Agenda

- Spring Boot 3 Migration

- What's New?

- What's Next?

- Q/A
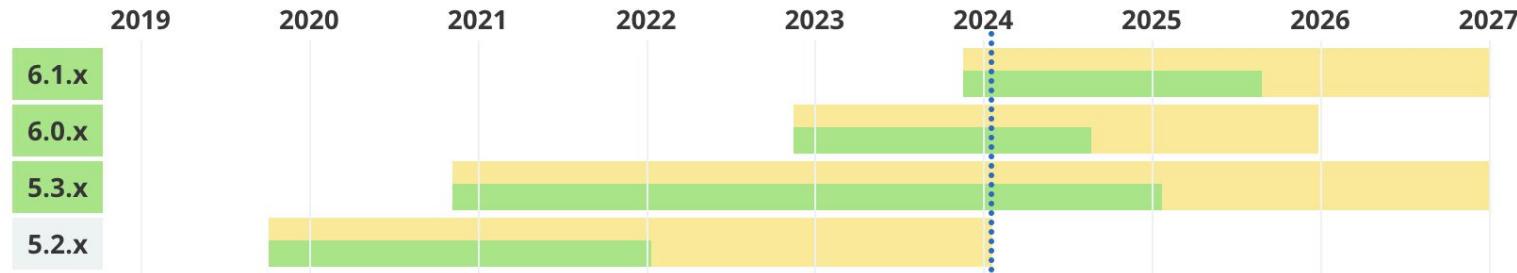
🌱 spring®

Spring Boot 3 Migration

spring®

Spring Framework Support

Branch	Initial Release	End of Support	End Commercial Support *
● 6.1.x	2023-11-16	2025-08-31	2026-12-31
● 6.0.x	2022-11-16	2024-08-31	2025-12-31
● 5.3.x	2020-10-27	2024-12-31	2026-12-31
○ 5.2.x	2019-09-30	2021-12-31	2023-12-31
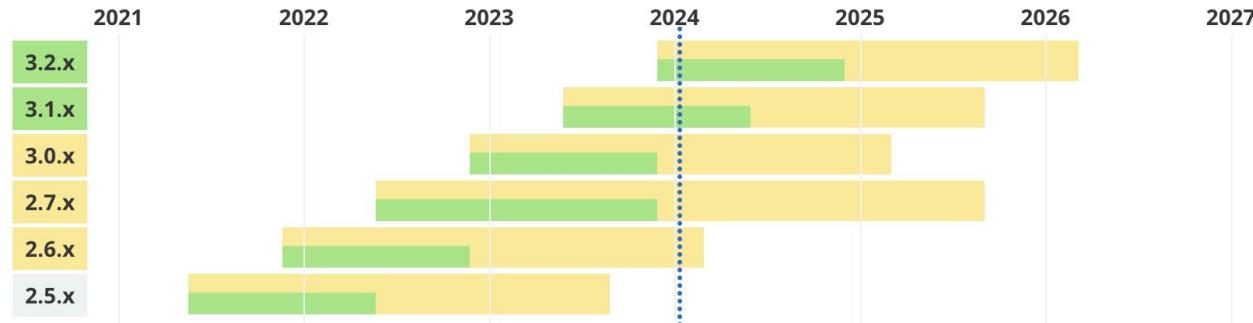
More ∨
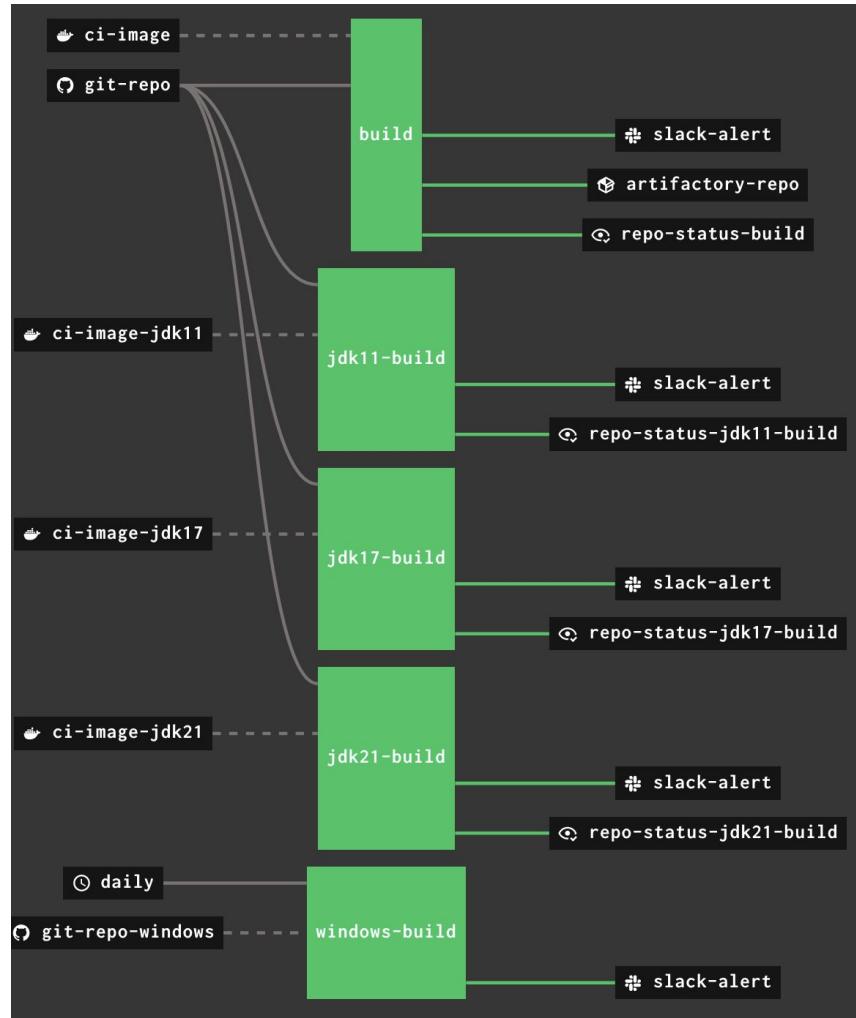
Spring Boot Support

