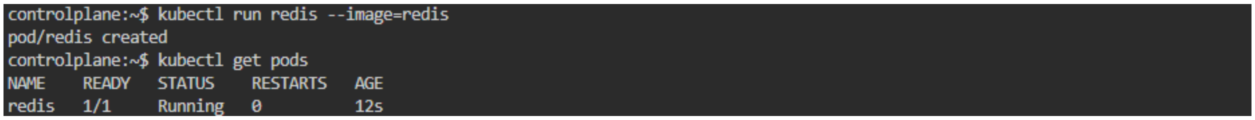
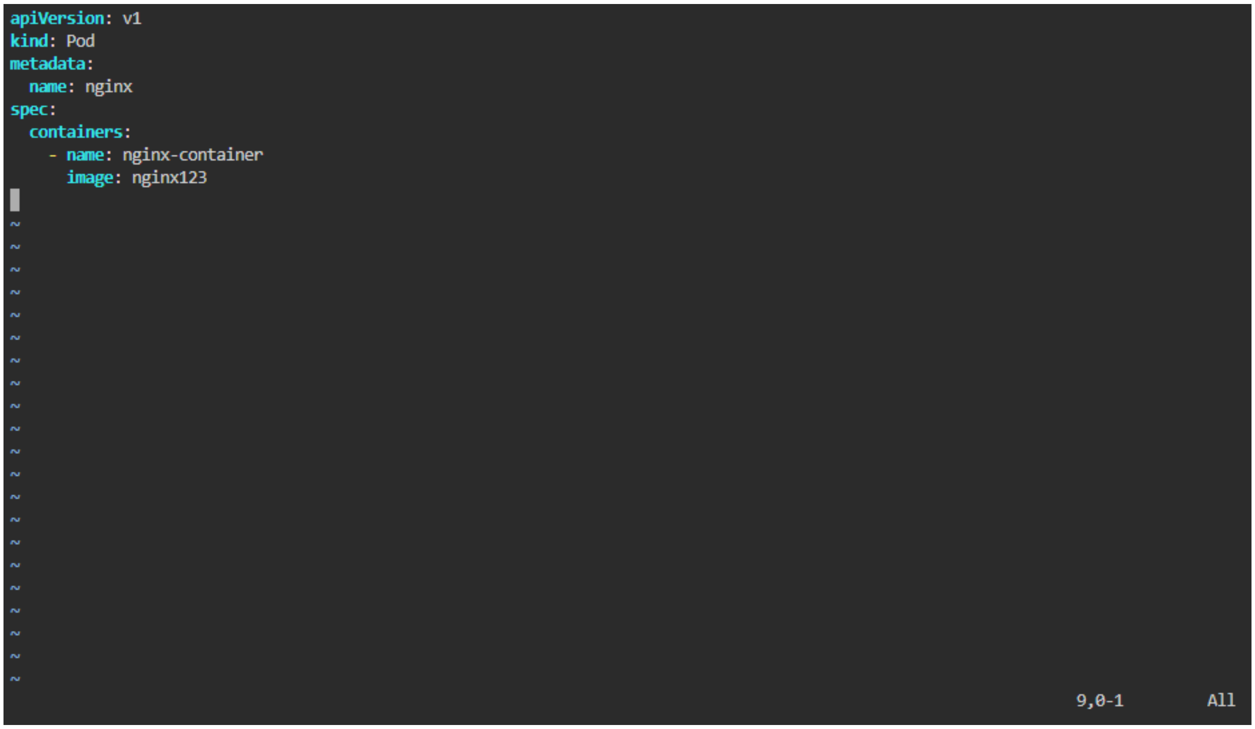
Create a pod with the name `redis` and with the image `redis`.



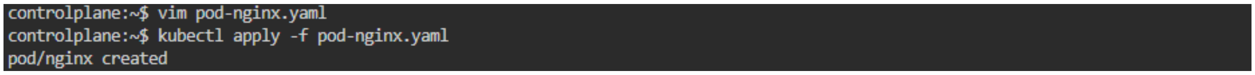
```
controlplane:~$ kubectl run redis --image=redis
pod/redis created
controlplane:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          12s
```

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



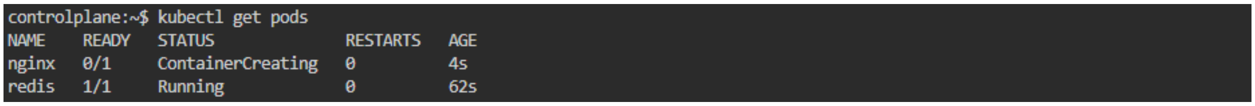
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx-container
      image: nginx123

