

Intro to Docker

Now with history!

Eric C. “echarlie” Landgraf

VTLUUG

January 31, 2019



Contents

History

Linux Namespaces

Demo

Limitations, features

Preface

Things I won't be addressing in this talk:

- ▶ Docker Compose (complex service definitions)
- ▶ Docker Swarm (clustering)
- ▶ Kubernetes (both of the above, with any OCI-compliant tools)
- ▶ Alternatives (RKT, podman, LXD)



Obligatory History

“Containers” also known as “operating-system-level virtualization” are a way to provide process-specific isolation with different views of user space.

- ▶ Chroot jails (1979/1982)
- ▶ Namespaces in Plan9 (1995)
- ▶ FreeBSD Jails (2000)
- ▶ Solaris Containers/Solaris Zones (before 2004)
- ▶ LXC and Docker (2008)



Chroot jails (1979/1982)

- ▶ chroot(2) implemented in BSD in '82 by Bill Joy, based on similar syscall introduced in Version 7 Unix
- ▶ Oldest form of isolation; first developed to test build/installation system for 4.2BSD
- ▶ Provides process visibility and access only to single tree in the filesystem
- ▶ Many Unix daemons chroot by default, or are easily configured to do so: e.g. almost every daemon in OpenBSD runs in a chroot



Plan9 Namespaces (1995)

- ▶ *Plan9 from Bell Labs* was an experimental operating system developed in the early '90s
- ▶ “[T]he foundations of the system are built on two ideas: a per-process name space and a simple message-oriented file system protocol.”
- ▶ Every application runs within a “namespace” which provides different view of the file system—IPC is facilitated by mounting a shared file in another process’ namespace.



FreeBSD Jails (2000)

- ▶ Extends chroot with the `jail(2)` syscall in FreeBSD 4.0 to prevent “second chroot” problem and provide network support.
- ▶ Provides more extensive sandboxing, and control over other process activities (disk use, processor limits, etc).



Solaris Zones (2004)

- ▶ Provides many of the same features as Jails, but with further extensions such as “Sparse Zones”—thin provisioning and overlay filesystems for zones
- ▶ Major feature is support for syscall translation in zones. Joyent *SmartOS* supports “lx” zones, which provide a binary-compatible Linux environment on a Solaris derivative.
- ▶ Zones allowed CPU and other resource assignment, which made Sun’s 4-core/32-thread T2 useful in 4-socket configurations.



LXC and Docker (2008)

- ▶ LXC was the first major tool to use cgroups (initially “process containers”) in Linux.
- ▶ cgroups and other Linux namespaces allowed LXC to provide functionality very similar to Zones or Jails using in-kernel building blocks



Linux namespaces

- ▶ Large portions of the kernel API are provided with namespaces
- ▶ Mount (filesystem), PIDs (procs), network, IPC, UTS (hostname), UIDs, and cgroups are all namespaces.
- ▶ Namespaced objects in a namespace (usually) cannot see those objects in other namespaces: i.e. multiple processes can be UID 1000 with different usernames if running in different UID namespaces.



More about Namespaces

- ▶ Notably, time is not namespaced in linux
- ▶ 3 syscalls: `clone(2)`, `unshare(2)`, `setns(2)`
- ▶ `unshare(1)` and `ip-netns(8)` can be used to manipulate namespaces from userspace without container infrastructure.



the part where I do a demo

This slide intentionally left blank.

Power of Docker

- ▶ Highly compatible—supported on Windows, OS X, and FreeBSD as well as Linux
- ▶ provides consistent and “portable” runtime for apps by bundling the whole runtime
- ▶ extensive support for most namespace functionality including cgroups (although rarely used).



Limitations and complaints

- ▶ dockerd/containerd runtime runs as root (solved by RedHat with their podman tool)
- ▶ docker group is root-equivalent
- ▶ Incomplete support for OCI (Open Container Initiative) standards
- ▶ Moving target—major interfaces change from version to version, and lots of backend details are only supported on “modern” kernels (e.g. overlays).
- ▶ Insecurity of multitenancy (c.f. gvisor, firecracker, Spectre/Meltdown)

