ComponentSoft
open cloud trainings and consultation

IST-102: Service Mesh Essentials and Implementation with Istio

Course Length: 2 days

Course Description: Moving applications to the cloud brings new challenges for both developers and operators. Developers will use microservices to build their applications, while operators will need to manage large numbers of microservices deployments. Istio will reduce the complexity of such microservices by providing a service mesh that can transparently integrate into existing distributed applications and provides a way to securely connect and monitor microservices. In this workshop we will first discuss the architecture of Istio, then we will install it into an existing Kubernetes cluster. Once we have Istio in place, we will deploy a sample application and will have a look at some of the functionalities of Istio.

Structure: 50% theory 50% hands on lab exercises

Target audience: System administrators, developers and DevOps who want to understand the basic concepts and features of service meshes generally and Istio specifically as well as to gain some basic hands-on experience with it.

Prerequisites: Proficiency with the Linux CLI. Basic understanding of Kubernetes administration. Basic knowledge of Linux containers, e.g. Docker. Basic understanding of networking.

Module 1: Introduction to Istio

- Service mesh
- What is Istio?
- Istio features
- Platform support
- Istio architecture
- Istio architecture
- Istio architecture – Data plane
- Istio Architecture - Control plane
- Istio’s design goals

Module 2: Installing Istio in Kubernetes

- Installing Istio on Kubernetes
- Injecting the sidecar container
- Lab:Installing Istio on Kubernetes

Module 3: Istio Traffic Management

- Traffic management
- Traffic routing concepts
- Request routing
- Discovery and load balancing
- Handling failures
- Rule configuration
- Virtual Services

- Virtual Services - examples
- Virtual Services – Timeouts and retries
- Virtual Services – Injecting faults
- Destination rules
- Destination rules – Circuit breakers
- Service entries
- Service entries - example
- Gateways
- Gateways - example
- Sidecars
- Lab:Traffic management

Module 4: **Security**

- Security
- Security - architecture
- Security – Identity
- Service Authentication
- Authentication policies - scope
- Authentication policies (cont)
- Authorization
- Service roles
- Service role binding
- Lab:Security

Module 5: **Policies and telemetry**

- Policies and telemetry
- Reliability and latency
- Attributes
- Configuration model
- Handlers
- Instances
- Rules
- Policies and telemetry