APACHECON @Home Sep 29 - Oct 1, 2020

What’s new with Apache Camel 3?

Claus Ibsen
Red Hat

Andrea Cosentino
Red Hat

About

Claus Ibsen

- Senior Principal Software Engineer at Red Hat
- ASF Member & Java Champion
- Apache Camel tech lead
- Author of Camel in Action books
- Based in Denmark
- Blog: http://www.davsclaus.com
- Twitter: @davsclaus
- Github: davsclaus

About

Andrea Cosentino

- Senior Software Engineer at Red Hat
- ASF Member
- Apache Camel PMC chair
- Apache Camel tech lead
- Based in Italy
- Twitter: @oscerd2 Github: oscerd

Agenda

- What’s Camel
- Camel 3 Projects
- Camel Releases

- Camel K
- Camel Quarkus
- Camel Kafka Connector

- More Material
- Q & A

What’s Camel

What is Apache Camel

The *swiss knife* of integration

What is Apache Camel

The *giant swiss* knife 😅

Packed with a lot of functionality

What is Apache Camel

- Java based integration framework (library)
- Based on Integration Patterns (EIPs)
- Comes with 300+ components (connectors)
- DSL (Java or XML) to describe integration flow (routes)
- Can integrate ~~everything~~ ... almost everything

Runs on popular Java Runtimes

WildFly

VERT.X

QUARKUS

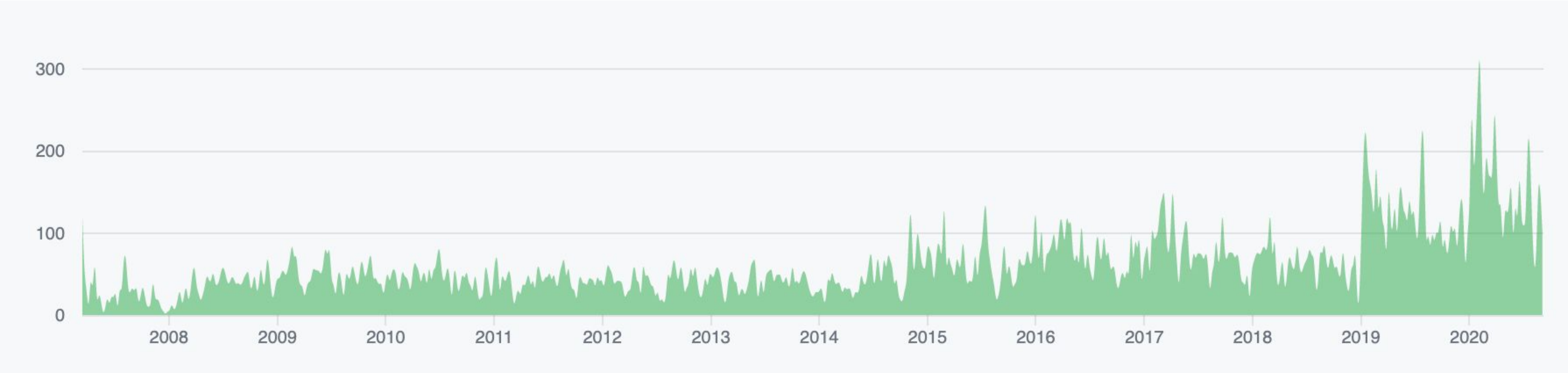
MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

APACHE kafka®

Karaf

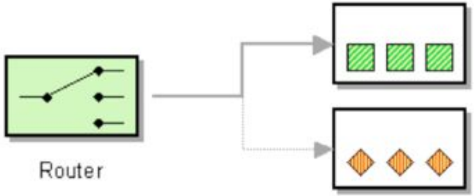
Camel Community

- The biggest and most active community for open source integration software
- 600+ unique contributors
- 3800+ pull requests (mostly from external contributors)
- 13+ years of development and one of the most active Apache projects



Camel Basics

Camel Routes



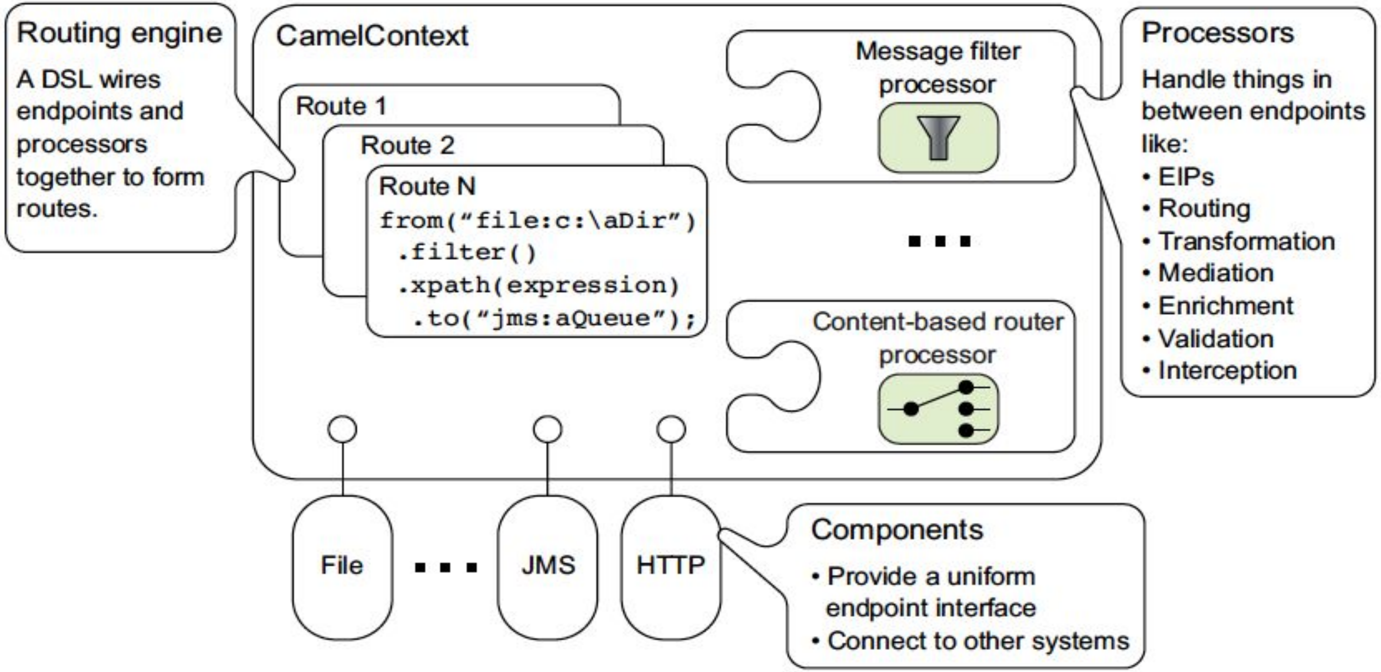
```
from("file:data/inbox")
  .to("jms:queue:order");
```

Java DSL

```
<route>
  <from uri="file:data/inbox"/>
  <to uri="jms:queue:order"/>
</route>
```

XML DSL

Camel Architecture


Routing engine
A DSL wires endpoints and processors together to form routes.
CamelContext
Route 1
Route 2
Route N
Message filter processor
...
Content-based router processor
Processors
Handle things in between endpoints like:
• EIPs
• Routing
• Transformation
• Mediation
• Enrichment
• Validation
• Interception
File
...
JMS
HTTP
Components
• Provide a uniform endpoint interface
• Connect to other systems

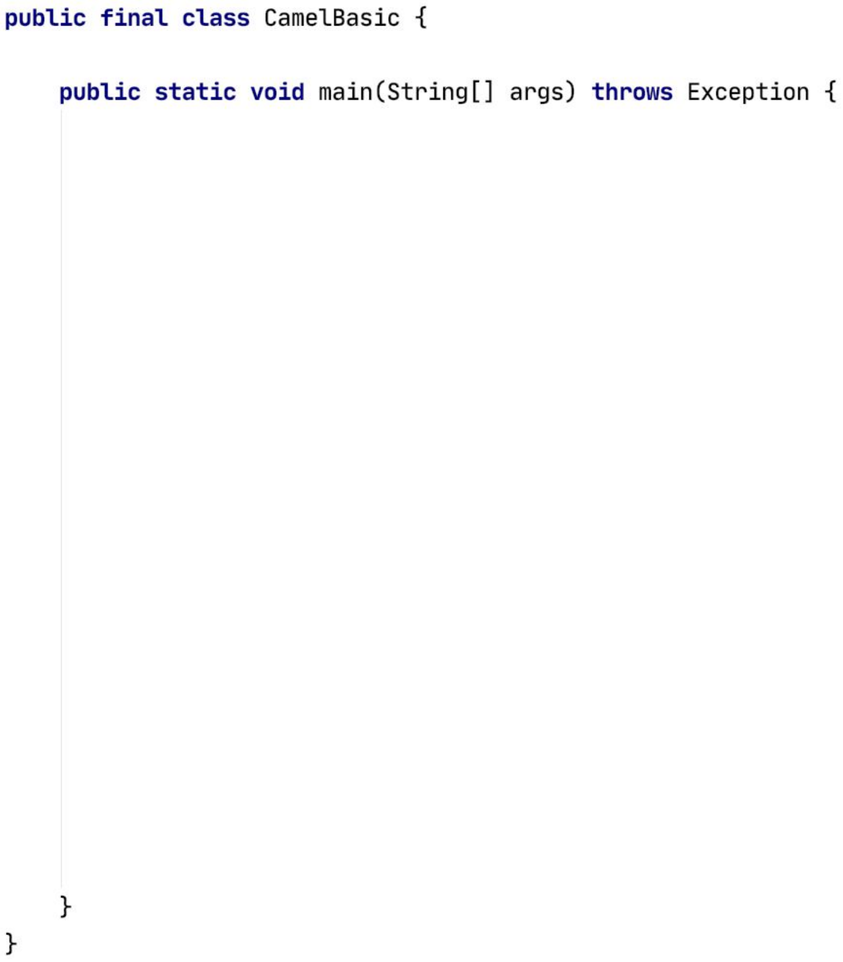
```
from("file:c:\aDir")
 .filter()
 .xpath(expression)
  .to("jms:aQueue");
```

