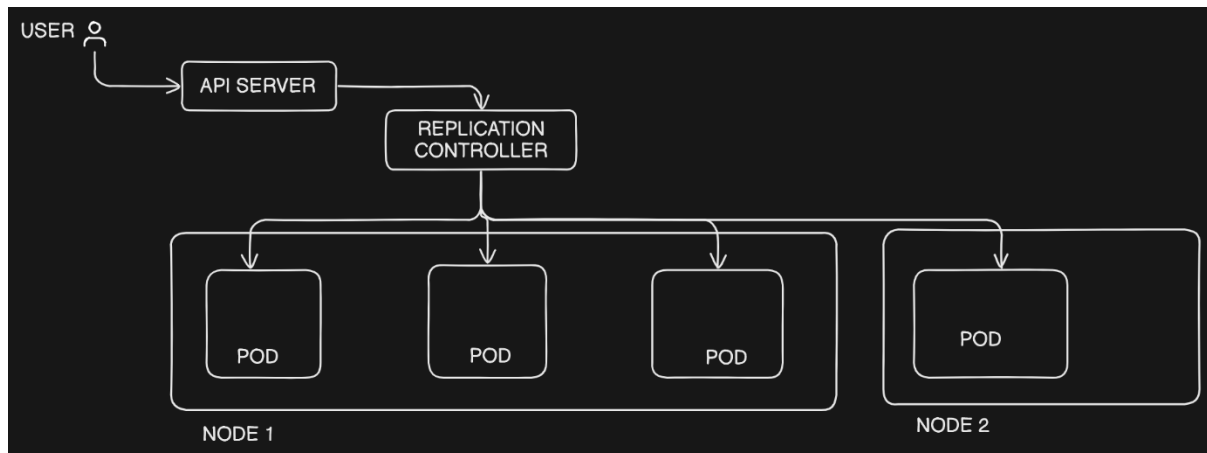


REPLICATION CONTROLLER, REPLICASET, AND DEPLOYMENT

In Kubernetes, **Replication Controller**, **ReplicaSet**, and **Deployment** are all used for ensuring that a certain number of pod replicas are running at any given time.

REPLICATION CONTROLLER (RC)



User is trying to access the API, the request will go to API server first and request will be forwarded to replication controller. This controller will choose the POD to which the request should be sent, basically it will choose the healthy POD.

In case the POD goes down replication controller will create another identical POD before it goes down and send the request to that. Let's a node run out of resource [eg: CPU, memory etc] in that case we can spin new node and re-direct that traffic to new POD at new node.

```
! deployment.yaml ! replicationcontroller.yaml • ! replicaset.yaml
cluster > ! replicationcontroller.yaml
1 #to know ther version ---> kubectl explain rc
2 apiVersion: v1
3 kind: ReplicationController
4 metadata:
5   name: nginx-rc
6   labels:
7     env: demo
8
9 spec:
10  template:
11    metadata:
12      labels:
13        env: demo
14        type: frontend
15    spec:
16      containers:
17        - name: nginx
18          image: nginx
19      replicas: 3
```

REPLICATION CONTROLLER, REPLICASET, AND DEPLOYMENT

```
! deployment.yaml | ! replicationcontroller.yaml | ! replicaset.yaml
cluster > ! replicationcontroller.yaml
1 #to know the version --> kubectl explain rc
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
manoj -->
manoj -->
manoj -->kubectl get pods
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f replicationcontroller.yaml
replicationcontroller/nginx-rc created
manoj -->
manoj -->kubectl get rc
NAME          DESIRED  CURRENT  READY  AGE
nginx-rc      3        3        3      7s
manoj -->
manoj -->kubectl get po
NAME          READY    STATUS    RESTARTS  AGE
nginx-rc-2qz5d 1/1      Running   0          18s
nginx-rc-gtt68 1/1      Running   0          18s
nginx-rc-p8drq 1/1      Running   0          18s
manoj -->
manoj -->
```

```
! deployment.yaml | ! replicationcontroller.yaml | ! replicaset.yaml
cluster > ! replicaset.yaml
1 #to know the version --> kubectl explain rs
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
manoj -->
manoj -->kubectl get rc
NAME          DESIRED  CURRENT  READY  AGE
nginx-rc      3        3        3      2m12s
manoj -->
manoj -->kubectl get all
NAME          READY    STATUS    RESTARTS  AGE
pod/nginx-rc-2qz5d 1/1      Running   0          2m18s
pod/nginx-rc-gtt68 1/1      Running   0          2m18s
pod/nginx-rc-p8drq 1/1      Running   0          2m18s
NAME          DESIRED  CURRENT  READY  AGE
replicationcontroller/nginx-rc 3        3        3      2m18s
NAME          TYPE        CLUSTER-IP  EXTERNAL-IP  PORT(S)  AGE
service/kubernetes ClusterIP  10.96.0.1   <none>       443/TCP  16d
manoj -->
manoj -->
```

