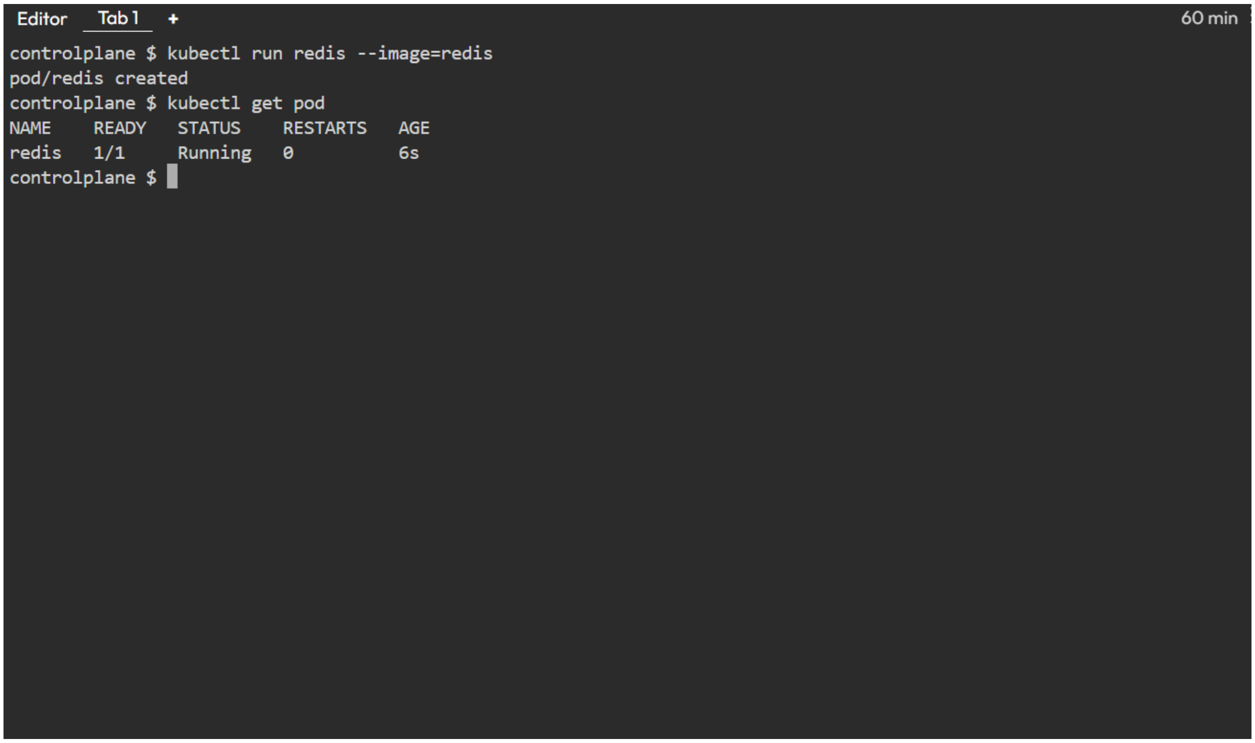
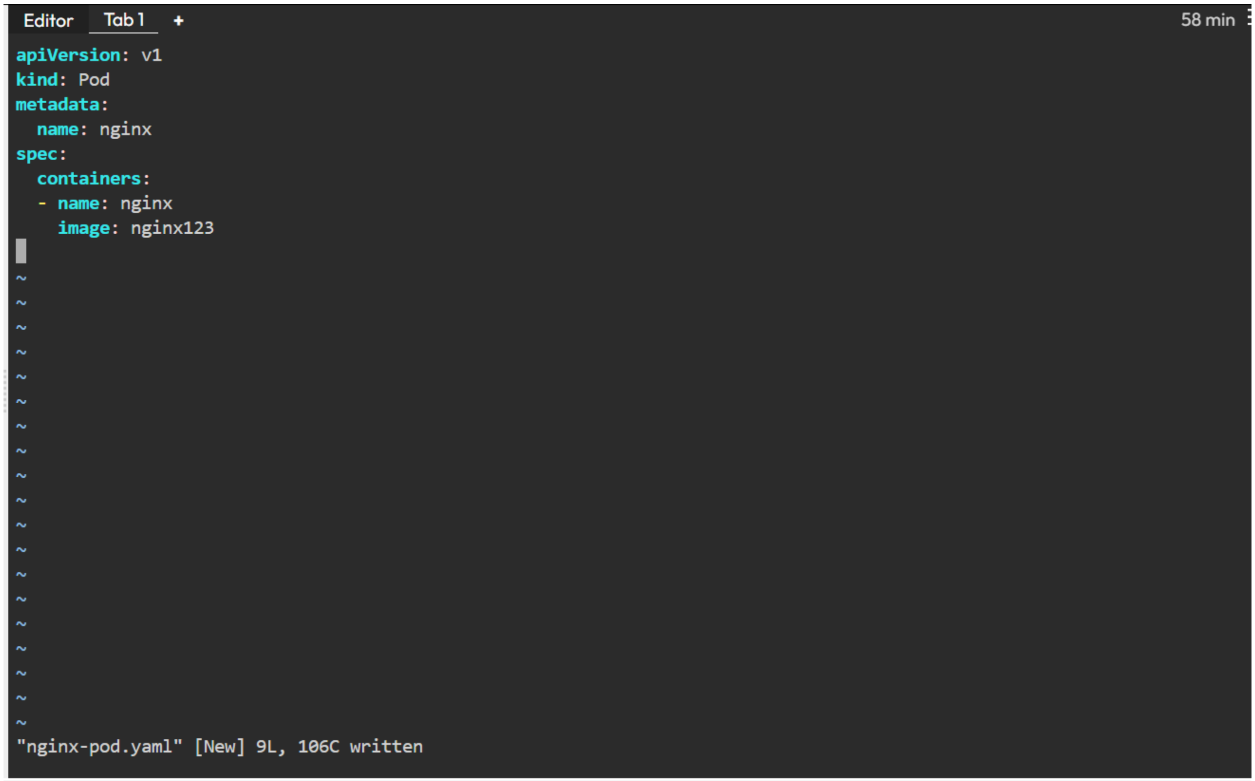
2- Create a pod with the name redis and with the image redis