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                9,0-1         All
```



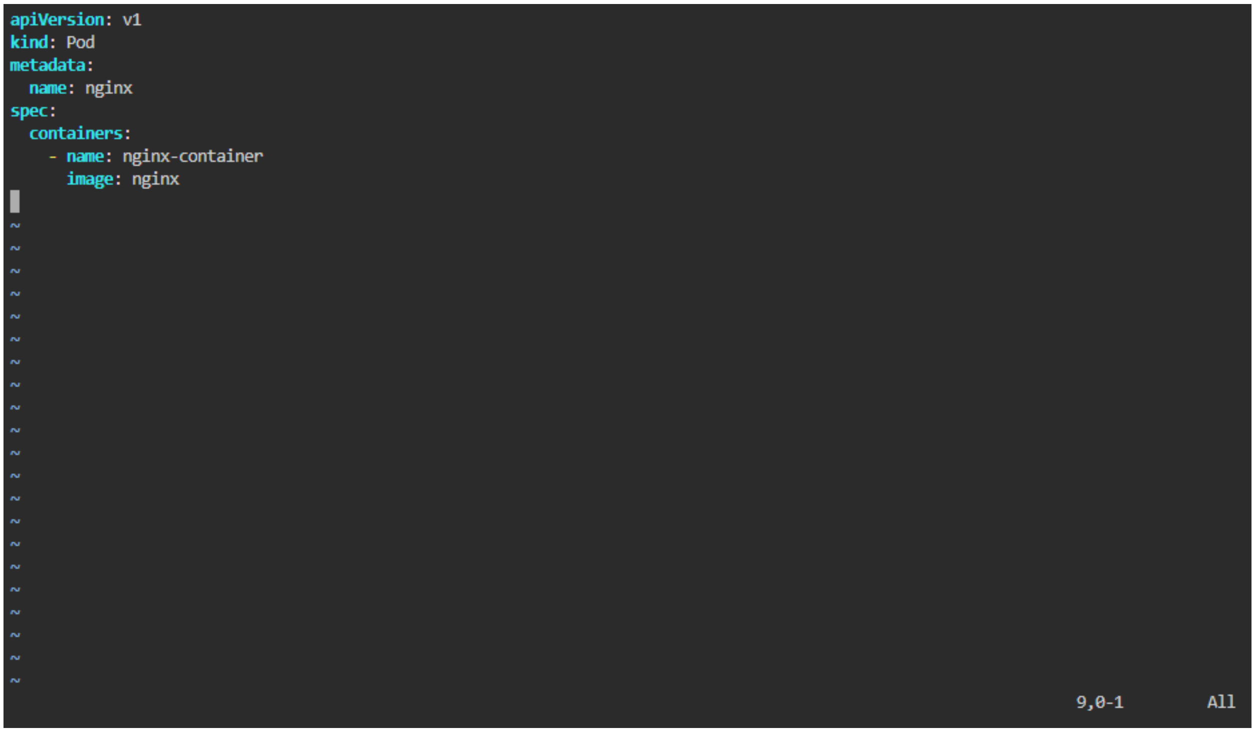
```
controlplane:~$ vim pod-nginx.yaml
controlplane:~$ kubectl apply -f pod-nginx.yaml
pod/nginx created
```

What is the nginx pod status?



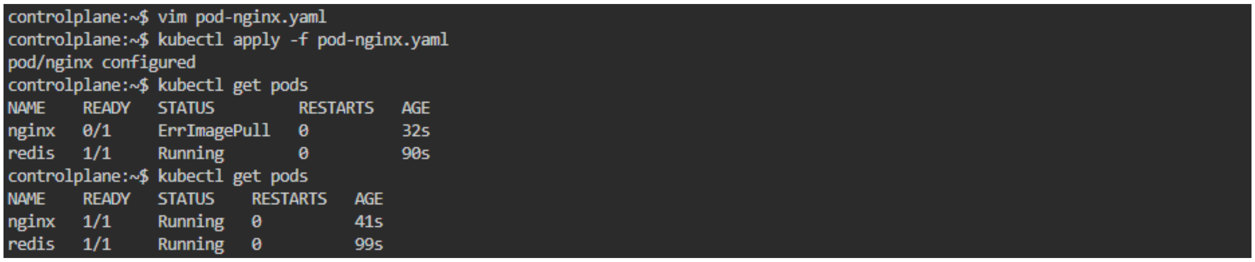
```
controlplane:~$ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
nginx   0/1     ContainerCreating   0          4s
redis   1/1     Running             0          62s
```

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



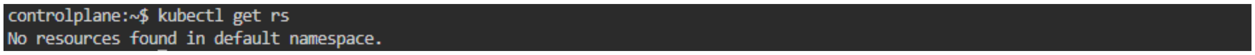
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx-container
      image: nginx

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                                                     9,0-1         All
```



