



# QUARKUS

QUARKUS WORLD TOUR 2021

JUG Germany  
Munich and Ingolstadt

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

A musician is silhouetted against a vibrant purple and pink stage light. They are holding an electric guitar and standing in front of a microphone stand. The background is filled with thick stage smoke. To the right, a drum set is partially visible in the shadows.

# Thanks for hosting us

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👋}



A large crowd of people is seen from behind, looking towards a stage. The scene is illuminated with vibrant red and blue stage lights, creating a hazy, atmospheric effect. The crowd is dense, and many people have their arms raised in the air. The text "Here's how it's going to work today ..." is overlaid in white on the left side of the image.

Here's how it's going to work  
today ...

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👉}





# line up

intros

A taster demo

More of an structured introduction to Quarkus

Code Jam: follow along demos and examples

with Q&A as we go

wrap up and last questions

# Who's on stage today?

A musician is silhouetted against a vibrant purple and pink stage light. They are holding an electric guitar and standing in front of a microphone stand. The background is filled with thick stage smoke, and a drum set is partially visible in the shadows to the right.

Wanja Pernath  
Markus Eisele

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {🎸}



A musician is silhouetted against a stage filled with smoke and purple lighting. The musician is holding a guitar and standing next to a microphone stand. The background is dark with wisps of smoke and a strong purple light source.

# Who's on stage today?

You too! ask questions on the chat,  
we'll try to answer them as we go  
and we'll wrap at the end



# Upfront setup

JDK 8 or 11+

Maven

Git

Editor

Code

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

<https://github.com/wpernath/quarkus-worldtour>



# QUARKUS

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👋}



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



## Developer Joy

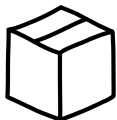
I know, right?  
SUPERSONIC JAVA, FTW!



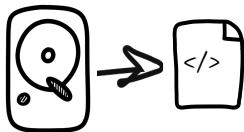
# How Does a Framework Start?

*Build Time*

*Runtime*



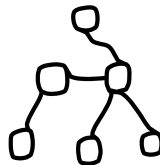
*Packaging  
(maven,*



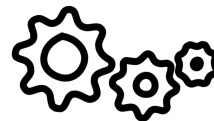
*Load config file  
from file system  
Parse it*



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



*Build its  
model of  
the world.*



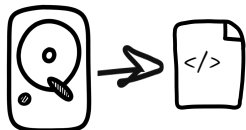
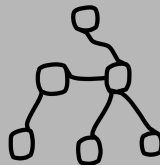
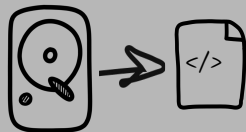
*Start the  
management  
(thread,  
pool...)*



# The Quarkus Way

*Build Time*

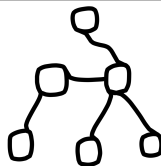
*Runtime*



*load*



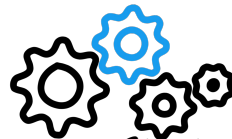
*scan*



*model*



*Package*



*Start*

*Build Time*

*Runtime*

# Developer Joy

Moved as much as possible to build phase

Minimized runtime dependencies

Maximize dead code elimination

Introduced clear metadata contracts

Spectrum of optimization levels

(all → some → no runtime reflection)



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



Since Quarkus is instrumenting & adapting code for performance reasons

It's seems sensible to include better developer tooling!

