EVENTS: A LOVE TRIANGLE IN INTEGRATION TESTING

HEISENBUG CONFERENCE

ANDRES SACCO

saccoandres

andres-sacco

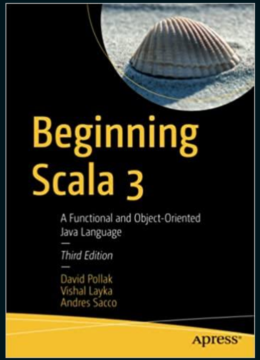
sacco.andres@gmail.com

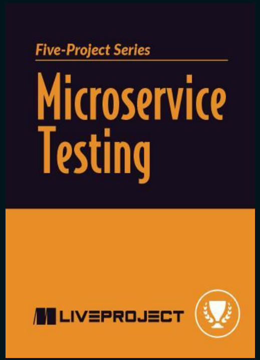
ABOUT ME

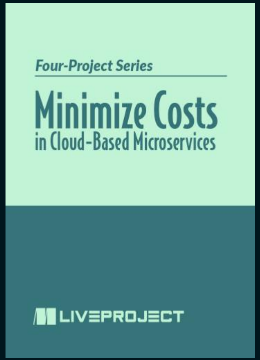
I’ve worked as developer since 2007 using different languages like Java, Kotlin, Scala, PHP, and NodeJS.

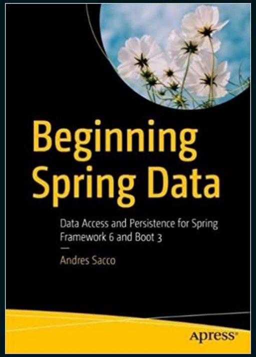
I've participated as a Technical Reviewer of Packt, Apress, and Manning books.

I published books, courses, and different theoretical-practical projects.





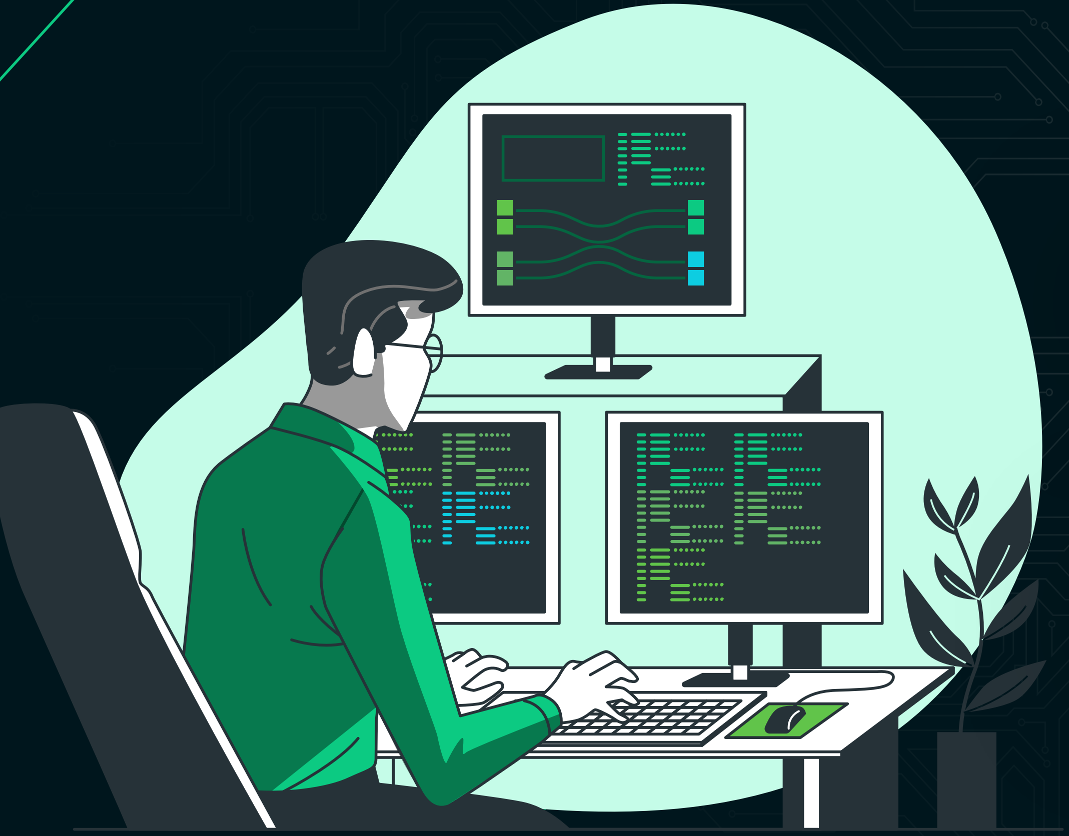



Beginning Spring Data
Data Access and Persistence for Spring Framework 6 and Boot 3
Andres Sacco
Apress

I participated as speaker on different conferences.

