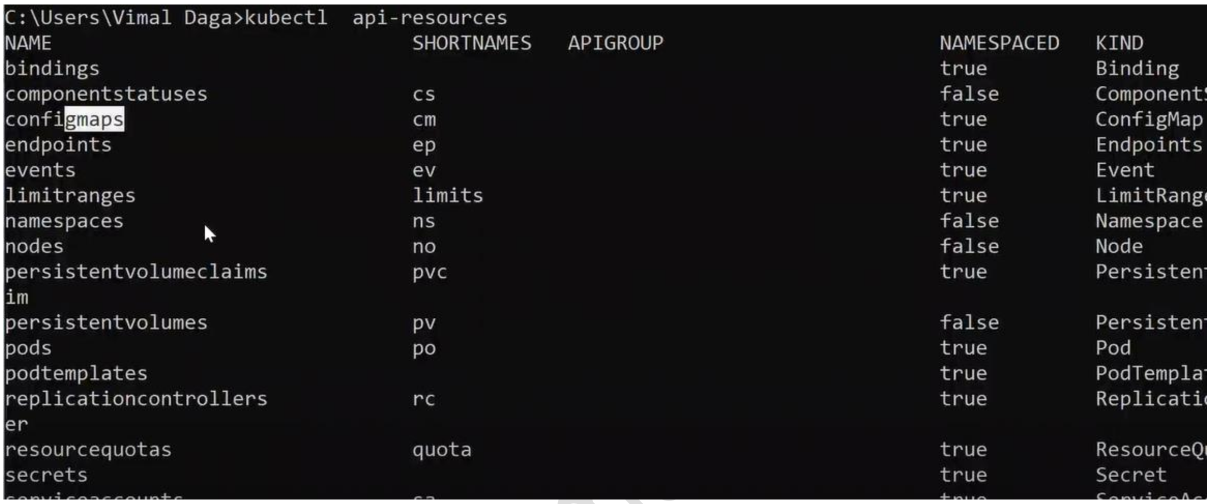
Summary

- Command for checking all available resources which are accessible through API

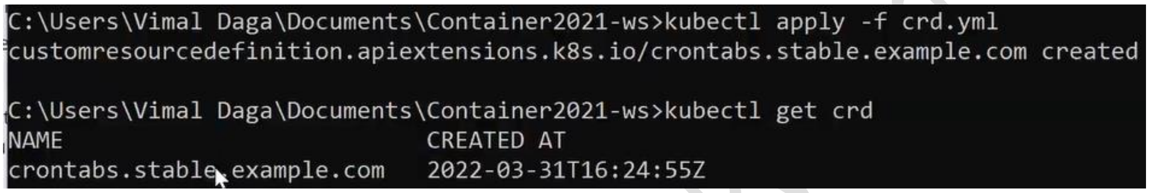


```
C:\Users\Vimal Daga>kubectl  api-resources
NAME                              SHORTNAMES   APIGROUP                       NAMESPACED   KIND
bindings                                                                      true         Binding
componentstatuses                 cs                                          false        Component
configmaps                        cm                                          true         ConfigMap
endpoints                         ep                                          true         Endpoints
events                            ev                                          true         Event
limitranges                       limits                                      true         LimitRang
namespaces                        ns                                          false        Namespace
nodes                             no                                          false        Node
persistentvolumeclaims            pvc                                         true         Persistent
im
persistentvolumes                 pv                                          false        Persistent
pods                              po                                          true         Pod
podtemplates                                                                  true         PodTempla
replicationcontrollers            rc                                          true         Replicati
er
resourcequotas                    quota                                       true         ResourceQ
secrets                                                                       true         Secret
serviceaccounts                   sa                                          true         ServiceAc
```

- A custom resource is an extension of Kubernetes API
- Through custom resource definition (CRD) we can create our custom resources in Kubernetes
- Example 1:-Creating Custom resource definition
  - Manifest file for CRD

