

## Chapter 13

# Custom Resource Definition



### 13.1 Labs

#### Exercise 13.1: Create a Custom Resource Definition

##### Overview

`ThirdPartyResource` is no longer included with the API in v1.8 and its use will return a validation error. If you have upgraded from a version prior to Kubernetes v1.7, you will need to convert them to `CustomResourceDefinitions` (CRD). A new resource often requires a controller to manage the resource. Creation of the controller is beyond the scope of this course, basically it is a watch-loop comparing a spec file to the current state and making changes until the states match. A good discussion of creating a controller can be found here: <https://coreos.com/blog/introducing-operators.html>.

##### Create a Custom Resource Definition

We will make a simple CRD, but without any particular action. It will be enough to find the object ingested into the API and responding to commands.

1. We will create a new YAML file.

```
student@lfs458-node-1a0a:~$ vim crd.yaml

apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: crontabs.training.lfs458.com
  # This name must match names below.
  # <plural>.<group> syntax
spec:
  scope: Cluster      #Could also be Namespaced
  group: training.lfs458.com
```

```

version: v1
names:
  kind: CronTab      #Typically CamelCased for resource manifest
  plural: crontabs   #Shown in URL
  singular: crontab  #Short name for CLI alias
  shortNames:
    - ct             #CLI short name

```

2. Add the new resource to the cluster.

```

student@lfs458-node-1a0a:~$ kubectl create -f crd.yaml
customresourcedefinition "crontabs.training.lfs458.com" created

```

3. View and describe the resource.

```

student@lfs458-node-1a0a:~$ kubectl get crd
NAME                                AGE
crontabs.training.lfs458.com        3m

student@lfs458-node-1a0a:~$ kubectl describe crd
Name:                                crontabs.training.lfs458.com
Namespace:
Labels:                               <none>
Annotations:                          <none>
API Version:  apiextensions.k8s.io/v1beta1
Kind:           CustomResourceDefinition
<output_omitted>

```

4. Now that we have a new API resource we can create a new object of that type. In this case it will be a crontab-like image, which does not actually exist, but is being used for demonstration.

```

student@lfs458-node-1a0a:~$ vim new-crontab.yaml

apiVersion: "training.lfs458.com/v1"
  # This is from the group and version of new CRD
kind: CronTab
  # The kind from the new CRD
metadata:
  name: new-cron-object
spec:
  cronSpec: "*/5 * * * *"
  image: some-cron-image
    #Does not exist

```

5. Create the new object and view the resource using short and long name.

```

student@lfs458-node-1a0a:~$ kubectl create -f new-crontab.yaml
crontab "new-cron-object" created

student@lfs458-node-1a0a:~$ kubectl get CronTab
NAME                                AGE
new-cron-object                    22s

student@lfs458-node-1a0a:~$ kubectl get ct
NAME                                AGE
new-cron-object                    29s

student@lfs458-node-1a0a:~$ kubectl describe ct
Name:                                new-cron-object
Namespace:
Labels:                               <none>

```

<output\_omitted>

Spec:

    Cron Spec: \*/5 \* \* \* \*  
    Image:      some-cron-image  
Events:      <none>

6. To clean up the resources we will delete the CRD. This should delete all of the endpoints and objects using it as well.

```
student@lfs458-node-1a0a:~$ kubectl delete -f crd.yaml
customresourcedefinition "cronabs.training.lfs458.com" deleted

student@lfs458-node-1a0a:~$ kubectl get ct
Error from server (NotFound): Unable to list "cronabs": the server
could not find the requested resource
(get cronabs.training.lfs458.com)
```