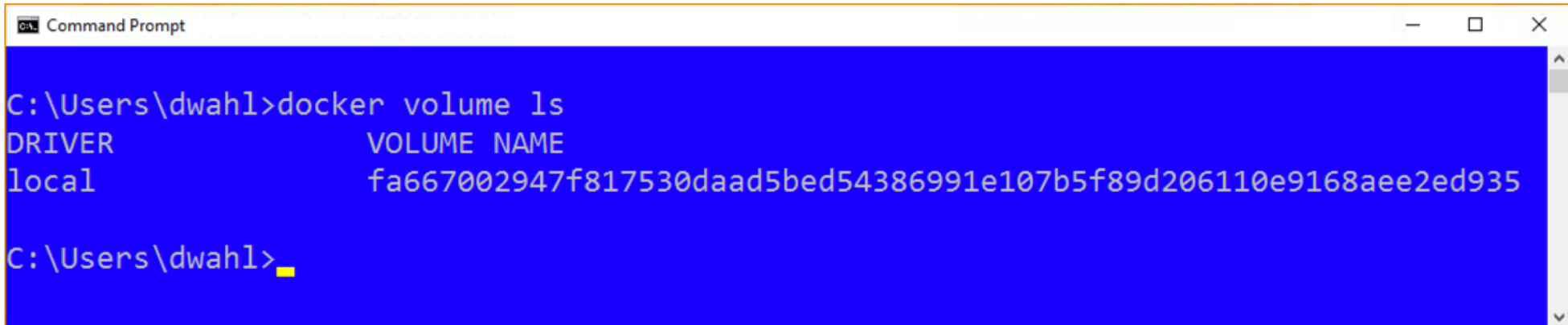
Volume Location  
in Container

# Listing Volumes

**docker volume ls**

A terminal window titled 'demos -- -bash -- 85x6' with a subtitle '~ /desktop/demos -- -bash'. The prompt is 'Dans-iMac:demos danwahlin\$'. The command 'docker volume ls' has been executed, showing a table with two columns: DRIVER and VOLUME NAME. The output shows a single volume named '3518267aa920284fb71acd21a481bf448fd486322024bf5d2eb94481a7ef6f01' on the 'local' driver.

```
Dans-iMac:demos danwahlin$ docker volume ls
DRIVER          VOLUME NAME
local           3518267aa920284fb71acd21a481bf448fd486322024bf5d2eb94481a7ef6f01
Dans-iMac:demos danwahlin$
```

A Windows Command Prompt window titled 'Command Prompt'. The prompt is 'C:\Users\dwahl>'. The command 'docker volume ls' has been executed, showing a table with two columns: DRIVER and VOLUME NAME. The output shows a single volume named 'fa667002947f817530daad5bed54386991e107b5f89d206110e9168aee2ed935' on the 'local' driver.