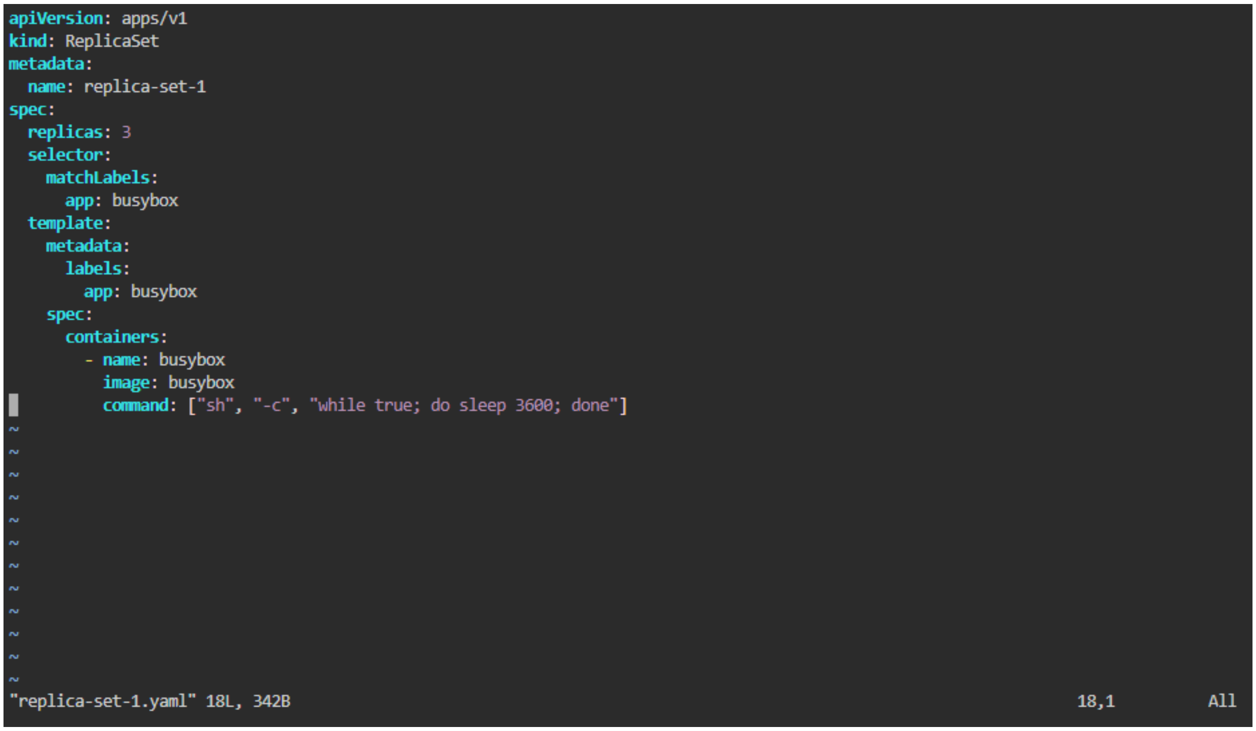
```
controlplane:~$ vim pod-nginx.yaml
controlplane:~$ kubectl apply -f pod-nginx.yaml
pod/nginx configured
controlplane:~$ kubectl get pods
NAME    READY   STATUS         RESTARTS   AGE
nginx   0/1     ErrImagePull   0          32s
redis   1/1     Running        0          90s
controlplane:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          41s
redis   1/1     Running   0          99s
```

How many ReplicaSets exist on the system?

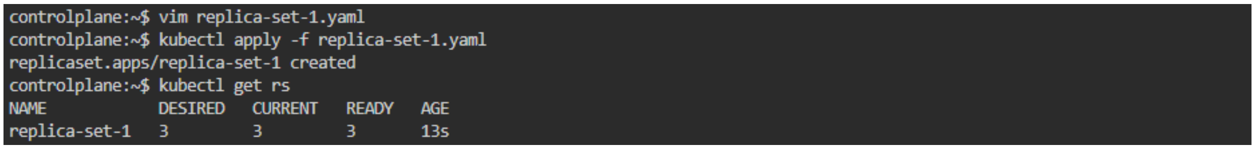


```
controlplane:~$ kubectl get rs
No resources found in default namespace.
```

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

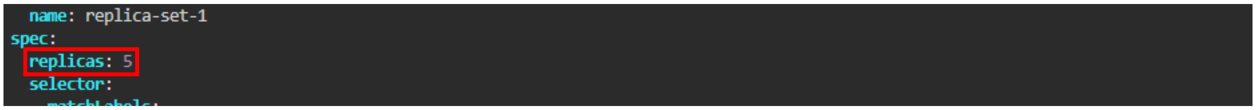


```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
        - name: busybox
          image: busybox
          command: ["sh", "-c", "while true; do sleep 3600; done"]
~
~
~
~
~
~
~
~
~
~
~
~
"replica-set-1.yaml" 18L, 342B                                                                          18,1          All
```

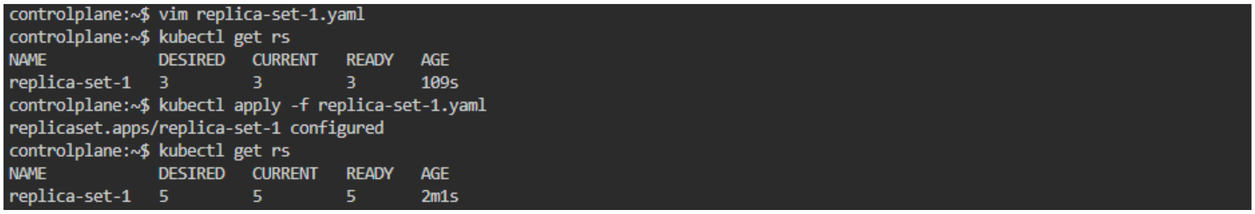


```
controlplane:~$ vim replica-set-1.yaml
controlplane:~$ kubectl apply -f replica-set-1.yaml
replicaset.apps/replica-set-1 created
controlplane:~$ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         3       13s
```

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

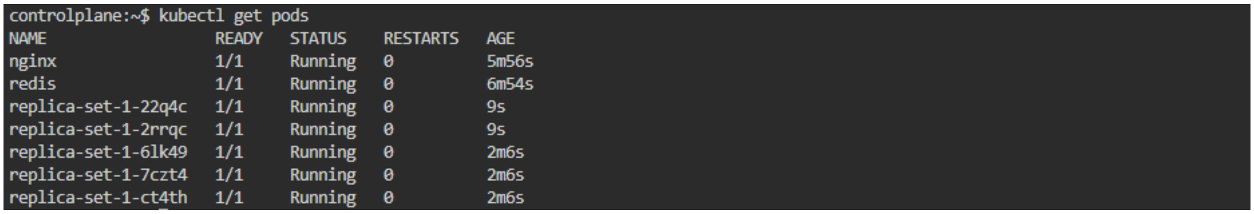


```
  name: replica-set-1
spec:
  replicas: 5
  selector:
    matchLabels:
```



```
controlplane:~$ vim replica-set-1.yaml
controlplane:~$ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         3       109s
controlplane:~$ kubectl apply -f replica-set-1.yaml
replicaset.apps/replica-set-1 configured
controlplane:~$ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         5       2m1s
```

