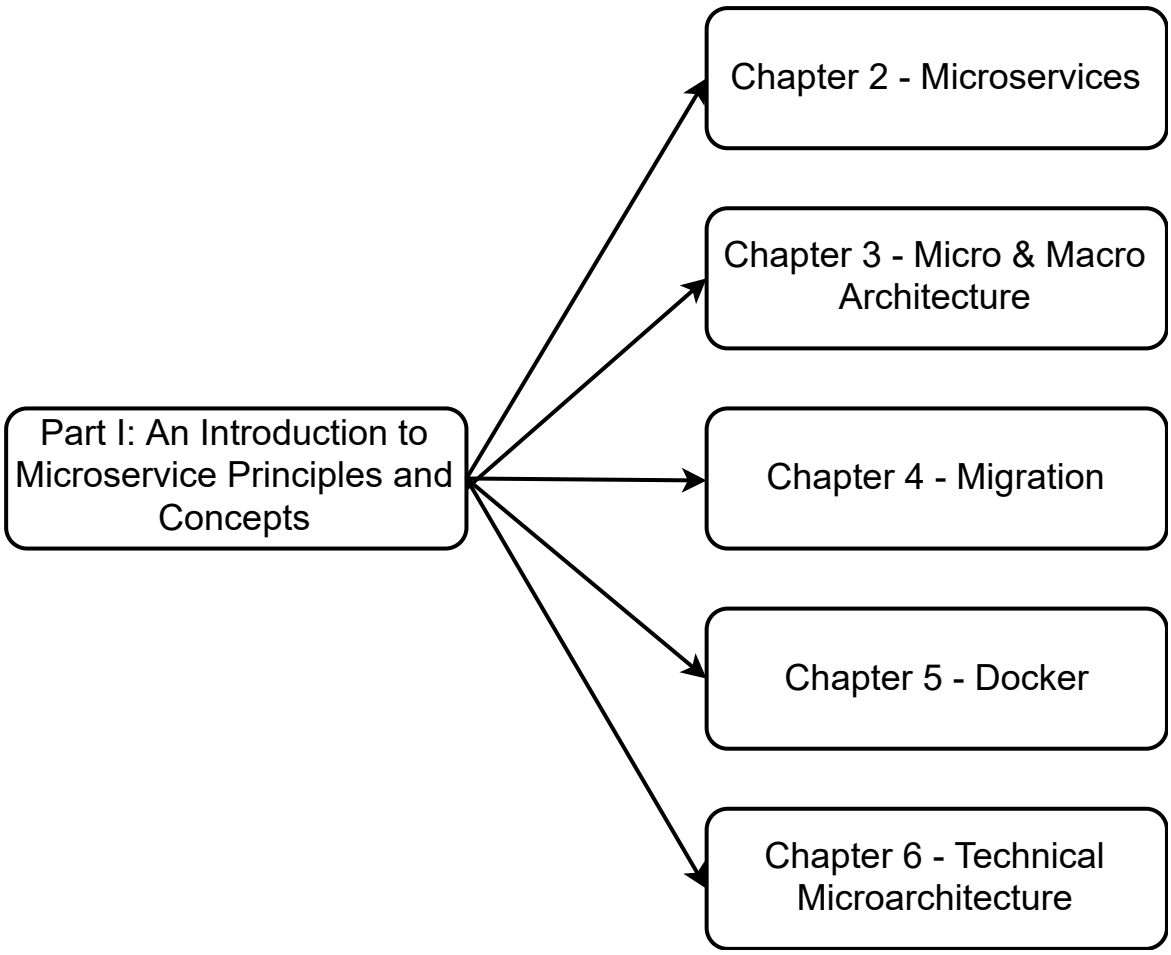
Structure of the Course

In this lesson, we'll go over the structure of this course!

WE'LL COVER THE FOLLOWING

Course structure

This course is **part I** of a series of courses on microservices to introduce the basic principles of microservices-based architecture and a few important technologies. It's one thing to define an architecture, and quite another to implement it. Hence, this course presents two technologies for the implementation of microservices and highlights the associated benefits and disadvantages.



Overview of this course

- Chapter 2 **defines the term *microservice***.
- Microservices architecture has two levels: **micro and macro architecture**. They represent global and local decisions as explained in chapter 3.
- Old systems are often supposed to be **migrated into microservices**, a topic covered in chapter 4.
- **Docker serves as the basis for many microservices architectures**. It facilitates the roll-out of software and the operation of the services and is discussed in chapter 5).
- The **technical micro architecture** describes technologies for implementing microservices and is looked at in (chapter 6) .

Target groups

This course explains basic principles and technical aspects of microservices.

Thus, it might be interesting for different audiences.

- For *developers*, this course explains the basic principles of architecture concepts.
- For *architects*, it contains fundamental knowledge about microservices.
- For experts in *DevOps* and *operations*, the recipes in this course provide background information about the concepts behind the microservices architecture approach.
- *Managers* are presented with an overview of the advantages and specific challenges of the microservices architecture approach.

Prior knowledge

This course assumes the reader has some **basic knowledge of software architecture and software development**. All practical examples are documented in such a way that they can be executed with **very little prior knowledge**. This course focuses on technologies that can be employed for microservices using different programming languages. However, the **examples are written in Java** using the Spring Boot and Spring Cloud frameworks so any changes to the code require knowledge of Java.

Quick start

This course focuses primarily on introducing microservices concepts. We will use an example e-commerce system throughout the course to illustrate these concepts.

Acknowledgements

I would like to thank everybody who discussed microservices with me, who inquired about them, or worked with me on this course. Unfortunately, these folks are far too numerous to name individually. The exchange of ideas is enormously helpful and also fun!

Many of the ideas and their implementation would not have been possible without my colleagues at INNOQ. I would especially like to thank Alexander Heusingfeld, Christian Stettler, Christine Koppelt, Daniel Westheide, Gerald

Preissler, Hanna Prinz, Jörg Müller, Lucas Dohmen, Marc Giersch, Michael

Simons, Michael Vitz, Philipp Neugebauer, Simon Kölsch, Sophie Kuna, Stefan Lauer, and Tammo van Lessen.

Also, Merten Driemeyer and Olcay Tümce provided important feedback.

Finally, I would like to thank my friends and family, whom I may have neglected while writing this course – especially my wife. She also did the translation into English.

Of course, my thanks goes out to the people who developed the technologies which I introduce in this course and thereby created the foundation for microservices.

I would also like to thank the developers of the tools of https://www.softcover.io/ and Leanpub.