WHAT IS A CONTAINER?

Lunch and Learn series - *remote edition*

Nick Brandaleone
AWS Specialist SA - Containers

GROUND RULES

- Ask questions
- Try to be on mute when not asking questions
- I will be moving quickly
- https://github.com/nbrandaleone/what-is-a-container
- I will be using Cloud-9 (Ubuntu OS) for a demo

AGENDA

LESSON	TIME
Why are containers so popular?	5 minutes
What is the magic?	5 minutes
Demo, using BASH	20 minutes
Firecracker demo	5 minutes

WHY SO POPULAR?

- Code portability issue is solved
- Faster start-up time makes them preferrable to VMs
- Greater hardware efficiency makes them cheaper
- Isolation provides security (*not perfect though*)
- Docker tooling is *easy to use*

WHAT IS THE MAGIC?

CONTAINERS DO NOT REALLY EXIST

- Namespaces
- cgroups (Linux capabilities + Seccomp)
- COW or layered filesystem

Linux Kernel tricks - Windows should use .Net Core

NAMESPACES

CGROUP - limit cpu/memory for a group of processes.
MNT: It allows a process to have its own filesystem.
PID: The pid namespace gives a process its own view of /proc.
NET: Isolated network stack.
UTS: System's hostname and domain name.
USER: The user namespace maps the uids to different uids.
IPC: message queues and shared memory.

CGROUPS

Where namespaces isolate a process, cgroups enforce fair resource sharing between processes.

For example:

- how much memory a process can use
- how much CPU can a process use
- how many children processes can be spawned

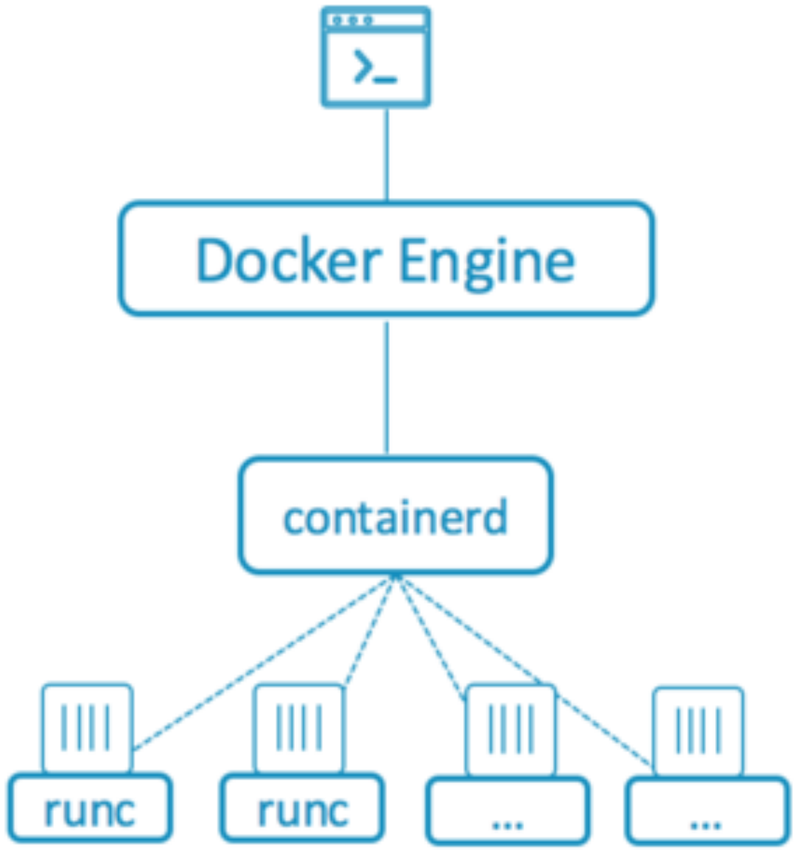
LAYERED FILESYSTEM

- Layered Filesystems are how we can efficiently move whole machine images around.
  - Also known as tarballs...
    - Storage Drivers:
      - overlay2
      - aufs (older version)

VOCABULARY

- Docker -> the company that made it easy to use containers. Purchased by Mirantis in late 2019.
- OCI -> The open source components of Docker
  - containerd (high-level interface)
    - runc (low level)

HOW THEY FIT TOGETHER



Same Docker UI and commands

User interacts with the Docker Engine

Engine communicates with containerd

containerd spins up runc or other OCI compliant runtime to run containers

UNDER THE HOOD

No `fork()` and `exec()`

We now `clone()` or `unshare()`

DOCKER AND KUBERNETES ARE WRITTEN IN GO, NOT C

DEMO TIME!

THANK YOU

Stay Safe and Sane

RESOURCES:

- https://www.infoq.com/articles/build-a-container-golang/
- https://gist.github.com/christophberger/5850541813 3d474486a88f958d8ea14b
- https://www.nickaws.net/linux/2020/03/05/Containers-are-not-magic.html

VIDEOS:

- https://www.youtube.com/watch?time_continue=2&v=sK5i-N34im8
- https://www.youtube.com/watch?v=Utf-A4r0DH8
- https://containersummit.io/events/nyc-2016/videos/building-containers-in-pure-bash-and-c

WORKSHOPS

- http://redhatgov.io/workshops/containersthehard_way/
- https://ericchiang.github.io/post/containers-from-scratch/
- https://github.com/riyazdf/dockercon-workshop