Playing with Containers

Marco Mambelli
Fermilab
6/6/2024

Container

Large metal box used for the transportation of freight by road, rail, sea, or air

https://dictionary.cambridge.org/dictionary/english/container

Container

Discrete environment set up within an operating system in which one or more applications may be run, typically assigned only those resources necessary for the application to function correctly.

Standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. (Docker 2013)

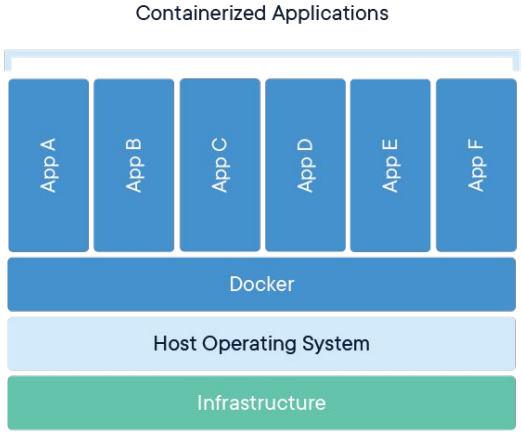
It's a Linux thing

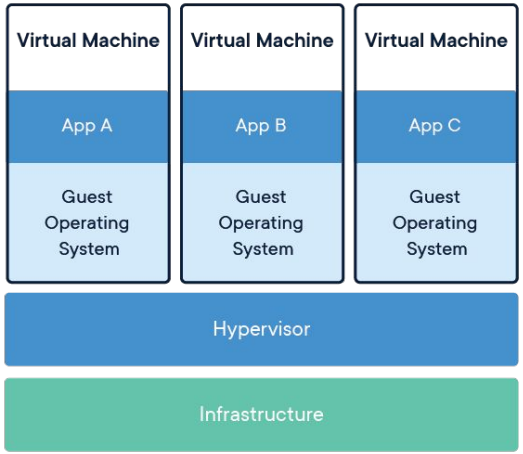
Shared kernel space

- Kernel of the host OS

Separate user space

- App + Library



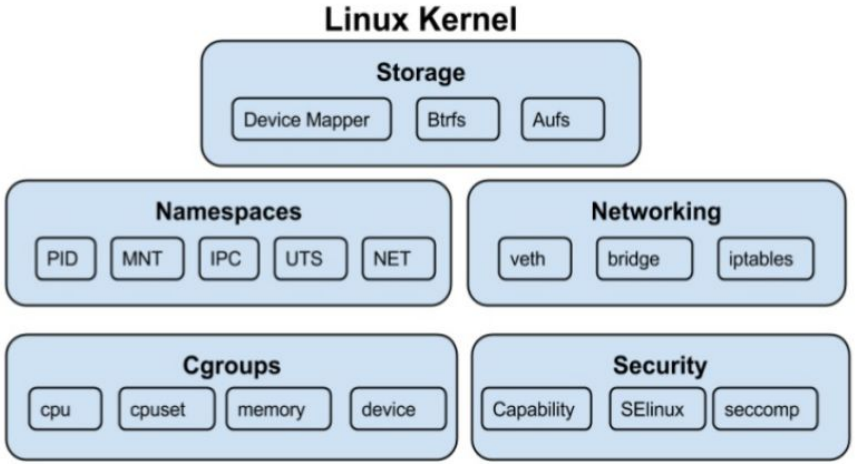


User mode	**User applications**	For example, bash, LibreOffice, Apache OpenOffice, Blender, 0 A.D., Mozilla Firefox, etc.				
	Low-level system components:	**System daemons**: *systemd, runit, loginд, networkd, soundd...*	**Windowing system**: *X11, Wayland, Mir, SurfaceFlinger (Android)*	**Other libraries:** *GTK+, Qt, EFL, SDL, SFML, FLTK, GNUstep*, etc.		**Graphics**: *Mesa 3D, AMD Catalyst, ...*
	C standard library	`open()`, `exec()`, `sbrk()`, `socket()`, `fopen()`, `calloc()`, ... (up to 2000 subroutines) *glibc* aims to be POSIX/SUS-compatible, *uClibc* targets embedded systems, *bionic* written for Android, etc.				
Kernel mode	**Linux kernel**	`stat`, `splice`, `dup`, `read`, `open`, `ioctl`, `write`, `mmap`, `close`, `exit`, etc. (about 380 system calls) The Linux kernel System Call Interface (SCI, aims to be POSIX/SUS-compatible)				
		Process scheduling subsystem	IPC subsystem	Memory management subsystem	Virtual files subsystem	Network subsystem
		Other components: ALSA, DRI, evdev, LVM, device mapper, Linux Network Scheduler, Netfilter Linux Security Modules: *SELinux, TOMOYO, AppArmor, Smack*				
Hardware (CPU, main memory, data storage devices, etc.)						