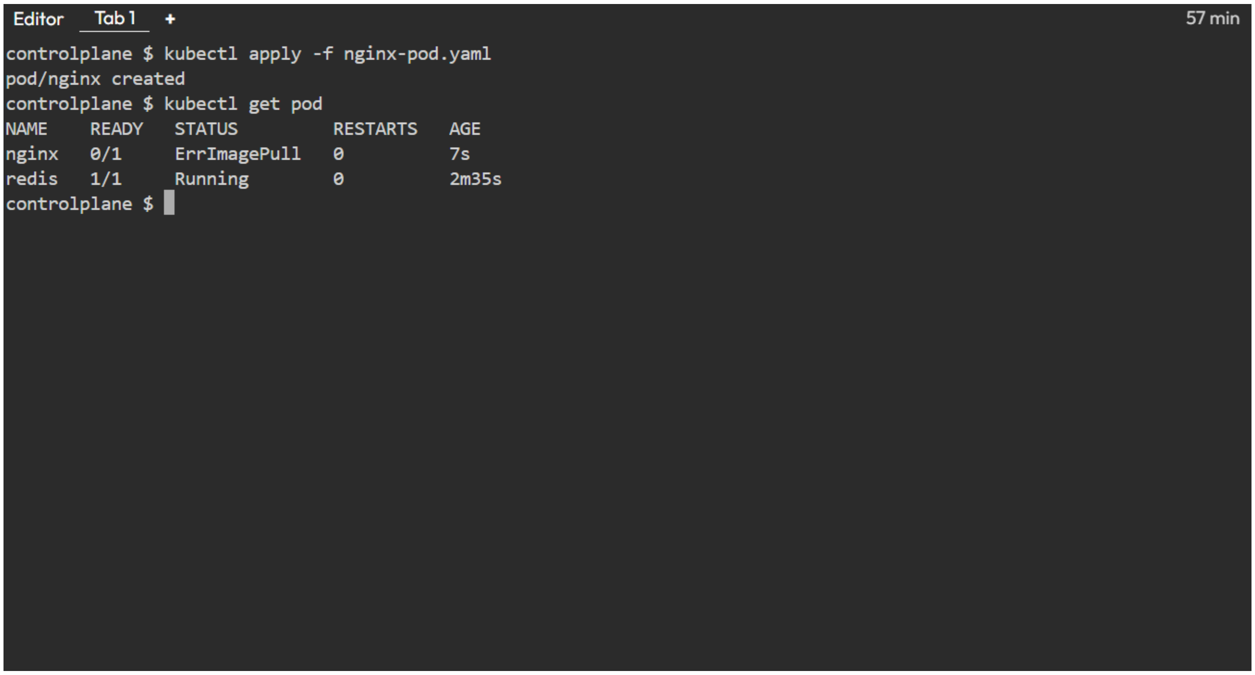
```
controlplane $ kubectl run redis --image=redis
pod/redis created
controlplane $ kubectl get pod
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          6s
controlplane $ 
```

