

RED HAT FORUMS

Quarkus

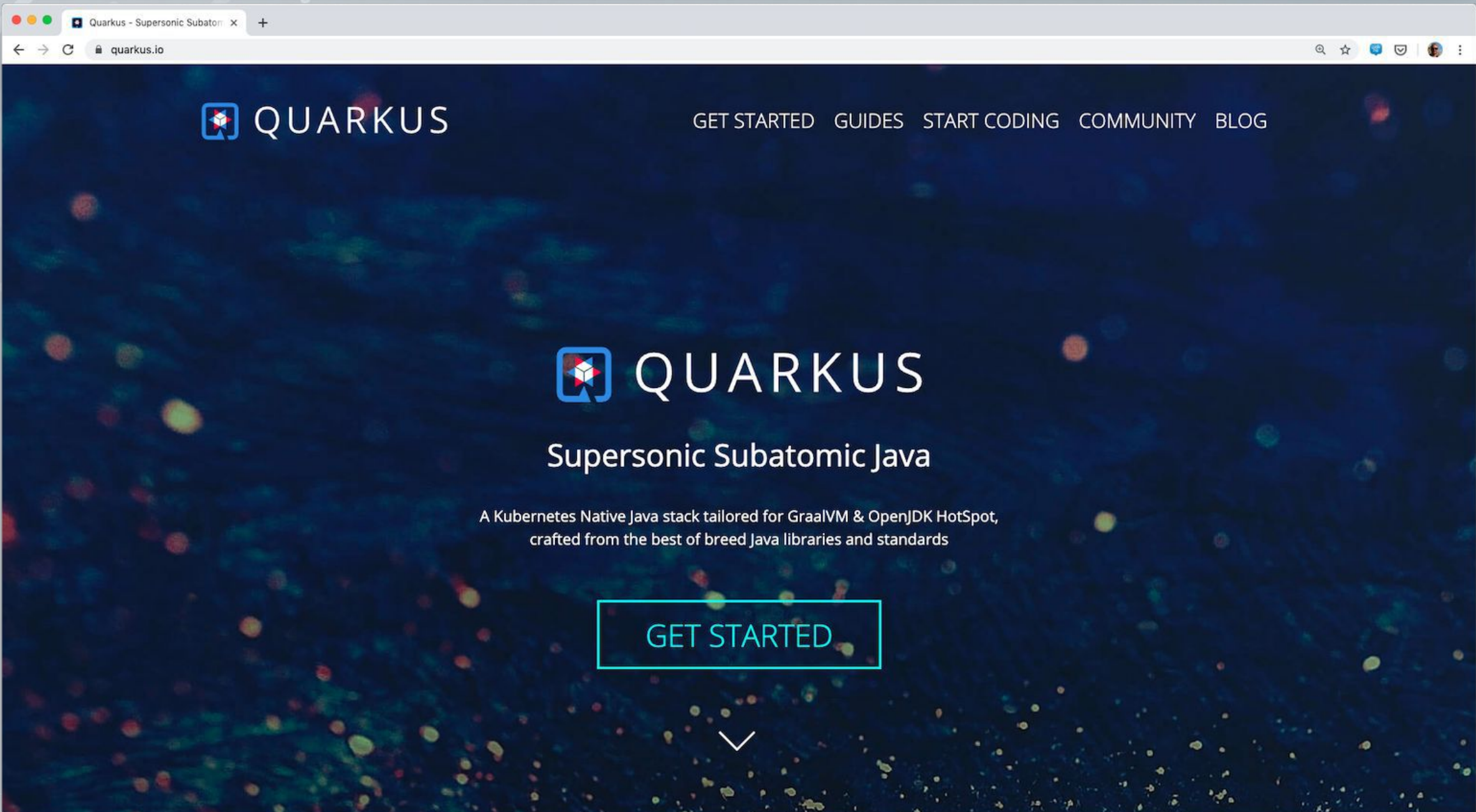
Hyperscaling Java in Kubernetes

Timo Friman
Solution Architect

Martin Östmark
Solution Architect

October 2019





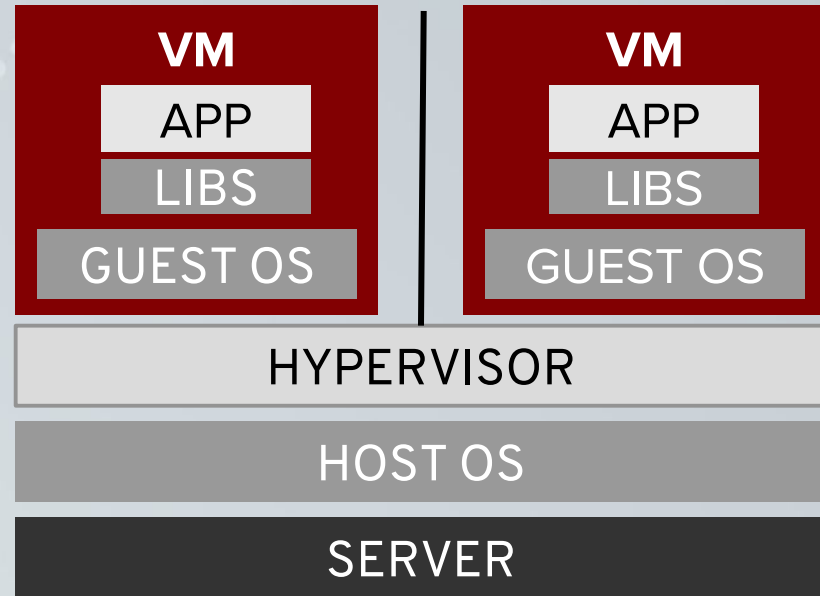
Supersonic Subatomic Java

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

