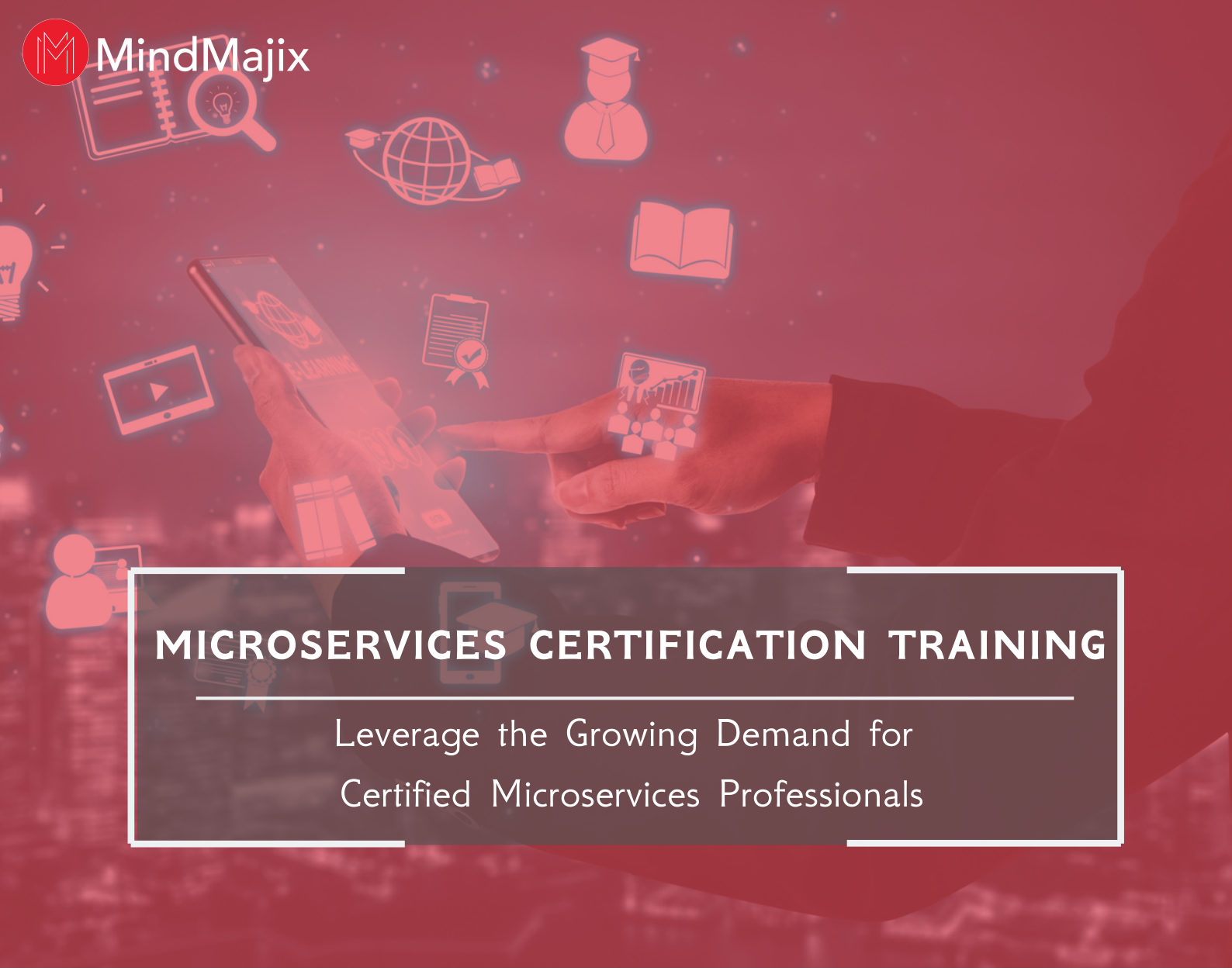
MindMajix



MICROSERVICES CERTIFICATION TRAINING

Leverage the Growing Demand for
Certified Microservices Professionals

About Microservices

Microservices is the architectural trend in the software industry that structures application as a collection of services. Top organizations like Amazon, Uber, Netflix, eBay, and more also use microservices. Through this course, you'll learn all the fundamental principles and practices that will let you successfully architect, build, and deliver microservice applications that are flexible, scalable, resilient, and secure.

Microservices Certification Course
Key Features

About Mindmajix

MindMajix is an edTech company offering quality training to professionals and corporates seeking skill upgrade. With more than 500 technology trainings to our LMS, we aim to deliver an industry-demand curriculum through live training and self-paced videos, Training Materials, Resume formats, and On Job Support. We have trained close to 500k professionals and more than 100 corporates winning impressive feedback.