Quarkus offers you less boilerplate, better defaults, and a container centric development approach, You've never seen Hot code replace like this

Three things:

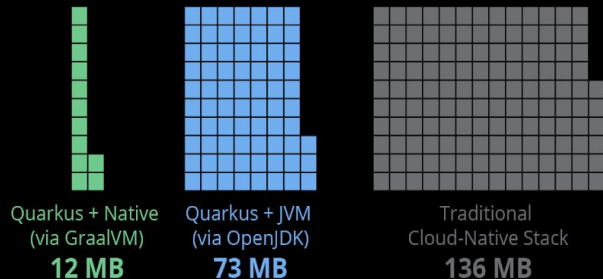
Performance  
Capabilities  
Community



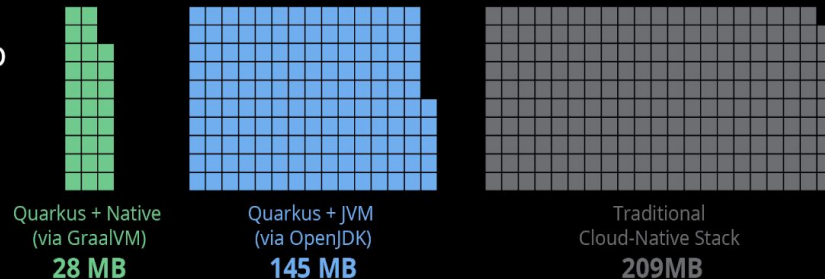
## Memory (RSS) in Megabytes\*

\*Tested on a single-core machine

REST



REST  
+ CRUD



## BOOT + First Response Time

REST



① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

REST  
+ CRUD



# Performance

REST

REST

# Performance is awesome

BOOT

Quarkus tailors your application for GraalVM and HotSpot.

REST

Amazing boot times, incredibly low RSS memory (not just heap size!)

REST  
+ CRUD

near instant scale up and high density memory utilization

Spoiler: Graal is optional - you still get awesome perf without it

conds



REST

REST  
+ CRUD

For performance, Quarkus framework + extensions

BOOT

REST

REST  
+ CRUD

1. Reduce unneeded overheads
2. Migrate some startup code to build time!
3. Makes code Kubernetes aware
4. Makes code more Graal friendly

Traditional  
Cloud-Native Stack

9.5 Seconds

# GraalVM

Standard Java is a challenge.

Java's dynamic, open view is at odds with an AOT approach

Quarkus makes using Graal so much easier



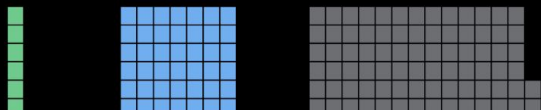
## Native Image Java Limitations

Native Image does not support all features of Java to keep the implementation small and to enable aggressive ahead-of-time optimizations. This page documents the limitations.

What	Support Status
<a href="#">Dynamic Class Loading / Unloading</a>	Not supported
<a href="#">Class-File Version 6 and Above</a>	Supported (Requires Configuration)
<a href="#">Class-File Version 7 and Above</a>	Supported (Requires Configuration)
<a href="#">JNI</a>	Mostly supported
<a href="#">Unsafe Class</a>	Mostly supported
<a href="#">Finalizers</a>	Supported
<a href="#">Invokedynamic</a>	Supported
<a href="#">Bytecode and Method Handles</a>	Not supported
<a href="#">Lambda Expressions</a>	Supported
<a href="#">Synchronized, wait, and notify</a>	Supported
<a href="#">Finalizers</a>	Not supported
<a href="#">References</a>	Mostly supported
<a href="#">Threads</a>	Supported
<a href="#">Identity Hash Code</a>	Supported
<a href="#">Security Manager</a>	Not supported
<a href="#">JVMTI, JMX, other native VM interfaces</a>	Not supported
<a href="#">JCA Security Services</a>	Supported



REST

REST  
+ CRUD

BOOT

REST

REST  
+ CRUD

Quarkus takes a “container first” approach because tuning for kubernetes means your Java application works well - everywhere else!

