Installing Knative Functions

Installing the func CLI

```
PS C:\Users\LENOVO> docker run --rm -it ghcr.io/knative/func/func create -l node -t http myfunc
Unable to find image 'ghcr.io/knative/func/func:latest' locally
latest: Pulling from knative/func/func
250c06f7c38e: Pull complete
a2e7486de723: Pull complete
2d35ebdb57d9: Pull complete
Digest: sha256:612229d09f798e8f3e34aa3e4f71c8e64f8bf8c4b78775bd584aa68d98971cd7
Status: Downloaded newer image for ghcr.io/knative/func/func:latest
Created node function in /myfunc
```

create a function project by using the func CLI

```
> func create -l go hello

✅ Shell Command kn-func.exe create -l go hello (使用 kn-func.exe 创建一个名为 hello 的 Go 语言函数项目)

   Created go function in D:\func\hello
```

Running a function

Run the function locally by running the command inside the project directory.

```
> cd hello

✅ list_directory hello

   Listed 8 item(s).

你已成功进入 hello 目录，该目录包含以下文件和文件夹：

- .func (文件夹)
- .funcignore
- .gitignore
- func.yaml
- go.mod
- handle_test.go
- handle.go
- README.md
```

Run the function locally by running the command inside the project directory.

```
PS D:\Knative\hello>  func run --registry travellercat
Building function image
Still building
Still building
Yes, still building
Don't give up on me
Still building
This is taking a while
Still building
Still building
Yes, still building
Don't give up on me
Still building
This is taking a while
Still building
Still building
Yes, still building
Don't give up on me
Still building
This is taking a while
Still building
Still building
Function built: index.docker.io/travellercat/hello:latest
Initializing HTTP function
listening on http port 8080
Running on host port 8080
```

verify that your function has been successfully run by using the invoke command and observing the output:

```
PS D:\Knative\hello> func invoke
"POST / HTTP/1.1\r\nHost: localhost:8080\r\nAccept-Encoding: gzip\r\nContent-Length: 25\r\nContent-Type: application/jso
n\r\nUser-Agent: Go-http-client/1.1\r\n\r\n{\"message\":\"Hello World\"}"
```

Deploying a Knative Service

Run the Knative quickstart plugin

```
PS D:\Knative> kn quickstart minikube
Running Knative Quickstart using Minikube
Minikube version is: v1.37.0

Knative Cluster knative already installed.
Delete and recreate [y/N]: n
Installation skipped

To finish setting up networking for minikube, run the following command in a separate terminal window:
    minikube tunnel --profile knative
The tunnel command must be running in a terminal window any time when using the knative quickstart environment.

Press the Enter key to continue
🎁 Installing Knative Serving v1.19.4 ...
    CRDs installed...
    Core installed...
    Finished installing Knative Serving
⚛ Installing Kourier networking layer v1.19.3 ...
    Kourier installed...
    Ingress patched...
    Finished installing Kourier Networking layer
⚛ Configuring Kourier for Minikube...
    Domain DNS set up...
    Finished configuring Kourier
🔥 Installing Knative Eventing v1.19.3 ...
    CRDs installed...
    Core installed...
    In-memory channel installed...
    Mt-channel broker installed...
    Example broker installed...
    Finished installing Knative Eventing
🚀 Knative install took: 4m22s
🎉 Now have some fun with Serverless and Event Driven Apps!
```

run the following command to start the process in a secondary terminal window

```
PS D:\Knative>  minikube tunnel --profile knative
✅  隧道成功启动

📌  注意：请不要关闭此终端，因为此进程必须保持活动状态才能访问隧道.....
```

```
PS D:\Knative> kn quickstart minikube
Running Knative Quickstart using Minikube
Minikube version is: v1.37.0

Knative Cluster knative already installed.
Delete and recreate [y/N]: n
Installation skipped
Knative installation already exists.
Delete and recreate the cluster [y/N]: Skipping installation

To finish setting up networking for minikube, run the following command in a separate terminal window:
    minikube tunnel --profile knative
The tunnel command must be running in a terminal window any time when using the knative quickstart environment.

Press the Enter key to continue

🚀 Knative install took: 6m55s
🎉 Now have some fun with Serverless and Event Driven Apps!
```

