Experiment 10: Operator SDK & Helm

The three tools you need for this experiment are:

1. SCM or DVCS

One of the following installed – Git, Mercurial, or Bazaar

2. make – should already be on macOS, but if you’re on Windows you’ll need GNU Make or similar

3. Go – should already be installed based on our first lab

Create a cluster, for the experiment

```
$ k3d cluster create local --api-port 6550 --agents
```

```
$> brew install operator-sdk

$> mkdir ~/k3d/nginx-operator

$> cd ~/k3d/nginx-operator

$> operator-sdk init --plugins=helm

$> operator-sdk create api --group=demo --version=v1 --kind=Nginx
```

Now we need to build the operator image

In macOS:

```
make docker-build docker-push IMG=<your-registry>/<image-name>:<tag>
```

To continue we need Kustomize or we could build out the kustomization via kubectl as we did in a previous lab.

Installation for Kustomize is available here:

For macOs:

We’ll use Homebrew

```
~/operator-sdk $ brew install kustomize
```

For both macOS and Windows:

Switch to our operator project build

```
$> cd ~/operator-sdk/nginx-operator
```

Run Kustomize against the nginx-operator configuration to create our Custom Resource Definition for the operator

```
~/operator-sdk/nginx-operator $> kustomize build config/crd > CRD-nginx-operator.yaml
```

Due to a current defect in the operator we'll have to do a bit of magic. Here we try to run our operator against the cluster.

```
~/operator-sdk/nginx-operator $> kustomize build config/default | kubectl apply -f -
namespace/operator-sdk-system created
customresourcedefinition.apiextensions.k8s.io/nginxalphas.demo.my.domain created
customresourcedefinition.apiextensions.k8s.io/nginxes.demo.my.domain created
serviceaccount/operator-sdk-controller-manager created
role.rbac.authorization.k8s.io/operator-sdk-leader-election-role created
clusterrole.rbac.authorization.k8s.io/operator-sdk-manager-role created
clusterrole.rbac.authorization.k8s.io/operator-sdk-metrics-reader created
clusterrole.rbac.authorization.k8s.io/operator-sdk-proxy-role created
rolebinding.rbac.authorization.k8s.io/operator-sdk-leader-election-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/operator-sdk-manager-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/operator-sdk-proxy-rolebinding created
service/operator-sdk-controller-manager-metrics-service created
deployment.apps/operator-sdk-controller-manager created
```

Create our sample Customer Resource against our cluster

```
~/operator-sdk/nginx-operator $> kubectl apply -f config/samples/demo_v1_nginx.yaml

nginx.demo.my.domain/nginx-sample created
```

To clean this up and get rid of our customer helm operator we

Remove our sample custom resource

```
~/operator-sdk/nginx-operator $>  kubectl delete -f config/samples/demo_v1_nginx.yaml
```

Remove our operator

```
~/operator-sdk/nginx-operator $> kustomize build config/default | kubectl delete -f –

namespace "system" deleted
customresourcedefinition.apiextensions.k8s.io "nginxes.demo.my.domain" deleted
role.rbac.authorization.k8s.io "nginx-operator-leader-election-role" deleted
clusterrole.rbac.authorization.k8s.io "nginx-operator-manager-role" deleted
clusterrole.rbac.authorization.k8s.io "nginx-operator-proxy-role" deleted
clusterrole.rbac.authorization.k8s.io "nginx-operator-metrics-reader" deleted
rolebinding.rbac.authorization.k8s.io "nginx-operator-leader-election-rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "nginx-operator-manager-rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "nginx-operator-proxy-rolebinding" deleted
service "nginx-operator-controller-manager-metrics-service" deleted
deployment.apps "nginx-operator-controller-manager" deleted
```

Delete the namespace that we had to create manually

```
~/operator-sdk/nginx-operator $> kubectl delete namespace nginx-operator-system

namespace/nginx-operator-system created
```

Change directory to our k3d folder

```
~/operator-sdk/nginx-operator $> cd \k3d
```

Disintegrate our k3d cluster

```
C:\k3d> k3d cluster delete local
[36mINFO[0m[0000] Deleting cluster 'local'
[36mINFO[0m[0000] Deleted k3d-local-serverlb
[36mINFO[0m[0001] Deleted k3d-local-server-0
[36mINFO[0m[0001] Deleting cluster network
'bd7bd4bd8ec595f0bbcc402f5f1090db29db7d27428ed2fa5877bc97a2189367'
[36mINFO[0m[0001] Deleting image volume 'k3d-local-images'
[36mINFO[0m[0001] Removing cluster details from default kubeconfig...
[36mINFO[0m[0001] Removing standalone kubeconfig file (if there is one)...
[36mINFO[0m[0001] Successfully deleted cluster local!
```