>CONTEXT OF THE SITUATION

You have a large number of microservices that use events to communicate each other



IT'S DIFFICULT TO CHECK IF EVERYTHING WORKS

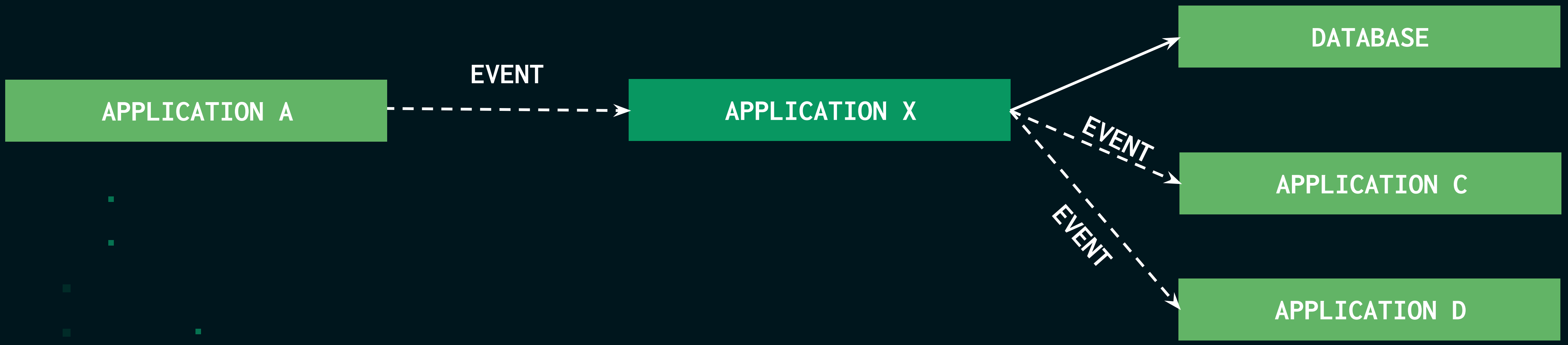
HOW CAN I CHECK IF SOMETHING IS MODIFIED IN THE DATABASE?

A NEW EVENT IS PUBLISHED AFTER PROCESS SOMETHING

AN UNIT TEST COULDN'T COVER ALL THIS ASPECTS

LET’S SEE THE INTERACTION

It’s **difficult** to understand the **role** of each component on this architecture



>WHY CREATE A TESTS IS RELEVANT?

The main problem with this type of architecture is that introduces a new problems