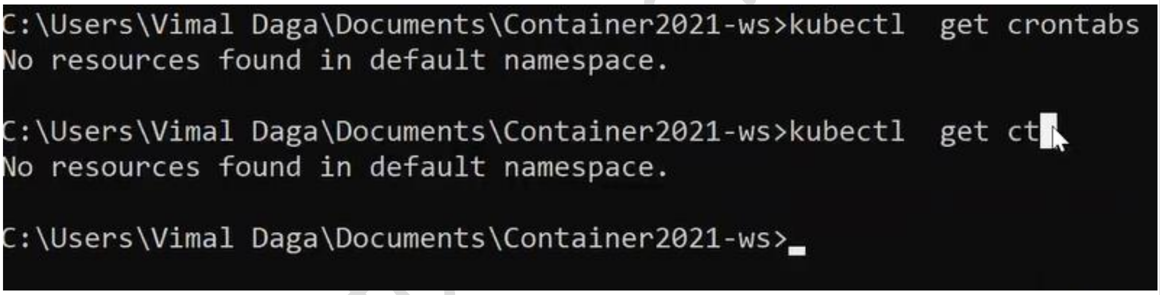
```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  # name must match the spec fields below, and be in the form: <plural>.<group>
  name: crontabs.stable.example.com
spec:
  # group name to use for REST API: /apis/<group>/<version>
  group: stable.example.com
  # list of 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
```

- Creating CRD from the manifest file



```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f crd.yml
customresourcedefinition.apiextensions.k8s.io/crontabs.stable.example.com created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get crd
NAME                            CREATED AT
crontabs.stable.example.com     2022-03-31T16:24:55Z
```

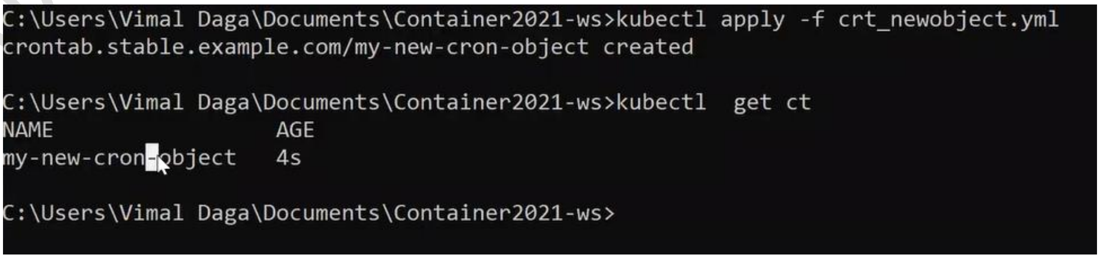
- Checking resources



```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get crontabs
No resources found in default namespace.

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get ct
No resources found in default namespace.

C:\Users\Vimal Daga\Documents\Container2021-ws>
```

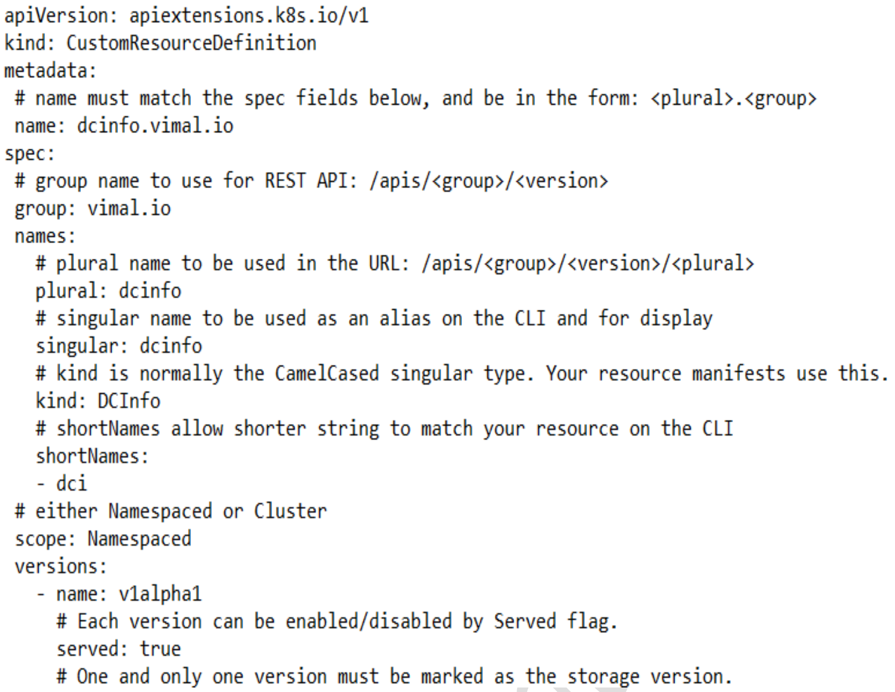
- Creating custom pod resource
  - Manifest file

```
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
  name: my-new-cron-object
spec:
  cronSpec: "* * * * */5"
  image: my-awesome-cron-image
```

