Firecracker as a container runtime
FOSDEM | 03.Feb.2019
Hi, I'm Dongsu

Dongsu Park
Software engineer, Kinvolk

Working on container runtimes, Flatcar Linux, kube-spawn, etc.

Github: dongsupark
Email: dongsu@kinvolk.io
Kinvolk

The Deep-stack Kubernetes Experts

Engineering services and products for Kubernetes, containers, process management and Linux user-space + kernel

Blog: kinvolk.io/blog
Github: kinvolk
Twitter: kinvolkio
Email: hello@kinvolk.io
Firecracker

- Lightweight Virtualization Machine Monitor (VMM)
- Spawns multiple micro-vms in an efficient way
- For short-lived workloads
- A good balance between traditional VMs and containers
- Heavily makes use of Linux KVM
- Based on crosvm from Google ChromeOS
- Open sourced in Dec 2018:
  - [https://github.com/firecracker-microvm/firecracker](https://github.com/firecracker-microvm/firecracker)
  - Written in Rust
Firecracker (example)

```bash
# ./firecracker --api-sock /tmp/firecracker.socket
# curl --unix-socket /tmp/firecracker.socket -i \
  -X PUT 'http://localhost/boot-source' \  
  -H 'Accept: application/json' \  
  -H 'Content-Type: application/json' \  
  -d '{
    "kernel_image_path": "/tmp/hello-vmlinux.bin",
    "boot_args": "console=ttyS0 reboot=k panic=1 pci=off"
  }'

# curl --unix-socket /tmp/firecracker.socket -i \
  -X PUT 'http://localhost/drives/rootfs' \  
  -H 'Accept: application/json' \  
  -H 'Content-Type: application/json' \  
  -d '{
    "drive_id": "rootfs",
    "path_on_host": "/tmp/hello-rootfs.ext4",
    "is_root_device": true,
    "is_read_only": false
  }'
```
Firecracker (example)

```sh
# curl --unix-socket /tmp/firecracker.socket -i \
   -X PUT 'http://localhost/actions'  \
   -H 'Accept: application/json'     \
   -H 'Content-Type: application/json' \
   -d '{
      "action_type": "InstanceStart"
   }'

### Then the microvm boots
```
Integration with container managers

- With containerd:
  - [https://github.com/firecracker-microvm/firecracker-containerd/](https://github.com/firecracker-microvm/firecracker-containerd/)
  - 3 components: Agent, Snapshotter, Runtime
  - Containerd-specific shim needs to be installed
  - Heavily depends on gRPC/ttRPC interface of containerd

- With Kata-container:
  - [https://github.com/kata-containers/runtime/pull/1044](https://github.com/kata-containers/runtime/pull/1044)
  - Relatively clean implementation
  - Basically a VM-based container runtime
CRI-O

- OCI-based Kubernetes Container Runtime Interface
  - https://github.com/kubernetes-sigs/cri-o
  - Implements Kubelet CRIs using OCI runtimes (runc)
  - Does not have complicated internal interfaces

- Crictl - Command-line tools
  - https://github.com/kubernetes-sigs/cri-tools/
  - Similar cmdline interface shown to users
CRI-O runtime for Firecracker

- Goal: make a runtime add-on in CRI-O, for Firecracker
  - Instead of the standard runtime v1 (oci)
  - Support VM-based container runtime is in progress
    - https://github.com/kubernetes-sigs/cri-o/pull/2025

- Rely on the Firecracker Go-SDK
  - https://github.com/firecracker-microvm/firecracker-go-sdk
  - A good wrapper around low-level KVM functionalities
  - Written in Go, easy to be integrated with container runtimes
CRI-O runtime for Firecracker

- Crio (daemon)
- Fc-runtime (add-on)
- Firecracker (daemon)
- Crictl (cmdline tool)
- micro-VMs
CRI-O runtime for Firecracker

- Current PoC available:
  - [https://github.com/kinvolk/cri-o/tree/dongsu/fc-runtime](https://github.com/kinvolk/cri-o/tree/dongsu/fc-runtime)
  - Reads config for setting up Kernel & rootfs for firecracker
  - When starting container, spawns a firecracker process
  - Still in heavy development
- A simple tool for creating Kernel & rootfs image
  - [https://github.com/dongsupark/debian-firecracker](https://github.com/dongsupark/debian-firecracker)
  - Based on a Dockerfile to create vmlinux.bin & rootfs.ext4
Demo
Future works

- Clean up the tree to create a pull request to upstream
  - In sync with VM runtime of CRI-O

- Missing features
  - attach, exec, etc.

- Similar work for rktlet (?)
Thank you!

Dongsu Park
Github: dongsupark
Email: dongsu@kinvolk.io

Kinvolk
Blog: kinvolk.io/blog
Github: kinvolk
Twitter: kinvolkio
Email: hello@kinvolk.io