

Next-Generation Cloud Native Apps with Spring Boot 3

Thomas Vitale
GOTO Aarhus
May 23rd, 2023

@vitaletomas

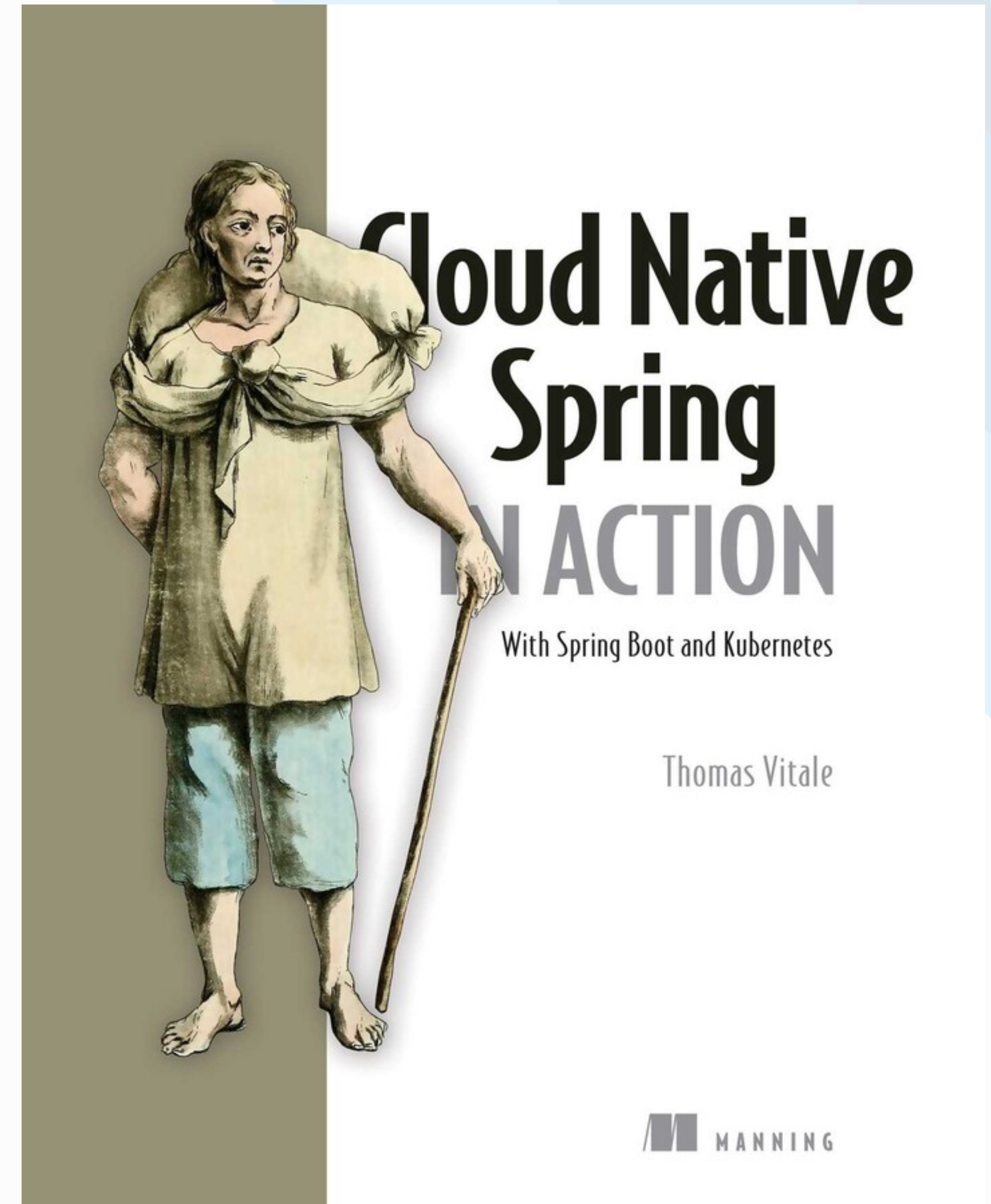
Thomas Vitale

Systematic

- **Software Engineer** and Cloud Architect.
- Author of “**Cloud Native Spring in Action**” (Manning).
- **OSS contributor** (Java, Spring, Cloud Native Technologies)

thomasvitale.com

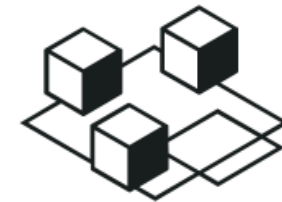
@vitaletomas



Spring



What Spring can do



Microservices

Quickly deliver production-grade features with independently evolvable microservices.



