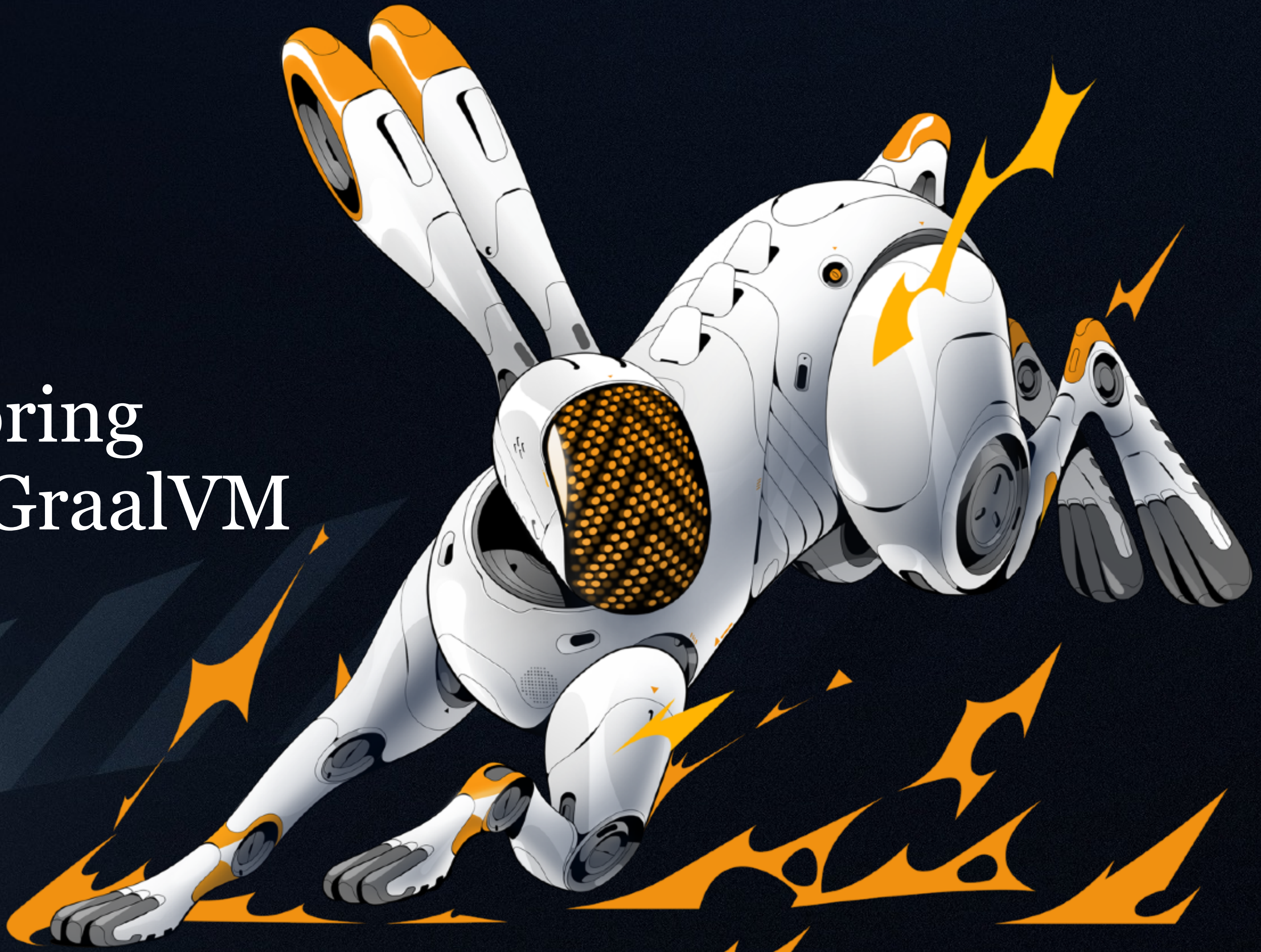
GraalVM™

Fast And Lightweight Spring Boot Applications With GraalVM



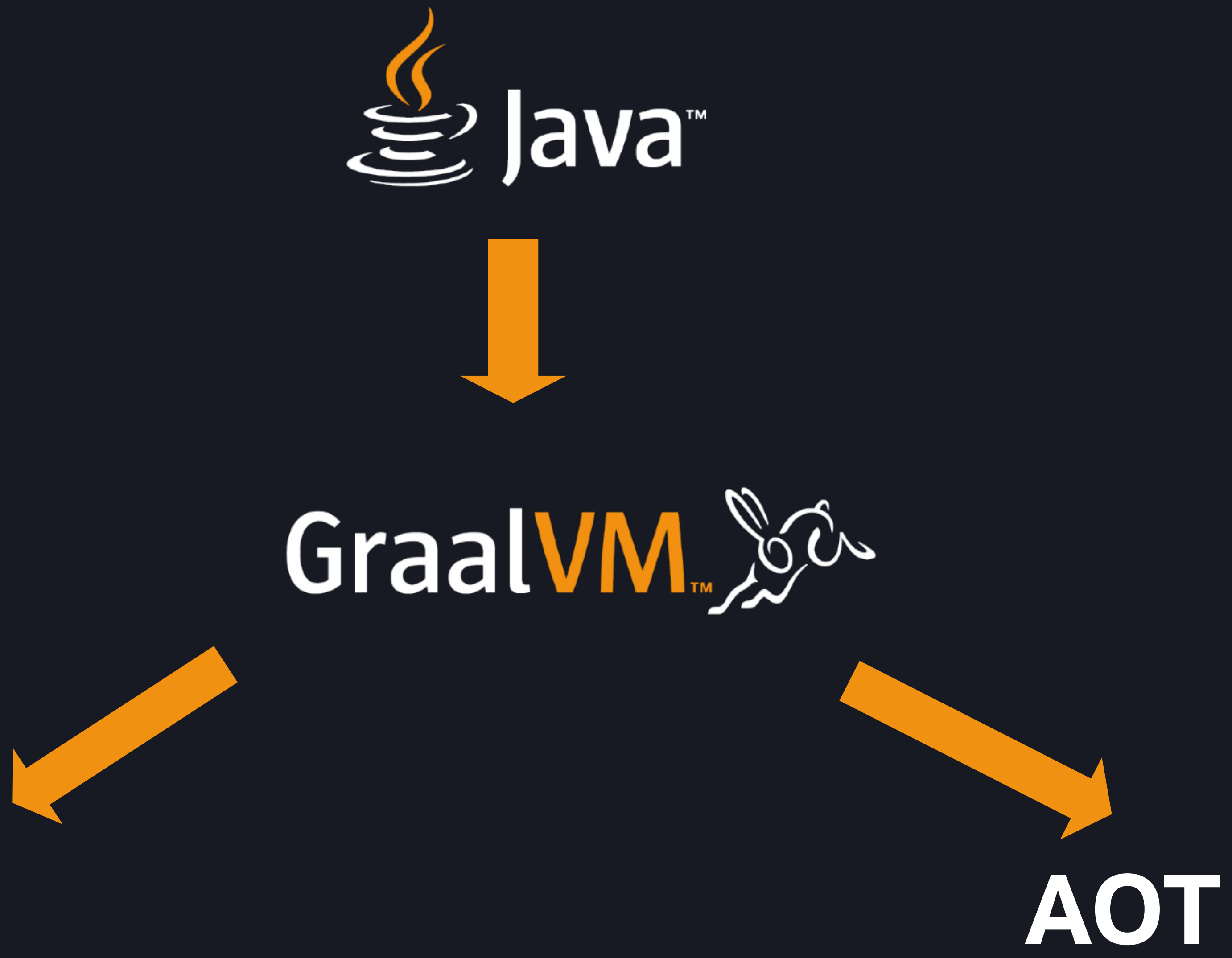
Alina Yurenko

Microsoft JDConf 2024

GraalVM and Native Image 🚀

GraalVM™

An advanced JDK with ahead-of-time
Native Image compilation



Java™

GraalVM™

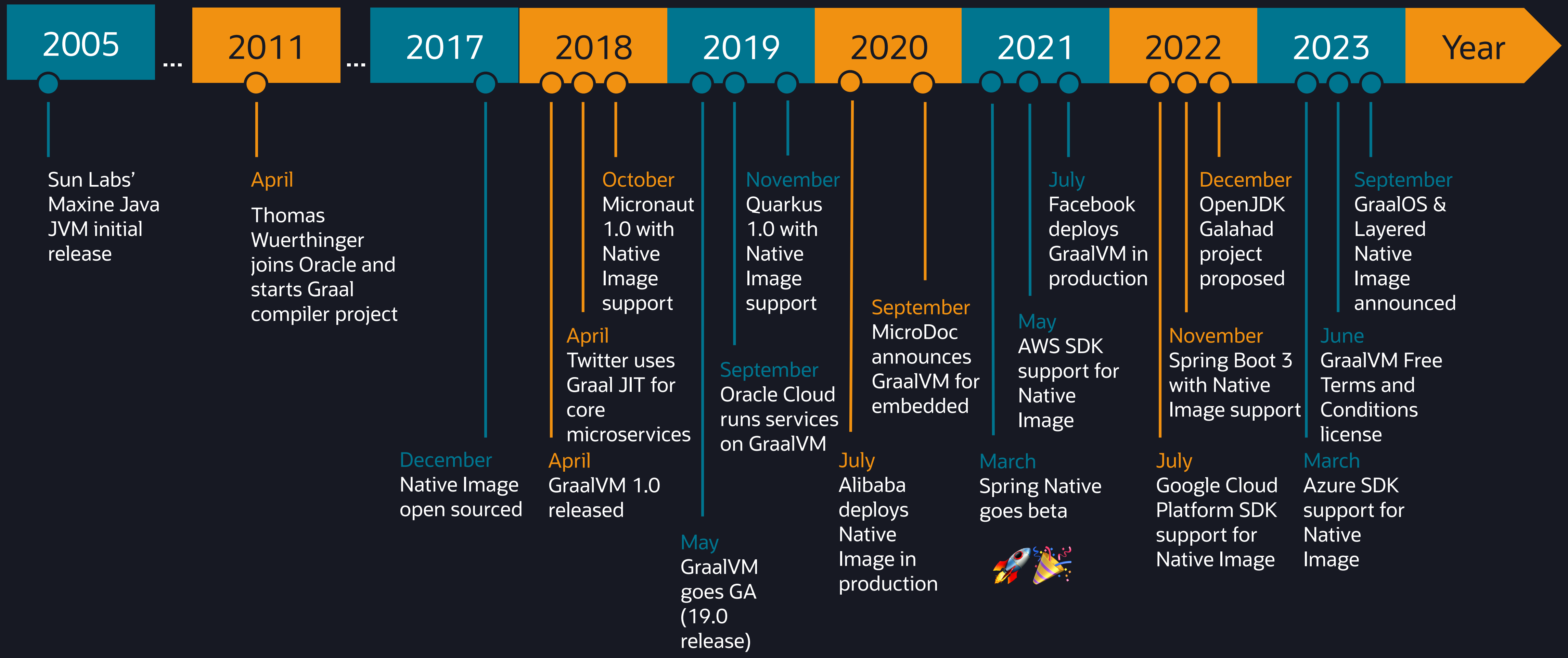
JIT

```
java MyMainClass
```

AOT

```
native-image MyMainClass
./mymainclass
```

Graal History



Sun Labs' Maxine Java JVM initial release

April
Thomas Wuerthinger joins Oracle and starts Graal compiler project

December
Native Image open sourced

April
GraalVM 1.0 released

April
