**Master Microservices with Spring Boot, Docker, Kubernetes**

<https://capgemini.udemy.com/course/master-microservices-with-spring-docker-kubernetes/>

## Instructors

[**Eazy Bytes**](https://capgemini.udemy.com/user/552e9eb07495d/)[**Madan Reddy**](https://capgemini.udemy.com/user/challa-madan/)Training on Java, Full Stack, Spring & ORM Java Full stack | Cloud enthusiast | Learner | Instructor

**GitHub page**:  
<https://github.com/eazybytes/microservices>  
---------------------------------------------------------------------------------------------------------------------------------------------------------  
**Microservices**: These days many organizations migrating monolithic application into microservices, the reason is microservice architecture makes application **easier to scale, faster to develop, enable innovation and also helps to organizations to accelerating the time to take new enhancement to the market**. So, microservice is going to be a demanding skillset for new, few years.

**Course agenda**:

* Welcome to the world of microservices.
* Building microservices building logic using spring boot.
* How do we right size our microservices and identify boundaries.
* How to containerize our microservices using Docker.
* Configurations management in microservices using spring cloud config.
* Service discovery and service registration in microservices using eureka.
* Building an edge server for microservices using spring cloud gateway.
* Making microservices resilient using Resiliency4J patterns.
* Observability and monitoring of microservices using Grafana, Prometheus etc.
* Securing microservices using OAuth2/OpenID, spring security.
* Event driven microservices using RabbitMQ, spring cloud Functions and stream.
* Event driven microservices using Kafka, spring cloud functions and stream.
* Container orchestration using Kubernetes.
* Deep dive on Helm (Kubernetes package manager).
* Deploying microservices into cloud Kubernetes cluster.
* Many best practices, techniques followed by real time microservice developers.

**Important points:**\* Explanation of DTO pattern  
<https://martinfowler.com/eaaCatalog/dataTransferObject.html>

\* Open Api documentation  
<https://www.openapis.org/>

<https://springdoc.org/>

**<dependency>**

**<groupId>org.springdoc</groupId>**

**<artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>**

**<version>2.3.0</version>**

**</dependency>**

This dependency is only supports for spring boot greater than or equal to 3.0  
<http://localhost:8080/swagger-ui/index.html>

Using this URL, we can access the swagger UI