Cloud Lab 6

Installing Knative Functions

My first step was to install func and the Knative plugins

```
git-graph                          statesmith
git-who                            terraform-iam-policy-validator
gnome-online-accounts              umockdev
grafana-alloy                      veccore
gtree                              yutu
gut                                zlib-rs
icu4c@77
==> New Casks
ace-studio                         font-special-gothic
advanced-renamer                   font-webdings
agent-tars                         font-winky-rough
charles@4                          hy-rpe2
chime@alpha                        irpf2025
chordpotion                        k6-studio
coterm                             kate
font-maple-mono-normal             mouseless@preview
font-maple-mono-normal-cn          nethlink
font-maple-mono-normal-nf          pastenow
font-maple-mono-normal-nf-cn       qobuz-downloader
font-playpen-sans-arabic           qt-design-studio
font-playpen-sans-hebrew           tal-drum
font-playpen-sans-thai             thelowtechguys-cling
font-pretendard-gov                triliumnext-notes


You have 28 outdated formulae installed.

==> Tapping knative-extensions/kn-plugins
Cloning into '/opt/homebrew/Library/Taps/knative-extensions/homebrew-kn-plugins'...
remote: Enumerating objects: 874, done.
remote: Counting objects: 100% (229/229), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 874 (delta 213), reused 198 (delta 197), pack-reused 645 (from 3)
Receiving objects: 100% (874/874), 239.93 KiB | 3.87 MiB/s, done.
Resolving deltas: 100% (598/598), done.
Tapped 41 formulae (149 files, 448.5KB).
aaron@MacBookAir Cloud-Data-Centres % brew install func
==> Fetching knative-extensions/kn-plugins/func
==> Downloading https://github.com/knative/func/releases/download/knative-v1.17.0/func_darwin_arm64
==> Downloading from https://objects.githubusercontent.com/github-production-release-asset-2e65be/242137258/ea7c11
################################################################################################### 100.0%
==> Installing func from knative-extensions/kn-plugins
🍺  /opt/homebrew/Cellar/func/knative-v1.17.0: 5 files, 78.6MB, built in 2 seconds
==> Running `brew cleanup func`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
aaron@MacBookAir Cloud-Data-Centres %
```

Creating A Function

I created a function through the CLI using this command

```
aaron@MacBookAir Cloud-Data-Centres % func create -l go hello
Created go function in /Users/aaron/Documents/GitHub/Cloud-Data-Centres/hello
aaron@MacBookAir Cloud-Data-Centres %
```

Here I am running it on my machine using my docker account.

```
aaron@MacBookAir hello % func run --registry docker.io/doyleaaron


function up-to-date. Force rebuild with --build
Running on host port 8080
Initializing HTTP function
listening on http port 8080

```

Here I am invoking the function and you can see the response hello world

```
aaron@MacBookAir hello % func invoke

"POST / HTTP/1.1\r\nHost: localhost:8080\r\nAccept-Encoding: gzip\r\nContent-Len
gth: 25\r\nContent-Type: application/json\r\nUser-Agent: Go-http-client/1.1\r\n\
r\n{\"message\":\"Hello World\"}"%
aaron@MacBookAir hello %
```

Then I built the function using my docker account

```
aaron@MacBookAir hello % func build
A registry for function images is required. For example, 'docker.io/tigerteam'.
? Registry for function images: docker.io/doyleaaron
Note: building a function the first time will take longer than subsequent builds
Building function image
Still building
🙌 Function built: index.docker.io/doyleaaron/hello:latest
aaron@MacBookAir hello %
```

Deploying a Knative Service:

Here I am creating the service and deploying it to port 8080

```
aaron@Aarons-MacBook-Air hello % kn service create hello \
--image ghcr.io/knative/helloworld-go:latest \
--port 8080 \
--env TARGET=World
Warning: Kubernetes default value is insecure, Knative may default this to secur
e in a future release: spec.template.spec.containers[0].securityContext.allowPri
vilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities,
 spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.sp
ec.containers[0].securityContext.seccompProfile
Creating service 'hello' in namespace 'default':

  0.029s The Route is still working to reflect the latest desired specification.
  0.049s Configuration "hello" is waiting for a Revision to become ready.
  0.071s ...
  2.053s ...
  2.068s Ingress has not yet been reconciled.
  2.095s Waiting for load balancer to be ready
  2.296s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
http://hello.default.192.168.49.2.sslip.io
aaron@Aarons-MacBook-Air hello % 
```

AutoScaling:

Here is the service responding

```
aaron@MacBookAir hello % curl -H "Host: hello.default.192.168.49.2.sslip.io" htt
p://localhost:8080

Hello World!
aaron@MacBookAir hello % 
```