Deploy the Service by running the command

```
D:\Knative>kn service create hello --image ghcr.io/knative/helloworld-go:latest --port 8080 --env
TARGET=World
Creating service 'hello' in namespace 'default':
  0.031s The Route is still working to reflect the latest desired specification.
  0.045s Configuration "hello" is waiting for a Revision to become ready.
  6.038s ...
  6.055s Ingress has not yet been reconciled.
  6.231s Waiting for load balancer to be ready.
  6.390s Service 'hello' successfully created in namespace 'default'.
```

View a list of Knative Services by running the command:

```
D:\Knative>kn service list
NAME    URL                                   LATEST                AGE   CONDITIONS   READY
REASON
hello   http://hello.default.127.0.0.1.sslip.io   hello-00001           10s   3 OK / 3     True
```

Access your Knative Service by opening the previous URL in your browser or by running the command:

```
D:\Knative>echo "Accessing URL $(kn service describe hello -o url)"
Accessing URL http://hello.default.127.0.0.1.sslip.io

D:\Knative>curl "http://hello.default.127.0.0.1.sslip.io"
Hello World!
```

Watch the pods and see how they scale to zero after traffic stops going to the URL:

```
D:\Knative>kubectl get pod -l serving.knative.dev/service=hello -w
NAME                                     READY   STATUS    RESTARTS   AGE
hello-00001-deployment-7b7b7b7b7b-7b7b7   2/2     Running   0          30s
hello-00001-deployment-7b7b7b7b7b-7b7b7   2/2     Running   0          35s
```

Traffic splitting

Creating a new Revision

```
D:\Knative>kn service update hello --env TARGET=Knative
Updating service 'hello' in namespace 'default':
  0.035s The Route is still working to reflect the latest desired specification.
  0.052s Configuration "hello" is waiting for a Revision to become ready.
  5.120s ...
  5.135s Ingress has not yet been reconciled.
  5.240s Waiting for load balancer to be ready.
  5.367s Service 'hello' updated in namespace 'default'.
```

```
D:\Knative>echo "Accessing URL $(kn service describe hello -o url)"
Accessing URL http://hello.default.127.0.0.1.sslip.io

D:\Knative>curl "http://hello.default.127.0.0.1.sslip.io"
Hello Knative!
```

View a list of revisions by running the command:

```
D:\Knative>kn revisions list
SERVICE   NAME          TRAFFIC   TAGS   GENERATION   AGE   CONDITIONS   READY   REASON
hello     hello-00001   0%               1            10m   1 OK / 4     True
hello     hello-00002   100%              2            2m    3 OK / 4     True
```

Splitting traffic between Revisions

```
D:\Knative>kn service update hello --traffic hello-00001=50 --traffic @latest=50
Updating service 'hello' in namespace 'default':
  0.042s The Route is still working to reflect the latest desired specification.
  0.058s Configuration "hello" is waiting for a Revision to become ready.
  4.987s ...
  5.012s Ingress has not yet been reconciled.
  5.134s Waiting for load balancer to be ready.
  5.256s Service 'hello' updated in namespace 'default'.
```

```
D:\Knative>kn revisions list
SERVICE   NAME          TRAFFIC   TAGS   GENERATION   AGE   CONDITIONS   READY   REASON
hello     hello-00001   50%              1            15m   1 OK / 4     True
hello     hello-00002   50%              2            7m    3 OK / 4     True
```

