Kubernetes Native Application Development

Course ID #: 7000-070-ZZ-Z
Hours: 14

Course Content

Course Description:

In this course, developers will learn how to build containerized applications targeted for enterprise-grade production environments. You'll explore patterns in containerized application architecture, techniques for eliminating friction in the development process, how to test and debug containerized applications, and how to instrument applications with healthchecks, monitoring tools, and common container logging patterns. After mastering these techniques, we'll turn our attention to devops and building container-native continuous integration pipelines powered by Jenkins and Kubernetes.

Course Objectives:

Upon completion, students will be able to:

- Rapid code-build-test iteration cycles
- Developer-local Kubernetes environments
- IDE & debugger container integration
- Optimizing container startup
- Gracefully handling container failure
- Containerizing legacy applications
- Instrumenting containers with health checks
- Log management strategies
- Instrumenting containers with Prometheus monitoring
- Unit and integration tests for containerized applications
- Containerized continuous integration pipelines
- Image and config hierarchies in CI
- Build server management
- Unit and integration testing pipelines
- Integrating Jenkins with registries, Kubernetes clusters and standalone Docker engines
- Securing and packaging applications for production

Target Audience:

This course is targeted at students with the following:

- Motivations: Develop container-native applications, and implement fully containerized CI
- Roles: Developers, application architects, devops

Prerequisites:

- CN120 course and prerequisites therein, or equivalent experience
- Familiarity with the Bash shell
- Filesystem navigation and manipulation
- Command line text editors like vim or nano

- Common tooling like curl, wget and ping
- Familiarity with YAML and JSON notation
- Basic familiarity with common software development patterns and tools like version control, testing, continuous integration and logging.