**Microservices with Kubernetes**

**Duration: 5 Days**

Day 1

Getting Started with Microservices

* Technical requirements
* Programming in the small – less is more
* Making your microservice autonomous
* Employing interfaces and contracts
* Exposing your service via APIs
* Using client libraries
* Managing dependencies
* Coordinating microservices
* Taking advantage of ownership
* Understanding Conway's law
* Troubleshooting across multiple services
* Utilizing shared service libraries
* Choosing a source control strategy
* Creating a data strategy

Introducing Continuous Delivery

* Understanding CD
* The automated deployment pipeline
* Prerequisites to CD
* Building the CD process
* Creating a complete CD system

Introducing Docker

* Technical requirements
* What is Docker?
* Installing Docker
* Running Docker hello world
* Docker applications
* Building images
* Docker container states
* Docker networking
* Using Docker volumes
* Using names in Docker
* Docker cleanup
* Docker commands overview

Day 2

Configuring Jenkins

* Technical requirements
* What is Jenkins?
* Installing Jenkins
* Jenkins Hello World
* Jenkins architecture
* Configuring agents
* Custom Jenkins images
* Configuration and management

Continuous Integration Pipeline

* Technical requirements
* Introducing pipelines
* The commit pipeline
* Code-quality stages
* Triggers and notifications
* Team development strategies

Clustering with Kubernetes

* Technical requirements
* Server clustering
* Introducing Kubernetes
* Advanced Kubernetes
* Application dependencies
* Scaling Jenkins
* Alternative cluster management systems

Day 3

Securing Microservices on Kubernetes

* Technical requirements
* Applying sound security principles
* Differentiating between user accounts and service accounts
* Managing secrets with Kubernetes
* Managing permissions with RBAC
* Controlling access with authentication, authorization, and admission
* Hardening your Kubernetes cluster using security best practices

Configuration Management with Ansible

* Technical requirements
* Introducing configuration management
* Installing Ansible
* Using Ansible
* Deployment with Ansible
* Ansible with Docker and Kubernetes

Continuous Delivery Pipeline

* Technical requirements
* Environments and infrastructure
* Nonfunctional testing
* Application versioning
* Completing the Continuous Delivery pipeline

Day 4

Advanced Continuous Delivery

* Technical requirements
* Managing database changes
* Pipeline patterns
* Release patterns
* Working with legacy systems

Talking to the World - APIs and Load Balancers

* Technical requirements
* Getting familiar with Kubernetes services
* East-west versus north-south communication
* Understanding ingress and load balancing
* Providing and consuming a public REST API
* Providing and consuming an internal gRPC API
* Sending and receiving events via a message queue
* Understanding service meshes

Working with Stateful Services

* Technical requirements
* Abstracting storage
* Storing data outside your Kubernetes cluster
* Storing data inside your cluster with StatefulSets
* Achieving high performance with local storage
* Using relational databases in Kubernetes
* Using non-relational data stores in Kubernetes

Running Serverless Tasks on Kubernetes

* Technical requirements
* Serverless in the cloud
* Link checking with Delinkcious
* Serverless link checking with Nuclio
* Other Kubernetes serverless frameworks

Testing Microservices

* Technical requirements
* Unit testing
* Integration testing
* Local testing with Kubernetes
* Isolating tests
* End-to-end testing
* Managing test data

Day 5

Monitoring, Logging, and Metrics

* Technical requirements
* Self-healing with Kubernetes
* Autoscaling a Kubernetes cluster
* Provisioning resources with Kubernetes
* Getting performance right
* Logging
* Collecting metrics on Kubernetes
* Alerting
* Distributed tracing

Service Mesh - Working with Istio

* Technical requirements
* What is a service mesh?
* What does Istio bring to the table?
* Delinkcious on Istio
* Alternatives to Istio

The Future of Microservices and Kubernetes

* The future of microservices
* The future of Kubernetes