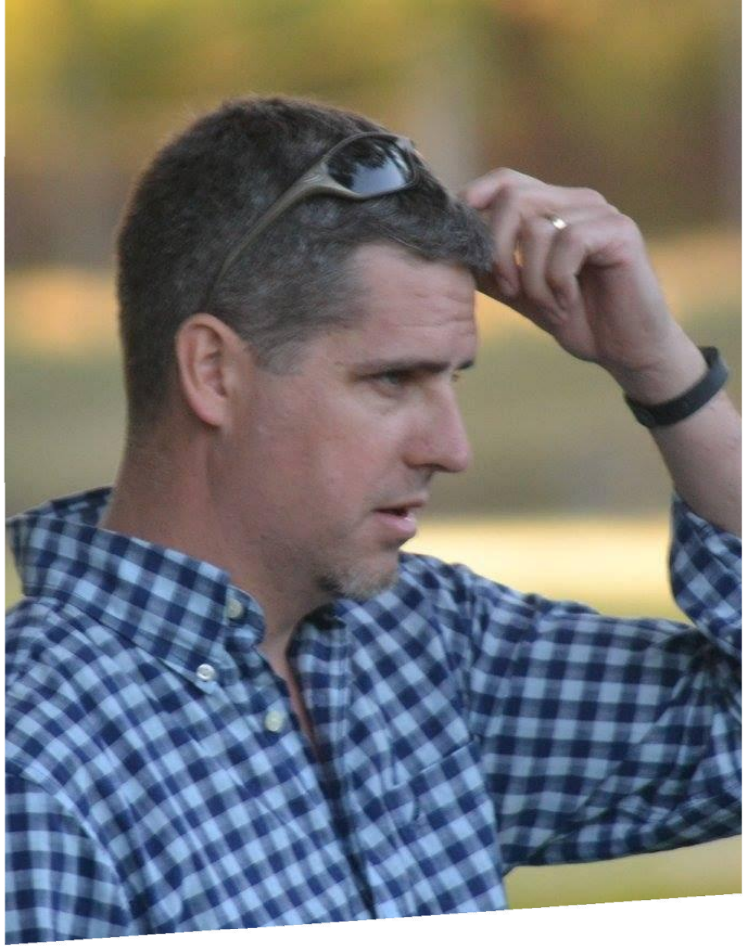

runC: The little engine that could (run Docker containers)

Phil Estes

IBM Cloud, Open Technology

dockercon 16

About Me

Phil Estes

Senior Technical Staff Member
IBM Cloud, Open Technology & Strategy
@estesp

> Docker core engine maintainer
> Member of “Docker Captains” program
> 10+ years involved in Linux/OSS

Community accomplishments:

> User namespace support in the Docker engine
> Helped design v2.2 image spec with multi-platform support
> Implemented first tool to create multi-platform images in Docker v2.3 registry & DockerHub

What is the OCI?

(Open Container Initiative)

Open Container Initiative (OCI)

- A **Linux Foundation** Collaborative Project
- Free from **control** by any particular vendor's specific cloud stack or ecosystem
- Includes:
  - **container runtime specification**
  - reference **runtime***
  - and now, an **image format specification**

dockercon 16

*seeded with **runc** + **libcontainer** by Docker

OCI: Specs and Status

> Runtime specification: Release 1.0.0-rc1 / June 2016

https://github.com/opencontainers/runtime-spec/releases/tag/v1.0.0-rc1

Approaching a finalized **1.0 release** (waiting on release criteria discussion)
Includes required core for containerization on Linux, Solaris, & Windows

> Image format specification: Milestone 0.3.0 / June 2016

https://github.com/opencontainers/image-spec/milestones/v0.3.0

Seeded with Docker registry v2.2 specification
Cadence of pre-releases underway in the repository:

- 0.2.0 release 3 weeks ago
- 0.3.0 targeted for this week

What is `runc`?

(and how do you pronounce it?)

Introduction to `runc`

> **runc** is a client wrapper around **libcontainer**
> **Libcontainer** is the operating system interface

runC requires two pieces of information: **a)** an OCI **config** (JSON) and **b)** a root **filesystem**

```
$ docker run -it --read-only -v /host:/hostpath alpine sh
/#
```

```
{
  "ociVersion": "0.6.0-dev",
  "platform": {
    "os": "linux",
    "arch": "amd64"
  },
  "process": {
  "terminal": true,
  "args": [
     "sh"
  ],
  "env": [
     "PATH=/usr/local/sbin:/usr/local/bin:/bin"
```

config.json

dockercon 16

runC: An open innovation platform

INTEREST **Implement low-level container features**

- Operating system level features should be defined in the OCI runtime specification
- New capabilities (PID cgroup controls, checkpoint/restore, seccomp) implemented in runC

INTEREST **OCI compliance/pluggable execution engine**

- Implement a OS/environment for containers via an OCI spec compliant binary
- Examples: runz (Solaris zones), runv (hypervisor-based), Intel Clear Containers

INTEREST **Iterative container configuration test/debug**

- Simple variant of "Docker-like" containers with less friction for quick modifications
- Low bar for dependencies: single **binary** + physical rootfs **bundle** + JSON **config**

Top 10 contributing companies to `opencontainers/runc`

1. Docker 2. OpenVZ 3. Huawei 4. Redhat 5. Google 6. IBM 7. SuSE 8. Pivotal 9. Fujitsu 10. Microsoft

Let's Demo `runc`!

```
# you'll see the following tools/projects during the demo:

/usr/bin/runc                https://github.com/opencontainers/runc

/usr/bin/ocitools            https://github.com/opencontainers/ocitools

/usr/local/bin/riddler       https://github.com/jfrazelle/riddler

/usr/local/bin/netns         https://github.com/jfrazelle/netns

/usr/local/bin/uidmapshift   http://bazaar.launchpad.net/~serge-hallyn/+junk/nsexec/view/head:/uidmapshift.c
```

OCI: Futures

- **Image Format Specification**
  - Implementation details under discussion; get involved if an area of interest for you or your company
- **More users/contributed implementations of the OCI spec(s)**
- **runC innovations moving up the stack**
  - checkpoint/restore underway (exposed via Docker engine)
  - Seccomp, user namespaces, PID limits are prior examples
- **What do you plan to do with OCI and/or the runC implementation?**

Thank you!

@estesp

github.com/estesp

estesp@gmail.com

https://integratedcode.us

IRC: estesp