Use Linux VM on other OSes

https://www.docker.com/resources/what-container/
https://i.stack.imgur.com/2mDPs.png
https://dockerlabs.collabnix.com/beginners/difference-vm-containers.html

Container abstraction

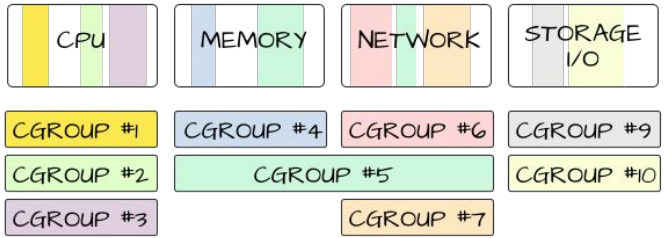


Isolation with Namespaces
(2002, EW Biederman, P Emelyanov, A Viro, and C Gorcunov)

- PID namespace for process isolation.
- NET namespace for managing network interfaces.
- IPC namespace for managing access to IPC resources.
- MNT namespace for managing filesystem mount points.
- UTS namespace for isolating kernel and version identifiers.

Resource Limitation with cgroup (2008, P Menage and R Seth)

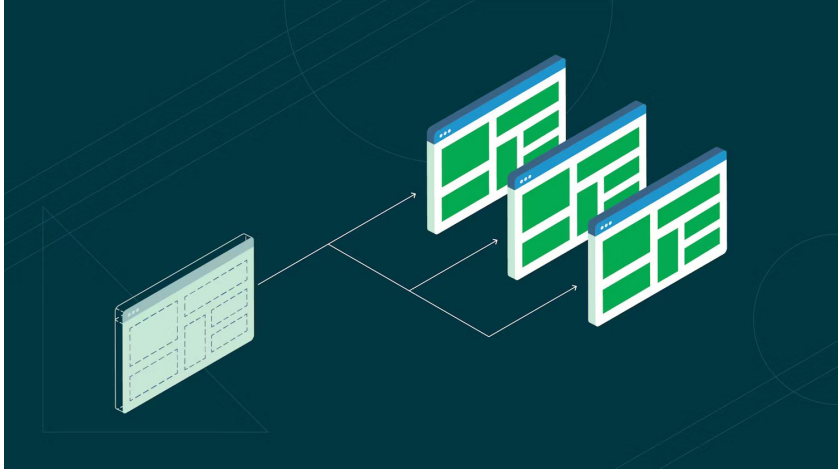
Packaging files and dependencies with rootfs

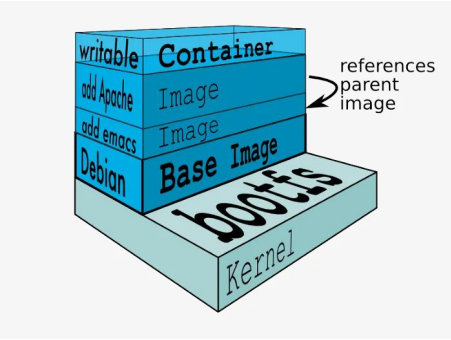


https://q15928.github.io/2021/01/09/container-101/
https://mairin.wordpress.com/2011/05/13/ideas-for-a-cgroups-ui/
https://en.wikipedia.org/wiki/Linux_kernel_interfaces
https://gcore.com/learning/containers-vs-virtual-machines/

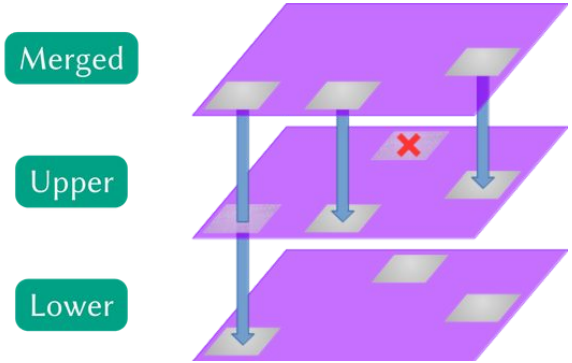
Container Image

Read only template used to create containers





Overlays and underlays (UnionFS/aufs)



https://circleci.com/blog/docker-image-vs-container/
https://velog.velcdn.com/images/koo8624/post/3e431335-53b5-4f0e-90bd-eb85b6c3c4fa/ufs.jpeg
https://www.nemunai.re/post/unveiling-whiteout-files/
https://embeddedcomputing.com/technology/processing/understand-what-an-overlayfs-is-and-how-it-works