Twitter uses Graal JIT for core microservices

October
Micronaut 1.0 with Native Image support

May
GraalVM goes GA (19.0 release)

September
Oracle Cloud runs services on GraalVM

November
Quarkus 1.0 with Native Image support

July
Alibaba deploys Native Image in production

September
MicroDoc announces GraalVM for embedded

March
Spring Native goes beta

🚀🎉

May
AWS SDK support for Native Image

July
Facebook deploys GraalVM in production

July
Google Cloud Platform SDK support for Native Image

November
Spring Boot 3 with Native Image support

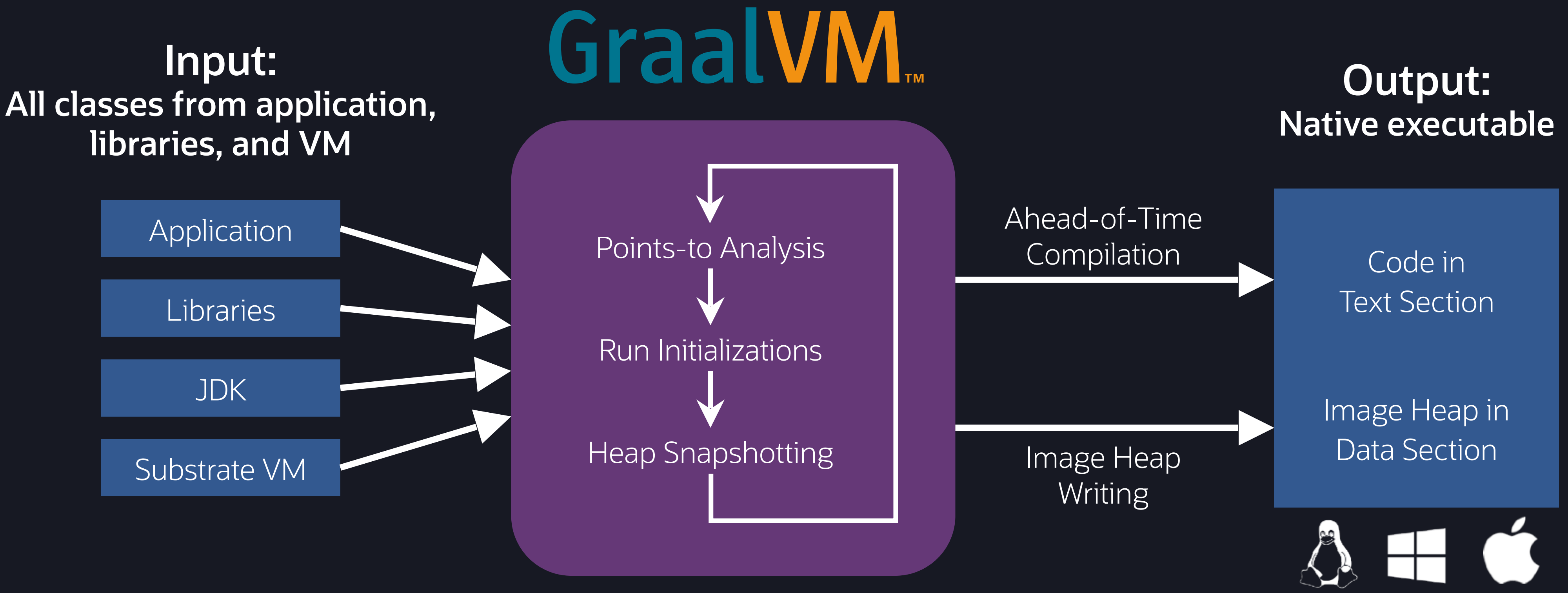
December
OpenJDK Galahad project proposed