GET STARTED



CONTAINERS PROVIDE HIGH EFFICIENCY, DENSITY AND APPLICATION PORTABILITY

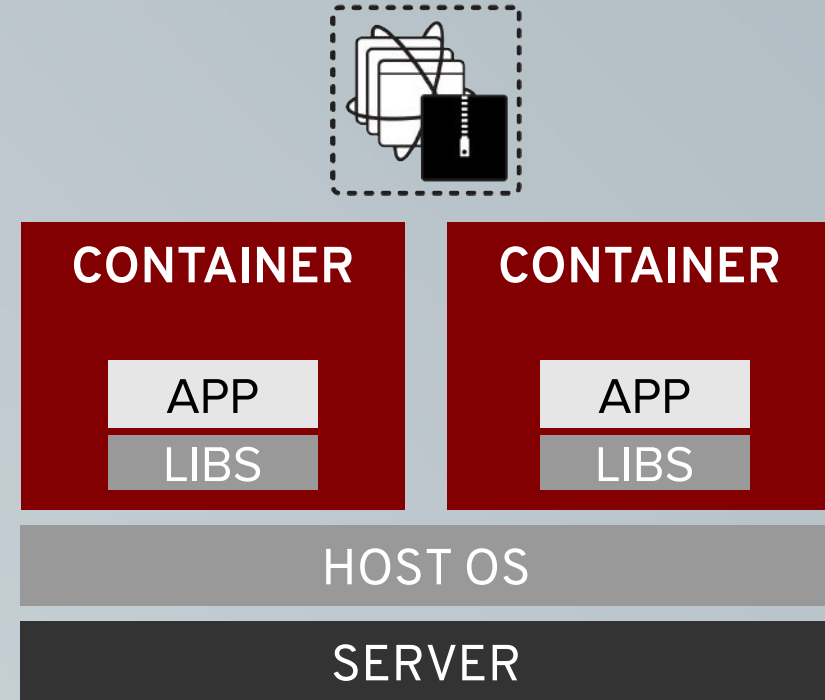


10 per
Server

PHYSICAL SERVER



27 HRS



100 per
Server

VIRTUAL MACHINE



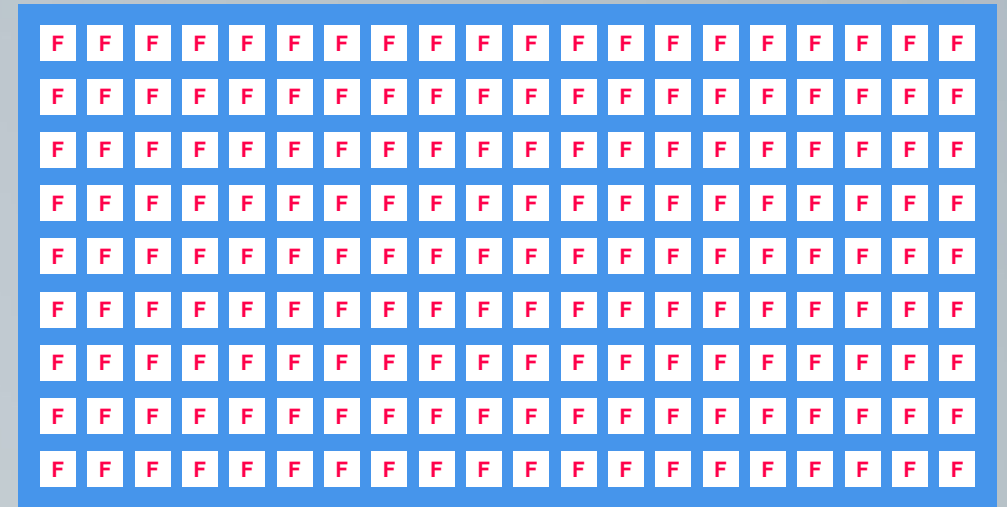
