Example 1:

BEFORE STARTING:

git clone https://github.com/appuio/operator-sdk-examples

Other Prerequisites

- dep version v0.5.0+.
- git
- go version v1.10+.
- docker version 17.03+.
- kubectl version v1.11.0+.- Access to a kubernetes v.1.11.0+ cluster.

- and Install also, to have a quick try of it, Minishift: https://github.com/minishift/minishift

- Set $GOPATH\

Note: This guide uses minikube version v0.25.0+ as the local kubernetes cluster and quay.io for the public registry.

For this first example of operator we provide a short and fast way to deploy it:

[SHORT VERSION]: few lines and you will deploy the operator

[LONG VERSION]: full steps to deploy the operator

[SHORT VERSION]: few lines and you will deploy the operator

Start and logging with Minishift:

git clone https://github.com/appuio/operator-sdk-examples

```
$ git clone https://github.com/spanichella/operator-sdk-examples
#MOVE THE FOLDER memcached-operator-long under "the GOPATH"
$ cd operator-sdk-examples
$ cd app-operator
$ minishift start
```

(instead of "kubectl" you can also use "*oc*" command instead)

```
$ oc login -u system:admin
$ oc new-project  blogpost-project
$ oc project blogpost-project
```

# Setup Service Account (instead of "kubectl" you can also use "*oc*" command instead)

```
$ kubectl create -f deploy/service_account.yaml
```

# Setup RBAC

```
$ kubectl create -f deploy/role.yaml
$ kubectl create -f deploy/role_binding.yaml
```

# Setup the CRD

```
$ kubectl create -f deploy/crds/app_v1alpha1_appservice_crd.yaml
```

# Deploy the app-operator

```
$ kubectl create -f deploy/operator.yaml
```

# Create an AppService CR

# The default controller will watch for AppService objects and create a pod for each CR

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

# Verify that a pod is created

```
$ kubectl get pod
NAME                                       READY     STATUS    RESTARTS   AGE
app-operator-bf4c4f8c6-jtljx               1/1       Running   0          7h
$ kubectl get deployment
NAME                      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
app-operator              1         1         1            1           7h
```

# Cleanup

```
$ kubectl delete -f deploy/crds/app_v1alpha1_appservice_cr.yaml
$ kubectl delete -f deploy/operator.yaml
$ kubectl delete -f deploy/role.yaml
$ kubectl delete -f deploy/role_binding.yaml
$ kubectl delete -f deploy/service_account.yaml
$ kubectl delete -f deploy/crds/app_v1alpha1_appservice_crd.yaml
$ oc delete project  blogpost-project
```

[END SHORT VERSION]

[LONG VERSION]: full steps to deploy the operator

Quick Start

First, checkout and install the operator-sdk CLI:

```
$ mkdir -p $GOPATH/src/github.com/operator-framework
$ cd $GOPATH/src/github.com/operator-framework
$ git clone https://github.com/operator-framework/operator-sdk
$ cd operator-sdk
$ git checkout master
$ make dep
$ make install
```

Create and deploy an app-operator using the SDK CLI:

# Create an app-operator project that defines the App CR. (See how to set a $GOPATH first)

```
$ mkdir -p $GOPATH/src/github.com/example-inc/
```

# Create a new app-operator project

```
$ cd $GOPATH/src/github.com/example-inc/
$ operator-sdk new app-operator


INFO[0000] Creating new Go operator 'app-operator'.
INFO[0000] Create cmd/manager/main.go
INFO[0000] Create build/Dockerfile
INFO[0000] Create deploy/service_account.yaml
INFO[0000] Create deploy/role.yaml
INFO[0000] Create deploy/role_binding.yaml
INFO[0000] Create deploy/operator.yaml
INFO[0000] Create pkg/apis/apis.go
INFO[0000] Create pkg/controller/controller.go
INFO[0000] Create version/version.go
INFO[0000] Create .gitignore
INFO[0000] Create Gopkg.toml
INFO[0000] Run dep ensure ...
...
```

```
$ 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
```

# Build and push the app-operator image to a public registry such as quay.io. OR...

```
$ operator-sdk build quay.io/example/app-operator
$ docker push quay.io/example/app-operator
```

...OR directly from DOCKER:

```
$ sudo docker login
$ operator-sdk build  <docker id>/app-operator:v.1.0
 (e.g.,  operator-sdk build  docker.io/spanichella/app-operator )
$ docker push   <docker id>/app-operator:v.1.0
(e.g., docker push  docker.io/spanichella/app-operator)
```

THE OPERATOR SHOULD BE THEN VISIBLE ON YOUR: https://cloud.docker.com/repository/list

# Update the operator manifest to use the built image name (if you are performing these steps on OSX, see note below) OR replace it according the Docker paths

```
$ sed -i 's|REPLACE_IMAGE|quay.io/example/app-operator|g' deploy/operator.yaml
```

# On OSX use (OR replace it according the Docker paths):

```
$ sed -i "" 's|REPLACE_IMAGE|quay.io/example/app-operator|g' deploy/operator.yaml
```

E.g., according the Docker paths:

```
$ sed -i "" 's|REPLACE_IMAGE|docker.io/<docker id>/app-operator|g' deploy/operator.yaml.
 (e.g., sed -i "" 's|REPLACE_IMAGE|docker.io/spanichella/app-operator|g' deploy/operator.yaml)
```

Start and logging with Minishift:

```
$ minishift start
```

(instead of "kubectl" you can also use "*oc*" command instead)

```
$ oc login -u system:admin
$ oc new-project  blogpost-project
$ oc project blogpost-project
```

# Setup Service Account (instead of "kubectl" you can also use "*oc*" command instead)

```
$ kubectl create -f deploy/service_account.yaml
```

# Setup RBAC

```
$ kubectl create -f deploy/role.yaml
$ kubectl create -f deploy/role_binding.yaml
```

# Setup the CRD

```
$ kubectl create -f deploy/crds/app_v1alpha1_appservice_crd.yaml
```

# Deploy the app-operator

```
$ kubectl create -f deploy/operator.yaml
```

# Create an AppService CR
# The default controller will watch for AppService objects and create a pod for each CR

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

# Verify that a pod is created

```
$ kubectl get pod
NAME READY STATUS RESTARTS AGE
example-appservice-pod 1/1 Running 0 1m
```

# Cleanup

```
$ kubectl delete -f deploy/crds/app_v1alpha1_appservice_cr.yaml
$ kubectl delete -f deploy/operator.yaml
$ kubectl delete -f deploy/role.yaml
$ kubectl delete -f deploy/role_binding.yaml
$ kubectl delete -f deploy/service_account.yaml
$ kubectl delete -f deploy/crds/app_v1alpha1_appservice_crd.yaml
$ oc delete project  blogpost-project
```

[END LONG VERSION]