Containers

What is Container?!

It feels like a VM

But it's not a VM

VM?

There are different kinds of virtual machines, each with different functions:

- **System virtual machines** (also termed full virtualization VMs) provide a substitute for a real machine. They provide functionality needed to execute entire operating systems. A hypervisor uses native execution to share and manage hardware, allowing for multiple environments which are isolated from one another, yet exist on the same physical machine. Modern hypervisors use hardware-assisted virtualization, virtualization-specific hardware, primarily from the host CPUs.
- **Process virtual machines** are designed to execute computer programs in a platform-independent environment.

https://en.wikipedia.org/wiki/Virtual_machine

Shared Kernel?

Yes, But ...

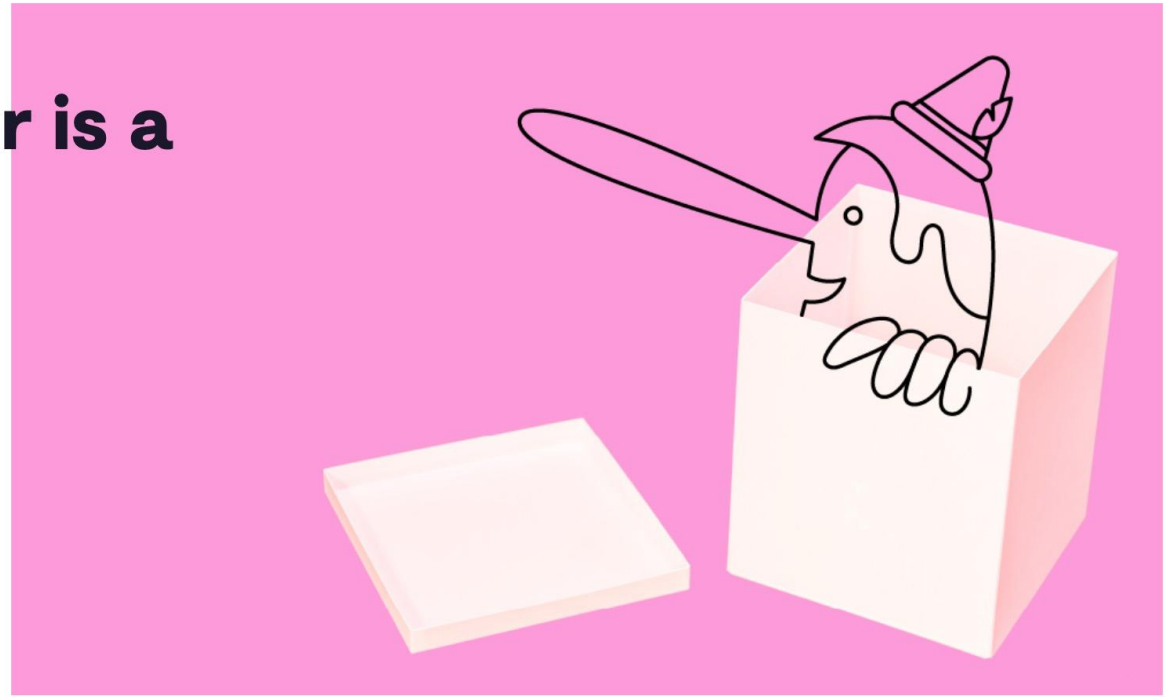
The Container is a lie!

« Blog

The container is a lie!

Larry Garfield
Director of Developer Experience

24 Mar 2020



https://platform.sh/blog/2020/the-container-is-a-lie/

Cgroup

Cgroups was originally written by Paul Menage and Rohit Seth, and mainlined into the Linux kernel in 2007 (2.6.24). Afterwards this is called cgroups version 1.

Resource limiting
: groups can be set to not exceed a configured memory limit, which also includes the file system cache

Prioritization
: some groups may get a larger share of CPU utilization or disk I/O throughput

