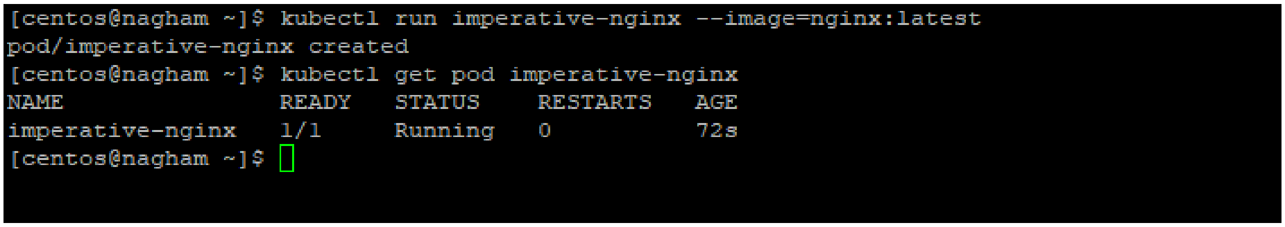
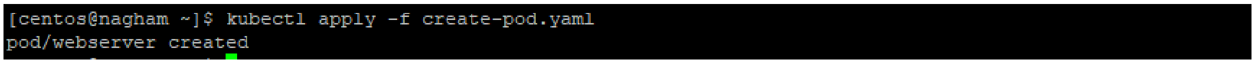
1- Create a pod with the name “imperative-nginx” and with the image nginx and latest tag. using Imperative command (not yaml).

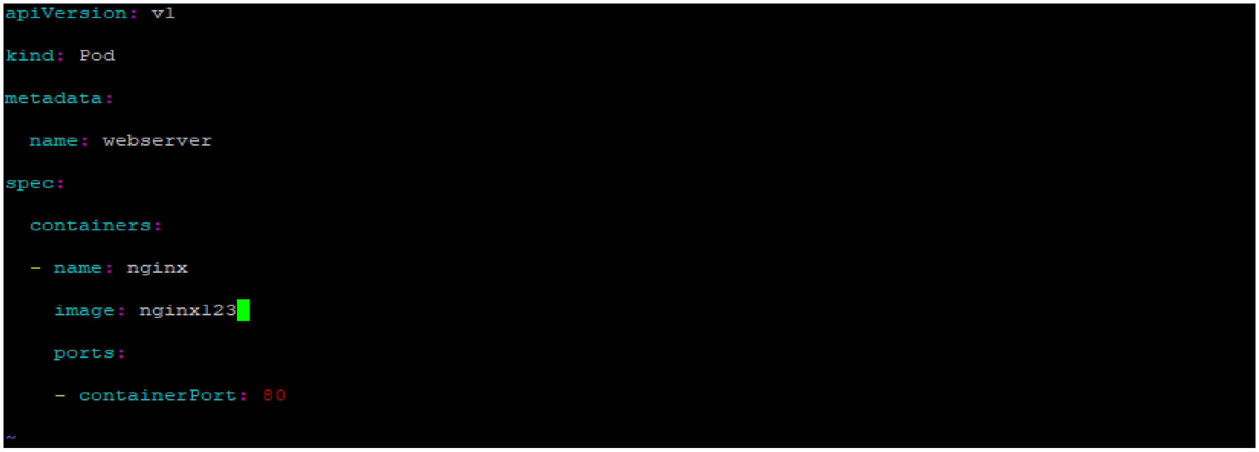


```
[centos@nagham ~]$ kubectl run imperative-nginx --image=nginx:latest
pod/imperative-nginx created
[centos@nagham ~]$ kubectl get pod imperative-nginx
NAME               READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          72s
[centos@nagham ~]$
```

2- Create a pod with the name webserver and with the image “nginx123”Use a pod-definition YAML file.

