Scaling & Replication

Pod does not have ability to recreate the container by itself once it is deleted.

Since Ks8 was designed to orchestrate multiple container and replications.

Need for multiple container/replications help us with these.

Reliability :- To make it reliable we should have atleast 2 pods at a time because if one is down so that others one will be replicated,

By having multiple versions of application you prevent problems if one or more fails.

Load Balancing :- Having multiple versions of a container enables you to easily send traffic to different instances

to prevent overloading of a single instances or node.

As more or more user come then its difficult tohandle by one pod.

So you should have multiple pods.

Rolling Update :- Update to a service by replacing pods one by one.

When you are trying to update old pod with new pod and if you do that at prod env so you make sure it should be done by one

SO these properties are not available with default pods.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Replication Controller:-

It is a Controller which enables you to easily create multiple pods then make sure that number of pods always active.

Replication Controller is a object where in on top of your pod, rather than creating pod we can create one object that has ability to create pod.

It has ability to maintain same number of pod.

If a pod is created using RC then it will be automatically replaced if they does crash, fails, deleted or terminated.

In This way your pod will be keep on running..

So its better to create 2 replica.

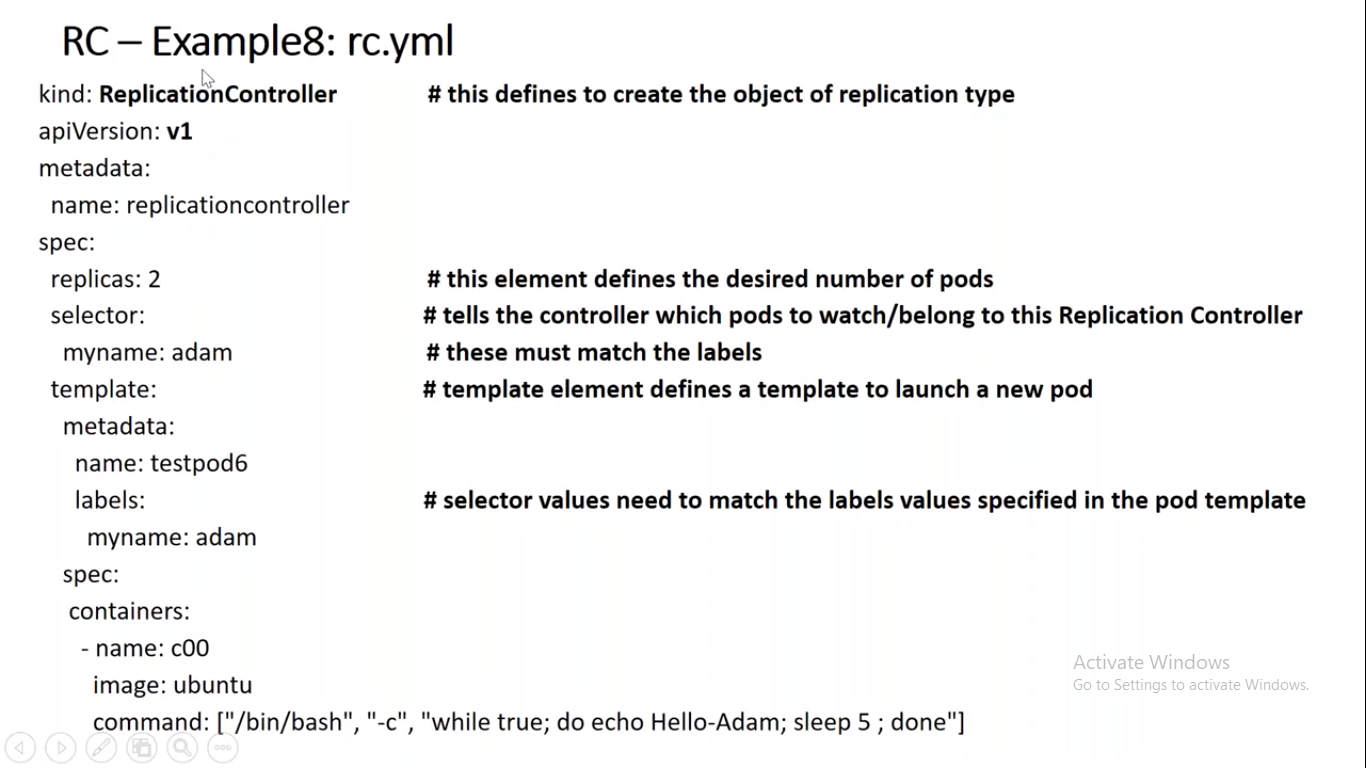
Pod (Label)