REDUCE RISK

SOMETHING BAD HAPPENS DURING THE PROCESS

VERIFY THE EVENTS

IT'S A WAY TO CHECK THE EVENTS PRODUCED

IT'S NOT AN UNIT TEST

AN UNIT TESTS NOT COVER ALL THE ASPECTS

ALL THE SCENARIOS

CHECK NOT ONLY THE SUCCESS SCENARIOS

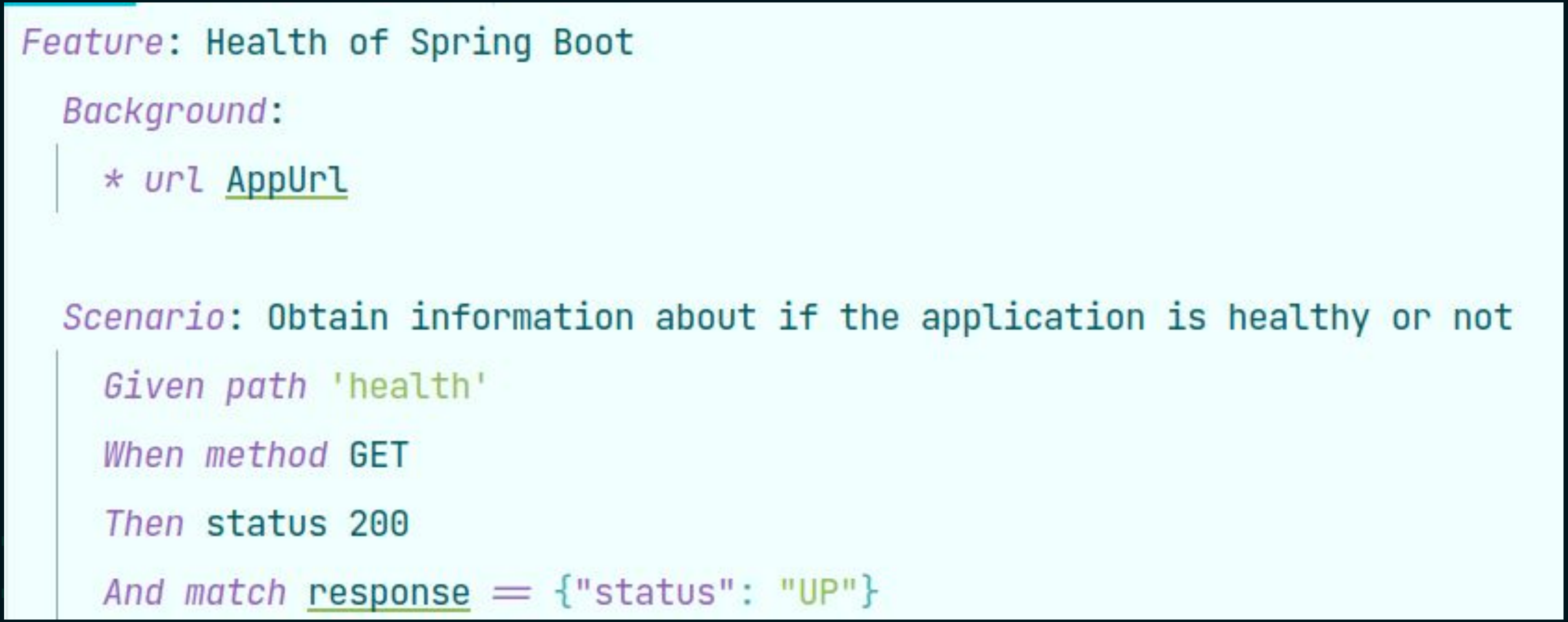
>WHICH LIBRARIES OR TOOLS EXISTS?

There are a series of libraries to solve the problems associated with the events

TESTCONTAINER	IT’S A TOOL TO RUN CONTAINERS IN YOUR APPLICATION
LOCALSTACK	IT'S A TOOL WHICH SIMULATE MOST OF THE AWS SERVICES
KARATE	IT’S A TOOL TO WRITE TESTS USING BDD
RESTHEART	IT’S A TOOL WHICH PROVIDE ENDPOINTS TO THE DATABASE

>HOW IS THE SYNTAX OF KARATE?

Let’s show you a simple example of the syntax

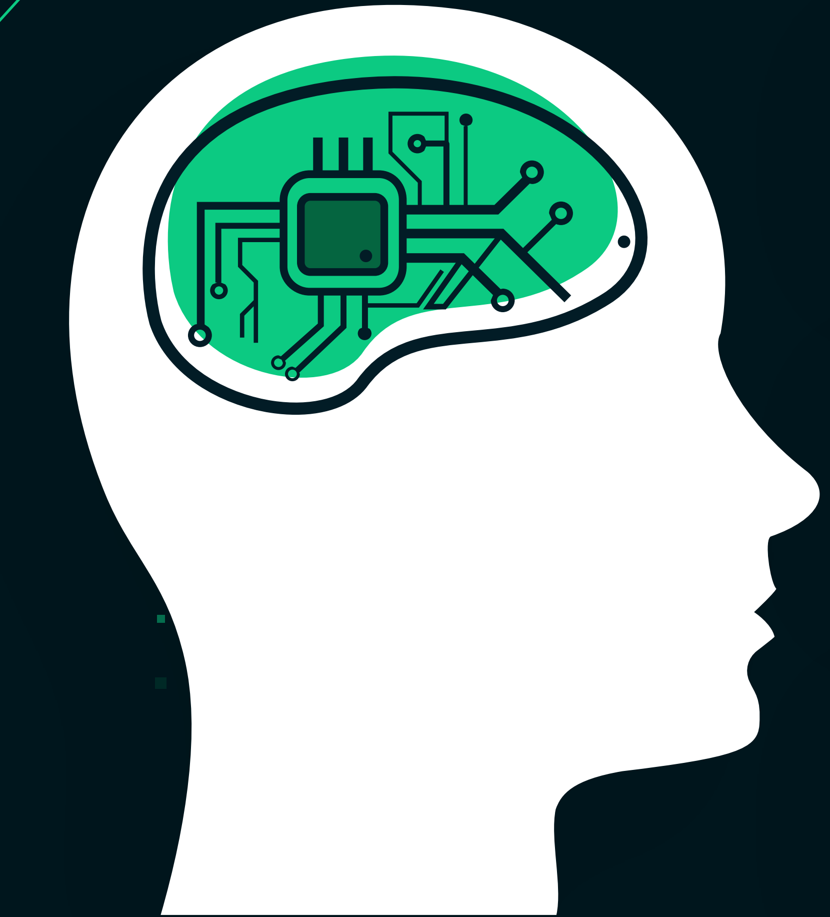


```
Feature: Health of Spring Boot
  Background:
    * url AppUrl

  Scenario: Obtain information about if the application is healthy or not
    Given path 'health'
    When method GET
    Then status 200
    And match response == {"status": "UP"}
```