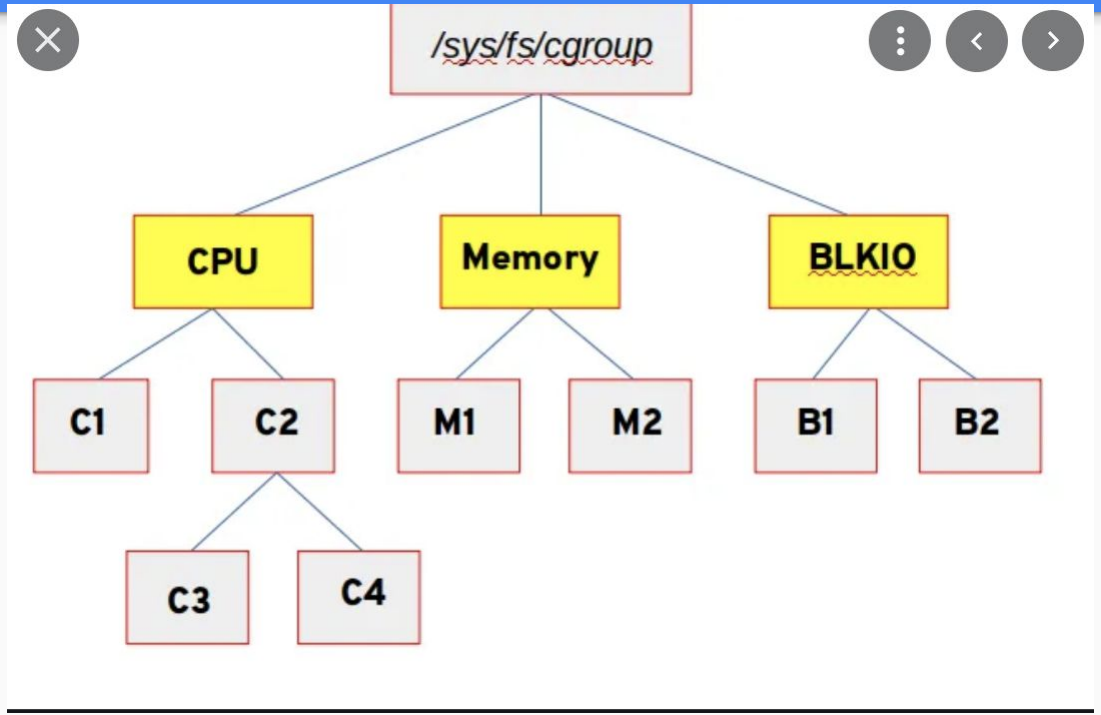
Accounting
: measures a group's resource usage, which may be used, for example, for billing purposes

Control
: freezing groups of processes, their checkpointing and restarting

https://en.wikipedia.org/wiki/Cgroups

Cgroup Demo

/sys/fs/cgroup



Namespaces

/proc/$pid/

How to start container?

