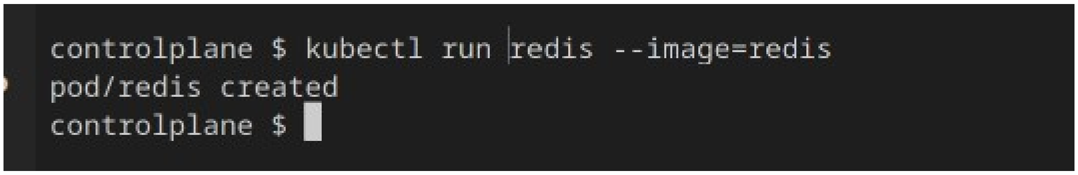
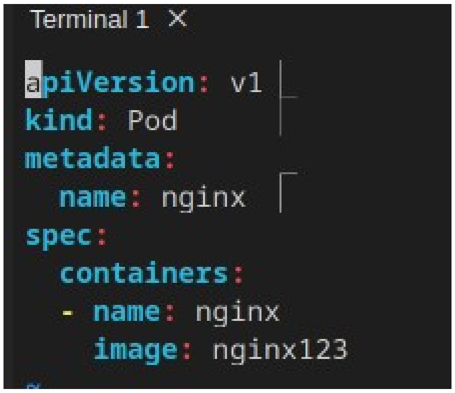
1- Install k8s cluster (minikube)
using killercoda

2: Create a pod with the name redis and with the image redis



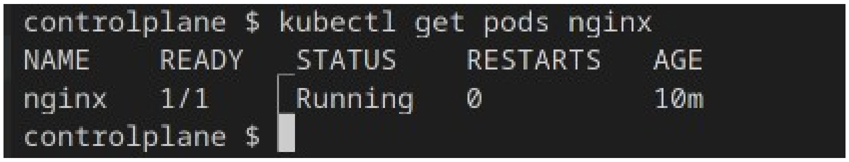
```
controlplane $ kubectl run redis --image=redis
pod/redis created
controlplane $ 
```

3- Create a pod with the name nginx and with the image “nginx123”



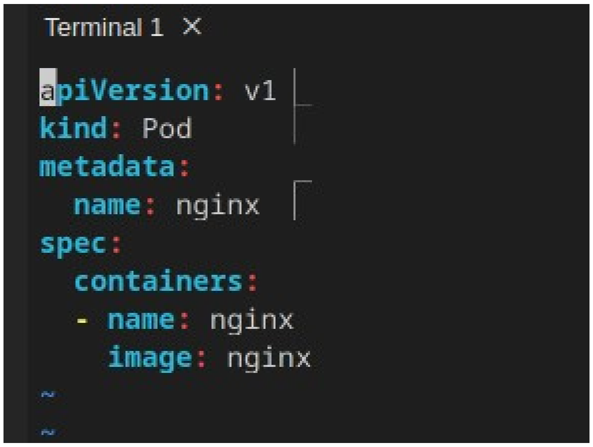
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx123
```

4: What is the nginx pod status?   ‘

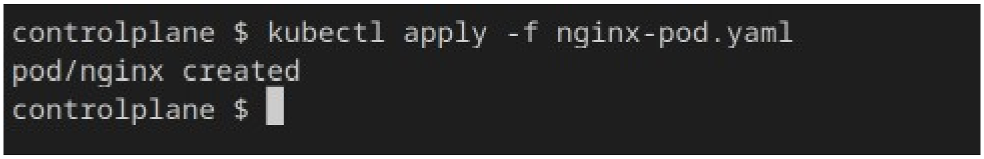


```
controlplane $ kubectl get pods nginx
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          10m
controlplane $ 
```

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

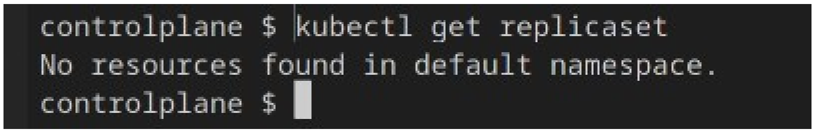