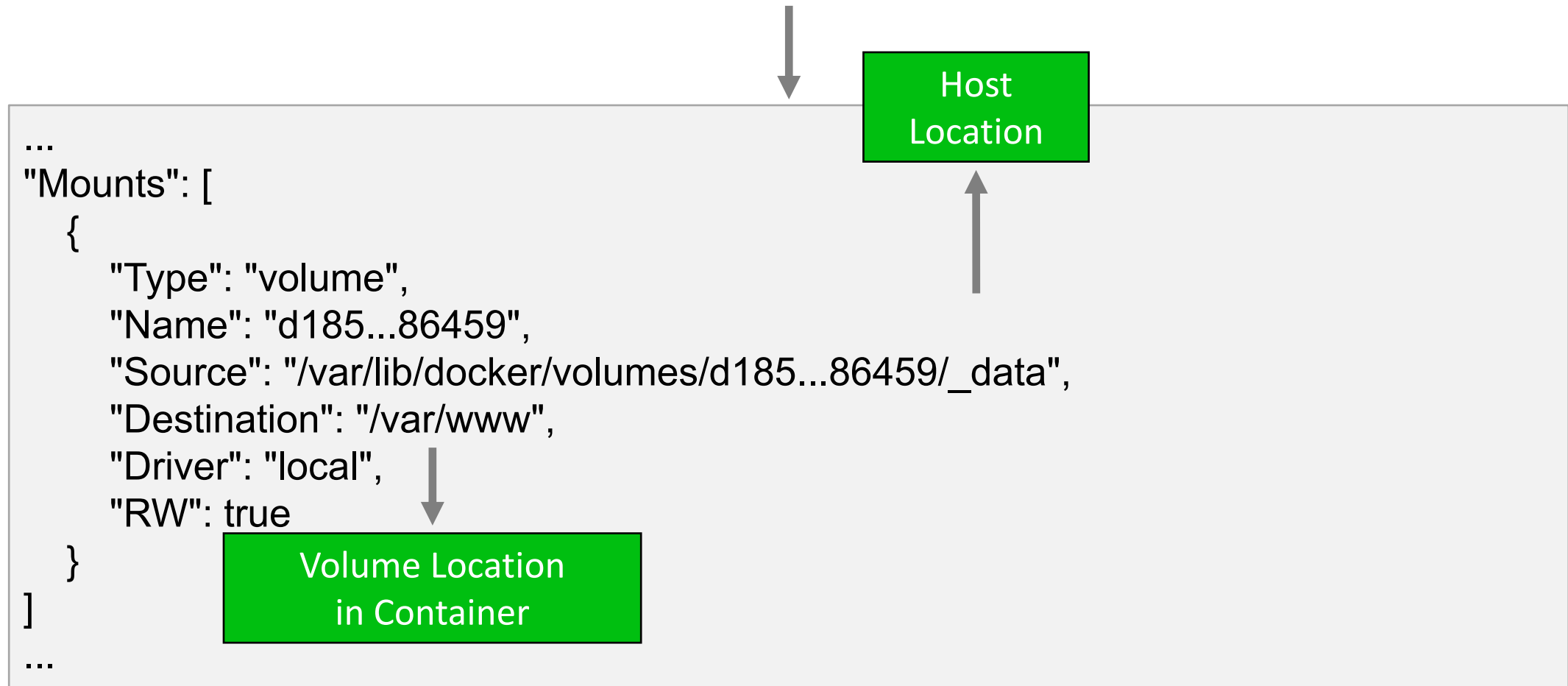
```
C:\Users\dwahl>docker volume ls
DRIVER          VOLUME NAME
local           fa667002947f817530daad5bed54386991e107b5f89d206110e9168aee2ed935
C:\Users\dwahl>
```

# Inspecting a Volume

```
docker volume inspect [volumeName]
```

# Inspecting a Container's Volumes

`docker inspect mycontainer`



# Removing Volumes

Remove container volumes (data loss!)



```
docker rm -v containerId
```

```
docker volume prune
```



Remove unused volumes



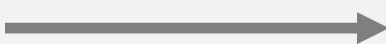
# Defining a Volume in a Dockerfile

# Defining Volumes

- Volumes can be defined multiple ways:
  - Using *docker run -v* when running a container (host mount can be defined)
  - In a docker-compose.yml file (covered later – host mount can be defined)
  - In a Dockerfile (host mount cannot be defined)
- Required volumes for data files, log files, etc. can be defined in a Dockerfile



# Dockerfile with a Volume

FROM	microsoft/dotnet:x.x.x-sdk
LABEL	author="Jimmy Docker"
ENV	ASPNETCORE_URLS=http://*:5000
VOLUME	/logs  Volume mounted on host
WORKDIR	/var/www/aspnetcoreapp
CMD	["/bin/bash", "-c", "dotnet restore && dotnet run"]

