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: Understanding of Kubernetes administration. Basic knowledge of Linux containers, e.g. Docker. Basic understanding of networking.

Module 1: Introduction

- Service mesh introduction
- Istio introduction
- Istio's architecture
- Installing Istio on Kubernetes

Module 2: Traffic management

- Traffic routing concepts
- Requiest routing
- Discovery and load balancing
- Handling failures
- Traffic management configuration resources

Module 3: Security

- Security needs of microservices based applications
- Security architecture of Istio
- Identites in Istio
- Authentication policies
- Authorization policies

Module 4: Observability

- Observability features of Istio
- Metrics
- Collecting and visualising metrics with Prometheus and Grafana
- Logging in Istio
- Distributed tracing with Jaegr
- Mesh management with Kiali

Module 5: Maintenance and troubleshooting

- Upgrading Istio with istioctl
- Analyzing mesh configuration
- Observing the status of the proxies
- Observing the configuration of the proxies
- Troubleshooting pod's status in the mesh
- Troubleshooting services in the mesh