PROS/CONS OF THIS APPROACH

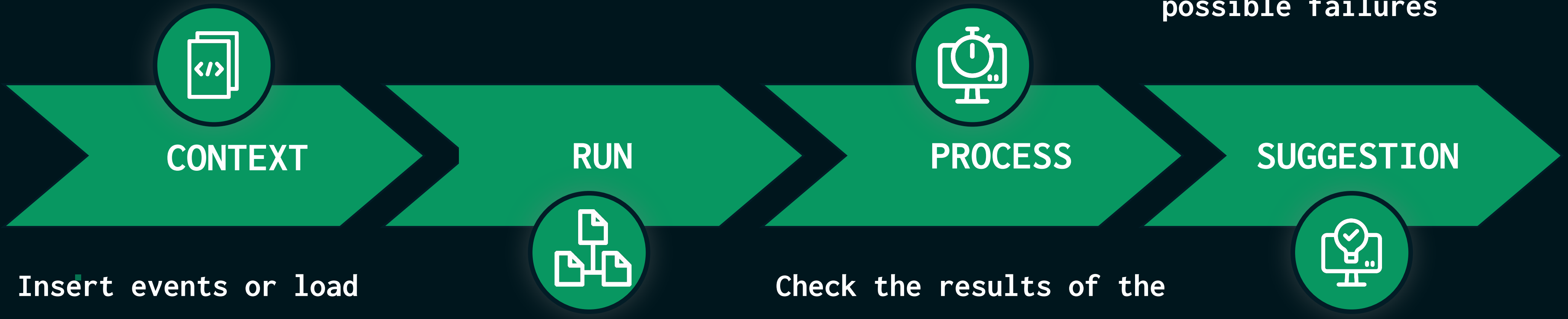
Using this technologies to create tests you have the following pros and cons



- NOT AFFECT THE REAL INFRASTRUCTURE ✓
- ENDPOINTS TO VERIFY THE DATABASES ✓
- YOU CAN SIMULATE THE PRODUCER OF EVENTS ✓
- IT'S AGNOSTIC TO THE LANGUAGES ✓
- TAKE MORE TIME TO EXECUTE THE TESTS ✗
- NOT HAVE SUPPORT TO ALL THE DATABASES ✗

> HOW IS THE PROCESS TO CHECK SOMETHING?

The process has four steps. Some of them are optional; others are mandatories



CONTEXT

Insert events or load information into the DB

RUN

Execute all the tests

PROCESS

Check the results of the execution

SUGGESTION

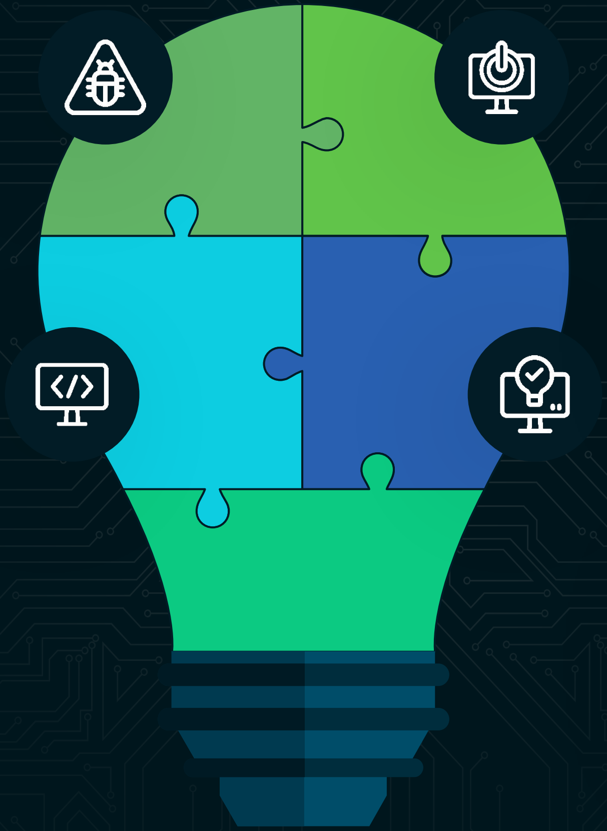
Remove any kind of events to prevent possible failures