Basic Camel Example

Just Standalone Camel

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic

Basic Camel Example

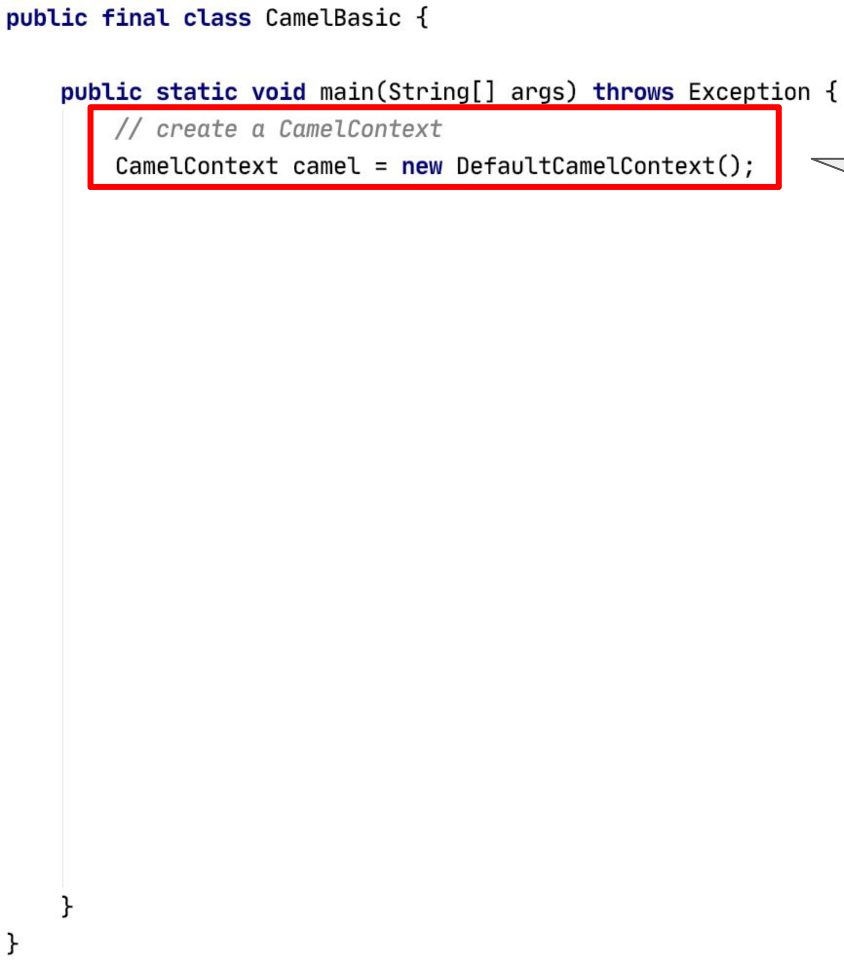


```
public final class CamelBasic {

    public static void main(String[] args) throws Exception {















    }
}
```

Main Class

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic

Basic Camel Example

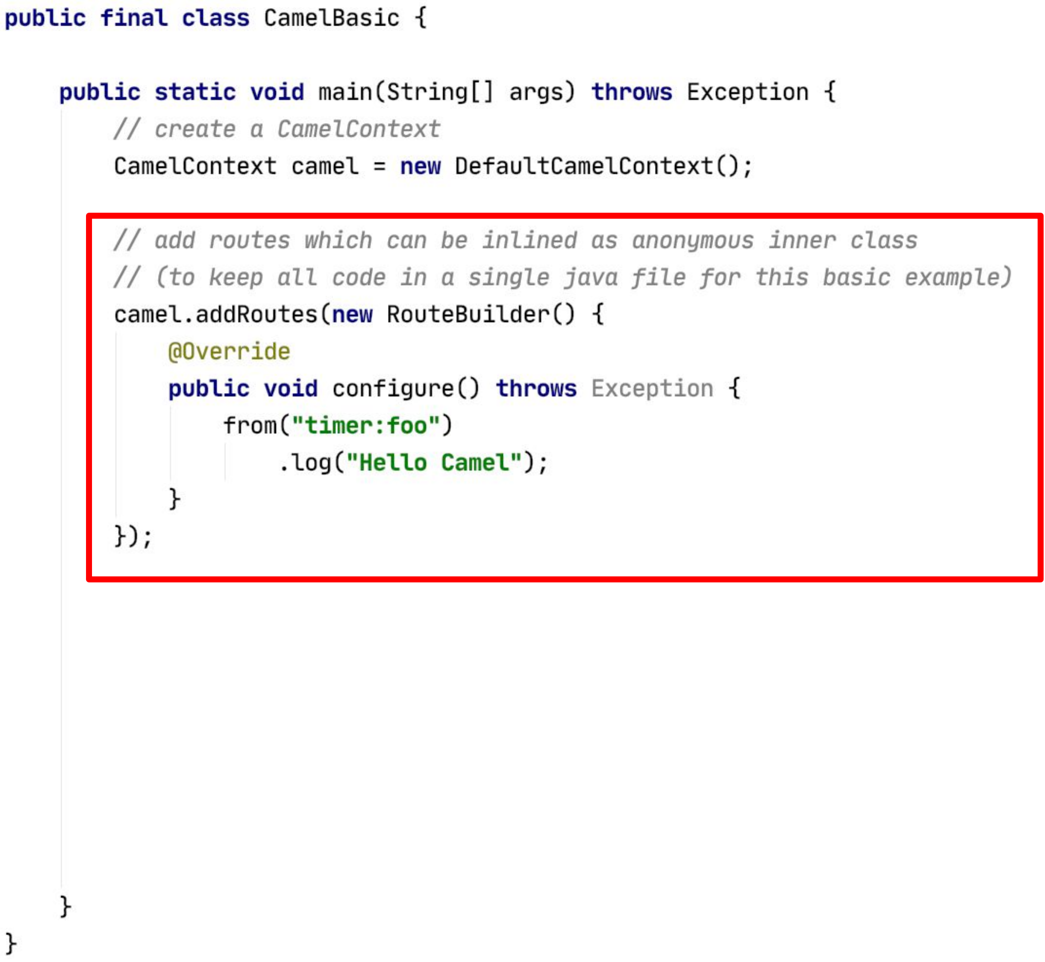


```
public final class CamelBasic {

    public static void main(String[] args) throws Exception {
        // create a CamelContext
        CamelContext camel = new DefaultCamelContext();











    }
}
```

Create CamelContext

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic

Basic Camel Example

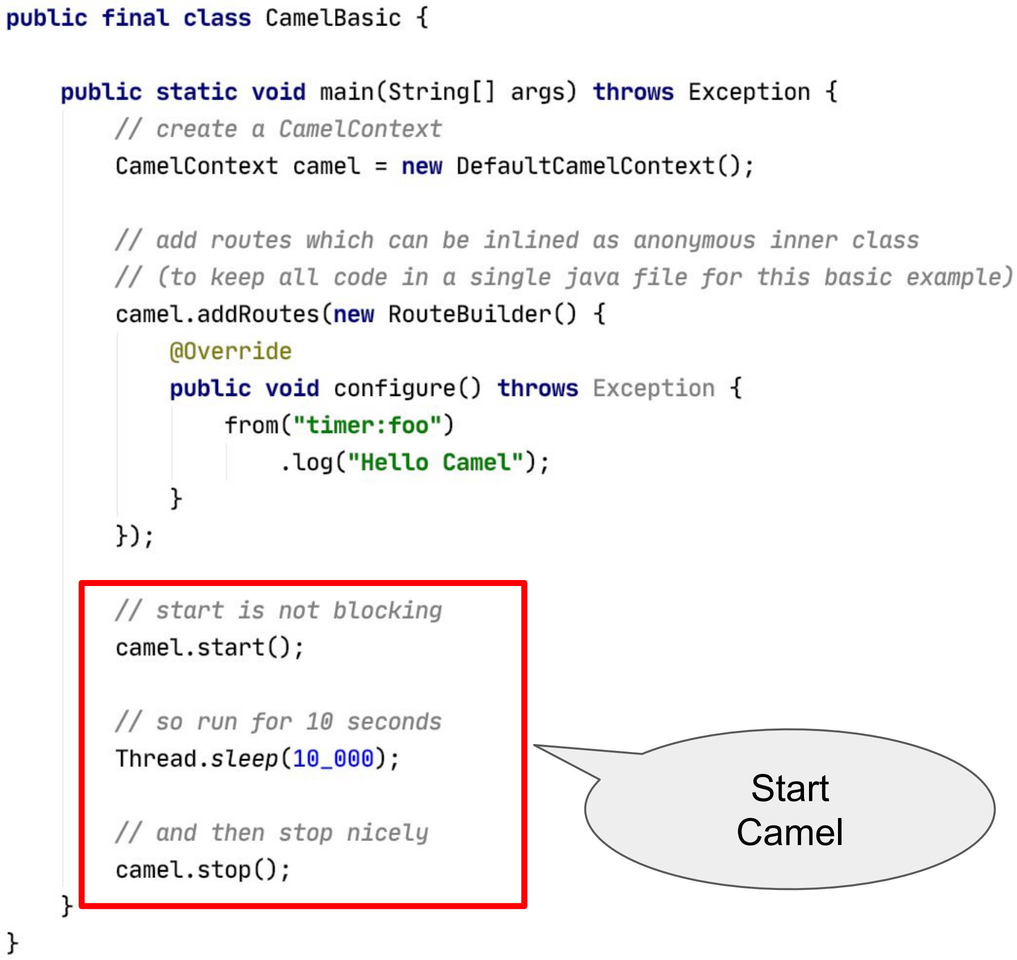


```java
public final class CamelBasic {

    public static void main(String[] args) throws Exception {
        // create a CamelContext
        CamelContext camel = new DefaultCamelContext();

        // add routes which can be inlined as anonymous inner class
        // (to keep all code in a single java file for this basic example)
        camel.addRoutes(new RouteBuilder() {
            @Override
            public void configure() throws Exception {
                from("timer:foo")
                    .log("Hello Camel");
            }
        });

    }
}
```

