



QUARKUS

QUARKUS WORLD TOUR 2022

QUARKUS FOR SPRING
DEVELOPERS

WHO'S ON STAGE TODAY?



ERIC
DEANDREA



22+ years software development experience



Contributor to Open Source projects



Quarkus



Spring Boot



Spring Framework



Spring Security



Spring Session



Build & deliver cloud-native development
training programs



Published author



Avoiding common pitfalls with modern microservices testing

Holly Cummins

Sr. Principal Software Engineer
Red Hat
@holly_cummins



Eric Deandrea

Sr. Principal Developer Advocate
Red Hat
@edeandrea



<https://youtu.be/vYwkDPrzqV8>

Quarkus World Tour 2022 {👋}

AGENDA

- 🤘 INTRO TO QUARKUS
- 🤘 QUARKUS FOR SPRING DEVELOPERS
- 🤘 ADDITIONAL RESOURCES
- 🤘 Q&A
- 🤘 JAM TIME! (ON YOUR OWN)



JAVA DESIGNED FOR A DIFFERENT TIME



Traditional

- *Throughput* at the expense of *footprint*
- *Long running* at expense of *startup speed*
- *Rich, dynamic behaviour* for *mutable systems*

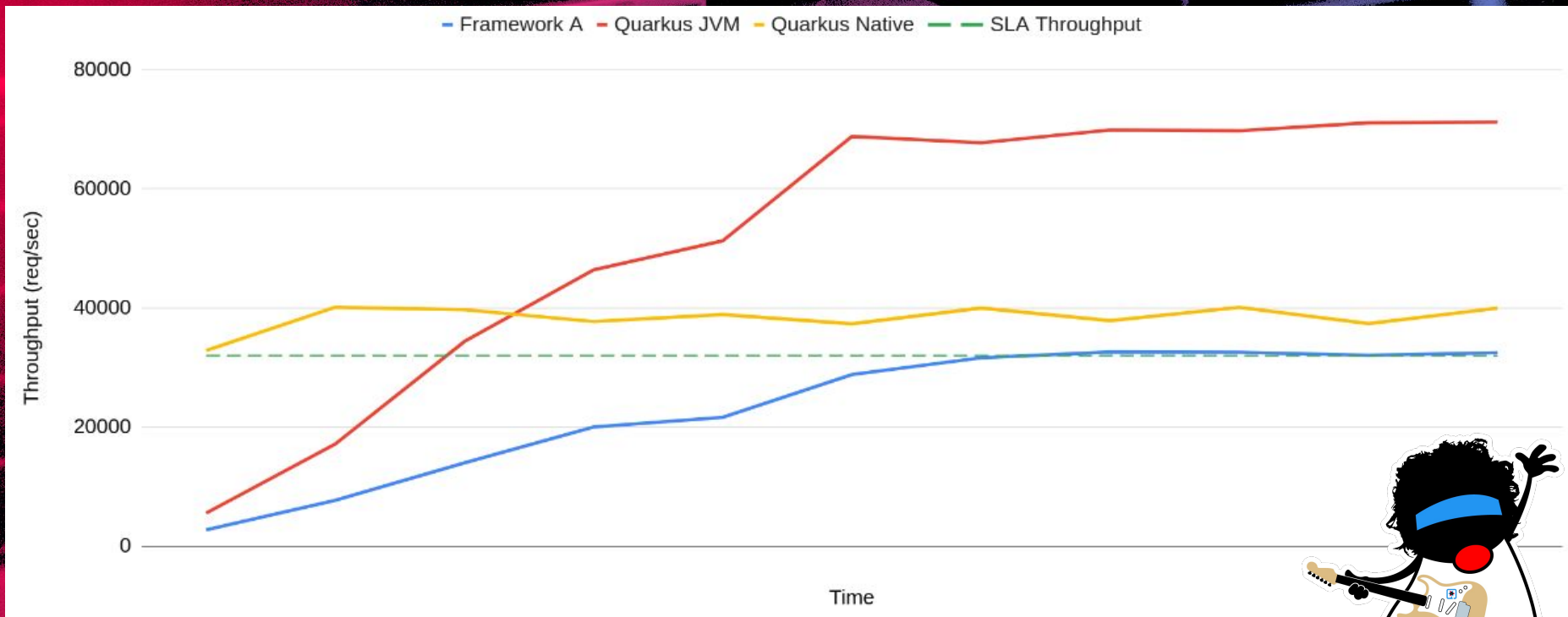


Cloud Native

- *Throughput* solved by *scaling*
- *Ephemeral, immutable systems*
- *Footprint and performance matter*