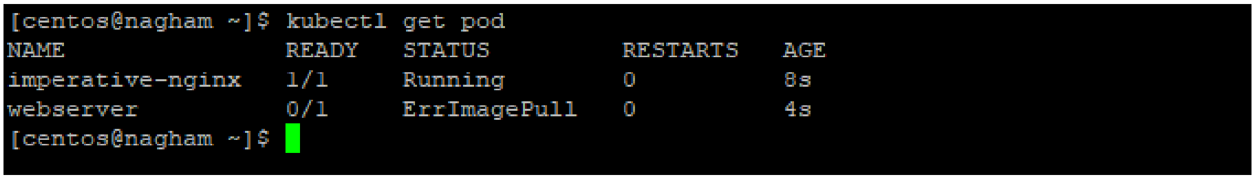


```
[centos@nagham ~]$ kubectl apply -f create-pod.yaml
pod/webserver created
```



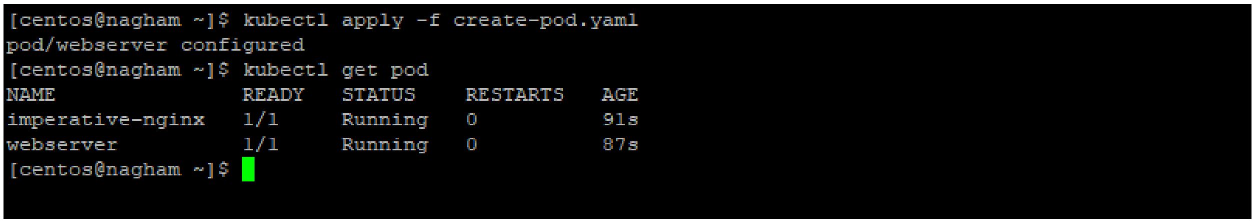
```
apiVersion: v1

kind: Pod

metadata:

  name: webserver

spec:

  containers:

  - name: nginx

    image: nginx123

    ports:

    - containerPort: 80

~
```

3- What is the nginx pod status?



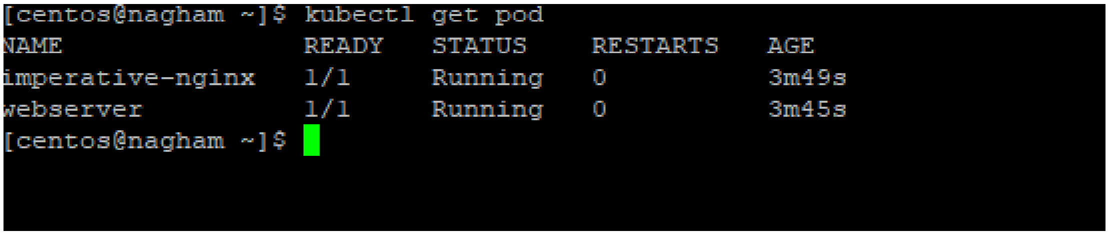
```
[centos@nagham ~]$ kubectl get pod
NAME               READY   STATUS         RESTARTS   AGE
imperative-nginx   1/1     Running        0          8s
webserver          0/1     ErrImagePull   0          4s
[centos@nagham ~]$
```

4- Change the nginx pod image to “nginx” check the status again



```
[centos@nagham ~]$ kubectl apply -f create-pod.yaml
pod/webserver configured
[centos@nagham ~]$ kubectl get pod
NAME               READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          91s
webserver          1/1     Running   0          87s
[centos@nagham ~]$
```

5- How many pods are running in the system? Type the command to show this

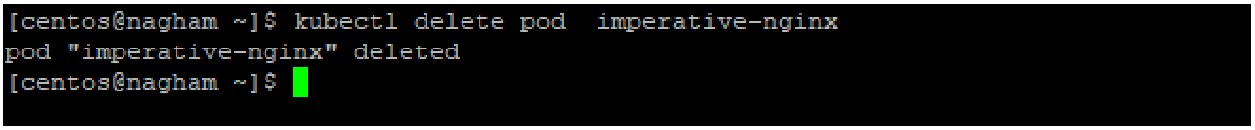


```
[centos@nagham ~]$ kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
imperative-nginx    1/1     Running   0          3m49s
webserver           1/1     Running   0          3m45s
[centos@nagham ~]$
```

6- What does READY column in the output of get pods command indicate?