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



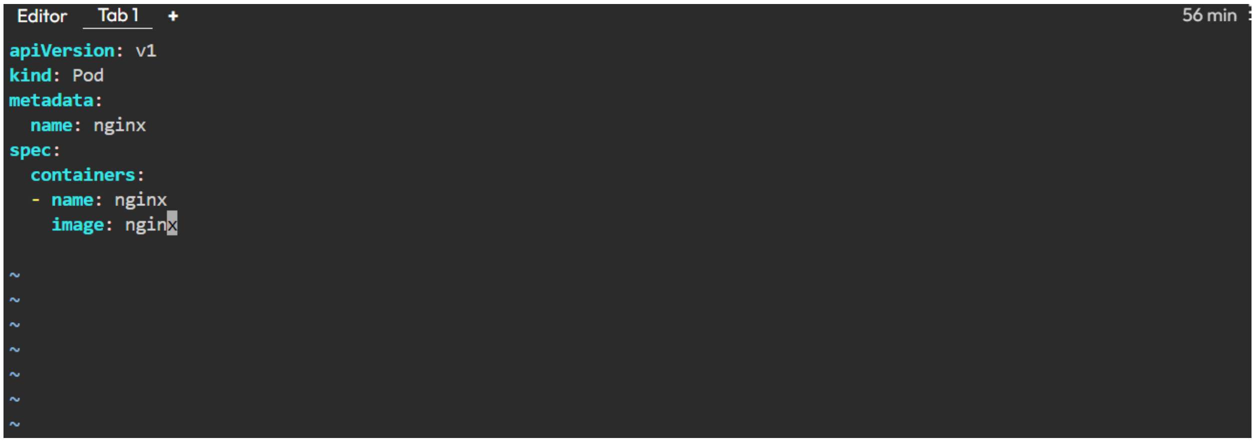
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx123

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"nginx-pod.yaml" [New] 9L, 106C written
```



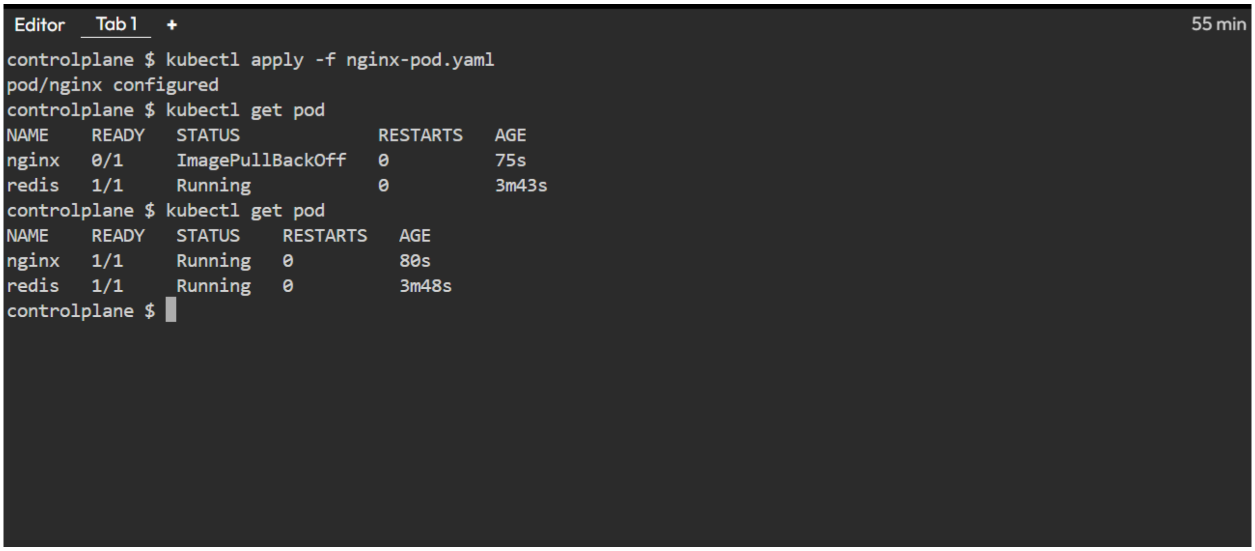
```
controlplane $ kubectl apply -f nginx-pod.yaml
pod/nginx created
controlplane $ kubectl get pod
NAME    READY   STATUS         RESTARTS   AGE
nginx   0/1     ErrImagePull   0          7s
redis   1/1     Running        0          2m35s
controlplane $ 
```

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



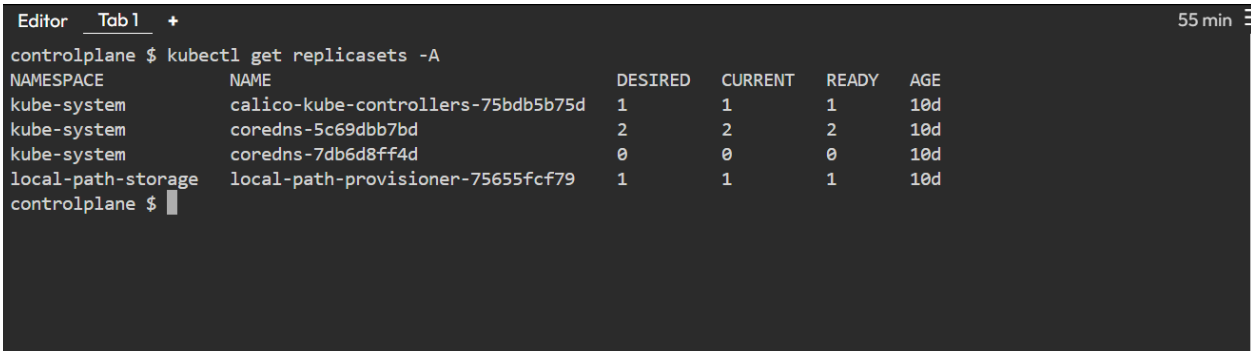
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx

~
~
~
~
~
~
~
```