Reactive

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.



Cloud

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.



Web apps

Frameworks for fast, secure, and responsive web applications connected to any data store.



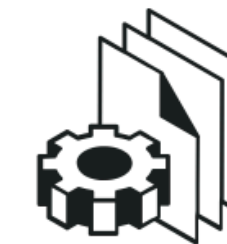
Serverless

The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.



Event Driven

Integrate with your enterprise. React to business events. Act on your streaming data in realtime.



Batch

Automated tasks. Offline processing of data at a time to suit you.

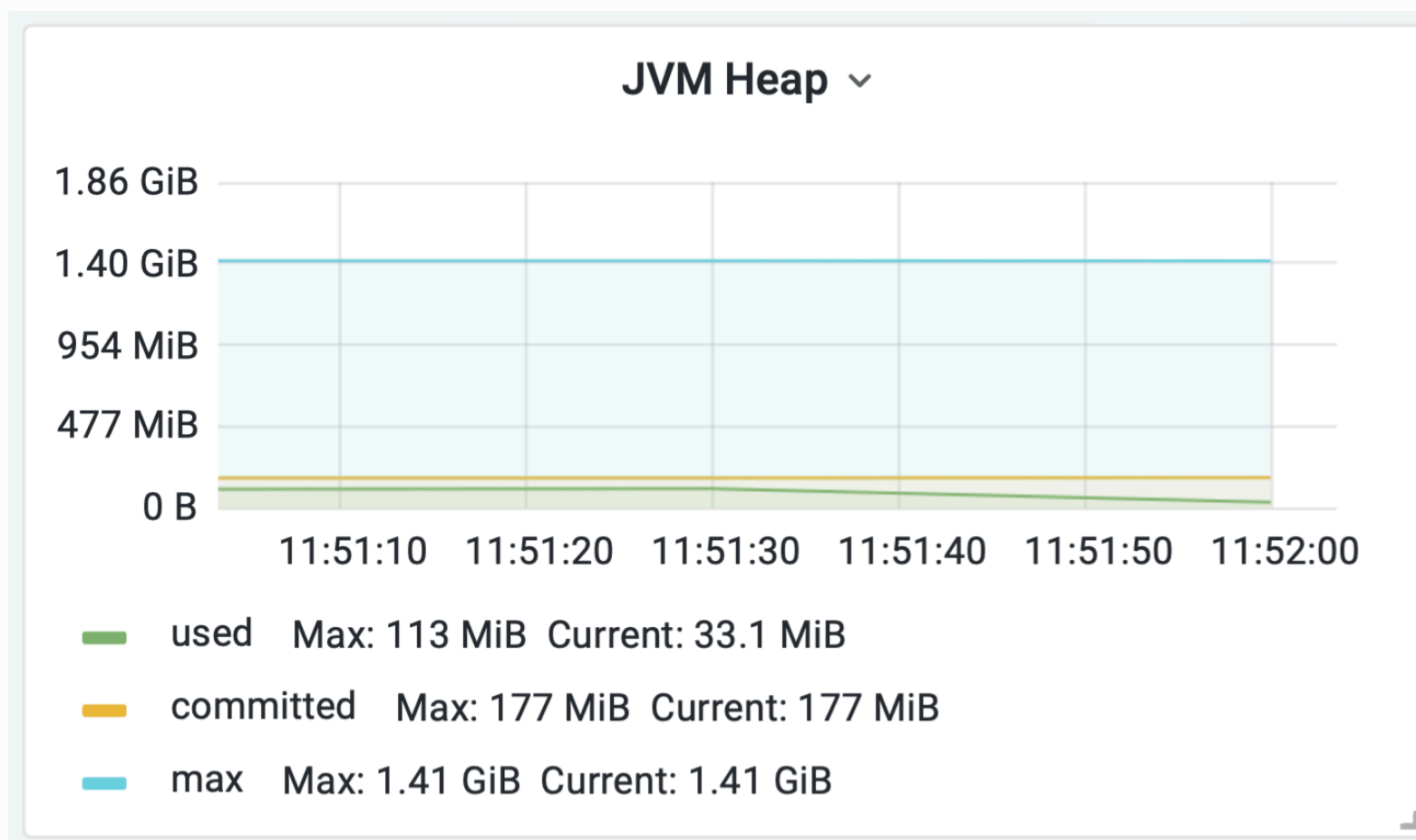
Java 17

Java 8 -> 11 -> 17

Performance improvements

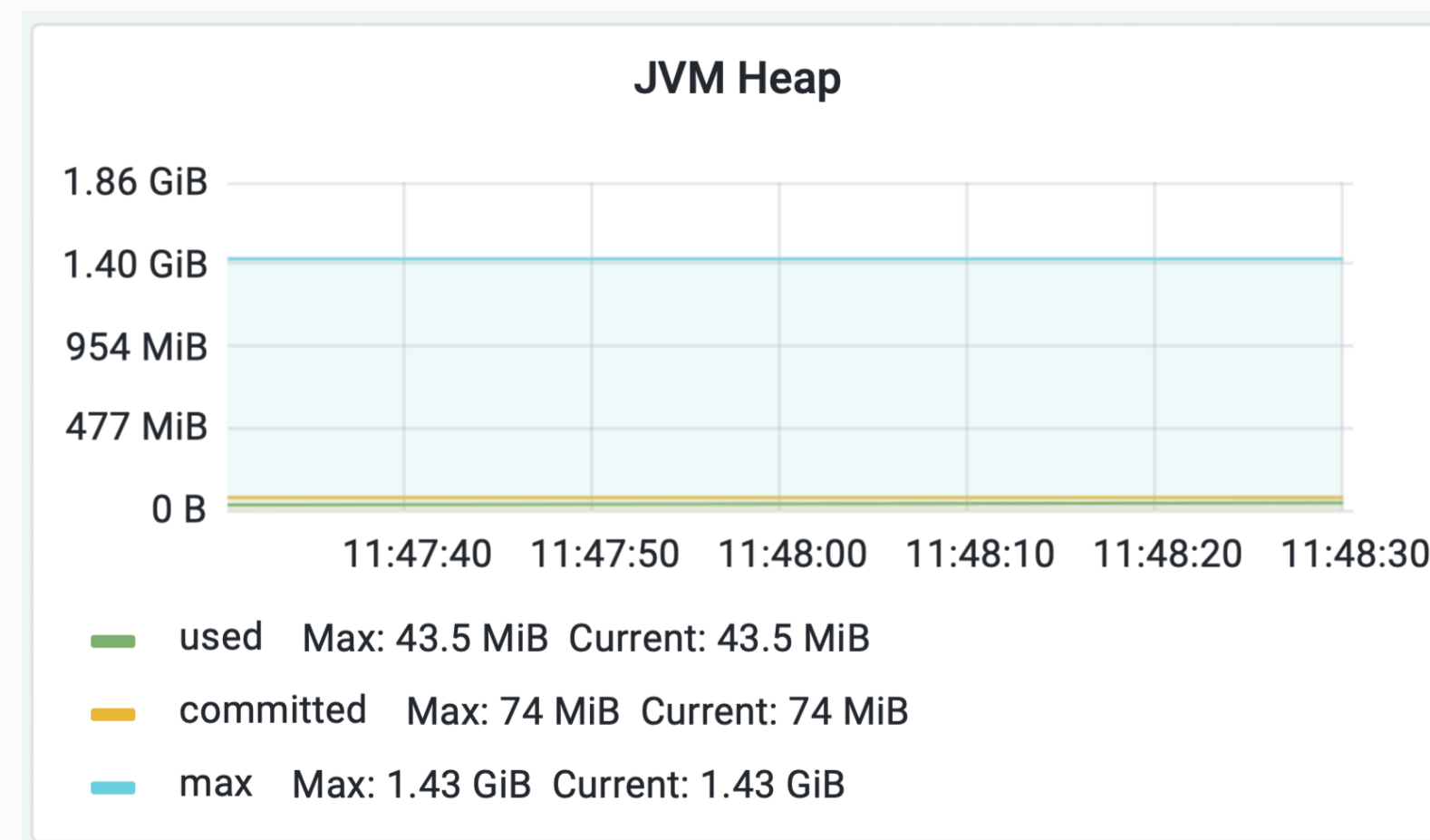
```
resources:
  limits:
    cpus: '2'
    memory: 2G
  reservations:
    cpus: '2'
    memory: 2G
```