Here we can see the pods scaling to 0 when they get no traffic

```
aaron@MacBookAir hello % kubectl get pod -l serving.knative.dev/service=hello -w

NAME                                        READY   STATUS        RESTARTS   AGE
hello-00001-deployment-7d9c65cc94-g6kzv     2/2     Running       0          37s
hello-00001-deployment-7d9c65cc94-g6kzv     2/2     Terminating   0          61s
hello-00001-deployment-7d9c65cc94-g6kzv     1/2     Terminating   0          91s
hello-00001-deployment-7d9c65cc94-g6kzv     0/2     Completed     0          91s
```

Here we can see the changed service running

```
aaron@MacBookAir hello % curl -H "Host: hello.default.192.168.49.2.sslip.io" htt
p://localhost:8080

Hello Knative!
aaron@MacBookAir hello % 
```

Here are the existing revisions:

```
aaron@MacBookAir hello % kn revisions list
NAME                     SERVICE        TRAFFIC    TAGS    GENERATION    AGE    CONDIT
IONS    READY    REASON
hello-00002              hello          100%               2             15m    3 OK /
  4      True
hello-00001              hello                              1             27d    3 OK /
  4      True
helloworld-go-00001      helloworld-go  100%               1             34d    3 OK /
  4      True
```

Here we can see the traffic split between the two revisions

```
aaron@MacBookAir hello % kn service update hello \
--traffic hello-00001=50 \
--traffic @latest=50
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.
containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spe
c.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfil
e
Updating Service 'hello' in namespace 'default':

  0.011s The Route is still working to reflect the latest desired specification.
  0.031s Ingress has not yet been reconciled.
  0.041s Waiting for load balancer to be ready
  0.247s Ready to serve.

Service 'hello' with latest revision 'hello-00002' (unchanged) is available at URL:
http://hello.default.192.168.49.2.sslip.io
aaron@MacBookAir hello % kn revisions list
NAME                     SERVICE        TRAFFIC    TAGS    GENERATION    AGE    CONDITIONS    READY    REASON
hello-00002              hello          50%                2             17m    3 OK / 4      True
hello-00001              hello          50%                1             27d    3 OK / 4      True
helloworld-go-00001      helloworld-go  100%               1             34d    3 OK / 4      True
```
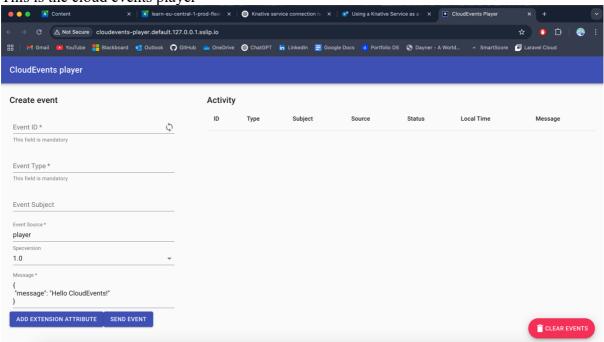
Using Knative service as a source:

Here I am creating the service

```
aaron@MacBookAir hello % kn service create cloudevents-player \
--image quay.io/ruben/cloudevents-player:latest
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.
containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spe
c.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfil
e
Creating service 'cloudevents-player' in namespace 'default':

  0.037s The Route is still working to reflect the latest desired specification.
  0.059s ...
  0.065s Configuration "cloudevents-player" is waiting for a Revision to become ready.
 21.484s ...
 21.506s Ingress has not yet been reconciled.
 21.540s Waiting for load balancer to be ready
 21.726s Ready to serve.

Service 'cloudevents-player' created to latest revision 'cloudevents-player-00001' is available at URL:
http://cloudevents-player.default.127.0.0.1.sslip.io
aaron@MacBookAir hello %
```
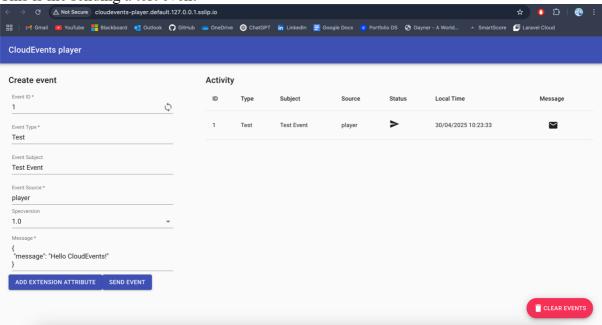
Then I created a sinkBinding

```
aaron@MacBookAir hello % kn source binding create ce-player-binding --subject "Service:serving.knative.dev/v1:cloudevents-
player" --sink broker:example-broker
Sink binding 'ce-player-binding' created in namespace 'default'.
aaron@MacBookAir hello %
```

This is the cloud events player

This is me sending a test event

Here I am creating a trigger

```
aaron@MacBookAir hello % kn trigger create cloudevents-trigger --sink cloudevents-player  --broker example-broker
Trigger 'cloudevents-trigger' successfully created in namespace 'default'.
aaron@MacBookAir hello %
```

Here we can see the trigger being created and seen on the cloud events player.