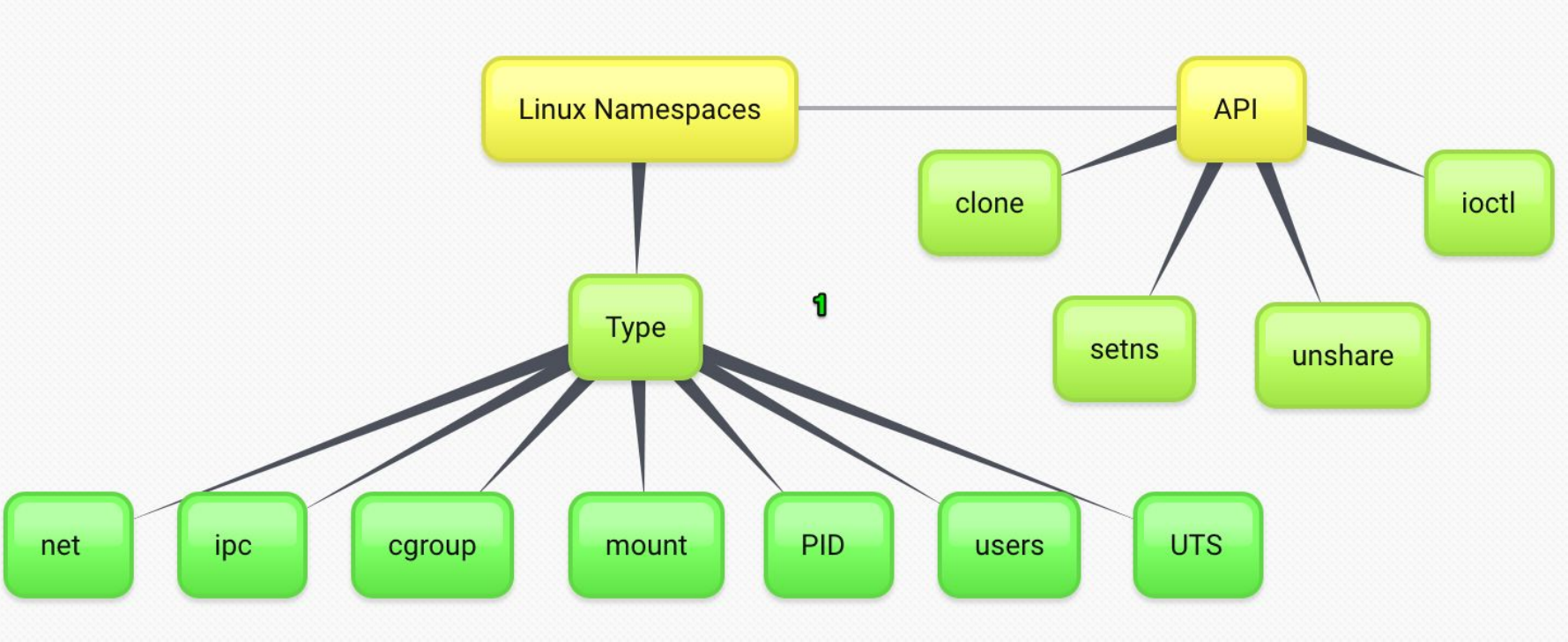
Linux Namespaces

A namespace wraps a global system resource in an abstraction that makes it appear to the processes within the namespace that they have their own isolated instance of the global resource. Changes to the global resource are visible to other processes that are members of the namespace, but are invisible to other processes. Namespaces are a fundamental aspect of containers on Linux.

The Linux Namespaces originated in 2002 in the 2.4.19 kernel with work on the mount namespace kind. Additional namespaces were added beginning in 2006 and continuing into the future.

https://www.man7.org/linux/man-pages/man7/namespaces.7.html , https://en.wikipedia.org/wiki/Linux_namespaces

