



Kubernetes native Spring apps with Quarkus

About me

- *Georgios Andrianakis - [@geoand86](https://twitter.com/geoand86)*
- *Red Hat Engineer*
 - *Spring Boot*
 - *Spring on Quarkus*
- *Athens Kubernetes Meetup Co-organizer*
- *Speaker*





Agenda

- *Short Quarkus intro*
- *Using Spring APIs on Quarkus*
- *Demo*
- *Cloud Native Deployment*
- *Demo*

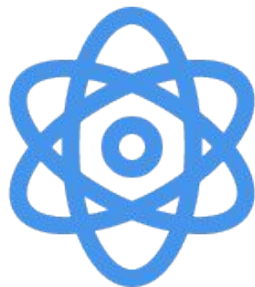


Quarkus

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

Best of Breed Frameworks & Standards



Eclipse Vert.x



Eclipse MicroProfile



Spring Compas



Hibernate



RESTEasy



Apache Camel



Kubernetes



OpenShift



Jaeger



Prometheus



Apache Kafka



Netty



Spring apps on Quarkus

- Why Spring on Quarkus
 - Capitalize your Spring Know-how
 - Speed up knowledge transfer and onboarding
 - Familiarity

Spring on Quarkus

- Spring DI & Web
 - Dependency injection model
 - `@Component`, `@Autowired`, etc.
 - Spring Web - Developing REST APIs
 - `@RestController`, etc.



Spring REST APIs

pom.xml

```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-spring-web</artifactId>
</dependency>
```

GreetingController.java

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class GreetingController {

    @GetMapping("/hello")
    public String hello(@RequestParam(defaultValue = "world") String name) {
        return "hello " + name;
    }
}
```



DEMO



Spring on Quarkus

- Spring Data JPA
 - Derived methods
 - Custom query methods
 - Fragments
 - Transactional support

Spring Data JPA

pom.xml

```
<dependency>  
  <groupId>io.quarkus</groupId>  
  <artifactId>quarkus-spring-data-jpa</artifactId>  
</dependency>
```

Book.java

```
import javax.persistence.Entity;  
import javax.persistence.Id;  
  
@Entity  
public class Book {  
  @Id  
  private Integer bid;  
  private Integer publicationYear;  
  private String name;  
}
```

Spring Data JPA

BookRepository.java

```
import java.util.List;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

public interface BookRepository extends CrudRepository<Book, Integer> {

    List<Book> findByPublicationYearBetween(Integer lower, Integer higher);

    @Modifying
    @Query("update Book b set b.name = concat(b.name, :suffix) where b.name like concat('%', :name, '%') ")
    void addSuffixToMatching(@Param("name") String name, @Param("suffix") String suffix);
}
```

DEMO

Cloud Native Deployment ...

- *Traditional Java stacks impose a mismatch between developer productivity and cloud economics*
 - *More developer productivity meant more dynamic behavior*
 - *More dynamic behavior means larger runtime footprint*

Cloud Native Deployment ...

- *The need to scale horizontally exacerbates the problem*
- *What about serverless?*

The logo for Devoxx, featuring the word "Devoxx" in a stylized, blocky font. The "x"s are orange and the other letters are dark grey. A small "TM" trademark symbol is at the top left.

What does Quarkus do different

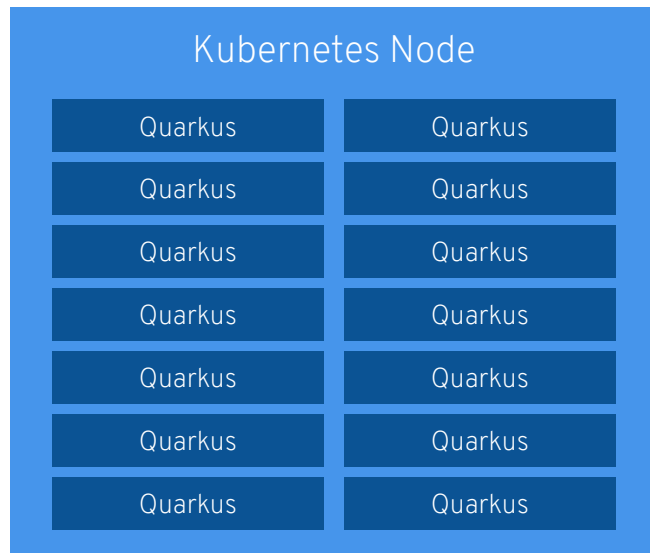
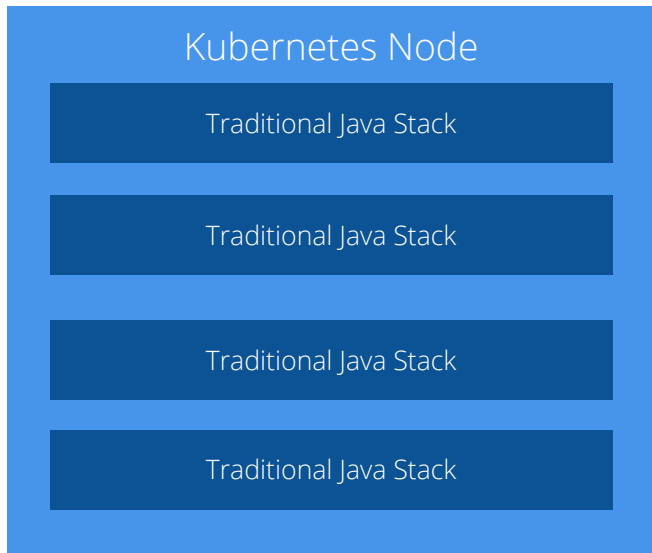
- *Move as much processing to build time as possible*
 - *Parse config files*
 - *Classes & classpath scanning*
 - *Read annotation and other metadata*
 - *Prepare reflection and build proxies*
 - *Wire-up components*

The logo for Devoxx, featuring the word "Devoxx" in a stylized, blocky font. The "x"s are orange, and the other letters are dark grey. A small "TM" trademark symbol is at the top left.

Benefits of build time work

- Minimize **runtime** dependencies
- Bootstrap classes **vanish**
- **Eliminate** dynamic behavior
- Pay the “startup” cost **once** - at build time
- Set the scene for native image compilation - GraalVM

Cloud Native Deployment



Container orchestration



DEMO



Recap

- *Leverage existing Spring knowledge*
- *Quarkus offers substantial productivity and economic benefits*
- *Embrace the brave new cloud-native world without compromise*





Quarkus Community

<https://quarkusio.zulipchat.com>

Documentation

<https://quarkus.io/guides/spring-di-guide>

<https://github.com/quarkusio/quarkus-quickstarts>

Learn/Discover

- <https://learn.openshift.com/middleware/courses/middleware-quarkus/getting-started>
- <https://github.com/geoand/spring-quarkus-demo1>
- <https://github.com/geoand/sidebyside>

Slides

- <https://github.com/geoand/devoxx-belgium2019-slides>