Add Routes

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic

Basic Camel Example

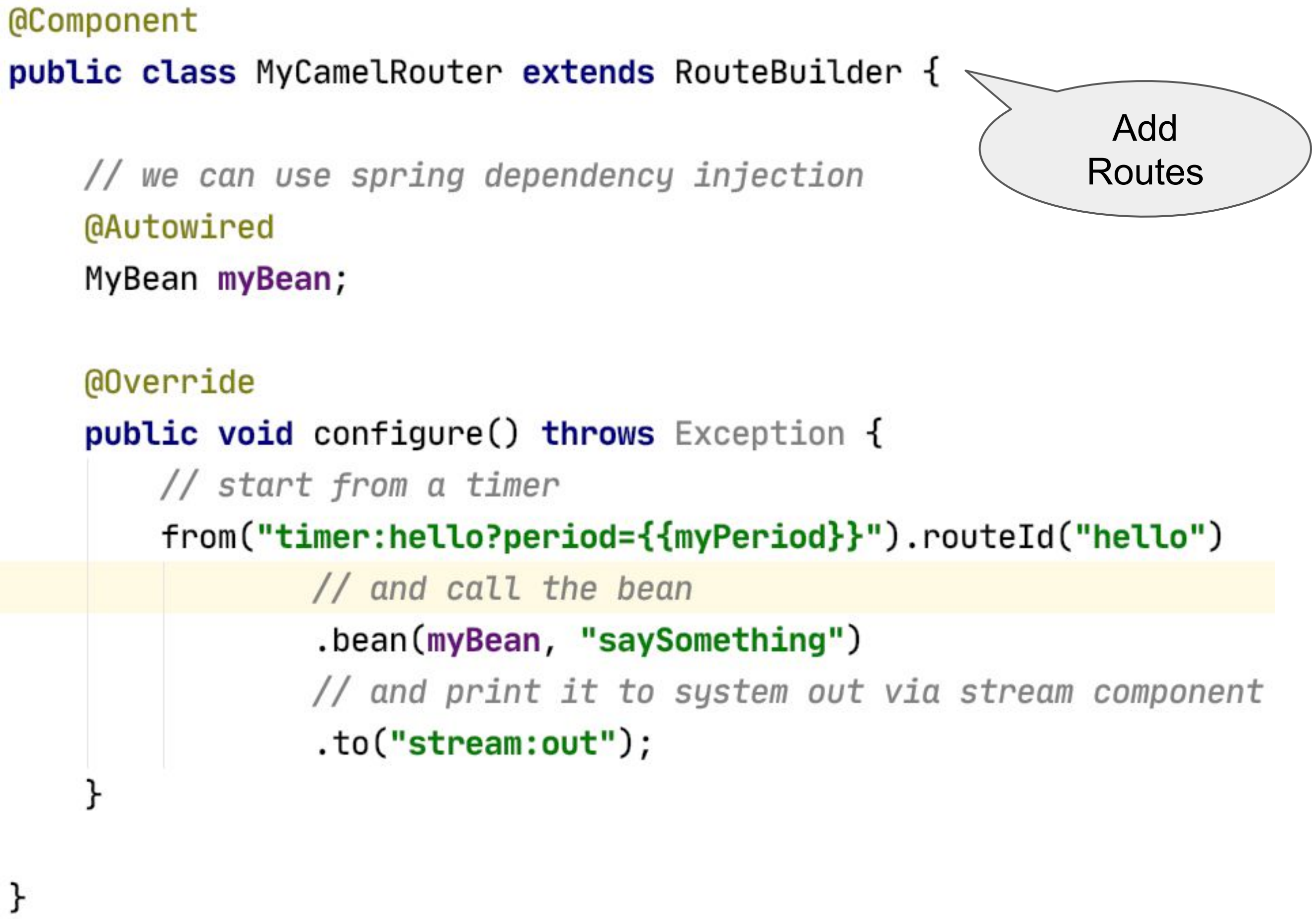


```
public final class CamelBasic {

    public static void main(String[] args) throws Exception {
        // create a CamelContext
        CamelContext camel = new DefaultCamelContext();

        // add routes which can be inlined as anonymous inner class
        // (to keep all code in a single java file for this basic example)
        camel.addRoutes(new RouteBuilder() {
            @Override
            public void configure() throws Exception {
                from("timer:foo")
                    .log("Hello Camel");
            }
        });

        // start is not blocking
        camel.start();

        // so run for 10 seconds
        Thread.sleep(10_000);

        // and then stop nicely
        camel.stop();
    }
}
```

Start Camel

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic

Camel Spring Boot



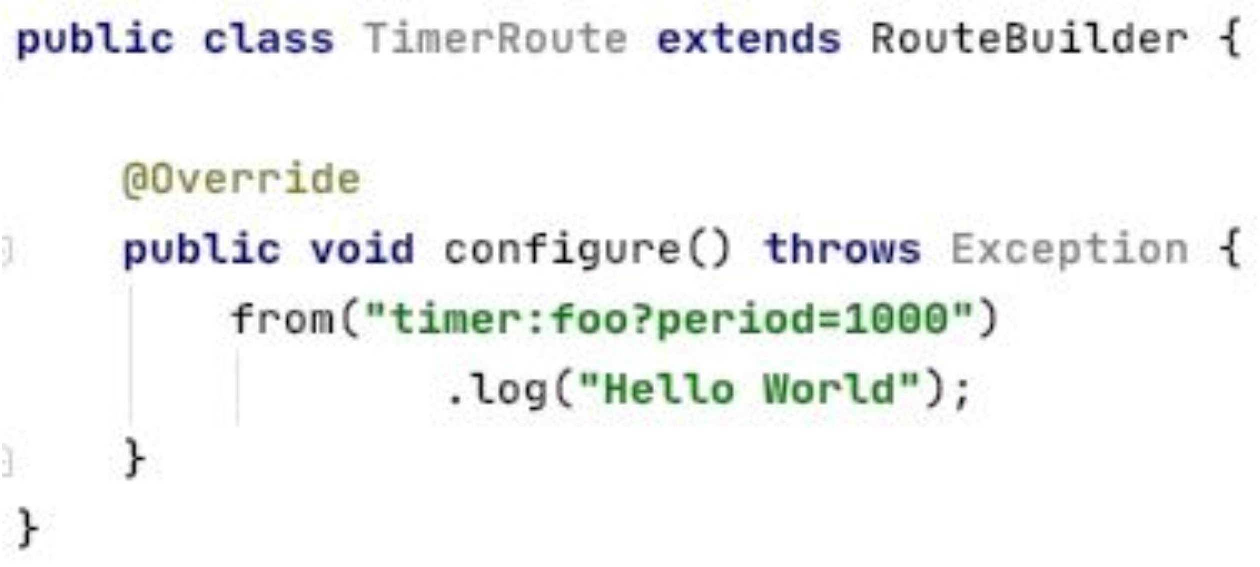
```java
@Component
public class MyCamelRouter extends RouteBuilder {

    // we can use spring dependency injection
    @Autowired
    MyBean myBean;

    @Override
    public void configure() throws Exception {
        // start from a timer
        from("timer:hello?period={{myPeriod}}").routeId("hello")
                // and call the bean
                .bean(myBean, "saySomething")
                // and print it to system out via stream component
                .to("stream:out");
    }

}
```

Add Routes

https://github.com/apache/camel-spring-boot-examples/tree/master/camel-example-spring-boot

Camel Quarkus

Add Routes

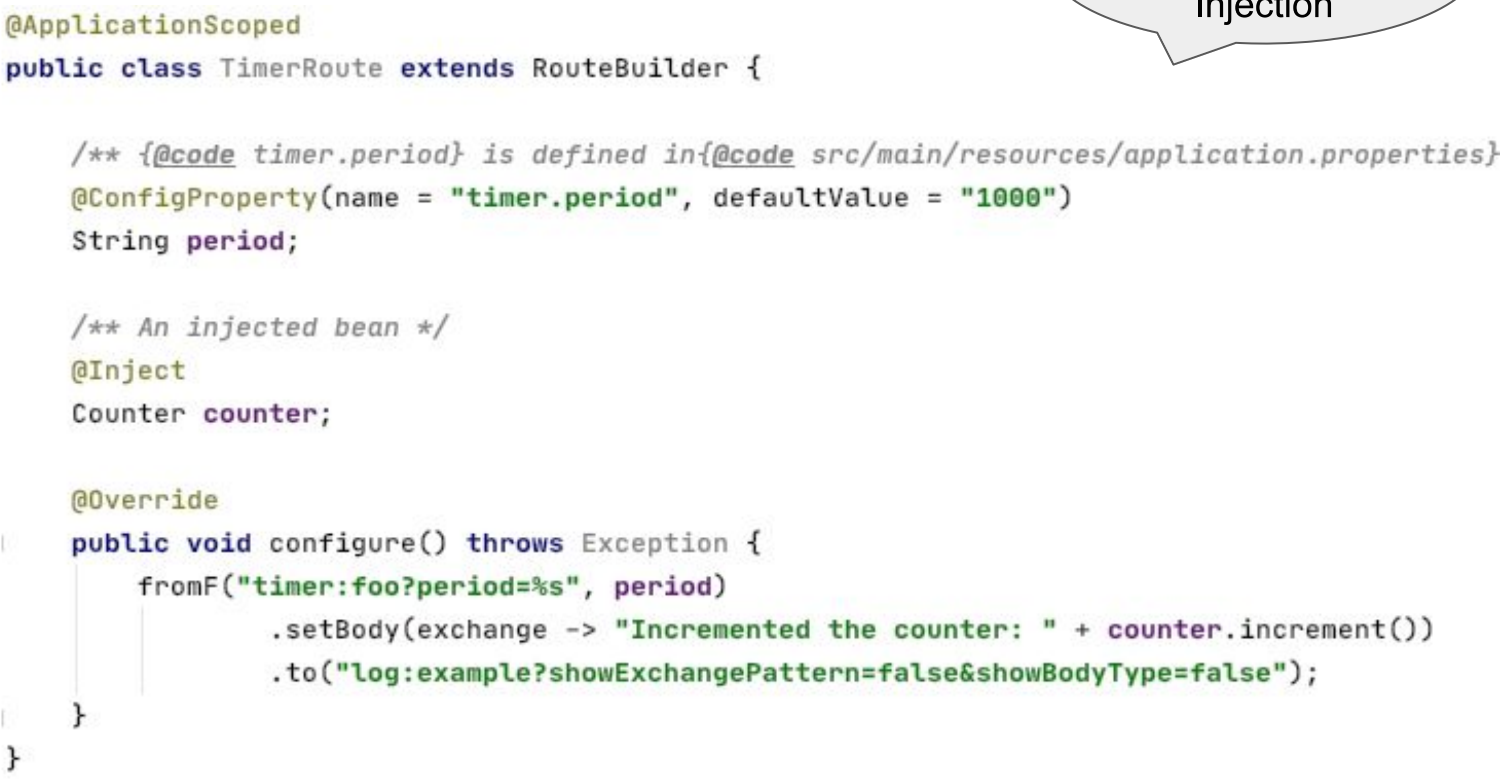


```java
public class TimerRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("timer:foo?period=1000")
                .log("Hello World");
    }
}
```

https://github.com/apache/camel-quarkus/tree/master/examples/timer-log

Camel Quarkus

CDI Dependency Injection



```
@ApplicationScoped
public class TimerRoute extends RouteBuilder {

    /** {@code timer.period} is defined in{@code src/main/resources/application.properties}
    @ConfigProperty(name = "timer.period", defaultValue = "1000")
    String period;

    /** An injected bean */
    @Inject
    Counter counter;

    @Override
    public void configure() throws Exception {
        fromF("timer:foo?period=%s", period)
                .setBody(exchange -> "Incremented the counter: " + counter.increment())
                .to("log:example?showExchangePattern=false&showBodyType=false");
    }
}
```

https://github.com/apache/camel-quarkus/tree/master/examples/timer-log-cdi

Camel 3 Projects

Apache Camel 3 - Projects

Camel

Swiss knife of integration

Apache Camel 3 - Projects

Camel

Swiss knife of integration

Camel Spring Boot

Camel on
Spring Boot

Apache Camel 3 - Projects

Camel

Swiss knife of integration

Camel Karaf

Camel on
Apache Karaf / OSGi

Camel Spring Boot

Camel on
Spring Boot

Apache Camel 3 - Projects

Camel K

Camel on Kubernetes & Knative

Camel

Swiss knife of integration

Camel Karaf

Camel on Apache Karaf / OSGi

Camel Spring Boot

Camel on Spring Boot

Apache Camel 3 - Projects

Camel K

Camel on Kubernetes & Knative

Camel

Swiss knife of integration

Camel Quarkus

Optimized JVM & Native compiled Java (GraalVM)