```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
~
~
```



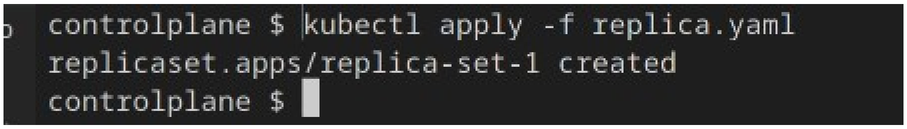
```
controlplane $ kubectl apply -f nginx-pod.yaml
pod/nginx created
controlplane $
```

6- How many ReplicaSets exist on the system?



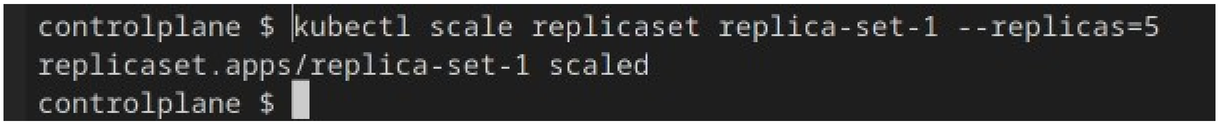
```
controlplane $ kubectl get replicaset
No resources found in default namespace.
controlplane $
```

7- create a ReplicaSet with
name= replica-set-1
image= busybox
replicas= 3



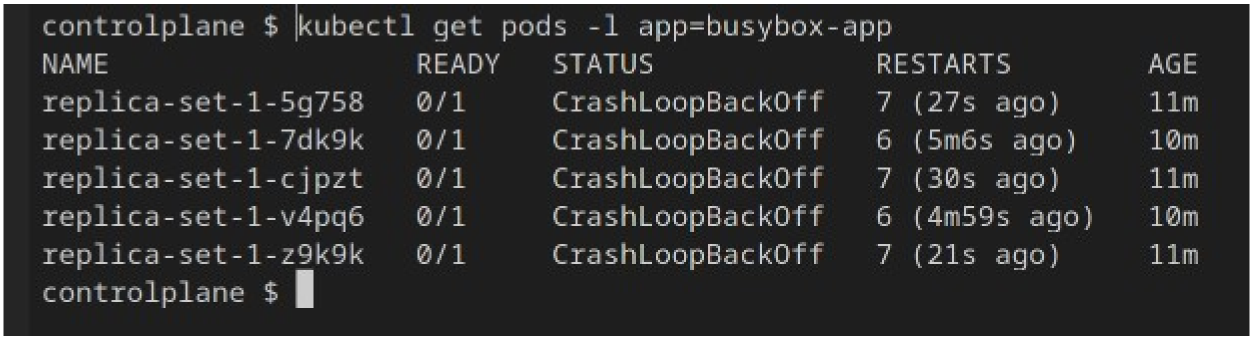
```
controlplane $ kubectl apply -f replica.yaml
replicaset.apps/replica-set-1 created
controlplane $
```

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



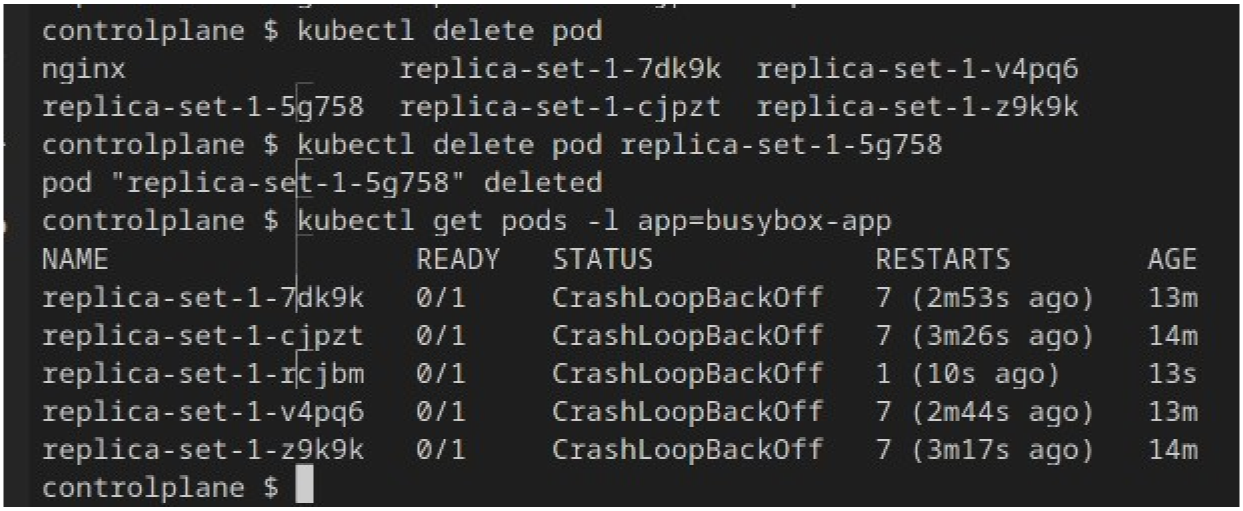
```
controlplane $ kubectl scale replicaset replica-set-1 --replicas=5
replicaset.apps/replica-set-1 scaled
controlplane $
```

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



