

Table of Contents

- Terminology 1
 - Function 1
 - Runtime 1
 - Connector 1
 - Flow..... 1
 - Funktion Operator..... 2

Terminology

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

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

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`.

For example here is a sample flow in YAML format.

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

Note that a Flow resource can contain multiple sequential flows. Each flow object in the YAML is a sequence of steps.

Creating a `flow` results in the `funktion operator` creating an associated `Deployment` which implements the flows.

Funktion Operator

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