Using the CLI

Table of Contents

You can get help on the available commands via:

```
funktion
```

Browsing resources

To list all the resources of different kind via:

```
funktion get connector
funktion get subscription
funktion get function
funktion get runtime
```

or to save typing you can use:

```
funktion get c
funktion get s
funktion get f
funktion get r
```

Deleting resources

You can delete a Connector or Subscription via:

```
funktion delete connector foo
funktion delete subscription bar
funktion delete function whatnot
funktion delete runtime nodejs
```

Or to remove all the Subscriptions or Connectors use `--all`

```
funktion delete subscription --all
```

Installing Runtimes and Connectors

To install the default function runtimes and connectors into your namespace type the following:

```
funktion install --all-connectors
```

There's over 200 connectors provided out of the box. If you only want to install a number of them you can specify their names as parameters

```
funktion install amqp kafka timer twitter
```

To just get a feel for what connectors are available without installing them try:

```
funktion install --list-connectors
```

or for short:

```
funktion install -l
```

Configuring Connectors

Various connectors have different configuration properties. For example the `twitter` connector has a number of properties to configure like the secret and token.

So to configure a connector you can type:

```
funktion edit connector twitter
```

You will then be prompted to enter new values; you can just hit `[ENTER]` to avoid changing a property.

To see a list of all the properties you can type

```
funktion edit connector twitter -l
```

Then you can pass in specific properties directly via the non-interactive version of the edit command:

```
funktion edit connector twitter accessToken=mytoken accessTokenSecret=mysecret
consumerKey=myconsumerkey consumerSecret=myconsumerSecert
```

Running the Operator

You can run the funktion operator from the command line if you prefer:

```
funktion operate
```

Though ideally we'd run the `funktion application` inside kubernetes; via a helm chart, `kubectl apply` or the `Run⋯` button in the fabric8 developer console

Subscribing to events

To create a new subscription for a connector try the following:

```
funktion subscribe --from timer://bar?period=5000 --to http://foo/
```

This will generate a new `Subscription` which will result in a new `Deployment` being created and one or more Pods should spin up.

Note that you must be running the `Operator` as described in the section above; its the `Operator` which actually creates a `Deployment` for each `Subscription`.

Also note that the first time you try out a new Connector kind it may take a few moments to download the docker image for this connector - particularly the first time you use a connector.

Once a pod has started for the `Deployment` you can then view the logs of a subscription via `kubectl`

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

Scaling a Subscription

If you want to stop a subscription type:

```
kubectl scale --replicas=0 deployment nameOfSubscription
```

To start it again:

```
kubectl scale --replicas=1 deployment nameOfSubscription
```

Using kubectl directly

You can also create a Subscription using `kubectl` if you prefer:

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

You can view all the Connectors and Subscriptions via:

```
kubectl get cm
```

Or delete them via

```
kubectl delete cm nameOfConnectorOrSubscription
```