```
controlplane $ kubectl get pods -l app=busybox-app
NAME                  READY   STATUS             RESTARTS        AGE
replica-set-1-5g758   0/1     CrashLoopBackOff   7 (27s ago)     11m
replica-set-1-7dk9k   0/1     CrashLoopBackOff   6 (5m6s ago)    10m
replica-set-1-cjpzt   0/1     CrashLoopBackOff   7 (30s ago)     11m
replica-set-1-v4pq6   0/1     CrashLoopBackOff   6 (4m59s ago)   10m
replica-set-1-z9k9k   0/1     CrashLoopBackOff   7 (21s ago)     11m
controlplane $
```

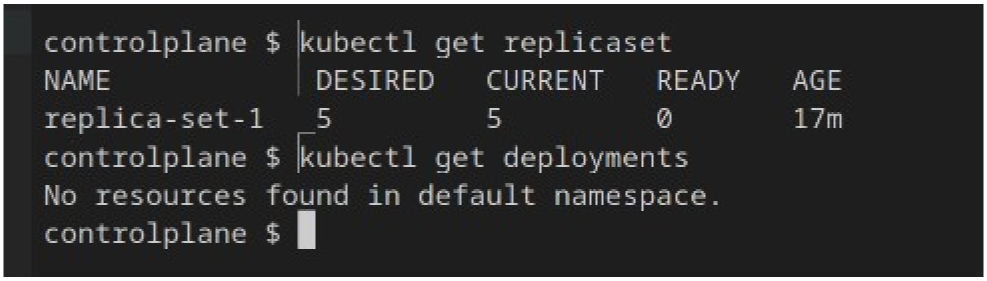
10- Delete any one of the 5 PODs then check How many PODs exist now?



```
controlplane $ kubectl delete pod
nginx                   replica-set-1-7dk9k  replica-set-1-v4pq6
replica-set-1-5g758  replica-set-1-cjpzt  replica-set-1-z9k9k
controlplane $ kubectl delete pod replica-set-1-5g758
pod "replica-set-1-5g758" deleted
controlplane $ kubectl get pods -l app=busybox-app
NAME                 READY   STATUS             RESTARTS        AGE
replica-set-1-7dk9k  0/1     CrashLoopBackOff   7 (2m53s ago)   13m
replica-set-1-cjpzt  0/1     CrashLoopBackOff   7 (3m26s ago)   14m
replica-set-1-rcjbm  0/1     CrashLoopBackOff   1 (10s ago)     13s
replica-set-1-v4pq6  0/1     CrashLoopBackOff   7 (2m44s ago)   13m
replica-set-1-z9k9k  0/1     CrashLoopBackOff   7 (3m17s ago)   14m
controlplane $ 
```

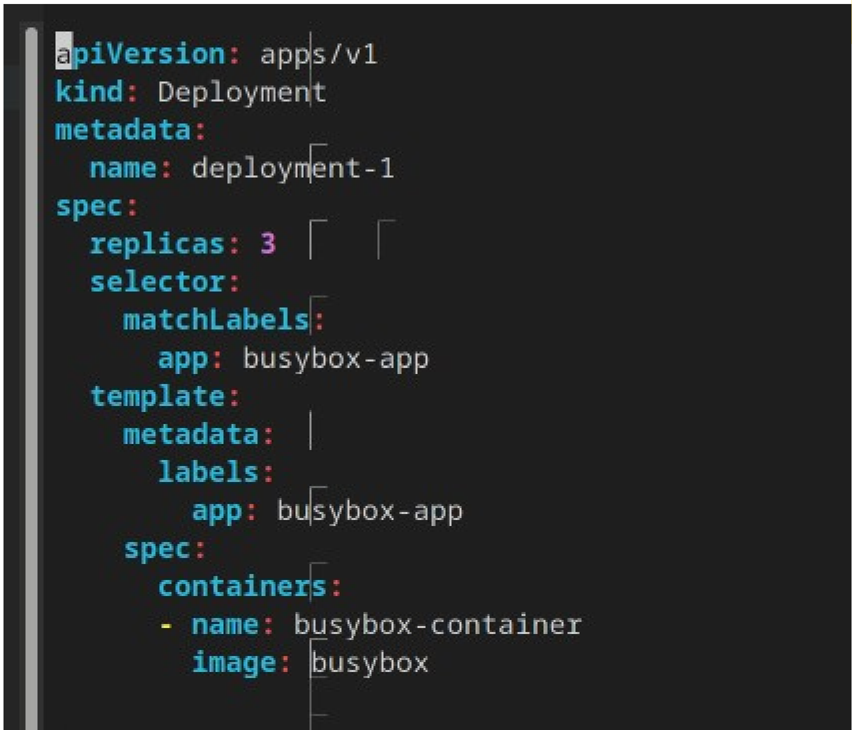
Why are there still 5 PODs, even after you deleted one?
ReplicaSet controller immediately notices that the pod has been terminated and takes action to maintain the desired number of replicas specified in its configuration.

11- How many Deployments and ReplicaSets exist on the system?

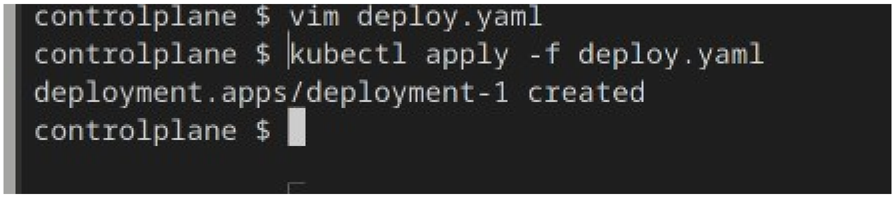