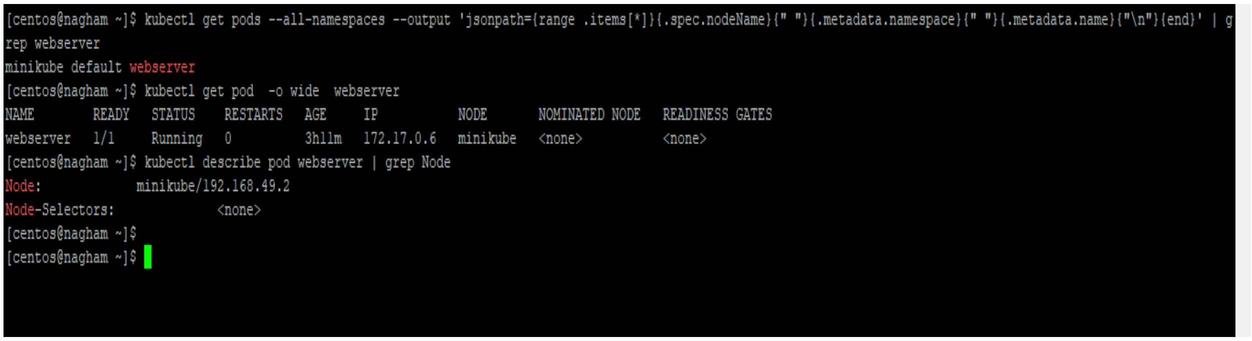
It shows how many containers in a pod are considered ready

7- Delete first pod named imperative-nginx you just created. Type the command to do this



```
[centos@nagham ~]$ kubectl delete pod  imperative-nginx
pod "imperative-nginx" deleted
[centos@nagham ~]$
```

8- Which node is pod named webserver running on (list two commands to do this)

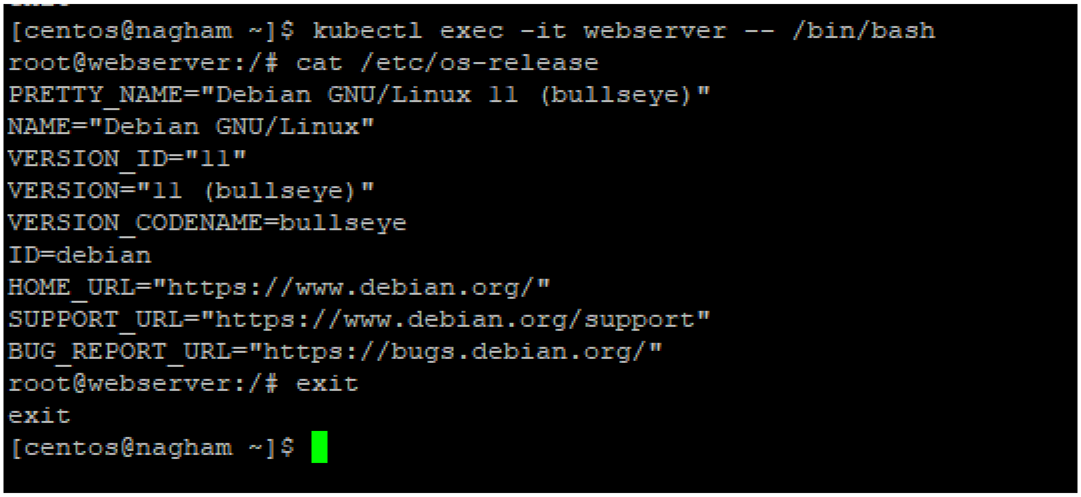


```
[centos@nagham ~]$ kubectl get pods --all-namespaces --output 'jsonpath={range .items[*]}{.spec.nodeName}{" "}{.metadata.namespace}{" "}{.metadata.name}{"\n"}{end}' | g
rep webserver
minikube default webserver
[centos@nagham ~]$ kubectl get pod  -o wide  webserver
NAME        READY   STATUS    RESTARTS   AGE     IP           NODE       NOMINATED NODE   READINESS GATES
webserver   1/1     Running   0          3h11m   172.17.0.6   minikube   <none>           <none>
[centos@nagham ~]$ kubectl describe pod webserver | grep Node
Node:             minikube/192.168.49.2
Node-Selectors:              <none>
[centos@nagham ~]$
[centos@nagham ~]$
```

9- Get a shell to the running container i.e ssh into it (figure out the command)

