

Volumes and Networking

Juan Osorio
Premier Field Engineer
Apps



Overview

- Storage
- Volumes
- Networking
 - Linux
 - Windows

Storage and Networking



Filesystem in Containers

The screenshot displays two windows from a Windows operating system. The top window is an Administrator Windows PowerShell terminal. It shows a command being executed to run a PowerShell command inside a Docker container named 'my_container'. The command is 'docker exec my_container powershell ls C:/'. The output shows the directory 'C:\' and a list of files and folders with their attributes (Mode, LastWriteTime, Length, Name).

The bottom window is a File Explorer window showing the contents of the 'C:\' drive. The 'Users' folder is selected. The left sidebar shows 'This PC' and 'Network' options. The main pane shows a list of files and folders with columns for Name, Date modified, Type, and Size.

Mode	LastWriteTime	Length	Name
d----	9/11/2018 6:59 PM		inetpub
d-r--	9/9/2018 10:21 AM		Program Files
d----	9/9/2018 10:14 AM		Program Files (x86)
d----	9/11/2018 7:03 PM		RoslynCompilers
d-r--	9/11/2018 7:00 PM		Users
d----	9/11/2018 6:59 PM		Windows
-a----	4/12/2018 3:27 AM	1894	License.txt
-a----	9/11/2018 7:01 PM	171712	ServiceMonitor.exe

Name	Date modified	Type	Size
inetpub	9/11/2018 6:59 PM	File folder	
Program Files	9/9/2018 10:21 AM	File folder	
Program Files (x86)	9/9/2018 10:14 AM	File folder	
RoslynCompilers	9/11/2018 7:03 PM	File folder	
Users	9/11/2018 7:00 PM	File folder	
Windows	9/11/2018 6:59 PM	File folder	
License.txt	4/12/2018 3:27 AM	Text Document	2 KB
ServiceMonitor.exe	9/11/2018 7:01 PM	Application	168 KB

Filesystem in Containers

- Containers get temporary storage
- Throw away storage when container goes away

```
[sujit@labs ~]$ sudo docker exec -it 63 bash
root@63e6d68fe47c:/#
root@63e6d68fe47c:/# ls -l
total 8
drwxr-xr-x.  2 root root 4096 Feb  4 00:00 bin
drwxr-xr-x.  2 root root    6 Jan 22 13:47 boot
drwxr-xr-x.  5 root root  340 Feb 14 04:54 dev
drwxr-xr-x.  1 root root   66 Feb 14 04:54 etc
drwxr-xr-x.  2 root root    6 Jan 22 13:47 home
drwxr-xr-x.  1 root root   45 Feb  4 00:00 lib
drwxr-xr-x.  2 root root   34 Feb  4 00:00 lib64
drwxr-xr-x.  2 root root    6 Feb  4 00:00 media
drwxr-xr-x.  2 root root    6 Feb  4 00:00 mnt
drwxr-xr-x.  2 root root    6 Feb  4 00:00 opt
dr-xr-xr-x. 159 root root    0 Feb 14 04:54 proc
drwx-----.  2 root root   37 Feb  4 00:00 root
drwxr-xr-x.  1 root root   38 Feb 14 04:54 run
drwxr-xr-x.  2 root root 4096 Feb  4 00:00 sbin
drwxr-xr-x.  2 root root    6 Feb  4 00:00 srv
dr-xr-xr-x. 13 root root    0 Feb 13 23:43 sys
drwxrwxrwt.  1 root root    6 Feb  6 08:11 tmp
drwxr-xr-x.  1 root root   66 Feb  4 00:00 usr
drwxr-xr-x.  1 root root   19 Feb  4 00:00 var
root@63e6d68fe47c:/#
```