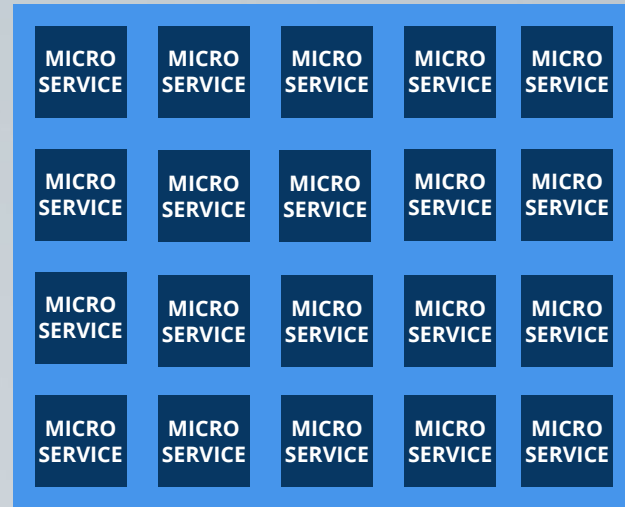
12 MINS

CONTAINER INSTANCE



10 SECS

From monolith to...



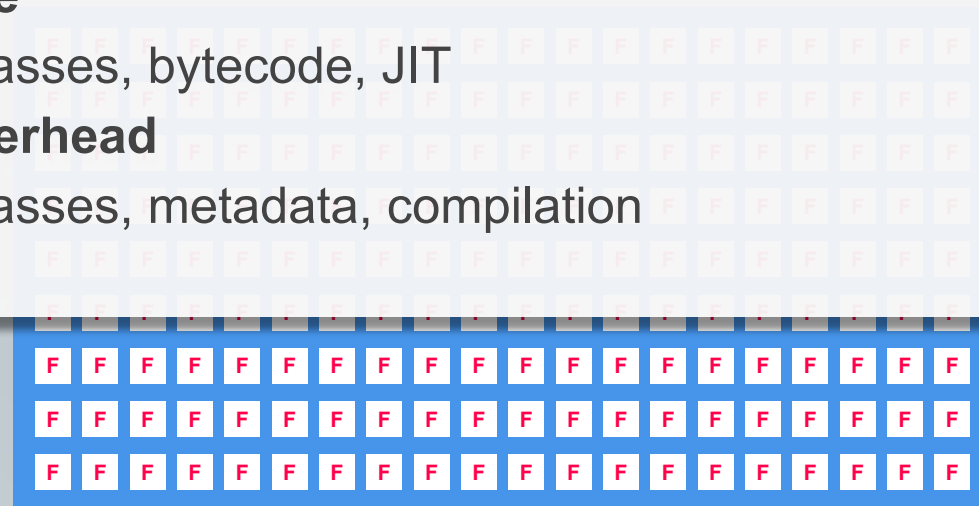
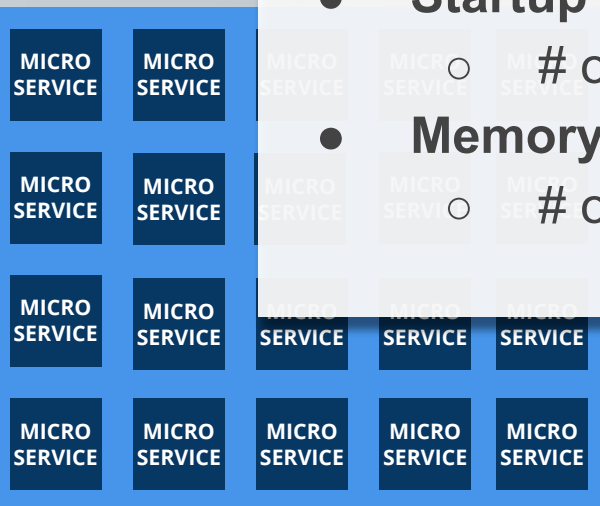
- 1 monolith $\approx$ 20 microservices $\approx$ 200 functions
- Scale to 1 vs scale to 0
- Startup time