> EXAMPLE

Let's see an example

https://github.com/andres-sacco/events-love-triangle

>BEST PRACTICES



CHECK EVERYTHING

CHECK EVENTS AND DATABASES STATUS

WRITE SIMPLE

TESTS NEED TO BE SIMPLE TO UNDERSTAND

REMOVE EVENTS

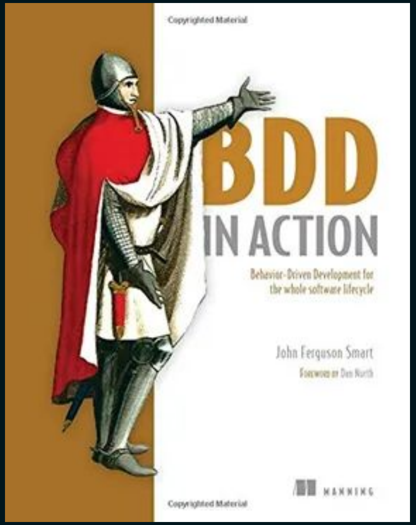
REMOVE EVENTS FROMS THE QUEUES

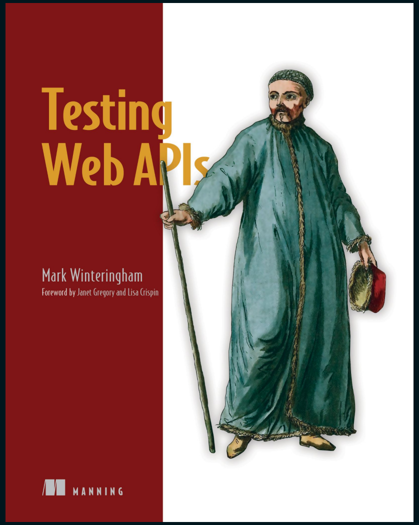
USE CONTAINERS

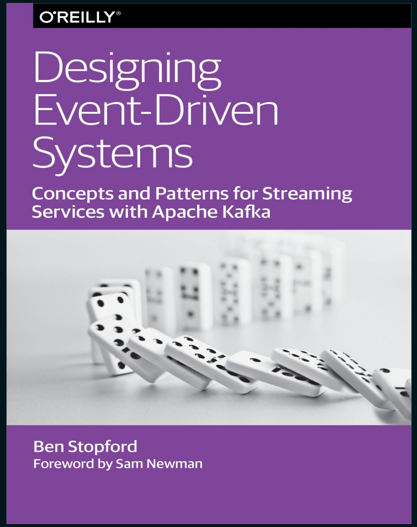
NOT USE THE REAL INFRASTRUCTURE

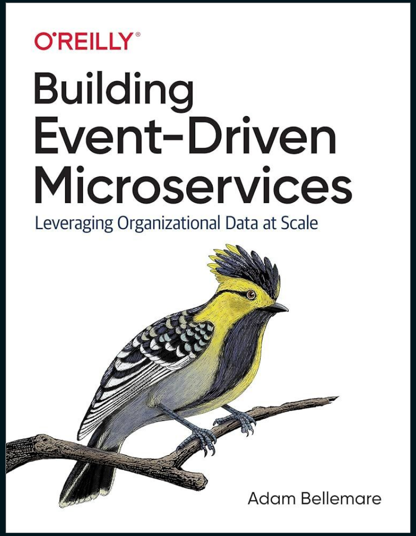
> ADDITIONAL RESOURCES

BOOKS






O'REILLY
Designing Event-Driven Systems
Concepts and Patterns for Streaming Services with Apache Kafka
Ben Stopford
Foreword by Sam Newman


O'REILLY
Building Event-Driven Microservices
Leveraging Organizational Data at Scale
Adam Bellemare

BLOGS

https://www.ministryoftesting.com/

https://martinfowler.com/

THANKS

HEISENBUG CONFERENCE