Branch	Initial Release	End of Support	End Commercial Support *
🟢 3.2.x	2023-11-23	2024-11-23	2026-02-23
🟢 3.1.x	2023-05-18	2024-05-18	2025-08-18
🟡 3.0.x	2022-11-24	2023-11-24	2025-02-24
🟡 2.7.x	2022-05-19	2023-11-24	2025-08-24
🟡 2.6.x	2021-11-17	2022-11-24	2024-02-24
⚪ 2.5.x	2021-05-20	2022-05-19	2023-08-24

More ⌄

spring

Migration pre-steps

1. Upgrade to the latest 2.7.x release (2.7.18).

2. Upgrade to Java 17 at least (why not 21?).

3. Check deprecations in code.

4. Check deprecations in application properties (properties migrator).

5. Using Spring Security? Upgrade to [Spring Security 5.8](#) first.

6. Check for warnings in your logs!

7. Check for dependency version overrides / Use our BOM

🌱 spring

Migrate to Spring Boot 3

You are now ready to get started!

- Upgrade to the **<u>latest</u>** 3.**0**.x (3.0.13).

- Each feature release has [dedicated release notes](#).

- Each major release has a [specific migration guide](#).

- The release notes link to more specific release notes and/or migration guides for the Spring Modules you might be using.

🍃 spring®

Jakarta EE 9+ upgrade

A community effort

- javax.* ➜ jakarta.* namespace

- Api spec dependencies upgrade

- Open source libraries upgrade

- Synchronizing with Spring Boot supported libraries

🍃 spring®

Migrate to Spring Boot 3

Rinse & Repeat

- Upgrade to the latest 3.**1**.x (3.1.8 at this time) and follow the [dedicated release notes](#).

- [One more time](#) and you are on 3.2.x!

spring®

What's New in Spring Boot 3?

spring®

Virtual Threads

Project Loom

A new degree of scalability for Spring MVC applications

spring®

Paul Bakker
@pbakker

...

Spring Boot 3.2 is released today! No surprise that I'm specially excited about the virtual thread support. Will start rolling out at Netflix starting next week already 👌 spring.io/blog/2023/11/2... #java

🍃 spring.

HTTP Client API

Revisited

```java
RestClient client = RestClient.create(baseUrl);
ResponseEntity<Person> person =
client.get().uri("/persons/1")
        .accept(MediaType.APPLICATION_JSON)
        .retrieve()
        .toEntity(Person.class);
```
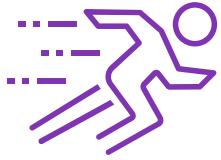
spring®

Spring AOT
GraalVM Native Images
A new deployment variant

As an alternative to classic HotSpot deployment

spring®

GraalVM advantages

Instant startup

Milliseconds for native instead of seconds for the JVM

No warmup

Peak performance available immediately

Low resource usage

Lower memory footprint and no JIT compilation

Reduced surface attack
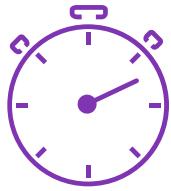
Closed world of dependencies with explicit reflection and serialization