Storage

Container Image

- Not designed for persistent data

- Not designed for secrets

Volumes

- Enables storage persistence

- Enables mapping of storage into containers

- Read-Only or Read/Write

- Multiple containers on the same host can access the same location

- Plug-In Architecture

Network Storage

- Containers access SMB shares

- Accessed through the containers network

Storage Options

1. Volumes

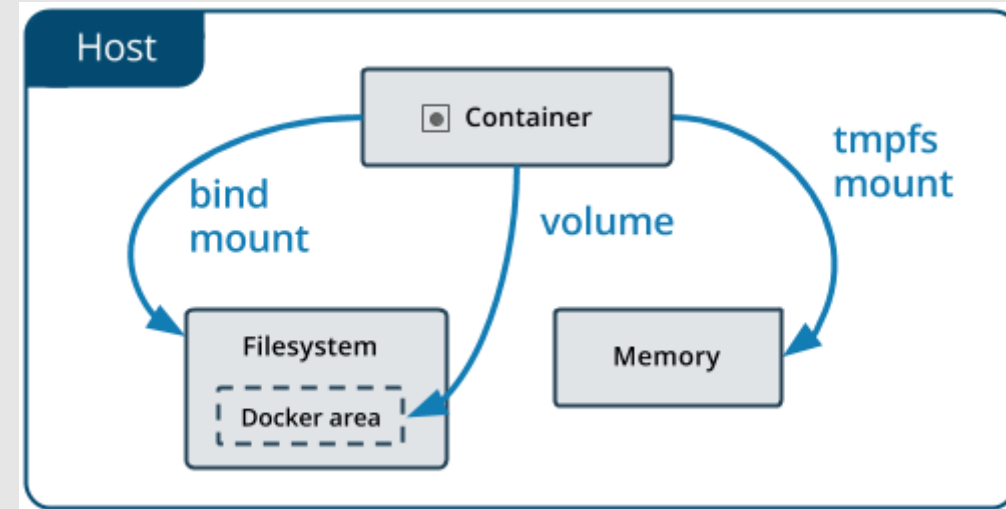
Stored in a part of the host filesystem which is managed by Docker

