



Container Deep Dive: Namespaces, Layers, and cgroups

Photo by [nikko osaka](#) on [Unsplash](#)

Daniel Mikusa

- Staff Software Engineer @ 7SIGNAL, Inc
- Paketo Steering Committee Member
- Cloud-Native Buildpacks Maintainer

Contact Me

- dan@mikusa.com
- <https://github.com/dmikusa>
- <https://www.mikusa.com>



Slides



<https://github.com/dmikusa/container-deep-dive-slides>

Feedback



<https://confoo.ca/f/9EA9EB26BB7244CB9B49199F19859999>

Why are we here today?

Learn & Understand Containers



Also, because ...

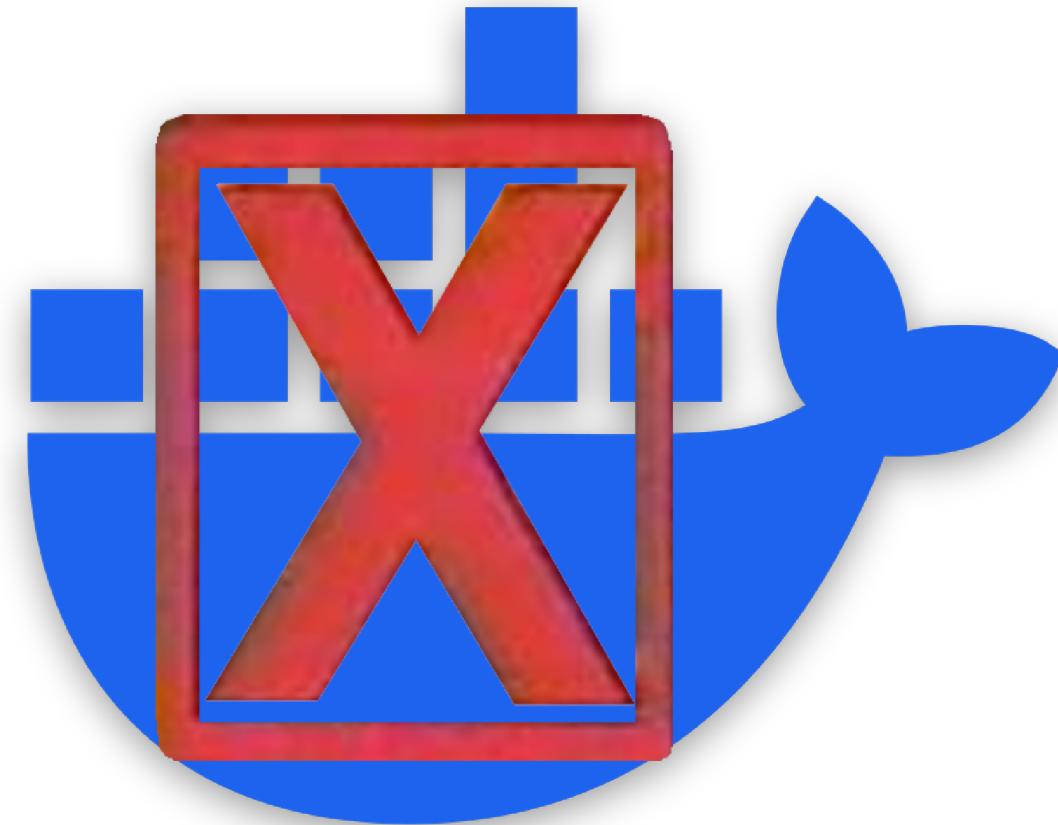


quickmeme.com

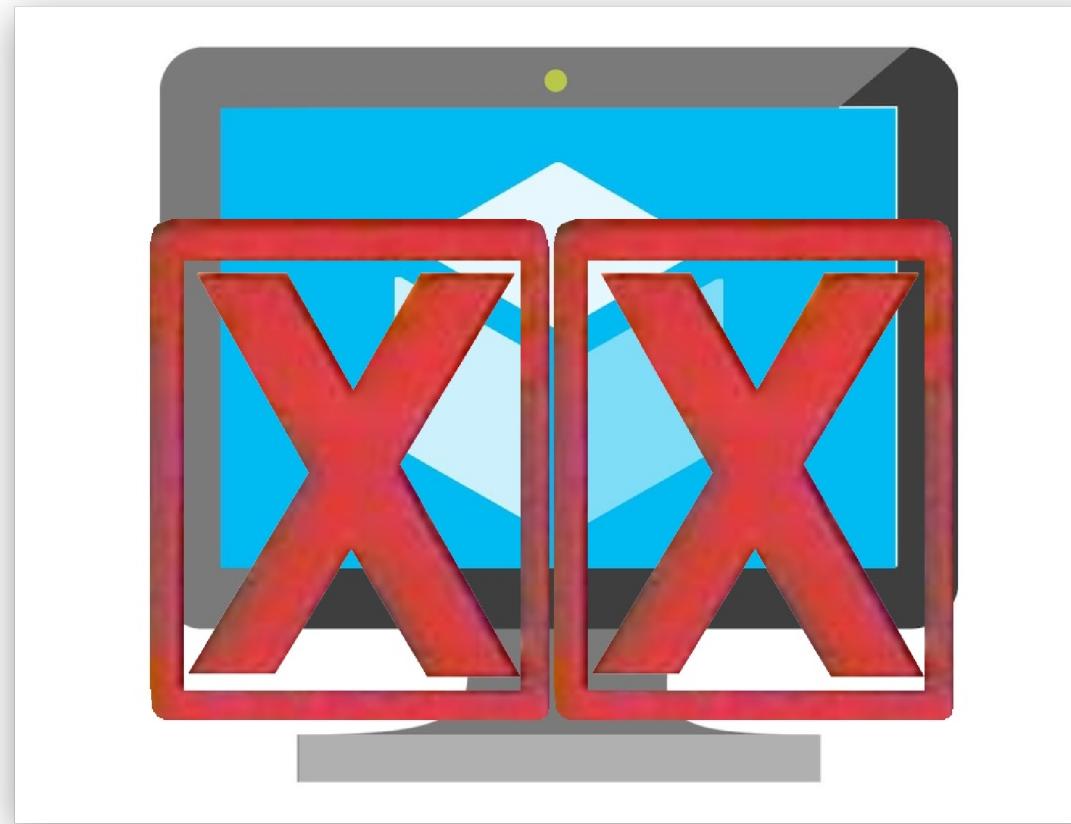
What is a container?



What's **NOT** a container?



What's **NOT** a container?



So What is a container then?

TL;DR - It's an isolated process.

Container Strengths & Weaknesses

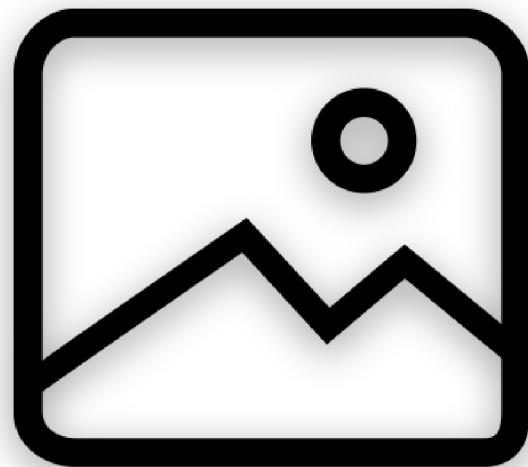
Strengths

- Cheap/Fast
- Easy way to package code
- Ubiquitous & battle tested
- Strong tooling & libraries
- Standardized (OCI)
- Efficient distribution protocol