March
Azure SDK support for Native Image

June
GraalVM Free Terms and Conditions license

September
GraalOS & Layered Native Image announced

GraalVM Native Image AOT Compilation



Input:
All classes from application, libraries, and VM

Application

Libraries

JDK

Substrate VM

GraalVM™

Points-to Analysis

Run Initializations

Heap Snapshotting

Ahead-of-Time Compilation

Image Heap Writing

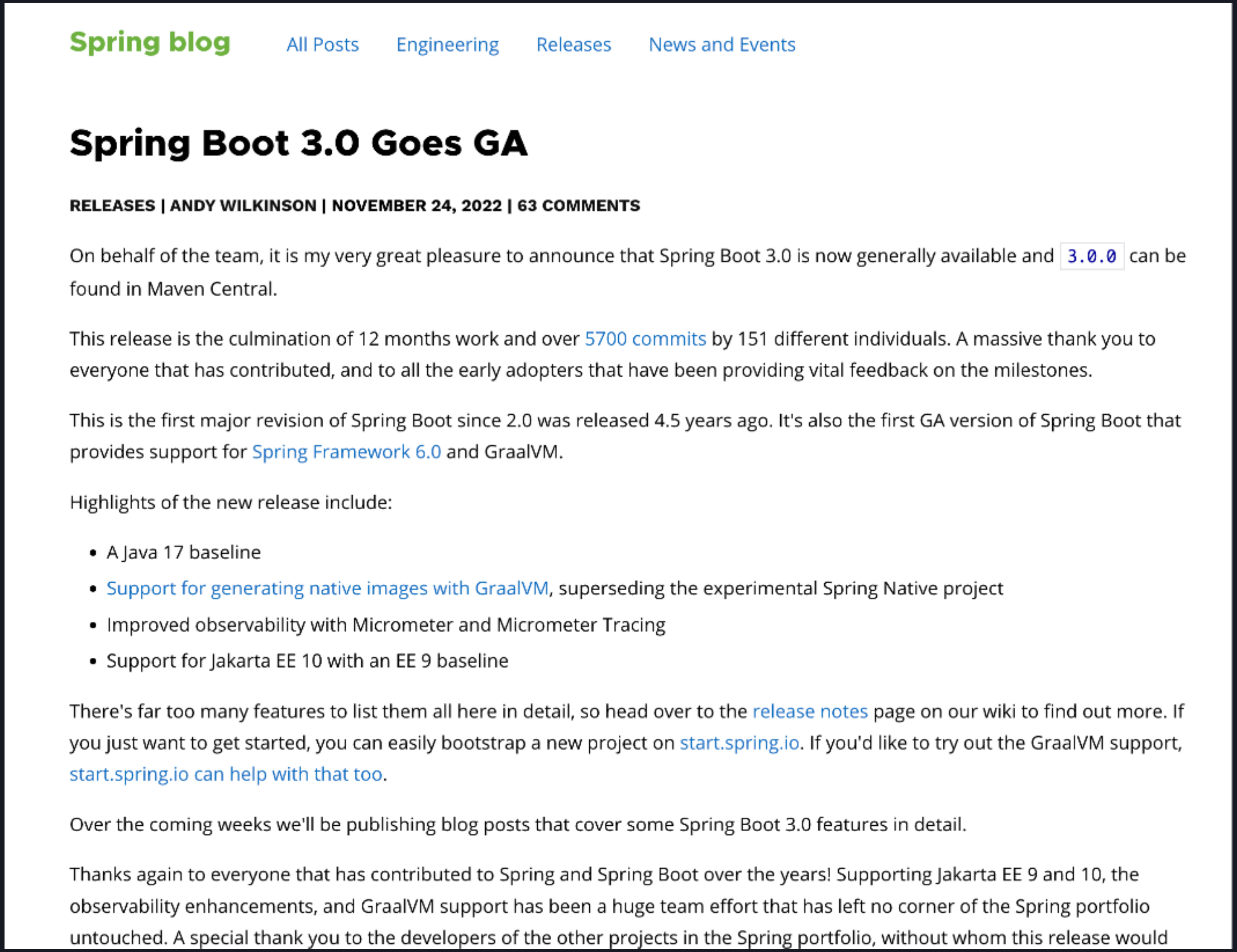
Output:
Native executable

Code in Text Section

Image Heap in Data Section

GraalVM meets Spring Boot! 🎉

Spring Boot and GraalVM



Spring Boot 3.0 Goes GA

RELEASES | ANDY WILKINSON | NOVEMBER 24, 2022 | 63 COMMENTS

On behalf of the team, it is my very great pleasure to announce that Spring Boot 3.0 is now generally available and `3.0.0` can be found in Maven Central.

This release is the culmination of 12 months work and over 5700 commits by 151 different individuals. A massive thank you to everyone that has contributed, and to all the early adopters that have been providing vital feedback on the milestones.

This is the first major revision of Spring Boot since 2.0 was released 4.5 years ago. It's also the first GA version of Spring Boot that provides support for Spring Framework 6.0 and GraalVM.

Highlights of the new release include:

- A Java 17 baseline
- Support for generating native images with GraalVM, superseding the experimental Spring Native project
- Improved observability with Micrometer and Micrometer Tracing
- Support for Jakarta EE 10 with an EE 9 baseline

