Replication

OW

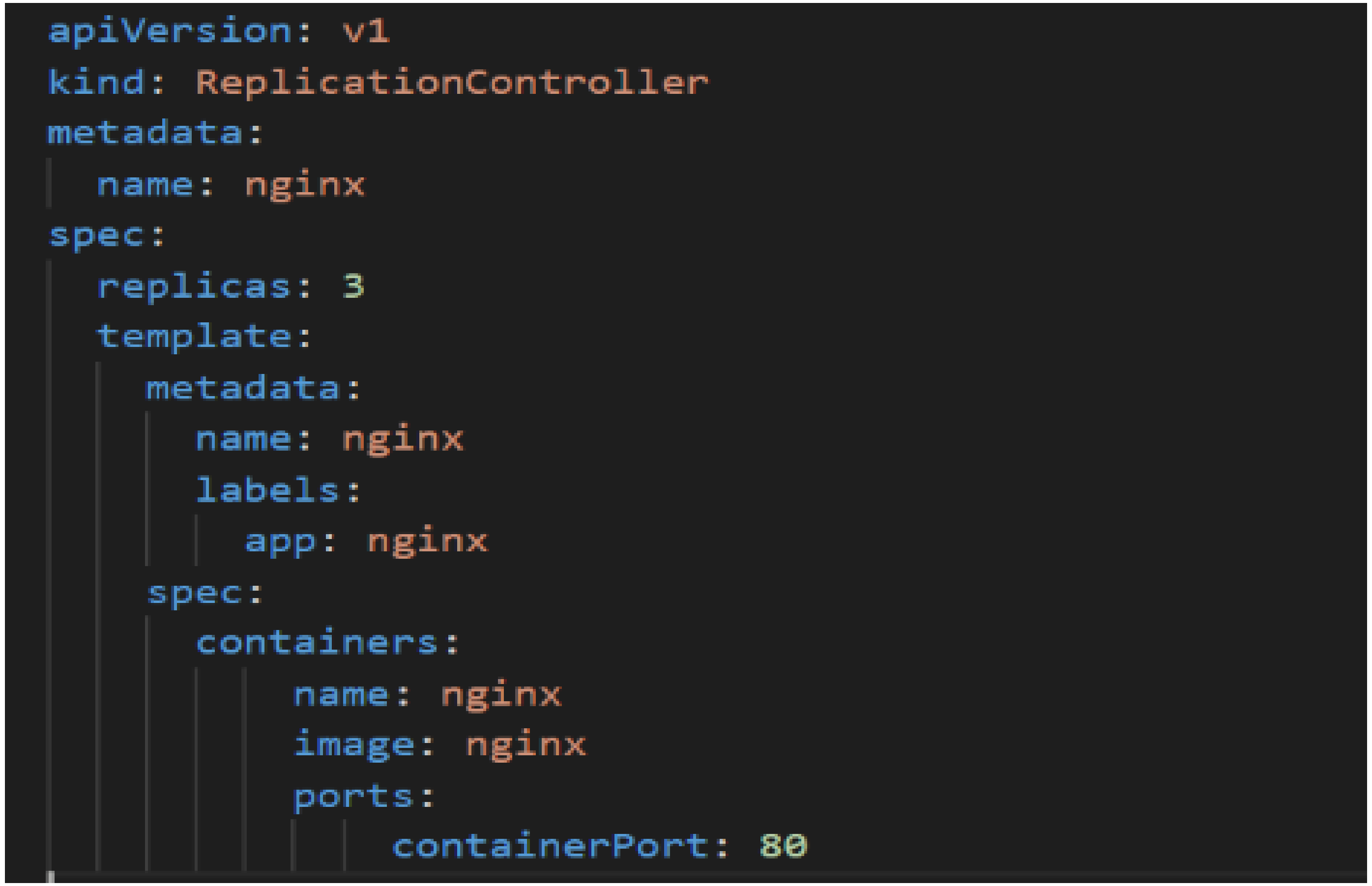
What is replication

- Typically we want to replicate our containers i.e. our application for several reasons which include reliability, load balancing, and scaling. By having multiple versions of the application we prevent problems if one or more pods fail.
- load balancing by having multiple versions of the containers enables us to easily send traffic to different instances to prevent overloading a single instance or not

Replication controller

- The Replication Controller is a structure that enables us to easily create multiple pods.
- If we make sure that a number of pods always exists. If a pod crashes, the Replication Controller replaces it with a new pod.
- The Replication Controller also provides other benefits such as the ability to scale the number of pods and to update or delete multiple pods with a single command.

yaml



```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx
spec:
  replicas: 3
  template:
    metadata:
      name: nginx
      labels:
        app: nginx
    spec:
      containers:
          name: nginx
          image: nginx
          ports:
              containerPort: 80
```