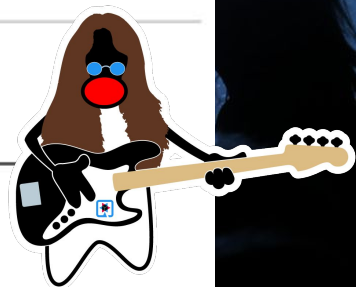
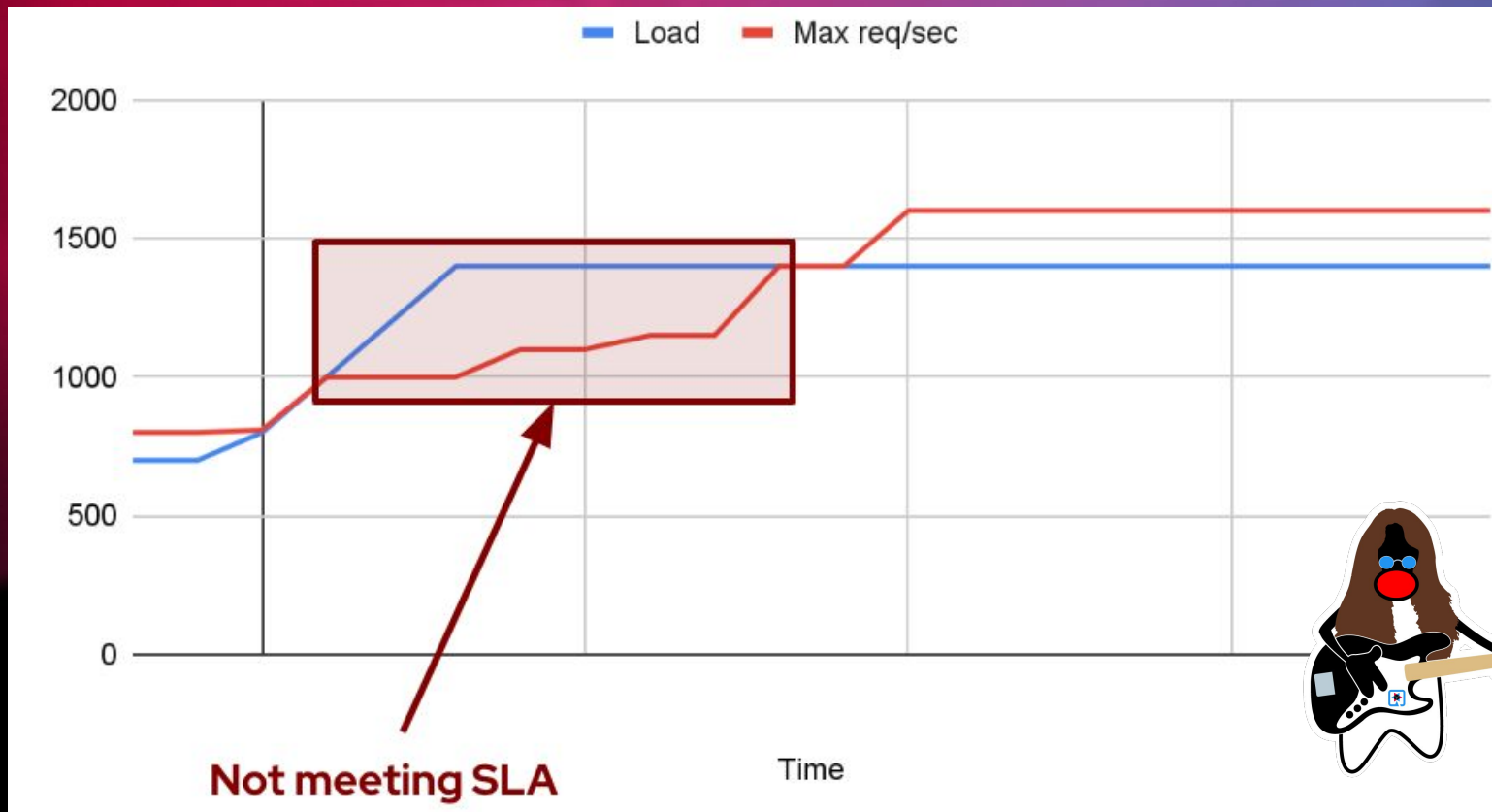
THE WARMUP ISSUE WITH JAVA



<https://youtu.be/bWmugh6wHgE> (first 13 minutes)

Quarkus World Tour 2022 {🎸}

JVM SCALING ON KUBERNETES





QUARKUS

SUPERSONIC, SUBATOMIC JAVA



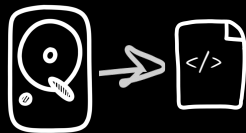
HOW DOES A FRAMEWORK START?

Build Time

Runtime



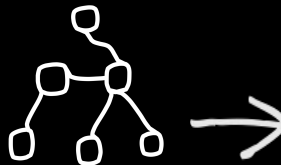
Packaging
(maven,
etc)



Load config file
from file system
Parse it



Classpath scanning
to find
annotated classes
Attempt to load
class to
enable/disable
features



Build its
model of
the world.



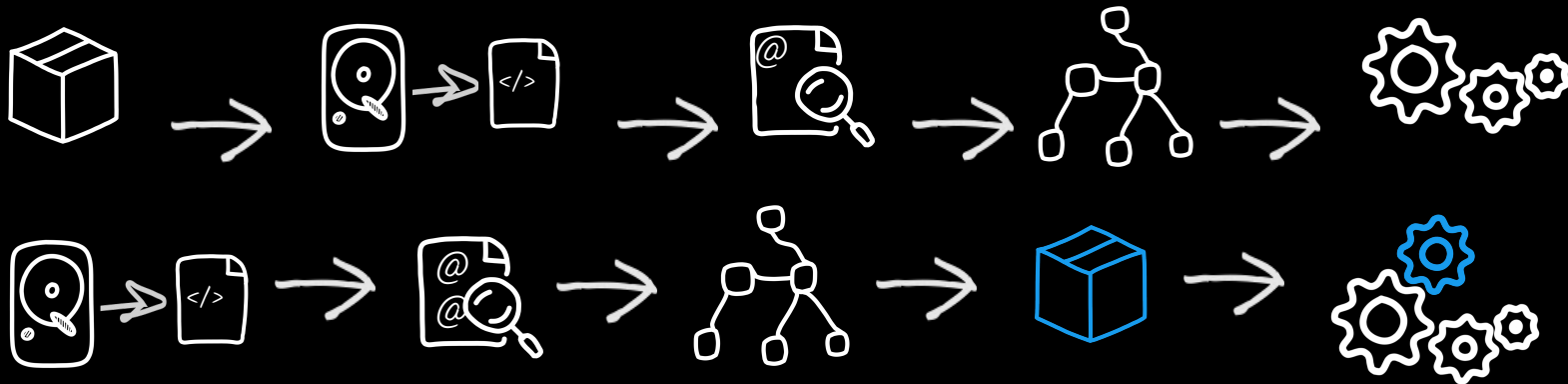
Start the
management
(thread,
pool...)