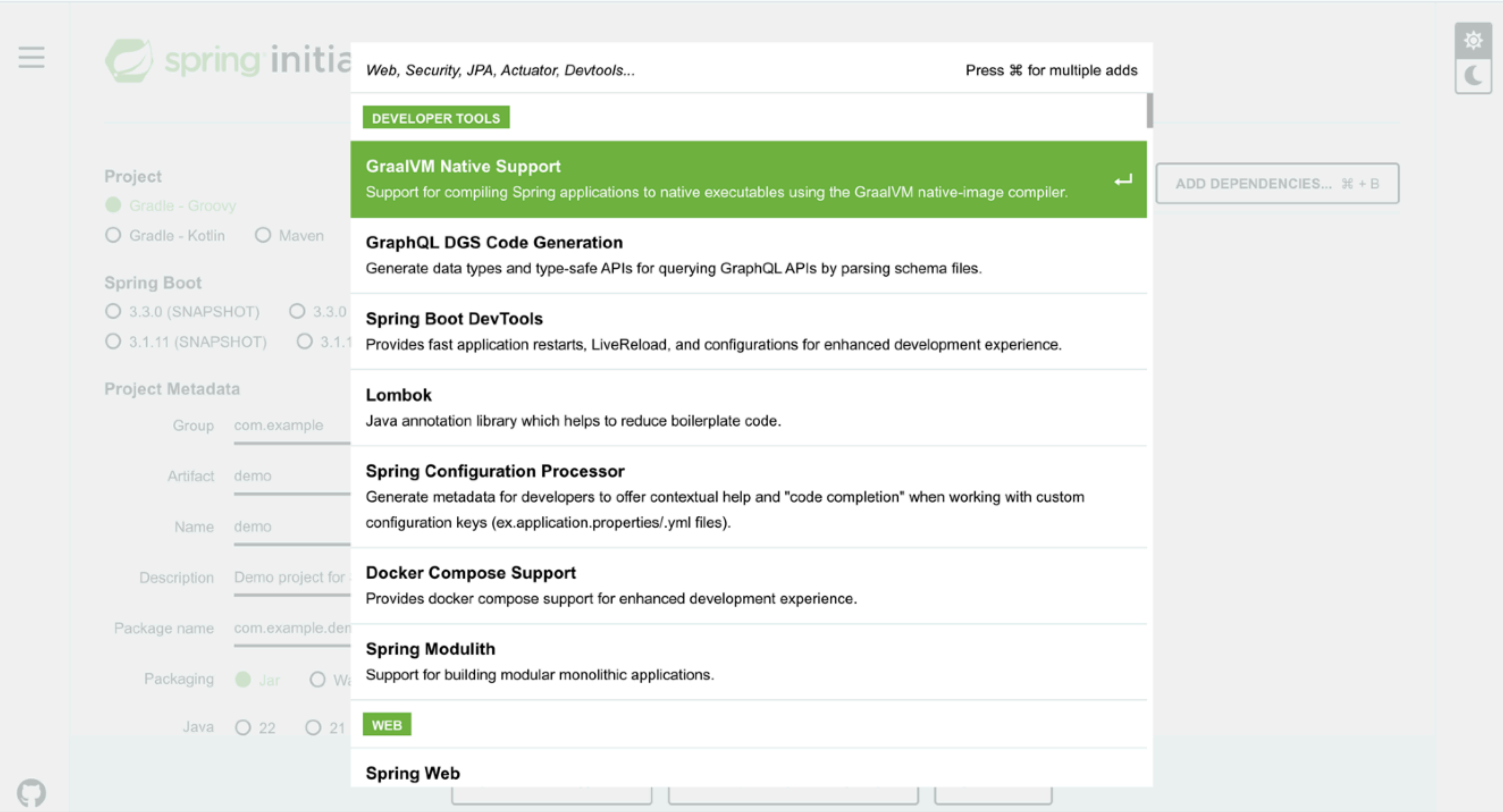
There's far too many features to list them all here in detail, so head over to the release notes page on our wiki to find out more. If you just want to get started, you can easily bootstrap a new project on start.spring.io. If you'd like to try out the GraalVM support, start.spring.io can help with that too.

Over the coming weeks we'll be publishing blog posts that cover some Spring Boot 3.0 features in detail.

Thanks again to everyone that has contributed to Spring and Spring Boot over the years! Supporting Jakarta EE 9 and 10, the observability enhancements, and GraalVM support has been a huge team effort that has left no corner of the Spring portfolio untouched. A special thank you to the developers of the other projects in the Spring portfolio, without whom this release would

spring.io/blog/2022/11/24/spring-boot-3-0-goes-ga

Spring Boot and GraalVM



Project

● Gradle - Groovy
○ Gradle - Kotlin ○ Maven

Spring Boot

○ 3.3.0 (SNAPSHOT) ○ 3.3.0
○ 3.1.11 (SNAPSHOT) ○ 3.1.1

Project Metadata

Group com.example

Artifact demo

Name demo

Description Demo project for

Package name com.example.den

Packaging ● Jar ○ Wa

Java ○ 22 ○ 21

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.

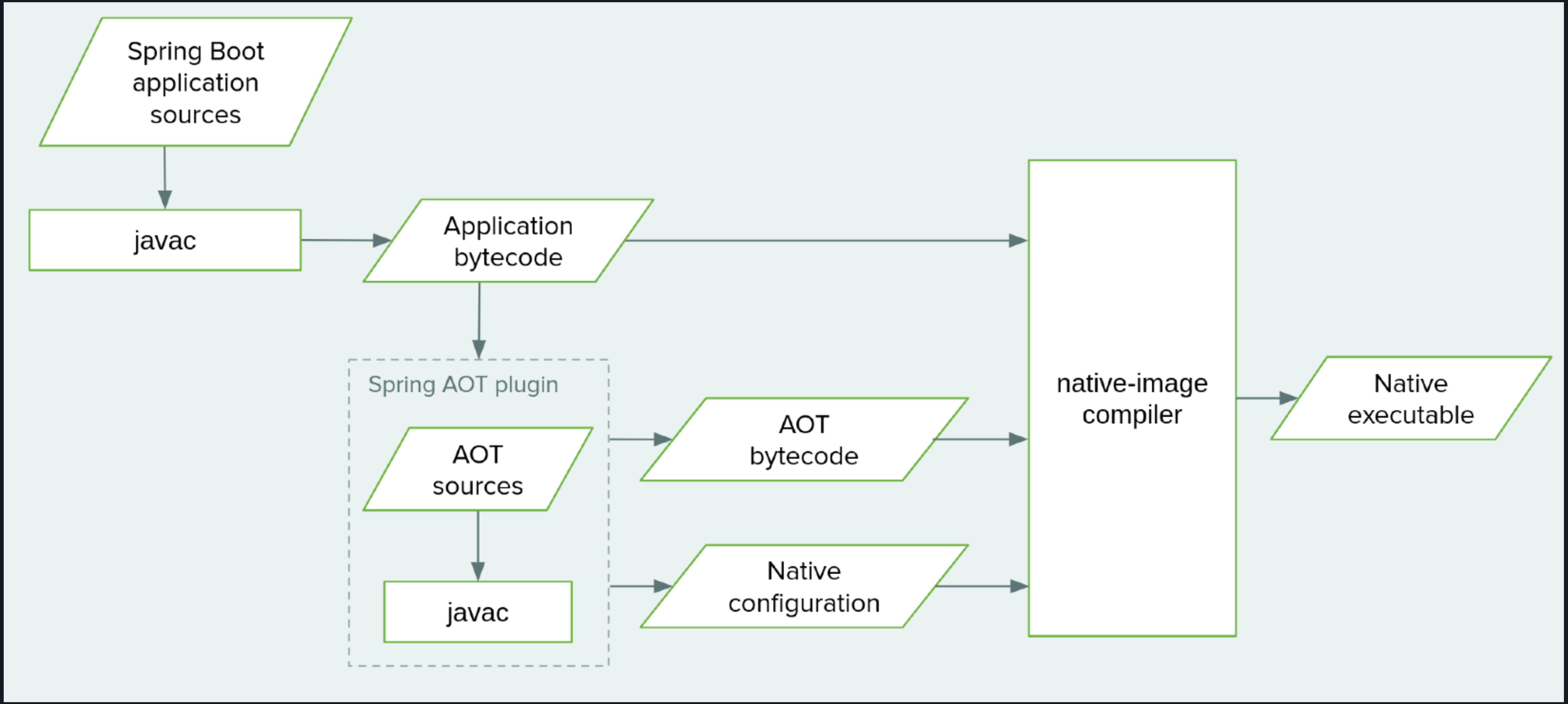
Spring Modulith
Support for building modular monolithic applications.