Yaml explanation

- here we will provide the name for the Replication Controller and under the spec, we will be defining how many replicas we want from the given pod definition under the template section. Under the template section, we will be providing the exact details that we have provided in a pod definition yml file that will include metadata labels and under the spec, we can define containers, init containers, different images, ports, different configmap volume mounts, environment variables all of those configurations basically related to pods we can provide it under the template section.
- The Replication Controller can have an optional selector and spec, where we can provide the labels used in the pods which is used to label query over the pods that should match with the replica count. When the selector is not provided it will assume that the provided template labels will be used as the selector.

cmds

```
kubectl apply -f <replication controller yaml>
```

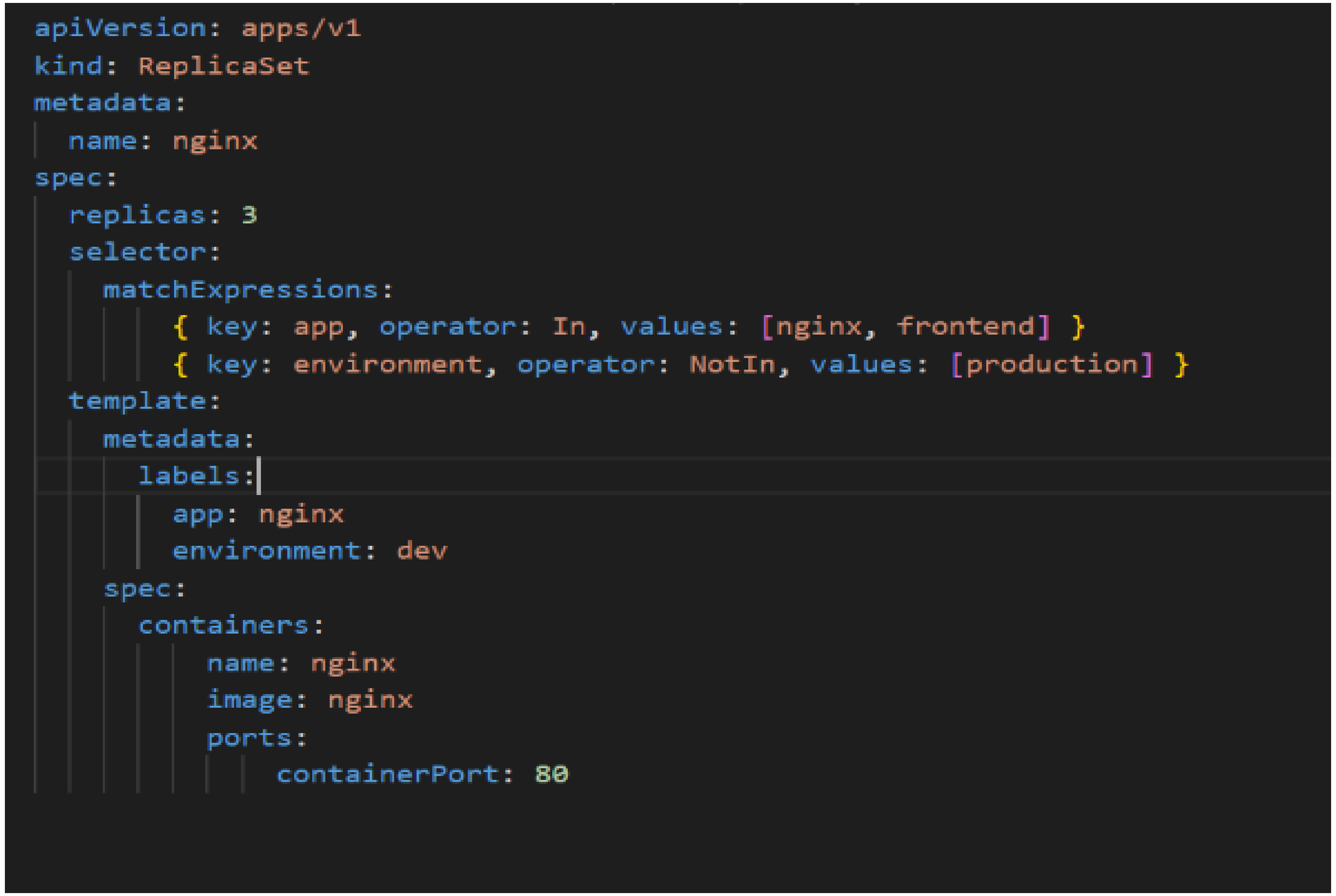
Check on the status of the ReplicationController using this command:

```
kubectl describe <name of replication controller>
```

Replica set

- Replica Sets are declared in the same way as Replication Controller except that they have more options for the selectors.
- The Selector is mandatory for Replica sets as match labels you can provide the pod labels to query the pods to match with the replica count.