THE QUARKUS WAY

Build Time

Runtime

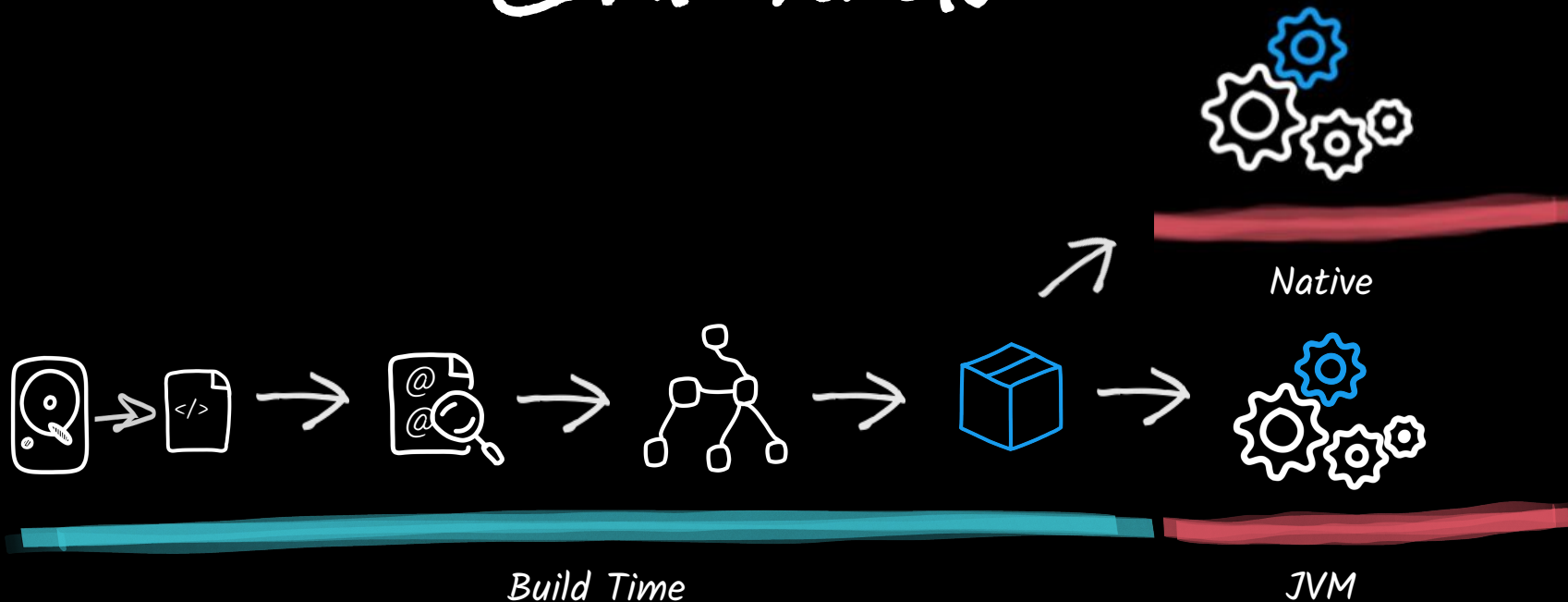


Build Time

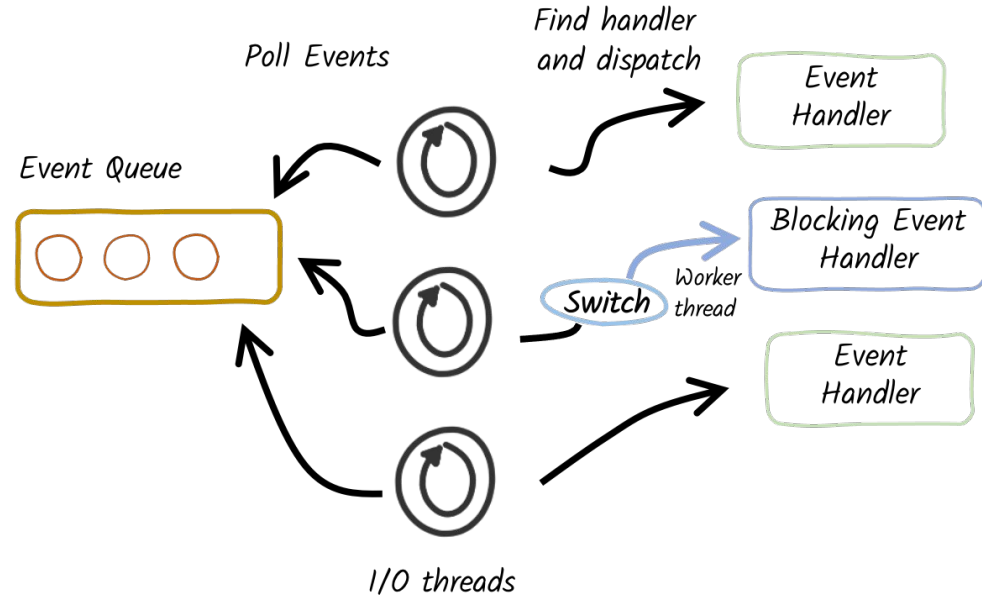
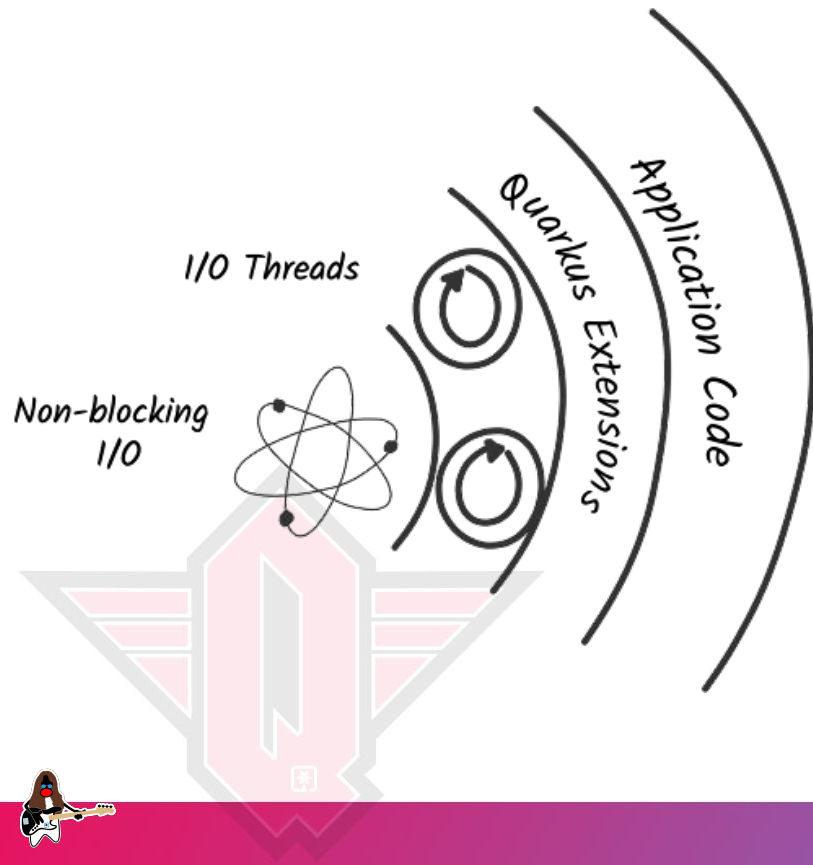
Runtime