10- Run cat /etc/os-release inside the container

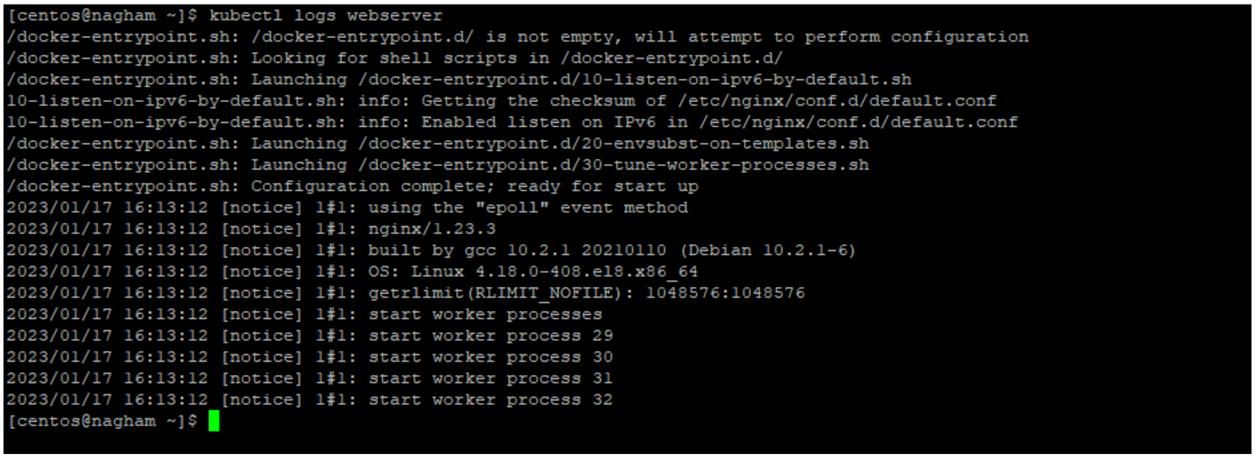
11- 11- Exit from the shell (/bin/bash) session



```
[centos@nagham ~]$ kubectl exec -it webserver -- /bin/bash
root@webserver:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@webserver:/# exit
exit
[centos@nagham ~]$
```

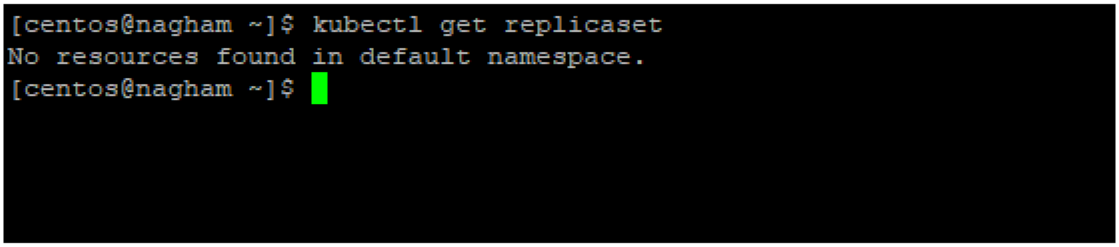
12- Get logs of pod, what are logs and what they are used for?

Kubernetes captures logs from each container in a running Pod and It is used to view container logs for debugging



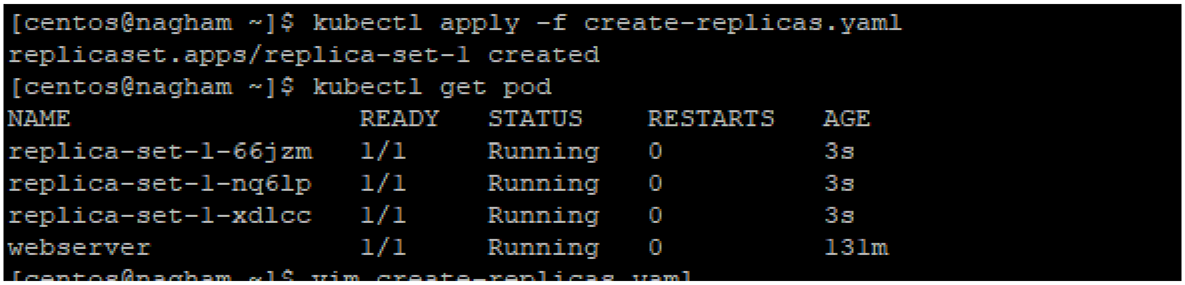
```
[centos@nagham ~]$ kubectl logs webserver
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/01/17 16:13:12 [notice] 1#1: using the "epoll" event method
2023/01/17 16:13:12 [notice] 1#1: nginx/1.23.3
2023/01/17 16:13:12 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/17 16:13:12 [notice] 1#1: OS: Linux 4.18.0-408.el8.x86_64
2023/01/17 16:13:12 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/17 16:13:12 [notice] 1#1: start worker processes
2023/01/17 16:13:12 [notice] 1#1: start worker process 29
2023/01/17 16:13:12 [notice] 1#1: start worker process 30
2023/01/17 16:13:12 [notice] 1#1: start worker process 31
2023/01/17 16:13:12 [notice] 1#1: start worker process 32
[centos@nagham ~]$
```

