Getting Started with Java Functions for Knative

Thomas Risberg, VMware

Introduction

Thomas Risberg, member of the Spring team at VMware

❏ Joined the Spring Framework project back in 2003
❏ 20 years of experience writing Spring "Hello world" demo apps
❏ Co-author of *Spring Data, Modern Data Access for Enterprise Java* and *Professional Java Development with the Spring Framework*
❏ Twitter: @trisberg
❏ GitHub: trisberg

SpringOne
@SpringOne

"I love start.spring.io it's my second favourite place on the internet after production." @starbuxman on stage at #SpringOne this morning.

10:20 AM · Oct 9, 2019 · Twitter for iPad

Thomas Risberg
@trisberg

Replying to @SpringOne and @starbuxman

My favorite starting point for creating Spring function apps is Knative func - github.com/knative/func#f…
See you at SpringOne 2022
springone.io/2022/sessions/…

11:28 PM · Oct 30, 2022 · Twitter Web App

Why Knative?

I want a platform that comes ready to use with "batteries included".

I want to just deploy my code without having to deal with platform details.

VMware Tanzu ✔
@VMwareTanzu

Haiku from @onsijoe:

here is my source code
run it on the cloud for me
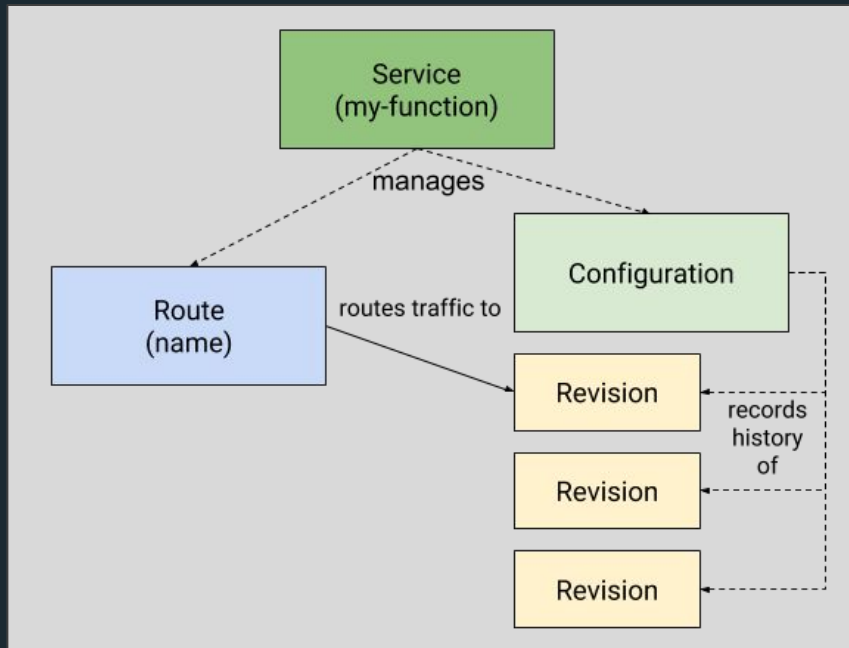I do not care how

Knative Serving

Knative is an Open-Source Enterprise-level solution to build Serverless and Event Driven Applications

Service - container accepting HTTP traffic

https://knative.dev

Why Functions?

I'm lazy, I don't want to write boilerplate code and YAML configuration!

I want to just write the code that deals with the problem at hand.

```java
package functions;
import java.util.function.Function;

public class Upper implements
Function<String, String> {
    public String apply(String in) {
        return in.toUpperCase();
    }
}
```

What are the building blocks?
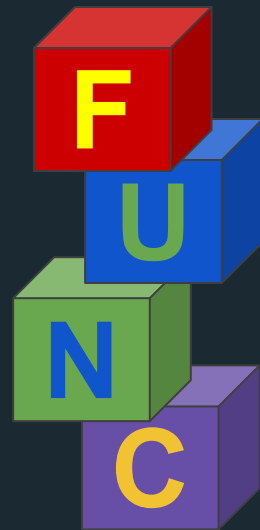
1. **Spring Cloud Function**

    supports a uniform programming model across serverless providers

2. **Knative func**

    provides a simple programming model for using functions on Knative

3. **Function Buildpacks for Knative**

    provides a FaaS experience for HTTP and CloudEvents leveraging buildpacks for Knative and Tanzu Application Platform

Spring Cloud Function

- Promote the implementation of business logic via functions.

- Decouple the development lifecycle of business logic from any specific runtime target so that the same code can run as a web endpoint, a stream processor, or a task.

- Support a uniform programming model across serverless providers, as well as the ability to run standalone (locally or in a PaaS).