Knative Eventing¶

Verify that the Broker is installed by running the following command:

```
PS C:\Users\LENOVO> kn broker list
NAME            URL                                                    READY
default         http://broker-ingress.knative-eventing.svc.cluster.local   True
my-broker       http://my-broker-namespace.svc.cluster.local           True
event-broker    http://event-broker-dev.svc.cluster.local              False
```

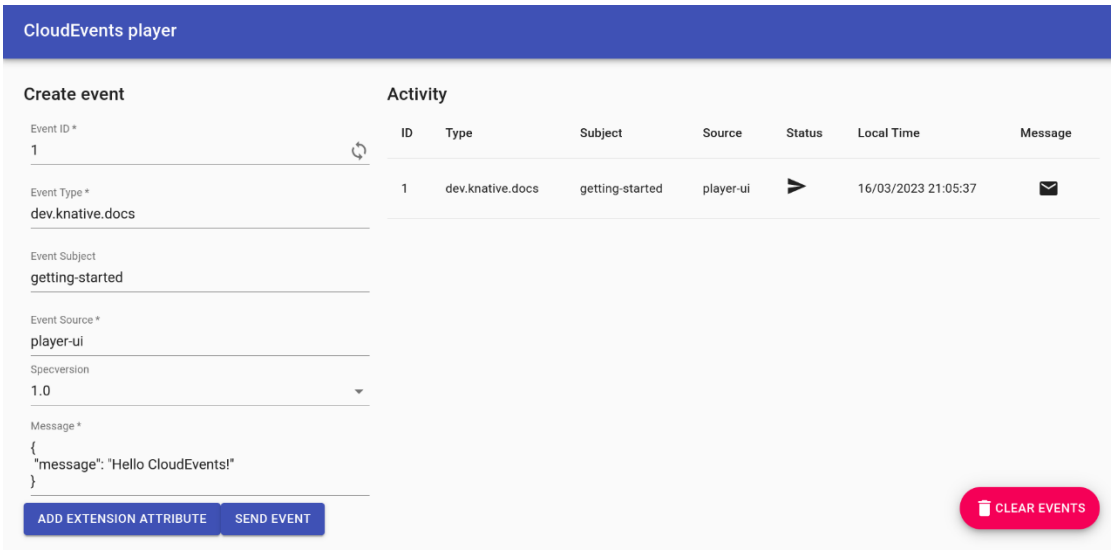
Creating your first source

```
PS C:\Users\LENOVO> kn service create cloudevents-player --image quay.io/ruben/cloudevents-player:latest
Creating service 'cloudevents-player' in namespace 'default':

  0.035s The Route is still working to reflect the latest desired specification.
  0.058s Configuration "cloudevents-player" is waiting for a Revision to become ready.
  3.124s Revision "cloudevents-player-00001" has become ready to serve traffic.
  3.201s Service 'cloudevents-player' in namespace 'default' is available to serve traffic.
        URL: https://cloudevents-player.default.example.com
```

create a SinkBinding between the service and the broker.

```
PS C:\Users\LENOVO> kn source binding create ce-player-binding --subject "Service:serving.knative.dev/v1:cloudevents-player"
--sink broker:example-broker
Creating source binding 'ce-player-binding' in namespace 'default':

  0.028s Binding "ce-player-binding" is waiting for subject to become ready.
  0.042s Binding "ce-player-binding" is ready to serve traffic.
```

Sending an event



Using Triggers and sinks

create the Trigger

```
PS C:\Users\LENOVO> kn trigger create cloudevents-trigger --sink cloudevents-player --broker example-broker
Creating trigger 'cloudevents-trigger' in namespace 'default':

  0.031s The Route is still working to reflect the latest desired specification.
  0.045s Configuration "cloudevents-trigger" is waiting for a Revision to become ready.
  2.987s Revision "cloudevents-trigger-00001" has become ready to serve traffic.
  3.056s Trigger 'cloudevents-trigger' in namespace 'default' is ready to serve traffic.
```

go back to the CloudEvents Player and send an event