Operators

» Custom Resource Definitions

Overview

Custom Resource Definitions allow Kubernetes users to extend the Kubernetes API….via the Kubernetes API!

Begin by running a proxy to the Kubernetes API server.

```
oc proxy --port=8001
```

Observe the **crd** *api resource* within the **extensions** *api group*.

```
curl http://localhost:8001/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions | jq
```

Let’s create a new crd resource object manifest for Postgres.

```
cat >> postgres-crd.yaml <<EOF
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: postgreses.rd.example.com
spec:
  group: rd.example.com
  names:
    kind: Postgres
    listKind: PostgresList
    plural: postgreses
    singular: postgres
    shortNames:
    - pg
  scope: Namespaced
  version: v1alpha1
EOF
```

Create the ***crd*** resource object.

```
oc create -f postgres-crd.yaml
```

You should now see the Kubernetes API reflect a brand new *api group* called **rd.example.com**.

```
curl http://localhost:8001/apis | jq .groups[].name
```

This will also be reflected in the `oc api-versions` command.

```
oc api-versions
```

Within the `rd.example.com` group there will an *api version* **v1alpha1** (per our crd resource object). The database resource resides here.

```
curl http://localhost:8001/apis/rd.example.com/v1alpha1 | jq
```

Notice how `oc` now recognize postgres as a present resource (although there will be no current resource objects at this time).

```
oc get postgres
```

Run the command again using the –v=8 option to see the API transactions. You will see an API call being made to
`curl http://localhost:8001/apis/rd.example.com/v1alpha1/namespaces/myproject/postgreses`

```
oc get postgres --v=8
```

Let’s create a new **resource object** manifest for the database **resource**.

```
cat >> wordpress-database.yaml <<EOF
apiVersion: "rd.example.com/v1alpha1"
kind: Postgres
metadata:
  name: wordpressdb
spec:
  user: postgres
  password: postgres
  database: primarydb
  nodes: 3
EOF
```

Create the new object.

```
oc create -f wordpress-database.yaml
```

Verify the resource was created.

```
oc get postgres
```

View the details about the wordpressdb object

```
oc get postgres wordpressdb -o yaml
```