Camel Karaf

Camel on Apache Karaf / OSGi

Camel Spring Boot

Camel on Spring Boot

Apache Camel 3 - Projects

Camel K

Camel on
Kubernetes & Knative

Camel

Swiss knife of integration

Camel Quarkus

Optimized JVM & Native
compiled Java (GraalVM)

Camel Karaf

Camel on
Apache Karaf / OSGi

Camel Spring Boot

Camel on
Spring Boot

Camel Kafka Connector

Kafka Connector with Camel

Camel Releases

Schedule is subject
for change

Camel - LTS vs Non-LTS Releases

Release	Date	LTS	LTS EOL	Java
2.25	**Jan 2020**	**Yes**	**Jun 2021**	**8**
3.0	Nov 2019	No		8,11
3.1	Feb 202	No		8,11
3.2	Apr 2020	No		8,11
3.3	May 2020	No		8,11
3.4	**Jun 2020**	**Yes**	**Jun 2021**	**8,11**
3.5	Aug 2020	No		8,11,14
3.6	Oct 2020	No		8,11,14
3.7	**Dec 2020**	**Yes**	**Dec 2021**	**8,11,14**

Camel K

What is Apache Camel K ?

A lightweight integration platform, based on Apache Camel,
born on Kubernetes, with serverless superpowers.

What is Apache Camel K ?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.

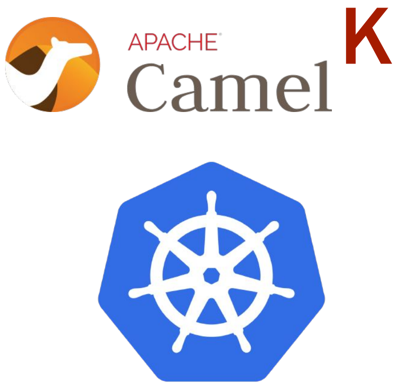
1.

Runs on “vanilla” Kubernetes (1) ...

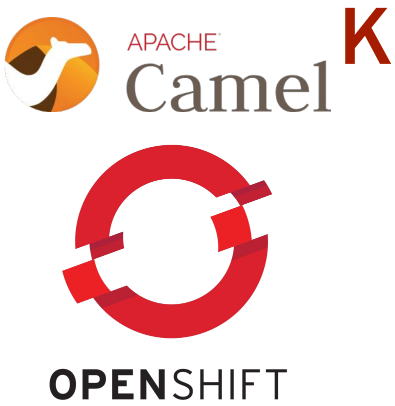
What is Apache Camel K ?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.

1.



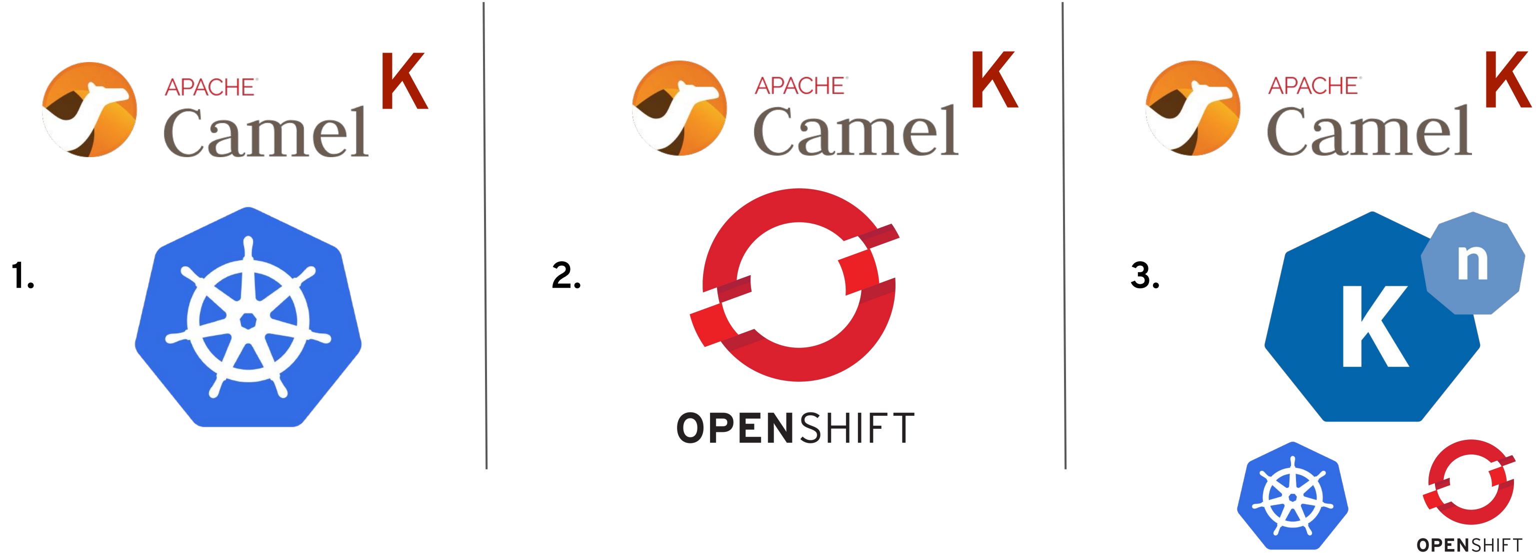
2.



Runs on “vanilla” Kubernetes (1), Openshift (2) ...

What is Apache Camel K ?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.



Runs on “vanilla” Kubernetes (1), Openshift (2) and gives its best on a Knative-powered cluster (3)!

How?

How?

Camel DSL, based on EIPs...

1. Create a integration file (Java, Groovy, Kotlin, JS, XML...)

```
from(“telegram:bots/bot-id”)
  .transform()...
  .to(“kafka:topic”);

from(“kafka:topic”)
  .to(“http:my-host/api/path”);
```

How?

1. Create a integration file (Java, Groovy, Kotlin, JS, XML…)

```
from(“telegram:bots/bot-id”)
  .transform()...
  .to(“kafka:topic”);

from(“kafka:topic”)
  .to(“http:my-host/api/path”);
```

Camel DSL, based on EIPs…

2. Run it

```
$ kamel run integration.groovy
```

How?

Camel DSL, based on EIPs...

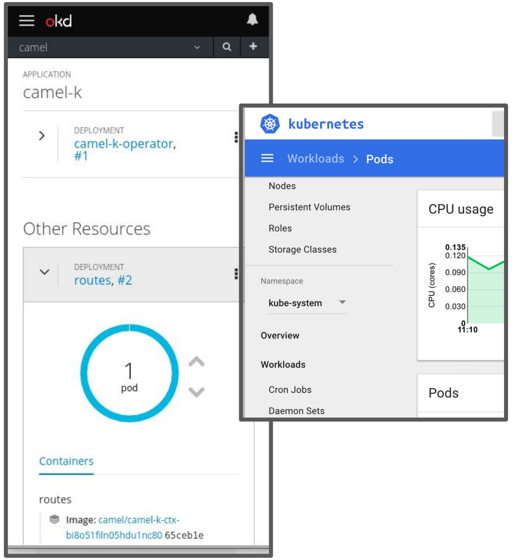
1. Create a integration file (Java, Groovy, Kotlin, JS, XML...)

```
from(“telegram:bots/bot-id”)
  .transform()...
  .to(“kafka:topic”);

from(“kafka:topic”)
  .to(“http:my-host/api/path”);
```