(via GraalVM)

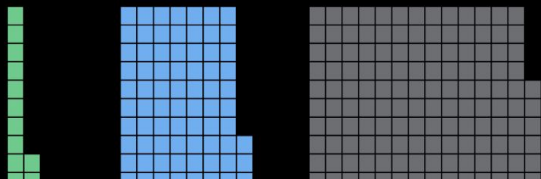
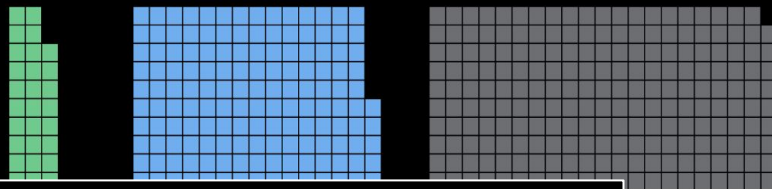
Quarkus + JIT  
(via OpenJDK)

2.033 Seconds

Traditional  
Cloud-Native Stack

9.5 Seconds

REST

REST  
+ CRUD

BOOT

REST

REST  
+ CRUD

Quarkus is “kubernetes native” not  
“cloud native” because kube centric  
code will run well almost everywhere  
*The converse is not true...*

Quarkus + Native  
(via GraalVM) 0.042 SecondsQuarkus + JIT  
(via OpenJDK) 2.033 SecondsTraditional  
Cloud-Native Stack

9.5 Seconds

Three things:

~~Performance~~

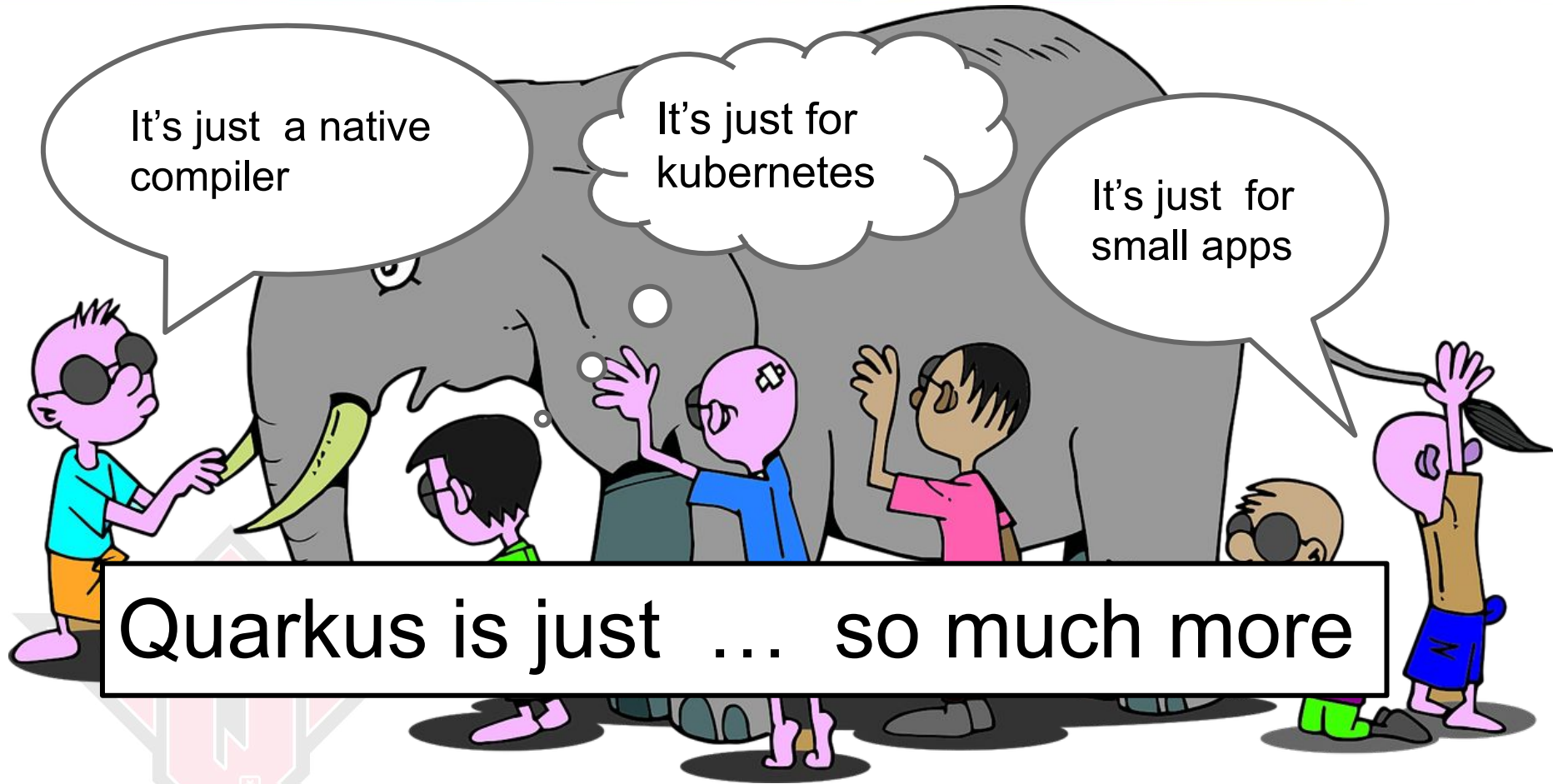
Capabilities

Community



You may have heard

Quarkus is just ...



## Supersonic Subatomic Java

A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.



# Comprehensive

## A cohesive, fun to use, full-stack framework

leveraging a growing list of  
over 100 best-of-breed  
libraries that you love and  
use.

## All wired on a standard backbone



## Quarkus is not

**Just a native compiler** - it's optional - use it where it makes sense. It's much easier with Quarkus

**Just for new apps** - Quarkus is applicable to all forms of Java application - new, old, web based, desktop. Even CLI's

**Just for small apps** - Quarkus scales across the range

**A Spring-lite copy** - Quarkus lets you use almost all your code and dependencies and all of your skills. The only differences are where it's essential for the cloud

**Just another framework!**

## So what is Quarkus?

Quarkus is a comprehensive, inclusive, community driven, open source, practical reimaging of Java for the Cloud era

Quarkus takes all your frameworks, all your dependencies, all your code and helps make them work together in the best performing way for the Cloud

Quarkus improves the developer experience too - by using the same techniques to give you a much more productive toolkit for cloud application development

**Bonus:** The performance and productivity improvements work for your existing code - even if its not moving to the cloud

## Quarkus Extensions

The jewel of Quarkus is the mechanisms for making your dependencies just work in a container environment

Whether adding container awareness, performance improvements, wiring into kubernetes services (or other clouds) etc

Even the hard work of using Graal is made easier (and often invisible) since the extension framework generates the config, adapter code etc at build time.

There are core extensions that are part of Quarkus  
There are many more that are community created and owned.



Agroal  
Amazon Alexa  
Amazon DynamoDB  
Amazon IAM  
Amazon KMS  
Amazon S3  
Amazon SES  
Amazon SNS  
Amazon SQS  
AMQP JMS client  
Apache Avro  
Apache Camel ActiveMQ  
Apache Camel AMQP  
Apache Camel ArangoDb  
Apache Camel AS2  
Apache Camel Async HTTP Client  
(AHC)  
Apache Camel Async HTTP Client  
(AHC) Websocket  
Apache Camel AtlasMap

Compute Cloud (EC2)  
Apache Camel AWS 2 Elastic  
Container Service (ECS)  
Apache Camel AWS 2 Elastic  
Kubernetes Service (EKS)  
Apache Camel AWS 2 Eventbridge  
Apache Camel AWS 2 Identity and  
Access Management (IAM)  
Apache Camel AWS 2 Key  
Management Service (KMS)  
Apache Camel AWS 2 Kinesis  
Apache Camel AWS 2 Lambda  
Apache Camel AWS 2 Managed  
Streaming for Apache Kafka (MSK)  
Apache Camel AWS 2 MQ  
Apache Camel AWS 2 S3 Storage  
Service  
Apache Camel AWS 2 Security  
Token Service (STS)  
Apache Camel AWS 2 Simple  
Email Service (SES)  
Apache Camel AWS 2 Simple  
Notification System (SNS)

