Operator SDK

» Initializing a Project with the Operator-SDK

Create a new directory called *redhat* (if it doesn't already exist in the `$GOPATH/src/github.com` directory.

```
mkdir $GOPATH/src/github.com/redhat
```

Navigate to the directory.

```
cd $GOPATH/src/github.com/redhat
```

Create the project scaffold for the new custom operator.

```
operator-sdk new app-operator
cd app-operator
```

Add a new API for the custom resource AppService

```
operator-sdk add api --api-version=app.example.com/v1alpha1 --kind=AppService
```

Add a new controller that watches for AppService

```
operator-sdk add controller --api-version=app.example.com/v1alpha1 --kind=AppService
```

Open the `app-operator` directory in the editor of your choice and follow along as we explore the structure.

Apply the app-operator CRD.

```
kubectl apply -f deploy/crds/app_v1alpha1_appservice_crd.yaml
```

Set the OPERATOR_NAME variable

```
export OPERATOR_NAME=app-operator
```

Run the operator from *outside* the Minishift environment.

```
operator-sdk up local --namespace myproject
```

Your operator is now watching for the existence of an object that matches: `GroupVersionKind(app.example.com/v1alpha1, Kind=App)`

Apply the provided CR.yaml to your cluster. This should trigger the default logic specified in the handler.

```
kubectl create -f deploy/crds/app_v1alpha1_appservice_cr.yaml
```

Verify a busybox pod was created.

```
kubectl get pods
```