Lab: Working with ReplicaSets

Introduction:

A ReplicaSet ensures that a specified number of pod replicas are running at any one time and a Replicaset makes sure that a pod or a homogeneous set of pods is always up and available.

A ReplicaSet is defined with fields, including a selector that specifies how to identify Pods it can acquire, a number of replicas indicating how many Pods it should be maintaining, and a pod template specifying the data of new Pods it should create to meet the number of replicas criteria.

A ReplicaSet then fulfills its purpose by creating and deleting Pods as needed to reach the desired number. When a ReplicaSet needs to create new Pods, it uses its Pod template.

Objectives:

- **Creating ReplicaSet**
- **Scale Up/Down ReplicaSet**
- **Cleanup**

Ensure that you have logged-in as **root** user on **eoc-controller** node

1. Creating Replicaset.

1.1 Let's **view** the yaml manifest file by executing below command

```
# cat -n ~/kubernetes/replicaset.yml
```

Output:

```
[root@eoc-controller ~]# cat -n ~/kubernetes/replicaset.yml
     1  apiVersion: apps/v1
     2  kind: ReplicaSet
     3  metadata:
     4    name: web-server
     5  spec:
     6    replicas: 4
     7    selector:
     8      matchLabels:
     9        tier: web-server
    10    template:
    11      metadata:
    12        labels:
    13          tier: web-server
    14      spec:
    15        containers:
    16        - name: web-server
    17          image: nginx:1.19
```

1.2 Let's **create** ReplicaSet by using the **~/kubernetes/replicaset.yml** file.

```
# kubectl apply -f ~/kubernetes/replicaset.yml
```

Output:

```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/replicaset.yml
replicaset.apps/web-server created
```

1.3 Let's **list** the ReplicaSet by executing the below command.

```
# kubectl get replicaset
```

Output:

```
[root@eoc-controller ~]#kubectl get replicaset
NAME         DESIRED   CURRENT   READY   AGE
web-server   4         4         4       78s
```

1.4 Let's **check** the details of the ReplicaSet by executing the command:

```
# kubectl describe replicaset web-server
```

Output:

```
[root@eoc-controller ~]#kubectl describe replicaset web-server
Name:         web-server
Namespace:    default
Selector:     tier=web-server
Labels:       <none>
Annotations:  <none>
Replicas:     4 current / 4 desired
Pods Status:  4 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  tier=web-server
  Containers:
   web-server:
    Image:        nginx:1.19
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age    From                   Message
  ----    ------            ----   ----                   -------
  Normal  SuccessfulCreate  2m13s  replicaset-controller  Created pod: web-server-vmrhv
  Normal  SuccessfulCreate  2m13s  replicaset-controller  Created pod: web-server-mlbk8
  Normal  SuccessfulCreate  2m13s  replicaset-controller  Created pod: web-server-grjrv
  Normal  SuccessfulCreate  2m13s  replicaset-controller  Created pod: web-server-8jqs2
```

1.5 Let's **list** the resources by executing the below command.

```
# kubectl get all
```

Output:

```
[root@eoc-controller ~]#kubectl get all
NAME                     READY   STATUS    RESTARTS   AGE
pod/web-server-8jqs2     1/1     Running   0          2m41s
pod/web-server-grjrv     1/1     Running   0          2m41s
pod/web-server-mlbk8     1/1     Running   0          2m41s
pod/web-server-vmrhv     1/1     Running   0          2m41s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   30h

NAME                            DESIRED   CURRENT   READY   AGE
replicaset.apps/web-server      4         4         4       2m41s
```

2. Scale up/down ReplicaSet

You can easily change the number of pods and particular ReplicaSet manages in one of 3 ways:

- **a.** Edit the ReplicaSet yaml manifest
- **b.** Edit the controller's configuration by using **kubectl edit rs ReplicaSet_name** and change the replicas count up or down as you desire.
- **c.** Use kubectl directly. For example, **kubectl scale --replicas=2 rs/name**.

2.1 Let's **list** the ReplicaSet by executing the below command.

```
# kubectl get rs web-server
```

Output:

```
[root@eoc-controller ~]#kubectl get rs web-server
NAME         DESIRED   CURRENT   READY   AGE
web-server   4         4         4       3m57s
```

2.2 Let's **verify** the pod labels to understand how RS is managing the replicas by executing the below command.

```
# kubectl get pods --show-labels
```

Output:

```
[root@eoc-controller ~]#kubectl get pods --show-labels
NAME               READY   STATUS    RESTARTS   AGE     LABELS
web-server-8jqs2   1/1     Running   0          4m57s   tier=web-server
web-server-grjrv   1/1     Running   0          4m57s   tier=web-server
web-server-mlbk8   1/1     Running   0          4m57s   tier=web-server
web-server-vmrhv   1/1     Running   0          4m57s   tier=web-server
```

2.3 Let's **scale up** ReplicaSet by editing **yaml manifest** with the help of sed command.

```
# sed -i 's/4/6/g' ~/kubernetes/replicaset.yml
```

2.4 Let's **view** the yaml manifest file by executing below command.

```
# cat -n ~/kubernetes/replicaset.yml
```

Output:

```
[root@eoc-controller ~]#cat -n ~/kubernetes/replicaset.yml
     1  apiVersion: apps/v1
     2  kind: ReplicaSet
     3  metadata:
     4    name: web-server
     5  spec:
     6    replicas: 6
     7    selector:
     8      matchLabels:
     9        tier: web-server
    10    template:
    11      metadata:
    12        labels:
    13          tier: web-server
    14      spec:
    15        containers:
    16        - name: web-server
    17          image: nginx:1.19
```

