



Building Microservices: Designing Fine-Grained Systems

Sam Newman

[Download now](#)

[Read Online ➔](#)

Building Microservices: Designing Fine-Grained Systems

Sam Newman

Building Microservices: Designing Fine-Grained Systems Sam Newman

Distributed systems have become more fine-grained in the past 10 years, shifting from code-heavy monolithic applications to smaller, self-contained microservices. But developing these systems brings its own set of headaches. With lots of examples and practical advice, this book takes a holistic view of the topics that system architects and administrators must consider when building, managing, and evolving microservice architectures.

Microservice technologies are moving quickly. Author Sam Newman provides you with a firm grounding in the concepts while diving into current solutions for modeling, integrating, testing, deploying, and monitoring your own autonomous services. You'll follow a fictional company throughout the book to learn how building a microservice architecture affects a single domain.

Discover how microservices allow you to align your system design with your organization's goals

Learn options for integrating a service with the rest of your system

Take an incremental approach when splitting monolithic codebases

Deploy individual microservices through continuous integration

Examine the complexities of testing and monitoring distributed services

Manage security with user-to-service and service-to-service models

Understand the challenges of scaling microservice architectures

Building Microservices: Designing Fine-Grained Systems Details

Date : Published February 20th 2015 by O'Reilly Media (first published December 25th 2014)

ISBN : 9781491950357

Author : Sam Newman

Format : Paperback 280 pages

Genre : Computer Science, Programming, Science, Technology, Technical, Software, Nonfiction



[Download Building Microservices: Designing Fine-Grained Systems ...pdf](#)



[Read Online Building Microservices: Designing Fine-Grained System ...pdf](#)

Download and Read Free Online Building Microservices: Designing Fine-Grained Systems Sam Newman

From Reader Review Building Microservices: Designing Fine-Grained Systems for online ebook

Andreas says

Microservices are a relatively new trend in computer science, coined around 2012. This conceptual book touches lots of aspects surrounding those little sisters of Service-oriented architectures (SOA): Starting from the basics, it covers topics like integration, splitting monoliths, deployment, testing, monitoring, security, system design and the role of architects, and scaling services.

The Good

Newman takes a holistic approach, analyzes the topic from lots of angles. It isn't a dry read but the author introduces a couple of real-life experiences leading to an enjoyable read. I found a couple of gems embedded in the text that I didn't know about, like references to Nagios, Hexagonal Architectures or Holistic Software Development, which I plan to read about later on.

The book is quite short with only 300 pages and will take only a couple of evenings to dig through. Each chapter ends with a nice summary.

The Bad

Given its size and the broad range of topics, it doesn't come as a surprise that each topic doesn't get a lot of screentime, leading to a very shallow discussion. This comes in handy, if you only want to get hints what could be thought about. In the end, the coverage isn't very helpful.

It is an egocentric text, stemming from thoughts of the author alone, it doesn't introduce community's knowledge or give any hints to further readings in this specific area. No references to articles, blogs, books covering more details with respect to Microservices.

The Ugly

The book is mainly about anything else but Microservices, roughly 20% are specific to the book's main topic. There are lots of chapters which don't or nearly don't talk about them at all. For example, there is a philosophical chapter about the role of architects without touching the topic at all. Other chapters are a nice read on computer science topics or service oriented architectures but don't cover specifics of Microservices. Although there are lots of hidden gems and references, the book doesn't contain a list of literature or a definition of terms. An overview book should give references to deeper analysis, but you won't find any here.

Architects who know their stuff, who haven't missed the last 15 years of SOA, should read the wikipedia article rather than this book. I don't recommend the book at all.

Robson Castilho says

Mixed feelings about this book. I had high expectations before starting reading it but after reading some chapters I've started feeling bored with so much shallow information about a lot of 'tooling stuff'.

The book is a big overview of a series of concepts and advices (and tooling!) you should care about in an environment of distributed systems such as integration, deployment, monitoring, security, scaling and the like. There's nothing practical in its content, which have disappointed me a lot (for a book named "Building" Microservices).

