



Container Runtimes

Ryan Tolboom

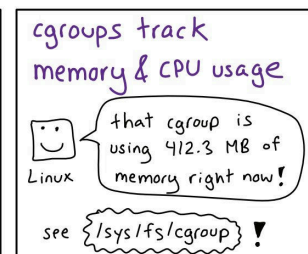
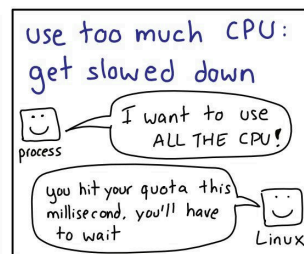
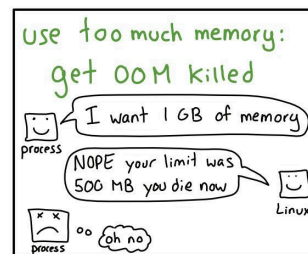
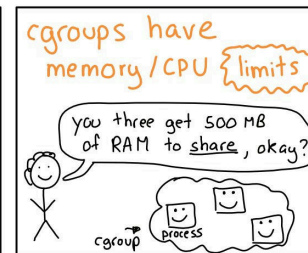
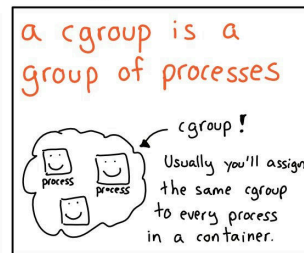
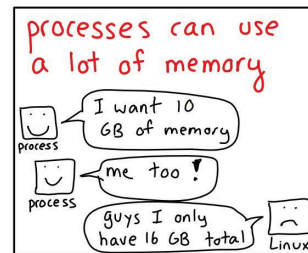
New Jersey Institute of Technology

Background

- Recall that containers are largely just namespaces and cgroups
- A container runtime is the software that is used to support the containerization of software

JULIA EVANS
@b0rk

cgroups



What specifically does a runtime do?



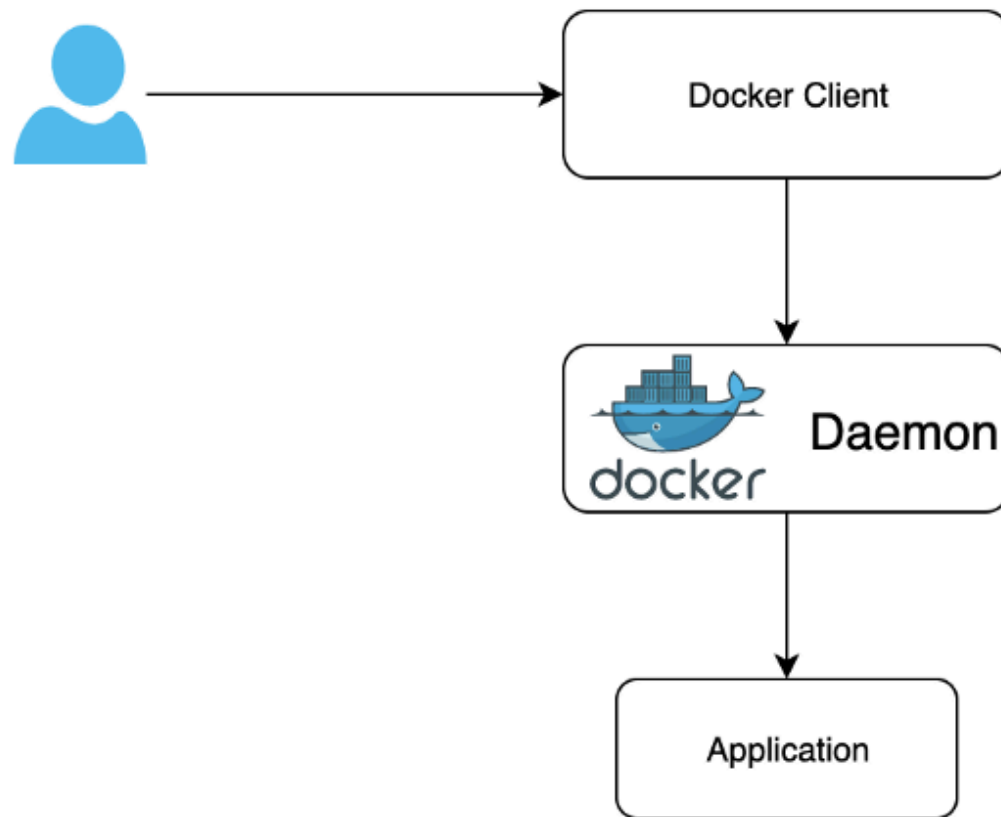
- image management
- lifecycle management
- creation
- resource management

