KUBERNETES (ENGLISH)

5. forduló

A kategória támogatója: Nokia

Ismertető a feladatlaphoz

> Kezdj neki minél hamarabb, mert a feladatot a forduló záró időpontjáig lehet befejezni, nem addig lehet elkezdeni!
>
> **A feladatot nem ajánlott telefonon / tableten megoldani!**

Sok sikert!

Kubernetes Custom Resources:

Custom Resources are extensions of the Kubernetes API. Custom resources can appear and disappear in a running cluster through dynamic registration, and cluster admins can update custom resources independently of the cluster itself. Once a custom resource is installed, users can create and access its objects just as they do for built-in resources.

Before you start, please read the following hint:

KDiff3 is a free and open-source diff and merge tool that can be helpful during the solution of coding-related exercises.
Download link: https://sourceforge.net/projects/kdiff3/files/

1. feladat 1 pont

Your task is to implement a custom Chicken resource for Kubernetes, but the specification is likely to change in the near future. Which definition would you choose for your first experiments?

Válasz

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: chickens.example.com
spec:
  group: example.com
  versions:
    - name: v1alpha1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec: *
  scope: Namespaced
  names:
    plural: chickens
    singular: chicken
    kind: Chicken
    shortNames:
    - chicken
```

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: chickens.example.com
spec:
  group: example.com
  versions:
    - name: v1
      served: true
      storage: true
  scope: Namespaced
  names:
    plural: chickens
    singular: chicken
```

```
    kind: Chicken
    shortNames:
    - chck
```



```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: chickens.example.com
spec:
  group: example.com
  versions:
    - name: v1alpha1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                nickName:
                  type: string
                age:
                  type: integer
  scope: Namespaced
  names:
    plural: chickens
    singular: chicken
    kind: Chicken
    shortNames:
    - chck
```



```
apiVersion: apiextensions.k8s.io/v1alpha1
kind: CustomResourceDefinition
metadata:
  name: chickens.example.com
spec:
  group: example.com
  versions:
    - name: v1
      served: true
      storage: true
      schema:
```

```
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                nickName:
                  type: string
                age:
                  type: integer
  scope: Namespaced
  names:
    plural: chickens
    singular: chicken
    kind: Chicken
    shortNames:
    - chck
```



```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: chickens.example.com
spec:
  group: example.com
  versions:
    - name: v1
      served: false
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                nickName:
                  type: string
                age:
                  type: integer
  scope: Namespaced
  names:
    plural: chickens
    singular: chicken
    kind: Chicken
```

```
    shortNames:
    - chickens
```

2. feladat 1 pont

Your company has released a product that includes a Stuff CRD for Kubernetes:

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: stuffs.mycompany.com
spec:
  group: mycompany.com
  versions:
    - name: v1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                size:
                  type: integer
  scope: Namespaced
  names:
    plural: stuffs
    singular: stuff
    kind: Stuff
    shortNames:
    - stuff
```

However, a competitor has appeared, and their definition looks like:

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: stuffs.anothercompany.com
spec:
```

```
group: anothercompany.com
versions:
  - name: v1
    served: true
    storage: true
    schema:
      openAPIV3Schema:
        type: object
        properties:
          spec:
            type: object
            properties:
              size:
                type: integer
scope: Namespaced
names:
  plural: stuffs
  singular: stuff
  kind: Stuff
  shortNames:
  - stuff
```

What happens when both of these are installed to the customer's Kubernetes cluster?

Válasz

- ( ) It is not possible to install these 2 CRDs to the same cluster.
- ( ) Instruction materials will need to be revised to use a more exact name, such as stuff.mycompany for kubectl commands.
- ( ) Any controllers your company released, will also handle competitor Stuff instances.
- ( ) It will not be possible any more to use kubectl to manage Stuff resources.
- ( ) Existing Stuff resources will be deleted when installing the second CRD.

3. feladat 3 pont

Your development team has implemented a business logic using Chicken and Egg custom resources, but there is a bug: the associated controller will not let Chicken resources be deleted as long as there are Egg resources, and will not let Egg resources be deleted as long as there are Chicken resources.

A redesign is being considered, but you have already have both Chicken and Egg instances in your cluster. How can the Kubernetes cluster be restored to its initial state?

Válasz

- Only by Kubernetes reinstall.
- Delete both Chicken and Egg resources with the same kubectl command, as this will initiate the two operations in the same API call.
- Delete the namespace, as this operation will bypass finalizers.
- Remove the metadata.finalizers entry from Chicken and/or Egg instances, as this will allow deletion of the instances.
- Delete the controller pod, as the default is to proceed with the deletion if the webhook does not respond.

Megoldások beküldése