I don't recommend it if you're a beginner programmer or you have little knowledge about networking, cloud, monitoring and other 'operations' concepts and tooling. For programmers or architects working with this kind of environment, it may be useful as a guide for reviewing/improving the architecture.

If there's a true good thing to say about the book, it's the fact that it may help to show that playing with microservices (one of those 'buzzwords' in the last years) is definitely not a simple thing to do. There's a lot of practices that must be running for this type of architecture work.

For all of that, I think the most appropriate name for this book should be "Managing an environment of distributed systems" or something like that.

Bragadeesh says

Microservices is still very young and this book does a fair job in covering all the aspects of the intricacies that involve in adopting to it. I was expecting something different before I picked this book, however, most of the concepts elicited here are some of the known concepts to me as an Architect. It still has a lot gold mines spread around the entire book. I would not say this is a must-read for a Software Architect, but the learnings of the book will definitely add value to your thinking process in designing fault-tolerant and large scale systems. In any case, getting hands on and having a first person point of view is vital to really digest the things the Author talks about.

Brian says

The Pragmatic Programmer of the microservices age. So many lessons learned the hard way are documented here. A quick and information packed read.

Daniel says

If you are new to micro services, or service-oriented architectures in general, this books provides a good overview of all the things you need to take into account, which trade-offs have to be made etc. That said, it can't serve as more than a starting point – every single chapter deserves a book of its own.

Elena B. says

A book that is more about ideas behind microservices and fine grained systems than technology specifics.

The key principles for microservice architectures:

- You want services that are loosely coupled and highly cohesive - so find boundaries that help ensure that related behaviour is in one place and that communicate with other boundaries as loosely as possible.

- Avoid database integration
- Understand the trade-offs of REST and RPC, but strongly consider REST as a good starting point for request/response integration.
- Prefer choreography over orchestration.
- Avoid breaking changes and the need to version by understanding Postel's Law and using tolerant readers.
- Think of user interfaces as compositional layers.
- Focus on maintaining the ability to release one service independently from another - you need one CI build per microservice.
- If possible, move to a single-service per host/container.
- Automate everything.
- Optimize for fast feedback, and separate types of tests accordingly
- Avoid the need for end-to-end tests whenever possible by using consumer-driven contracts.
- Use consumer-driven contracts to provide focus points for conversation between teams.
- Understand the trade-off between putting more effort into testing and detecting issues faster in production (optimizing MTBF vs MTTR)
- It is preferable that teams aligned along bounded contexts
- An essential part of building a resilient system is the ability to safely degrade functionality
- Prepare for the sorts of failure that can happen with distributed architectures. Handle failure by using timeouts, circuit breakers, bulkheads and isolation
- Consider using blue/green or canary release techniques to separate deployment from release
- Use semantic monitoring to see if your system is behaving correctly, by injecting synthetic transactions into your system to simulate real-user behavior.
- Aggregate your logs, and aggregate your stats, so that when you see a problem you can drill down to the source.

I liked the cautionary tales and examples, as well as the idea of embracing the concept of evolutionary architecture, where your system bends and flexes and changes over time as you learn new things.

Change is inevitable. Embrace it.

Also, funny side remark:

The main thing I noticed, though, was that the hard drives were attached by velcro. I asked one of the Googlers why that was. “Oh,” he said, “the hard drives fail so much we don’t want them screwed in. We just rip them out, throw them in the bin, and velcro in a new one.”

Alexander says

I think this book is going to be useful for any engineer onboarding the world of microservices. It gives a broad and complete overview of all stages of the process: from design to deployment and testing.

On the other hand, from my perspective, for more experienced engineers already familiar with similar systems this book would rather look like a reiteration of all the difficulties and problems one might face while working with microservices. In many cases, the author agrees with the complexity of certain problems (e.g. database scaling/sharding, 2 phase commit, etc) but does not dive into details of how to approach these problems in real world. However, this is probably intentional, as many of these problems go far beyond a

single book.