[Source: Cloud Native Landscape](#)

Docker (original)

- One of the first runtimes, and as such many things strive to be Docker compatible.
- Runs on many platforms and is easy to use
- Extremely popular *not just for the runtime* but also for the tools that work with it such as their container registry Docker Hub.
- Supports app containers
- Requires a daemon to be running as root
- Losing ground in the production environment, but still common on dev machines

Old Docker



Docker Version < 1.11.0

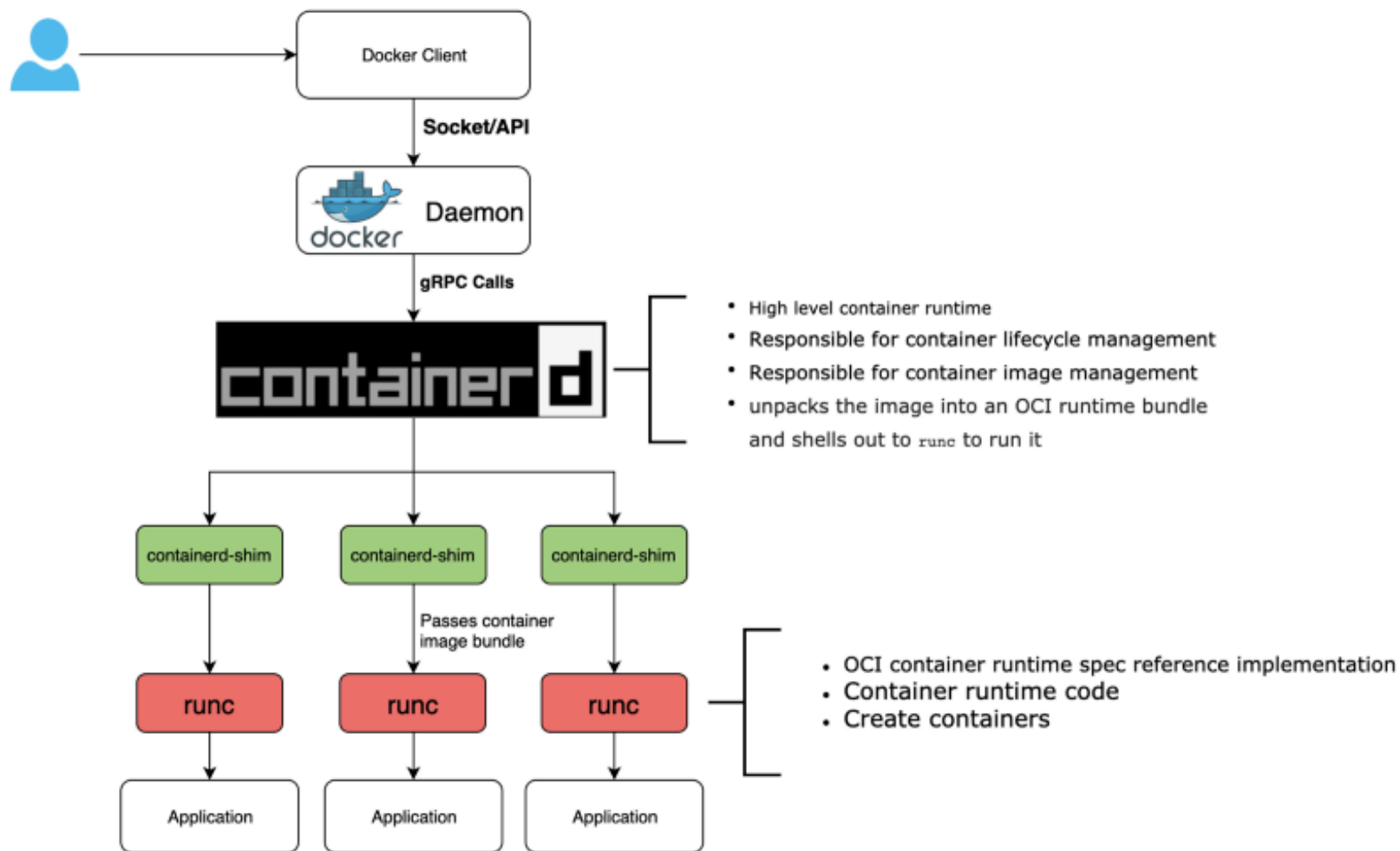


- Born from the Open Container Initiative (OCI) which was created to make open standards for containers
- Daemon that runs in the background and supports app containers (sound familiar?)
- Relies on a lower-level tool called runc (written by Docker)
- Current versions of Docker run on containerd



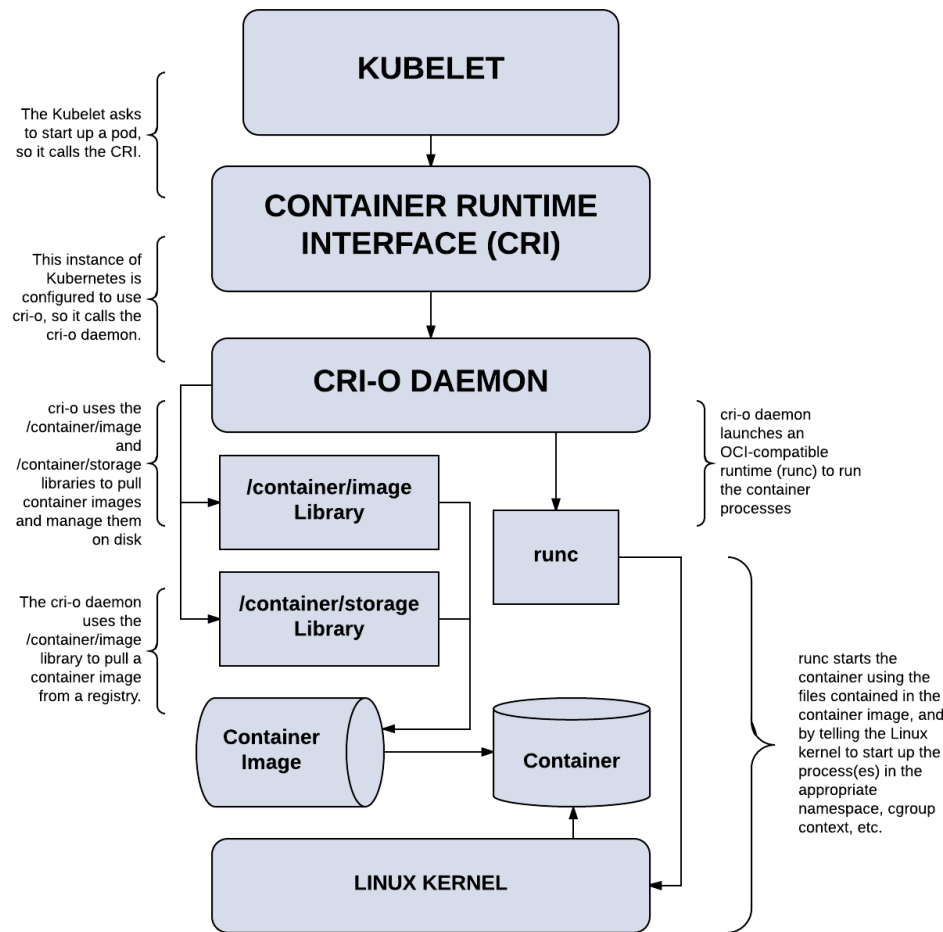
- Universal, lightweight container runtime
- Designed for security
- No Docker dependencies
- Supports a very simple model of container execution