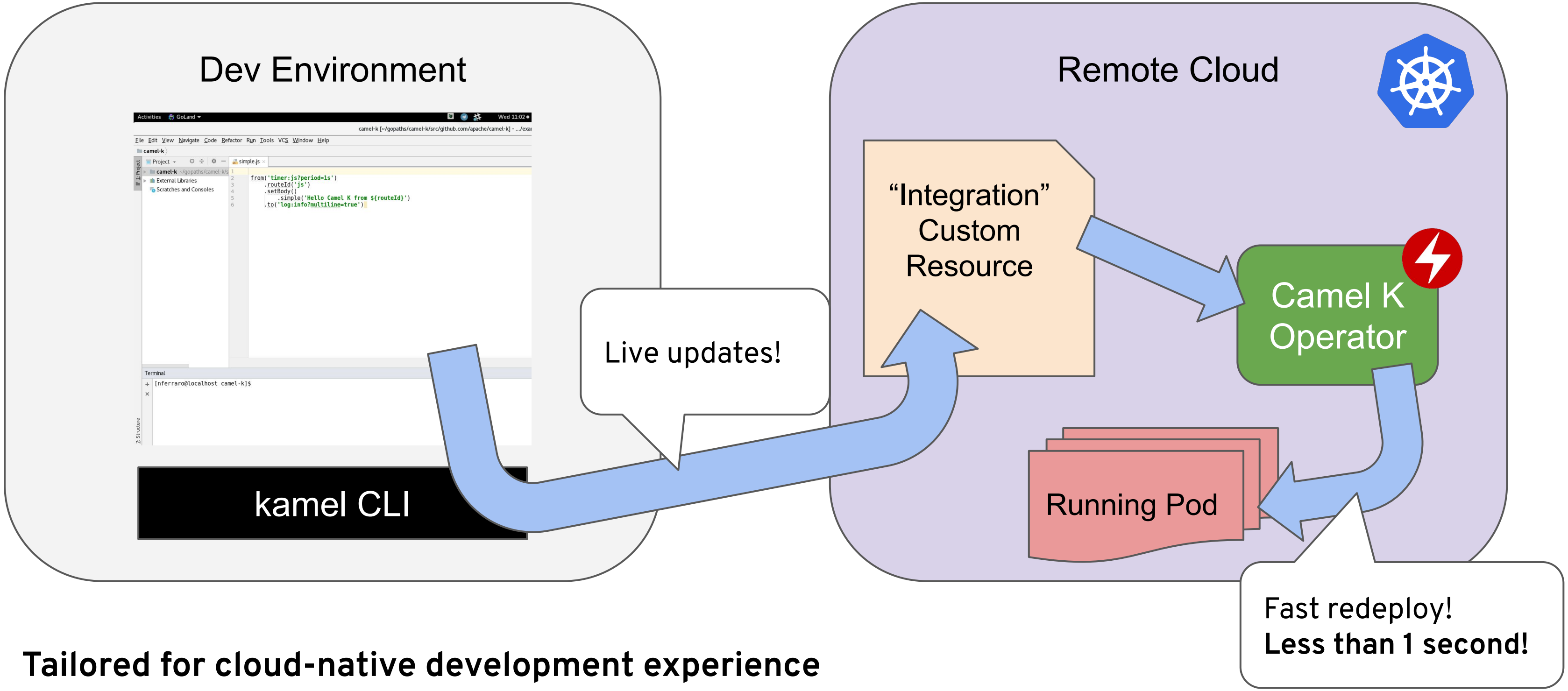
2. Run it

```
$ kamel run integration.groovy
```

3. Runs on Openshift or Kubernetes



Architecture of Camel K

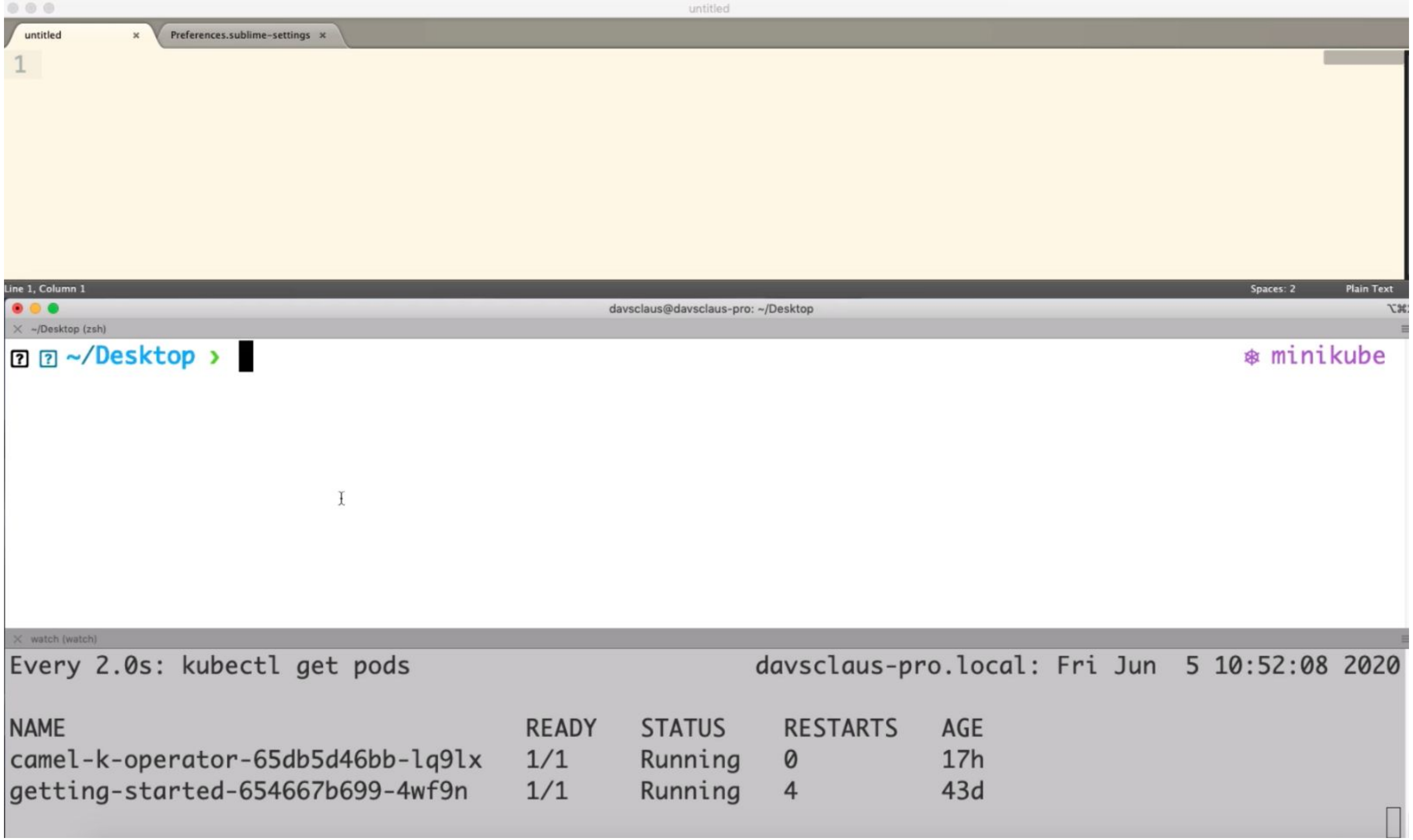


```
from('timer:js?period=1s')
    .routeId('js')
    .setBody()
        .simple('Hello Camel K from ${routeId}')
    .to('log:info?multiline=true')
```

```
[nferraro@localhost camel-k]$
```

Tailored for cloud-native development experience

Quick Camel K Demo



```
~/Desktop ›                                                    minikube
```

```
Every 2.0s: kubectl get pods                     davsclaus-pro.local: Fri Jun  5 10:52:08 2020

NAME                                READY   STATUS    RESTARTS   AGE
camel-k-operator-65db5d46bb-lq9lx   1/1     Running   0          17h
getting-started-654667b699-4wf9n    1/1     Running   4          43d
```

[redacted]

Camel K is not only Java

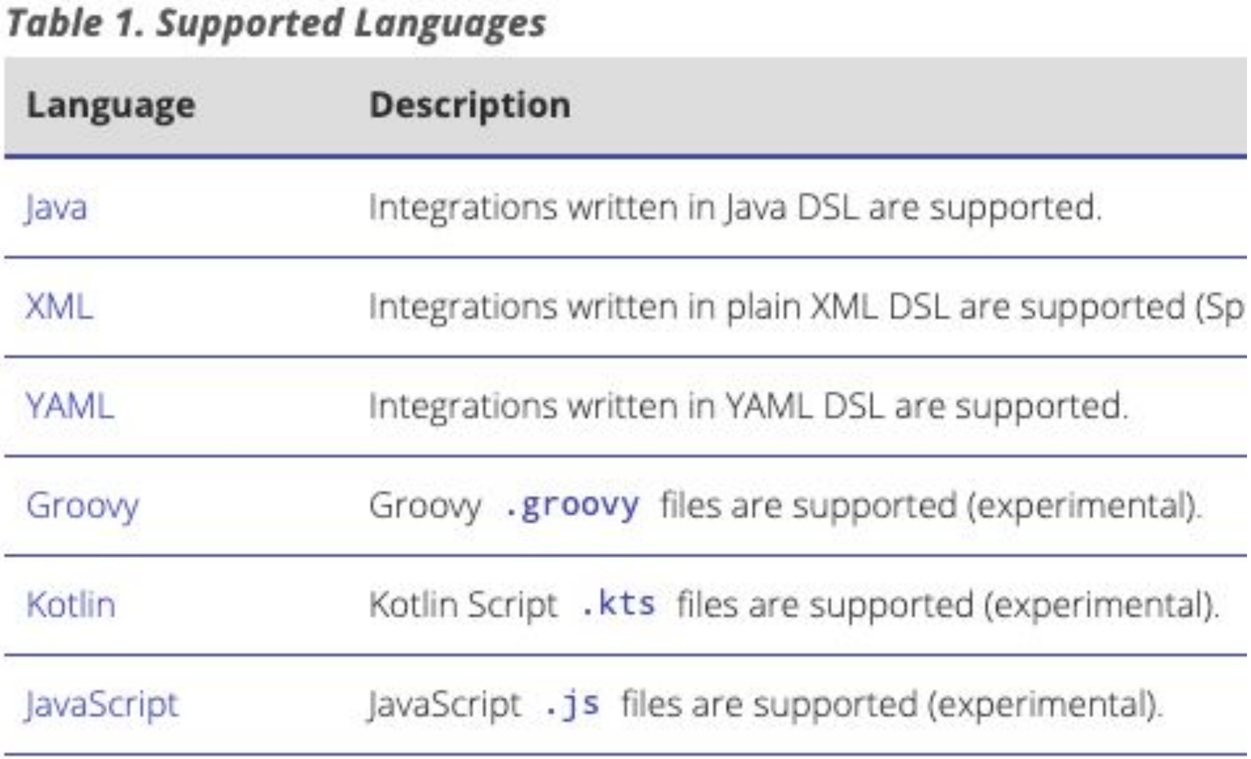


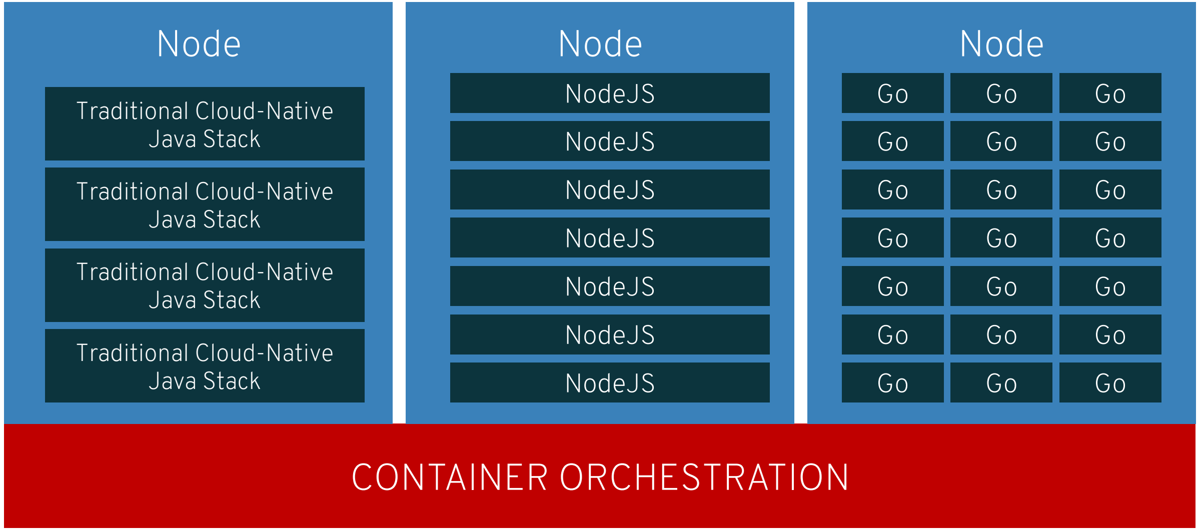
Table 1. Supported Languages

Language	Description
Java	Integrations written in Java DSL are supported.
XML	Integrations written in plain XML DSL are supported (Sp
YAML	Integrations written in YAML DSL are supported.
Groovy	Groovy `.groovy` files are supported (experimental).
Kotlin	Kotlin Script `.kts` files are supported (experimental).
JavaScript	JavaScript `.js` files are supported (experimental).

