



Spring Application on Quarkus & Kubernetes

Charles Moulliard - Software Engineer Manager



@cmoulliard

Quarkus - Agenda

- Quarkus runtime - new Java stack
- Spring on Quarkus - can we ?
- What about deployment on k8s
- and serverless ...
- Demos :-)

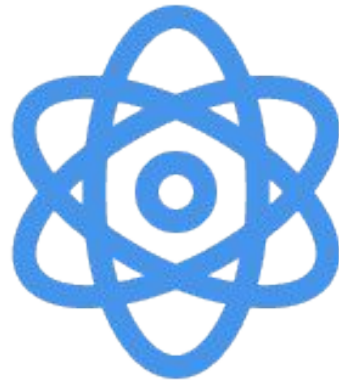
Quarkus New Java Stack

Quarkus

An Open Source
stack to write Java apps



Cloud Native,



Microservices,



Serverless

Benefits of Quarkus

Developer **Joy**

Supersonic
Subatomic Java

Unifies

imperative and reactive

Best of breed

libraries and **standards**

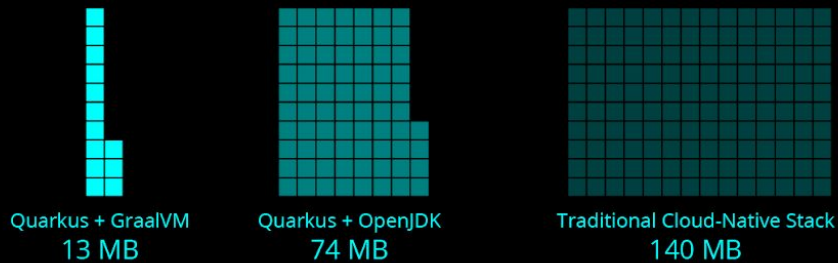
QUARKUS

Supersonic Subatomic Java

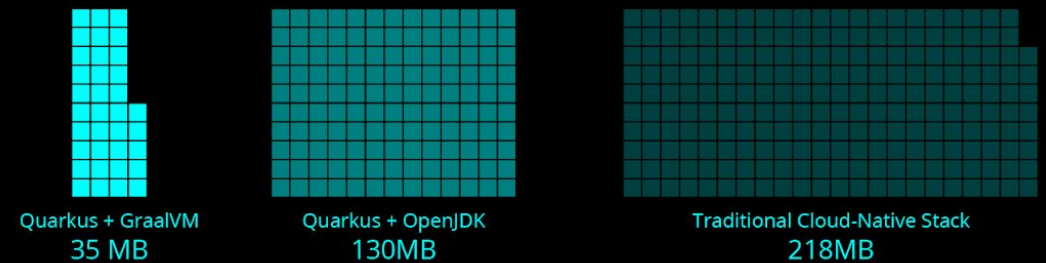
MEMORY AND BOOT + FIRST RESPONSE TIME

Memory (RSS) in Megabytes

REST



REST + JPA

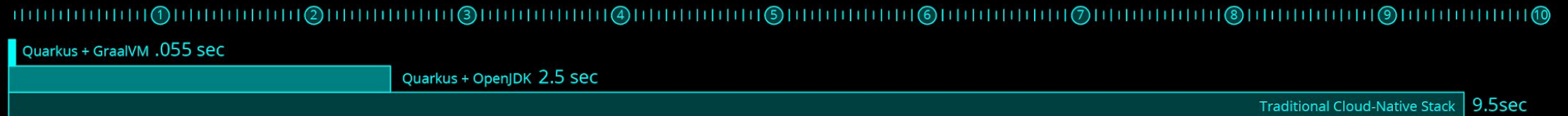


Boot + First Response Time in Seconds

REST



REST + JPA



Best of Breed Frameworks & Standards



VERT.X

Eclipse Vert.x



Eclipse MicroProfile



Spring Compas



Hibernate



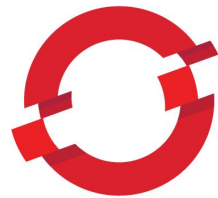
RESTEasy



Apache Camel



Kubernetes



OpenShift



Jaeger



Prometheus



Apache Kafka



Netty

<https://code.quarkus.io/>

Quarkus - How to

- Project Making off
 - Create a Java (maven | gradle) project
 - Add Quarkus dependencies (= **extensions**)
 - Launch it : **mvn quarkus:dev**
 - Call Endpoint(s)
 - Experiment **hot reload**
 - Use **native** build (graalvm)
 - Move to docker, deploy on k8s/OpenShift

A red stamp with the word "DEMO" in a bold, sans-serif font, enclosed in a red rectangular border with rounded corners. The stamp has a slightly distressed or ink-like texture.

<https://github.com/snowdrop/quarkus-demo/blob/master/REST-spring.md#create-a-quarkus-rest-project>

Spring on Quarkus

Quarkus - Spring

- **Spring** developers can **leverage** their **skills** :-)
- How ? use simply Spring Annotations
- We support **Spring extensions** :
 - CDI
 - Web
 - JPA
 - Security, config
- More will come :-)



<https://github.com/snowdrop/spring-quarkus-fruit>

S

What about packaging & deployment on k8s



DEMO

<https://github.com/snowdrop/spring-quarkus-fruit>
[s#deploy-on-the-cluster](#)



QUARKUS

Supersonic Subatomic Java

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

[GET STARTED WITH QUARKUS](#)

Now Available

QUARKUS 1.1.1



Halkyon Operator

Halkyon: To get back to the halcyon days of local development in a modern kubernetes setting of microservices!

Home > Halkyon Operator

Halkyon Operator

Introduction

Deploying modern micro-services applications that comply with the [12-factor](#) guidelines to Kubernetes is difficult, mainly due to the host of different and complex Kubernetes Resources involved. In such scenarios developer experience becomes very important.

This projects aims to tackle said complexity and vastly **simplify** the process of deploying micro-service applications to Kubernetes and get back to the halcyon days of local development! 😊

By providing several, easy-to-use Kubernetes [Custom Resources \(CRs\)](#) and an [Operator](#) to handle them, the Halkyon project provides the following features:

- Install micro-services (components in Halkyon's parlance) utilizing runtimes such as Spring Boot, Vert.x, Thorntail, Quarkus or Nodejs, serving as base building blocks for your application
- Manage the relations between the different components using `link` CR allowing one micro-service for example to consume a REST endpoint provided by another
- Deploy various infrastructure services like a database which are bound to a component via the

Install

CHANNEL

alpha

VERSION

0.1.8 (Current) ▾

CAPABILITY LEVEL ⓘ

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights

Quarkus - References

- Project : <https://quarkus.io/>
- Dev Exp
 - Operator: <https://github.com/halkyonio/operator/>
 - Hal : <https://github.com/halkyonio/hal>
- Demo <https://github.com/snowdrop/quarkus-demo>

Quarkus

- Questions
- Contact: cmoulliard@redhat.com

