

# Systems Administration with Containers

Ryan Tolboom

New Jersey Institute of Technology



# What is a container?



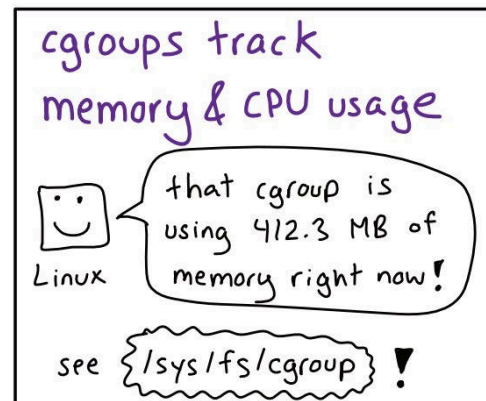
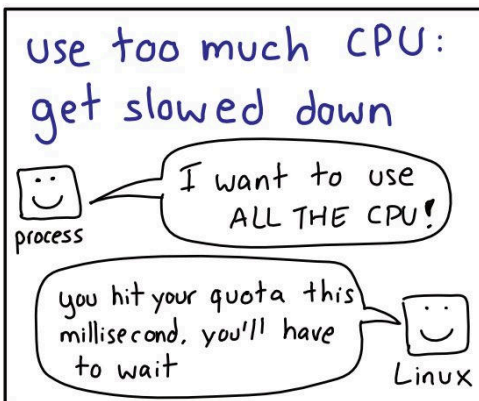
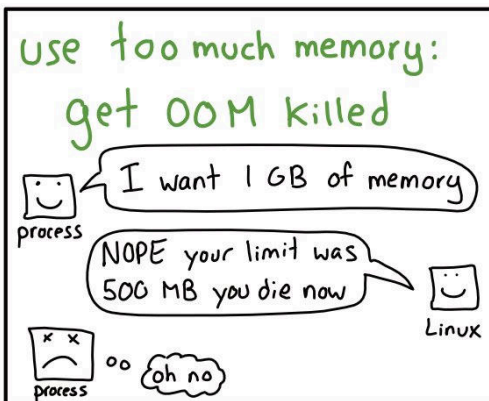
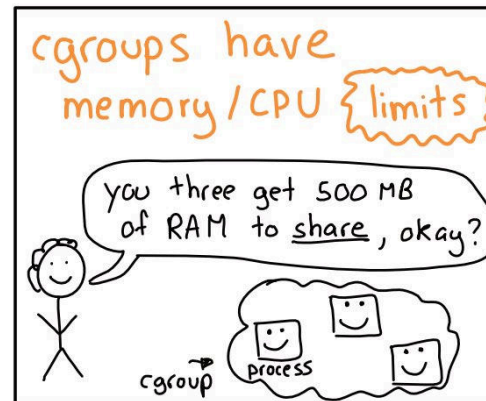
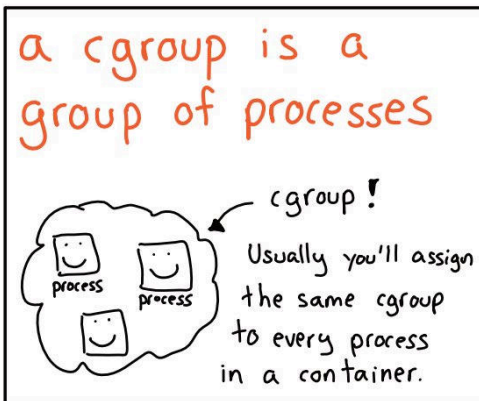
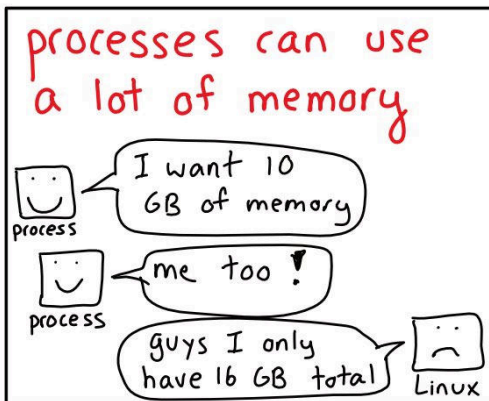
[“Multicolored containers”](#) by [Håkan Dahlström](#) is licensed under [CC BY 2.0](#)

- fully contained Linux system that configured exactly how the dev / ops team wants it to be
- uses chroot / cgroups for isolation
- allows multiple independent Linux environments to run on one host
- more lightweight than virtual machines (VMs)
- typically coordinated through a daemon process (docker for example)