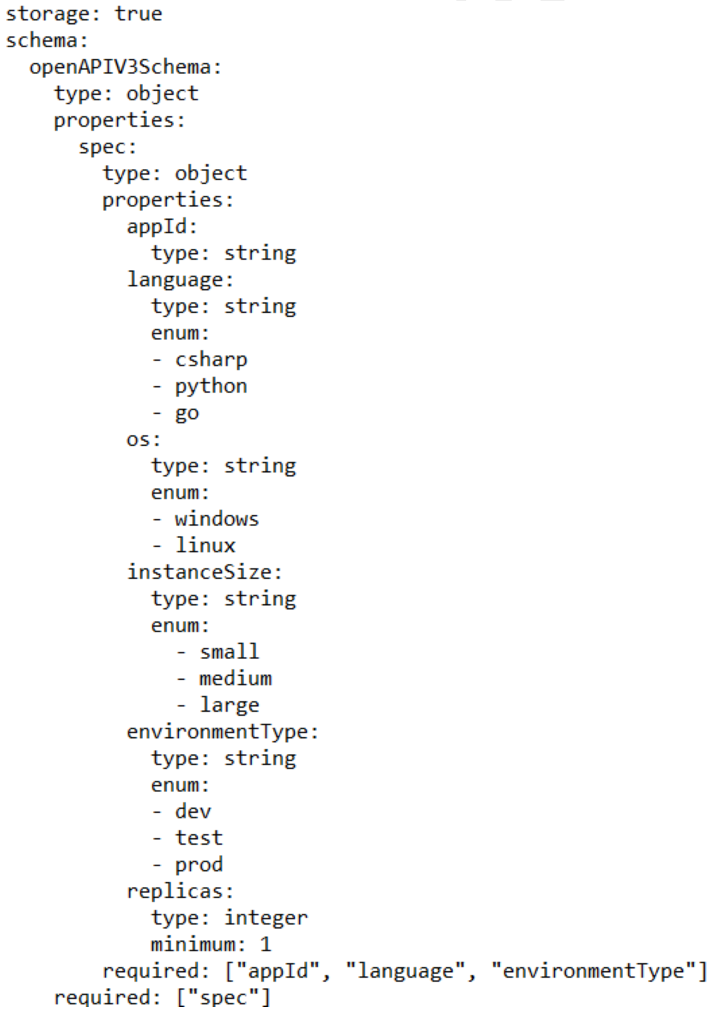


```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f crt_newobject.yml
crontab.stable.example.com/my-new-cron-object created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get ct
NAME                 AGE
my-new-cron-object   4s

C:\Users\Vimal Daga\Documents\Container2021-ws>
```

- Example 2:- Creating a custom resource definition
    - Manifest file

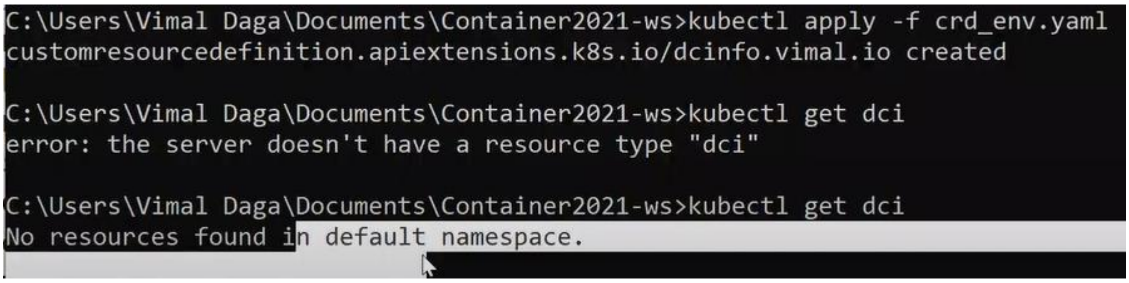


```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
 # name must match the spec fields below, and be in the form: <plural>.<group>
 name: dcinfo.vimal.io
spec:
 # group name to use for REST API: /apis/<group>/<version>
 group: vimal.io
 names:
   # plural name to be used in the URL: /apis/<group>/<version>/<plural>
   plural: dcinfo
   # singular name to be used as an alias on the CLI and for display
   singular: dcinfo
   # kind is normally the CamelCased singular type. Your resource manifests use this.
   kind: DCInfo
   # shortNames allow shorter string to match your resource on the CLI
   shortNames:
   - dci
 # either Namespaced or Cluster
 scope: Namespaced
 versions:
   - name: v1alpha1
     # Each version can be enabled/disabled by Served flag.
     served: true
     # One and only one version must be marked as the storage version.
```