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

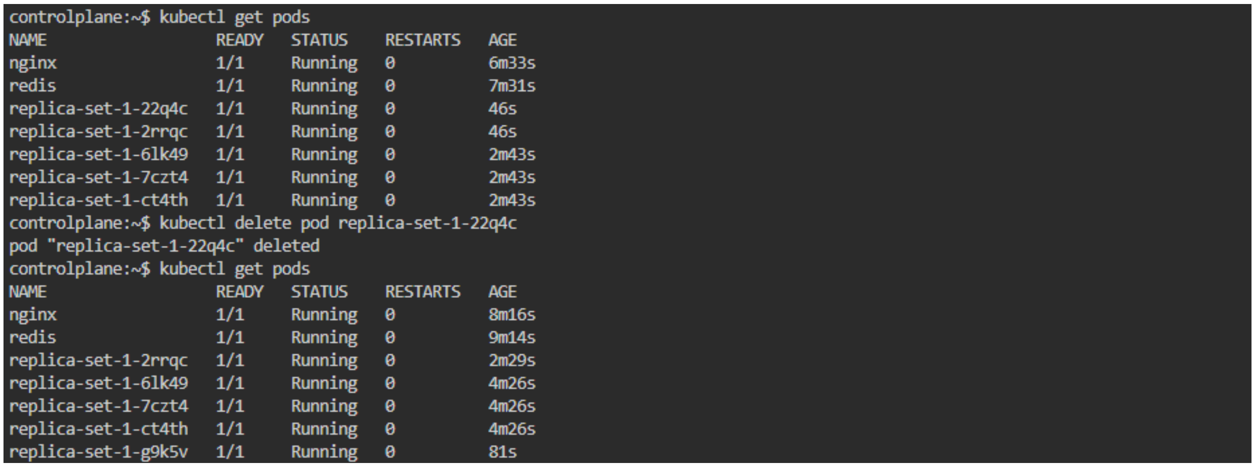


```
controlplane:~$ kubectl get pods
NAME                 READY   STATUS    RESTARTS   AGE
nginx                1/1     Running   0          5m56s
redis                1/1     Running   0          6m54s
replica-set-1-22q4c  1/1     Running   0          9s
replica-set-1-2rrqc  1/1     Running   0          9s
replica-set-1-6lk49  1/1     Running   0          2m6s
replica-set-1-7czt4  1/1     Running   0          2m6s
replica-set-1-ct4th  1/1     Running   0          2m6s
```

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

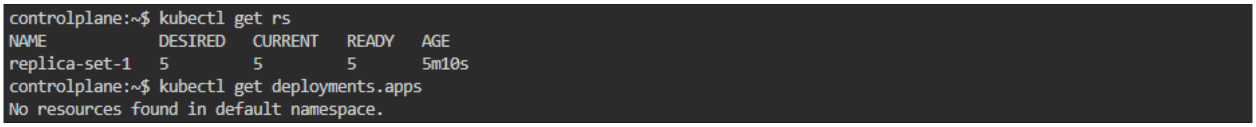
Why are there still 5 PODs, even after you deleted one?

because Replica-set ensures that a specified number of pod replicas are running at all times.