Camel Quarkus

SAY JAVA IS SLOW
ONE MORE TIME
quickmeme.com

Java Density Problem



Node

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Node

NodeJS

NodeJS

NodeJS

NodeJS

NodeJS

NodeJS

NodeJS

Node

Go Go Go

Go Go Go

Go Go Go

Go Go Go

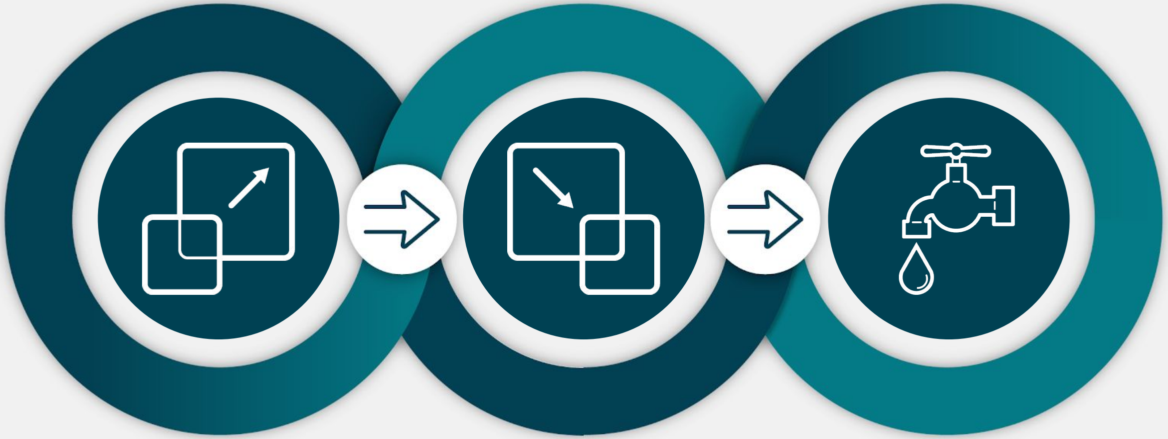
Go Go Go

Go Go Go

Go Go Go

CONTAINER ORCHESTRATION

Java Fast Scaling Problem



SCALE-UP

SCALE-DOWN

SCALE-TO-ZERO

What is Quarkus ?

QUARKUS

What is Quarkus ?

QUARKUS

Supersonic Subatomic Java

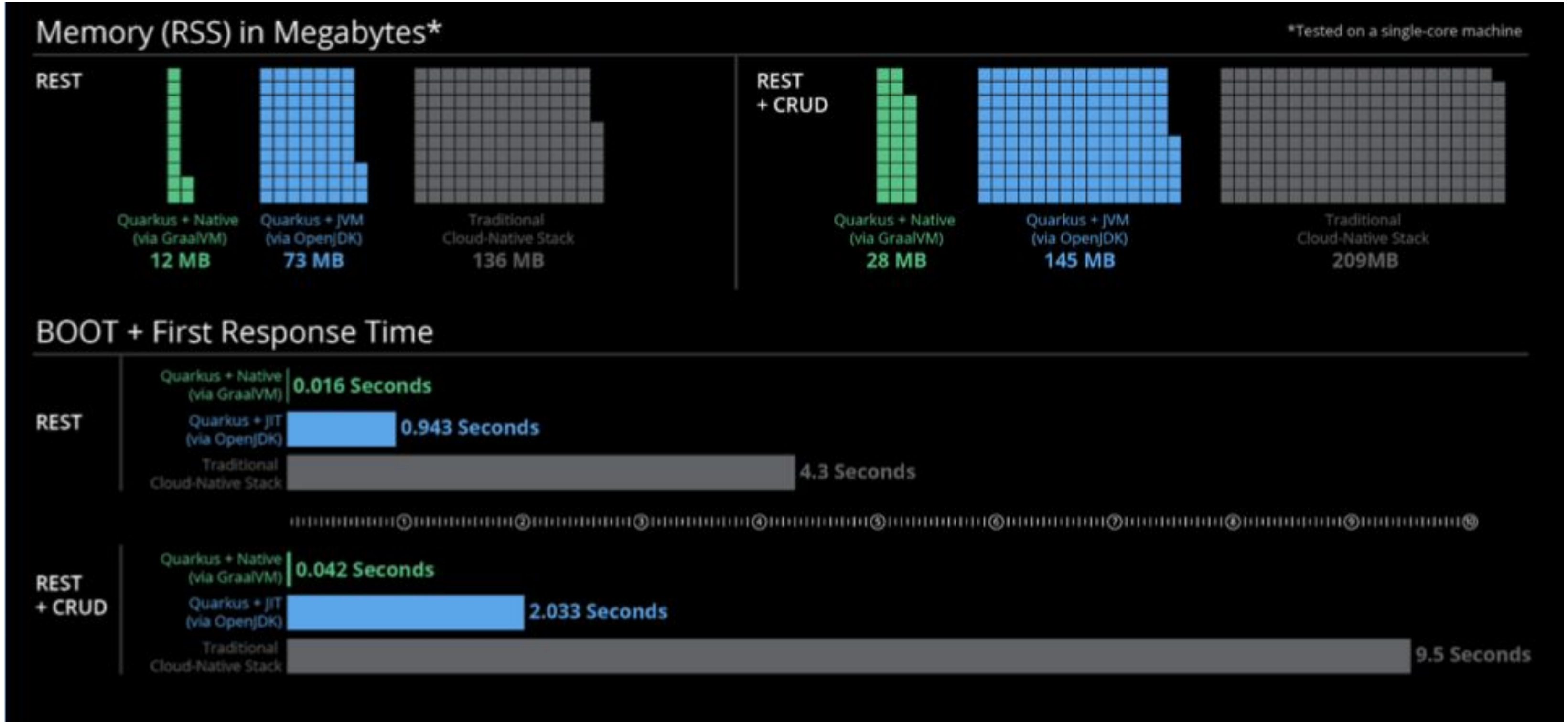
What is Quarkus ?

