Getting Started

Table of Contents

Type the following commands.

To make it easier to see what kubernetes resources are being created you may wish to create a new namespace for this experiment first:

```
kubectl create namespace funky
kubectl config set-context `kubectl config current-context` --namespace=funky
```

Now we'll install the runtimes and a couple of connectors

```
funktion install http4 timer twitter
```

Now lets run the `funktion operator` to watch for funktion resources and create the necessary kubernetes `Deployment` and `Services`.

```
funktion operate
```

Open another terminal then type:

```
kubectl apply -f https://raw.githubusercontent.com/fabric8io/funktion-
operator/master/examples/subscription1.yml
```

You should now have created a subscription flow. You can view the subscription via

```
funktion get subscription
```

To view the output of the subscription you can use the following (assuming you've enabled tab completion for kubectl

```
kubectl logs -f subscription1-[TAB]
```

If you don't have tab completion you can specify the exact pod name, or you can use this command to find it and use it:

```
kubectl logs -f `kubectl get pod -oname -lfunktion.fabric8.io/kind=Subscription`
```

To delete the subscription:

```
funktion delete subscription subscription1
```

Now lets create a function:

```
kubectl apply -f https://raw.githubusercontent.com/fabric8io/funktion-
operator/master/examples/function1.yml
```

If you are running the fabric8 console then you will have the link:[exposecontroller] microservice running and will be able to invoke it via running one of these commands:

```
minikube service function1 -n funky
gofabric8 service function1 -n funky
```

Or clicking on the `funktion1` service in the fabric8 console in the `Services` tab for the `funky` namespace.

A more complex example

To see a more real world style example check out the blog splitting and counting example using functions and flows