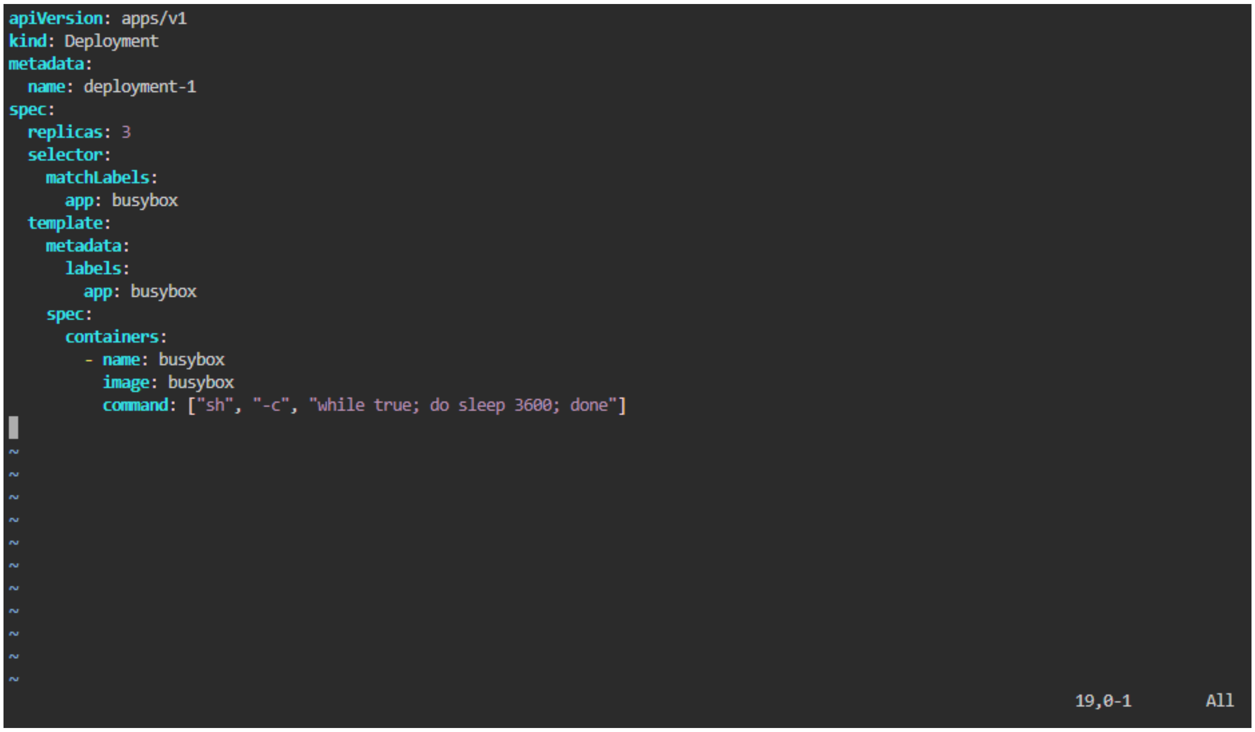
```
controlplane:~$ kubectl get pods
NAME                 READY   STATUS    RESTARTS   AGE
nginx                1/1     Running   0          6m33s
redis                1/1     Running   0          7m31s
replica-set-1-22q4c  1/1     Running   0          46s
replica-set-1-2rrqc  1/1     Running   0          46s
replica-set-1-6lk49  1/1     Running   0          2m43s
replica-set-1-7czt4  1/1     Running   0          2m43s
replica-set-1-ct4th  1/1     Running   0          2m43s
controlplane:~$ kubectl delete pod replica-set-1-22q4c
pod "replica-set-1-22q4c" deleted
controlplane:~$ kubectl get pods
NAME                 READY   STATUS    RESTARTS   AGE
nginx                1/1     Running   0          8m16s
redis                1/1     Running   0          9m14s
replica-set-1-2rrqc  1/1     Running   0          2m29s
replica-set-1-6lk49  1/1     Running   0          4m26s
replica-set-1-7czt4  1/1     Running   0          4m26s
replica-set-1-ct4th  1/1     Running   0          4m26s
replica-set-1-g9k5v  1/1     Running   0          81s
```

How many Deployments and ReplicaSets exist on the system?



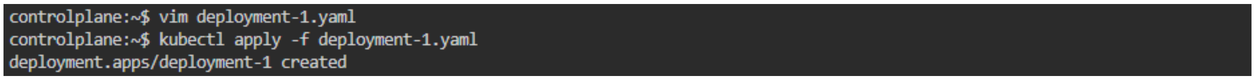
```
controlplane:~$ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         5       5m10s
controlplane:~$ kubectl get deployments.apps
No resources found in default namespace.
```

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



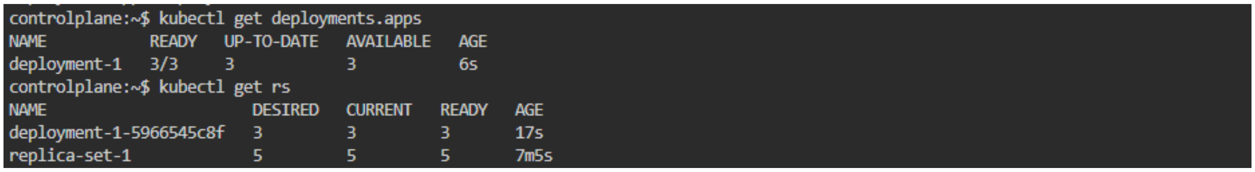
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
        - name: busybox
          image: busybox
          command: ["sh", "-c", "while true; do sleep 3600; done"]