Java 8



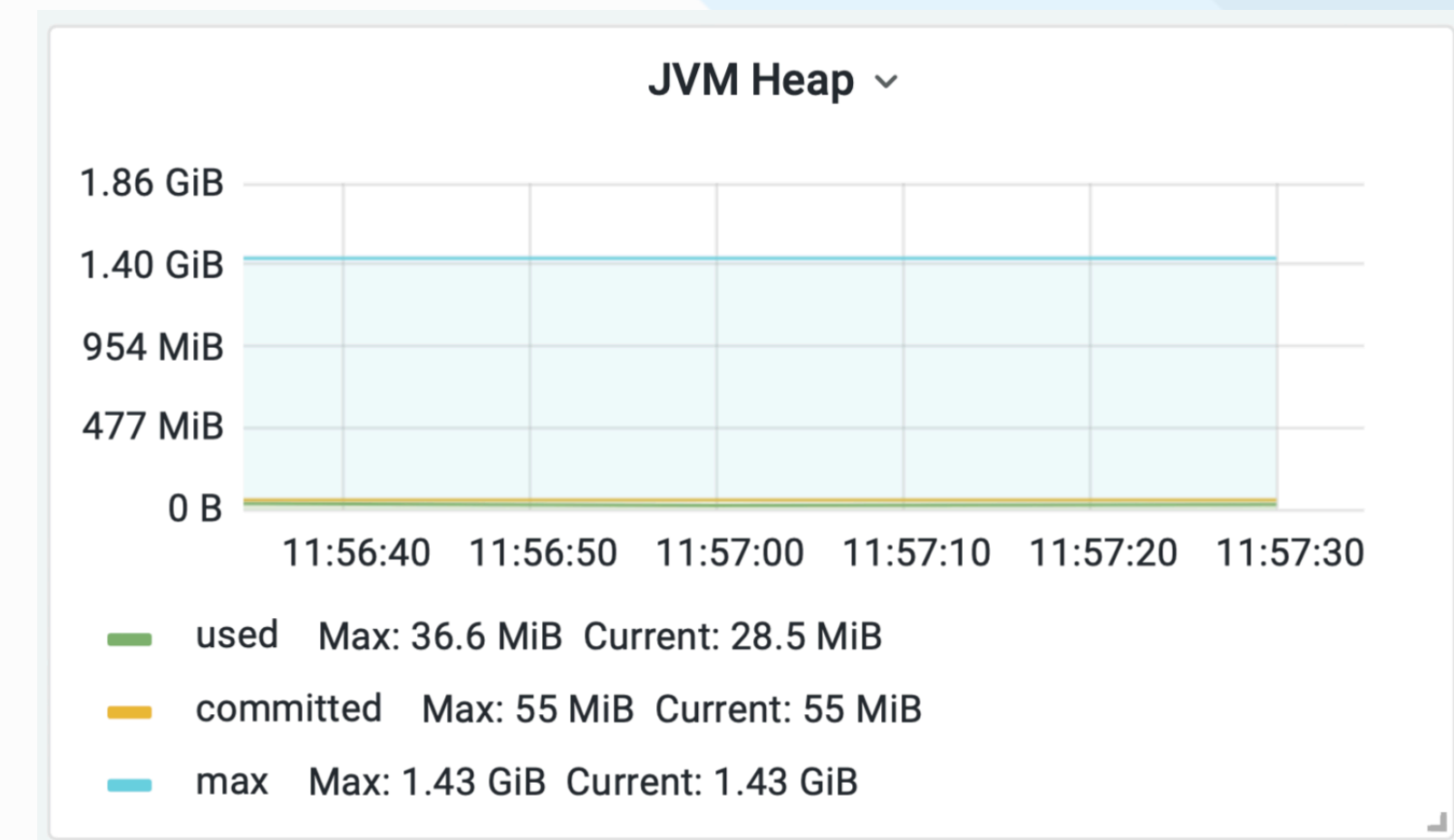
- Startup: **3.784 s**
- Max heap used: **113 MiB**
- Committed heap: **177 MiB**

