**Lesson 3 Demo 9**

**Understanding the Working of API Servers**



Steps to be followed:

1. Defining Custom API Resources

**Step 1: Defining Custom API Resources**

1. Create a file with the following command:

***vi customresourcedef.yaml***



1. Write the following code in the **customresourcedef.yaml** file:

***apiVersion: apiextensions.k8s.io/v1***

***kind: CustomResourceDefinition***

***metadata:***

***# name must match the spec fields below, and be in the form: <plural>.<group>***

***name: crontabs.stable.example.com***

***spec:***

***# group name to use for REST API: /apis/<group>/<version>***

***group: stable.example.com***

***# list of versions supported by this CustomResourceDefinition***

***versions:***

***- name: v1***

***# Each version can be enabled/disabled by Served flag.***

***served: true***

***# One and only one version must be marked as the storage version.***

***storage: true***

***schema:***

***openAPIV3Schema:***

***type: object***

***properties:***

***spec:***

***type: object***

***properties:***

***cronSpec:***

***type: string***

***image:***

***type: string***

***replicas:***

***type: integer***

***# either Namespaced or Cluster***

***scope: Namespaced***

***names:***

***# plural name to be used in the URL: /apis/<group>/<version>/<plural>***

***plural: crontabs***

***# singular name to be used as an alias on the CLI and for display***

***singular: crontab***

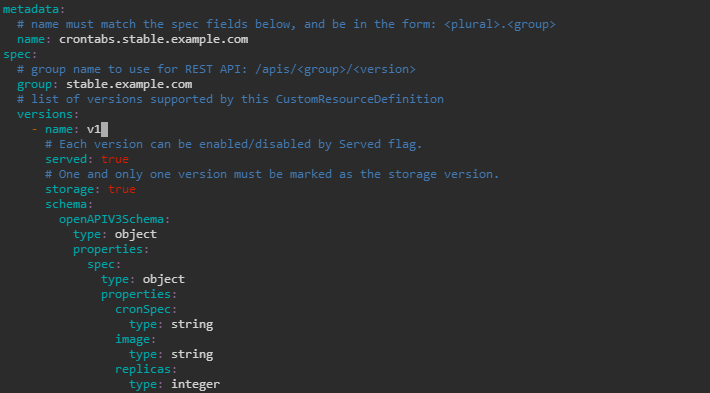
***# kind is normally the CamelCased singular type. Your resource manifests use this.***

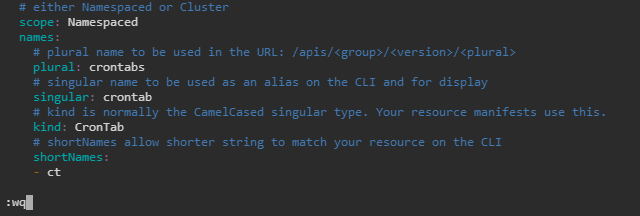
***kind: CronTab***

***# shortNames allow shorter string to match your resource on the CLI***

***shortNames:***

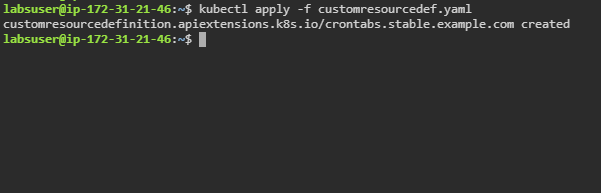
***- ct***





1. Run the following command to create the custom resource:

***kubectl apply -f customresourcedef.yaml***



1. Check the newly created REST Endpoint using the following command:

***kubectl api-resources***