13- How many ReplicaSets exist on the system?

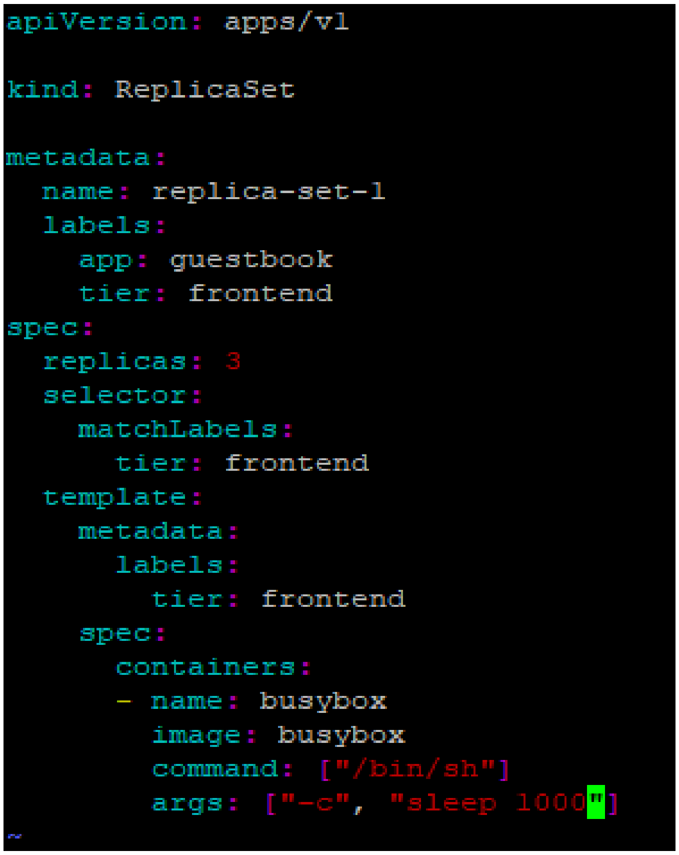


```
[centos@nagham ~]$ kubectl get replicaset
No resources found in default namespace.
[centos@nagham ~]$
```