Course Overview

Mindmajix Microservices training enables you to master all the key fundamentals and advanced features of Microservices with real-time examples. You will gain in-depth knowledge of architecture design principles and tools to implement and manage Microservices based applications through hands-on projects, and this will help you to clear the Microservices certification exam.

1. Microservices Evolution

1. Monolithic Architecture
   - Traditional Monolithic Applications and Their Place
   - Disadvantages of Monoliths
   - Architecture Modernization
   - Architecture Modernization Challenges
   - Summary
2. Service oriented Architecture
   - Understanding SOA
3. Defining entities of Microservices
4. Microservices in nutshell
   - What are Microservices?
   - Microservices v/s Classic SOA
   - Principles of Microservices Architecture Design
   - Domain-Driven Design
   - Domain-Driven Design – Benefits
   - Microservices and Domain-Driven Design
   - Summary

2. MicroServices Architecture

- REST Architecture principles
- Microservices Characteristics
- Designing for failure
- Inter-Process Communications
- Microservices Transaction Management
- Microservices Architecture – Pros
- Microservices Architecture – Cons
- Summary

3. Design of Micro Services

- Domain-Driven Design
- Untangling the Ball of MUD
- Big Mud Ball to Sweet Gems
- Kill the MUD Ball growth
- Repackaging/Refactoring
- Decouple the User interface and BackendBusiness Logic
- MUD Ball to Services
- Microservices Design Patterns
- Microservices Architecture Decisions
- Summary

4. Spring Boot and Microservices

- What is the Spring Boot?
- Spring Boot Main Features
- Understanding Java Annotations
- Lombok library
- REST Services With Spring MVC
- Spring MVC @RequestMapping with REST
- Working With the Request Body and Response Body
- @RestController AnnotationImplementing JAX-RS Services and
- Spring
- JAX-RS Annotations
- Java Clients Using RestTemplate
- RestTemplate Methods
- Spring MVC Annotations
- Spring MVC-based RESTful Web Service
- Spring Booting Your RESTful Web Service
- Spring boot built-in servers
- Setting up the root project
- Returning JSON entity as a response
- Spring Boot dev tools

- Accessing an SQL database
- Spring Data Rest and HATEOAS
- Converting a Spring Boot Application to a WAR File
- Running spring boot application on external Tomcat/Jboss
- Summary

5. RESTful Services

- Many Flavours of Services
- Understanding REST
- Principles of RESTful Services
- SOAP Equivalent Examples
- REST Example – JSON
- Famous RESTful Services
- Additional Resources
- Protocol Buffers
- Protobuf vs. JSONHTTP/2 vs. HTTP 1.1
- HTTP/2 vs. HTTP 1.1 (Contd.)
- Messages vs. Resources and Verbs
- REST Resource Examples
- REST v/s SOAP
- REST Example – Create
- REST Example – Retrieve
- REST Example – Update
- REST Example – Delete
- REST Example – Client Generated ID
- Summary

6. Security in Microservices

- Securing Web Applications with Spring Security 3.0
- Spring Security 3.0
- Authentication and Authorization
- Programmatic v Declarative Security
- Spring Security Configuration

- Spring Security Configuration Example
- Authentication Manager
- Using Database User Authentication
- LDAP Authentication
- What is Security Assertion Markup Language (SAML)?
- What is a SAML Provider?
- Spring SAML2.0 Web SSO Authentication
- Setting Up an SSO Provider
- Adding SAML Dependencies to a Project
- Dealing with the State
- How Can I Maintain State?
- SAML vs. OAuth2
- OAuth2 Overview
- OAuth – Facebook Sample Flow
- OAuth Versions
- OAuth2 Components
- OAuth2 – End Points
- OAuth2 – Tokens
- OAuth – Grants
- Authenticating Against an OAuth2 API
- OAuth2 using Spring Boot – Dependencies
- OAuth2 using Spring Boot – application.yml
- OAuth2 using Spring Boot – Main Class
- OAuth2 using Spring Boot – SPA Client
- JSON Web Tokens
- JSON Web Token Architecture
- How JWT Works
- JWT Header
- JWT Payload
- JWT Example Payload
- JWT Example Signature
- How JWT Tokens are Used
- Adding JWT to HTTP Header

- How The Server Makes Use of JWT Tokens
- What is “Scopes”?
- JWT with Spring Boot – Dependencies
- JWT with Spring Boot – Main Class
- Open ID Connect (OIDC)
- Summary

