Document Generated: 12/17/2025

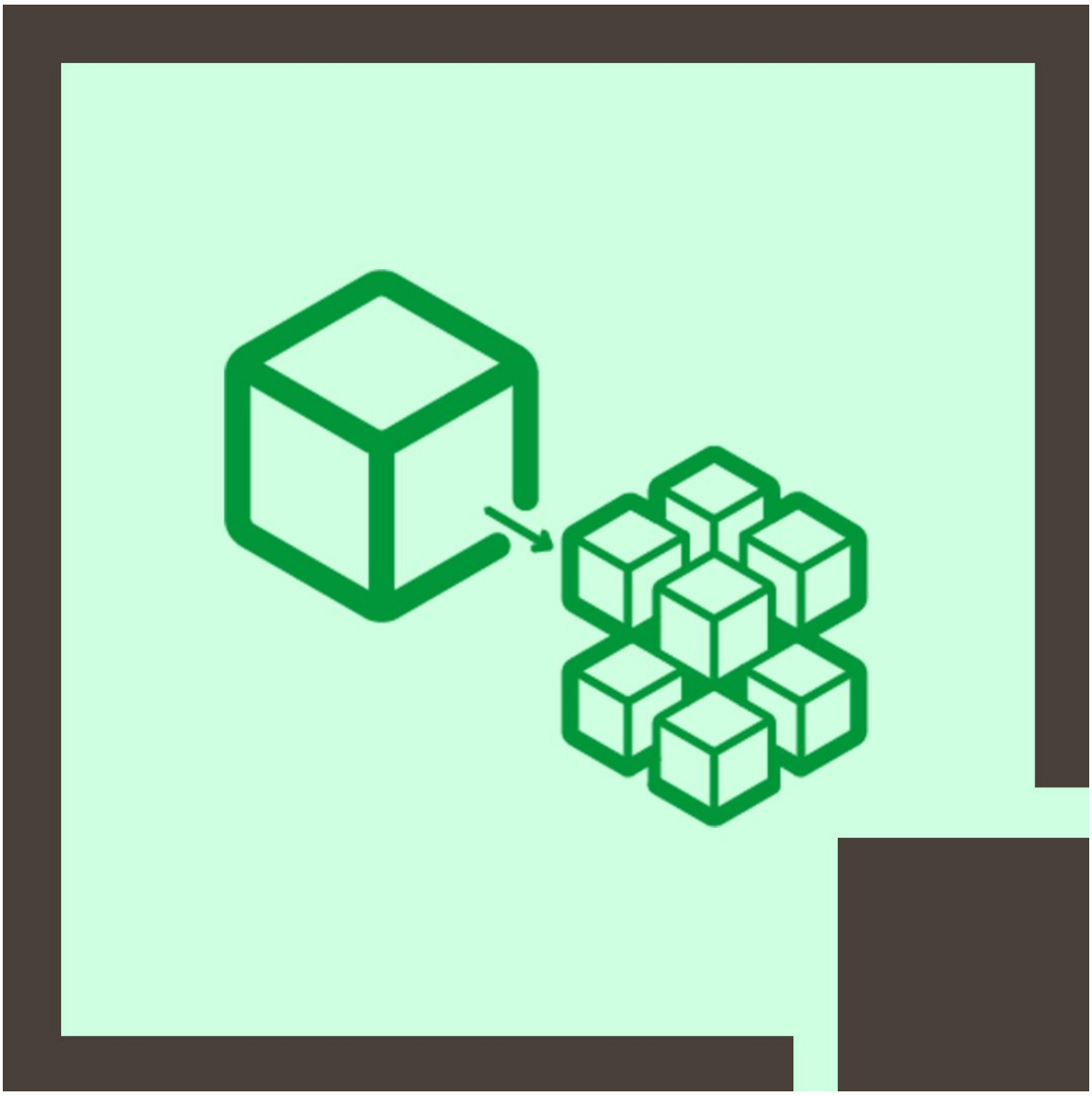
Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 2 Days

Understanding MicroServices: A Technical Overview (TT7050)



About this course:

Microservices understanding is a two-day training course of technical overview services intended for project stakeholders and managers who need to comprehend the effect of Microservices, what is distinctive about the direction of microservice and the general procedure of implementation and adoption. This course aims around what isolates an ad hoc set of services from a reusable, vibrant, managed, catalog of microservices. It gives an outline of the whole range from the guarantee of shattering monolithic computing to the coarseness of changing information. The course shows a perfect from of how a microservice direction can in a general sense change the elements of how programming is created and "lives" inside an enterprise.

Businesses are struggling with how to decipher the DevOps / microservice / Continuous Deployment vision into handy ideas and terms. Getting Microservices: A Technical Overview is intended for big business chiefs and partners who wish to comprehend what Microservices and a microservice-direction are and how actualizing Microservices will affect their tasks.