Compact packaging

Smaller container easier to deploy

spring®

GraalVM trade-offs

Very slow compilation
Minutes instead of seconds
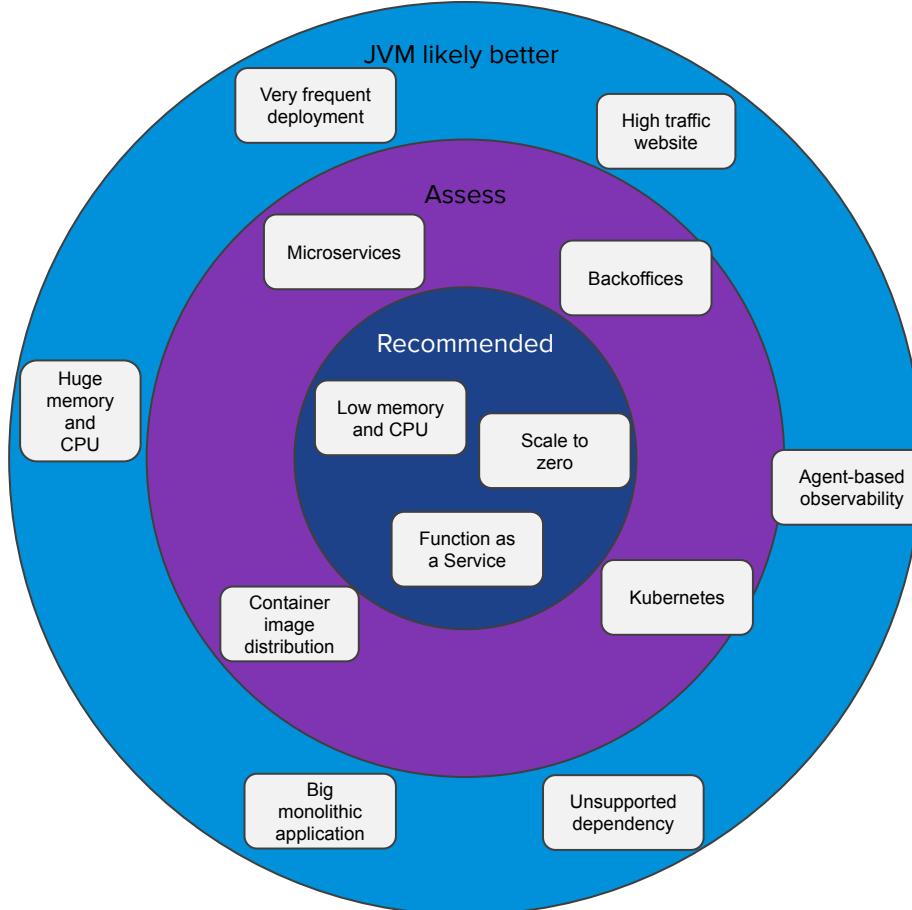
Compatibility
Additional metadata required for reflection, proxies, resources

Closed-world Assumptions
Bean conditions fixed at build time
No dynamic class loading
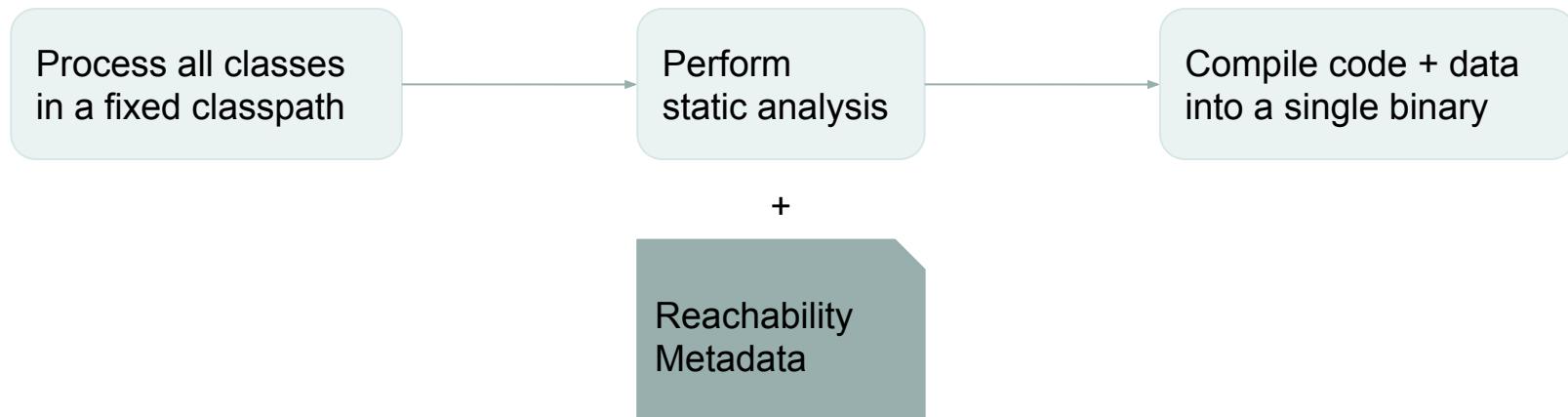
spring®