```
controlplane $ kubectl get replicaset
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         0       17m
controlplane $ kubectl get deployments
No resources found in default namespace.
controlplane $ 
```

12- create a Deployment with
name= deployment-1
image= busybox
replicas= 3

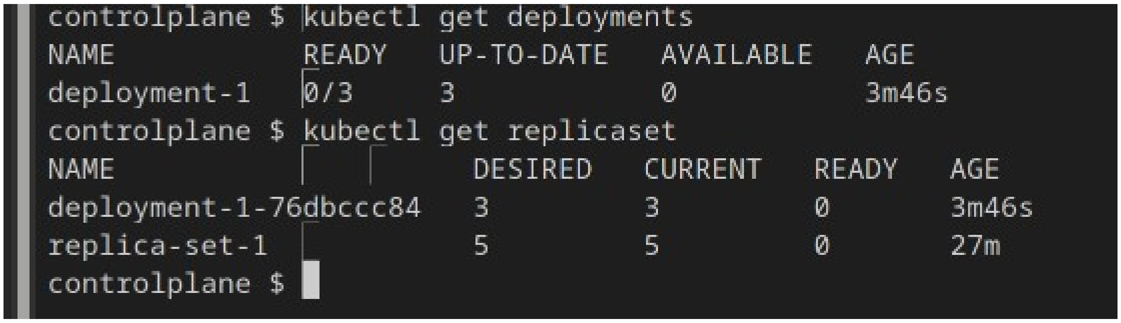


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox-app
  template:
    metadata:
      labels:
        app: busybox-app
    spec:
      containers:
      - name: busybox-container
        image: busybox
```



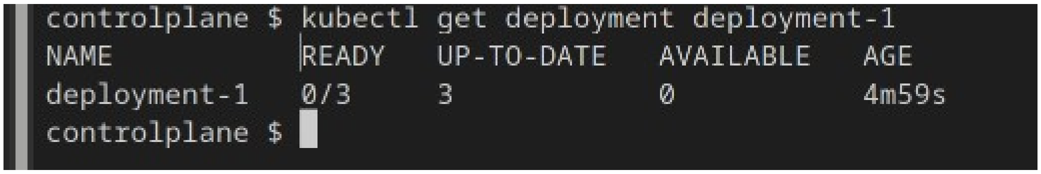
```
controlplane $ vim deploy.yaml
controlplane $ kubectl apply -f deploy.yaml
deployment.apps/deployment-1 created
controlplane $ 
```

13- How many Deployments and ReplicaSets exist on the system now?