Artemis JMS  
AWS Lambda  
AWS Lambda HTTP  
AWS Lambda HTTP V1  
(deprecated)  
AWS Lambda X-Ray  
Azure Functions HTTP  
Blaze-Persistence  
Container Image Docker  
Container Image Jib  
Container Image OpenShift  
Container Image S2I  
DataStax Apache Cassandra client  
Debezium Quarkus Outbox  
Eclipse Vertx GraphQL  
Elasticsearch REST client  
Elasticsearch REST High Level  
Client  
Elytron Security JDBC  
Elytron Security LDAP Realm  
Elytron Security OAuth 2.0  
Elytron Security Properties File

Three things:

~~Performance~~

~~Capabilities~~

Community

# Community

over 450 committers:

Inclusive, open, practical, pragmatic, inventive. Not all from Red Hat!

But that's just Quarkus.

\*

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {🌍}



# Community

## the unique Red Hat contribution

Red Hat - or rather Red Hatters - connect across the open source communities that make up software stack we use today.

Whether Linux kernel, OpenJDK, Kubernetes, Docker, Java Frameworks or small project

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👋}





Quarkus isn't just another framework

Quarkus is a practical vision of modern Java

Bringing together all the parts we need to  
build the next generation of Java  
masterpieces

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {🍷}



[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {8}



A large crowd of people is gathered at night, illuminated by vibrant red and blue stage lights. The scene is energetic, with many people raising their hands in the air. In the background, there are silhouettes of buildings and structures, suggesting an outdoor festival or concert setting.

# What we're going to show

Quarkus project generation

Quarkus “dev mode”

Quarkus Panache

Quarkus container image

Quarkus on OpenShift

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {🌍}



[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👉}

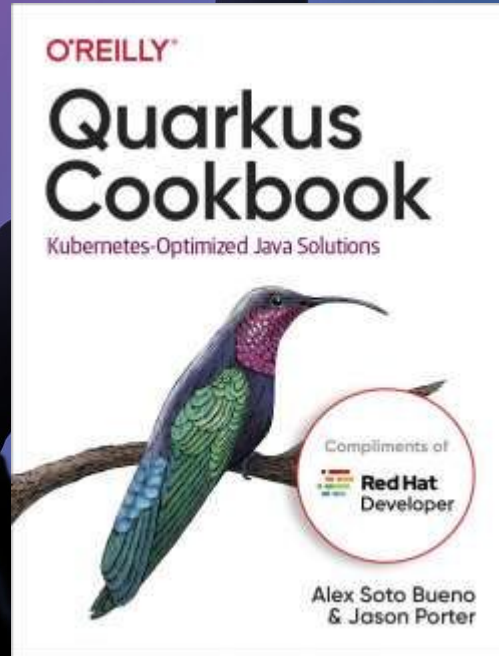


**Had a good time?  
let everyone know!  
#quarkusrocks**

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

**Web** [quarkus.io](https://quarkus.io)  
**Talk** [quarkusio.zulipchat.com](https://quarkusio.zulipchat.com)  
**Twitter** [@quarkusio](https://twitter.com/quarkusio)  
**tour** [worldtour@quarkus.io](https://worldtour@quarkus.io)  
**More** [quarkus.io/worldtour/soundcheck](https://quarkus.io/worldtour/soundcheck)  
**book** [bit.ly/q-book](https://bit.ly/q-book)

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)





Swag is sent to your  
JUG leaders

Swag varies by country

Swag will be as  
awesome as we can  
make it !



[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👉}

# Three things:

**Quarkus:** accelerates your day-2-day Java development

**Quarkus:** makes all your Java applications supersonic

**Quarkus:** for Java everywhere



Next tour stop

[bit.ly/quarkus-worldtour-jam](https://bit.ly/quarkus-worldtour-jam)

Quarkus Road Tour 2021 {👉}