From monolith to...

Some challenges with Java in context of Serverless/FaaS...

- **Startup time**
 - # of classes, bytecode, JIT
- **Memory overhead**
 - # of classes, metadata, compilation

MONOLITH



- 1 monolith $\approx$ 20 microservices $\approx$ 200 functions
- Scale to 1 vs scale to 0
- Startup time



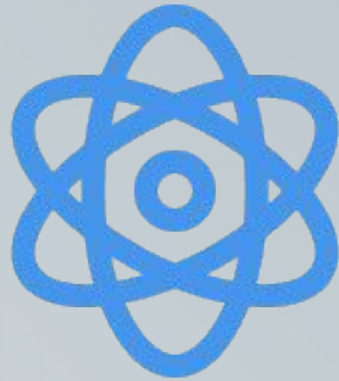
QUARKUS

Supersonic. Subatomic. Java.

A stack to write Java apps



Cloud Native,



Microservices,

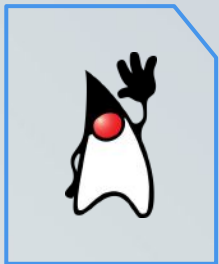
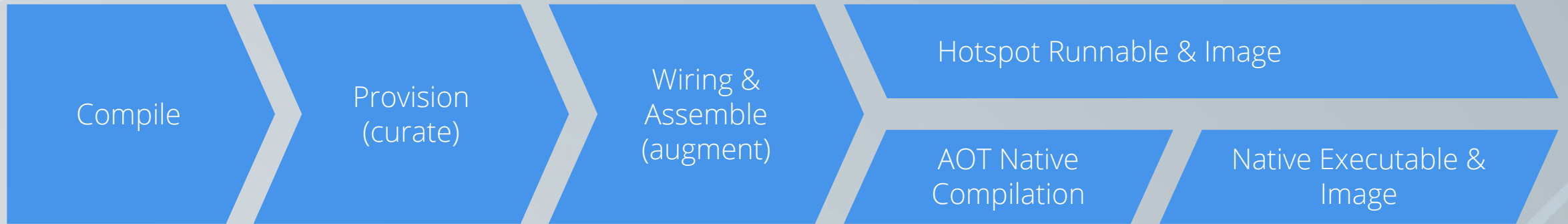


Serverless

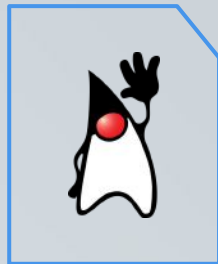
WHAT MAKES QUARKUS DIFFERENT?



Runnable java app



app.jar



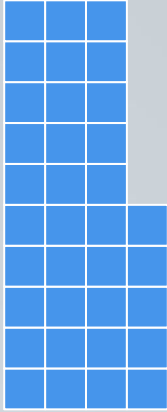
frameworks



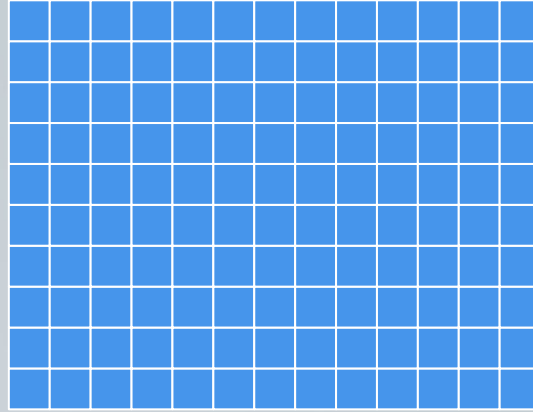
Native-app
(GraalVM)

