Day 4:

Replication Controller

Before Starting Todays Session, let's start with starting minikube by going to that directory.
Minikube start
Minikube status

```
c:\Program Files\Kubernetes\Minikube>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
timeToStop: Nonexistent
```

But What is RC?

To relaunch the deleted Pod, we need a Replication Controller for monitoring it.
Let's Check for pods available
Kubectl get pods

```
c:\Program Files\Kubernetes\Minikube>kubectl get pods
No resources found in default namespace.

c:\Program Files\Kubernetes\Minikube>
```

Now, first we will launch POD, for that we need that pod.yml(day3 of K8's).
With some changes in it- (labels : app=web) and it will be exactly like

pod.yml - Notepad

File Edit Format View Help

```
apiVersion: v1
kind: Pod
metadata:
  name: "mypod1"
  labels:
    app: web

spec:
  containers:
    - name: "myc1"
      image: "vimal13/apache-webserver-php"
```

Launch it with kubectl create -f pod.yml

Then kubectl get pods
Later kubectl get pods -L app which will tell the app name called web.

```
C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl create -f pod.yml
pod/mypod1 created

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mypod1    1/1     Running   0          13s

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get pods -L app
NAME      READY   STATUS    RESTARTS   AGE   APP
mypod1    1/1     Running   0          21s   web
```

Now The role of RC comes in play we need a RC file so created -Rc.yml

rc.yml - Notepad

File Edit Format View Help

```
apiVersion : v1
kind : ReplicationController
metadata:
  name : myrc1

spec:
  selector:
    app: web

  template:
    metadata:
      name: "abhispod2"
      labels:
        app: web

    spec:
      containers:
      - name: "myc1"
        image: "vimal13/apache-webserver-php"
```

Now to run this rc file we need to run command

Kubectl create -f rc.yml

```
C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl create -f rc.yml
replicationcontroller/myrc1 created

C:\Users\Abhishek kumar\Desktop\temp\k8>
```

Now, we will run kubectl get rc
And after that exposing it to port 80 by type NodeNort
Kubectl expose rc myrc1 --port=80 --type=NodePort
After that lets get services as it is Node Port
Kubectl get services

```
C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl create -f rc.yml
replicationcontroller/myrc1 created

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get rc
NAME     DESIRED   CURRENT   READY   AGE
myrc1    1         1         1       3m31s

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl expose rc myrc1 --port=80 --type=NodePort
service/myrc1 exposed

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get services
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        3d
myrc1        NodePort    10.103.194.133   <none>        80:32494/TCP   14s
```

Here in service, only 1 pod is there. If we create a replica of that it will automatically create a replica, if one down then another one will be created instantly by setting
DESIRED STATE==CURRENT STATE
Lets set replica=5, 1 is already there, 4 more will be created, so make changes in rc.yml file

rc.yml - Notepad

File Edit Format View Help

```
apiVersion : v1
kind : ReplicationController
metadata:
  name : myrc1


spec:
  replica: 5
  selector:
    app: web
```

Use Apply not create again as rc.yml is already created

Kubect apply -f rc.yml

```
C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl apply -f rc.yml
Warning: resource replicationcontrollers/myrc1 is missing the kubectl.kubernetes.io/last-applied-c
onfiguration annotation which is required by kubectl apply. kubectl apply should only be used on r
esources created declaratively by either kubectl create --save-config or kubectl apply. The missin
g annotation will be patched automatically.
replicationcontroller/myrc1 configured
```

Now, let's check for the status of the services

Kubectl get rc

Kubectl get pods -L app

Kubectl get service

```
C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get rc
NAME     DESIRED   CURRENT   READY   AGE
myrc1    5         5         5       13m

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get pods -L app
NAME          READY   STATUS    RESTARTS   AGE     APP
mypod1        1/1     Running   0          17m     web
myrc1-54tmc   1/1     Running   0          4m52s   web
myrc1-lt6t6   1/1     Running   0          4m52s   web
myrc1-nkrrl   1/1     Running   0          4m52s   web
myrc1-zv8fl   1/1     Running   0          4m52s   web

C:\Users\Abhishek kumar\Desktop\temp\k8>kubectl get service
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        3d
myrc1        NodePort    10.103.194.133   <none>        80:32494/TCP   10m
```

See, i have highlighted the port with that port number and with minikube IP you can view the webpage hosted in it. So let's use this

https://minikubeip:thisport

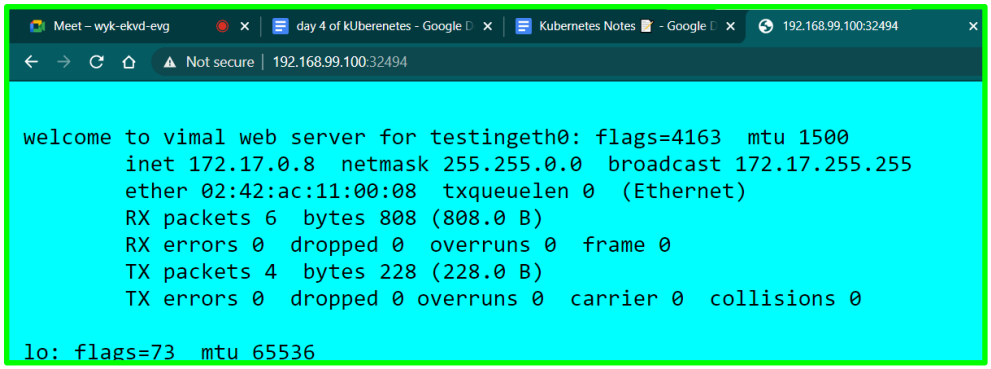
Using ip of minikube i.e

Ifconfig | less to find that in VM with username:docker pass:tcuser

```
          RX packets:2259 errors:0 dr
          TX packets:1485 errors:0 dr
          collisions:0 txqueuelen:100
          RX bytes:427024 (417.0 KiB)

th1       Link encap:Ethernet  HWaddr
          inet addr:192.168.99.100  B
          UP BROADCAST RUNNING MULTIC
          RX packets:2570 errors:0 dr
          TX packets:1401 errors:0 dr
          collisions:0 txqueuelen:100
```

So my link will be for client:- https:192.168.99.100:32494



Now, lets's refresh again, the IP will change as it is balanced by services and act as magic.