THE QUARKUS WAY ENABLES NATIVE COMPILE



UNIFICATION OF IMPERATIVE AND REACTIVE



ENHANCING DEVELOPER JOY



Based on standards



Developer productivity

- Zero-config Live coding
- Developer services
- Continuous testing
- Dev UI
- CLI



Streamlined for the 80% common usages, flexible for the rest



No hassle native executable generation

Wait.
So you just save it,
And your code is running?
And it's Java?!



I know, right?
SUPERSONIC JAVA, FTW!





IT'S
STILL
JAVA!



NATIVE COMPILATION

🤘 Red Hat sits on the GraalVM advisory board

🤘 Red Hat contributes to GraalVM

- Debugging support
- ARM support
- Java Flight Recorder

🤘 Quarkus is part of the GraalVM test suite

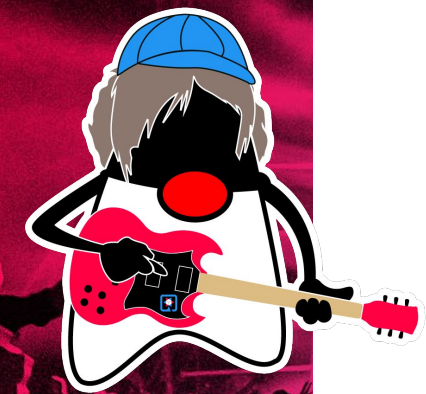
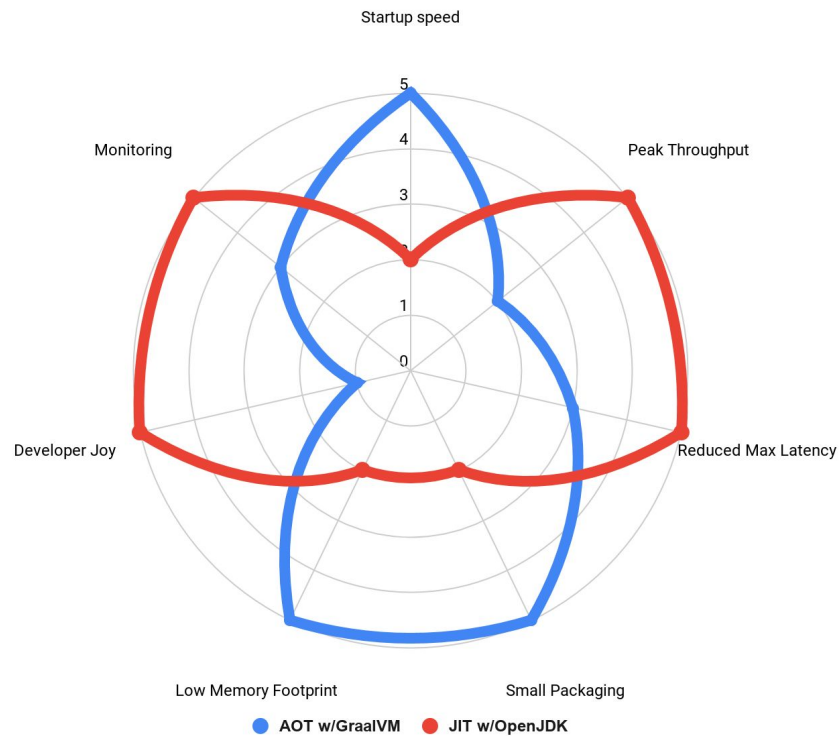