# cgroups?

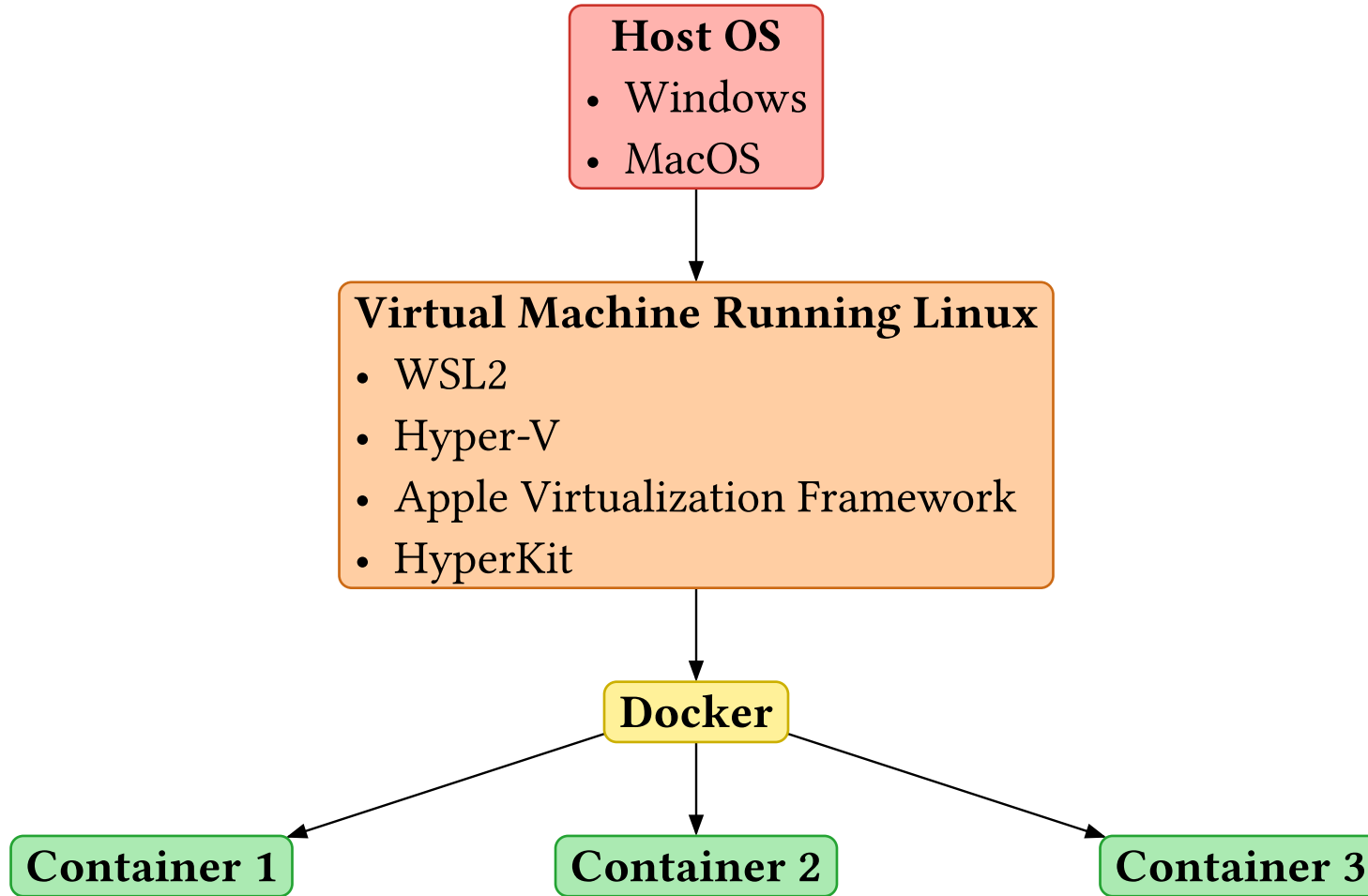
JULIA EVANS  
@b0rk

## cgroups

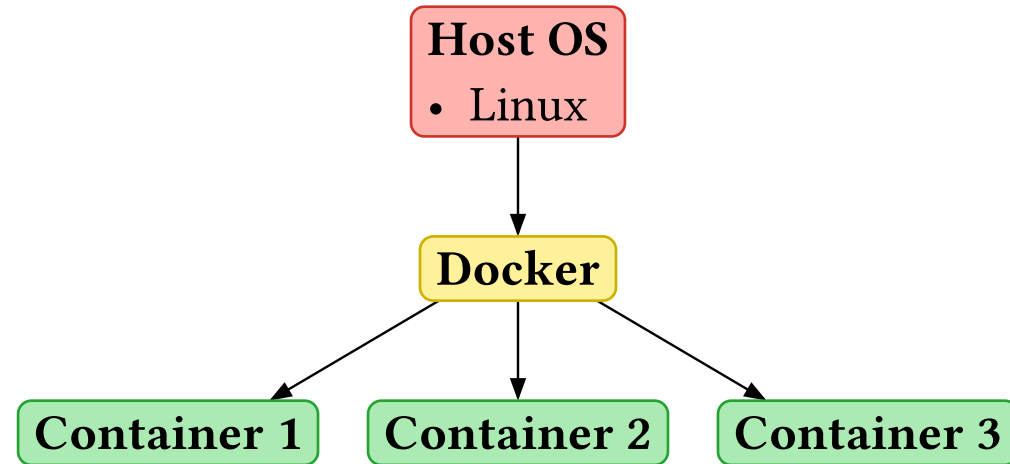


"cgroups" by Julia Evans is licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

# Docker on a Non-Linux Machine

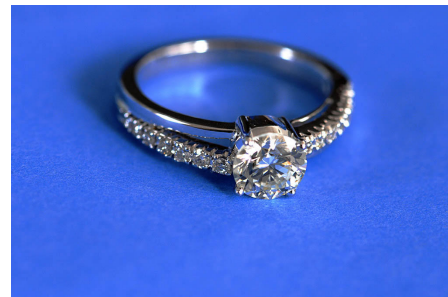


# Docker on a Linux Machine



# Precious vs. Expendable

- containers have a brief lifecycle, they are expendable
- the configuration information for an image matters (Dockerfile)
- the actual information inside the running container does not