← Replication controller

← running on default service

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
manoj -->
manoj -->kubectl get po
NAME          READY    STATUS    RESTARTS  AGE
nginx-rc-2qz5d 1/1      Running   0          4m53s
nginx-rc-gtt68 1/1      Running   0          4m53s
nginx-rc-p8drq 1/1      Running   0          4m53s
manoj -->
manoj -->kubectl delete po nginx-rc-2qz5d
pod "nginx-rc-2qz5d" deleted
manoj -->
manoj -->kubectl get po
NAME          READY    STATUS    RESTARTS  AGE
nginx-rc-66js2 1/1      Running   0          4s
nginx-rc-gtt68 1/1      Running   0          5m15s
nginx-rc-p8drq 1/1      Running   0          5m15s
manoj -->
manoj -->
```

Replication controller only be used to manage the resource that pod are created from that replica controller

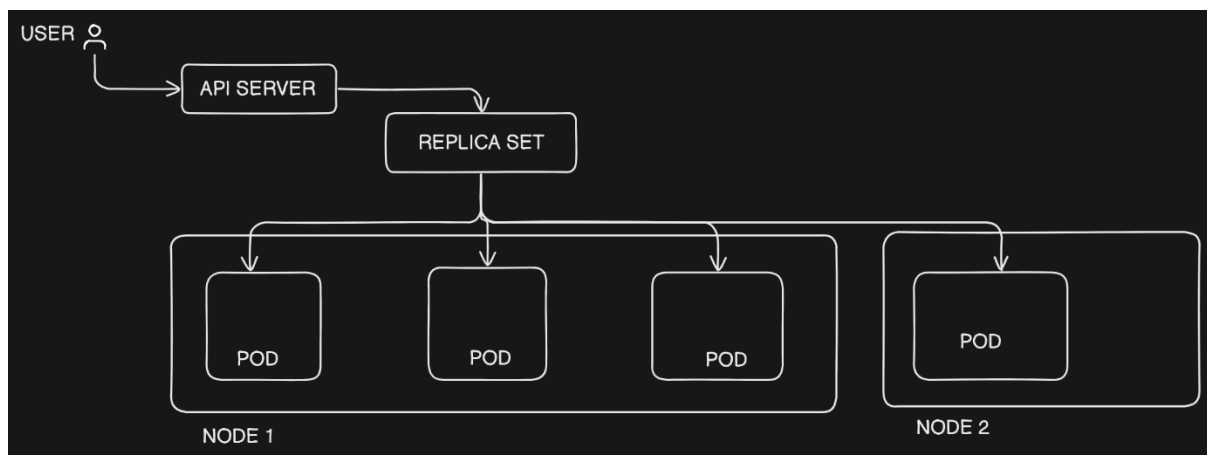
REPLICATION CONTROLLER, REPLICASET, AND DEPLOYMENT

Key points:

- Ensures a specific number of replicas of a pod are running at any time.
- Does not support advanced features like rolling updates or rollbacks.
- Deprecated in favor of ReplicaSet and Deployment but is still available for backward compatibility.

REPLICASET (RS)

ReplicaSet is the successor to the Replication Controller. It has a similar role but with more features and flexibility, like support for set-based label selectors.