- Enable Spring Boot features (auto-configuration, dependency injection, metrics) on serverless providers.

https://spring.io/projects/spring-cloud-function

Knative Func

- A simple programming model for using functions on Knative, without requiring in-depth knowledge of Knative, Kubernetes, containers, or dockerfiles

- Create, build, and deploy stateless, event-driven functions as Knative Services by using the func CLI

- Provides language packs that support HTTP and CloudEvents for Node.js, Python, Go, Quarkus, Rust, Spring Boot, and TypeScript

- Provides function templates for above languages/frameworks, and you can add your own templates

- Recently graduated from "sandbox" to top-level Knative project and is part of 1.8 release

https://knative.dev/docs/functions

Function Buildpacks for Knative

- ⚡ Build and deploy your functions fast -- a FaaS experience for HTTP and CloudEvents leveraging buildpacks

- Function Buildpacks for Knative (aka "Functions") brings functions as a programming model, to allow you to quickly build and deploy independent units of logic

- Provides buildpacks for Java/Spring Boot, Python and Node.js

- **Soon**, func support will allow users to deploy Functions via CLI in a matter of seconds

https://github.com/vmware-tanzu/function-buildpacks-for-knative

Live Demo

MacBook M1/M2 users - problems?

1. **No buildpack support for ARM yet**

   `kn func build` a bit flaky for JVM builds and no native image support on Apple ARM chips - **Support ARM #435**

2. **Func buildpacks produce X86_64 images**

   X86_64 images are slow on ARM clusters

3. **`Jib` can build multi-arch images**

   Jib images are JVM based

4. **Can use kpack for native image in-cluster builds**

More Advanced Sample: a function for looking up stock quotes

```java
@Bean
public Function<Message<Map<String, List<Object>>>, Map<String, List<Object>>> quote() {
  return (inputMessage) -> {
    Map<String, List<Object>> result = new HashMap<>();
    if (!inputMessage.getPayload().isEmpty()) {
      String symbol = inputMessage.getPayload().keySet().iterator().next();
      try {
        Stock stock = YahooFinance.get(symbol);
        if (stock != null) {
          var quote = stock.getQuote();
          if (quote == null || quote.getPrice() == null) {
            result.put(symbol.toUpperCase(), Collections.singletonList("No price for symbol found!"));
          } else {
            result.put(symbol.toUpperCase(), Collections.singletonList(quote.getPrice()));
          }
        } else {
          result.put(symbol.toUpperCase(), Collections.singletonList("Symbol not found!"));
        }
      } catch(IOException e) {
        result.put("ERROR", Collections.singletonList(e.toString()));
      }
    }
    return result;
  };
}
```

Adding custom function templates

- You can add custom function templates to your local **func** install
- Templates can use additional language packs or use the ones shipped with **func**

```
$ func repository add trisberg  https://github.com/trisberg/func-templates

$ func create --help
…
        Available Language Runtimes and Templates:
          Language      Template
          --------      --------
          go            cloudevents
          go            http
          node          cloudevents
          node          http
          python        cloudevents
          python        http
          quarkus       cloudevents
          quarkus       http
          rust          cloudevents
          rust          http
          springboot    cloudevents
          springboot    http
          springboot    trisberg/convert-case
          springboot    trisberg/hello-fun
          typescript    cloudevents
          typescript    http


        To install more language runtimes and their templates see 'func repository'.
```
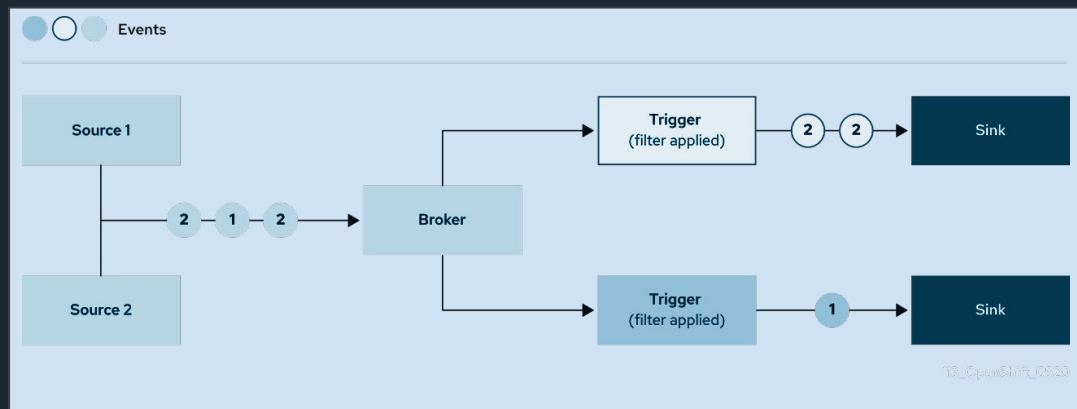
Knative Eventing and CloudEvents

- Knative Eventing is a collection of APIs that enable you to use an event-driven architecture with your applications

- Knative Eventing uses standard HTTP POST requests to send and receive events between event producers and sinks

- Events conform to the **CloudEvents** specifications, which enables creating, parsing, sending, and receiving events in any programming language

Live Demo

Thank you!

Resources:

- https://spring.io/projects/spring-cloud-function

- https://knative.dev/docs/functions

- https://github.com/vmware-tanzu/function-buildpacks-for-knative

Demos:

- https://github.com/trisberg/springone-2022

- https://github.com/trisberg/quote

- https://github.com/trisberg/func-templates

vmware® ©2022 VMware, Inc.