Table of Contents

Terminology

This section defines all the terms used in the `funktion` project

Connector

A `connector` represents a way to connect to some event source, including most network protocols, transports, databases, messaging systems, social networks, cloud services and SaaS offerings. Funktion supports over 200 event sources.

At the implementation level a `connector` represents the kubernetes `deployment` metadata required to take the `flow` and implement it as one or more kubernetes `pods`.

Flow

A `flow` is a sequence of `steps` such as consuming events from an `endpoint` or invoking a `function`.

Subscription

A subscription consists of one or more `flows` which bind events to HTTP endpoints and functions.

For example here is a sample subscription with a single flow:

```
flows:
  - steps:
    - kind: endpoint
      uri: timer://foo?fixedRate=true&period=5000
    - kind: endpoint
      uri: http://myendpoint/
```

Creating a `subscription` results in the `funktion operator` creating an associated `deployment` which implements the flows.

Function

A `function` is some source code to implement a function in some programming language like JavaScript, python or ruby.

Runtime

A `runtime` represents the kubernetes `deployment` metadata required to take a function source in some programming language and implement it as one or more pods.

The `funktion operator` then detects a new `function` resource being created or updated and creates the associated `runtime` deployment

Funktion Operator

The `funktion operator` is a runnning `pod` in kubernetes which monitors for all the funktion resources like `function`, `runtime`, `connector` and `subscription` and creates, updates or deletes the associated kubernetes `deployments` and `services` so that as you create a `subscription` or `function` the associated kubenretes resources are created.