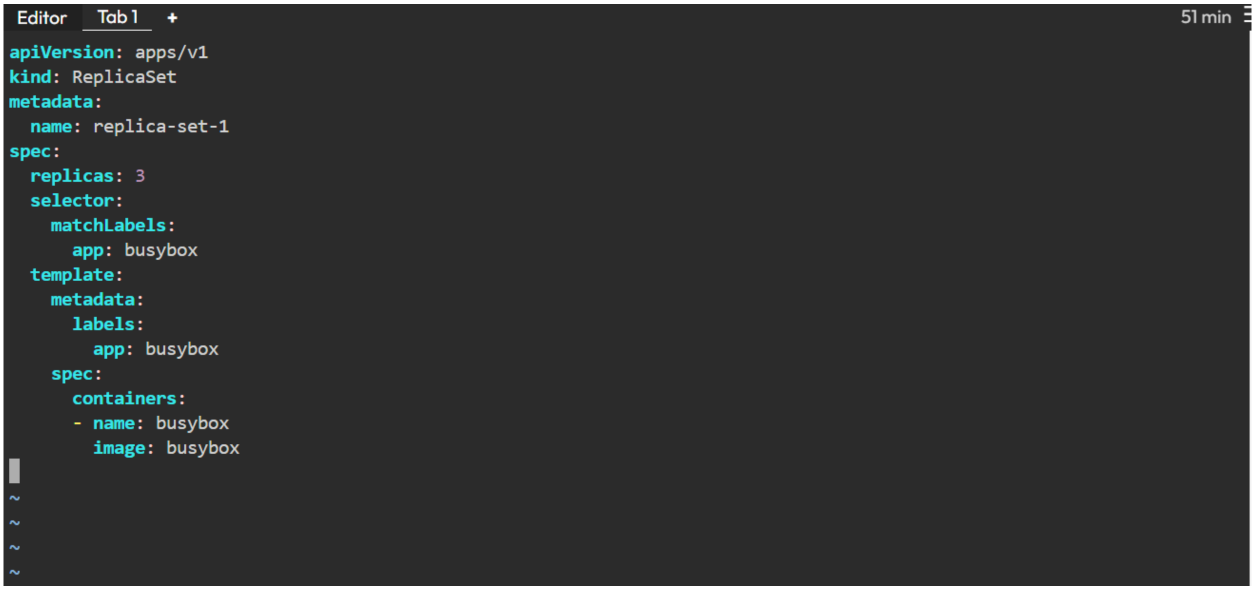
```
controlplane $ kubectl apply -f nginx-pod.yaml
pod/nginx configured
controlplane $ kubectl get pod
NAME    READY   STATUS             RESTARTS   AGE
nginx   0/1     ImagePullBackOff   0          75s
redis   1/1     Running            0          3m43s
controlplane $ kubectl get pod
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          80s
redis   1/1     Running   0          3m48s
controlplane $
```

6- How many ReplicaSets exist on the system?



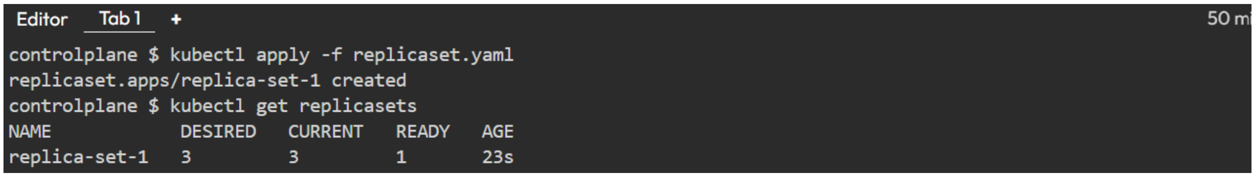
```
controlplane $ kubectl get replicasets -A
NAMESPACE            NAME                                 DESIRED   CURRENT   READY   AGE
kube-system          calico-kube-controllers-75bdb5b75d   1         1         1       10d
kube-system          coredns-5c69dbb7bd                   2         2         2       10d
kube-system          coredns-7db6d8ff4d                   0         0         0       10d
local-path-storage   local-path-provisioner-75655fcf79    1         1         1       10d
controlplane $
```

