

# Simple Operator Example:

## BEFORE STARTING:

1) **Install required tools:** (Short link: <https://bit.ly/2F5IUfz>)

2) **Install Operator SDK** (Short link: <https://bit.ly/2WuJWri>)

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

**[LONG VERSION]:** full steps to deploy the operator

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

**[LONG VERSION]: full steps to deploy the operator**

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

```
$ minishift start

"
The server is accessible via web console at:
https://192.168.64.2:8443

You are logged in as:
User: developer
Password: developer

To login as administrator:
oc login -u system:admin
"

$ oc login -u system:admin

# Create an app-operator project that defines the App CR.
$ mkdir -p $GOPATH/src/github.com/example-inc/
# Create a new app-operator project
$ cd $GOPATH/src/github.com/example-inc/
$ export GO111MODULE=on
$ 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

INFO[0000] Generating api version app.example.com/v1alpha1 for kind AppService.
INFO[0001] Created pkg/apis/app/v1alpha1/appservice_types.go
INFO[0001] Created pkg/apis/addtoscheme_app_v1alpha1.go
INFO[0001] Created pkg/apis/app/v1alpha1/register.go
INFO[0001] Created pkg/apis/app/v1alpha1/doc.go
INFO[0001] Created deploy/crds/app_v1alpha1_appservice_cr.yaml
...
```

```

# Add a new controller that watches for AppService
$ operator-sdk add controller --api-version=app.example.com/v1alpha1 --kind=AppService

INFO[0000] Generating controller version app.example.com/v1alpha1 for kind AppService.
INFO[0000] Created pkg/controller/appservice/appservice_controller.go
INFO[0000] Created pkg/controller/add_appservice.go
INFO[0000] Controller generation complete.

# Build and push the app-operator image to a public registry 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)

# Update the operator manifest to use the built image name (if you are performing these steps on
OSX, see note below)
# $ 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 pods
      NAME                                READY   STATUS             RESTARTS   AGE
app-operator-77fc4bcddf-g7h2h            0/1     ContainerCreating   0           29s

# Test the new Resource Type
$ kubectl describe appservice example-appservice
Name:         example-appservice
Namespace:    blogpost-project
Labels:       <none>
Annotations:  <none>
API Version:  app.example.com/v1alpha1
Kind:         AppService
Metadata:
  Creation Timestamp: 2019-06-12T18:22:23Z
  Generation:        1
  Resource Version:   1488007
  Self Link:          /apis/app.example.com/v1alpha1/namespaces/blogpost-project/appservices/example-appservice
  UID:                062bf8e8-8d3f-11e9-8435-ceff9172fca2
Spec:
  Size: 3
  Events: <none>

# 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

```

**[END LONG VERSION]**

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

```
# 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 pods
      NAME                                READY   STATUS             RESTARTS   AGE
app-operator-77fc4bcd9f-g7h2h            0/1     ContainerCreating   0           29s

# Test the new Resource Type
$ kubectl describe appservice example-appservice
Name:      example-appservice
Namespace: blogpost-project
Labels:    <none>
Annotations: <none>
API Version: app.example.com/v1alpha1
Kind:      AppService
Metadata:
  Creation Timestamp: 2019-06-12T18:22:23Z
  Generation:        1
  Resource Version:   1488007
  Self Link:          /apis/app.example.com/v1alpha1/namespaces/blogpost-project/appservices/example-appservice
  UID:                062bf8e8-8d3f-11e9-8435-ceff9172fca2
Spec:
  Size: 3
  Events: <none>

# 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
```

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