runc: The Little (container) Engine That Could

Phil Estes
IBM Cloud Open Technologies
Twitter: @estesp
About Me

Phil Estes
Senior Technical Staff Member
IBM Cloud, Open Technologies
Container Strategy/Open Source Leader
Docker community core engine maintainer
Linux/open source expertise for 15 years @ IBM

Community activities & accomplishments
› Brought user namespace support to the Docker engine
› Helped design v2.2 image specification with multi-platform support
› Implemented first tool to create multi-platform images in Docker v2.3 registry
› Member of the “Docker Captains” program
What is the OCI?
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

*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 (-rc2 being cut very soon)**
Includes required core for containerization on Linux, Solaris, & Windows

> **Image format specification: Milestone 0.4.0 / August 2016**
  
  https://github.com/opencontainers/image-spec/milestones/v0.4.0

Seeded with Docker registry v2.2 specification
Cadence of pre-releases underway in the repository:
  - 0.4.0 release last week
  - 0.5.0 -> 1.0.0 releases underway
What is runc?
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
runc: An open innovation platform

**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

**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

**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*

Let’s Demo `runc`!

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

- `/usr/bin/runc`  [https://github.com/opencontainers/runc](https://github.com/opencontainers/runc)
- `/usr/bin/ocitools`  [https://github.com/opencontainers/ocitools](https://github.com/opencontainers/ocitools)
- `/usr/local/bin/riddler`  [https://github.com/jfrazelle/riddler](https://github.com/jfrazelle/riddler)
- `/usr/local/bin/netns`  [https://github.com/jfrazelle/netns](https://github.com/jfrazelle/netns)
- `/usr/local/bin/uidmapshift`  [http://bazaar.launchpad.net/~serge-hallyn/+junk/nsexec/view/head:/uidmapshift.c](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