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



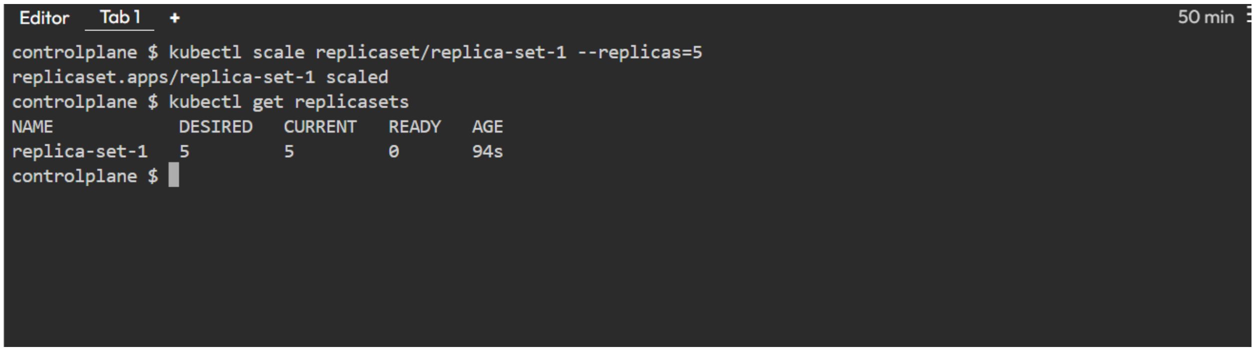
```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox

~
~
~
~
```



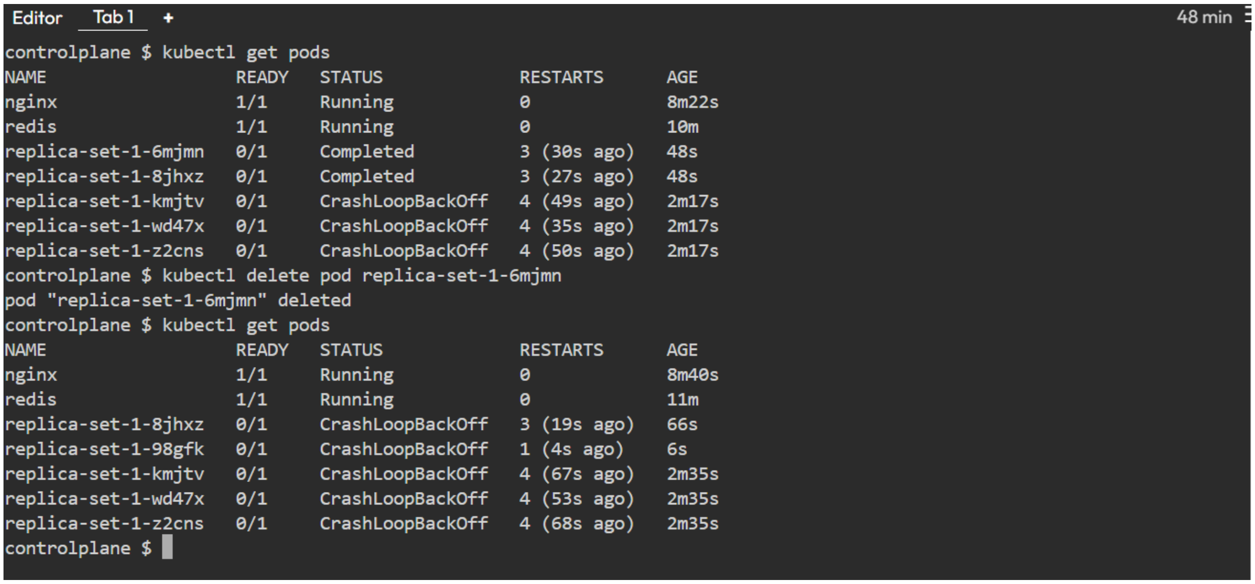
```
controlplane $ kubectl apply -f replicaset.yaml
replicaset.apps/replica-set-1 created
controlplane $ kubectl get replicasets
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         1       23s
```