yaml

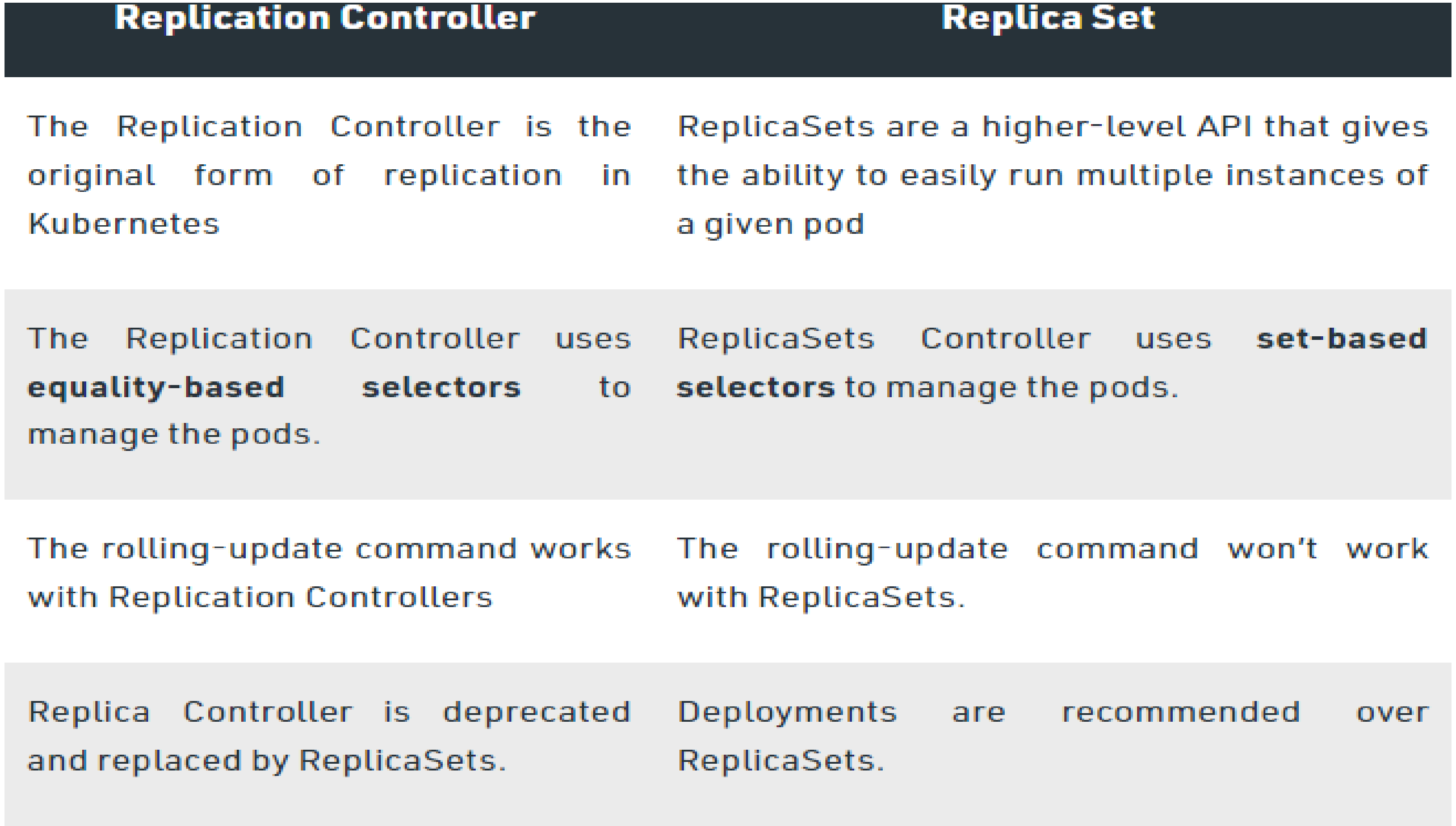


```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchExpressions:
        { key: app, operator: In, values: [nginx, frontend] }
        { key: environment, operator: NotIn, values: [production] }
  template:
    metadata:
      labels:
        app: nginx
        environment: dev
    spec:
      containers:
          name: nginx
          image: nginx
          ports:
              containerPort: 80
```

..

- Here we can see, the API version is apps v1, the kind is ReplicaSet, and then we will be defining what is our ReplicaSet name under the spec section we can define the number of replicas, and under selectors match labels we will define what are our pod labels which need to be considered for the replica counts.
- Under the template section, we can provide our pod template which includes the containers, images, and labels, under the selector section not only the matching label we can use match expressions as well to configure different conditions for the selector.

Difference



Replication Controller	Replica Set
The Replication Controller is the original form of replication in Kubernetes	ReplicaSets are a higher-level API that gives the ability to easily run multiple instances of a given pod
The Replication Controller uses **equality-based selectors** to manage the pods.	ReplicaSets Controller uses **set-based selectors** to manage the pods.
The rolling-update command works with Replication Controllers	The rolling-update command won't work with ReplicaSets.
Replica Controller is deprecated and replaced by ReplicaSets.	Deployments are recommended over ReplicaSets.