SUPERSONIC SUBATOMIC JAVA

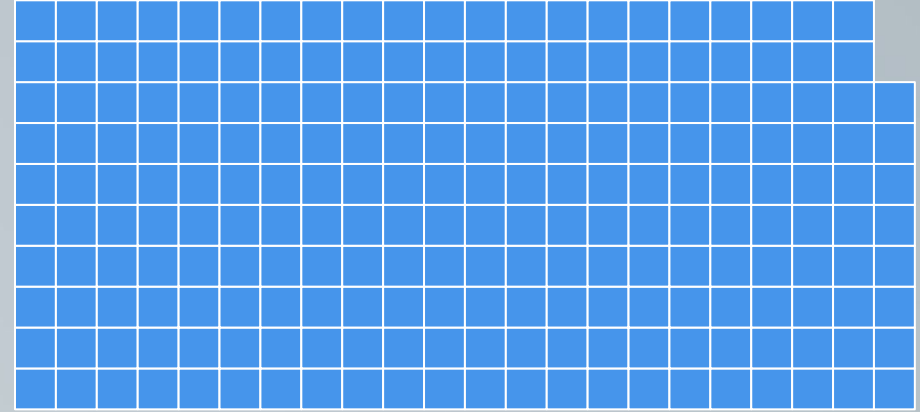
REST + CRUD



Quarkus + GraalVM
35 MB



Quarkus + OpenJDK
130 MB



Traditional Cloud-Native Stack
218 MB

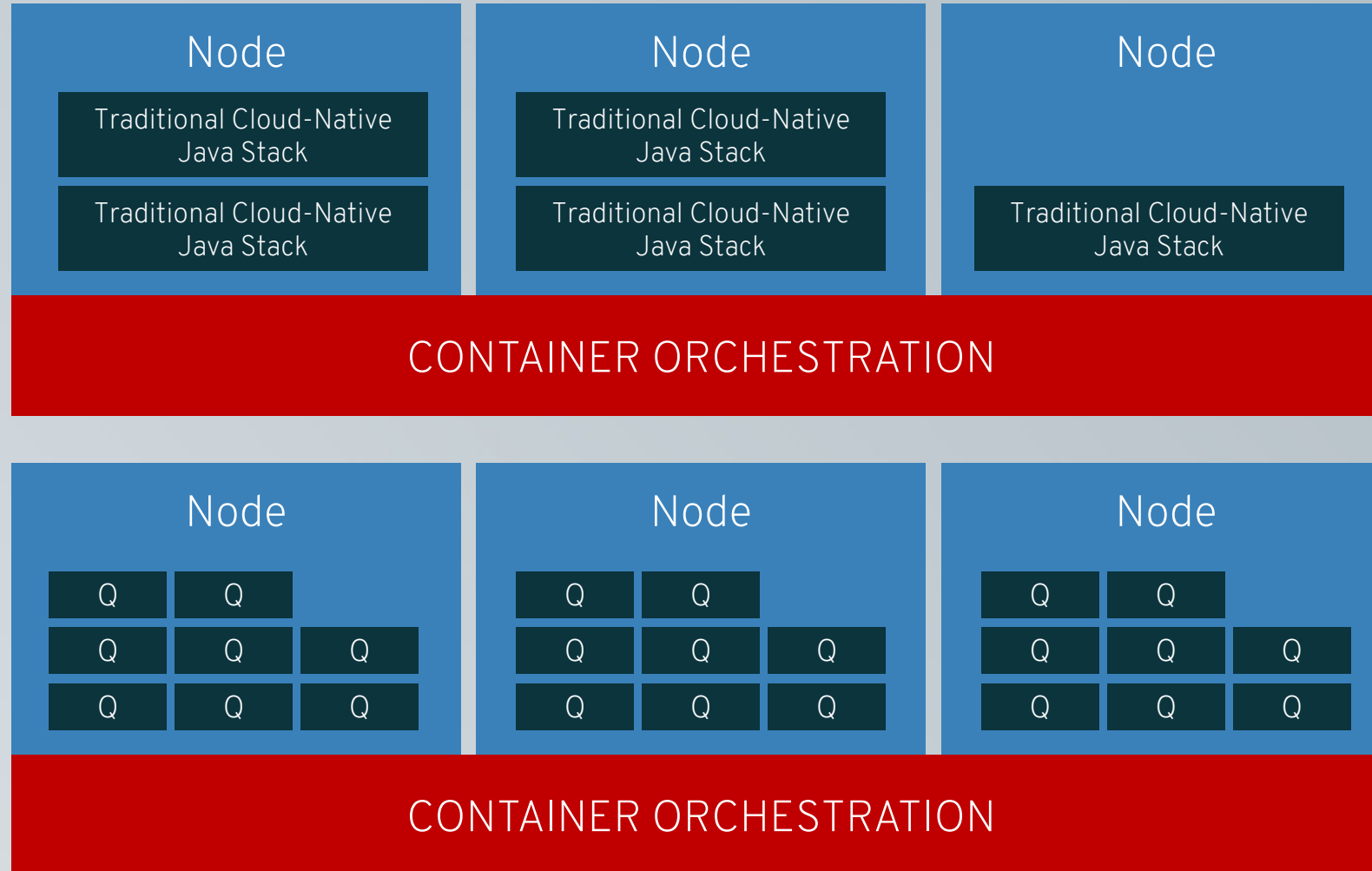
REST + CRUD

Quarkus + GraalVM **0.055 Seconds**

Quarkus + OpenJDK **2.5 Seconds**

Traditional Cloud-Native Stack **9.5 Seconds**

BETTER RESOURCE UTILIZATION



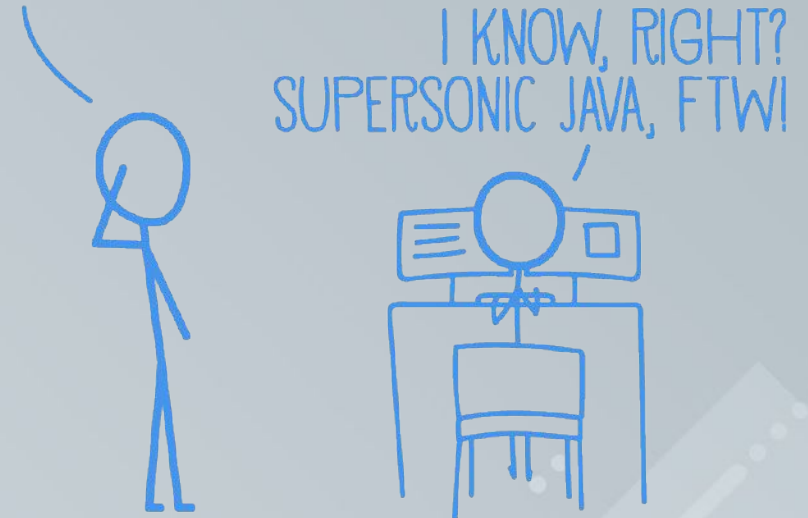
MAKING APP DEVELOPERS HAPPY

Quarkus is designed for Developer Joy

A cohesive platform for optimized developer joy:

- Based on standards, but not limited
- Unified configuration
- Zero config, live reload in the blink of an eye
- Streamlined code for the 80% common usages, flexible for the 20%
- No hassle native executable generation

WAIT,
SO YOU JUST SAVE IT,
AND YOUR CODE IS RUNNING?
AND IT'S JAVA?!



Unifies Imperative and Reactive

```
@Inject
SayService say;

@GET
@Produces(MediaType.TEXT_PLAIN)
public String hello() {
    return say.hello();
}
```

```
@Inject @Stream("kafka")
Publisher<String> reactiveSay;

@GET
@Produces(MediaType.SERVER_SENT_EVENTS)
public Publisher<String> stream() {
    return reactiveSay;
}
```

- Combine both Reactive and imperative development in the same application
- Inject the EventBus or the Vertx context
- Use the technology that fits your use-case

