Containers And Microservices

Concepts, OCI, runC, CNCF, Containerd, Docker, Microservices, App Architecture, Networking, Project

Microservices have revolutionized server-side application development. Most modern engineering teams have evolved to running as much code as possible in containers and benefit from the range of enhancements they offer.

As an application developer, you can think of a container as an efficient sandbox within which your process runs (multiple processes can run in the one container, but usually it is one process per container). Containers offer a sandbox based on unique namespaces and c(ontrol) groups (e.g. resource limits/accounting).

The world of containers is undergoing rapid transformation (Docker and its components - e.g. runc - are important but they certainly are not the whole story). Developers really need to understand how all the moving parts fit together in the modern container world. They also need to be aware of how similar and contrasting containers are with traditional "process in OS" approach. This course focuses on individual containers and how attendees can build microservices (slice of their application) to run inside-a recommended follow-on course on *Kubernetes* explores how to orchestrate clusters of containers in innovative ways.

Target Audience

Server application developers wishing to compose applications from many microservices running in containers.

Prerequisites

Developers experienced in server-side software.

Knowledge of the Go language is useful.

Note

Please note this course does not cover Kubernetes (apart from a brief introduction). We have a separate course dedicated to *Kubernetes*, which we recommend attendees takes after this course.

Contents of One-Day Training Course

Container Landscape

What containers offers over VM approach
Intro to microservice development
Big picture review of the sometimes confusing container landscape: its standard bodies, specs, commercial businesses, cloud offerings, tooling

OCI -Open Container Initiative

https://www.opencontainers.org and https://github.com/opencontainers
"An open governance structure for .. creating open industry standards around container formats and runtime"
Three specs:
* Image is for file system layout
* Runtime is how to run a container
* Distribution (new) is how images travel
Open source projects – runc (includes libcontainer), image-tools, runtime-tools (runc is the runtime used by Docker and by most Kubernetes installations)

CNCF - Cloud-Native Computing Foundation

https://www.cncf.io
"builds sustainable ecosystems .. around a constellation of high-quality projects that orchestrate containers as part of a microservices architecture."
Review of CNCF projects, including ...
* Containerd - https://containerd.io/

Docker

The Docker toolsuite is the market-leading container platform – let's explore what services it offers and how to use them
Available for Linux, macOS and Windows

Kata Containers

OCI-compliant open-source project to run containers via a light-weight hypervisor (combine containers and VM approaches)

Building Microservices

Sub-dividing a large app into microservices
As an application developer, what steps you need to take to prepare your code to run inside a container

Application Architecture

What steps are needed to build a microservice and how to optimize?
Handling data, IPC, config, lifecycle, etc.

Microservices And Security

Under what security context does your microservice code run?
Importance of isolation

Microservices And Networking

* CNI (container networking) - https://github.com/containernetworking
* Envoy (distributed proxy) - https://www.envoyproxy.io

Tour of Source Trees

Understanding internals is good: exploring runc CLI and libcontainer (written in Go), containerd (written in Go), Docker (written in Go), Kata (yup, also written in Go)

Project

How best to partition a large server app to run as microservices in many containers
Practical architectural guidance