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



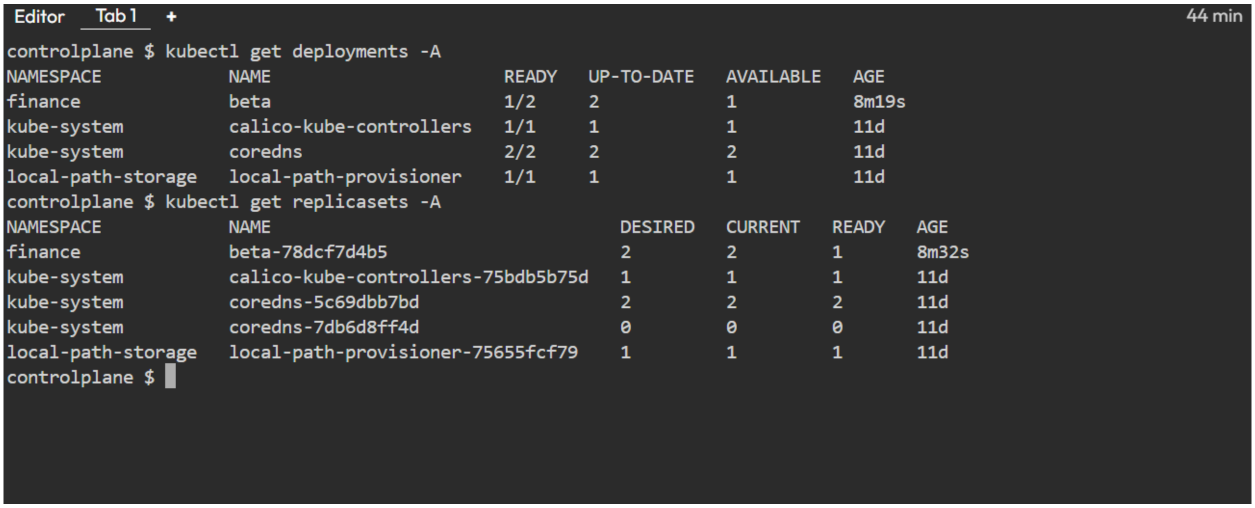
```
controlplane $ kubectl scale replicaset/replica-set-1 --replicas=5
replicaset.apps/replica-set-1 scaled
controlplane $ kubectl get replicasets
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         0       94s
controlplane $ 
```

10- 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?



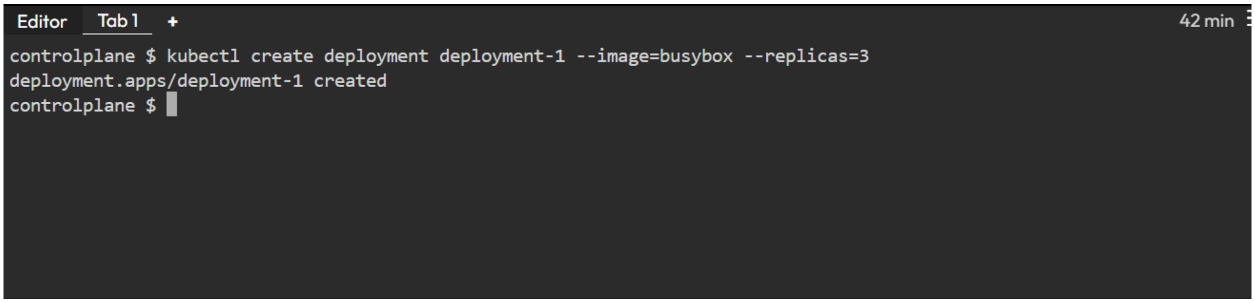
```
controlplane $ kubectl get pods
NAME                 READY   STATUS             RESTARTS     AGE
nginx                1/1     Running            0            8m22s
redis                1/1     Running            0            10m
replica-set-1-6mjmn  0/1     Completed          3 (30s ago)  48s
replica-set-1-8jhxz  0/1     Completed          3 (27s ago)  48s
replica-set-1-kmjtv  0/1     CrashLoopBackOff   4 (49s ago)  2m17s
replica-set-1-wd47x  0/1     CrashLoopBackOff   4 (35s ago)  2m17s
replica-set-1-z2cns  0/1     CrashLoopBackOff   4 (50s ago)  2m17s
controlplane $ kubectl delete pod replica-set-1-6mjmn
pod "replica-set-1-6mjmn" deleted
controlplane $ kubectl get pods
NAME                 READY   STATUS             RESTARTS     AGE
nginx                1/1     Running            0            8m40s
redis                1/1     Running            0            11m
replica-set-1-8jhxz  0/1     CrashLoopBackOff   3 (19s ago)  66s
replica-set-1-98gfk  0/1     CrashLoopBackOff   1 (4s ago)   6s
replica-set-1-kmjtv  0/1     CrashLoopBackOff   4 (67s ago)  2m35s
replica-set-1-wd47x  0/1     CrashLoopBackOff   4 (53s ago)  2m35s
replica-set-1-z2cns  0/1     CrashLoopBackOff   4 (68s ago)  2m35s
controlplane $ 
```