# Can a Host Mount be Defined in a Dockerfile?

FROM	microsoft/dotnet:x.x.x-sdk
LABEL	"Jane Docker"
ENV	ASPNETCORE_URLS=http://*:5000
VOLUME	/host-logs /container-logs →
WORKDIR	/var/www/aspnetcoreapp
CMD	["/bin/bash", "-c", "dotnet restore && dotnet run"]

Not supported! Host is always independent from the Dockerfile

# Changing the Volume Within a Dockerfile

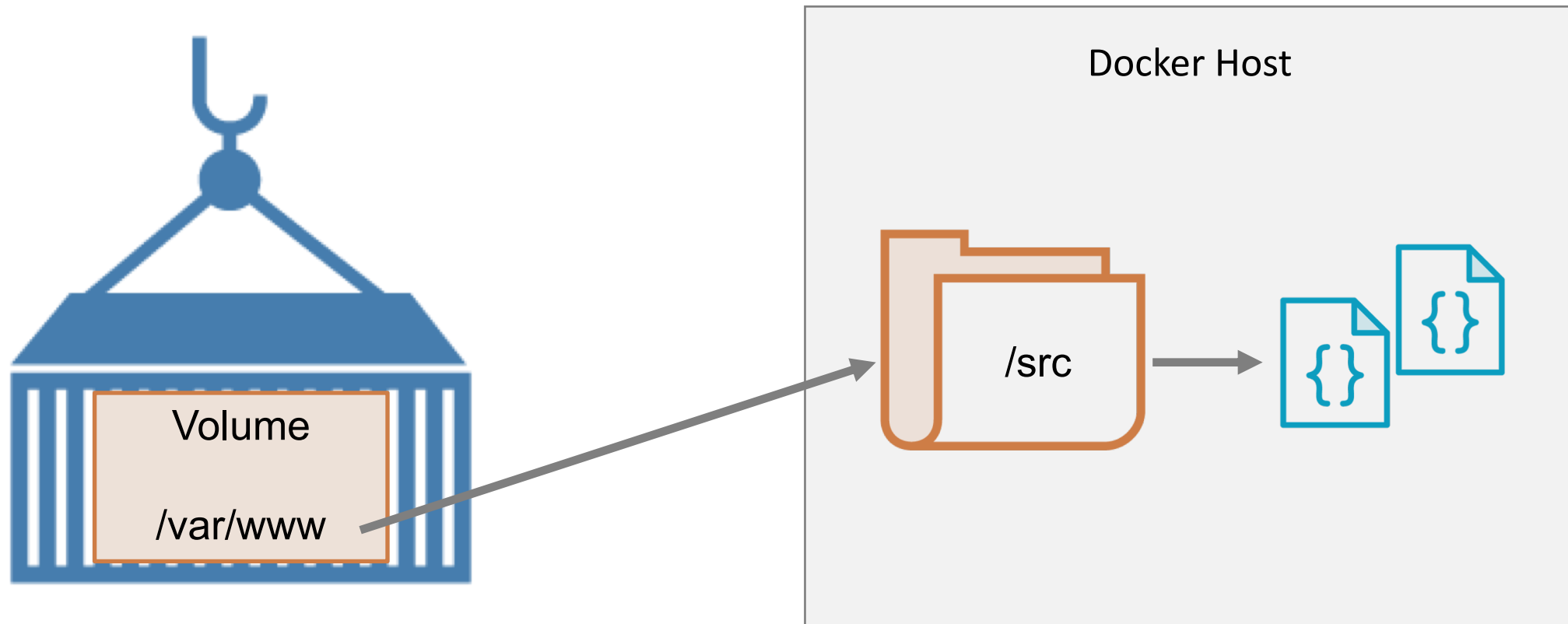
## **Important!**

If any build steps change the data within the volume after it has been declared, those changes will be discarded.