- store your *data* somewhere that isn't inside a container, it's precious

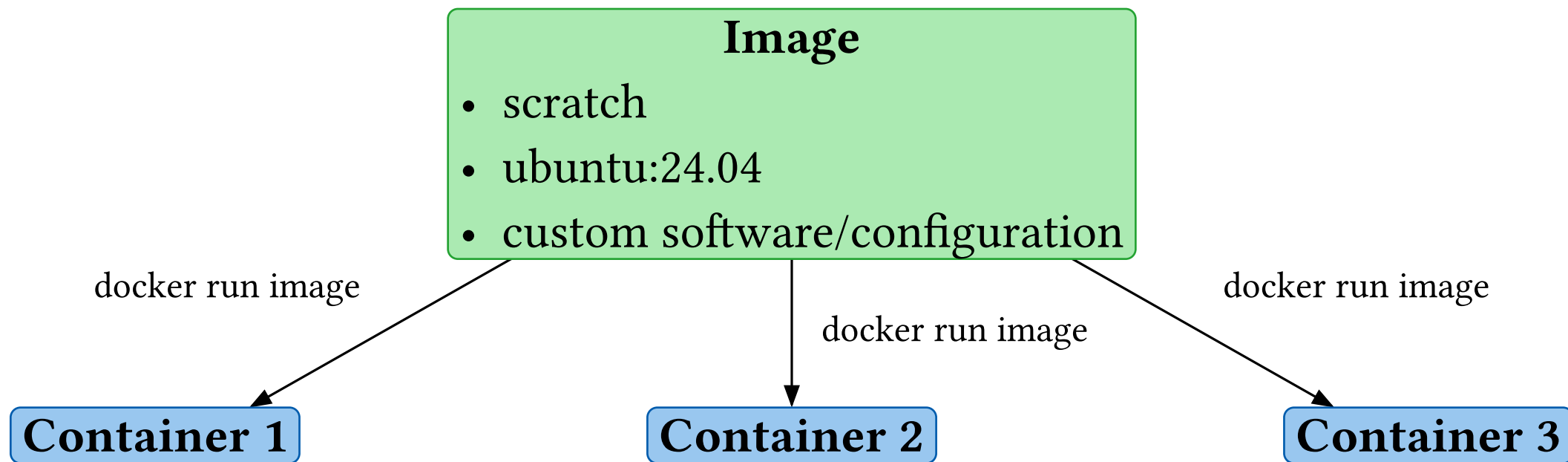


["Diamond ring"](#) by [Koshy Koshy](#) is licensed under [CC BY 2.0](#)



[Vintage Gumball Machine](#) is in the public domain under [CC0](#)

# Containers vs. Images



# Wrap it Up!



[Short mummy](#) is in the public domain

- traditional Linux systems administration still matters
- focus on *how* you set up something more than maintaining a set up you made earlier
- you should be able to make a script that deploys servers on demand
- everything is code