While our preparation is "innovation-driven", it is merchant independent, implying that the substance isn't clouded by marketing and sales messages. Though a particular item can be talked about, the detailed exercises are outfitted towards showing the useful application of microservices, as opposed to concentrating on the better purposes of the devices being used. Obviously, microservices isn't innovation but a set of concepts and practices that hold the guarantee of all the more intently adjusting business to IT applications.

Salary Estimate:

The normal compensation of a Java Developer is $90,992 annually.

Course Objective:

Microservices proceeds with the development of applications and frameworks being decayed into littler segments that work together to achieve bigger business services and tasks. Microservices are the autonomous, single purpose, and independently

deployable.

This course gives a strong comprehension of what microservices bring to the bench and the way to encourage the deployment, implementation, and management of microservices.

Learners are taken on an inside journey through the fundamental concepts of microservices and also how they identify with SOA and other styles of architecture. They will analyze the difficulties related to microservices and also how to recognize and plan them. The course at that point moves into the building and designing REST-based administrations and working those administrations utilizing different parts. The course tends to extra difficulties, for example, securing, monitoring, scaling, and containerizing microservices.

Working in an interactive, dynamic demo environment and discussion, driven by our accomplished SOA proficient, understudies will have the option to:

1. Clarify the issues related to monolithic, huge applications and how microservices present a chance to address huge numbers of those issues.
2. Comprehend the connections between SOA and microservices, SOAP administrations, and other current activities.
3. Apply RESTful microservices that take benefits of containers and the definitive idea of collecting basic segments into executable substances.
4. Work with executed microservices to deploy, package, and manage them.
5. Work and understand with the Cloud to scale microservices
6. Containerized microservices utilizing Docker
7. Use and recognize best practices relative to working and designing with microservices.

Targeted Audience:

This is a review level services instructional class, intended for individuals who need to comprehend and oversee existing or up and coming microservice ventures. Involvement in managing and working with big business applications will be useful. We will explore the specification, the terminology, the technologies and processes specific to microservices.

Prerequisite:

Participants ought to have at least 2 years of working information in the industry of IT. A fundamental comprehension of web-based applications and software development is important. Genuine improvement working information is useful but not essential.

Course Outline:

Module 1: Microservices Overview

Lesson: Monolithic Versus Micro

- Monolithic Applications
- Deploying and Scaling Monolithic Applications
- Microservices Described
- Deploying Microservices
- Scaling Microservices
- Benefits of Microservices
- Implementing Microservices

Lesson: Microservices: The Business Proposition

- Dealing with Change
- Leveraging business processes
- Challenges to adoption
- The Service Analysis and Design Process
- Relating Microservices to Agile Practices
- DevOps Defined
- Relating Microservices to DevOps
- Continuous Deployment Defined
- Relating Microservices to Continuous Deployment

Lesson: Designing Microservices

- Microservice Boundaries
- Size of Deployable Unit
- Communication Patterns
- Microservice endpoints
- Data Stores and Transaction Boundaries
- Challenges with Microservices

Module 2: RESTful Microservices in Spring

Lesson: Overview of REST

- REpresentational State Transfer
- REST Characteristics
- REST Elements
- REST Architectural Principles
- REST and HTTP
- REST/HTTP: Representation-Oriented
- REST Design Principles

Lesson: Designing RESTful Microservices

- Effectively Designing RESTful Services
- Best Practices for Endpoint Definition
- Using Query Parameters
- Working with HTTP GET and DELETE
- Working with HTTP PUT
- Working with HTTP POST
- Best Practices for HTTP Methods
- Handling Additional Operations

Module 3: The Microservices Ecosystem

Lesson: Working with Microservices

- Typical Microservices Stack
- Monitoring Microservices
- Logging
- Containerizing with Docker
- Deploying into Docker
- Orchestration of Microservices

Lesson: Microservice Best Practices

- Motivation and Mindset
- Minimum Viable Product
- Challenges of Data and Data Islands
- Spring Data and Microservices
- PrePersist, PreUpdate, and Repository Interface
- A DevOps-Style Microservice Life Cycle
- Continuous Delivery Pipeline
- Governance
- Tracking APIs and API Consumers

Lesson: Microservice Patterns

- Aggregator Pattern
- Branch Pattern
- Proxy Pattern

- Chained Pattern
- Circuit Breaker/Bulkhead Isolation Pattern
- Continuous Integration/Delivery Pattern
- Shared Resources as an Anti-Pattern
- Shared Resources as a Pattern
- Async Messaging Pattern

Lesson: Microservice Anti-Patterns and Challenges

- Microservice Costs
- When to Apply and NOT Apply
- Data Islands
- Dependency management
- Cohesion Creep
- Avoiding Versioning