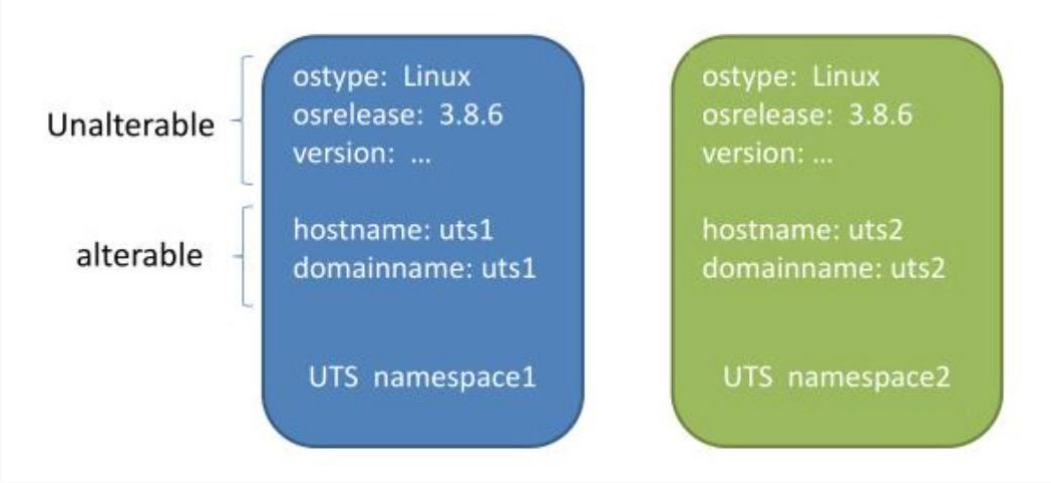
Linux Namespaces



https://www.man7.org/linux/man-pages/man7/namespaces.7.html

Linux Namespaces - UTS

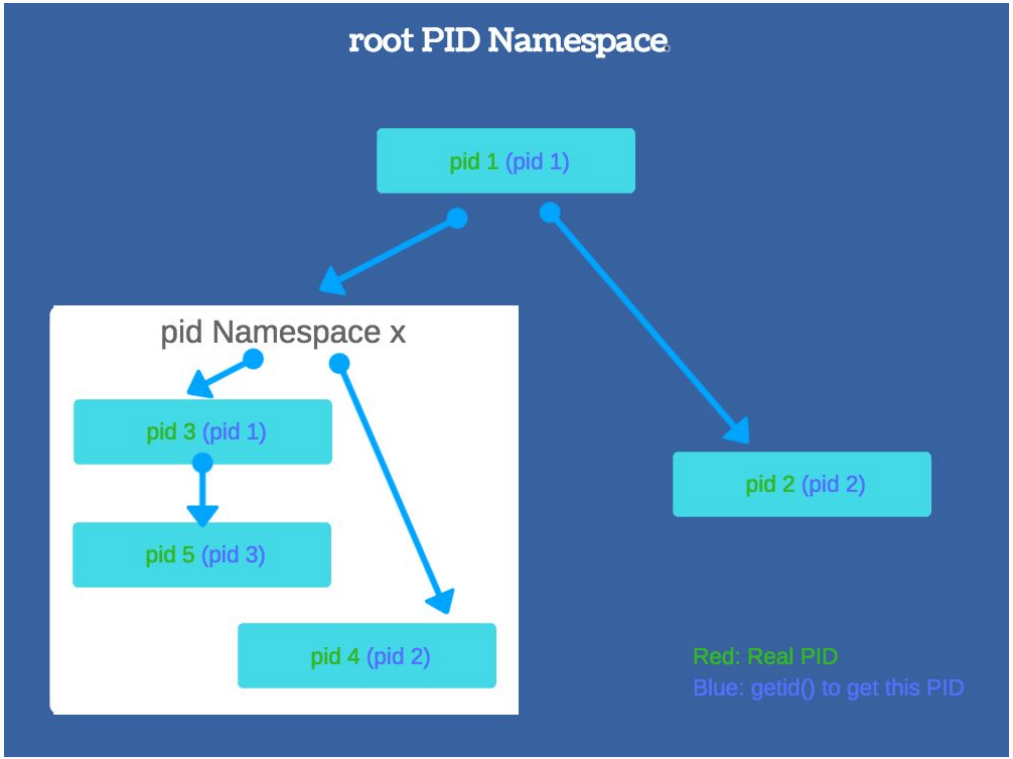
UTS namespaces provide isolation of two system identifiers: the hostname and the NIS domain name. These identifiers are set using sethostname(2) and setdomainname(2), and can be retrieved using uname(2), gethostname(2), and getdomainname(2). Changes made to these identifiers are visible to all other processes in the same UTS namespace, but are not visible to processes in other UTS namespaces.



https://www.man7.org/linux/man-pages/man7/uts_namespaces.7.html

Linux Namespaces - PID

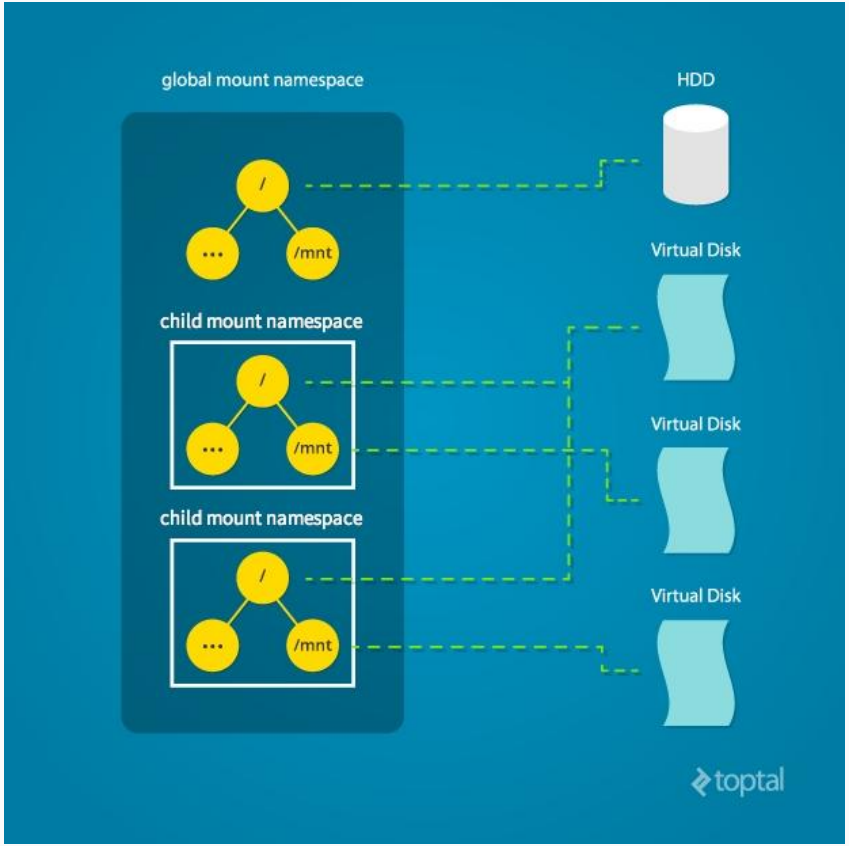
PID namespaces isolate the process ID number space, meaning that processes in different PID namespaces can have the same PID. PID namespaces allow containers to provide functionality such as suspending/resuming the set of processes in the container and migrating the container to a new host while the processes inside the container maintain the same PIDs.



