Lab6 Notes/Findings

Step1: Install kn:

```
kacper@UNKNOWN:~$ curl -Lo kn https://github.com/knative/client/releases/latest/download/kn-linux-amd64 && chmod +x kn && sudo mv kn /usr/local/bin
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 84.7M  100 84.7M    0     0  23.9M      0  0:00:03  0:00:03 --:--:-- 33.5M
[sudo] password for kacper:
kacper@UNKNOWN:~$ kn version
Version:      v1.18.0
Build Date:   2025-04-23 19:42:52
Git Revision: 96721e59
Supported APIs:
* Serving
  - serving.knative.dev/v1 (knative-serving v1.18.0)
* Eventing
  - sources.knative.dev/v1 (knative-eventing v1.18.0)
  - eventing.knative.dev/v1 (knative-eventing v1.18.0)
```

```
kacper@UNKNOWN:~$ kn quickstart kind
Running Knative Quickstart using Kind
✅ Checking dependencies...
    Kind version is: 0.28.0

A local registry is no longer created by default.
    To create a local registry, use the --registry flag.

☀Creating Kind cluster...
Creating cluster "knative" ...
 ✓ Ensuring node image (kindest/node:v1.31.6) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
 ✓ Waiting ≤ 2m0s for control-plane = Ready ⏳
 • Ready after 17s 💚
Set kubectl context to "kind-knative"
You can now use your cluster with:

kubectl cluster-info --context kind-knative

Not sure what to do next? 😄  Check out https://kind.sigs.k8s.io/docs/user/quick-start/

🍿 Installing Knative Serving v1.18.0 ...
    CRDs installed...
    Core installed...
    Finished installing Knative Serving
🕸Installing Kourier networking layer v1.18.0 ...
    Kourier installed...
    Ingress patched...
    Finished installing Kourier Networking layer
🕸Configuring Kourier for Kind...
    Kourier service installed...
    Domain DNS set up...
    Finished configuring Kourier
🔥 Installing Knative Eventing v1.18.0 ...
    CRDs installed...
    Core installed...
    In-memory channel installed...
    Mt-channel broker installed...
    Example broker installed...
    Finished installing Knative Eventing
🎉 Knative install took: 3m25s
🎉 Now have some fun with Serverless and Event Driven Apps!
```

Step2:Build into the local registry:

```
kacper@UNKNOWN:~$ curl -Lo func https://github.com/knative/func/releases/latest/download/func_linux_amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 82.9M  100 82.9M    0     0  22.7M      0  0:00:03  0:00:03 --:--:-- 30.3M
kacper@UNKNOWN:~$ chmod +x func
kacper@UNKNOWN:~$ sudo mv func /usr/local/bin/
[sudo] password for kacper:
kacper@UNKNOWN:~$ func version
v0.45.1
kacper@UNKNOWN:~$ mkdir ~/hello-func && cd ~/hello-func
kacper@UNKNOWN:~/hello-func$ func init --language go
Created go function in /home/kacper/hello-func
kacper@UNKNOWN:~/hello-func$ export FUNC_REGISTRY=localhost:5001/kn-func
kacper@UNKNOWN:~/hello-func$ func build
Building function image
Still building
Still building
Yes, still building
🙌 Function built: localhost:5001/kn-func/hello-func:latest
```

Step3: Create a Knative Service:

```
kacper@UNKNOWN:~/hello-func$ kn service create hello \
  --image ghcr.io/knative/helloworld-go:latest \
  --port 8080 \
  --env TARGET="Knative"

URL=$(kn service describe hello -o url)
curl $URL
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].s
ecurityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'hello' in namespace 'default':

  0.044s The Route is still working to reflect the latest desired specification.
  0.049s ...
  0.075s Configuration "hello" is waiting for a Revision to become ready.
  7.764s ...
  7.782s Ingress has not yet been reconciled.
  7.818s Waiting for load balancer to be ready
  8.015s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
http://hello.default.127.0.0.1.sslip.io
Hello Knative!
```

Step4: Autoscale Demo:

```
kacper@UNKNOWN:~/hello-func$ for i in {1..100}; do curl -s $URL & done; wait
watch kubectl get pod -l serving.knative.dev/service=hello
```

```
kacper@fedora:~/hello-func — watch kubectl get pod -l serving.knative.dev/service=hello
                                              ~/hello-func
Every 2.0s: kubectl get pod -l serving.knative.dev/service=hello                    UNKNOWN: Sun May 18 17:25:42 2025

NAME                                     READY   STATUS        RESTARTS   AGE
hello-00001-deployment-5ffbb95579-5dhvw  2/2     Running       0          11s
hello-00001-deployment-5ffbb95579-fwwdw  2/2     Running       0          12s
hello-00001-deployment-5ffbb95579-rl67b  1/2     Terminating   0          93s
```

Step5: Knative Event:

```
kacper@UNKNOWN:~/hello-func$ kubectl -n knative-eventing port-forward svc/broker-ingress 8080:80 &    #
sleep 2
curl -v http://localhost:8080/default/default \
  -H 'Ce-Specversion: 1.0' \
  -H 'Ce-Type: demo.hello' \
  -H 'Ce-Source: cli' \
  -H 'Ce-Id: 1' \
  -H 'Content-Type: application/json' \
  -d '{"msg":"Hi Eventing"}'
[3] 156582
Unable to listen on port 8080: Listeners failed to create with the following errors: [unable to creat
ten tcp6 [::1]:8080: bind: address already in use]
error: unable to listen on any of the requested ports: [{8080 8080}]
[3]+  Exit 1                  kubectl -n knative-eventing port-forward svc/broker-ingress 8080:80
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8080...
* Connected to localhost (::1) port 8080
* using HTTP/1.x
> POST /default/default HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.11.1
> Accept: */*
> Ce-Specversion: 1.0
> Ce-Type: demo.hello
> Ce-Source: cli
> Handling connection for 8080
Ce-Id: 1
> Content-Type: application/json
> Content-Length: 21
>
* upload completely sent off: 21 bytes
< HTTP/1.1 202 Accepted
< Allow: POST, OPTIONS
< Date: Sun, 18 May 2025 16:49:04 GMT
< Content-Length: 0
<
* Connection #0 to host localhost left intact
```

```
kacper@UNKNOWN:~/hello-func$ kubectl logs -l serving.knative.dev/service=event-display --tail=20
2025/05/18 16:49:04 Failed to read tracing config, using the no-op default: empty json tracing config
☁️  cloudevents.Event
Context Attributes,
  specversion: 1.0
  type: demo.hello
  source: cli
  id: 1
  datacontenttype: application/json
Extensions,
  knativearrivaltime: 2025-05-18T16:49:03.258173321Z
Data,
  {
    "msg": "Hi Eventing"
  }
kacper@UNKNOWN:~/hello-func$
```