~
~
~
~
~
~
~
~
~
~
~
                                                                                           19,0-1        All
```



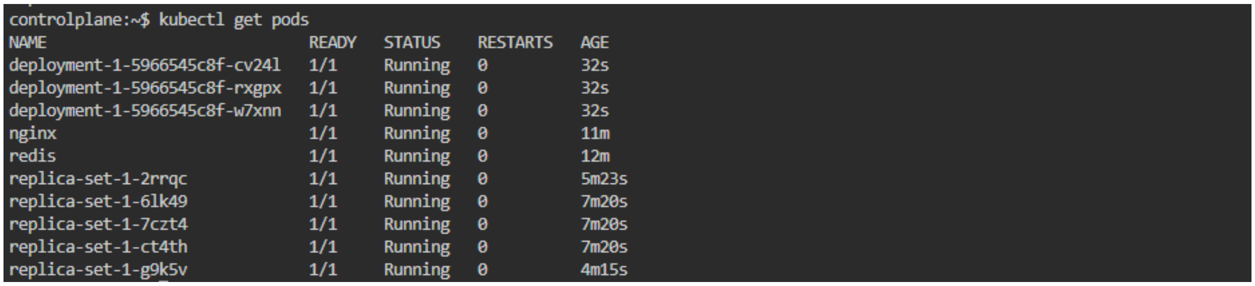
```
controlplane:~$ vim deployment-1.yaml
controlplane:~$ kubectl apply -f deployment-1.yaml
deployment.apps/deployment-1 created
```

How many Deployments and ReplicaSets exist on the system now?



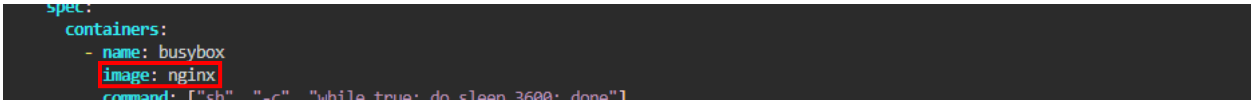
```
controlplane:~$ kubectl get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           6s
controlplane:~$ kubectl get rs
NAME                      DESIRED   CURRENT   READY   AGE
deployment-1-5966545c8f   3         3         3       17s
replica-set-1             5         5         5       7m5s
```

How many pods are ready with the deployment-1?

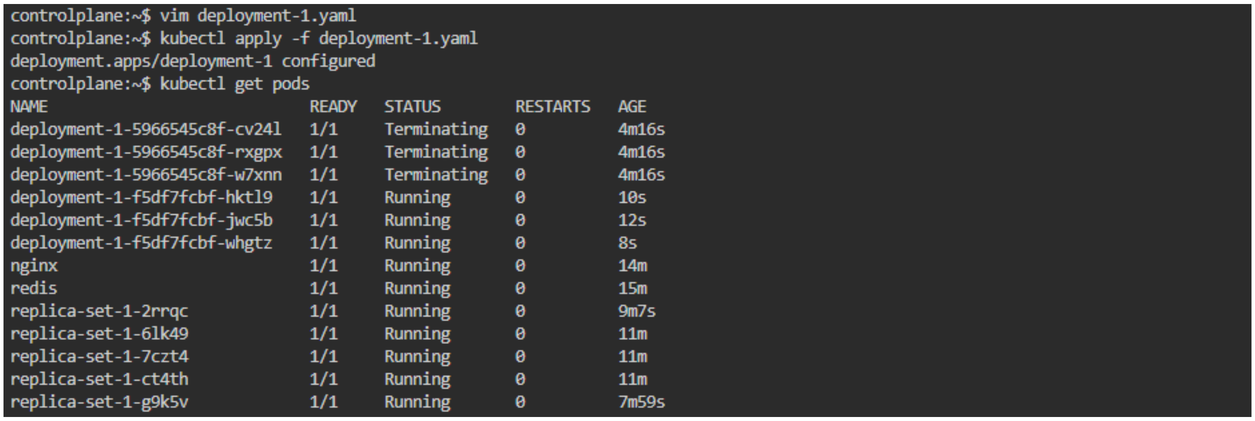


```
controlplane:~$ kubectl get pods
NAME                            READY   STATUS    RESTARTS   AGE
deployment-1-5966545c8f-cv24l   1/1     Running   0          32s
deployment-1-5966545c8f-rxgpx   1/1     Running   0          32s
deployment-1-5966545c8f-w7xnn   1/1     Running   0          32s
nginx                           1/1     Running   0          11m
redis                           1/1     Running   0          12m
replica-set-1-2rrqc             1/1     Running   0          5m23s
replica-set-1-6lk49             1/1     Running   0          7m20s
replica-set-1-7czt4             1/1     Running   0          7m20s
replica-set-1-ct4th             1/1     Running   0          7m20s
replica-set-1-g9k5v             1/1     Running   0          4m15s
```

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

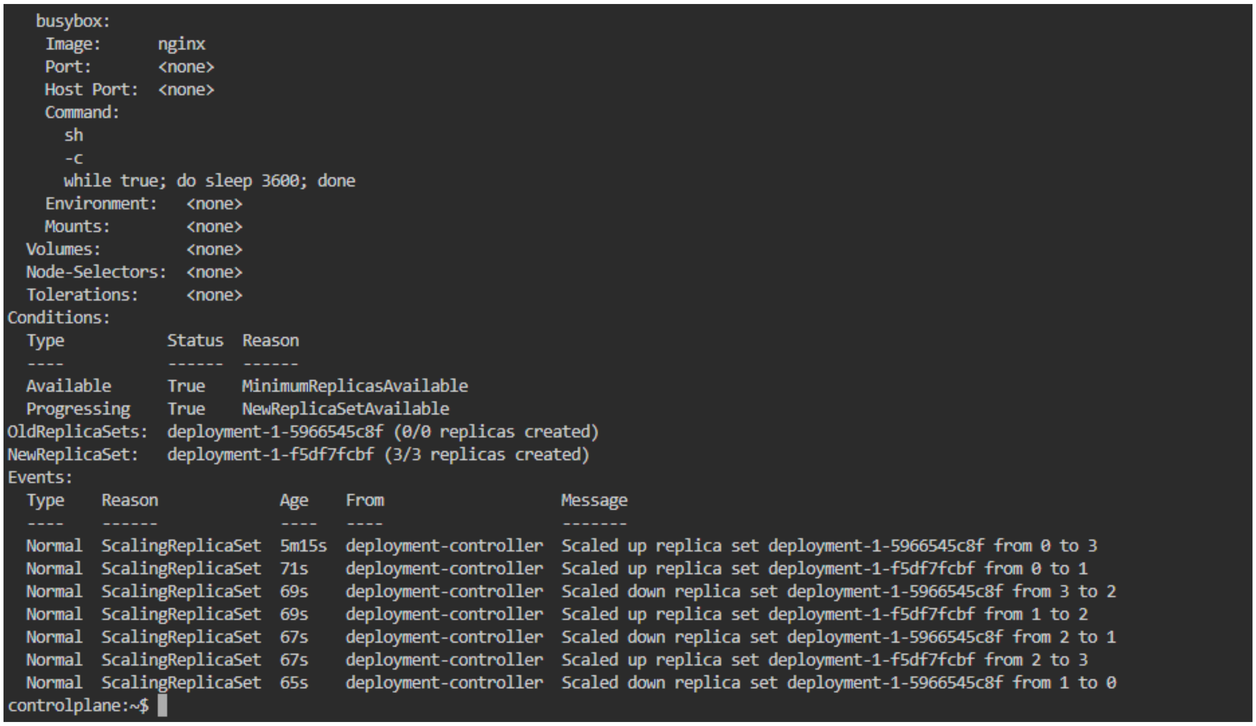