QUARKUS

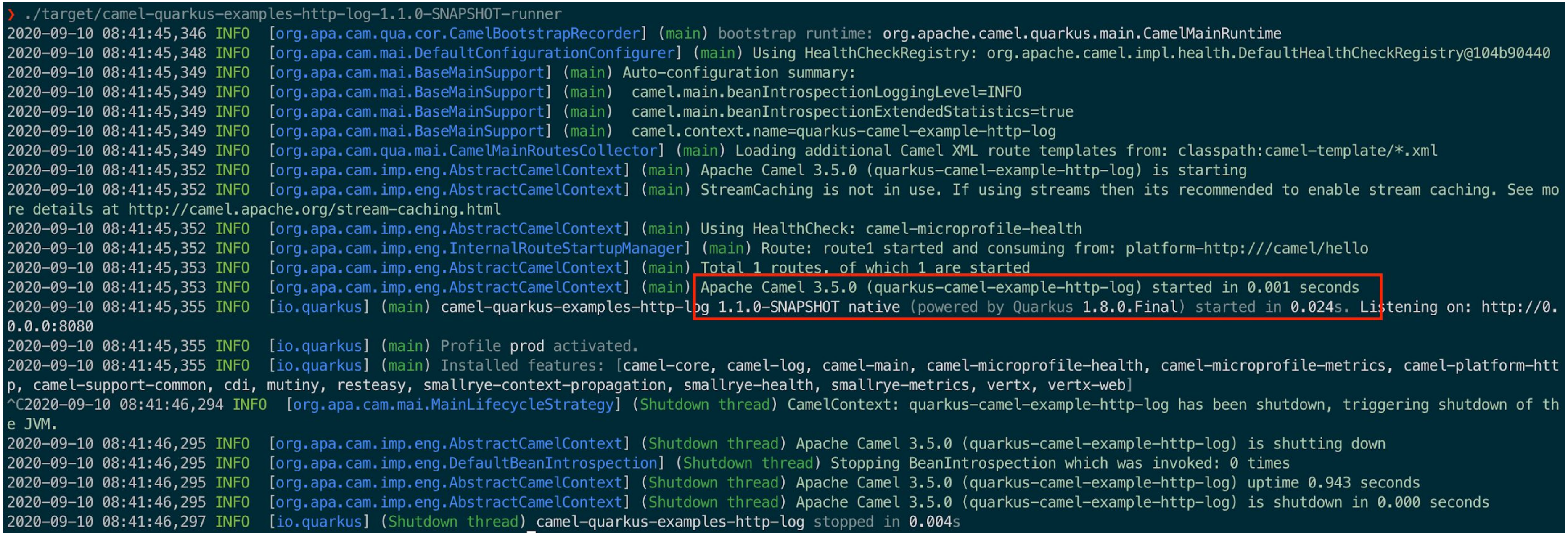
Supersonic Subatomic Java

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

Minimal Footprint



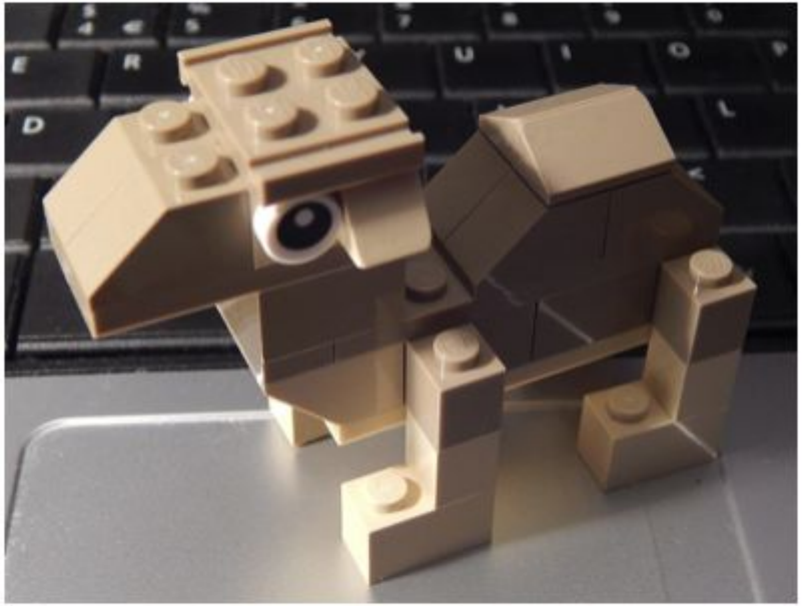
Apache Camel and Quarkus runs blazingly fast with native compiled binaries.
Startup is 24 msec, 50mb binary file, and 27mb RSS memory used.



```
❯ ./target/camel-quarkus-examples-http-log-1.1.0-SNAPSHOT-runner
2020-09-10 08:41:45,346 INFO  [org.apa.cam.qua.cor.CamelBootstrapRecorder] (main) bootstrap runtime: org.apache.camel.quarkus.main.CamelMainRuntime
2020-09-10 08:41:45,348 INFO  [org.apa.cam.mai.DefaultConfigurationConfigurer] (main) Using HealthCheckRegistry: org.apache.camel.impl.health.DefaultHealthCheckRegistry@104b90440
2020-09-10 08:41:45,349 INFO  [org.apa.cam.mai.BaseMainSupport] (main) Auto-configuration summary:
2020-09-10 08:41:45,349 INFO  [org.apa.cam.mai.BaseMainSupport] (main)  camel.main.beanIntrospectionLoggingLevel=INFO
2020-09-10 08:41:45,349 INFO  [org.apa.cam.mai.BaseMainSupport] (main)  camel.main.beanIntrospectionExtendedStatistics=true
2020-09-10 08:41:45,349 INFO  [org.apa.cam.mai.BaseMainSupport] (main)  camel.context.name=quarkus-camel-example-http-log
2020-09-10 08:41:45,349 INFO  [org.apa.cam.qua.mai.CamelMainRoutesCollector] (main) Loading additional Camel XML route templates from: classpath:camel-template/*.xml
2020-09-10 08:41:45,352 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (main) Apache Camel 3.5.0 (quarkus-camel-example-http-log) is starting
2020-09-10 08:41:45,352 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (main) StreamCaching is not in use. If using streams then its recommended to enable stream caching. See mo
re details at http://camel.apache.org/stream-caching.html
2020-09-10 08:41:45,352 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (main) Using HealthCheck: camel-microprofile-health
2020-09-10 08:41:45,352 INFO  [org.apa.cam.imp.eng.InternalRouteStartupManager] (main) Route: route1 started and consuming from: platform-http:///camel/hello
2020-09-10 08:41:45,353 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (main) Total 1 routes, of which 1 are started
2020-09-10 08:41:45,353 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (main) Apache Camel 3.5.0 (quarkus-camel-example-http-log) started in 0.001 seconds
2020-09-10 08:41:45,355 INFO  [io.quarkus] (main) camel-quarkus-examples-http-log 1.1.0-SNAPSHOT native (powered by Quarkus 1.8.0.Final) started in 0.024s. Listening on: http://0.
0.0.0:8080
2020-09-10 08:41:45,355 INFO  [io.quarkus] (main) Profile prod activated.
2020-09-10 08:41:45,355 INFO  [io.quarkus] (main) Installed features: [camel-core, camel-log, camel-main, camel-microprofile-health, camel-microprofile-metrics, camel-platform-htt
p, camel-support-common, cdi, mutiny, resteasy, smallrye-context-propagation, smallrye-health, smallrye-metrics, vertx, vertx-web]
^C2020-09-10 08:41:46,294 INFO  [org.apa.cam.mai.MainLifecycleStrategy] (Shutdown thread) CamelContext: quarkus-camel-example-http-log has been shutdown, triggering shutdown of th
e JVM.
2020-09-10 08:41:46,295 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (Shutdown thread) Apache Camel 3.5.0 (quarkus-camel-example-http-log) is shutting down
2020-09-10 08:41:46,295 INFO  [org.apa.cam.imp.eng.DefaultBeanIntrospection] (Shutdown thread) Stopping BeanIntrospection which was invoked: 0 times
2020-09-10 08:41:46,295 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (Shutdown thread) Apache Camel 3.5.0 (quarkus-camel-example-http-log) uptime 0.943 seconds
2020-09-10 08:41:46,295 INFO  [org.apa.cam.imp.eng.AbstractCamelContext] (Shutdown thread) Apache Camel 3.5.0 (quarkus-camel-example-http-log) is shutdown in 0.000 seconds
2020-09-10 08:41:46,297 INFO  [io.quarkus] (Shutdown thread) camel-quarkus-examples-http-log stopped in 0.004s
```

Compiled natively with GraalVM CE 20.2.0 with JDK 11.0.8 on Mac OSX

Camel Quarkus Demo



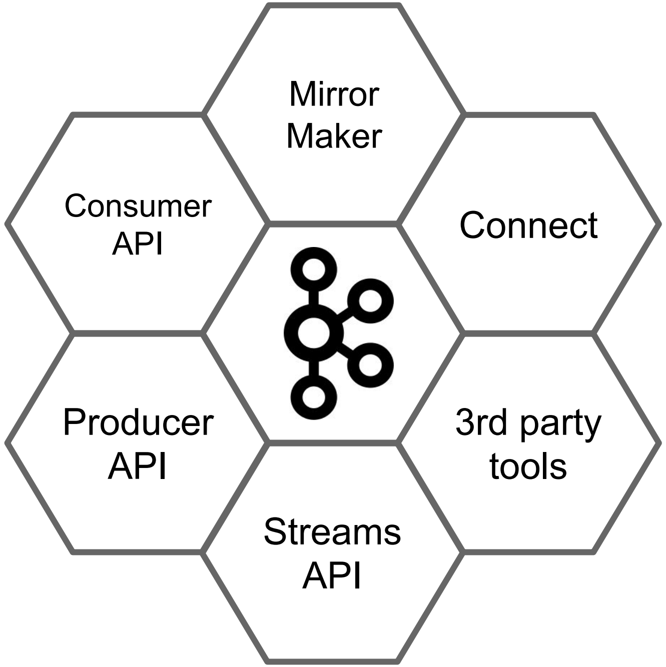
QUARKUS

[redacted]

Camel
Kafka Connector

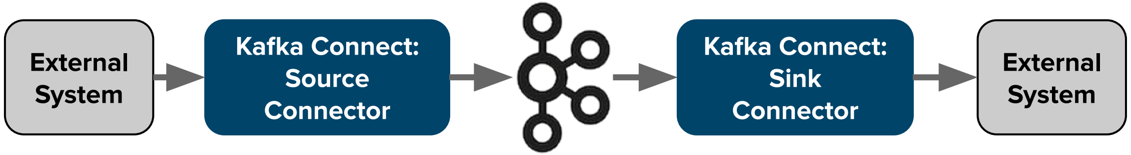
What is Apache Kafka?

- Kafka is known as
  - a **distributed streaming platform**
  - or a **pub/sub messaging broker**
- It’s an ecosystem
  - Multiple components part of Apache Kafka
  - A lot of 3rd party integrations
  - One of them is **Kafka Connect**



What is Kafka Connect?

- It is a **framework** that helps you to integrate Kafka with other systems
  - A user can define **source** and **sink** connectors to stream data in/out of Kafka brokers
  - Connectors are **plugable** - you can use one of the many connectors available or write your own
- **Distributed** and **scalable** by default
- Automatic **offset** management
- Simple transformations
- Streaming / batch integration


External System
Kafka Connect: Source Connector
Kafka Connect: Sink Connector
External System

What is Camel Kafka Connector?

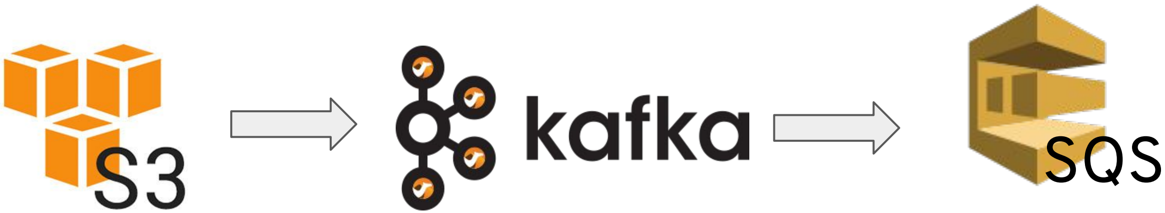
- A Kafka Connector built on top of Apache Camel
- Started as an internal proof-of-concept
  - A sub-project of the Apache Camel
  - Donated by Red Hat to the ASF on December 2019
- Reuses in a **simple way** most of the Camel components as Kafka sink and sources
- Latest release is 0.5.0 based on Camel 3.5.0
- Autogenerated documentation and connectors list:

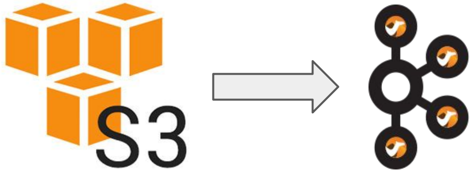
  https://camel.apache.org/camel-kafka-connector/latest/connectors.html

