AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated 06/04/2025

Sign up

Firecracker training

3 days (21 hours)

Presentation

Created and developed by Amazon in 2018, Firecracker is a minimally designed open source virtualization technology running Apache 2.0. It can create and manage secure, multi-tenant containers and functional services. This makes it possible to use secure services while combining the speed, resource efficiency and performance offered by containers with security and isolation offered by traditional virtual machines.

By multiplying minimalist devices, it excludes unnecessary peripherals and guest functionality to reduce the memory footprint and attack surface of each microVM. This improves security, reduces boot time, increases hardware utilization and enables a secure sandbox environment for each container.

FireCracker combines fast/high-intensity start-up times with security based hardware virtualization.

MicroVMs offer improved security and workload isolation compared to traditional virtual machines, while enabling the speed and resource efficiency of containers.

If you're still not convinced of Firecracker's effectiveness, I invite you to read this article, which presents this new secure serverless architecture based on AWS Lambda, MicroVMs and Fargate.

Firecracker currently supports Intel processors, and is integrated with Kata containers, Weave FireKube (via Weave Ignite) and containerd (via firecracker-containerd). Firecracker also runs on Linux.

Objectives

- How to use Firecracker
- Create secure, multi-tenant containers
- Mastering sandboxes
- Creating and managing MicrosVMs

Target audience

- AWS & Apache developers
- Rust developers

Prerequisites

- Knowledge of AWS and Apache
- Developing on Rust

Firecracker training program

API REQUEST

- ACTIONS : Instance Start/ Flush Metrics/ SendCtrlAltDel
- Logers API REQUEST
- Updating a Block Device
- Update Network Interface
- Removing Rate Limiting Supression de la Limitation de Débit

FIRECRACKER DESIGN

- SCOPE
  - What is FireCracker?
  - Features
  - Specifications
  - Host integration
  - Host network integration
  - Storage
- INTERNAL ARCHITECTURE
  - Threat containment
- COMPONENTS AND FEATURES
  - Machine model: guest CPU layout/exposure/Clocksources available for guests
  - I/O: Storage, networking and rate limiting
  - MicroVM Metadata Service: Jailling/ Cgroups and quotas/ Monitoring

Setting up a Development Environment for FireCracker

- Local: Local Bare-Metal Machine / Local Virtual Machine (macOS with VMware Fusion)
- Cloud: AWS/GCP/Addemdum/Microsoft Azure

Getting started with FireCracker

- Prerequisites
- Get the FireCracker binary
- Running FireCracker
- Building from the source
- Execution following Integration Test
- Appendix A: KVM Access configuration
- Appendix B: Docker configuration

The FireCracker guard

- The FireCracker guard
- Using the keeper
- Janitor operations
- Example of Execution and Notes
- Comments
- Warnings

Micro VM Metadata Services

- MMDS BackEnd: Example of use: Rotation of identification information
- The Data store
- Dumbo: MMDS network stack / TCP manager / MMDS endpoint / Connection

Reactive FormsFirecracker network configuration

- For the host
- FireCracker configuration
- For the guest
- Clean

Production host configuration recommendations

- Guard configuration
- Host Security Configuration : Mitigating side-channel problems / Known kernel problems

Creating custom roofts and kernel images

- Creating a kernel image

- Creating a Rootfs image

Using the Firecracker Virtio-vsock device

- Using the Firecracker Virtio-vsock device
- Prerequisites
- Firecracker Virtio-vsock Design (Host/Guest initiated connections)
- Virtio-vsock device configuration
- Examples (Using internal plug tools (nc-vsock and socat)

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.