Java 11



- Startup: **2.204 s**
- Max heap used: **43 MiB**
- Committed heap: **74 MiB**

Java 17



- Startup: **1.776 s**
- Max heap used: **37 MiB**
- Committed heap: **55 MiB**

Deployment

Packaging Spring Boot

JAR & Container Image

JAR

Gradle

`bootJar`

Maven

`spring-boot:repackage`

Container Image

Gradle

`bootBuildImage`

Maven

`spring-boot:build-image`

“Friends don’t let friends write Dockerfiles!”

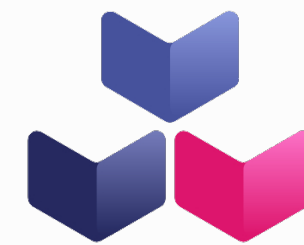
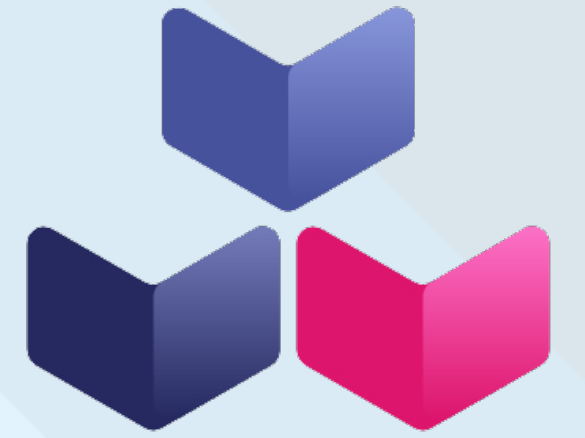
- Josh Long

Dockerfiles

“Dockerfiles are **easy to write**, but the current development guidelines do **not** produce containers that are **repeatable and hardened.**”

Cloud Native Buildpacks

From source code to container image



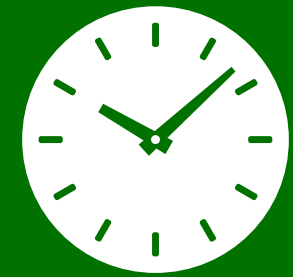
pack build

Image

Native Images

Spring Boot 3 and Native Executables

The benefits of GraalVM



Instant
Startup



Instant
Peak
Performance



Low
Resource
Usage



Reduced
Surface
Attack



Compact
Packaging

Spring Boot 3 and Native Executables

The downsides of GraalVM



Slower
Heavier
Build



Additional
configuration
might be
needed

Data Persistence

Data Integration

Spring Data for Relational Databases

Spring Data JPA

- JPA & Hibernate
- Repositories
- JDBC

Spring Data JDBC

- DDD Principles
- Repositories
- JDBC

Spring Data R2DBC

- DDD Principles
- Repositories
- R2DBC

Data Integration

Spring Data for Relational Databases

Spring Data JPA

- JPA & Hibernate
- Repositories
- JDBC

Spring Data Relational

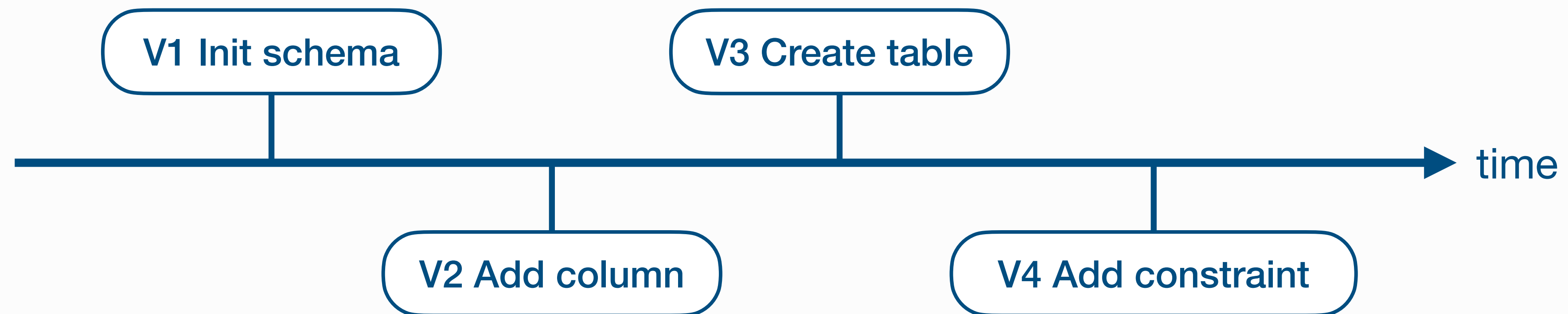
- DDD Principles
- Repositories
- JDBC/R2DBC

