

En-lighten Technology Ltd

Testing Cloud Native Software Overview

Overview

Cloud native is an approach to building applications that takes full advantage of cloud computing. It requires not only embracing new technology but also new ways of working. To realise the significant benefits of cloud native, changes in the way software is built and tested have to be made. This course presents the latest testing tools and techniques to achieve continuous delivery and deployment of cloud native software. A high-level view of the end-to-end software delivery process is presented. All the testing stages and example tools that can be used to achieve high-quality software are explained and demonstrated.

Attendees will learn the following:

- What cloud native is
- Continuous deployment pipelines and quality gates
- Microservice testing
- Contract tests that verify request responses against a specification
- Non-functional testing
- The importance of defining KPIs and SLOs for microservices
- Why observability is vital
- Testing in production techniques

Audience and Recommended Background

The course is focused on how to test cloud native systems but is suitable for anybody interested in how high quality software delivery can be engineered. This includes engineers, architects and managers.

Course Approach

The course is presentation based and there will be demonstrations of the various tools and testing techniques presented.

Course Duration: 1/2 day

Course Content

What is Cloud Native?

- What are we trying to achieve with cloud native
- Microservice architectures
- Key benefits of a cloud native transformation

Continuous Delivery and Deployment

- The importance of continuous delivery
- Delivery pipelines that protect a product
- Delivering independent services
- No more end-to-end testing

En-lighten Technology Ltd

Functional Testing of Microservices

- The role of unit tests
- Consumer driven contract testing for independent service testing
- Test stubs and sharing stubs
- Building confidence to release a microservice

Non-Functional Testing

- Defining KPI's and SLO's
- Load testing
- Stress testing

Observability

- The need for observability
- The three pillars of observability:
 - Logs, metrics, traces
- Key microservice metrics:
 - Four golden signals
 - Apdex score
 - RED
 - USE
- Visualising metrics

Testing in Production

- Executing zero downtime deployments
- Releasing features
 - Canary deployments
 - Blue-green deployments