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



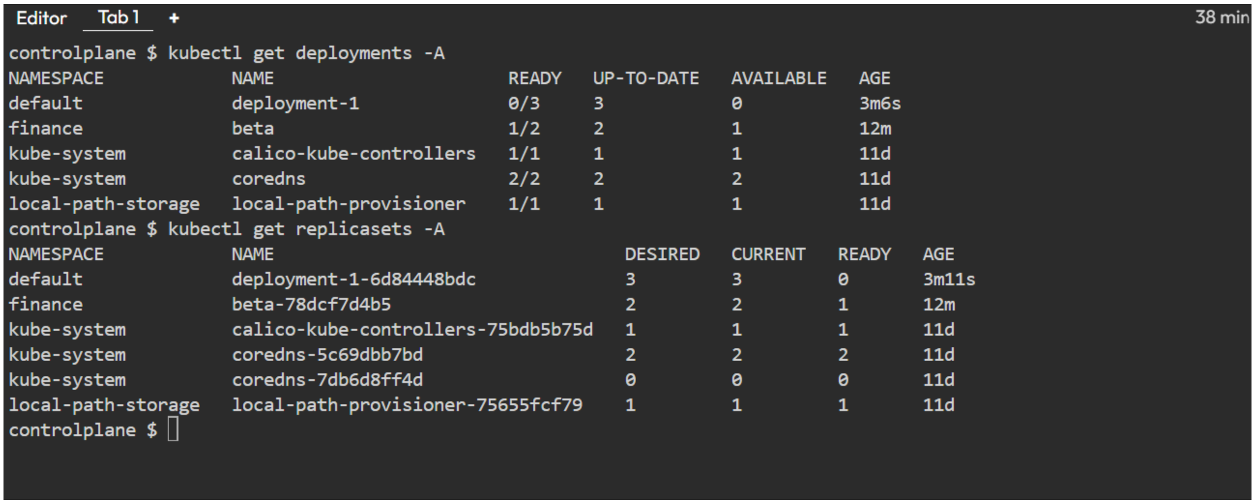
```
controlplane $ kubectl get deployments -A
NAMESPACE           NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
finance             beta                     1/2     2            1           8m19s
kube-system         calico-kube-controllers  1/1     1            1           11d
kube-system         coredns                  2/2     2            2           11d
local-path-storage  local-path-provisioner   1/1     1            1           11d
controlplane $ kubectl get replicasets -A
NAMESPACE           NAME                                DESIRED   CURRENT   READY   AGE
finance             beta-78dcf7d4b5                     2         2         1       8m32s
kube-system         calico-kube-controllers-75bdb5b75d  1         1         1       11d
kube-system         coredns-5c69dbb7bd                  2         2         2       11d
kube-system         coredns-7db6d8ff4d                  0         0         0       11d
local-path-storage  local-path-provisioner-75655fcf79   1         1         1       11d
controlplane $ 
```

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



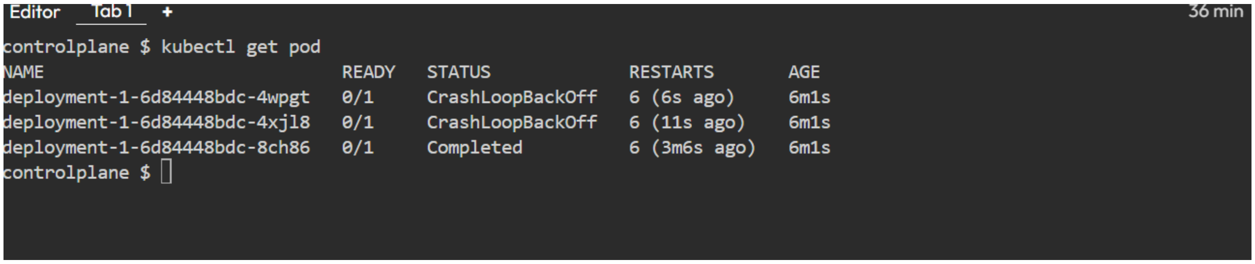
```
controlplane $ kubectl create deployment deployment-1 --image=busybox --replicas=3
deployment.apps/deployment-1 created
controlplane $ 
```

