Cloud - Introduction to Microservices Architecture Training

Code:	WA2755
Length:	2 days
URL:	View Online

This training course will help the attendees understand the value proposition and technical aspects of microservices. You will learn about the pros and cons of breaking up the monolithic type of applications prevalent in the enterprise space and converting them into microservices-based solutions. Detailed analysis of some of the common patterns and motivation for using them in microservices architecture is also provided.

Who Can Benefit

IT Architects, Software Designers, Developers.

Prerequisites

Foundational knowledge of programming and software design principles.

Course Details

Outline of Introduction to Microservices Architecture Training

Chapter 1. Breaking Up Monoliths – Pros and Cons

- Traditional Monolithic Applications and Their Place
- Disadvantages of Monoliths
- Developer's Woes
- Architecture Modernization
- Architecture Modernization Challenges
- Microservices Architecture is Not a Silver Bullet!
- What May Help?
- In-Class Discussion
- Summary

Chapter 2. Microservices

- What is a "Microservice"?
- Unix Analogy
- Principles of Microservices
- Services within an SOA vs Microservices

- Properties and Attributes of Microservices
- Benefits of Using Microservices
- The Two-Pizza Teams
- Beware of Microservices Cons
- Anti-Pattern: Nanoservices
- The Twelve-Factor App Methodology
- The Select Factors
- Serverless Computing
- Microservices – Operational Aspects
- Summary

Chapter 3. Microservices Architecture Defined

- The Microservices Architecture
- SOA Promises and Expectations
- Microservices Architecture vs SOA
- The ESB Connection
- Microservices Architecture Benefits
- Microservices Architecture Choices and Attributes
- Example: On-Line Banking Solution Based on MsA
- Distributed Computing Challenges
- Replaceable Component Architecture
- The Actor Model
- MapReduce Distributed Computing Framework
- Hadoop's MapReduce Word Count Job Example
- What Can Make a Microservices Architecture Brittle?
- 4+1 Architectural View Model
- Summary

Chapter 4. Containerization Systems for Microservices

- Infrastructure as Code
- Why Not Just Deploy My Code Manually?
- What is Docker
- Docker Containers vs Traditional Virtualization
- Docker is a Platform-as-a-Service
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Container Registries
- Your Own Docker Image Registry
- Starting, Inspecting, and Stopping Docker Containers
- One Process per Container

- Asynchronous Message Consumption Patterns
- Popular Messaging Systems
- Challenges of Managing Microservices
- Options for Managing Microservices
- In-Class Discussion
- Summary

Chapter 9. Working with Data in Microservices

- Monolithic Databases
- The Traditional Two-phase Commit (2PC) Protocol
- Table Sharding and Partitioning
- The CAP Theorem
- Mechanisms to Guarantee a Single CAP Property
- The CAP Triangle
- Eventual Consistency
- Handling Transactions in Microservices Architecture
- The Event-Driven Data Sharing Diagram
- The Saga Pattern
- The Saga Log and Execution Coordinator
- The Saga Happy Path
- A Saga Compensatory Request Example
- In-Class Discussion
- The Need for Micro Databases
- Migrating Data from Existing Databases (Breaking up the Monolith Database)
- One Data Migration Approach
- One Data Migration Approach (Cont'd)
- In-Class Discussion
- Command Query Responsibility Segregation (CQRS)
- The CQRS Communications Diagram
- A Word of Caution
- The Event Sourcing Pattern
- Event Sourcing Example
- Applying Efficiencies to Event Sourcing
- Summary

Chapter 10. Robust Microservices

- What Can Make a Microservices Architecture Brittle?
- Making it Resilient – Mechanisms
- Techniques and Patterns for Making Your Microservices Robust
- Fail Fast or Quiesce?

- Synchronous Communication Timeouts / Retries
- Asynchronous Communication Timeouts / Retries
- In-Class Discussion
- The Circuit Breaker Pattern
- The Circuit Breaker Pattern Diagram
- The Bulkhead Pattern
- Factor IX of the 12 App Methodology
- Feature Enablement
- Designing for Test and Failure
- Making Microservices Testable
- Test for Failure
- Continuous Testing and Integration
- Continuous Release and Deployment
- SLAs
- Where and What to Monitor
- Logging and Monitoring
- Summary

Lab Exercises

- Lab 1. Monolith vs Microservices Design
- Lab 2. Using Databases with Microservices
- Lab 3. The Event Sourcing Pattern

Schedule (as of 4)

Date	Location