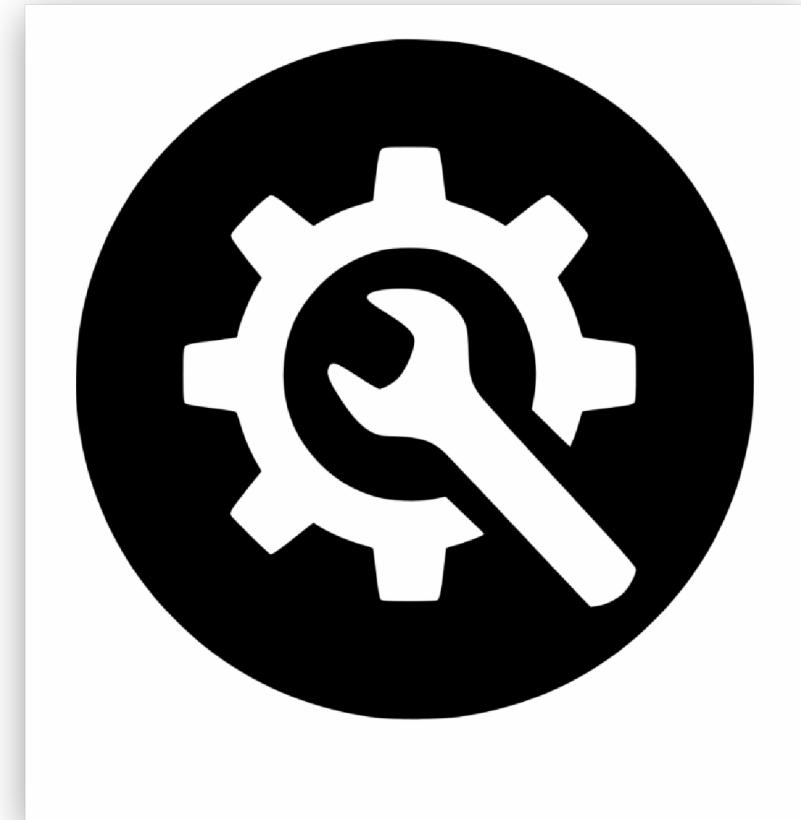
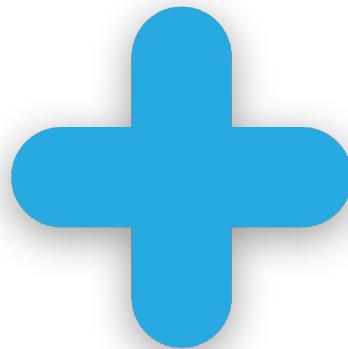
Weaknesses

- Weaker isolation
- Container security config *
- CPU shares are challenging
- Noisy neighbors

What else do we need?



What's a container image?