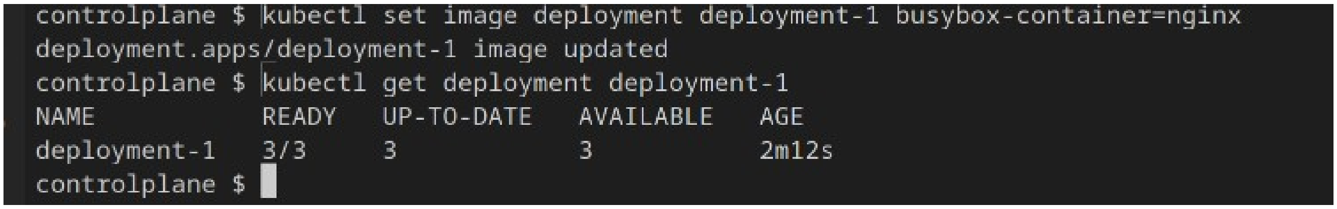
```
controlplane $ kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   0/3     3            0           3m46s
controlplane $ kubectl get replicaset
NAME                      DESIRED   CURRENT   READY   AGE
deployment-1-76dbccc84    3         3         0       3m46s
replica-set-1             5         5         0       27m
controlplane $ 
```

14- How many pods are ready with the deployment-1?



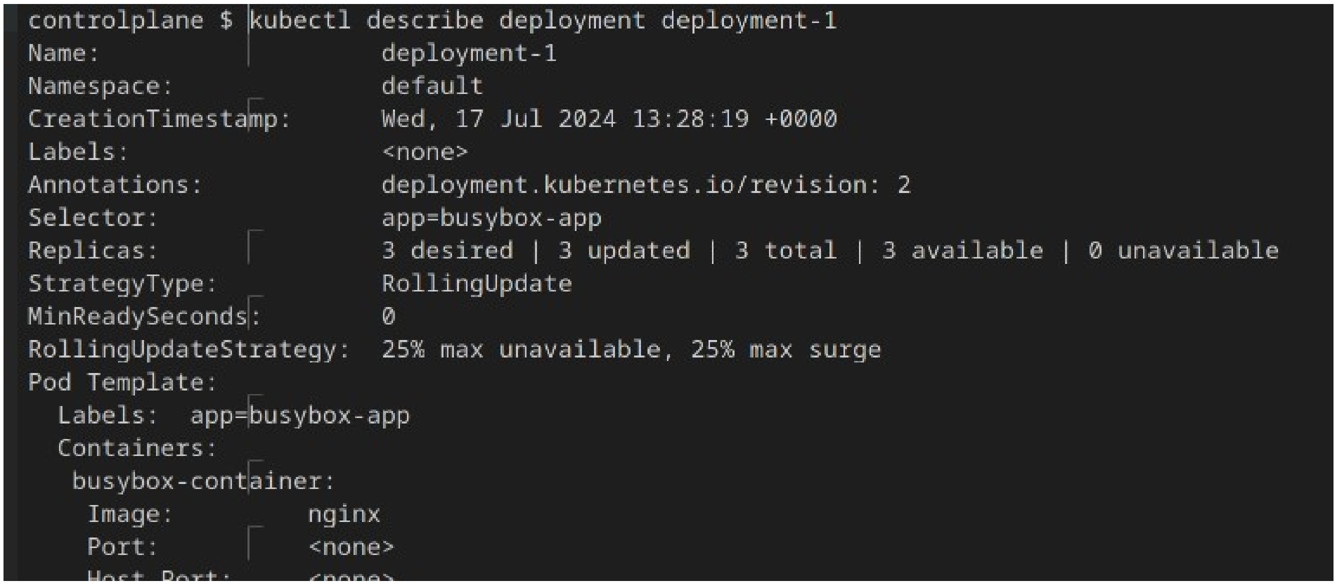
```
controlplane $ kubectl get deployment deployment-1
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   0/3     3            0           4m59s
controlplane $ 
```

15- Update deployment-1 image to nginx then check the ready pods again



```
controlplane $ kubectl set image deployment deployment-1 busybox-container=nginx
deployment.apps/deployment-1 image updated
controlplane $ kubectl get deployment deployment-1
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           2m12s
controlplane $ 
```

