Course Outline

Microservices Bootcamp

Course Summary

Description

In this course, you'll learn about the fundamentals of microservices with the help of several examples and many hands on labs.

Next, you will look at how to handle practical scenarios that will arise in microservices development, use the microservices architecture to see how to translate your learnings into practice and how Docker and Kubernetes can help in transforming your microservices use case to an Internet-scale deployment. And to finalize, you'll examine the relationship between microservices and DevOps and the microservices development life cycle.

Topics

- Demystifying Microservices
- Applying Microservices Concepts
- Microservices Evolution
- Docker
- Understanding the Docker Model
- The Container Network Model
- The Docker Development Life Cycle
- The Microservices Development Life Cycle
- The Kubernetes Network Model
- Setting Up Kubernetes
- Microservices Healthchecks
- Namespaces and kubectl proxy
- The Microservices Security Model
- Volumes and Microservices
- Stateful Microservices
- The Microservices Logging and Metrics Model
- Extending Microservices APIs
- The Microservices Cluster
- Putting it all Together

Audience

This course is designed for anyone interested in learning how to program Microservices or manage a team new to Microservices.

Prerequisites

Basic analytic or programming skills.

Duration

Five days

Course Outline

Microservices Bootcamp

Course Outline

I. Demystifying Microservices
- A. The evolution of microservices
- B. What are microservices?
- C. Characteristics of microservices
- D. Microservices examples
- E. Microservices benefits
- F. Relationship with other architecture styles
- G. Microservice use cases

II. Applying Microservices Concepts
- A. Patterns and common design decisions
- B. Microservices challenges
- C. The microservices capability model

III. Microservices Evolution
- A. Reviewing the microservices capability model
- B. Understanding the PSS application
- C. Death of the monolith
- D. Microservices to the rescue
- E. The business case
- F. Plan the evolution
- G. Target architecture

IV. Docker
- A. Docker 30,000ft overview
- B. Our training environment
- C. Installing Docker
- D. Our first containers
- E. Background containers
- F. Restarting and attaching to containers

V. Understanding the Docker Model
- A. Understanding Docker images
- B. Building images interactively
- C. Building Docker images with a Dockerfile
- D. CMD and ENTRYPOINT
- E. Copying files during the build
- F. Multi-stage builds
- G. Publishing images to the Docker Hub
- H. Tips for efficient Dockerfiles

VI. The Container Network Model
- A. Naming and inspecting containers
- B. Container networking basics
- C. Container network drivers
- D. The Container Network Model
- E. Service discovery with containers
- F. Ambassadors

VII. The Docker Development Life Cycle
- A. Local development workflow with Docker
- B. Working with volumes
- C. Compose for development stacks
- D. Advanced Dockerfiles
- E. Tying it all together with CI/CD and continuous integration patterns

VIII. The Microservices Development Life Cycle
- A. Our sample application
- B. Kubernetes concepts
- C. First contact with kubectl
- D. Running our first containers on Kubernetes
- E. Accessing logs from the CLI
- F. Declarative vs imperative

IX. The Kubernetes Network Model
- A. Kubernetes network model
- B. Exposing containers
- C. Shipping images with a registry
- D. Running our application on Kubernetes
- E. Deploying with YAML

X. Setting Up Kubernetes
- A. Setting up Kubernetes
- B. The Kubernetes dashboard
- C. Security implications of kubectl apply
- D. Scaling our demo app
- E. Daemon sets
- F. Labels and selectors
- G. Authoring YAML

XI. Microservices Healthchecks
- A. Rolling updates
- B. Healthchecks
- C. Recording deployment actions

Course Outline

Microservices Bootcamp

Course Summary (cont'd)

XII. Namespaces and kubectl proxy

A. Namespaces
B. Accessing the API with kubectl proxy
C. Controlling a Kubernetes cluster remotely
D. Accessing internal services
E. Exposing Microservices with Ingress
F. Exposing HTTP services with Ingress resources
G. Kustomize
H. Managing stacks with Helm
I. Creating Helm charts

XIII. The Microservices Security Model

A. Network policies
B. Authentication and authorization
C. Pod Security Policies
D. The CSR API
E. OpenID Connect
F. Securing the control plane

XIV. Volumes and Microservices

A. Volumes
B. Building images with the Docker Engine
C. Building images with Kaniko

XV. Stateful Microservices

A. Managing configuration
B. Stateful sets
C. Running a Consul cluster
D. Local Persistent Volumes
E. Highly available Persistent Volumes

XVI. The Microservices Logging and Metrics Model

A. Centralized logging
B. Collecting metrics with Prometheus
C. Resource Limits
D. Defining min, max, and default resources
E. Namespace quotas
F. Limiting resources in practice
G. Checking pod and node resource usage
H. Cluster sizing
I. The Horizontal Pod Autoscaler

XVII. Extending Microservices APIs

A. Extending the Kubernetes API
B. Operators
C. Owners and dependents

XVIII. The Microservices Cluster

A. Building our own cluster
B. Adding nodes to the cluster
C. The Container Network Interface
D. Interconnecting clusters
E. API server availability
F. Static pods
G. Upgrading clusters
H. Backing up clusters
I. The Cloud Controller Manager
J. Git-based workflows

XIX. Putting it all Together

A. DevOps
B. Developing Microservices
C. Microservices, DevOps, and cloud
D. Practice points for microservices development
E. More real world use cases and case studie