A ReplicaSet (RS) is one of the Kubernetes controllers that ensures a specified number of pod replicas are running at a given time. Without ReplicaSets, we would have to create multiple manifests for the number of pods needed which is very tedious.

In previous versions of Kubernetes, this was called Replication Controller. The main difference between the two is that ReplicaSets allow us to us something called Label Selector. Labels are key value pairs used to specify attributes of objects that are meaningful and useful to users, so keep in mind that It doesn't change the way the core system works. Label Selectors is used to identify a set of objects in Kubernetes.

Creating Manifest for replica Set :

```
#vim nginx-rs.yml
apiVersion : apps/v1
kind : ReplicaSet
metadata :
 name : nginx-rs
spec :
 replicas : 3
 template :
  metadata :
   name : nginx-pod
   labels :
    app : nginx-app
    tier : frontend
  spec :
   containers :
   - name : nginx-container
     image : nginx
     ports :
     - containerPort : 80
 selector :
  matchExpressions :
  - {key: tier,app, operator: In, values: [frontend, nginx-app]}
:wq!
```

#kubectl create -f nginx-rs.yml **(Deploying the Manifest file for creating Replica Set)**
#kubectl get pods -o wide
#kubectl get pods -l app=nginx-app
#kubectl get rs
#kubectl get rs nginx-rs
#kubectl get rs nginx-rs -o wide
#kubectl describe rs nginx-rs

NOTE :

To perform how Replicaset is providing the availability to the pods, Do manually power of your worker node and test Whether Replica Set is recreating the new POD as per the desired number mentioned in the manifest file.

Scale up :
#kubectl scale rs nginx-rs --replicas=7
#kubectl get pods -o wide
#kubectl get rs nginx-rs -o wide

Scale Down :
#kubectl scale rs nginx-rs --replicas=3
#kubectl get pods -o wide
#kubectl get rs nginx-rs -o wide

Cleanup :
#kubectl delete -f nginx-rs.yml
#kubectl get rs
#kubectl get pods -l app=nginx-app