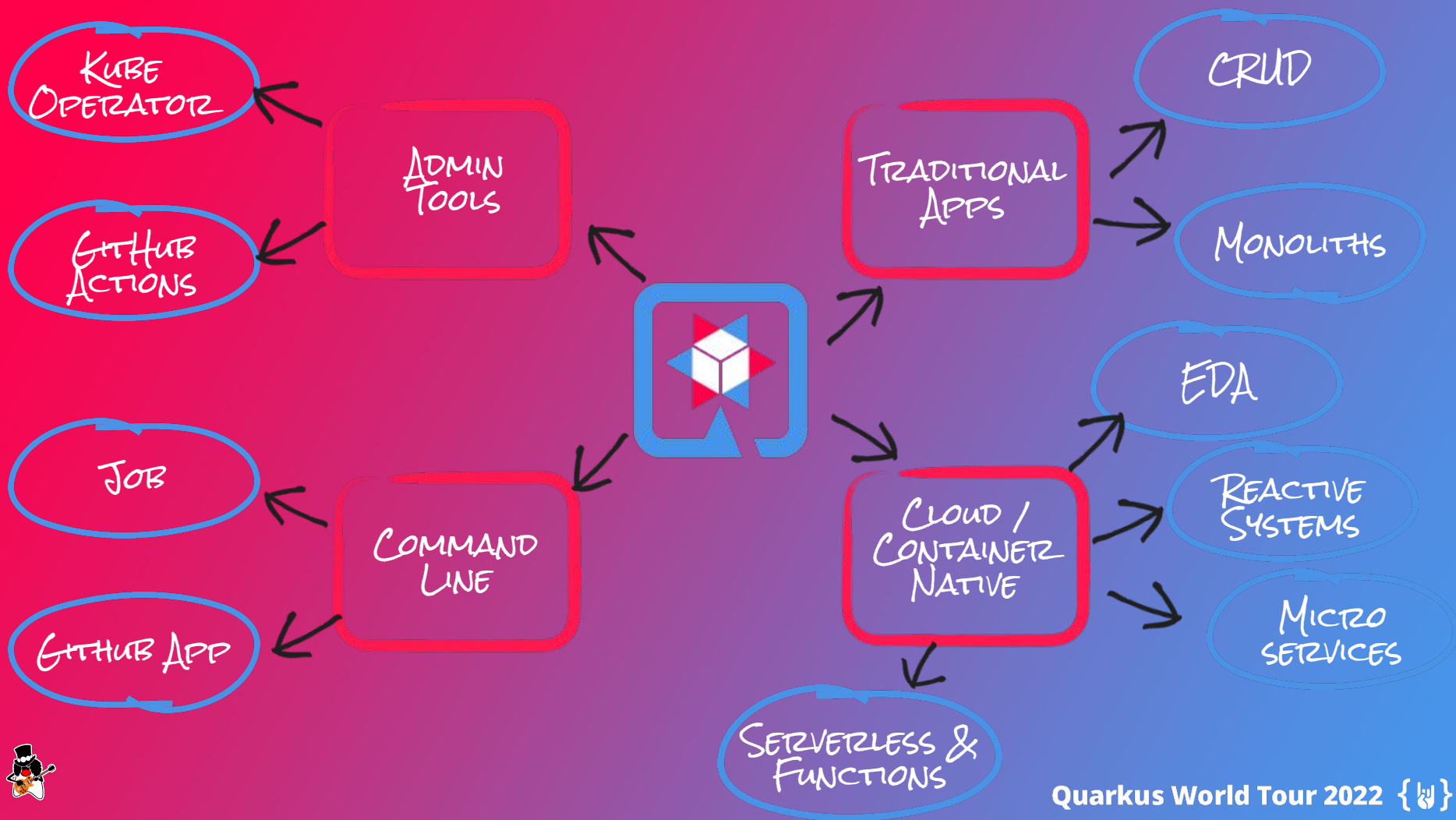
NATIVE COMPILE WITH MANDREL

- 👉 Mandrel: direct downstream distribution of GraalVM
- 👉 Combines best of GraalVM + OpenJDK to offer native builds
- 👉 Makes GraalVM easier to consume
 - 👉 Open Source licensing & maintenance
 - 👉 Consumes patches & CVEs from OpenJDK
- 👉 Near drop-in replacement for GraalVM
 - 👉 Focused on native compilation for Java only
- 👉 No change for user experience with Quarkus
- 👉 No need to pay Oracle for support



AOT/NATIVE-IMAGE VS VM MODE







Quarkus



Quarkus World Tour 2022 {Qu}

- Showcase & explain Quarkus, how it enables modern Java development & the Kubernetes-native experience
- Introduce familiar Spring concepts, constructs, & conventions and how they map to Quarkus
- Equivalent code examples between Quarkus and Spring as well as emphasis on testing patterns & practices
- Chapters devoted to
 - Why the need for Quarkus in the first place?
 - Getting started
 - RESTful applications
 - Persistence
 - Event-driven services
 - Cloud environments, containers, and Kubernetes



<https://red.ht/quarkus-spring-devs>



Quarkus World Tour 2022 {👋}

BUT I ALREADY KNOW SPRING...



Can I do reactive/imperative/blocking?

- When do I need to make that choice?



What if there isn't an extension for my library?



Can I reuse my existing Spring code?

- Spring compatibility extensions? When should I use them?
 - Dependency injection
 - Spring Web/MVC
 - Spring Data JPA
 - Spring Data REST
 - Spring Security
 - Spring Cache
 - Spring Boot Configuration Properties
 - Spring Scheduled
 - Spring Cloud Config Client

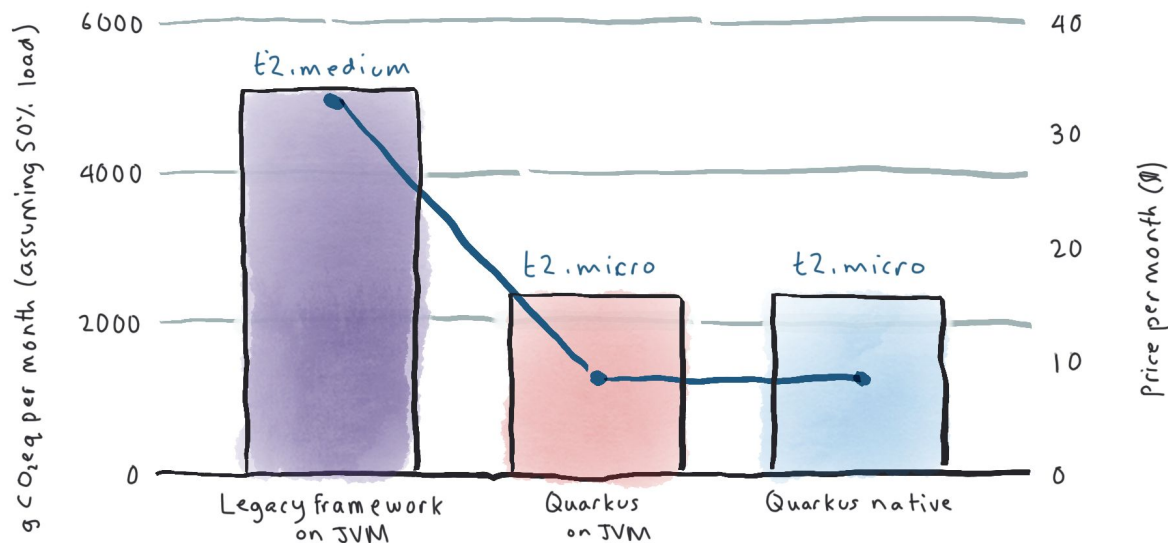


BUT I ALREADY KNOW SPRING...

