

Microservices Certification Training Course

Course Curriculum: Your 8 module Learning Plan

https://www.edureka.co/microservices-architecture-training

About Edureka

Edureka is a leading e-learning platform providing live instructor-led interactive online training. We cater to professionals and students across the globe in categories like Big Data & Hadoop, Business Analytics, NoSQL Databases, Java & Mobile Technologies, System Engineering, Project Management and Programming. We have an easy and affordable learning solution that is accessible to millions of learners. With our students spread across countries like the US, India, UK, Canada, Singapore, Australia, Middle East, Brazil and many others, we have built a community of over 1 million learners across the globe.

About Course

Edureka's Microservices course is designed with the help of top industry experts, making you proficient in designing Microservices applications, establishing communication between services, securing, monitoring, dockerizing, and deploying your services on Spring cloud with hands-on demonstration. Enroll now to become a Certified Microservices Developer with Edureka.

Microservices Training Curriculum

Introduction to Microservices

Goal: This module aims to Introduce Microservices.. Understand its basic working and equipping learners to be well versed in Microservices.

Objectives:

At the end of this module, you should be able to:

- Understand the need for Microservices
- Learn about Monolithic architecture
- Understand SOA
- Learn benefits and drawbacks of each architecture
- Know why there is a need to shift from Monolithic to Microservices
- Implement Microservices and Monolithic architecture codes

Topics:

- Introduction to Microservices
- Monolithic Architecture
- SOA Architecture
- Key benefits of Microservices
- Challenges in Microservices
- Comparisons between Monolithic, SOA and Microservices
- Microservices: Process and Organization
- Use Case #1 FTGO

Hands on:

• Tuscan website compares the architecture of both the applications, see how Microservice application is better than Monolithic

Microservices Design and Architecture

Goal: This module aims to learn about designing Microservices applications based on their specific architecture.

Objectives:

At the end of this module, you should be able to:

- Learn Microservices Architecture
- Understand Microservices Pattern
- Design applications with Microservice Architecture
- Understand IPC-Inter Process Communication

Topics:

- Introducing Microservices Architecture
- Microservices Design Patterns
- Use case: Apollo Store
- Decomposition Strategies
- Obstacles in Decomposition
- Inter-process communication
- Partial failure in communication
- Service Discovery
- Transaction management

Hands on:

• Design an E-commerce application with Microservice architecture

Introduction to Springboot and Spring framework - I

Goal: The aim of this module is to build a strong base in Spring framework and to build application using Springboot

Objectives:

At the end of this module, you should be able to:

- Understand Springboot
- Realize the different aspects of Spring boot
- Make use of Springboot in Eclipse
- Use various annotations in Spring boot
- Make use of databases in Springboot

Topics:

- Introduction to Springboot
- Comparisons between Spring, Springboot
- Working of Spring in Eclipse/any other IDE of your choice
- Necessary files in Springboot
- Annotations
- Spring boot database H2 and JPA

Hands on:

• Build a Course Management service with Spring boot using H2 In-memory database

Springboot and Spring framework - II

Goal: The focus of this module is to build a strong base in spring framework to build application using Springboot and understand how Spring boot integrates with other software and services

Objectives:

At the end of this module, you should be able to:

- Understand Springboot
- Use Springboot for AOP
- Make better applications on Web
- Cache your data from your application

Topics:

- Spring boot AOP
- Springboot caching
- Sprinboot database MySQL
- Springboot RESTful Web Services
- Swagger documentation format

Hands on:

• Build Transaction Management Service with Springboot

Microservices with Springboot and Cloud

Goal: The aim is to make the learner capable of performing and ready the projects for deploying on the Spring Cloud.

Objectives:

At the end of this module, you should be able to:

- Deploy applications on cloud
- Gain practical knowledge of Eureka
- Learn about Spring cloud features
- Configure Spring cloud

Topics:

• Introduction to Spring cloud

- Difference between Springboot and Spring cloud
- Spring Cloud features
- Spring Cloud and Netflix Eureka
- Registration and Discovery
- Spring Load Balancer
- Introducing Spring Cloud Gateway

Hands on:

• Deploy an Ordering and Payment Service application on Spring cloud

Microservices Security

Goal: The aim is to make the learner capable of using Security in their application and understand how the application performs.

Objectives:

At the end of this module, you should be able to:

- Make your applications secure
- Secure applications with different techniques

Topics:

- Introduction to Spring Security
- Microservices Security Principles
- Introduction to OAuth 2.0
- Authentication and Authorization
- Principal Security
- Springboot Security

Hands on:

• Secure microservices application

Docker with Microservices using Spring Boot - I

Goal: The aim of this module is to use Docker with your Microservice application.

Objectives:

At the end of this module, you should be able to:

- Build application using Docker Microservices
- Use Docker concepts in your microservices project
- Use Docker containers

Topics:

- Introduction of Docker with Microservices
- Introduction to linux
- Docker Basics
- Advantages of Docker
- Installation and commands
- Docker Hub and Desktop
- Docker hosts and machine

Hands on:

• Docker containers in Microservice application

Docker with Microservices using Spring Boot - II

Goal: The aim of this module is to use Docker with Microservices and make use of dockerised containers for Microservices application development

Objectives:

At the end of this module, you should be able to:

- Understand Docker in Spring Application
- Make Docker containers in Spring
- Using Docker in applications
- Learn about Docker compose

Topics:

- Introduction to Docker Images
- Docker containers
- Docker Images
- Docker Repositories
- Managing containers for Microservices
- Monitoring with Prometheus and Grafana

Hands on:

• Build a Microservice application using Docker

Microservices Course Project

What are the system requirements for this Microservices Training?

A System with an Intel i3 Processor or above, minimum 4 GB RAM (8 GB recommended) and an Operating System either of 32bit or 64bit with Java-11 (preferably) or Java – 8 installed

How will I execute the Practical?

Practicals can be executed on any IDE (eg. Eclipse, STS, Intellij) and applications are run on local server.

Which projects will be a part of this Microservices Certification Course?

The Microservices Certification Course project will test your ability to work with services. Following is the problem statement for the same:

- Spring boot 2.5.x version or higher
- Java 11
- Use In-Memory(H2) Database wherever applicable
- Use any IDE of your choice
- Use JSON for exchanging data through RESTful APIs
- For security, use OAuth2. Add customers to OAuth2 server(in-memory)
- Swagger Documentation
- Dockerize your services
- Focus should be on implementing several design patterns relevant in Microservices
- Keep the Data Model simple, eg. A book may have only one author
- Call out assumptions that you may make in implementation in terms of domain
- Use OAuth2 server to manage users(in-memory)



• Monitor your Microservices