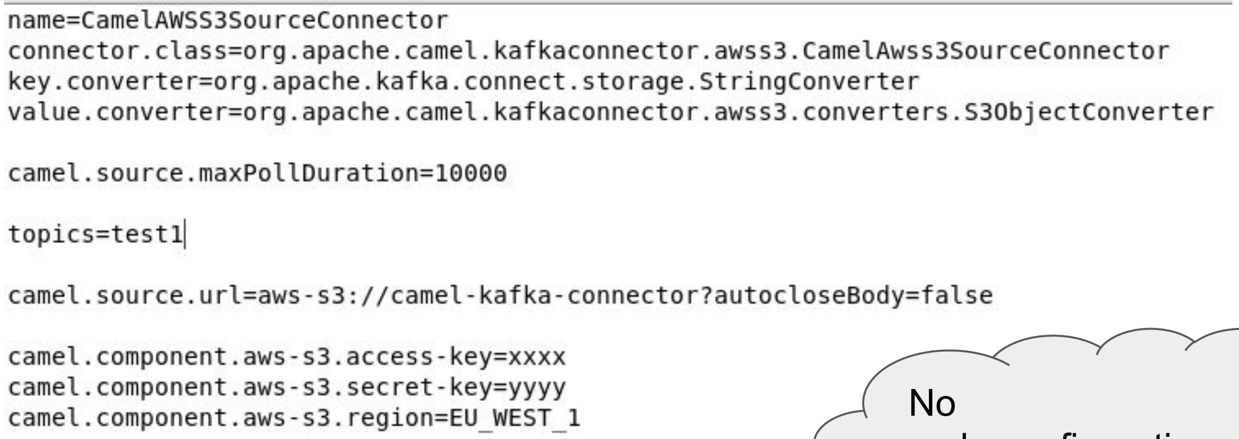
To run Kafka on Kubernetes

Demo S3 to SQS

- S3 Source connector (camel-aws-s3)
- SQS Sink connector (camel-aws-sqs)
- A file loaded on a bucket -> file content in a SQS Queue

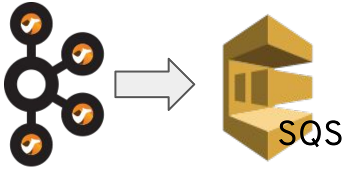


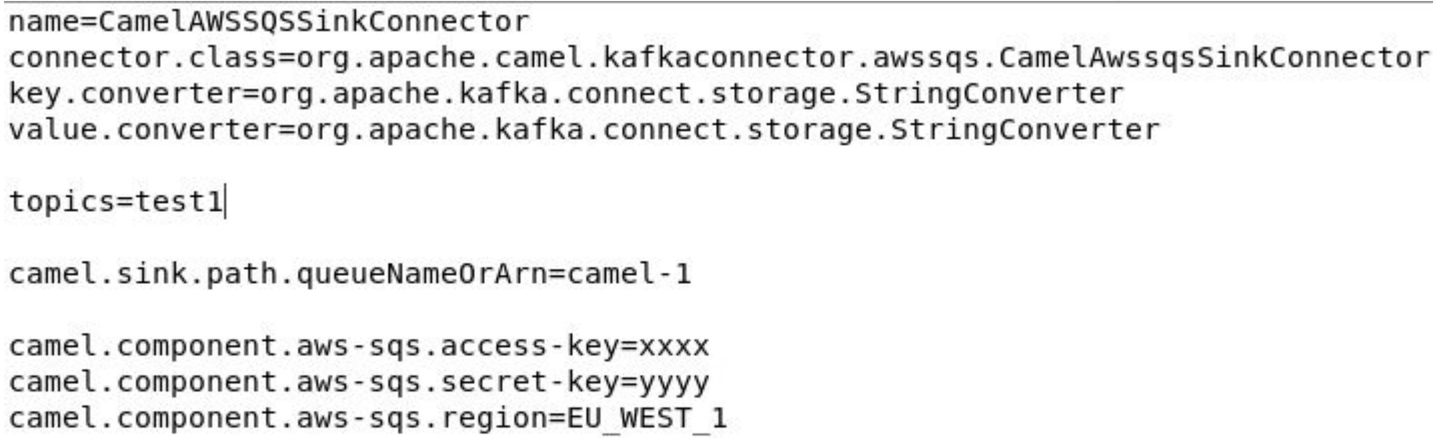




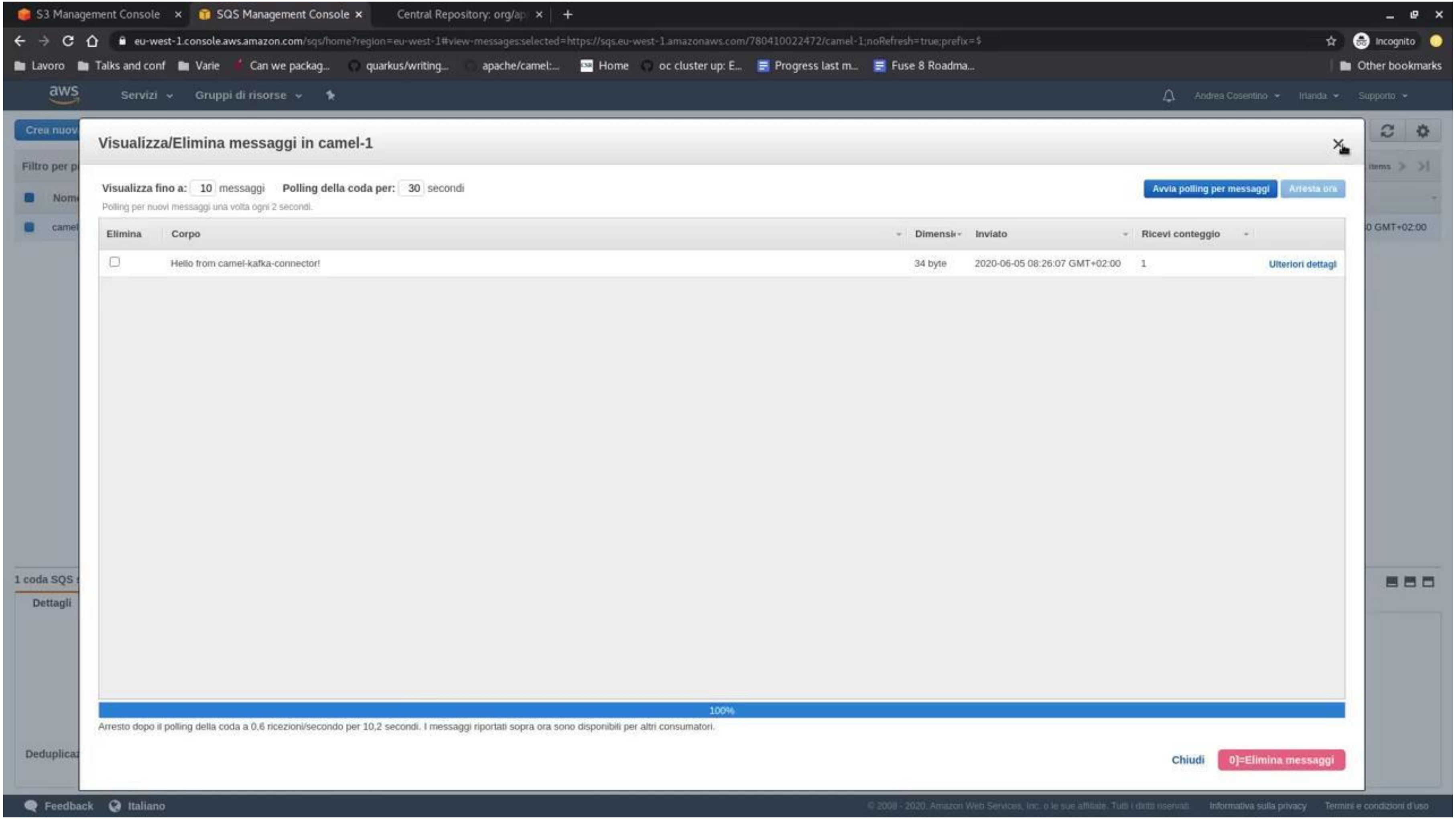
```
name=CamelAWSS3SourceConnector
connector.class=org.apache.camel.kafkaconnector.awss3.CamelAwss3SourceConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.camel.kafkaconnector.awss3.converters.S3ObjectConverter

camel.source.maxPollDuration=10000

topics=test1

camel.source.url=aws-s3://camel-kafka-connector?autocloseBody=false

camel.component.aws-s3.access-key=xxxx
camel.component.aws-s3.secret-key=yyyy
camel.component.aws-s3.region=EU_WEST_1
```

No code,configuration only!





```
name=CamelAWSSQSSinkConnector
connector.class=org.apache.camel.kafkaconnector.awssqs.CamelAwssqsSinkConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter

topics=test1

camel.sink.path.queueNameOrArn=camel-1

camel.component.aws-sqs.access-key=xxxx
camel.component.aws-sqs.secret-key=yyyy
camel.component.aws-sqs.region=EU_WEST_1
```



Visualizza/Elimina messaggi in camel-1

Visualizza fino a: 10 messaggi **Polling della coda per:** 30 secondi

Polling per nuovi messaggi una volta ogni 2 secondi.

Elimina	Corpo	Dimensi	Inviato	Ricevi conteggio	
☐	Hello from camel-kafka-connector!	34 byte	2020-06-05 08:26:07 GMT+02:00	1	Ulteriori dettagli

Arresto dopo il polling della coda a 0,6 ricezioni/secondo per 10,2 secondi. I messaggi riportati sopra ora sono disponibili per altri consumatori.

More Material

More Material

- Camel website
  https://camel.apache.org
- Quick Camel K demo (2 min video)
  https://www.youtube.com/watch?v=l-g2xt-Qcb8
- Run 100 Camels quickly with Came, Quarkus and GraalVM (10 min video)
  https://www.youtube.com/watch?v=4lXSf8DBQkQ
- Camel Kafka Connector (blog and 25 min video)
  https://developers.redhat.com/blog/2020/05/19/extending-kafka-connectivity-with-apache-camel-kafka-connectors
- Serverless Integration on Kubernetes with Camel K - KubeCon 2020 (35 min video)
  https://www.youtube.com/watch?v=beJOCndVfaU

Examples

- Camel Examples
  https://github.com/apache/camel-examples
- Camel K Examples
  https://github.com/apache/camel-k/tree/master/example
- Camel K Examples (detailed as tutorials)
  https://github.com/apache/camel-k-examples
- Camel Quarkus Examples (on the way)
  https://github.com/apache/camel-quarkus-examples
- Camel Kafka Connector Examples
  https://github.com/apache/camel-kafka-connector-examples
- Camel Spring Boot Examples
  https://github.com/apache/camel-spring-boot-examples

Any Questions ?

APACHE Camel

https://github.com/apache/camel

A star on github is appreciated

Follow us on Twitter

@davsclaus

@oscerd2

@ApacheCamel