2. Bind Mounts

anywhere on the host system

3. Tmpfs mounts

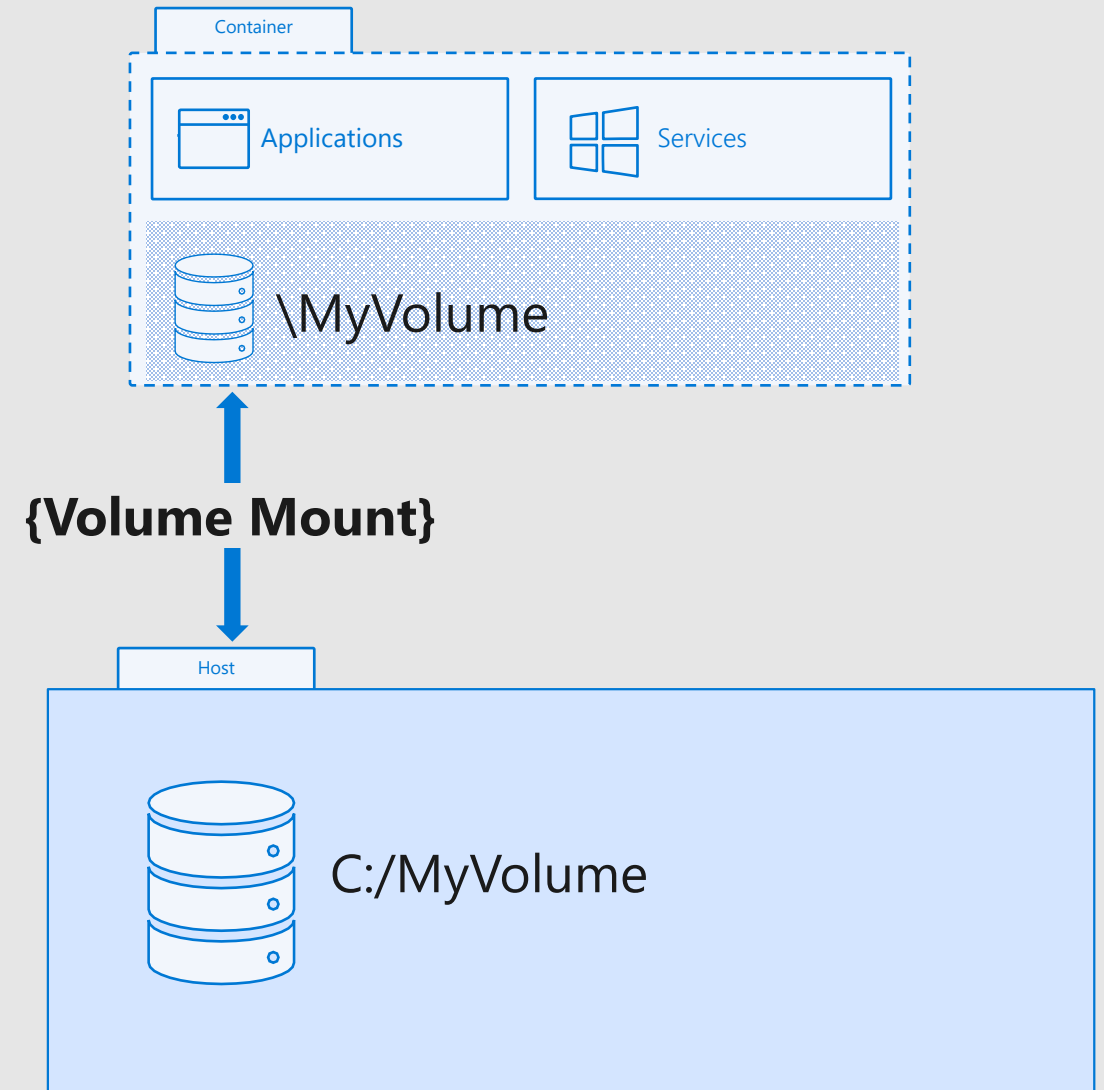
stored in the host system's memory only



<https://docs.docker.com/storage/>

Persistent Storage for containers

- You can mount folders or files from host to container.
- Changes preserved on host



Mount a host folder from a container

- To mount a container host folder to a container:
 - `docker run -v [host folder path]:[container folder path]`

Example:

```
docker run -it -v c:\containers_shared:c:\shared  
microsoft/windowsservercore cmd
```

Mount a host folder from a container

Docker inspect [container]

```
Administrator: Command Prompt - docker run -it -v C:\containers_shared:c:\shared microsoft/windowsservercore cmd
Volume in drive C has no label.
Volume Serial Number is 944B-A758

Directory of C:\
11/23/2016 06:45 AM          1,894 License.txt
07/16/2016 09:18 PM          <DIR> PerfLogs
01/04/2017 12:41 AM          <DIR> Program Files
07/16/2016 09:18 PM          <DIR> Program Files (x86)
01/04/2017 12:41 AM          <SYMLINKD> shared [\\?\ContainerMappedDirectories\F3415BE9-C71C-49A5-8BC6-0606670
01/04/2017 12:41 AM          <DIR> Users
01/04/2017 12:41 AM          <DIR> Windows
               1 File(s)          1,894 bytes
               6 Dir(s) 21,224,865,792 bytes free

C:\>dir c:\shared
Volume in drive C has no label.
Volume Serial Number is 944B-A758

Directory of c:\shared
01/04/2017 12:17 AM          <DIR> .
01/04/2017 12:17 AM          <DIR> ..
01/04/2017 12:17 AM          2,309 beatup-windows.ps1
01/03/2017 11:40 PM          2,172 Invoke-CPUStressTest.ps1
10/14/2016 07:55 AM       135,864 ServiceMonitor.exe
               3 File(s)        140,345 bytes
               2 Dir(s) 122,435,170,304 bytes free

C:\>
```

Inside the container

```
Administrator: Command Prompt

    "CpuPercent": 0,
    "IOMaximumIops": 0,
    "IOMaximumBandwidth": 0
  },
  "GraphDriver": {
    "Name": "windowsfilter",
    "Data": {
      "dir": "D:\\\\Docker\\\\windowsfilter\\\\c1fc2c27bcb6b
    }
  },
  "Mounts": [
    {
      "Type": "bind",
      "Source": "c:\\\\containers_shared",
      "Destination": "c:\\\\shared",
      "Mode": "",
      "RW": true,
      "Propagation": ""
    }
  ],
  "Config": {
    "Hostname": "c1fc2c27bcb6",
    "Domainname": "",
    "User": "",
    "AttachStdin": true,
    "AttachStdout": true,
```

Volume

- Create a volume

```
docker volume create <volume name>
```

- Mount the volume

```
docker run -it --mount source=<volume name>,target=<container path>  
mcr.microsoft.com/windows/nanoserver:1909 cmd
```

Tmpfs

- Only for Docker on Linux
- can't share tmpfs mounts between containers
- This is useful to temporarily store sensitive files that you don't want to persist in either the host or the container writable layer.