14- create a ReplicaSet withname= replica-set-1 image= busybox replicas= 3



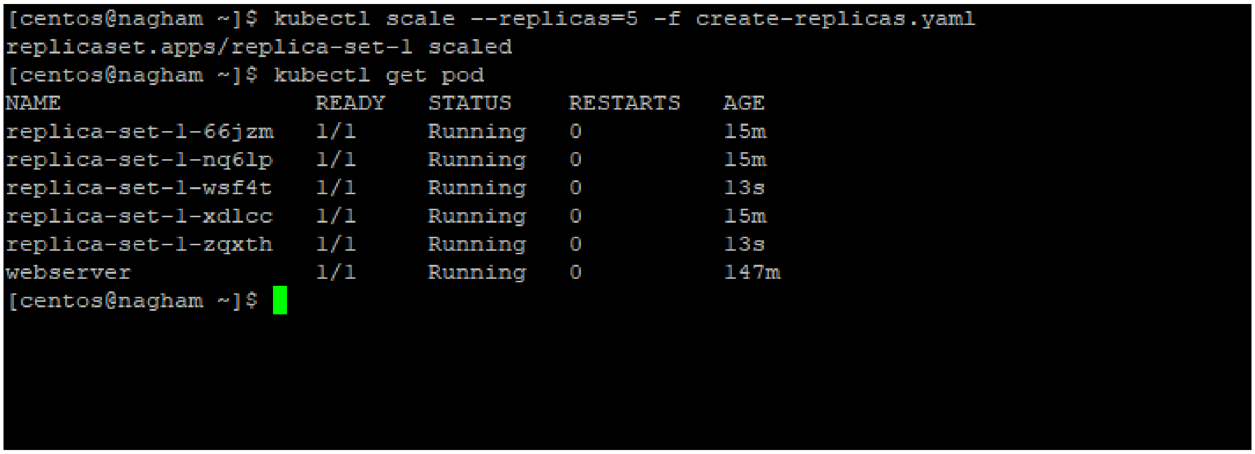
```
[centos@nagham ~]$ kubectl apply -f create-replicas.yaml
replicaset.apps/replica-set-1 created
[centos@nagham ~]$ kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
replica-set-1-66jzm   1/1     Running   0          3s
replica-set-1-nq6lp   1/1     Running   0          3s
replica-set-1-xdlcc   1/1     Running   0          3s
webserver             1/1     Running   0          131m
[centos@nagham ~]$ vim create-replicas.yaml
```



```
apiVersion: apps/v1

kind: ReplicaSet

metadata:
  name: replica-set-1
  labels:
    app: guestbook
    tier: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: busybox
        image: busybox
        command: ["/bin/sh"]
        args: ["-c", "sleep 1000"]
~
```

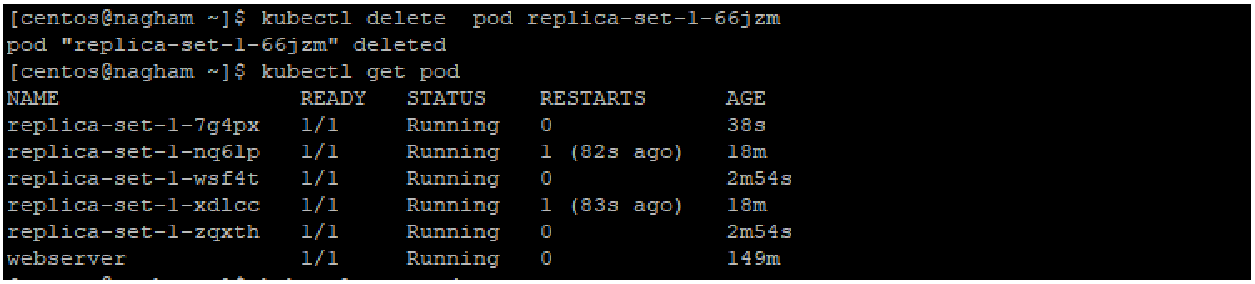
15- Scale the ReplicaSet replica-set-1 to 5 PODs.

16- 16- How many PODs are READY in the replica-set-1?



```
[centos@nagham ~]$ kubectl scale --replicas=5 -f create-replicas.yaml
replicaset.apps/replica-set-1 scaled
[centos@nagham ~]$ kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
replica-set-1-66jzm   1/1     Running   0          15m
replica-set-1-nq6lp   1/1     Running   0          15m
replica-set-1-wsf4t   1/1     Running   0          13s
replica-set-1-xdlcc   1/1     Running   0          15m
replica-set-1-zqxth   1/1     Running   0          13s
webserver             1/1     Running   0          147m
[centos@nagham ~]$
```

17- Delete any one of the 5 PODs then check How many PODs exist now? Why are there still 5 PODs, even after you deleted one?



```
[centos@nagham ~]$ kubectl delete  pod replica-set-1-66jzm
pod "replica-set-1-66jzm" deleted
[centos@nagham ~]$ kubectl get pod
NAME                  READY   STATUS    RESTARTS      AGE
replica-set-1-7g4px   1/1     Running   0             38s
replica-set-1-nq6lp   1/1     Running   1 (82s ago)   18m
replica-set-1-wsf4t   1/1     Running   0             2m54s
replica-set-1-xdlcc   1/1     Running   1 (83s ago)   18m
replica-set-1-zqxth   1/1     Running   0             2m54s
webserver             1/1     Running   0             149m
```