Replicas 2 Label

Pod (Label)

Scaling means if you want 50 pods at any time that it well scale up to 50.

Multiple copy of same pod is called as Replica, here we have 2 copy of replica.



Here 2 Pod is created from RC.

Here we have 2 copy of pod.

Selector is a label, find out the pod which has its name is called adam.

Template:- Properties of pod, it is nothing but definition of pod.

Here it should have 2 pod at any time.

Make sure that the name which you defining in your

# Example analysis:-

Here you are not creating pod, you have created replication controller object, which will create internally a pod.

ubuntu@ip-172-31-41-23:~$ kubectl get rc

NAME DESIRED CURRENT READY AGE

myrc 2 2 2 12m

Here two RC object created.

ubuntu@ip-172-31-41-23:~$ kubectl get pod

NAME READY STATUS RESTARTS AGE

myrc-gnn7v 1/1 Running 0 13m

myrc-sqvqk 1/1 Running 0 13m

Here 2 Pod is created from RC.

ubuntu@ip-172-31-41-23:~$ kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-172-31-37-102 Ready <none> 39h v1.18.2

ip-172-31-41-23 Ready master 41h v1.18.2

ip-172-31-45-172 Ready <none> 39h v1.18.2

Here we can see that 2 workers and 1 master is available.

ubuntu@ip-172-31-41-23:~$ kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

myrc-gnn7v 1/1 Running 0 16m 10.244.2.4 ip-172-31-45-172 <none> <none>

myrc-sqvqk 1/1 Running 0 16m 10.244.1.4 ip-172-31-37-102 <none> <none>

Here we can see that one pod is running on 172 worker and another is running on 102 worker.

ubuntu@ip-172-31-41-23:~$ kubectl delete pod myrc-gnn7v

pod "myrc-gnn7v" deleted

ubuntu@ip-172-31-41-23:~$ kubectl get pods

NAME READY STATUS RESTARTS AGE

myrc-jhtws 1/1 Running 0 62s

myrc-sqvqk 1/1 Running 0 21m

Here we can see that we have deleted one pod and it recreated another pod so it proffs that everytime desired state should be 2.

ubuntu@ip-172-31-41-23:~$ kubectl apply -f Rc.yml

replicationcontroller/myrc configured

ubuntu@ip-172-31-41-23:~$ kubectl get rc

NAME DESIRED CURRENT READY AGE

myrc 3 3 3 30m

Here we can scale the replica as well by going to yml file.

ubuntu@ip-172-31-41-23:~$ kubectl scale --replicas=1 rc/myrc

replicationcontroller/myrc scaled

ubuntu@ip-172-31-41-23:~$ kubectl get rc

NAME DESIRED CURRENT READY AGE

myrc 1 1 1 33m

We can do it by using imperative command as well.

ubuntu@ip-172-31-41-23:~$ kubectl get pod --show-labels

NAME READY STATUS RESTARTS AGE LABELS

myrc-sqvqk 1/1 Running 0 35m myname=adam

ubuntu@ip-172-31-41-23:~$ kubectl scale --replicas=2 rc -l myname=adam

replicationcontroller/myrc scaled

Here we can see that we can scale using label name as well.

ubuntu@ip-172-31-41-23:~$ kubectl delete -f Rc.yml

replicationcontroller "myrc" deleted

Here if we delete the object itself then nothing will be available.

we can delete it via using yml file, or via imperative command as well i.e rc/myrc or rc myrc

Commands:-

