

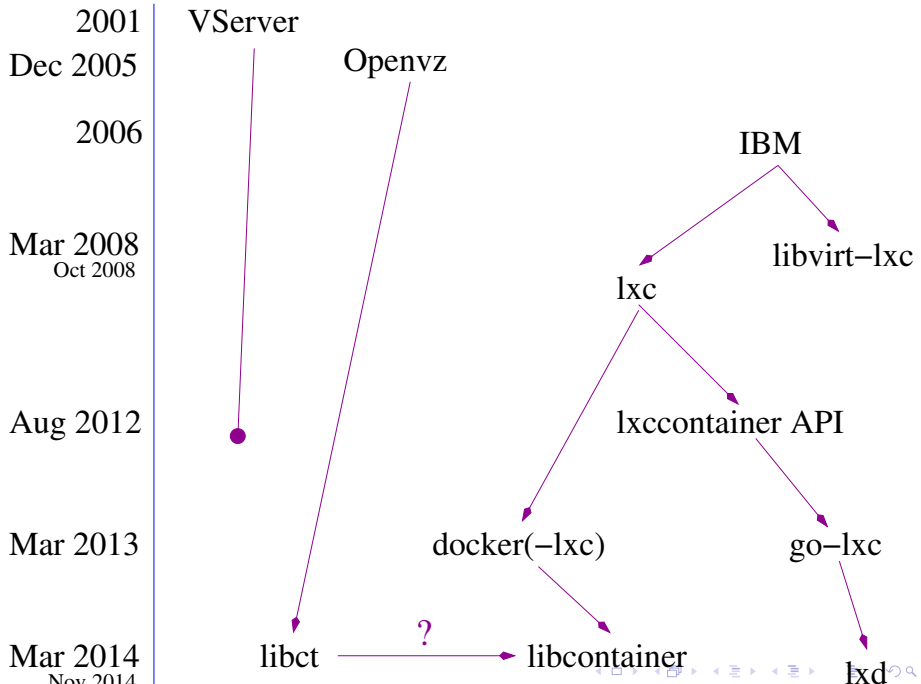
Container management APIs

Serge Hallyn

Canonical, Ltd

serge.hallyn@ubuntu.com

October 4, 2015



Regular Usage: Overview

- Let's first go over the basics:
 - Download an image
 - Run a command in a container
 - query images and containers
 - Publish the new image
- Demos
 - lxc
 - lxd
 - docker
 - libct

Digging Deeper

- Getting familiar with the commands is important
- Docker and LXD have REST APIs
- LXD
 - locally use `/var/lib/lxd/unix.socket`
 - endpoints:
 - `/1.0`
 - `/1.0/images`
 - `/1.0/containers`
 - <https://github.com/lxc/lxd/blob/master/specs/rest-api.md>
- Docker
 - locally use `/run/docker.sock` unix socket
 - examples:
 - example: `http:/images/json`
 - example: `http:/containers/ID/start`
 - https://docs.docker.com/reference/api/docker_remote_api/

- A new way to run OpenContainers format containers
- (Usable from other languages - which go pkgs are not)
- Requires
 - untarred rootfs
 - config.json: mounts (brief), user, env, capabilities, arch
 - runtime.json: mounts(details), devices, security, hooks, more
- `github.com/opencontainers/runc`

Building new tools

- LXD/LXC

- Designed around LXC containers
- Basis for LXD
- Written in C
- Bindings in python, go, haskell, lua, ruby, ...
- github.com/lxc/lxc

- libcontainer

- Written in go
- Focus on driving docker containers
- <https://github.com/opencontainers/runc/tree/master/libcontainer>

- libct

- Designed to implement container drivers
- Written in C
- Bindings in python and go
- github.com/xemul/libct