```
    spec:
      containers:
        - name: busybox
          image: nginx
          command: ["sh", "-c", "while true; do sleep 3600; done"]
```



```
controlplane:~$ vim deployment-1.yaml
controlplane:~$ kubectl apply -f deployment-1.yaml
deployment.apps/deployment-1 configured
controlplane:~$ kubectl get pods
NAME                            READY   STATUS        RESTARTS   AGE
deployment-1-5966545c8f-cv24l   1/1     Terminating   0          4m16s
deployment-1-5966545c8f-rxgpx   1/1     Terminating   0          4m16s
deployment-1-5966545c8f-w7xnn   1/1     Terminating   0          4m16s
deployment-1-f5df7fcbf-hktl9    1/1     Running       0          10s
deployment-1-f5df7fcbf-jwc5b    1/1     Running       0          12s
deployment-1-f5df7fcbf-whgtz    1/1     Running       0          8s
nginx                           1/1     Running       0          14m
redis                           1/1     Running       0          15m
replica-set-1-2rrqc             1/1     Running       0          9m7s
replica-set-1-6lk49             1/1     Running       0          11m
replica-set-1-7czt4             1/1     Running       0          11m
replica-set-1-ct4th             1/1     Running       0          11m
replica-set-1-g9k5v             1/1     Running       0          7m59s
```

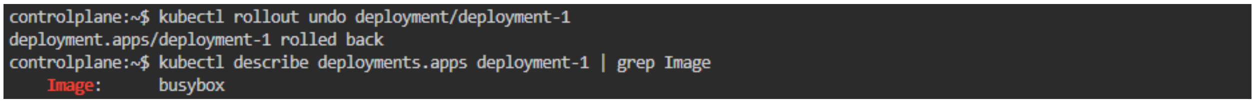
Run kubectl describe deployment deployment-1 and check events
What is the deployment strategy used to upgrade the deployment-1?



