

# Designing Microservice Architectures - Basic Info

## This course is for organizations who:

- are considering to adopt microservices,
- want understand and tackle the challenges associated with microservices,
- want to learn how to design an architecture to get the most out of their distributed system.

## Prerequisites

- Node.js knowledge is helpful but not necessary as the code examples are written in Node.

## Training length: 2 days

## This training is available in:

- [Tel-Aviv](#) (March 12-13)
- [Barcelona](#) (April 12-13)
- [Warsaw](#) (May 10-11)
- [Berlin](#) (June 14-15)

# Designing Microservice Architectures - Full Agenda

## Day One:

1. Microservice Architecture in the Company
2. Moving to Microservices
3. Decentralized Data Management
4. Architectural patterns in distributed systems

## Day Two:

5. Reliability in Microservices
6. Testing Microservices
7. Debugging and Monitoring Microservices

In case you have any questions regarding this training, or you'd like to invite our team to provide a training exclusively for your company, please reach out to us at [info@risingstack.com](mailto:info@risingstack.com)

# Designing Microservice Architectures - Day 1

## 1. Microservice Architecture in the Company

We will compare monolithic and microservice systems while taking a look at organizational changes needed to make the transition from the first to the latter. We'll also learn how to decide whether you should move or not in the first place. We'll introduce a monolithic application that we'll redesign into a set of small, independent processes for better scaling and faster development.

## 2. Moving to Microservices

You'll learn the evolutionary way of the migration from a monolithic system to the microservices architecture via the API gateway approach.

# Designing Microservice Architectures - Day 1

## **3. Decentralized Data Management**

We will dive into ways of handling data outside the ACID compliant world, using http and queues as a way of sharing data between your services while taking a look at common pitfalls.

## **4. Architectural patterns in distributed systems**

You'll learn about some patterns that may come useful later, such as creating common interfaces and the Data Pump model. We'll also talk about anti-patterns that may feel convenient but should be avoided.

# Designing Microservice Architectures - Day 2

## **5. Reliability in Microservices**

We will discuss how much reliability and errors can cost you and your company, and what you can do to mitigate the effects of inevitable failures or even profit from the fact that things can and will go wrong.

## **6. Testing Microservices**

You're going to learn the way to test an architecture where your logic lives on several different machines and about the cost of local development.

## **7. Debugging and Monitoring Microservices**

Logging in a distributed system is a lot more challenging than in a single service application. You need to introduce distributed tracing to track down errors that propagate through several HTTP calls.