Use cases for native images

JVM likely better

Very frequent deployment

High traffic website

Assess

Microservices

Backoffices

Recommended

Low memory and CPU

Scale to zero

Huge memory and CPU

Function as a Service

Agent-based observability

Container image distribution

Kubernetes

Big monolithic application

Unsupported dependency
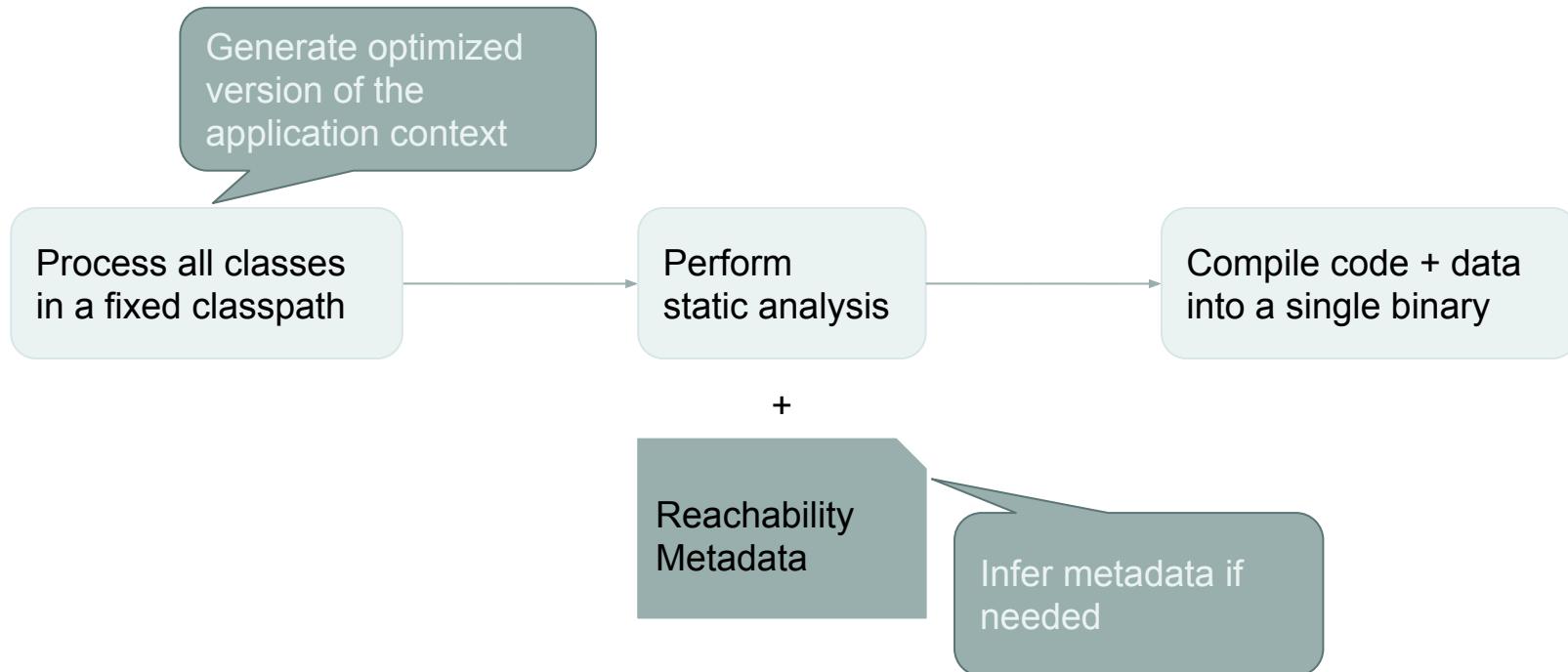
spring

Building a native image with GraalVM
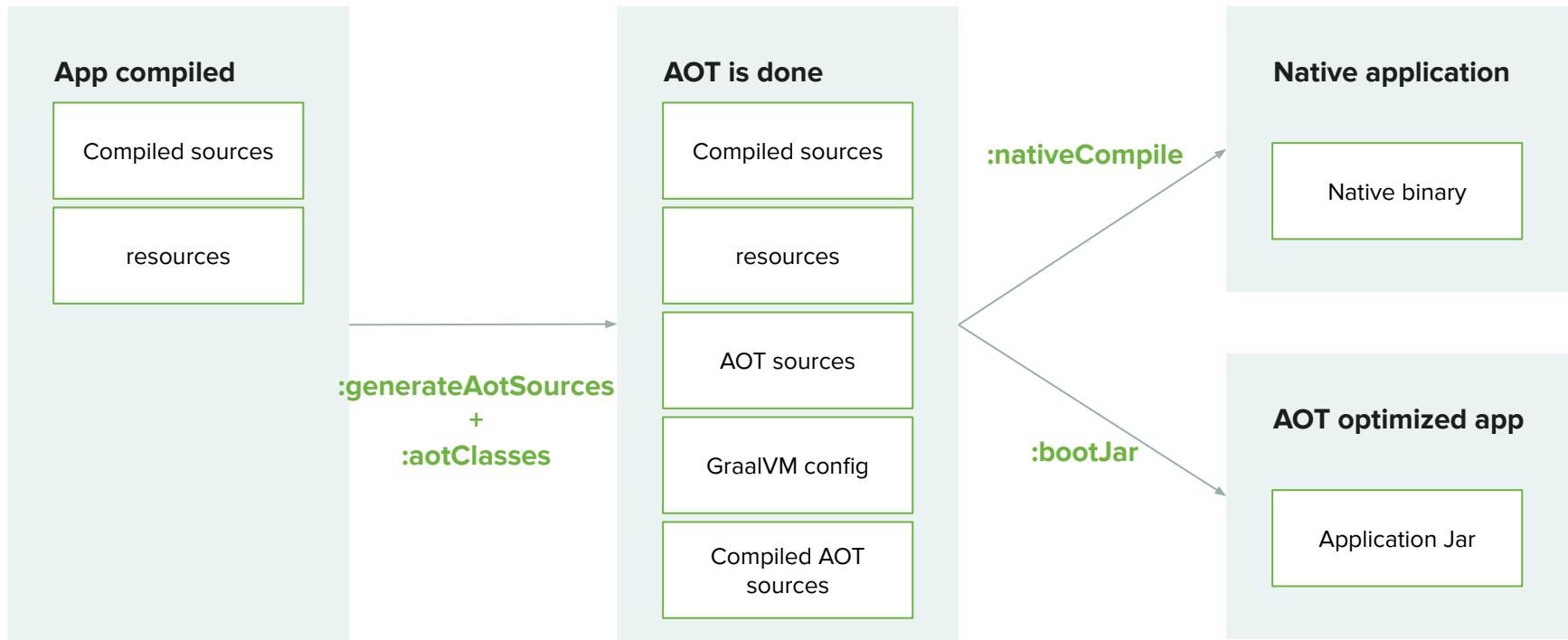
Using the "native-image" binary

Process all classes in a fixed classpath	Perform static analysis	Compile code + data into a single binary

+

Reachability
Metadata

 spring

How Spring adapts for native

Generate optimized
version of the
application context

Process all classes
in a fixed classpath

Perform
static analysis

Compile code + data
into a single binary

+

Reachability
Metadata

Infer metadata if
needed

Also applies to resources, JDK proxies... See class metadata features

spring®

AOT processing with Spring

App compiled

> Compiled sources

> resources

:generateAotSources
+
:aotClasses

AOT is done

> Compiled sources

> resources

> AOT sources

> GraalVM config

> Compiled AOT sources

:nativeCompile

:bootJar

Native application

> Native binary

AOT optimized app

> Application Jar

spring®

spring initia

Web, Security, JPA, Actuator, Devtools...   Press ⌘ for multiple adds

DEVELOPER TOOLS

GraalVM Native Support   ↵
Support for compiling Spring applications to native executables using the GraalVM native-image compiler.

GraphQL DGS Code Generation
Generate data types and type-safe APIs for querying GraphQL APIs by parsing schema files.

Spring Boot DevTools
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Lombok
Java annotation library which helps to reduce boilerplate code.

Spring Configuration Processor
Generate metadata for developers to offer contextual help and "code completion" when working with custom
configuration keys (ex.application.properties/.yml files).

Docker Compose Support
Provides docker compose support for enhanced development experience.

Spring Modulith
Support for building modular monolithic applications.