- 👉 How do I migrate?
 - If doing REST/HTTP (server or client), make sure to use RESTEasy Reactive!
- 👉 How stable is it?
 - There are lots of issues on GitHub...
- 👉 GraalVM compilation time is memory intensive...
- 👉 What if I have my own “meta” framework on “top” of Spring?



THE COST / CARBON IMPACT



Setup / Assumptions

- 👉 AWS (us-east-1)
- 👉 SLA > 99%
- 👉 800 req/sec over 20 days
- 👉 50% load



<https://www.redhat.com/en/resources/greener-java-applications-detail> (published 11/14/22)

<https://hollycummins.com/writing-greener-java-applications>

Quarkus World Tour 2022 {👉}

MIGRATION TOOLKIT FOR APPLICATIONS

"SIMPLIFIES THE MIGRATION OF SPRING APPS TO QUARKUS"

Migration Mandatory

Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
Replace Spring datasource property key/value pairs with Quarkus properties	3	1	Trivial change or 1-1 library swap	3
Replace the Spring Web artifact with Quarkus 'spring-web' extension	1	1	Trivial change or 1-1 library swap	1
Replace the Spring Boot Actuator dependency with Quarkus Smallrye Health extension	1	5	Requires re-design or library change	5
Replace the Spring Parent POM with Quarkus BOM	1	1	Trivial change or 1-1 library swap	1
Replace the Springboot Data JPA artifact with Quarkus 'spring-data-jpa' extension	1	1	Trivial change or 1-1 library swap	1
	7			11

Migration Potential

Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
Component springdoc-openapi-ui requires investigation	1	5	Requires re-design or library change	5
Component spring-boot-starter-test requires investigation	1	5	Requires re-design or library change	5
	2			10

Migration

Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
BOM (pom.xml)	1	0	Info	0
	1			0

- AUTOMATE APPLICATION ANALYSIS
- ESTIMATE LEVEL OF EFFORT
- ACCELERATE CODE TRANSFORMATION & MIGRATION
- INCLUDES RULES FOR DI, METRICS, SECURITY, WEB, SHELL, & MORE

Proprietary
Java EE runtimes

Java EE & JDK
upgrades

JBoss EAP migrations

Spring Boot, Quarkus,
Tomcat (JWS)

Cloud readiness,
containerization

Pluggable:
add your own rules

OracleJDK to OpenJDK Migrations



Command Line
Interface



Web
Console



IDE
Plugins



Maven
Plugin



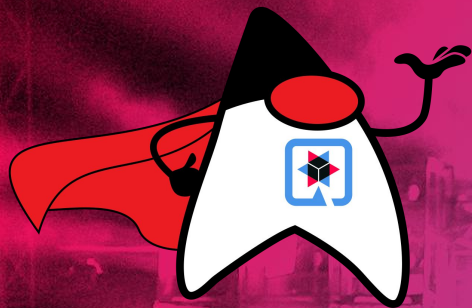


JAM TIME!

