Experiment 09: Operator SDK & Helm

OperatorSDK – Helm

Prereqs for building an operator with the Operator SDK

https://sdk.operatorframework.io/build/

https://sdk.operatorframework.io/docs/building-operators/helm/quickstart/

https://sdk.operatorframework.io/docs/installation/install-operator-sdk/#compile-and-install-from-master

The three things you need 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

GNU Make for Windows

https://sourceforge.net/projects/gnuwin32/files/make/3.81/make-3.81.exe/download?use_mirror=netactuate&download=

{ If you were not using Git, you could download or use the following SCM tools

http://wiki.bazaar.canonical.com/Download

http://wiki.bazaar.canonical.com/WindowsDownloads

https://www.mercurial-scm.org/downloads

}

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

This is a reference around the Operator SDK

https://sdk.operatorframework.io/docs/building-operators/helm/quickstart/

Create a cluster, for the experiment

$ **k3d cluster create local --api-port 6550 --agents 2**

Set your KUBECONFIG and load it from k3d kubeconfig get as done in previous labs.

Note: Ensure that your GOPROXY is set with its default value for Go versions 1.13+ which is https://proxy.golang.org,direct.

```
$ C:\projects\kind\operator-sdk> set | grep GOPROXY
```

GOPROXY=https://proxy.golang.org,direct

C:\projects\kind> **git clone https://github.com/operator-framework/operator-sdk**
C:\projects\kind> **cd operator-sdk**
C:\projects\kind\operator-sdk> **set GOPROXY=https://proxy.golang.org,direct**

C:\projects\kind\operator-sdk> **C:\gnu-make\bin\make tidy**
or
~/operator-sdk $ make tidy

Watch as it downloads about a bajillion modules, and of course take note that there are gomodules from lots of individuals, Ross Ross, Peter Bourgon, Jeff Czapiewski, Maya Madeline, and Zachary George William
C:\projects\kind\operator-sdk> **C:\gnu-make\bin\make install**
or
~/operator-sdk $ make install

C:\projects\kind\operator-sdk> **mkdir nginx-operator**
C:\projects\kind\operator-sdk> **cd nginx-operator**
C:\projects\kind\operator-sdk> **operator-sdk init --plugins=helm**

C:\projects\kind\operator-sdk> 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=<some-registry>/<project-name>:<tag>

Unfortunately, the Makefile for the operator-sdk created operators doesn't know diddly about Windows, so we have to do a bunch of extra work. You'll find that in just about everything in the past, but not surprisingly that's been changing quite a bit with the arrival of AKS, Microsoft GitHub and RedHat change in ownership

In Windows:
Set the IMG target as the following
c:\projects\kind\operator-sdk\nginx-operator>set IMG=georgeniece/nginx-operator:v1.0.0

Validate the IMG export created correctly
c:\projects\kind\operator-sdk\nginx-operator>set | grep IMG
IMG=georgeniece/nginx-operator:v1.0.0

georgeniece is my Docker Hub Account User, so you would use yours, that we created in the earlier labs or if you had an already existing, or you can create one now at hub.docker.com

Note: Alternatively we could have started up the local registry and been using our localhost reference that we saw in the Air-Gapped-Kind experiment/lab

Build the docker image for our operator
c:\projects\kind\operator-sdk\nginx-operator>docker build . -t %IMG%
Sending build context to Docker daemon 67.58kB
Step 1/5 : FROM quay.io/operator-framework/helm-operator:v1.0.0

```
v1.0.0: Pulling from operator-framework/helm-operator
41ae95b593e0: Pull complete                                    f20f68829d13:
Pull complete                                                  1e62a48ba86f: Pull complete
dcdba66e4f89: Pull complete                                    f4211e386701:
Pull complete                                                  Digest:
sha256:a6e23dc4f9a14253f2b13c4a1aa4fb1fc116ed347916f3fbdd4b3579d33ea491
Status: Downloaded newer image for quay.io/operator-framework/helm-operator:v1.0.0
 ---> 3bf52f2b9176
Step 2/5 : ENV HOME=/opt/helm
 ---> Running in bae2898cb171
Removing intermediate container bae2898cb171
 ---> 11d2bb25f92b
Step 3/5 : COPY watches.yaml ${HOME}/watches.yaml
 ---> f3c4eebca696
Step 4/5 : COPY helm-charts  ${HOME}/helm-charts
 ---> 482021225333
Step 5/5 : WORKDIR ${HOME}
 ---> Running in d3d82652c1ff
Removing intermediate container d3d82652c1ff
 ---> a492b5562788
Successfully built a492b5562788
Successfully tagged georgeniece/nginx-operator:v1.0.0
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows
Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is
recommended to double check and reset permissions for sensitive files and directories.

c:\projects\kind\operator-sdk\nginx-operator>
```

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:

https://kubernetes-sigs.github.io/kustomize/installation/

For MacOs: we'll use brew

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

For Windows: we'll install this with Chocolatey

https://kubernetes-sigs.github.io/kustomize/installation/chocolatey/

c:\projects\kind\operator-sdk\nginx-operator> **choco install kustomize**

```
Chocolatey v0.10.15
Installing the following packages:
kustomize
By installing you accept licenses for the packages.
Progress: Downloading kustomize 3.8.2... 100%

kustomize v3.8.2 [Approved]
kustomize package files install completed. Performing other installation steps.
The package kustomize wants to run 'chocolateyinstall.ps1'.
```

```
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y

Downloading kustomize 64 bit
  from 'https://github.com/kubernetes-
sigs/kustomize/releases/download/kustomize%2Fv3.8.2/kustomize_v3.8.2_windows_amd64.tar.
gz'
Progress: 100% - Completed download of
C:\ProgramData\chocolatey\lib\kustomize\tools\kustomize_v3.8.2_windows_amd64.tar.gz
(12.59 MB).
Download of kustomize_v3.8.2_windows_amd64.tar.gz (12.59 MB) completed.
Hashes match.
C:\ProgramData\chocolatey\lib\kustomize\tools\kustomize_v3.8.2_windows_amd64.tar.gz
Extracting
C:\ProgramData\chocolatey\lib\kustomize\tools\kustomize_v3.8.2_windows_amd64.tar.gz to
C:\ProgramData\chocolatey\lib\kustomize\tools...
C:\ProgramData\chocolatey\lib\kustomize\tools
Extracting
C:\ProgramData\chocolatey\lib\kustomize\tools\kustomize_v3.8.2_windows_amd64.tar to
C:\ProgramData\chocolatey\lib\kustomize\tools...
C:\ProgramData\chocolatey\lib\kustomize\tools
Installing 64-bit kustomize...
kustomize has been installed.
Added C:\ProgramData\chocolatey\bin\kustomize.exe shim pointed to
'..\lib\kustomize\tools\kustomize.exe'.
 The install of kustomize was successful.
  Software installed to 'C:\ProgramData\chocolatey\lib\kustomize\tools'

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

c:\projects\kube\operator-sdk\nginx-operator> cd
C:\ProgramData\chocolatey\lib\kustomize\tools



C:\ProgramData\chocolatey\lib\kustomize\tools> dir

 Volume in drive C is OS
 Volume Serial Number is 5081-CA53

 Directory of C:\ProgramData\chocolatey\lib\kustomize\tools

09/13/2020  05:57 PM    <DIR>          .
09/13/2020  05:57 PM    <DIR>          ..
09/13/2020  05:56 PM             1,794 chocolateyinstall.ps1
09/13/2020  05:56 PM             1,205 chocolateyuninstall.ps1
08/29/2020  12:50 PM        41,163,776 kustomize.exe
09/13/2020  05:57 PM                 6 kustomize.exe.ignore
09/13/2020  05:57 PM        41,165,312 kustomize_v3.8.2_windows_amd64.tar
```

09/13/2020 05:57 PM   13,203,931 kustomize_v3.8.2_windows_amd64.tar.gz
               6 File(s)   95,536,024 bytes
               2 Dir(s) 137,211,785,216 bytes free

Move the Kustomize.exe that we built to the Bin folder

C:\ProgramData\chocolatey\lib\kustomize\tools> **copy kustomize.exe c:\bin\.**

        1 file(s) copied.

Switch to our operator project build

C:\ProgramData\chocolatey\lib\kustomize\tools> **cd \projects\kind\operator-sdki\nginx-operator**

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

C:\projects\kind\operator-sdk\nginx-operator> **c:\bin\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.

C:\projects\kind\operator-sdk\nginx-operator> **c:\bin\kustomize build config/default | kubectl apply -f -**

namespace/system created
customresourcedefinition.apiextensions.k8s.io/nginxes.demo.my.domain created
clusterrole.rbac.authorization.k8s.io/nginx-operator-manager-role created
clusterrole.rbac.authorization.k8s.io/nginx-operator-proxy-role created
clusterrole.rbac.authorization.k8s.io/nginx-operator-metrics-reader created
clusterrolebinding.rbac.authorization.k8s.io/nginx-operator-manager-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/nginx-operator-proxy-rolebinding created
Error from server (NotFound): error when creating "STDIN": namespaces "nginx-operator-system" not found
Error from server (NotFound): error when creating "STDIN": namespaces "nginx-operator-system" not found
Error from server (NotFound): error when creating "STDIN": namespaces "nginx-operator-system" not found
Error from server (NotFound): error when creating "STDIN": namespaces "nginx-operator-system" not found

We see that the defect in the namespace creation caused a failure of a number of our resource creations. We'll create the missing namespace and retry.

C:\projects\kind\operator-sdk\nginx-operator> **kubectl create namespace nginx-operator-system**

namespace/nginx-operator-system created

Delete the failed resources

C:\projects\kind\operator-sdk\nginx-operator> **c:\bin\kustomize build config/default | kubectl delete -f -**

namespace "system" deleted
customresourcedefinition.apiextensions.k8s.io "nginxes.demo.my.domain" 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
clusterrolebinding.rbac.authorization.k8s.io "nginx-operator-manager-rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "nginx-operator-proxy-rolebinding" deleted
Error from server (NotFound): error when deleting "STDIN": roles.rbac.authorization.k8s.io "nginx-operator-leader-election-role" not found
Error from server (NotFound): error when deleting "STDIN": rolebindings.rbac.authorization.k8s.io "nginx-operator-leader-election-rolebinding" not found
Error from server (NotFound): error when deleting "STDIN": services "nginx-operator-controller-manager-metrics-service" not found
Error from server (NotFound): error when deleting "STDIN": deployments.apps "nginx-operator-controller-manager" not found

C:\projects\kind\operator-sdk\nginx-operator> **c:\bin\kustomize build config/default | kubectl apply -f -**

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

Create our sample Customer Resource against our cluster
C:\projects\kind\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

C:\projects\kind\operator-sdk\nginx-operator> **kubectl delete -f config/samples/demo_v1_nginx.yaml**

Remove our operator

C:\projects\kind\operator-sdk\nginx-operator> **c:\bin\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

C:\projects\kind\operator-sdk\nginx-operator> **kubectl delete namespace nginx-operator-system**

namespace/nginx-operator-system created

Change directory to our k3d folder
C:\projects\kind\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!