Building a Container Supervisor

Michael Crosby

dockercon 16

> whoami

Docker since 0.3 - maintainer

dockerui - author

libcontainer - author

nsinit - author

runc - author

OCI - maintainer

containerd - author

> man containerd

containerd

- Fast, lightweight container supervisor
- runc (OCI) multiplexer
- Container lifecycle operations

> why

- runc integration
- Multiple runtime support
- Execution v2
- Decouple Execution from filesystem
- daemonless containers
- cleaner development

Benchmarks

```
> ./benchmark -count 100
INFO[0001] 1.149902846 seconds
```

> events

- lock free event loop
- concurrency control
  - 10 > 100 at a time

> daemonless

I want to upgrade the Docker daemon but I want my containers to keep running.

> container state

Managing state is easy when you don’t have any.

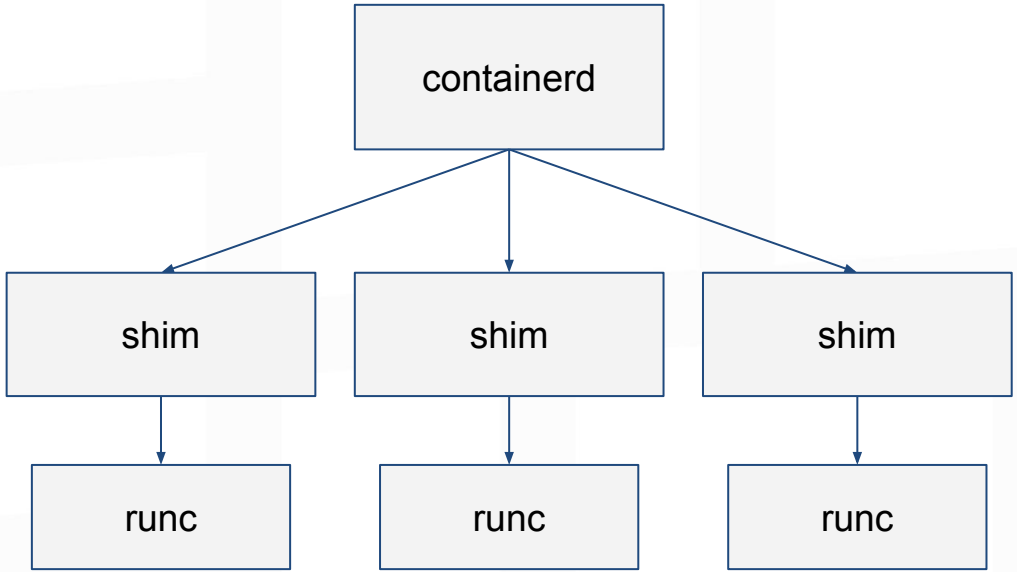
Don’t keep anything in memory.

/run is your friend

> daemonless problems

- exit code and wait4()
- tty / stdio
- reparenting
- facilitated by a shim

> containerd-shim



> exit status

- FIFO for blocking + file
  - fifo for exit event
  - file for exit status
- O_CLOEXEC
- RDONLY/WRONLY

O_CLOEXEC

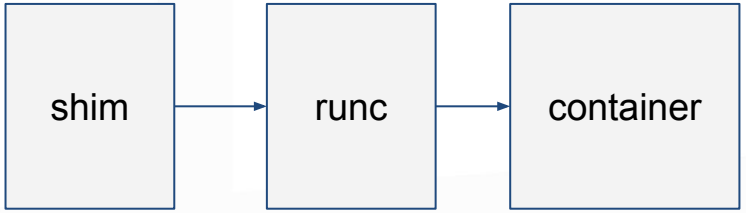
```
if (mkfifo("exit-fifo", 0666) != 0) {
    printf("%s\n", strerror(errno));
    exit(EXIT_FAILURE);
}
int fd = open("exit-fifo", O_WRONLY | O_CLOEXEC, 0);
```

> stdio

- FIFOs for data
- fifos have a buffer
  - /proc/sys/fs/pipe-max-size

> re-parenting

1. shim launches runc
2. runc launches container



> re-parenting

1. shim launches runc
2. runc launches container
3. runc exits
4. shim becomes parent of container

> re-parenting rules

1. Your parent is the process that forked you
2. If your parent dies, your new parent is PID 1

> subreaper

- prct - PR_SET_CHILD_SUBREAPER
- *“In effect, a subreaper fulfills the role of init(1) for its descendant processes.”*

PR_SET_CHILD_SUBREAPER

```
> ./parent
main() parent 27538
child process 27540 with parent 27539
parent 27539 exiting
child process 27540 with new parent 2391

> ps x | grep 2391
 2391 ?        Ss     0:00 /sbin/upstart --user
```

> The OOM Problem

How do you connect to OOM notifications before the user process starts?

> runtime workflow

- create
  - initialize namespaces and config
- start
  - exec the user’s process
- delete
  - destroy the container

create/start/delete

```
> runc create test
> runc start test
> runc delete test
```

> code

https://github.com/crosbymichael/dockercon-2016

Thank you!