Schema and data management

Flyway: Version control for your database

SQL Migrations
Schema changes

Java Migrations
Data changes



Testing

Testing Spring Boot

Types of auto tests for Spring Boot applications

Unit Tests

- JUnit
- Mockito
- No Spring context

Integration Tests

- @SpringBootTest
- Web server
- Full Spring context

Slice Tests

- @SpringWebMvc
- @SpringDataR2dbc
- Sliced Spring context

Testcontainers

Development and testing with external dependencies

OCI containers

Run external dependencies as OCI containers

Data Layer Tests

Ensure environment parity by testing the data layer with the real database

Integration Tests

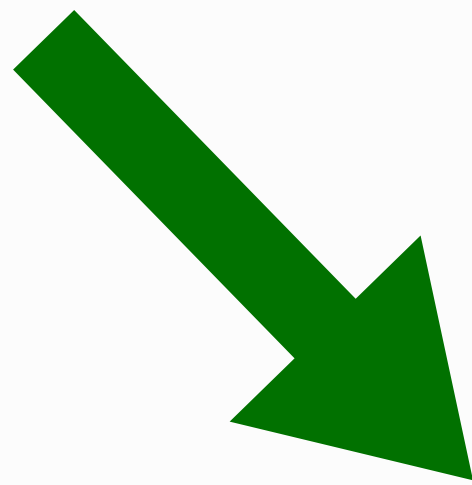
Use containers for databases, message queues, and web servers

Jakarta EE 10

From Java EE to Jakarta EE

New package names since Jakarta EE 9

```
import javax.persistence.Entity;  
import javax.servlet.Servlet;  
import javax.validation.Valid;
```



```
import jakarta.persistence.Entity;  
import jakarta.servlet.Servlet;  
import jakarta.validation.Valid;
```

Introduction to OpenRewrite

RUNNING RECIPES

Quickstart: Setting up your project and running recipes

Running Rewrite on a Gradle project without modifying the build

Running Rewrite without build tool plugins

Popular recipe guides >

AUTHORING RECIPES

Recipe development environment

Writing a Java refactoring recipe

Recipe testing

Recipe conventions and best practices

Modifying methods with JavaTemplate

Creating multiple visitors in one recipe

Introduction to OpenRewrite

⋮

Large-scale automated source code refactoring

OpenRewrite enables large-scale distributed source code refactoring for framework migrations, vulnerability patches, and API migrations with an early focus on the Java language.

Semantic code search and transformation

The OpenRewrite project is a [semantic code search](#) and transformation ecosystem for Java and other source code. It consists of a platform of prepackaged refactoring recipes for common framework migration and stylistic consistency tasks. It also allows you to define custom recipes to achieve a wide range of source code transformations.

What does OpenRewrite do?

OpenRewrite works by making changes to [Lossless Semantic Trees](#) (LST) representing your source code and printing the modified trees back into source code. You can then review the changes in your code and commit. Modifications to the LST are performed in [Visitors](#) and visitors are aggregated into [Recipes](#). OpenRewrite recipes make minimally invasive changes to your source code that honor the original formatting.