WEB

Spring Web

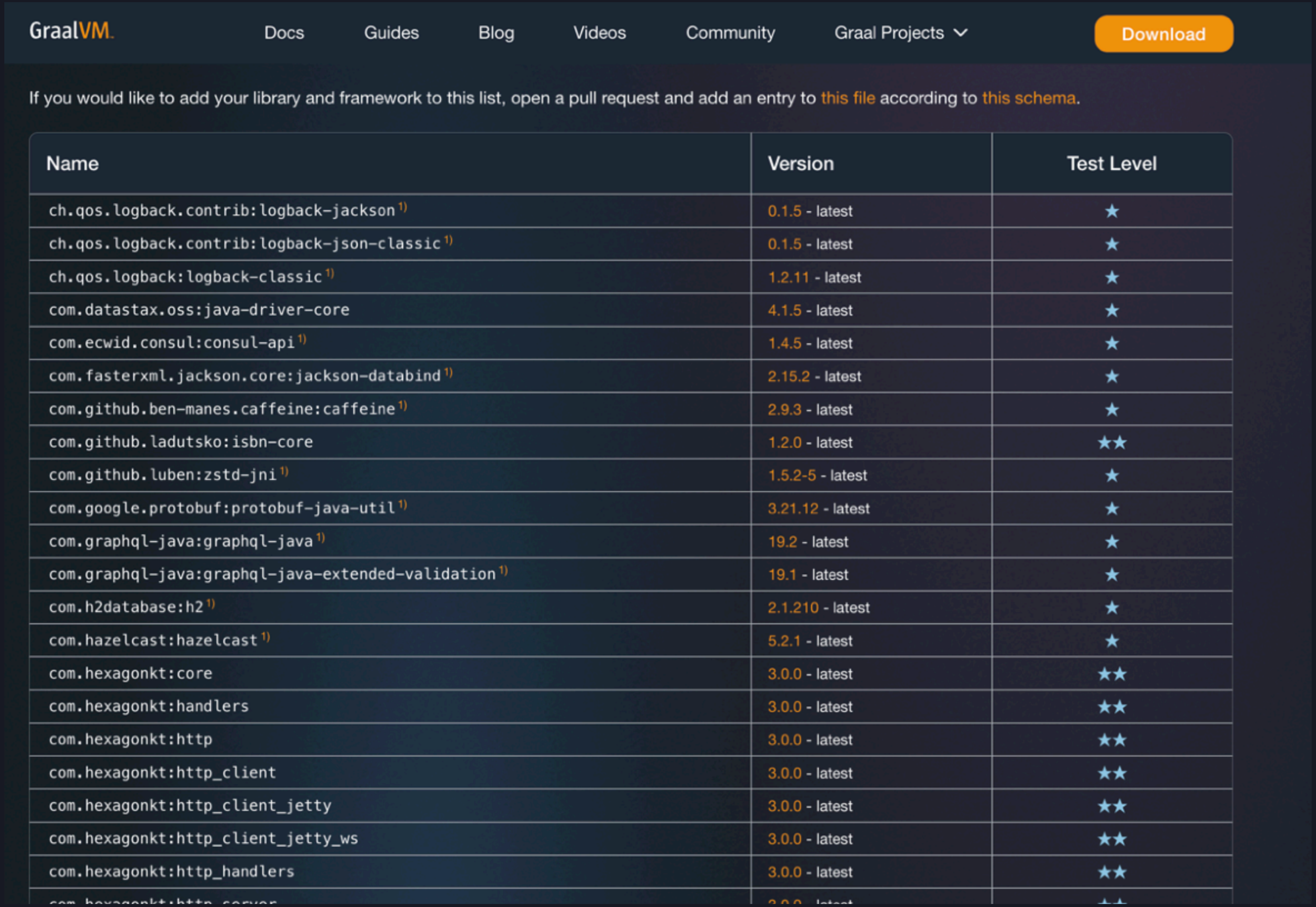
start.spring.io

Spring AOT Engine



spring.io/blog/2021/12/09/new-aot-engine-brings-spring-native-to-the-next-level

Ready for GraalVM Native Image



If you would like to add your library and framework to this list, open a pull request and add an entry to this file according to this schema.

Name	Version	Test Level
ch.qos.logback.contrib:logback-jackson[1]	0.1.5 - latest	★
ch.qos.logback.contrib:logback-json-classic[1]	0.1.5 - latest	★
ch.qos.logback:logback-classic[1]	1.2.11 - latest	★
com.datastax.oss:java-driver-core	4.1.5 - latest	★
com.ecwid.consul:consul-api[1]	1.4.5 - latest	★
com.fasterxml.jackson.core:jackson-databind[1]	2.15.2 - latest	★
com.github.ben-manes.caffeine:caffeine[1]	2.9.3 - latest	★
com.github.ladutsko:isbn-core	1.2.0 - latest	★★
com.github.luben:zstd-jni[1]	1.5.2-5 - latest	★
com.google.protobuf:protobuf-java-util[1]	3.21.12 - latest	★
com.graphql-java:graphql-java[1]	19.2 - latest	★
com.graphql-java:graphql-java-extended-validation[1]	19.1 - latest	★
com.h2database:h2[1]	2.1.210 - latest	★
com.hazelcast:hazelcast[1]	5.2.1 - latest	★
com.hexagonkt:core	3.0.0 - latest	★★
com.hexagonkt:handlers	3.0.0 - latest	★★
com.hexagonkt:http	3.0.0 - latest	★★
com.hexagonkt:http_client	3.0.0 - latest	★★
com.hexagonkt:http_client_jetty	3.0.0 - latest	★★
com.hexagonkt:http_client_jetty_ws	3.0.0 - latest	★★
com.hexagonkt:http_handlers	3.0.0 - latest	★★
com.hexagonkt:http_server	3.0.0 - latest	★★