2.5 Let's **apply** the changes by executing below command.

```
# kubectl apply -f ~/kubernetes/replicaset.yml
```

Output:

```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/replicaset.yml
replicaset.apps/web-server configured
```

2.6 Let's **list** the **ReplicaSet** (**rs**)to see the changes by executing below commnad.

```
# kubectl get rs
```

Output:

```
[root@eoc-controller ~]#kubectl get rs
NAME         DESIRED   CURRENT   READY   AGE
web-server   6         6         6       23m
```

2.7 Let's **edit** the replicas using kubectl edit command

```
# kubectl edit rs web-server
```

Output:

```
[root@eoc-controller ~]#kubectl edit rs web-server
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"apps/v1","kind":"ReplicaSet","metadata":{"annotations":{},"name":"web-
server","namespace":"default"},"spec":{"replicas":6,"selector":{"matchLabels":{"tier":"web-
server"}},"template":{"metadata":{"labels":{"tier":"web-server"}},"spec":{"containers":[{"i
mage":"nginx:1.19","name":"web-server"}]}}}}
  creationTimestamp: "2023-09-05T13:05:17Z"
  generation: 2
  name: web-server
  namespace: default
  resourceVersion: "198109"
  uid: d6c21942-d301-46aa-9da1-0ab98ccb19e1
spec:
  replicas: 8
  selector:
    matchLabels:
      tier: web-server
```

```
[root@eoc-controller ~]#kubectl edit rs web-server
replicaset.apps/web-server edited
```

Note: **After editing the manifest to exit type Esc:wq!**

2.8 Let's **view** the manifest file by executing the below command.

```
# cat -n ~/kubernetes/replicaset.yml
```

Output:

```
[root@eoc-controller ~]#cat -n ~/kubernetes/replicaset.yml
     1  apiVersion: apps/v1
     2  kind: ReplicaSet
     3  metadata:
     4    name: web-server
     5  spec:
     6    replicas: 6
     7    selector:
     8      matchLabels:
     9        tier: web-server
    10    template:
    11      metadata:
    12        labels:
    13          tier: web-server
    14      spec:
    15        containers:
    16        - name: web-server
    17          image: nginx:1.19
```

2.9 Let's **check** the ReplicaSet by executing below command.

```
# kubectl get rs
```

Output:

```
[root@eoc-controller ~]#kubectl get rs
NAME         DESIRED   CURRENT   READY   AGE
web-server   8         8         8       31m
```

2.10 Let's **scale down** the nginx app to **4** replicas by executing below command

```
# kubectl scale rs web-server --replicas=4
```

Output:

```
[root@eoc-controller ~]#kubectl scale rs web-server --replicas=4
replicaset.apps/web-server scaled
```

2.11 Let's **check** the ReplicaSet by executing below command.

```
# kubectl get rs
```

Output:

```
[root@eoc-controller ~]#kubectl get rs
NAME         DESIRED   CURRENT   READY   AGE
web-server   4         4         4       33m
```

2.12 Let's **scale up** the nginx app to **6** replicas by executing below command

```
# kubectl scale rs web-server --replicas=6
```

Output:

```
[root@eoc-controller ~]#kubectl scale rs web-server --replicas=6
replicaset.apps/web-server scaled
```

2.13 Let's **list** the all details by executing the below command.

```
# kubectl get all
```

Output:

```
[root@eoc-controller ~]#kubectl get all
NAME                     READY   STATUS    RESTARTS   AGE
pod/web-server-2mfdc     1/1     Running   0          8m45s
pod/web-server-dlvww     1/1     Running   0          8m45s
pod/web-server-dz62f     1/1     Running   0          8m45s
pod/web-server-fwtxm     1/1     Running   0          8m45s
pod/web-server-txcx5     1/1     Running   0          52s
pod/web-server-wmw9b     1/1     Running   0          52s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   3d

NAME                           DESIRED   CURRENT   READY   AGE
replicaset.apps/web-server     6         6         6       8m45s
```

2.14 Let's **expose** the ReplicaSet on Port 80 by running below command.

```
# kubectl expose replicaset web-server --port=80
```

Output:

```
[root@eoc-controller ~]#kubectl expose replicaset web-server --port=80
service/web-server exposed
```

2.15 Let's **check** the services by executing below command

```
# kubectl get services
```

Output:

```
[root@eoc-controller ~]#kubectl get services
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   31h
web-server   ClusterIP   10.97.115.36   <none>        80/TCP    19s
```

2.16 Let's expose the service-ip.

```
# curl SERVICEIP
```

Note: Replace "**SERVICEIP**" with web-server service actual IP.

Output:

```
[root@eoc-controller ~]#curl 10.97.115.36
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

2.17 Let's **check** the **endpoint(ep)** by executing below command.

```
# kubectl get ep web-server
```

Output:

```
[root@eoc-controller ~]#kubectl get ep web-server
NAME         ENDPOINTS                                         AGE
web-server   10.32.0.2:80,10.32.0.3:80,10.32.0.4:80 + 3 more...   99s
```

3 Cleanup.

3.1 Let's **delete** the ReplicaSet and notice that it deletes the pods as well.

```
# kubectl delete rs web-server
```

Output:

```
[root@eoc-controller ~]#kubectl delete rs web-server
replicaset.apps "web-server" deleted
```

3.2 Let's **delete** the service by executing below command.

```
# kubectl delete service web-server
```

Output:

```
[root@eoc-controller ~]#kubectl delete service web-server
service "web-server" deleted
```