New Docker

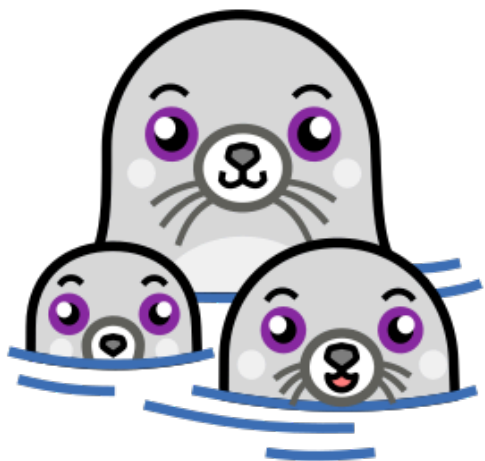


CRI-O

- Implements Kubernetes Container Runtime Interface (CRI) with OCI standards
- Runs as a daemon
- Support app containers
- Bridges the gap between K8s and containers, no need to run a `docker` command in-between
- Not meant for use by devs



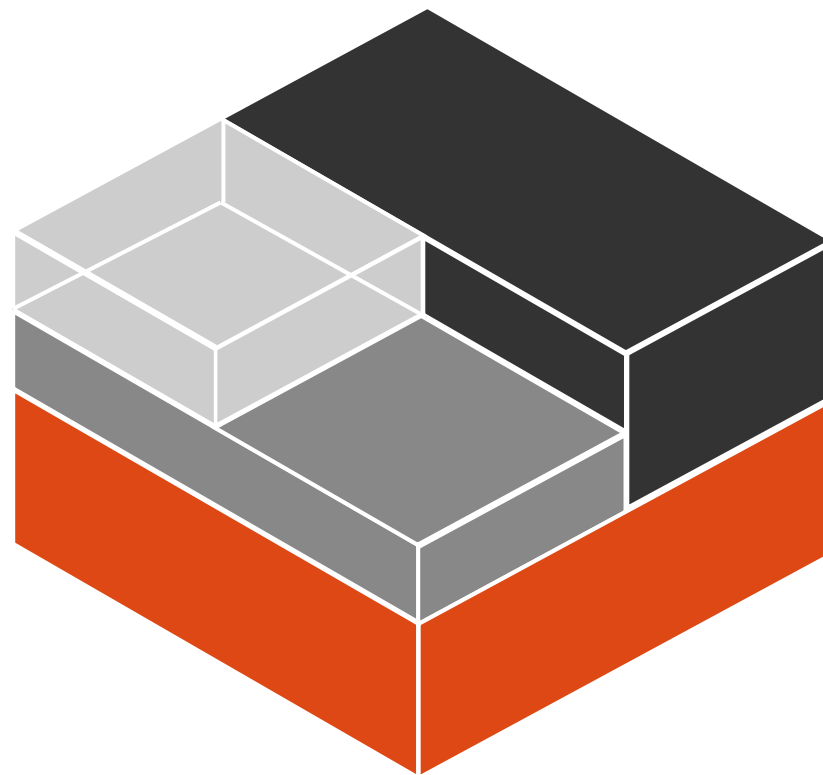
Podman



podman

- daemonless tool to use OCI containers
- has a similar CLI to `docker`
- supports similar features as `docker`
- Linux only