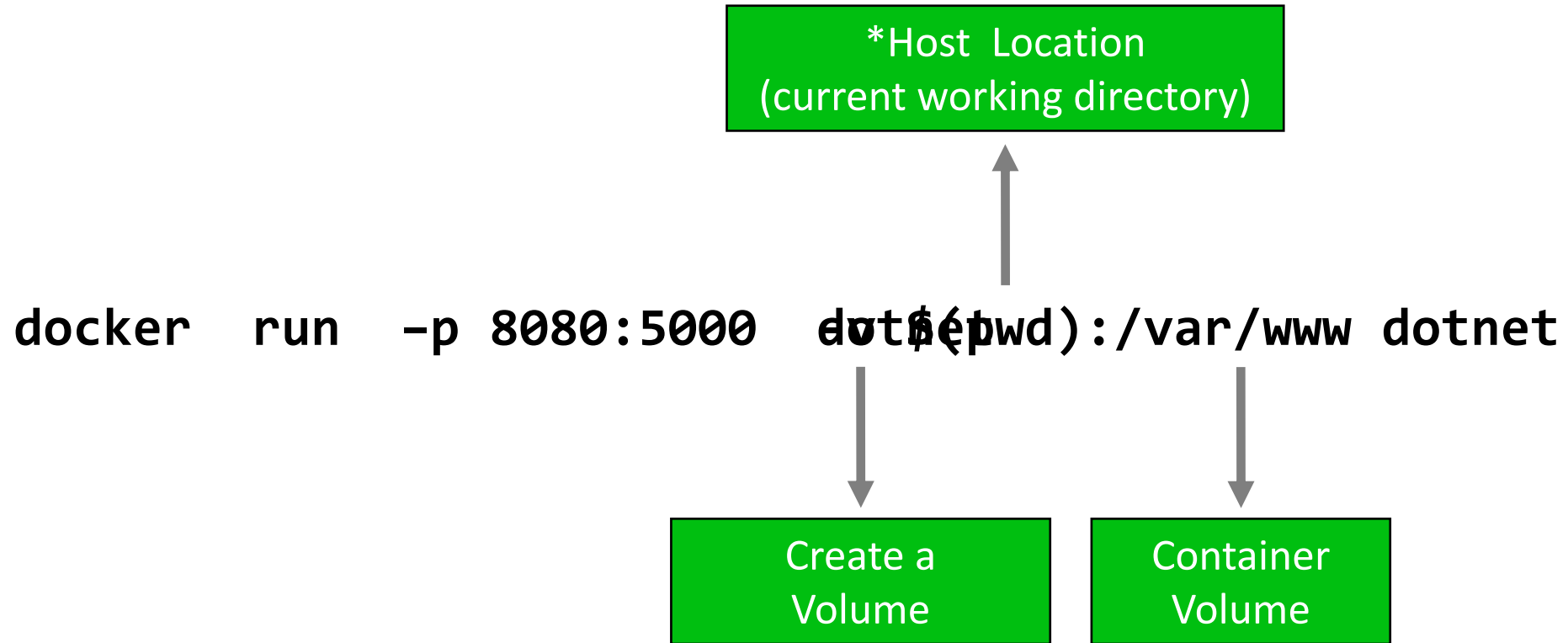
<https://docs.docker.com/engine/reference/builder/#volume>

# Local Source Code and Containers

# Binding a Volume to Your Source Code



# Customizing the Host Location



\* On Windows you'll need to use different syntax to reference the local folder:

<https://blog.codewithdan.com/docker-volumes-and-print-working-directory-pwd>



# Summary

- Volumes can "hook" a container folder to a host folder
- Source code can be hooked into to a running container using the -v switch
- Volumes can be listed and inspected using *docker volume*
- Volumes are persisted on the Docker host (even after a container is removed)



# Lab

## Running Containers and Using Volumes

<https://cloudskills.io/labs>



# Bonus: Windows Volume Syntax

**To create a volume that points to the folder that you're currently in:**

DOS (cmd):                -v %cd%:/data

Powershell:             -v \${PWD}:/data

Git Bash:                -v /\$(pwd)/data

**You can use an absolute path as well:**

-v /c/folder:/data