graalvm.org/native-image/libraries-and-frameworks

GraalVM Native Image & Unit

@EnabledInNativeImage

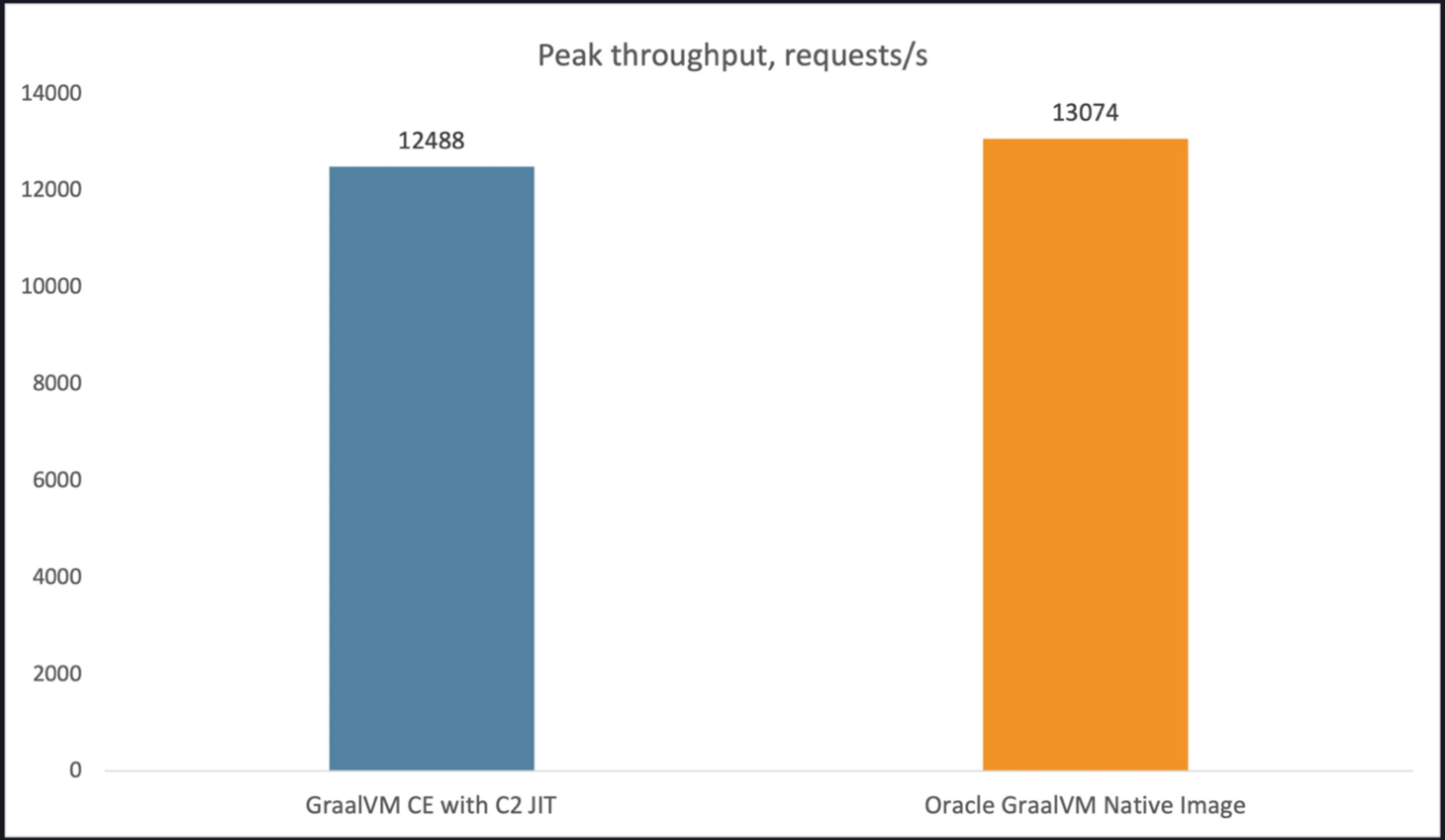
- used to signal that the annotated test class or test method is only *enabled* when executing within GraalVM native images
- when applied at the class level, all test methods within that class will be enabled within a native image

@DisabledInNativeImage

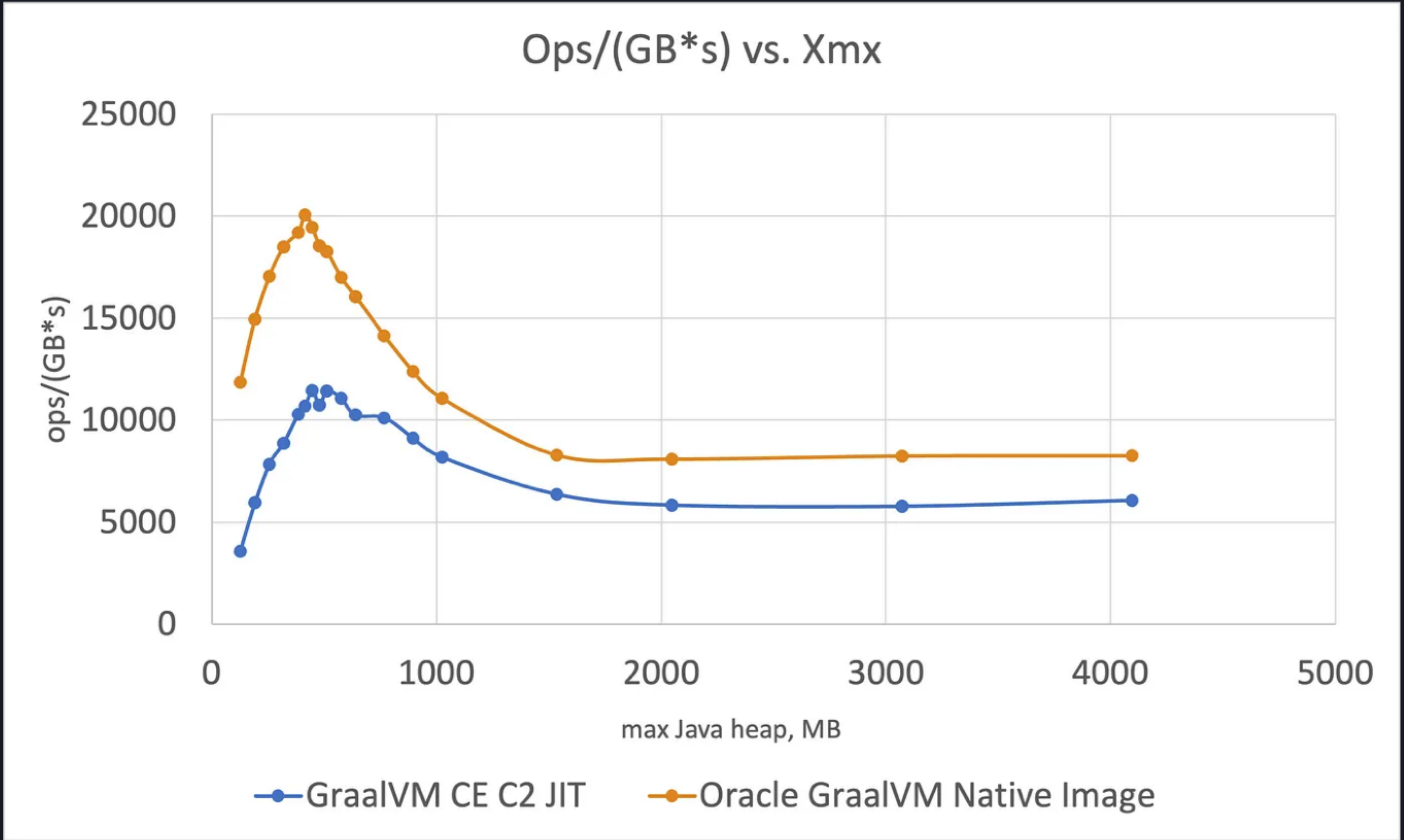
- used to signal that the annotated test class or test method is only *disabled* when executing within a GraalVM native image.

Demo 🚀

Spring PetClinic on Oracle GraalVM - Peak Throughput



Spring PetClinic on Oracle GraalVM - Memory Efficiency



Spring PetClinic Performance on Oracle GraalVM

	GraalVM CE with C2 JIT	**Oracle GraalVM Native Image**	
Memory Usage (max RSS)	1,029 MB	641 MB	**-38% lower**
Peak throughput	11,066 req/s	11,902 req/s	**+8% higher**
Throughput per memory	12,488 req/(GB*s)	18,569 req/(GB*s)	**+49% better**
Tail latency (P99)	7.2ms	5.15ms	**-28% lower**
Startup	7,090ms	210ms	**34x faster**