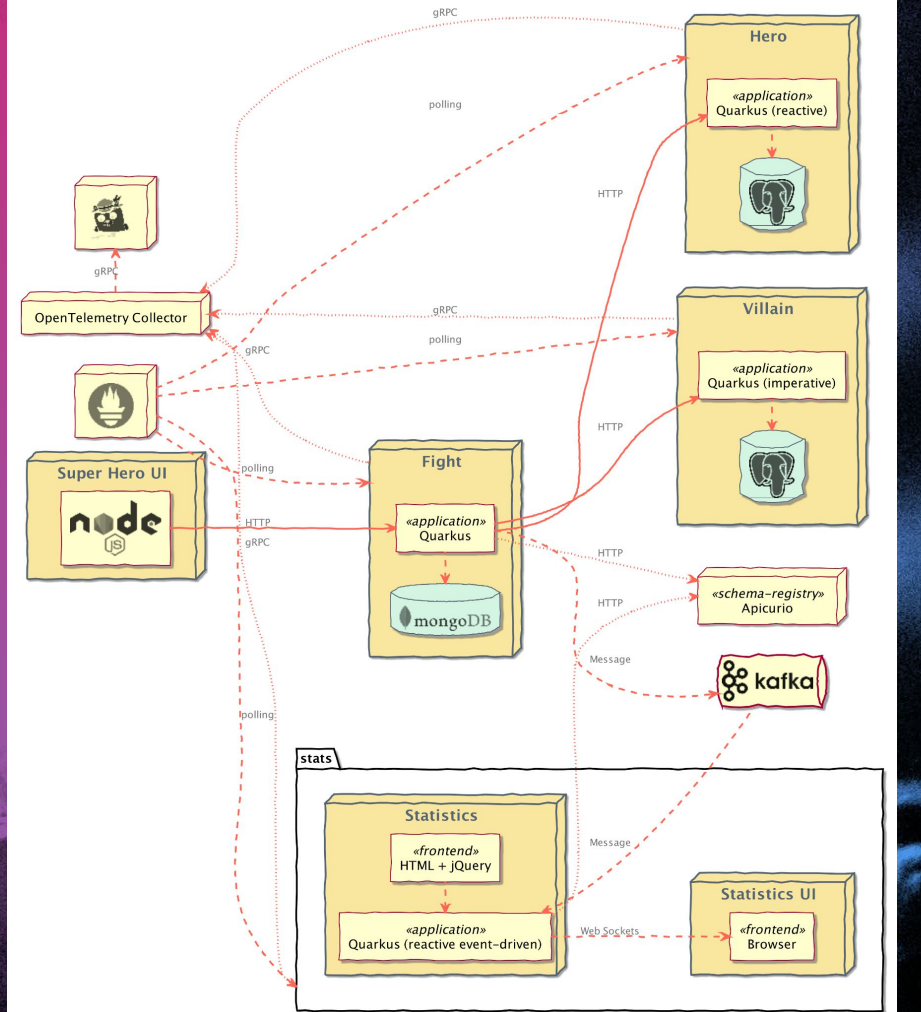


Quarkus World Tour 2022 { ʘ } ʘ

QUARKUS SUPERHEROES



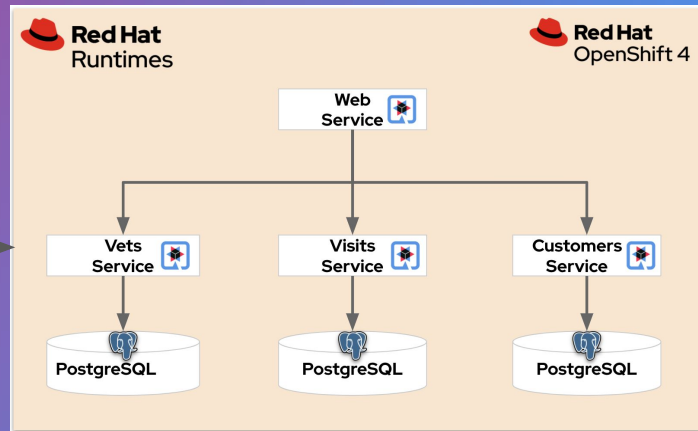
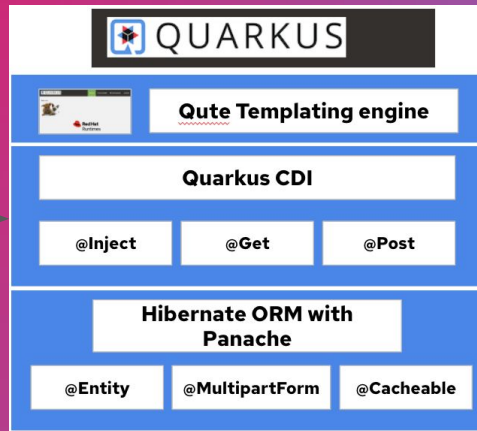
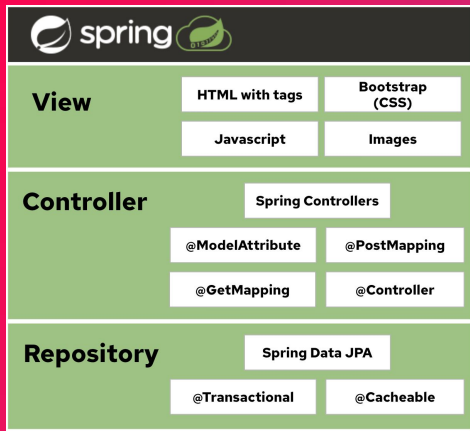
<https://github.com/quarkusio/quarkus-super-heroes>



Quarkus World Tour 2022 { }

QUARKUS WORKSHOP

- *Getting Started with Spring Petclinic*
- *Spring Web to Quarkus Qute*
- *Spring DI to Quarkus CDI*
- *Spring Data JPA to Quarkus Hibernate ORM with Panache*
- *Strangling the Spring Monolith with Quarkus*
- *Spring to Kubernetes Native with Quarkus*



DEVELOPER SANDBOX

- 30 days
- Limited access
- 4 vCPU, 8 GB RAM
- 3 projects
- Cloud IDE, Pipelines



red.ht/dev-sandbox

Add

Select a way to create an Application, component or service from one of the options.

Details on

Getting started resources

Create applications using samples

Choose a code sample to get started creating an application with.

[Basic Quarkus](#) →
[Basic Spring Boot](#) →

Build with guided documentation

Follow guided documentation to build applications and familiarize yourself with key features.

[Get started with Quarkus using s2i](#) →
[Get started with Spring](#) →

[View all quick starts](#)

Explore new developer features

Explore new features and resources within the developer perspective.

[Discover certified Helm Charts](#) →
[Start building your application quickly in topology](#) →

[What's new in OpenShift](#)

Developer Catalog

All services

Browse the catalog to discover, deploy and connect to services

Database

Browse the catalog to discover database services to add to your Application

Operator Backed

Browse the catalog to discover and deploy operator managed services

Helm Chart

Browse the catalog to discover and install Helm Charts

Managed Services

Discover managed services to simplify deployments and reduce operational overhead & complexities

Git Repository

From Git

Import code from your Git repository to be built and deployed

From Devfile

Import your Devfile from your Git repository to be built and deployed

From Dockerfile

Import your Dockerfile from your Git repository to be built and deployed

Container images

Deploy an existing Image from an Image registry or Image stream tag

Samples

Create an Application from a code sample

From Local Machine

Import YAML

Create resources from their YAML or JSON definitions

Upload JAR file

Upload a JAR file from your local desktop to OpenShift

Quick Starts

Learn how to create, import, and run applications on OpenShift with step-by-step instructions and tasks.



Binding your Quarkus application to Streams for Apache Kafka

10 minutes

Binding your Quarkus application to the OpenShift Streams for Apache Kafka cloud service

Prerequisites (3)



Get started with Quarkus using a Helm Chart

10 minutes

Deploy a Quarkus application using a Helm Chart.



Get started with Quarkus using s2i

10 minutes

Import a Quarkus Application from git, build, and deploy it onto OpenShift.

ADDITIONAL RESOURCES

- 👉 Spring Boot on Quarkus: Magic or Madness?
- 👉 Why should I choose Quarkus over Spring for my microservices?
- 👉 Quarkus for Spring Developers: Getting Started
- 👉 Evolution of the Quarkus Developer Experience
- 👉 Quarkus Insights: Quarkus for Spring Developers
- 👉 Quarkus Insights: Quarkus Superheroes



DON'T TAKE OUR WORD FOR IT!



asiakastieto

"We went from 1 min startup times to 400 ms."

[RH Press Release Community](#)



vodafone

"We became increasingly worried about resource consumption that Spring Boot was having while being deployed on the Kubernetes cluster... It became increasingly cumbersome to find ways to circumvent the methodology we were using just to squeeze every little bit of performance out of Spring Boot" - Christos Sotiriou DXL Backend Chapter Lead, Vodafone Greece

[Information Week Community](#)



Lufthansa

"We could run 3x denser deployments without sacrificing availability and response times of service."

