

Ansible Operator

» Ansible Operator

Create a new directory for your Ansible-Operator project.

```
mkdir ~/ansible-operator
cd ~/ansible-operator
```

Use the CLI to create a new Ansible-based Operator project with the new command. `operator-sdk new --type ansible` has two required flags `--api-version` and `--kind`. These flags are used to generate proper Custom Resource files and an Ansible Role whose name matches the input for `--kind`.

```
operator-sdk new memcached-operator --api-version=cache.example.com/v1alpha1 --kind=Memcached --type=ansible --skip-git-init
cd memcached-operator
```

To speed things up, we can reuse a role that has been written inside of our operator. The role we will use is `dymurray.memcached_operator_role`. The section at the bottom of this document will also show the reader how to create that Ansible Role from scratch. To get started, install the Ansible Role inside of the project.

```
ansible-galaxy install dymurray.memcached_operator_role -p ./roles
```

Delete generated scaffolding role

```
rm -rf ./roles/Memcached
```

This role provides the user with a variable `size` which is an integer to control the number of replicas to create. You can find the default for this variable in `roles/dymurray.memcached_operator_role/defaults/main.yml`.

```
cat roles/dymurray.memcached_operator_role/defaults/main.yml
```

View the Ansible role.

```
cat roles/dymurray.memcached_operator_role/tasks/main.yml
```

Update the `watches.yaml` file

```
cat > watches.yaml <<EOF
---
- version: v1alpha1
  group: cache.example.com
  kind: Memcached
  role: /home/vagrant/ansible-operator/memcached-operator/roles/dymurray.memcached_operator_role
EOF
```

Before running the operator, Kubernetes needs to know about the new custom resource definition the operator will be watching.

```
kubectl create -f deploy/crds/cache_v1alpha1_memcached_crd.yaml
```

Run the operator locally with the default kubernetes config file present at \$HOME/.kube/config.

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

Create the Memcached CR.

```
kubectl create -f deploy/crds/cache_v1alpha1_memcached_cr.yaml
```