WHAT IS IT WE WANT IN CONTAINERS ANYWAYS?

Vincent Batts  @vbatts

bit.ly/vbatts-containers-anyways
bit.ly/252kBnL
$>\text{finger } $(\text{whoami})$
\text{Login: vbatts}
\text{Directory: /home/vbatts}
\text{Name: Vincent Batts}
\text{Shell: /bin/bash}

\text{Such mail.}
\text{Plan:}
\text{OHMAN}

$>\text{id -Gn}$
\text{devel opencontainers docker appc redhat golang slackware}
SO,
WHY, CONTAINERS?

Single Application
  Full System
    But Not a VM
      Except Maybe a VM
    Pods of applications
  Labels of services
Non-root
  Desktop Applications
    OMG AND CATS

https://www.flickr.com/photos/27549668@N03/
But wait,
What does "container" mean to you?
But wait,
What does "container" mean to you?
USE-CASE

Reproducibility
USE-CASE

Ephemeral Environments

THERE USED BE A DEV MACHINE HERE

NOW THERE ISN'T

memegenerator.net
USE-CASE

Freedom from host restrictions
USE-CASE

Easy delivery
USE-CASE

Integrate to existing process
USE-CASE

Controls and knobs
BUILD

Dockerfile

Source to Image (github.com/OpenShift/source-to-image)

appc/acbuild (github.com/appc/acbuild)

Your own Makefiles?
SHARING

registry (i.e. docker-registry, dockyard)

Host it yourself (i.e. tarballs on an http server)

Share recipes!
TOOLS

LXC / LXD

lmctfy

Docker

systemd-nspawn

runC

bubblewrap
• Namespaces
• Resource Controls
• Security and Isolation
systemd directives

Security Slides (CoreOS Fest2016)
**SHELL**

- `unshare(1)`

- `mount(8)` shared subtrees

- `chroot(1)`

- `procfs, sysfs, tmpfs`

- `cgroup filesystem (not for the faint of heart)`
STANDARDS!

Standard
/ˈstændəd/

noun
something used as a measure, norm, or model in comparative evaluations
STANDARDS!

Areas to Standardize:
- Packaging
- Runtime
- Networking
- Cloud
Call to Action!

Define your use-cases first

Get involved in the conversations

Ensure your container integration touchpoint stay generic, to avoid lock-in to a particular platform.

PoC tooling for your integration
THANKS!

VINCENT BATTs

@VBATTs  VBATTs@REDHAT.com

THANKS!