```
   busybox:
    Image:      nginx
    Port:       <none>
    Host Port:  <none>
    Command:
      sh
      -c
      while true; do sleep 3600; done
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  deployment-1-5966545c8f (0/0 replicas created)
NewReplicaSet:   deployment-1-f5df7fcbf (3/3 replicas created)
Events:
  Type    Reason             Age    From                   Message
  ----    ------             ----   ----                   -------
  Normal  ScalingReplicaSet  5m15s  deployment-controller  Scaled up replica set deployment-1-5966545c8f from 0 to 3
  Normal  ScalingReplicaSet  71s    deployment-controller  Scaled up replica set deployment-1-f5df7fcbf from 0 to 1
  Normal  ScalingReplicaSet  69s    deployment-controller  Scaled down replica set deployment-1-5966545c8f from 3 to 2
  Normal  ScalingReplicaSet  69s    deployment-controller  Scaled up replica set deployment-1-f5df7fcbf from 1 to 2
  Normal  ScalingReplicaSet  67s    deployment-controller  Scaled down replica set deployment-1-5966545c8f from 2 to 1
  Normal  ScalingReplicaSet  67s    deployment-controller  Scaled up replica set deployment-1-f5df7fcbf from 2 to 3
  Normal  ScalingReplicaSet  65s    deployment-controller  Scaled down replica set deployment-1-5966545c8f from 1 to 0
controlplane:~$ 
```

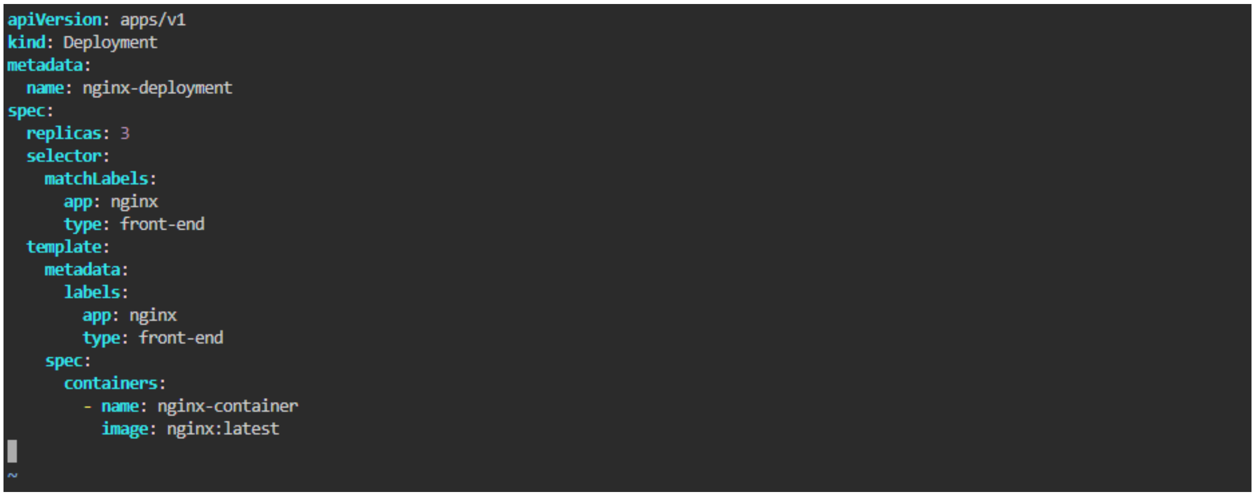
Rollback the deployment-1

What is the used image with the deployment-1?



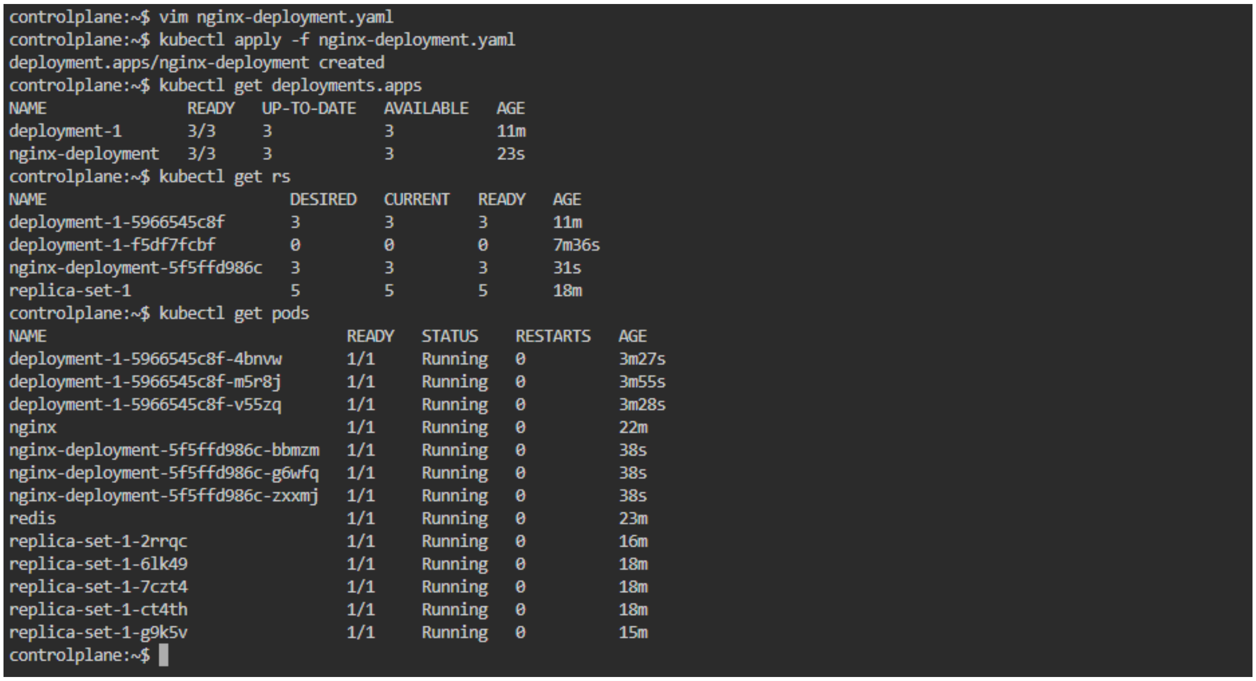
```
controlplane:~$ kubectl rollout undo deployment/deployment-1
deployment.apps/deployment-1 rolled back
controlplane:~$ kubectl describe deployments.apps deployment-1 | grep Image
    Image:      busybox
```

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.



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

~
```



```
controlplane:~$ vim nginx-deployment.yaml
controlplane:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
controlplane:~$ kubectl get deployments.apps
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1       3/3     3            3           11m
nginx-deployment   3/3     3            3           23s
controlplane:~$ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
deployment-1-5966545c8f       3         3         3       11m
deployment-1-f5df7fcbf        0         0         0       7m36s
nginx-deployment-5f5ffd986c   3         3         3       31s
replica-set-1                 5         5         5       18m
controlplane:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
deployment-1-5966545c8f-4bnvw       1/1     Running   0          3m27s
deployment-1-5966545c8f-m5r8j       1/1     Running   0          3m55s
deployment-1-5966545c8f-v55zq       1/1     Running   0          3m28s
nginx                               1/1     Running   0          22m
nginx-deployment-5f5ffd986c-bbmzm   1/1     Running   0          38s
nginx-deployment-5f5ffd986c-g6wfq   1/1     Running   0          38s
nginx-deployment-5f5ffd986c-zxxmj   1/1     Running   0          38s
redis                               1/1     Running   0          23m
replica-set-1-2rrqc                 1/1     Running   0          16m
replica-set-1-6lk49                 1/1     Running   0          18m
replica-set-1-7czt4                 1/1     Running   0          18m
replica-set-1-ct4th                 1/1     Running   0          18m
replica-set-1-g9k5v                 1/1     Running   0          15m
controlplane:~$ 
```