WEB

Project
● Gradle - Groovy
○ Gradle - Kotlin   ○ Maven

Spring Boot
○ 3.2.1 (SNAPSHOT)   ● 3.2.0

Project Metadata
Group   com.example

Artifact   demo

Name   demo

Description   Demo project for

Package name   com.example.den

Packaging   ● Jar   ○ Wa

Java   ○ 21   ● 17

ADD DEPENDENCIES... ⌘ + B

Observability
Metrics and Tracing
Unified through Micrometer

Across the runtime stack

spring®

# Instrumenting libraries for observability
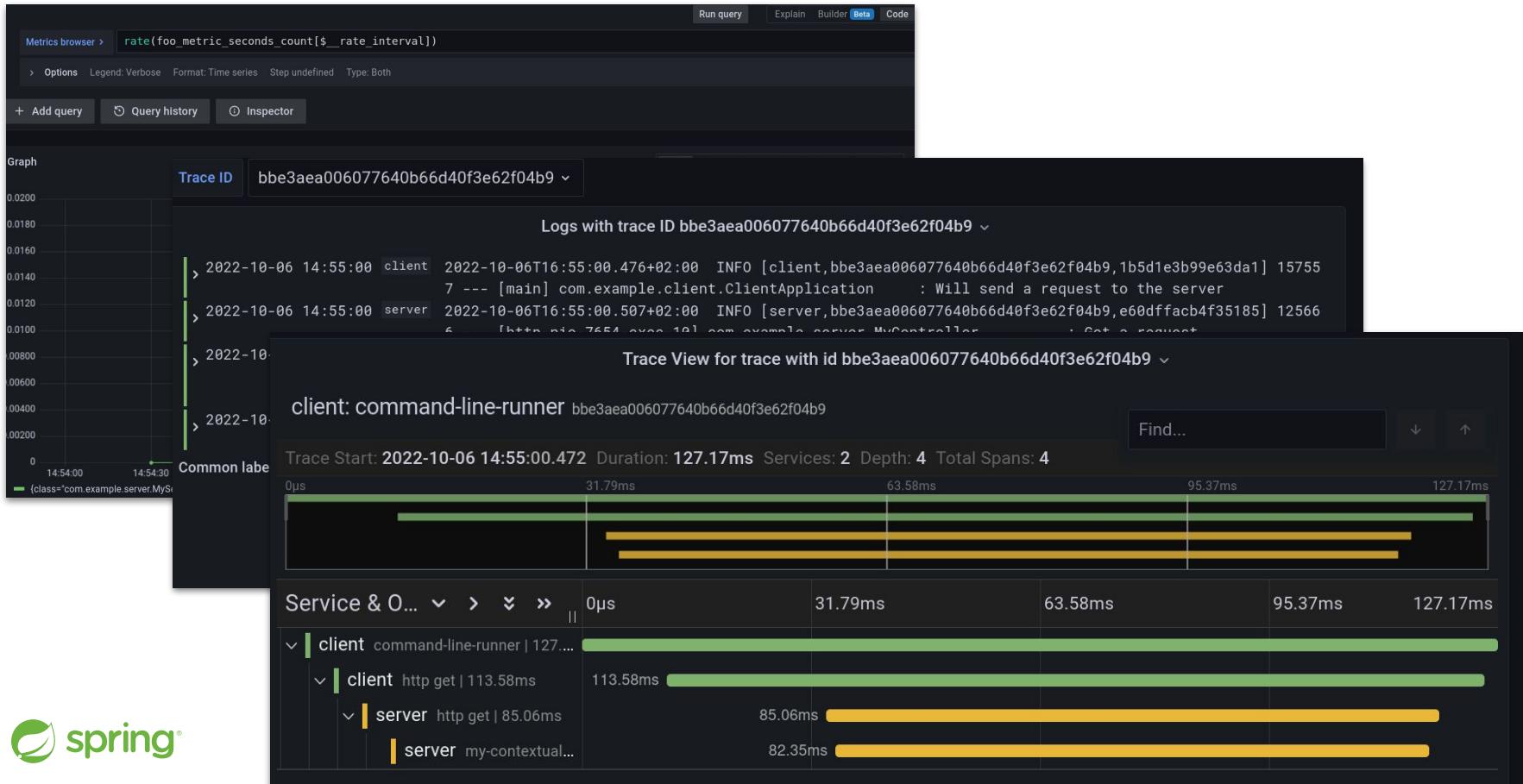
Instrumentation strategies

- Java agent, at the bytecode level

- Using public extension points

- Direct instrumentation in the library

Micrometer 1.10.0
with io.micrometer.observation.Observation

spring®

Metrics, Logs correlation and Traces in Spring apps

Metrics browser › `rate(foo_metric_seconds_count[$__rate_interval])`

Run query   Explain   Builder Beta   Code

› Options   Legend: Verbose   Format: Time series   Step undefined   Type: Both

+ Add query   Query history   ⓘ Inspector

Graph

0.0200
0.0180
0.0160
0.0140
0.0120
0.0100
.00800
.00600
.00400
.00200
0
14:54:00   14:54:30   Common labe

{class="com.example.server.MyS

Trace ID   bbe3aea006077640b66d40f3e62f04b9 ˅

Logs with trace ID bbe3aea006077640b66d40f3e62f04b9 ˅

› 2022-10-06 14:55:00 `client` 2022-10-06T16:55:00.476+02:00 INFO [client,bbe3aea006077640b66d40f3e62f04b9,1b5d1e3b99e63da1] 15755
7 --- [main] com.example.client.ClientApplication   : Will send a request to the server