Best of Breed Frameworks & Standards



Eclipse Vert.x



Hibernate



RESTEasy



Apache Camel



Eclipse MicroProfile



Netty



Kubernetes



OpenShift



Jaeger



Prometheus



Apache Kafka



Infinispan

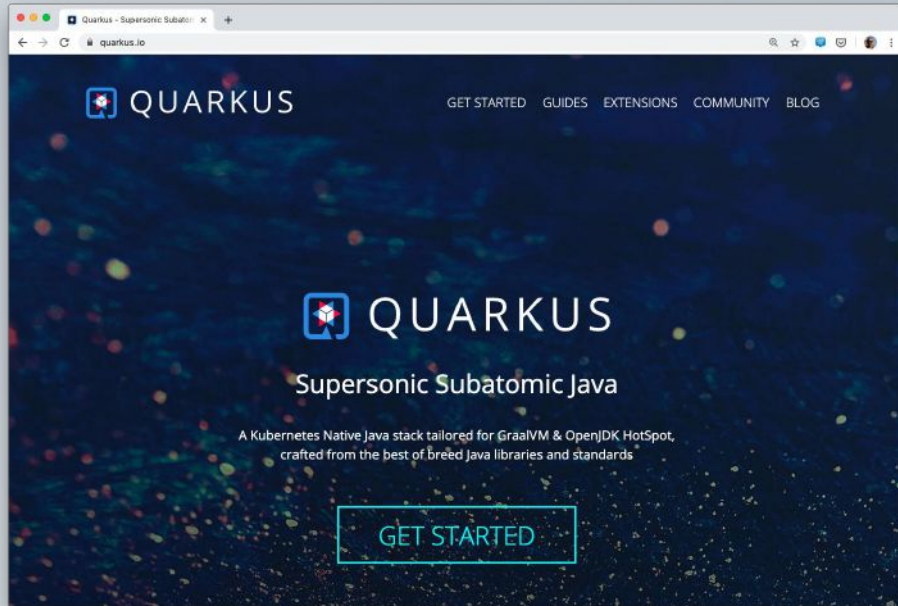
DEMO

DEMO - OVERVIEW

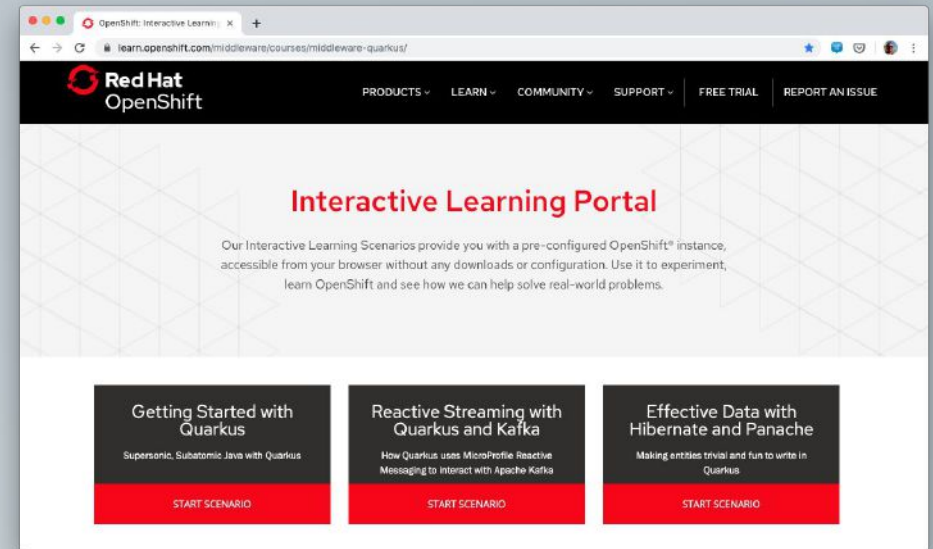
1. Generate a Quarkus project
2. Context and Dependency Injection (CDI) out of the box
3. Extensions - add more capabilities to our app
4. Packaging the application - Java and Native compilation
5. Running a Quarkus app on Kubernetes (OpenShift)

LET'S GO!

WANT TO LEARN MORE?



quarkus.io



bit.ly/quarkus-lab

RED HAT FORUMS

THANK YOU



plus.google.com/+RedHat



facebook.com/RedHatinc



linkedin.com/company/Red-Hat



twitter.com/RedHat



youtube.com/user/RedHatVideos