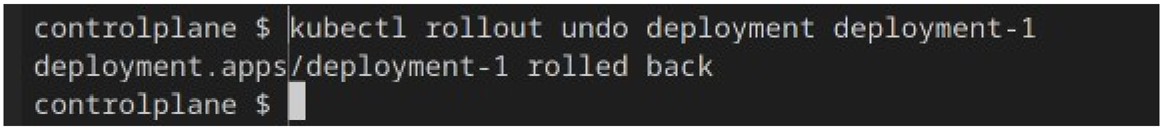
16- Run kubectl describe deployment deployment-1 and check events



```
controlplane $ kubectl describe deployment deployment-1
Name:                   deployment-1
Namespace:              default
CreationTimestamp:      Wed, 17 Jul 2024 13:28:19 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=busybox-app
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=busybox-app
  Containers:
   busybox-container:
    Image:         nginx
    Port:          <none>
    Host Port:     <none>
```

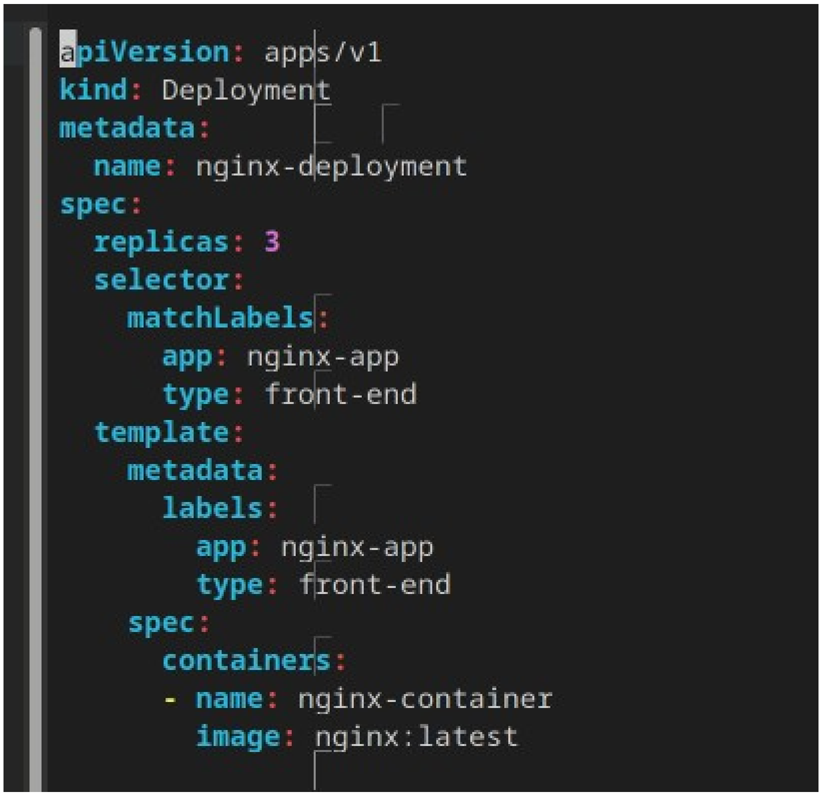
What is the deployment strategy used to upgrade the deployment-1?
RollingUpdate

17- Rollback the deployment-1 What is the used image with the deployment-1?

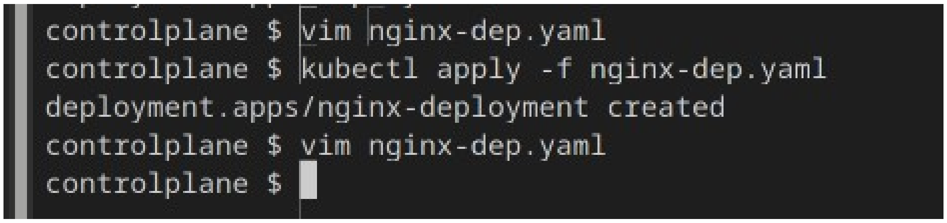


```
controlplane $ kubectl rollout undo deployment deployment-1
deployment.apps/deployment-1 rolled back
controlplane $ 
```

18- Create a deployment using nginx image with latest tag only and remember to mention tag i.e nginx:latest and name it as nginx-deployment. App labels should be app: nginx-app and type: front-end. The container should be named as nginx-container; also make sure replica counts are 3.



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-app
      type: front-end
  template:
    metadata:
      labels:
        app: nginx-app
        type: front-end
    spec:
      containers:
      - name: nginx-container
        image: nginx:latest
```



```
controlplane $ vim nginx-dep.yaml
controlplane $ kubectl apply -f nginx-dep.yaml
deployment.apps/nginx-deployment created
controlplane $ vim nginx-dep.yaml
controlplane $ 
```