Replica Set ensures how many replica of pod should be running. It can be considered as a replacement of replication controller. The key difference between the replica set and the replication controller is, the replication controller only supports equality-based selector whereas the replica set supports set-based selector.

```
apiVersion: extensions/v1beta1 --------------------->1
kind: ReplicaSet --------------------------> 2
metadata:
   name: Tomcat-ReplicaSet
spec:
   replicas: 3
   selector:
      matchLables:
         tier: Backend ------------------> 3
      matchExpression:
{ key: tier, operation: In, values: [Backend]} --------------> 4
template:
   metadata:
      lables:
         app: Tomcat-ReplicaSet
         tier: Backend
      labels:
         app: App
         component: neo4j
   spec:
      containers:
      - name: Tomcat
      image: tomcat: 8.0
      ports:
      - containerPort: 7474
```

Setup Details

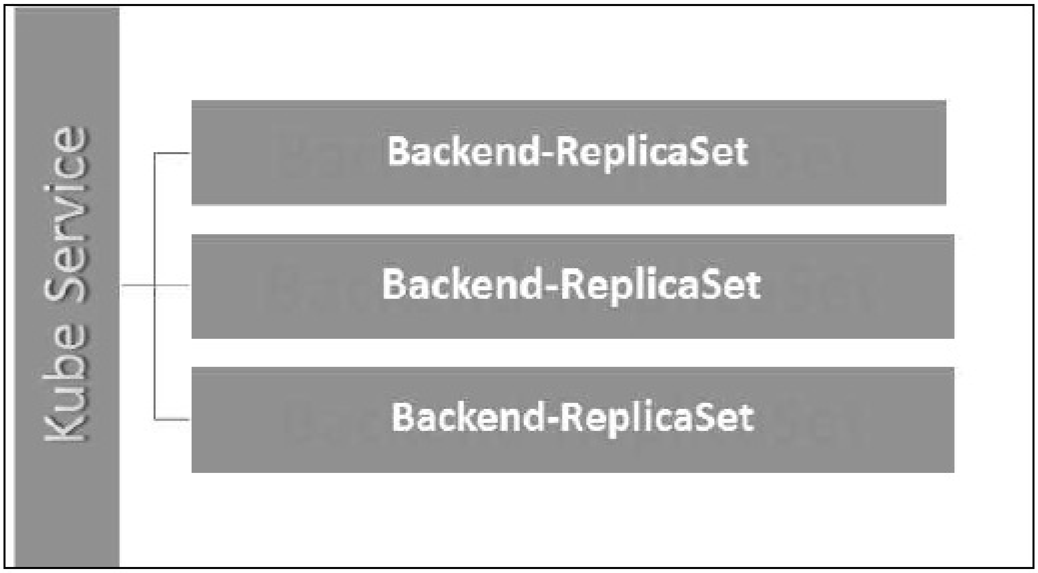
apiVersion: extensions/v1beta1 → In the above code, the API version is the advanced beta version of Kubernetes which supports the concept of replica set.

kind: ReplicaSet → We have defined the kind as the replica set which helps kubectl to understand that the file is used to create a replica set.

tier: Backend → We have defined the label tier as backend which creates a matching selector.

{key: tier, operation: In, values: [Backend]} → This will help **matchExpression** to understand the matching condition we have defined and in the operation which is used by **matchlabel** to find details.

Run the above file using **kubectl** and create the backend replica set with the provided definition in the **yaml** file.


Kube Service
Backend-ReplicaSet
Backend-ReplicaSet
Backend-ReplicaSet