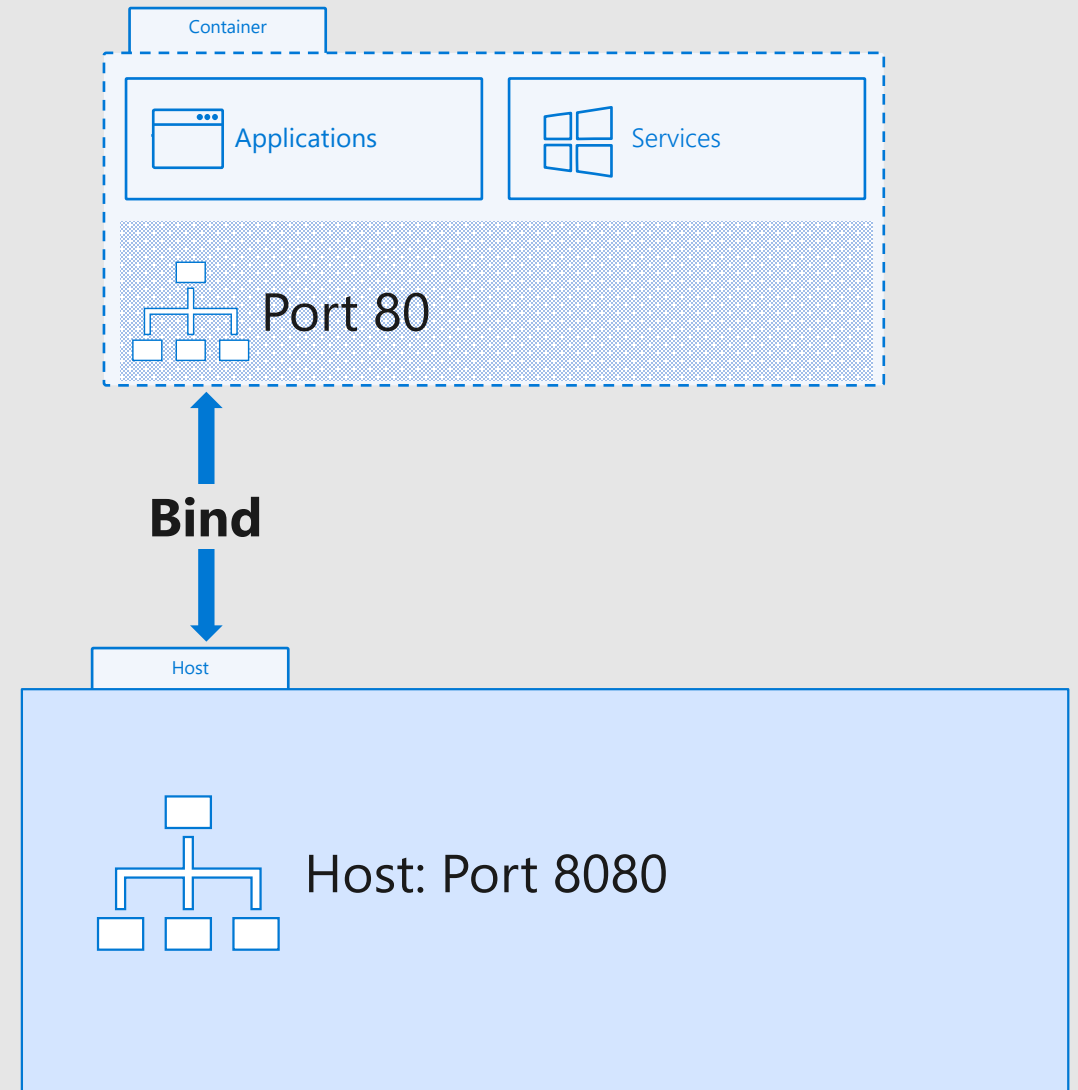
```
docker run -d -it --name container1 /  
    --mount type=tmpfs, destination=/app nginx:latest
```

Demo

- Bind mount
- Volume

Container Networking

- Each container gets their own network isolation
- Bind container to host ports.



Linux Networking

- **User-defined bridge networks** are best when you need multiple containers to communicate on the same Docker host.
- **Host networks** are best when the network stack should not be isolated from the Docker host, but you want other aspects of the container to be isolated.
- **Overlay networks** are best when you need containers running on different Docker hosts to communicate, or when multiple applications work together using swarm services.
- **Macvlan networks** are best when you are migrating from a VM setup or need your containers to look like physical hosts on your network, each with a unique MAC address.
- **Third-party network plugins** allow you to integrate Docker with specialized network stacks.

Windows networking

- **NAT (Default)** Dynamic IP allocation and assignment by Host Networking Service (HNS) from internal NAT subnet prefix
- **Transparent** Static or dynamic (using external DHCP server) IP allocation and assignment from IP addresses within container host's network prefix
- **Overlay** Dynamic IP allocation from Docker Engine Swarm Mode managed prefixes and assignment through HNS
- **L2Bridge** Static IP allocation and assignment from IP addresses within container host's network prefix (could also be assigned through HNS)
- **L2Tunnel** Azure only - Dynamic IP allocation and assignment from plugin

User defined networks

- docker network create \
 --driver=nat \
 --subnet=172.28.0.0/16 \
 --ip-range=172.28.5.0/24 \
 --gateway=172.28.5.254 \
 mynetwork

Demo

- Networking

Knowledge Check

- Difference between bind mount and volume
- Parameters to mount folders in containers?
- What are default networks for Windows and Linux containers?



Thank you! Questions?