7. Testing of Micro Services

- Testing Microservices with Postman
- Testing Microservices with Swagger
- The Java rest service client
- The Angular HTTP Client
- Using The HTTP Client – Overview
- Importing HttpClientModule
- Simple Example
- Service Using HttpClient
- ES6 Import Statements
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- Error Handling
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request
- Summary

8. Reference Architecture of Micro Services

- Reading properties in various ways
- Implementing config server
- Setting up Discovery Server
- Setting up Discovery Client
- Overview of Actuator Endpoints
- API Gateway and Dynamic Routing

- IDeclarative Rest Client
- Hystrix Fault Tolerance
- Distributed Caching
- Distributed Sessions
- Need for Event Driven Systems
- Building Event Driven Systems
- Implementing Distributed Tracing
- Understanding Metrics
- Monitoring Microservices
- Spring Boot Admin
- Summary

9. Logs Analysis of Microservices

- Logging Challenges
- Leading Practices
- Correlate Requests with a Unique ID
- Include a Unique ID in the Response
- Send Logs to a Central Location
- Structure Your Log Data
- Add Context to Every Record
- Examples of Content
- Write Logs to Local Storage
- Collecting Logs with Fluentd
- Leading Practices for Microservices Logging Summary
- Metrics Using Prometheus
- Prometheus Architecture
- Service Discovery
- File-based Service Discovery
- Istio and Prometheus
- Exposing Metrics in Services
- Querying in Prometheus
- Summary

10. Design Patterns in Microservices

- Edge Proxy Server
- Request Handling
- Filters
- Filter Architecture
- API Gateway for Routing Requests
- API Gateway – Example
- Rate Limiting
- Rate Limiting – Business Cases
- Circuit Breaker
- Design Principles
- Cascading Failures
- Bulkhead Pattern
- Circuit Breaker Pattern
- Thread Pooling
- Request Caching
- Request Collapsing
- Fail-Fast
- Fallback
- Circuit Breaker / Fault tolerance Solutions
    1. Hystrix
    2. Resilience4j
    3. Hystrix v/s resilience4j
- Load Balancing in Microservice
    1. Server-side load balance
    2. Client-side Load Balance
- Service Mesh
- Service Mesh Solutions
- Content Delivery Network (CDN)
- How does a CDN Work?
- Benefits of using a CDN
- CDN Solutions
- Summary

11. Docker

- What is Docker?
- Where Can I Run Docker?
- Installing Docker Container Engine
- Docker Machine
- Docker and Containerization on Linux
- Linux Kernel Features: cgroups and namespaces
- The Docker-Linux Kernel Interfaces
- Docker Containers v/s Traditional Virtualization
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Competing Systems
- Docker Command Line
- Starting, Inspecting, and Stopping Docker Containers
- Docker Volume
- Dockerfile
- Docker Compose
- Using Docker Compose
- Dissecting docker-compose.yml
- Specifying services
- Dependencies between containers
- Injecting Environment variables
- runC Overview
- runC Features
- Using runC
- Running a Container using runC
- Summary

12. Jenkins

- Jenkins Continuous Integration
- Jenkins Features
- Running Jenkins
- Downloading and Installing Jenkins
- Running Jenkins as a Stand-Alone Application
- Running Jenkins on an Application Server
- Installing Jenkins as a Windows Service
- Different types of Jenkins jobs
- Configuring Source Code Management(SCM)
- Working with Subversion
- Working with Git
- Build Triggers
- Schedule Build Jobs
- Polling the SCM
- Maven Build Steps
- Jenkins Pipeline
- Jenkins Pipeline Output
- Installing Jenkins Plugins
- Summary

13. CI/CD pipeline for Microservices

- Creating Jenkins pipeline to deploy microservices on cloud

Projects Covered

Project 01 - Design a Web application for e-learning organization using MySql

To support read and write scalability using web technologies such as HTML, javascript,JSP, Servlet,Angular and Java

Project 02 - Design an e-commerce application ,customer browse the website to place orders for various brands and items

In this process, they must register themselves, sign in with a username and password, search for specific brands and items and place an order. While placing the order they can add/remove products from the shopping cart, discard the shopping cart , or proceed with the payment using various payment options.

Trusted By Companies Worldwide & 410K+ Learners

Successfully trained 120+ enterprises around the globe!

User Ratings on Microservices Course

(4.5) Google Ratings

(4.3) Trust Pilot Score

(4.5) Facebook Ratings

JOIN US !

Mindmajix Online Training Portal

Over 420K+ Happy Students  *Still growing*

https://mindmajix.com

Contact info

info@mindmajix.com

INDIA: +91-9246333245

USA : +1 917 456 8403