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

Updated 03/06/2024

Sign up

Apache OpenWhisk training

3 days (21 hours)

Presentation

Apache OpenWhisk is a distributed, open source serverless platform that executes functions (fx) in response to events at any scale. It is designed to facilitate the development of applications in the cloud. Serverless computing dramatically simplifies software development. It allows you to concentrate solely on your application, while your cloud provider manages the servers you need. The absence of a server saves you time, which can be used for debugging, monitoring and maintaining the infrastructure. In our Apache OpenWhisk training course, you'll discover a vendor-independent approach using preconfigured containers, microservices and Kubernetes as the cloud operating system.

Objectives

- Discover Apache OpenWhisk
- Build complex applications without server configuration
- Examine how OpenWhisk's serverless architecture works, including the use of packets, actions, sequences, triggers, rules and flows.
- Learn how OpenWhisk differs from existing architectures, such as Java Enterprise Edition.
- Manipulate OpenWhisk functionalities using the command line interface or a JavaScript API.
- Design applications using common Gang of Four design patterns.
- Use architectural design models such as the view-controller model to combine several OpenWhisk actions.
- Test and debug your code in a serverless environment.

Target audience

- Developers interested in an open source server-less cloud platform
- Contributors
- Operators

Prerequisites

- Knowledge of Docker and Kubernetes
- Programming experience in at least one of the languages supported by OpenWhisk

OpenWhisk training program

Introduction to Apache OpenWhisk

- What is OpenWhisk?
- OpenWhisk architecture
  - Severless model
- Microservices architecture
- Java Enterprise Edition architecture with OpenWhisk
- Installing and configuring OpenWhisk

Openwhisk application

- CLI Bash
- IBM cloud
- Creating a simple contact form
- Form validation
- Address validation
- Return result
- Saving form data
- Database storage
- Mailgun configuration

Programming models

- Classic design models
- Actions (a functional logic)
- Creating and invoking actions
- Consuming an action via a REST API
- OpenWhisk CLI
- Triggers & Rules
  - Automating OpenWhisk actions
  - Managing actions with OpenWhisk packages

OpenWhisk programming languages

- Javascript
- Python
- PHP
- Go
- Ruby
- Swift

Deployment with Apache OpenWhisk

- Infrastructure management
- Server management
- Scaling with Docker containers
- Building components using containers
- Different deployment options
- Container frameworks: Kubernetes, OpenShift, Compose
- Helm diagram

Developing Openwhisk with Python

- Python runtime and its contents
- Runtime libraries
- Packaging a Python application in a Zip file
- How Virtualenv and Pip work
- Virtual environment automation
- Using the REST API with Python
- Invocations, activations and triggers with Python

CoucheDB with OpenWhisk

- What is CoucheDB and how do I use it?
- How CoucheDB works
- Exploring CoucheDB from the command line
- Creating a database
- CoucheDB design documents

Kafka with OpenWhisk

- Introducing Apache Kafka
- Messages and keys
- Creating a Kafka instance in IBM Cloud
- Creating a topic
- Binding and Feed creation
- Send messages using kafkacat

Deploying OpenWhisk with Kubernetes

- Kubernetes installation type: Helm, local, on the coud
- Architecture for deploying kubernetes in the cloud
- Generic Kubernetes installation procedure with cloud-init
- Installing OpenWhisk
- Kubectl and Helm configuration
- Configuring the OpenWhisk command-line interface

Apache OpenWhisk flows

- Flow implementation
- Event flow
- Trigger T
- A flow action
- Creating a flow
- Deleting a feed
- Pause and resume flow

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, with 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.