"Quarkus is close to what our developers are already doing with Spring and it's familiar to them. This is a big benefit"

[Community](#)



"Before we introduced Quarkus, many of our customers had started to look at alternative stacks like Go and node.js to improve performance and efficiency. These customers were weary of selecting a new language and having to hire new developers or retrain their existing Java developers." - Arijit Mazumdar

"There was a low learning curve with Quarkus. It took one of our developers one week to get up to speed on Quarkus and another week to migrate a Spring application to Quarkus." - Arijit Mazumdar

"Quarkus and the Spring API compatibility reduced the migration time and complexity which is critically important for our customers." - Arijit Mazumdar

[Community](#)



<https://quarkus.io/blog/tag/user-story>

Quarkus World Tour 2022 {👋}

DON'T TAKE OUR WORD FOR IT!



"using Spring with AWS Lambda would have been prohibitive because the startup time of Spring in AWS Lambda is too big from my research" - Dennis Baerten

[Blog](#)

"As costs increase, this is when the benefit of using Quarkus will be experienced due to its more efficient use of cloud resources and fast startup time compared to plain Java and Spring Boot" - Dennis Baerten

"It took me about 3 days to get familiar with the Quarkus stack" - Dennis Berten, Spring Developer



"Some of Payair's developers had mainly Spring experience, we were concerned that it would be difficult for them to "switch sides". It turned out that our fear of the unknown was completely unfounded. Quarkus leverages some good old Jakarta EE standards that all Java developers are familiar with. We did not have to learn a bunch of new APIs." - Hubert Lewandowski

[Blog](#)

"As a long term Spring developer I realized that Spring is slowly becoming the very thing it swore to destroy. The initial premises of Spring (which basically can be summed up as a lightweight alternative to Jakarta EE) are way past the expiry date now. Spring is the undisputed heavyweight champion that can handle everything you imagine but is not your best option for fast and light services. And that applies to Spring Boot as well. - Hubert Lewandowski



"When you adopt Quarkus, you will be productive from day one since you don't really need to learn new technologies." -TalkDesk

[Blog](#)



"After deploying, we found that Quarkus used about 15% of the CPU, 12% of the memory compared to Spring Boot. So far, we are sticking with Quarkus!" - Sam Dacanay, Lead Software Engineer

[Blog](#)



<https://quarkus.io/blog/tag/user-story>

Quarkus World Tour 2022 {👋}

QUARKUS IDC LAB VALIDATION

ANALYST PAPER

Red Hat Quarkus Lab Validation

Quarkus can save as much as 64% of cloud resources as compared to Framework A*

This report compares Quarkus with another widely used Java™ framework for cloud-native development. The comparison is based on criteria important for container, Kubernetes, and cloud deployments. In this analyst paper, you'll find key findings on:

- Cost savings
- Deployment density
- Startup memory usage
- Throughput
- Startup time
- Developer productivity



Based on the reduced memory utilization demonstrated by this lab, Quarkus JVM and Quarkus Native deliver cost savings as follows:

Application stack	Estimated Saving
Framework A	0%
Quarkus on JVM	37%
Quarkus on Native	64%

These savings in reduced cloud memory utilization are commensurate to the savings in real dollars that an organization would save by using Quarkus over Framework A. For illustration purposes, if you spend \$100K USD annually in memory consumption on AWS using Framework A, you would spend \$63K with Quarkus JVM and \$36K with Quarkus Native.

DOWNLOAD THE ANALYST PAPER

First name *

Last name *

Work email *

Work phone *

Quarkus_LabValReport_#US46175920_final
May22.pdf (page 12 of 40)

Red Hat Quarkus Lab Validation Report

SCENARIO 1.2A

Memory Utilization for 10 Pods

Background

This lab quantified the ability of the Quarkus development framework to optimize memory utilization by measuring the following:

- Memory utilization after starting 10 pods

Deployment Environment and Relevant Code

OpenShift 4.2 hosted by AWS. For more detail about the environment, see "Container Orchestration Cluster" in the appendix.

Code and configuration details: Appendix: Code and configuration for Lab 1.2

Key Findings

For 10 Kubernetes pods on OpenShift 4.2 on AWS, the memory utilization of Quarkus was:

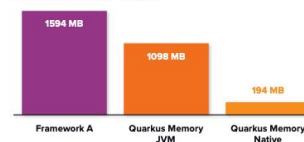
- ~1/8 that of Framework A for Quarkus Native
- ~2/3 that of Framework A for Quarkus on JVM

Public-cloud-based memory usage of Quarkus was lower than Framework A as shown below:

Memory utilization after starting 10 pods

Application stack	Memory utilization
Framework A	1594 MB
Quarkus on JVM	1098 MB
Quarkus on Native	194 MB

Memory usage for 10 instances



IDC Lab Validation Report, Sponsored by Red Hat
Doc. #JH46238020

Table of Contents



<https://www.redhat.com/en/blog/key-findings-idc-red-hat-quarkus-lab-validation>

Quarkus World Tour 2022 {👋}

*Had a good time?
let everyone know!
#quarkusrocks*

*Want to know more?
Links are in the jam doc below*

**Web
Talk
Twitter
Tour
More
Book**

*quarkus.io
quarkus.io/community
@quarkusio
worldtour@quarkus.io
quarkus.io/worldtour/soundcheck
red.ht/quarkus-spring-devs*





Quarkus World Tour 2022 { ୯ }



JAM TIME!



Quarkus World Tour 2022 { ʘ } ʘ

JAM TIME!

- 🤘 Convert and run a full-featured Spring Boot application on Quarkus with no code changes
- 🤘 Spring Web
- 🤘 Spring Data JPA
- 🤘 Spring Actuator
- 🤘 Spring Boot Config
- 🤘 SpringDoc OpenAPI
- 🤘 Micrometer metrics
- 🤘 PostgreSQL (JDBC)



JAM TIME!

- 👉 Replace Spring Boot starters with equivalent Quarkus extensions
- 👉 Replace Spring configuration with equivalent Quarkus configuration
- 👉 **Bonus Track:** No hassle native image generation

WHY WAIT FOR SPRING BOOT 3? USE QUARKUS TODAY!



JAM TIME!

<https://red.ht/spring-to-quarkus-todo>



Quarkus World Tour 2022 { ʘ } }

Thank you!