In short: highly recommended to everybody dealing with the concepts of microservices for the first time ever.

Sunil rajashekhar says

I work in a firm which firmly believes that MicroServices are the way to move forward.

I had vague idea of why things are structured / built in certain way, it helped me to assert my understanding. The book touches most of the concepts very well with practical use cases and scenarios. I was able to appreciate and understand the rational for making design choices at my firm. The book will help you enhance your overall thinking process and you would be in a better choice to make some decisions around your services.

It covers topics like why and how , integration techniques, splitting monolith, testing, deployment, monitoring and how to scale [basic pointers].

Felt it could have been more elaborate on few topics, if you have started working on micro services and want to understand core principles it's a must read.

Líbene Fernandes says

"The need to change our systems to deal with scale isn't a sign of failure. It is a sign of success."

Yevgeniy Brikman says

If you're new to microservices, this book is a decent intro, covering most of the major topics you need to be aware of. It only covers the topics at a surface level (to be fair, it would've been a very long book if it went in-depth on each one), which is just enough to show you what questions you should be asking. Of course, you'll have to seek elsewhere to find those answers, but at least now you'll know what to look for.

However, the biggest weakness of the book is that it makes microservices seem like a good idea for just about everyone. Obviously, the author has a vested interest in this perception, since he wants to sell more books, but the reality is that microservices are NOT a good choice for many use cases. They make perfect sense at the scale of a company like Google or Facebook, but most companies do not face that kind of scale, and, even more importantly, most companies do not have the resources of a Google or Facebook to invest in making microservices usable.

Running microservices requires a massive investment in terms of orchestration, configuration management, automated deployment, build tooling, monitoring, alerting, feature toggles, service discovery, service APIs, I/O management, and versioning. Breaking up a monolith or any sort of large rewrite is also a MAJOR undertaking—and a major risk for a company. The author doesn't hide these drawbacks, but in my opinion,

he massively understates them. Microservices are not for everyone (just like NoSQL and distributed systems aren't for everyone!), and it's a disservice to the industry to tell people otherwise.

Kirill says

I was somewhat skeptical about the book. Microservices are hype and seems that everyone nowadays has own advice on how to break the scary monolith. Still I found this book very good. Apart from the concrete "break the monolith" subject, it addresses many common architectural - and on my opinion - much more important questions - about good software design, art of decision making, coupling and cohesion, DDD and much more. Technical advices and sum up of different tooling are also quite useful.

Regis Hattori says

This book shows a lot of aspects to consider before using microservices.

It is hard to rate it because it is the first book I read about the topic. I really liked some parts, but I feel that the author did not put the same effort in other ones. There is not a consistent narrative among the topics. Some of them are focused on the concepts with detailed explanation. And some of them are more focused on tools.

Jose says

The book does a good job exposing the ideas and good practices behind a migration to (micro)services. if you've been doing it for a while it's a good way to checklist what you've done. If you're jumping on the wagon it has a nice overview about the practices and the tools that are out there to help you. I feel that the chapter about Scaling gives a good summary of the whole book.

Luís Soares says

If you have to read just one book about microservices, choose this. Quick&easy to read but very complete; it summarizes related subjects (not only the technical ones). A book to read from beginning to end, but also one to consult. It mentions a lot of technologies, but never without the corresponding concept/topic so it'll hardly get outdated. In the end, you just feel confident to develop this way.

Irina Lukyanenko says

????? ?? ??????? ??????? ?????????????? ? ????????, ???, ??????, ?????????????? ?????? ?????? ??????????????
?????????? ??????????, ?? ?????? ??????? ????? ?????????? ?????????????? ??????????????.

? ?????? ??????? ?????? ??????? ?????? ???, ??? ? ?????????? ?????????????? ?????????? ?????? ?????? ????????

"????????????? ??????". ?????? ??????????, ?????????????? ?????? ?? ??? ?????? ?????? ?????? ?????? ????

???????????? ?? ?????????? "Domain Driven Design".