› 2022-10-06 14:55:00 `server` 2022-10-06T16:55:00.507+02:00 INFO [server,bbe3aea006077640b66d40f3e62f04b9,e60dffacb4f35185] 12566
6   [http nio 7654 exec 10] com example server MyController   : Got a request

› 2022-10-

› 2022-10-

Trace View for trace with id bbe3aea006077640b66d40f3e62f04b9 ˅

client: command-line-runner bbe3aea006077640b66d40f3e62f04b9

Find...   ↓   ↑

Trace Start: **2022-10-06 14:55:00.472** Duration: **127.17ms** Services: **2** Depth: **4** Total Spans: **4**

0µs   31.79ms   63.58ms   95.37ms   127.17ms

Service & O... ˅ › » »»   0µs   31.79ms   63.58ms   95.37ms   127.17ms

˅ ▌ **client** command-line-runner | 127....

 ˅ ▌ **client** http get | 113.58ms   113.58ms

  ˅ ▌ **server** http get | 85.06ms   85.06ms

   ▌ **server** my-contextual...   82.35ms

🌱 spring®

Creating your own observations

```java
String conference = "Spring Meetup";
return Observation.createNotStarted("spring.talk", registry)
        .lowCardinalityKeyValue( "conference", conference)
        .observe(() -> {
            logger.info("Talk about Observations" );
            return restClient.get()
                        .uri( "https://example.com/conferences/{id}" , conference)
                        .retrieve().body( String.class);
        });
```

Service Connection

Improved support for Testcontainers and Docker Compose

See "Docker Compose Support in Spring Boot 3.1" and "Improved Testcontainers Support in Spring Boot 3.1".

spring®

Problem Details Support

RFC 7807

```java
@ExceptionHandler(UnknownProjectException.class)
ProblemDetail handleUnknownProjectException(UnknownProjectException exc) {
    ProblemDetail problemDetail = ProblemDetail.forStatus(HttpStatus.NOT_FOUND);
    problemDetail.setTitle("Unknown Project");
    problemDetail.setType(URI.create(
            "https://example.org/problems/unknown-project"));
    problemDetail.setDetail(String.format(
            "No project found with slug '%s'", exc.getSlug()));
    return problemDetail;
}
```

JDBC Client API

Revisited

```java
JdbcClient client = JdbcClient.create(dataSource);
Optional<Person> person = client.sql("SELECT ... WHERE ID = :id")
        .param("id", 3)
        .query(Person.class)
        .optional();
```

spring

Java Interface Clients

HTTP, RSocket, ...

```java
interface SpringProjectsService {

    @GetExchange("/projects/{slug}")
    ProjectSummary findProject(@PathVariable String slug);

}
```

spring®

SSL Improvements

(Reloadable) SSL Bundles

See "[Securing Spring Boot Applications With SSL](#)" and "[SSL hot reload in Spring Boot 3.2.0](#)".