- Container engine designed for *system* containers
- Rootless, daemonless, and fast
- Keep this in mind if you encounter a *tough* virtualization feature
- Works with tools Linux admins are already familiar with
- Linux only



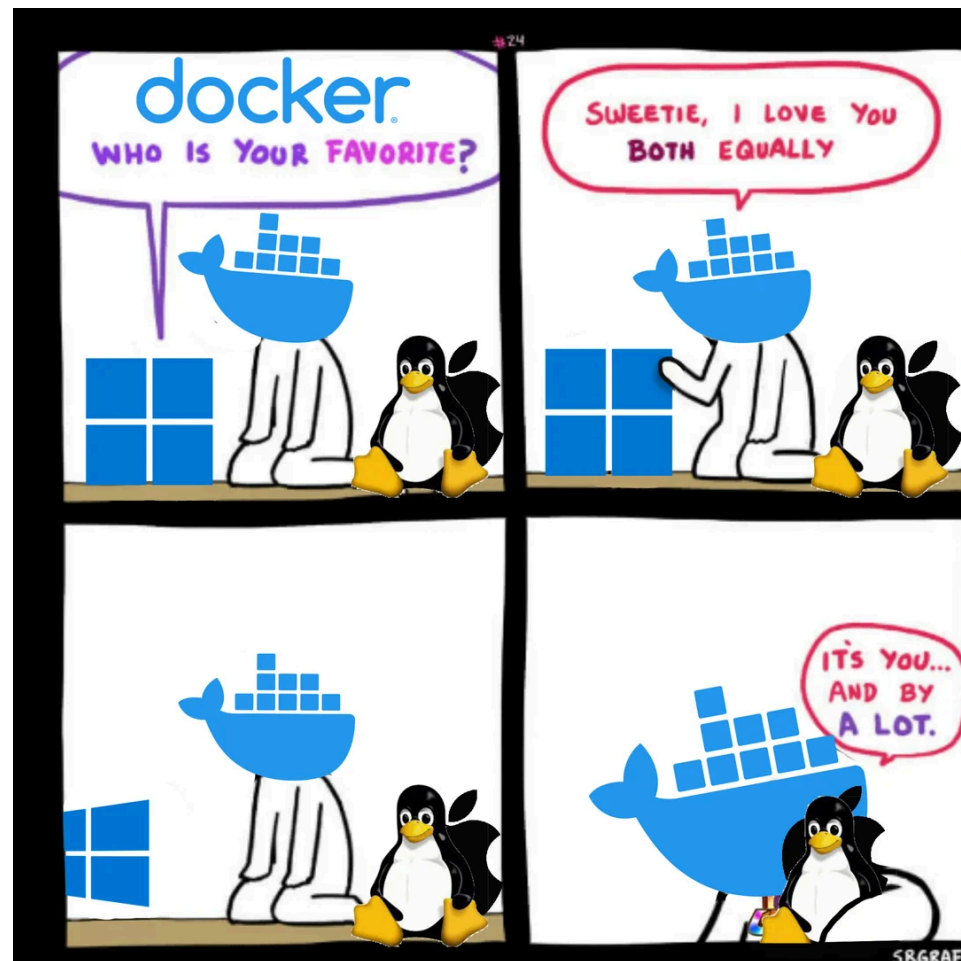
Just for fun: Bocker



- A lot of what Docker does can be implemented in around 100 lines of BASH!
- Remember these are things that are *already* built in to the Linux kernel
- Obviously Linux only

What does this mean?

- As a sysadmin, you have options
- Almost all Linux-based
- A few kernel features can spawn a massive shift in software deployment



Don't forget to demo...

```
demo@demo-VirtualBox: ~$  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
demo@demo-VirtualBox:~$
```

- the Docker daemon running
- a `docker save` to show the structure of the filesystem