Layers



Sharing Layers

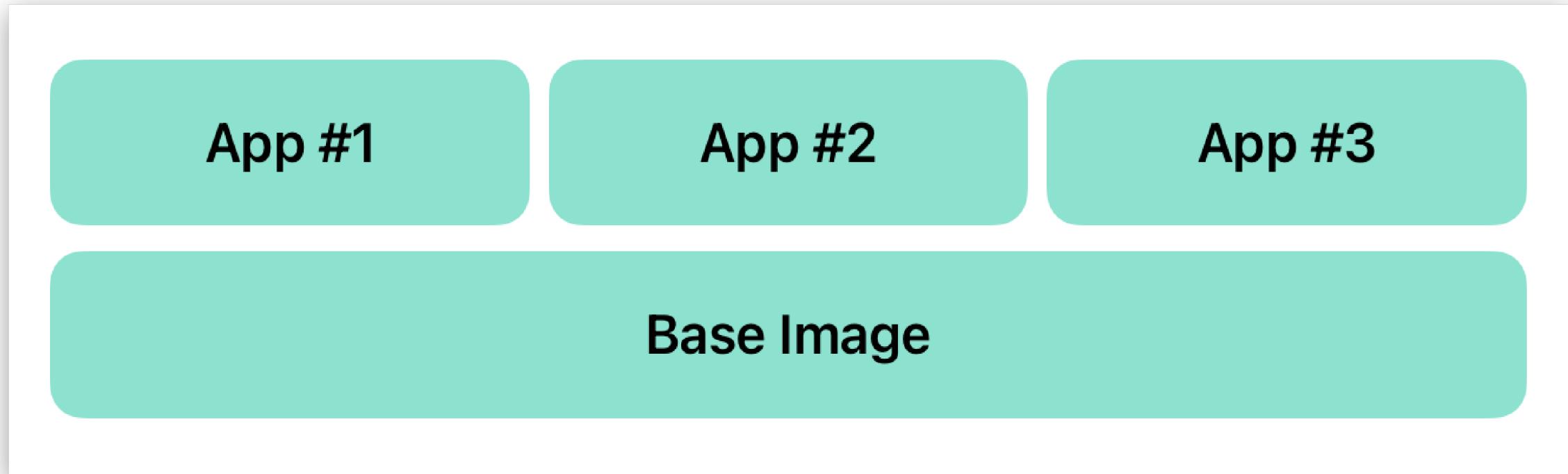


Image Hashes

App #1

sha256:~~fffc122~~

App #2

sha256:d12faa

App #3

sha256:6751aa

Base Image

sha256:cc12df

sha256:32acffd

Volume Mounts



Demo: Deconstructing a Container Image

Image Distribution

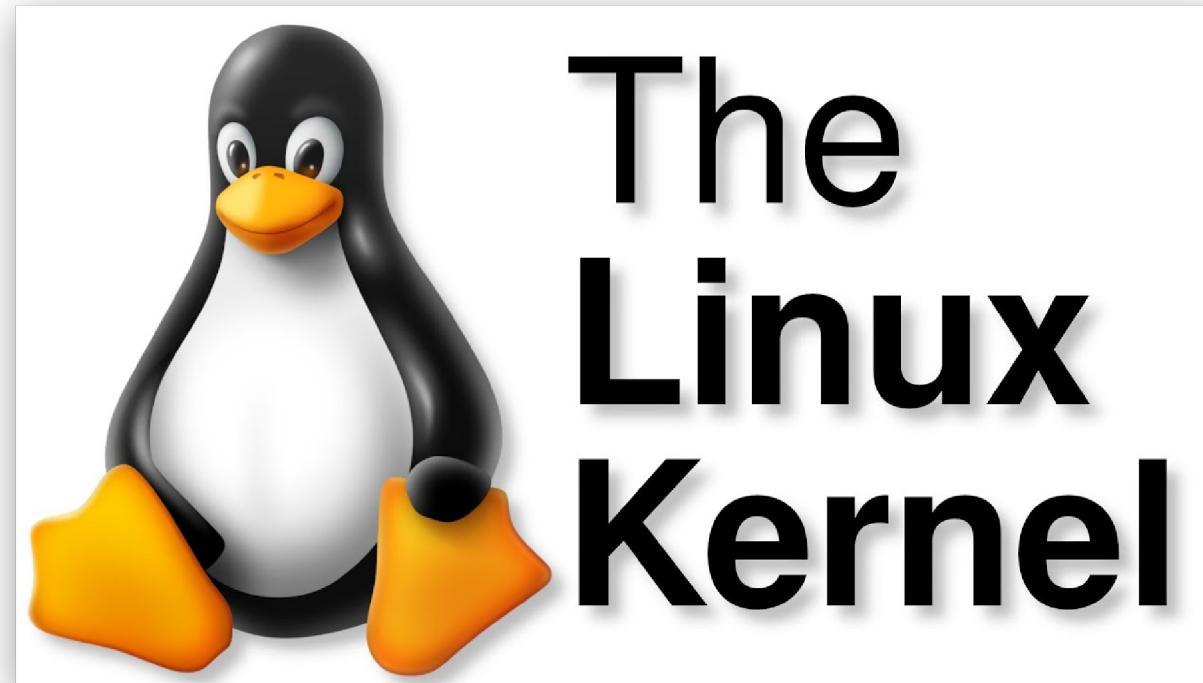


Demo: Fetching Images

Running Containers



What's Used to Run a Container



Linux Kernel Primitives

- CGroups
- Namespaces
- Capabilities
- Seccomp
- AppArmor

A Filesystem



Configuration / Settings





Run Docker Run

- docker run
- podman run
- kubectl apply
- AWS ECS, Lambda (or other Cloud Provider)

The Hard Way



Demo: namespaces & cgroups

Slides



<https://github.com/dmikusa/container-deep-dive-slides>



Feedback



<https://confoo.ca/f/9EA9EB26BB7244CB9B49199F19859999>

Questions?