Observability

Spring Observability

Production-grade features

Spring Boot Actuator

- Health (liveness and readiness)
- Metrics (Prometheus, OpenMetrics)
- Flyway, Thread Dumps, Heap Dumps

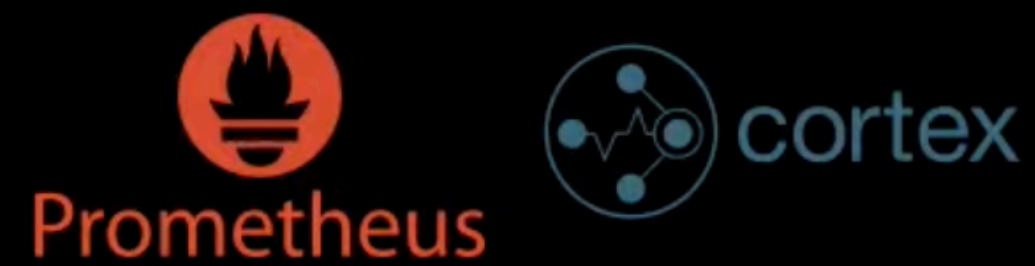
Micrometer

- Unified Observation API
- Instrumentation for metrics and traces
- OpenZipkin, OpenTelemetry

The curated open-source observability stack



Traces



Metrics



Logs

Declarative Clients

Problem Details

[\[RFC Home\]](#) [\[TEXT\]](#) [\[PDF\]](#) [\[HTML\]](#) [\[Tracker\]](#) [\[IPR\]](#) [\[Errata\]](#) [\[Info page\]](#)

PROPOSED STANDARD

Errata Exist

Internet Engineering Task Force (IETF)

M. Nottingham

Request for Comments: 7807

Akamai

Category: Standards Track

E. Wilde

ISSN: 2070-1721

March 2016

Problem Details for HTTP APIs

Abstract

This document defines a "problem detail" as a way to carry machine-readable details of errors in a HTTP response to avoid the need to define new error response formats for HTTP APIs.

Status of This Memo

This is an Internet Standards Track document.


This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7807>.




<https://www.rfc-editor.org/rfc/rfc7807>


@vitaletomas





Resources




[Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)


  


 [ThomasVitale / spring-boot-next-gen-apps](#) Public

 Pin  Unwatch 1  Fork 1  Star 2


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)








 main



 2 branches

 0 tags

[Go to file](#) [Add file](#) [Code](#)

 **ThomasVitale** Update buildpacks config a70268a 4 minutes ago 3 commits

	book-service	Update buildpacks config	4 minutes ago
	order-service	Update buildpacks config	4 minutes ago
	platform	Initial commit	yesterday
	.gitignore	Initial commit	yesterday
	LICENSE	Initial commit	5 days ago
	README.md	Initial commit	yesterday
	docker-compose.yml	Initial commit	yesterday

 README.md 






Next-Generation Apps with Spring Boot 3

The recent release of Spring Boot 3 and Spring Framework 6 laid the foundation for the next generation of modern Java applications. This session will highlight what's new, and demonstrate patterns and techniques for cloud-native development.

About

Samples showcasing new features and capabilities in Spring Boot 3 and Spring Framework 6

[java](#) [spring-boot](#) [cloud-native](#) [spring-framework](#) [graalvm](#) [buildpacks](#)

 Readme  Apache-2.0 license  2 stars  1 watching  1 fork

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

<https://github.com/ThomasVitale/spring-boot-next-gen-apps>

ProductTeamEnterpriseExploreMarketplacePricing

Search / Sign in Sign up

ThomasVitale / awesome-spring Public

Notifications Fork 89 Star 514

<> Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights

main 1 branch 0 tags Go to file Code

ThomasVitale Add new resources ef1eba1 on Feb 21 37 commits

.github/workflows	Add linter	2 years ago
.gitignore	Add first version of awesome list	2 years ago
CONTRIBUTING.md	Add first version of awesome list	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Add new resources	3 months ago

README.md

awesome Lint Code Base passing

A curated list of awesome books, tutorials, courses, and resources for the Spring framework ecosystem.

If you're new to Spring, start with watching the [What is Spring?](#) video.

Contents

- Books
- Courses
- Podcasts
- Events
- Educational Projects
- Other Resources
 - Blogs
 - YouTube Channels and Playlists
- Contributing

About

A curated list of awesome books, tutorials, courses, and resources for the Spring framework ecosystem.

awesome spring spring-boot spring-data spring-cloud spring-security awesome-list spring-framework

Readme

CC0-1.0 license

514 stars

39 watching

89 forks

Releases

No releases published

Packages

No packages published

Contributors 9

<https://github.com/ThomasVitale/awesome-spring>

Next-Generation Cloud Native Apps with Spring Boot 3

Thomas Vitale
GOTO Aarhus
May 23rd, 2023

@vitaletomas