spring

What's Next?*

*under consideration
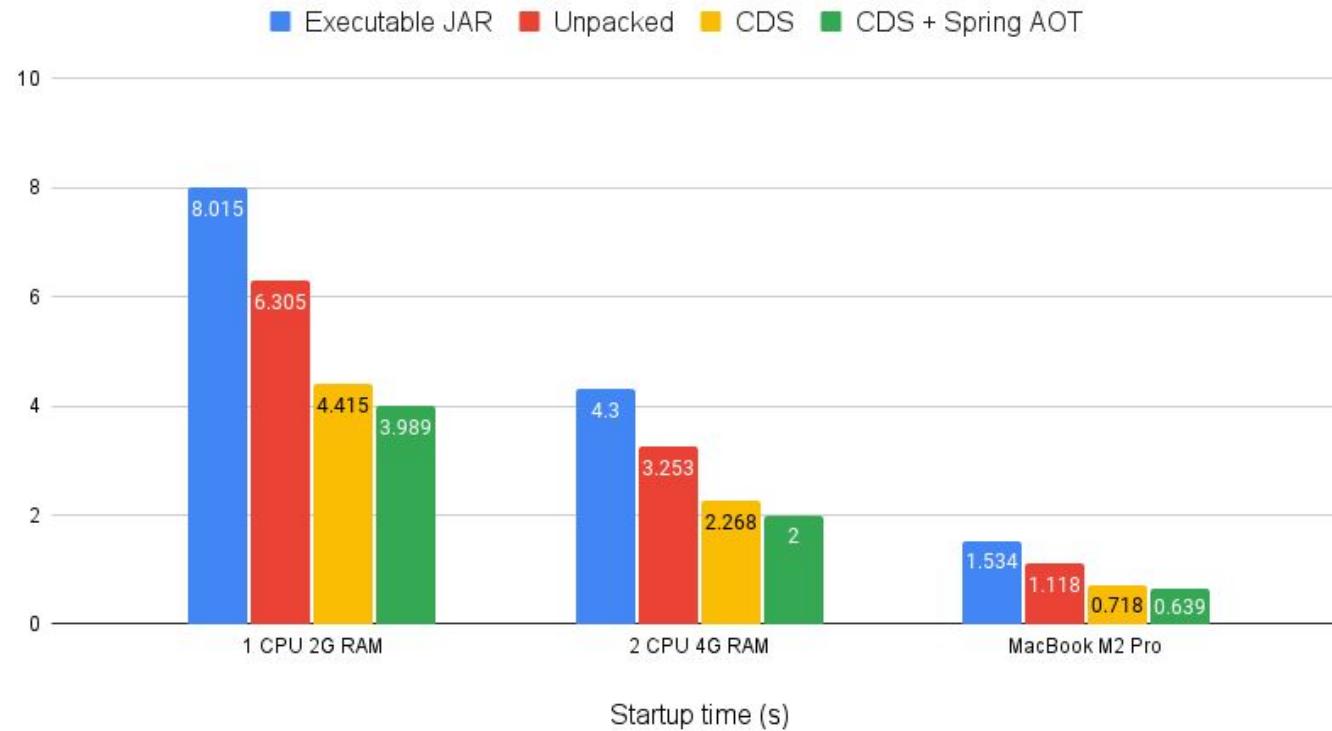
OpenJDK Startup

Class Data Archive

See "CDS On Spring Framework 6.1"

spring®

Executable JAR, Unpacked, CDS and CDS + Spring AOT

Startup time (s)

spring

OpenJDK Startup

Project Leyden

"CDS on steroids" for Spring deployments on HotSpot
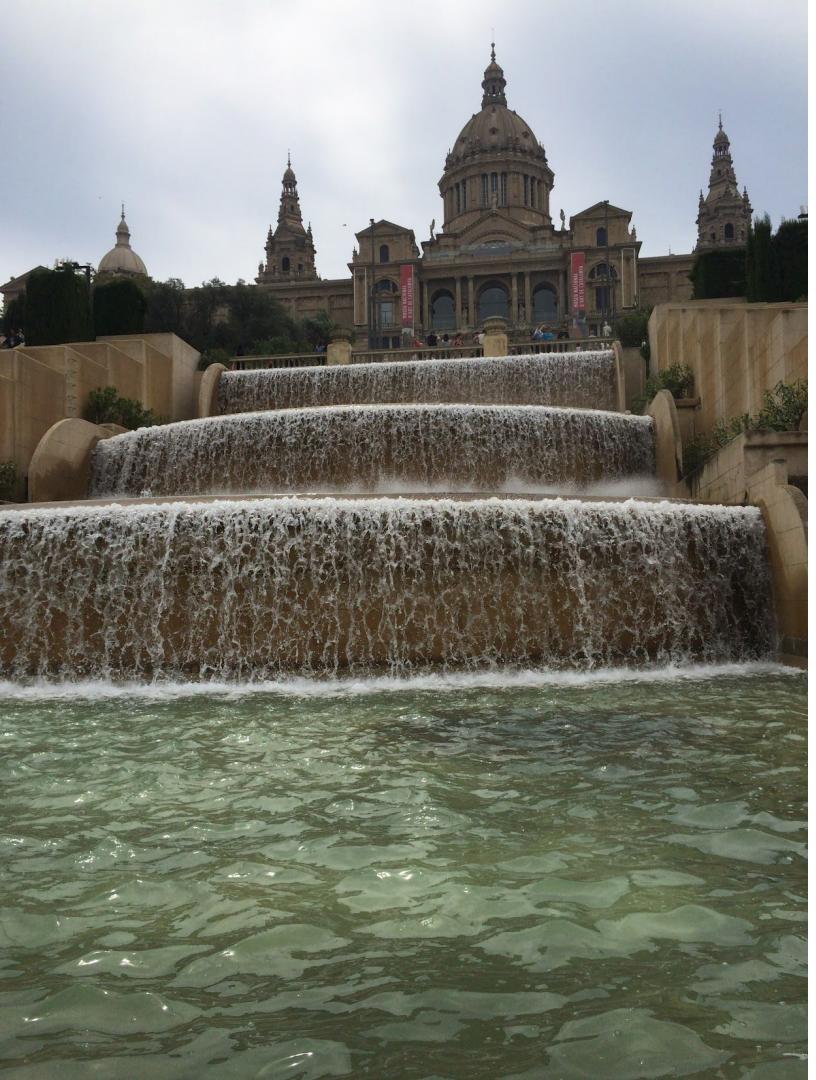
AOT on the JVM

Revisited

For HotSpot, CRaC, Leyden

spring®

Q/A

spring®

Tanzu Academy

spring®

Thank you

Contact me at @snicoll[@mastodon.online]