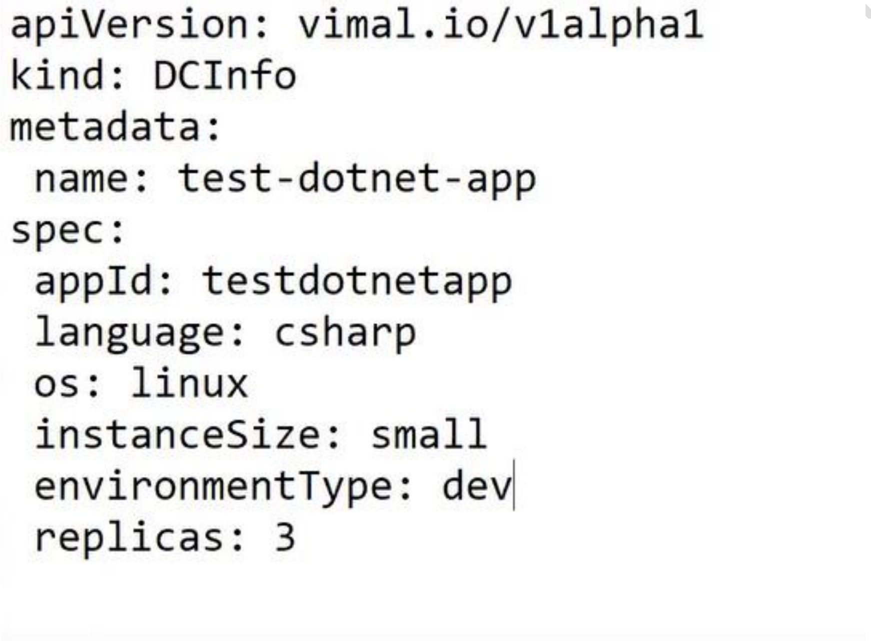
```
     storage: true
     schema:
       openAPIV3Schema:
         type: object
         properties:
           spec:
             type: object
             properties:
               appId:
                 type: string
               language:
                 type: string
                 enum:
                 - csharp
                 - python
                 - go
               os:
                 type: string
                 enum:
                 - windows
                 - linux
               instanceSize:
                 type: string
                 enum:
                   - small
                   - medium
                   - large
               environmentType:
                 type: string
                 enum:
                 - dev
                 - test
                 - prod
               replicas:
                 type: integer
                 minimum: 1
             required: ["appId", "language", "environmentType"]
         required: ["spec"]
```

- Creating resources from a manifest file



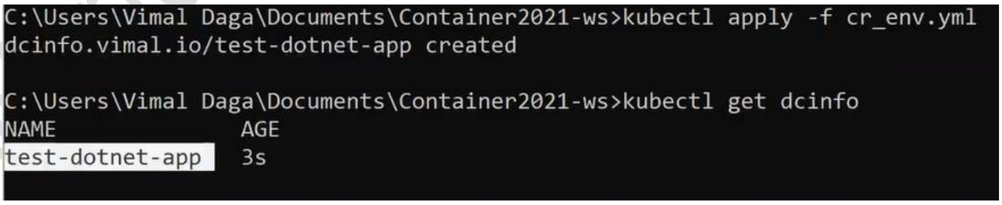
```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f crd_env.yaml
customresourcedefinition.apiextensions.k8s.io/dcinfo.vimal.io created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get dci
error: the server doesn't have a resource type "dci"

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get dci
No resources found in default namespace.
```

- Creating resource
  - Manifest file



```
apiVersion: vimal.io/v1alpha1
kind: DCInfo
metadata:
 name: test-dotnet-app
spec:
 appId: testdotnetapp
 language: csharp
 os: linux
 instanceSize: small
 environmentType: dev
 replicas: 3
```

  - Creating resources from a manifest file



```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f cr_env.yml
dcinfo.vimal.io/test-dotnet-app created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get dcinfo
NAME              AGE
test-dotnet-app   3s
```