



# Microservices Architecture: Building Enterprise-Grade Solutions

**Subtitle - Transform your messy monoliths into organized, maintainable Microservices to design and build Microservice architectures**

**Short subtitle - Migrate to Microservices with this easy-to-follow guide**

Dimos Raptis

---

## DURATION

3 hours 30 minutes

## OVERVIEW

All major companies are now evolving from monolithic to microservices architecture in their websites and applications due to its many advantages. Early adopters of microservices have been able to provide an unprecedented level of control and speed as they deliver innovative web experiences to customers and Netflix is a leading example.

The enterprise-grade application that we will build through the course will be highly maintainable and scalable. By exploring the intricacies of the Microservices architecture you will understand its advantages over the monolithic architecture and Service Oriented Architecture (SOA) principles. By going through multiple scenarios you would be taught where to use Microservices in the architectures of the applications. Advanced tasks with concepts such as Serverless computing and DevOps using AWS will also be undertaken to help you in your next venture with confidence.

By the end of the course you will be ready to migrate to Microservices in your organisation by building architecture using Microservices Architecture.

## TARGET AUDIENCE

This course will appeal to architects, designers & developers, who are primarily new to Microservices architecture and interested in migrating to it from a Monolith architecture.

## KEY FEATURES

1. Discover the multiple facets of Microservices Architecture using while building a Microservice framework from scratch
2. Successfully migrate from monolith to microservices by understanding the key success factors
3. Explore industry case-studies along with best-practices & lessons learnt in the journey of Microservices based development approach.

## APPROACH

This step-by -step course helps the customer understand the complexity behind basic tenets of Microservices Architecture, key architectural principles, key benefits of MSA, key foundational building blocks required for building in MSA and the role of organizational culture for the success of MSA building architecture using Microservices architecture.

## WHAT WILL YOU LEARN

1. Understand foundational elements behind Microservices Architecture
2. Gain knowledge on key tenets/elements required to build using MSA
3. Observe key industry examples, lessons learnt & pitfalls of MSA
4. Understand development & design practices/approaches behind Microservices Architecture
5. Explore non-functional requirements related to MSA such as resilience, availability, performance, security, scalability and maintainability.
6. Gain knowledge on role of DevOps in MSA particularly Continuous Integration & Deployment, importance of CI pipeline with large number of Microservices
7. Understand with examples whether Serverless Architecture is suitable in MSA

## ABOUT THE AUTHOR

Dimos is a Software Engineer with Computer Science foundations, coupled with several years of industrial experience, designing and implementing software systems in an agile mode. He especially loves challenges in the field of distributed and/or multi-threaded, concurrent systems. He is a big fan of agile

manifesto and XP practices. Whatever he is working on, he tries to do his best to make complex problems look simple through the use of Domain-Driven Design.

His technical expertise lies on the ecosystem of Java and Linux, having some hands-on experience with emergent open-source technologies. Also he has a soft spot for the fields of Big Data and Machine Learning.

## **SUMMARY OF CONTENTS**

**SECTION 1 - Diving into Microservices (30 mins)**

**SECTION 2 - Building our first Microservice (30 mins)**

**SECTION 3 - Integration of multiple Microservices (30 mins)**

**SECTION 4 - Scaling our Microservices Architecture (30 mins)**

**SECTION 5 - Serverless Architectures (30 mins)**

**SECTION 6 - Design Practices and Tips (30 mins)**

**SECTION 7 - Industry Examples (30 mins)**

## **COURSE ROADMAP**

**SECTION 1 - Diving into Microservices (30 mins)**

- Definition of Microservices (5 mins)
- Driving forces & Conway's law (7 mins)
- Coupling & Cohesion (7 mins)
- Domain-Driven Design (7 mins)
- Exploring our sample project (5 mins)

**SECTION 2 - Building our first Microservice (30 mins)**

- Building our first microservice
- Continuous Integration using AWS Code Build
- Automated deployment using AWS CodeDeploy
- Continuous Delivery using AWS Code Pipeline
- Best Practices for CI/CD

**SECTION 3 - Integration of multiple Microservices (30 mins)**

- Integration methods
- Adding a gRPC interface to our service
- Building our second async microservice
- Best Practices for Monitoring & Logging
- Monitoring using AWS Cloudwatch

#### **SECTION 4 - Scaling our Microservices Architecture (30 mins)**

- Scaling & load balancing
- Service Discovery
- Using AWS EC2 auto-scaling
- Service mesh VS API Gateway

#### **SECTION 5 - Serverless Architectures (30 mins)**

- Introduction to serverless computing
- Leveraging AWS Lambda
- Development lifecycle of serverless applications
- Caveats of serverless model

#### **SECTION 6 - Design Practices and Tips (30 mins)**

- Microservices Design & Boundaries
- Operations in a microservices environment
- Polyglot programming & persistence
- Moving from a monolith to microservices
- Considerations for a microservices architecture

#### **SECTION 7 - Industry Examples (30 mins)**

- Microservices at Netflix
- Microservices at Gilt

## **Setup and Installation**

### **Minimum Hardware Requirements**

For successful completion of this course, students will require the computer systems with at least the following:

- OS: Linux/MacOS/Microsoft Windows
- Processor: 2.0 Ghz

- Memory: 2 GB
- Storage: 20GB free space

## **Recommended Hardware Requirements**

For an optimal experience with hands-on labs and other practical activities, we recommend the following configuration:

- OS: Linux (Ubuntu 14.04 or later)
- Processor: 2.4 Ghz
- Memory: 4 GB
- Storage: 20GB free space

## **Software Requirements**

- Browser: Google chrome
- Code Editor: IntelliJ IDEA