GraalVM Native Image—Ideal for Cloud Native Applications

Fast Start & Scale

Lower Resource Usage

Predictable Performance

Improved Security

Compact Packaging

MICRONAUT
spring
QUARKUS
helidon.io

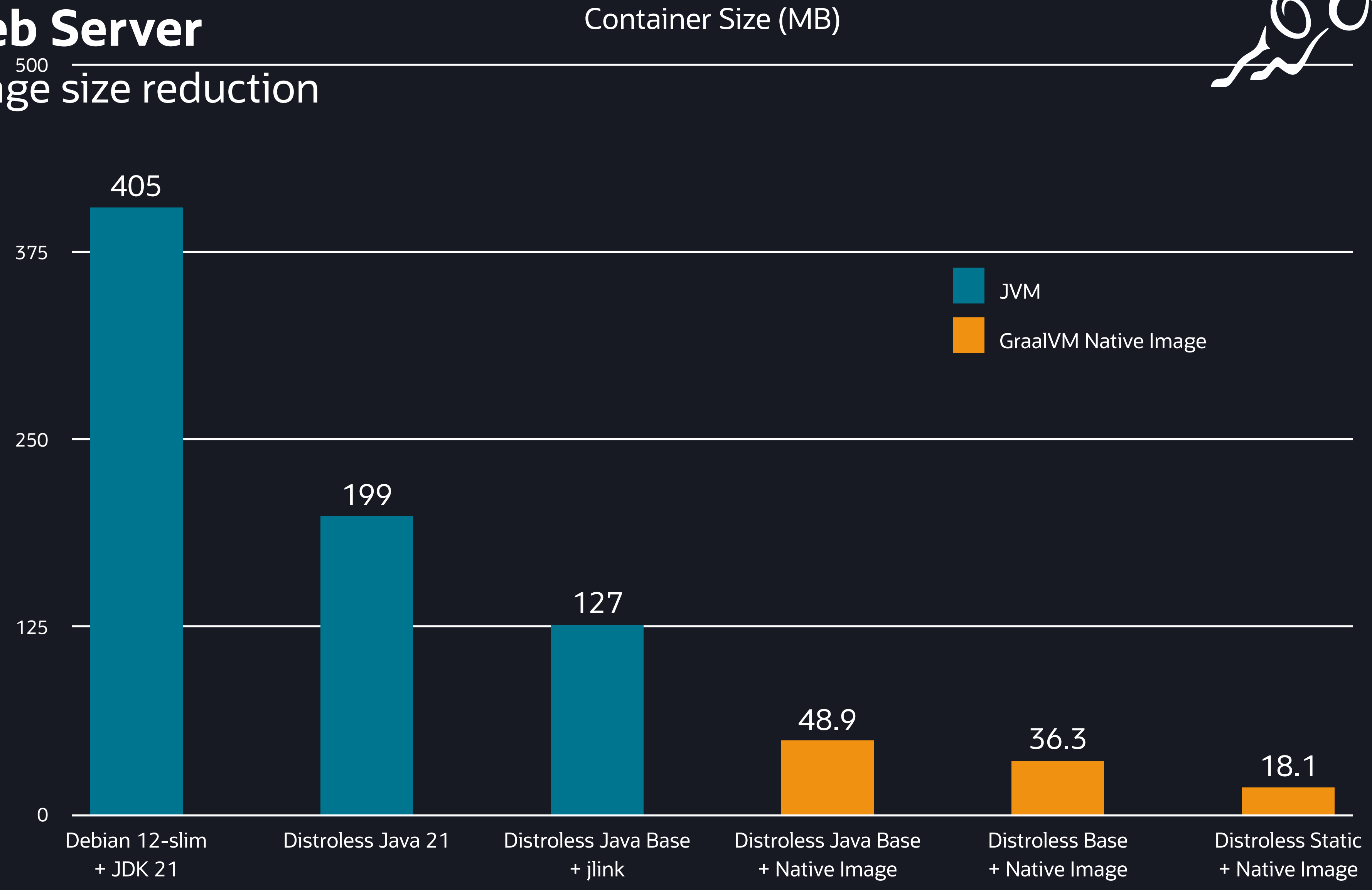
Azure
AWS
GCP
OCI

Supported

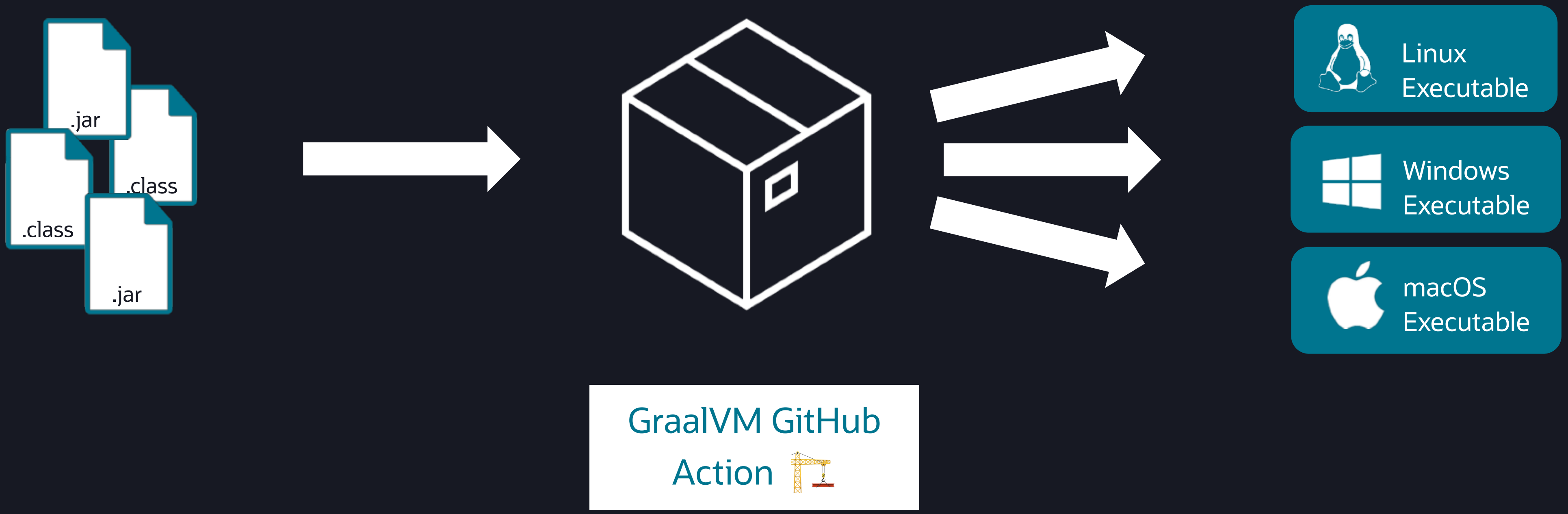
JDK’s Simple Web Server

95.5% container image size reduction

Compact Packaging



Cross-Platform Builds on GitHub Actions



.jar

.class

.class

.jar

GraalVM GitHub Action 🏗️

Linux Executable

Windows Executable

macOS Executable

Reduced Attack Surface

Improved Security

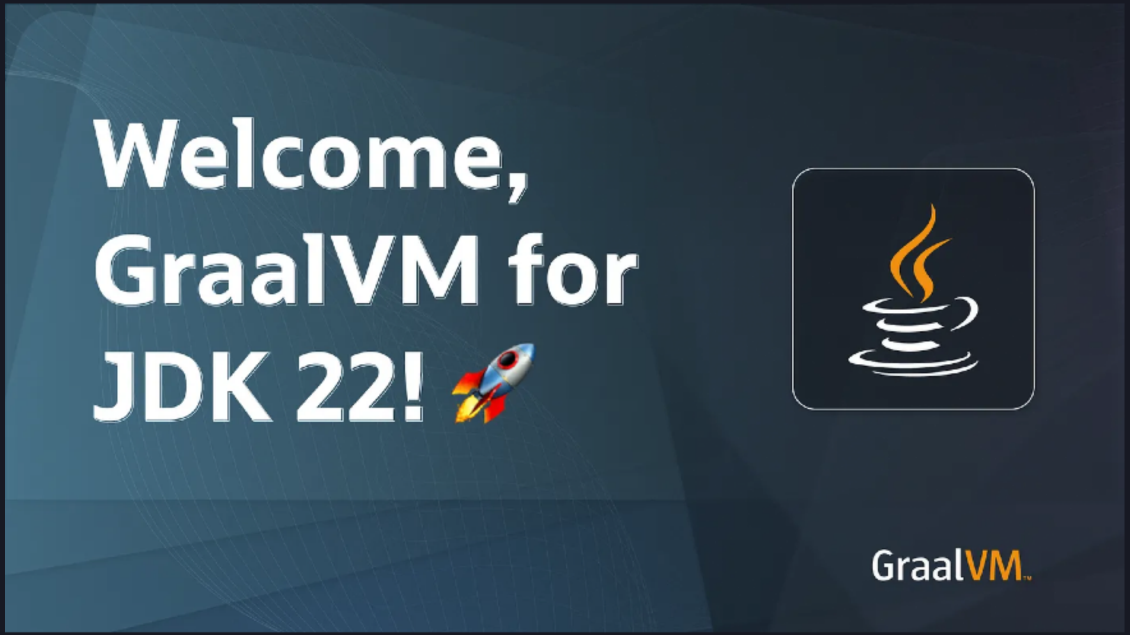
- Reduced attack surface area due to dead code removal—unused classes, methods, and fields not included in executable
- SBOM supporting industry standards
  Embedded in executables
  CycloneDX format
- Not vulnerable to deserialization attacks via class loading—executable includes only required and specified classes
- Not vulnerable to JIT compiler attacks
  all code is AOT compiled

What’s next for GraalVM

GraalVM for JDK 22 🚀

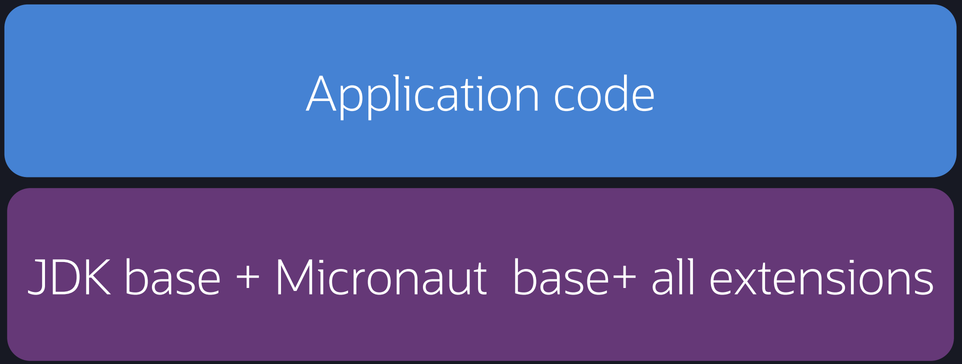
- Java 22 features
- The fastest GraalVM yet :)
- Developer experience improvements

Learn more: medium.com/graalvm



Layered Native Images

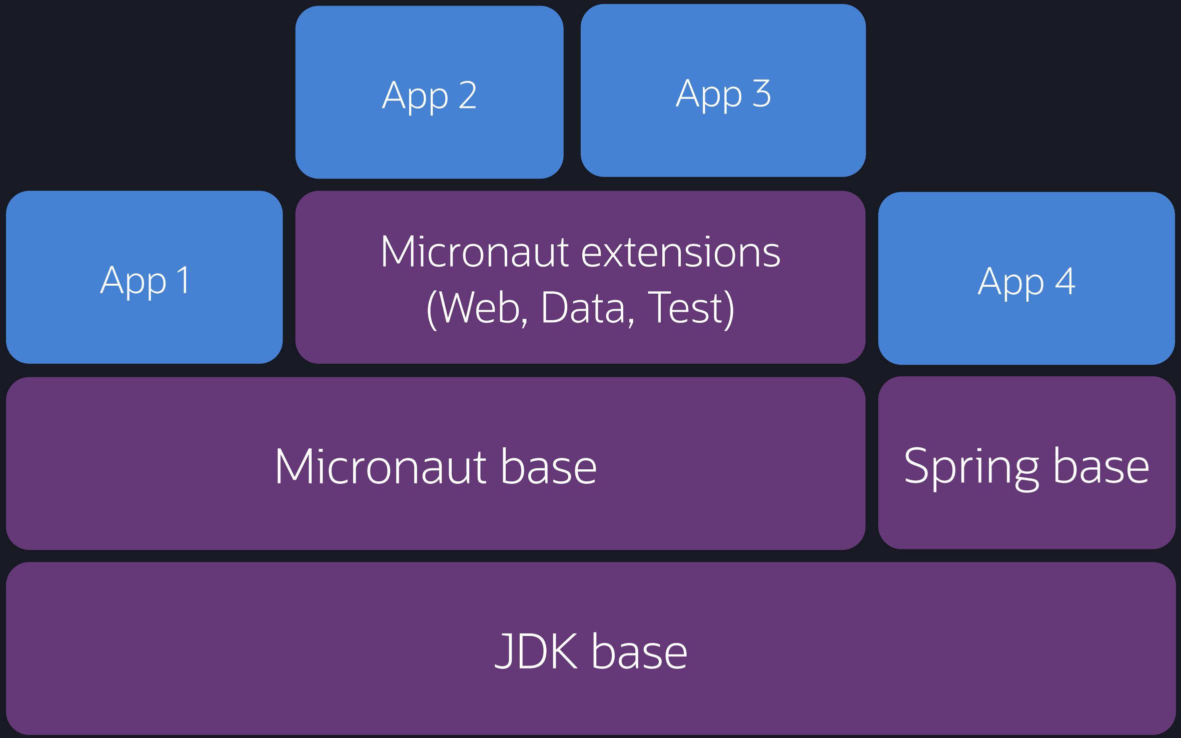
Development: fast recompilation 🚀



Application code

JDK base + Micronaut base+ all extensions

Deployment: resources sharing ☁️



App 2

App 3

App 1

Micronaut extensions (Web, Data, Test)