https://www.man7.org/linux/man-pages/man7/pid_namespaces.7.html

Linux Namespaces - Mount

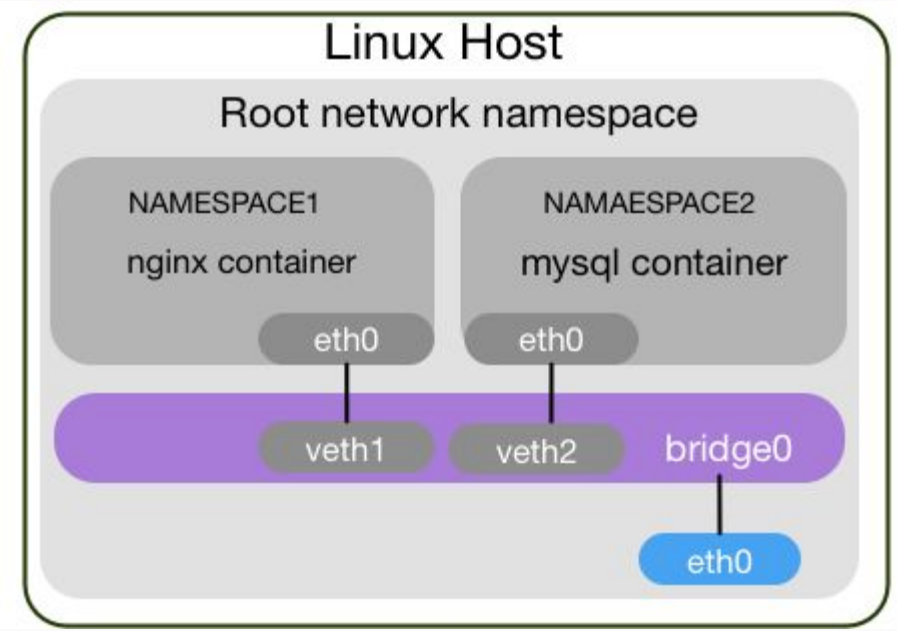
Mount namespaces provide isolation of the list of mount points seen by the processes in each namespace instance. Thus, the processes in each of the mount namespace instances will see distinct single-directory hierarchies.



https://www.man7.org/linux/man-pages/man7/mount_namespaces.7.html

Linux Namespaces - Network

Network namespaces provide isolation of the system resources associated with networking: network devices, IPv4 and IPv6 protocol stacks, IP routing tables, firewall rules, the /proc/net directory (which is a symbolic link to /proc/PID/net), the /sys/class/net directory, various files under /proc/sys/net, port numbers (sockets), and so on. In addition, network namespaces isolate the UNIX domain abstract socket namespace



https://man7.org/linux/man-pages/man7/network_namespaces.7.html, https://www.youtube.com/watch?v=6v_BDHIgOY8

Linux Namespaces - Others

- **Interprocess Communication (ipc):** IPC namespaces isolate processes from SysV style inter-process communication. This prevents processes in different IPC namespaces from using, for example, the SHM family of functions to establish a range of shared memory between the two processes. Instead each process will be able to use the same identifiers for a shared memory region and produce two such distinct regions.
- **User ID (user):** User namespaces are a feature to provide both privilege isolation and user identification segregation across multiple sets of processes available since kernel 3.8. A user namespace contains a mapping table converting user IDs from the container's point of view to the system's point of view. This allows, for example, the root user to have user id 0 in the container but is actually treated as user id 1,400,000 by the system for ownership checks. A similar table is used for group id mappings and ownership checks.
- **Control group (cgroup) Namespace:** The cgroup namespace type hides the identity of the control group of which process is a member. A process in such a namespace, checking which control group any process is part of, would see a path that is actually relative to the control group set at creation time, hiding its true control group position and identity. This namespace type has existed since March 2016 in Linux 4.6.
- **Time Namespace:** The time namespace allows processes to see different system times in a way similar to the UTS namespace. It was proposed in 2018 and landed on Linux 5.6, which was released in March 2020

https://en.wikipedia.org/wiki/Linux_namespaces#Namespace_kinds

What's next

- Running containers without docker (or runc, lxc,...) !
- CoW
- chroot, pivot_root
- SELinux
- Linux Caps

Thank You!

Any Questions?!