Lab 9

Environment:

Windows + Docker Desktop (Kubernetes enabled)

Local container registry running

kubectl and curl available

Part 1. Knative Functions

Objective

Enable a simple development workflow to create and build stateless event-driven functions without understanding containers or Kubernetes internals.

Steps

1. Initialize a Knative function project using CLI or templates.
2. Configure function build to push images into a local registry instead of remote cloud registry.
3. Build the function code into an OCI container image stored in the local registry.
4. (Skip deploy step as requested in instructions.)

Notes

- Knative Functions automatically generates container images when building functions.
- No Dockerfile is required.
- Developers interact using "func" or "kn func" commands.
- Image builds update automatically after code changes.
- Images are stored in a registry and can later be deployed as Knative services.

Part 2. Knative Serving

Objective

Deploy a serverless application using Knative Serving and observe activation and autoscaling behavior.

Steps

1. Copy and save the YAML below into a file named hello.yaml:

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: hello
spec:
  template:
    spec:
      containers:
        - image: ghcr.io/knative/helloworld-go:latest
          ports:
            - containerPort: 8080
          env:
            - name: TARGET
              value: "World"
```

2. Deploy the service using:

```
kubectl apply -f hello.yaml
```

3. Confirm Knative Service creation using:

kubectl get ksvc hello

4. Retrieve the external URL via:

kubectl get ksvc

5. Send a request to the service:

curl http://<service-url>

Expected behavior:

- Knative activates pods on request.
- Returns “Hello World!”

Part 3. Knative Eventing

Objective

Enable event-driven execution by wiring event producers and consumers through Knative Eventing components.

Steps

1. Ensure Knative Eventing is installed.
2. Create a Broker to receive events:

kubectl apply -f broker.yaml

3. Create a Trigger to select events and forward them to the Knative Service:

kubectl apply -f trigger.yaml

4. Send a CloudEvent manually into Broker:

```
kubectl -n default run -i --tty event-sender \
  --image=ghcr.io/knative/eventing-sender \
  -- -target http://broker.default.svc.cluster.local
```

5. Confirm event delivery by checking logs:

kubectl logs <pod-name>

Notes

- Event-driven processing decouples event producers and consumers.
- Knative components involved: Broker, Trigger, Sink, Event Source.
- Events follow CloudEvents specification.

```
C:\Users\lenovo>docker run -d -p 5000:5000 --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
8e82f80af0de: Pull complete
3493bf46cdec: Pull complete
6d464ea18732: Pull complete
44cf07d57ee4: Pull complete
bbbdd6c6894b: Pull complete
Digest: sha256:a3d8aaa63ed8681a604f1dea0aa03f100d5895b6a58ace528858a7b332415373
Status: Downloaded newer image for registry:2
b54f1db20f952b7f85561e45d2288eed3e5aa3e6703eff32d21efeb9a39ebb06

C:\Users\lenovo>docker ps
CONTAINER ID   IMAGE        COMMAND                  CREATED          STATUS             PORTS
               NAMES
b54f1db20f95   registry:2   "/entrypoint.sh /etc…"   20 seconds ago   Up 23 seconds      0.0.0.0:5000->5000/tcp, [::]:50
00->5000/tcp   registry
d510f710d9d0   n8nio/n8n    "tini -- /docker-ent…"   2 months ago     Up About an hour   0.0.0.0:5678->5678/tcp, [::]:56
78->5678/tcp   assessment-n8n-1
23576ddce95b   mysql:8      "docker-entrypoint.s…"   3 months ago     Up About an hour   3306/tcp, 33060/tcp
               gitea-db-1
```

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: hello
  namespace: default
spec:
  template:
    spec:
      containers:
        - image: gcr.io/knative-samples/helloworld-go
          env:
            - name: TARGET
              value: "Knative user"
```

```
NAME                   STATUS   AGE
default                Active   31d
istio-system           Active   30d
knative-eventing       Active   4d2h
knative-serving        Active   4d2h
kube-node-lease        Active   31d
kube-public            Active   31d
kube-system            Active   31d
kubernetes-dashboard   Active   31d
```

```
C:\Users\lenovo\Desktop>kubectl apply -f https://github.com/knative/serving/releases/download/knative-
re.yaml
namespace/knative-serving created
role.rbac.authorization.k8s.io/knative-serving-activator created
clusterrole.rbac.authorization.k8s.io/knative-serving-activator-cluster created
clusterrole.rbac.authorization.k8s.io/knative-serving-aggregated-addressable-resolver created
clusterrole.rbac.authorization.k8s.io/knative-serving-addressable-resolver created
clusterrole.rbac.authorization.k8s.io/knative-serving-namespace-admin created
clusterrole.rbac.authorization.k8s.io/knative-serving-namespace-edit created
clusterrole.rbac.authorization.k8s.io/knative-serving-namespace-view created
clusterrole.rbac.authorization.k8s.io/knative-serving-core created
clusterrole.rbac.authorization.k8s.io/knative-serving-podspecable-binding created
serviceaccount/controller created
clusterrole.rbac.authorization.k8s.io/knative-serving-admin created
clusterrolebinding.rbac.authorization.k8s.io/knative-serving-controller-admin created
clusterrolebinding.rbac.authorization.k8s.io/knative-serving-controller-addressable-resolver created
serviceaccount/activator created
rolebinding.rbac.authorization.k8s.io/knative-serving-activator created
clusterrolebinding.rbac.authorization.k8s.io/knative-serving-activator-cluster created
customresourcedefinition.apiextensions.k8s.io/images.caching.internal.knative.dev unchanged
customresourcedefinition.apiextensions.k8s.io/certificates.networking.internal.knative.dev unchanged
customresourcedefinition.apiextensions.k8s.io/configurations.serving.knative.dev unchanged
customresourcedefinition.apiextensions.k8s.io/clusterdomainclaims.networking.internal.knative.dev unch
customresourcedefinition.apiextensions.k8s.io/domainmappings.serving.knative.dev unchanged
customresourcedefinition.apiextensions.k8s.io/ingresses.networking.internal.knative.dev unchanged
customresourcedefinition.apiextensions.k8s.io/metrics.autoscaling.internal.knative.dev unchanged
customresourcedefinition.apiextensions.k8s.io/podautoscalers.autoscaling.internal.knative.dev unchange
```

```
NAME                          READY   STATUS              RESTARTS   AGE
activator-f56b94b44-ldjvr     0/1     ContainerCreating   0          4d2h
autoscaler-74d66ffcd-6n4gc    0/1     ContainerCreating   0          4d2h
controller-5d68d6d797-lwgk9   0/1     ContainerCreating   0          4d2h
webhook-c47fc76d8-sfrff       0/1     ContainerCreating   0          4d2h
```