```
! deployment.yaml | ! replicationcontroller.yaml | ! replicaset.yaml X
cluster > ! replicaset.yaml
1  #to know the version ---> kubectl explain rs
2  apiVersion: apps/v1
3  kind: ReplicaSet
4  metadata:
5    name: nginx-rs
6    labels:
7      env: demo
8
9  spec:
10   template:
11     metadata:
12       labels:
13         env: demo
14         type: frontend
15     spec:
16       containers:
17       - name: nginx
18         image: nginx
19   replicas: 3
20   selector:
21     matchLabels:
22       env: demo
```

← using "selector" we can manage the existing set as well as that are not part of replicaset

REPLICATION CONTROLLER, REPLICASET, AND DEPLOYMENT

```
manoj -->
manoj -->kubectl get po
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f replicaset.yaml
replicaset.apps/nginx-rs created
manoj -->
manoj -->kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
nginx-rs-lkcw 1/1     Running   0           4s
nginx-rs-w64fg 0/1     ContainerCreating 0           4s
nginx-rs-zjsb6 0/1     ContainerCreating 0           4s
manoj -->
manoj -->kubectl get rs
NAME          DESIRED   CURRENT   READY   AGE
nginx-rs      3          3         3       15s
manoj -->kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx-rs-lkcw 1/1     Running   0           23s
pod/nginx-rs-w64fg 1/1     Running   0           23s
pod/nginx-rs-zjsb6 1/1     Running   0           23s
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/nginx-rs	3	3	3	23s

manoj -->

← Replicaset

```
! deployment.yaml ! replicationcontroller.yaml • ! replicaset.yaml X
cluster > ! replicaset.yaml
1 #to know the version --> kubectl explain rs

manoj -->
manoj -->kubectl describe rs nginx-rs
Name:          nginx-rs
Namespace:     default
Selector:      env=demo
Labels:        env=demo
Annotations:   <none>
Replicas:      3 current / 3 desired
Pods Status:   3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  env=demo
           type=frontend
  Containers:
    nginx:
      Image:          nginx
      Port:           <none>
      Host Port:      <none>
      Environment:    <none>
      Mounts:         <none>
      Volumes:        <none>
      Node-Selectors:  <none>
      Tolerations:    <none>
Events:
  Type     Reason             Age   From               Message
  ----     -
  Normal   SuccessfulCreate    7m23s replicaset-controller Created pod: nginx-rs-zjsb6
  Normal   SuccessfulCreate    7m23s replicaset-controller Created pod: nginx-rs-w64fg
  Normal   SuccessfulCreate    7m23s replicaset-controller Created pod: nginx-rs-lkcw
manoj -->
```

```
manoj -->
manoj -->kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
nginx-rs-lkcw 1/1     Running   0           8m50s
nginx-rs-w64fg 1/1     Running   0           8m50s
nginx-rs-zjsb6 1/1     Running   0           8m50s
manoj -->
manoj -->kubectl scale --replicas=5 rs nginx-rs
replicaset.apps/nginx-rs scaled
manoj -->
manoj -->kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
nginx-rs-klfbv 0/1     ContainerCreating 0           2s
nginx-rs-lkcw 1/1     Running   0           9m48s
nginx-rs-w64fg 1/1     Running   0           9m48s
nginx-rs-xf2kl 0/1     ContainerCreating 0           2s
nginx-rs-zjsb6 1/1     Running   0           9m48s
manoj -->
manoj -->kubectl scale --replicas=2 rs nginx-rs
replicaset.apps/nginx-rs scaled
manoj -->kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
nginx-rs-lkcw 1/1     Running   0           10m
nginx-rs-zjsb6 1/1     Running   0           10m
manoj -->
```

← scale up the replica's using command line. OR
kubectl edit rs <name present in yaml> ..
eg: kubectl edit rs nginx-rs

← scale down the replica's

REPLICATION CONTROLLER, REPLICASET, AND DEPLOYMENT

NOTE:

We will use replica set over replication controller because replication controller only be used to manage the resource that pod are created from that replica controller.

Where as in replica set we can manage the existing set as well that are not part of the replica set with the help of “selector” with the “matchLabels”

Key points:

- Ensures a specific number of pod replicas are running.
- Supports set-based label selectors for more flexible pod matching.
- Primarily used by Deployment to control pod scaling, but can also be used on its own.