App 4

Micronaut base

Spring base

JDK base

GraalOS—Advanced cloud native application deployment platform

Runs applications as small, fast GraalVM Native Image compiled machine executables

Fast Start

GraalOS applications start fast with virtually no cold start cost

Low Latency

Excellent 99th percentile latency makes GraalOS applications highly responsive

Reduced Memory

GraalOS applications require significantly less memory resulting in reduced operating costs

Run On Demand

GraalOS applications are automatically suspended and resumed on demand—with no idle cost

Applications, not Containers

GraalOS uses the latest advances in x86 and AArch64 processor architectures for hardware enforced application isolation without containers

Cloud Native

With support for stateful and stateless services and functions, GraalOS is ideal for cloud native applications

Recommendations

- Migrate 🚀
  - Move to Spring Boot 3.X
  - If not possible, start with adding Native Build Tools
- Build and deploy 👷‍♀️
  - Build and test on GraalVM as the JVM, build with Native Image closer to the deployment
  - Quick build mode with `-Ob`
  - Use CI/CD systems for deployment and cross-platform builds
- Run faster 🚀
  - PGO
  - Machine Learning PGO
  - G1 GC
  - `-march=native`

Get started with GraalVM 🚀

graalvm.org

docker pull container-registry.oracle.com/graalvm/jdk:22

sdk install java 22-graal

github.com/graalvm/graalvm-demos

Thank you!

@alina_yurenko

ORACLE