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



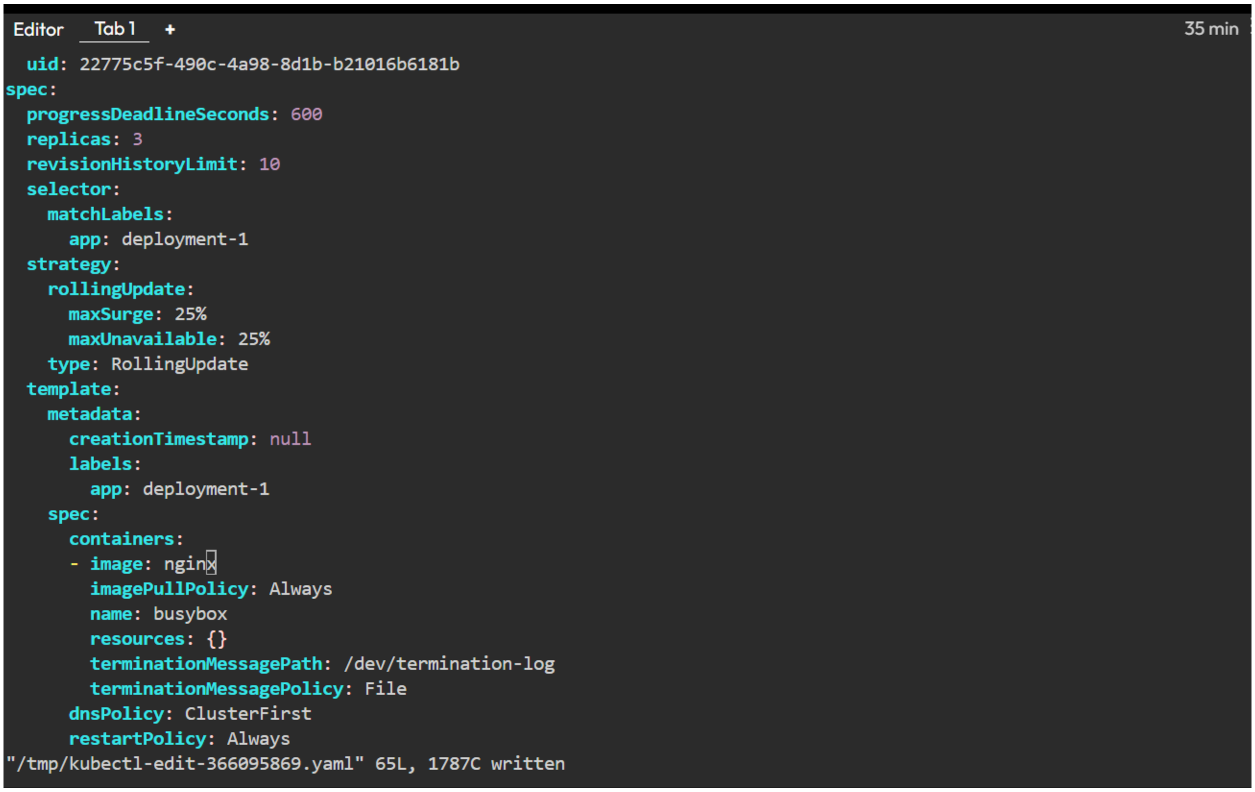
```
controlplane $ kubectl get deployments -A
NAMESPACE            NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
default              deployment-1              0/3     3            0           3m6s
finance              beta                      1/2     2            1           12m
kube-system          calico-kube-controllers   1/1     1            1           11d
kube-system          coredns                   2/2     2            2           11d
local-path-storage   local-path-provisioner    1/1     1            1           11d
controlplane $ kubectl get replicasets -A
NAMESPACE            NAME                                 DESIRED   CURRENT   READY   AGE
default              deployment-1-6d84448bdc              3         3         0       3m11s
finance              beta-78dcf7d4b5                      2         2         1       12m
kube-system          calico-kube-controllers-75bdb5b75d   1         1         1       11d
kube-system          coredns-5c69dbb7bd                   2         2         2       11d
kube-system          coredns-7db6d8ff4d                   0         0         0       11d
local-path-storage   local-path-provisioner-75655fcf79    1         1         1       11d
controlplane $ 
```

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

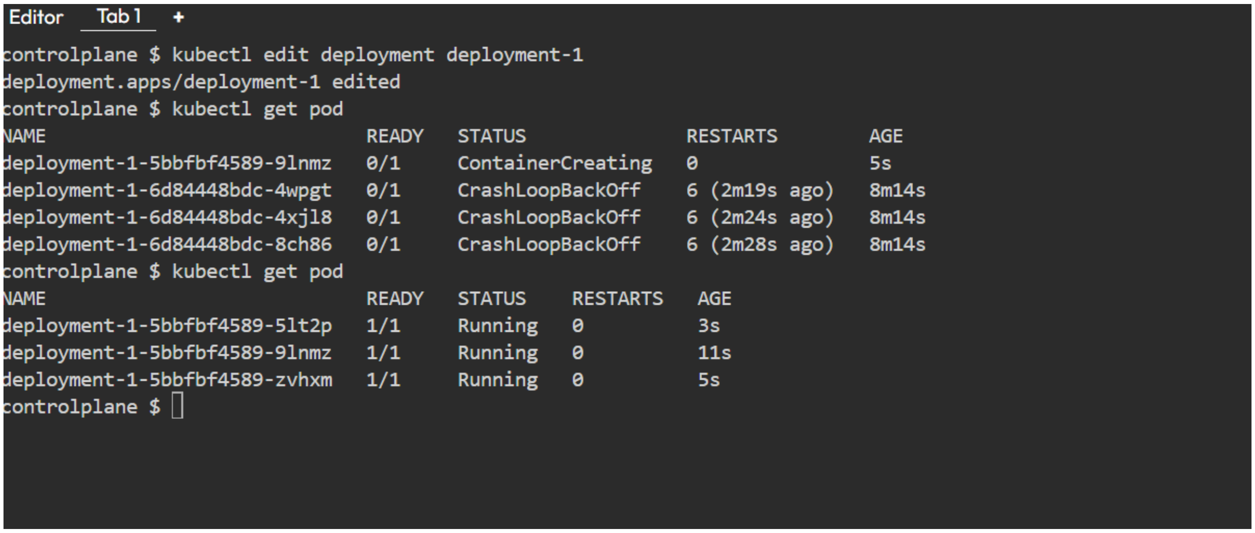


```
controlplane $ kubectl get pod
NAME                            READY   STATUS             RESTARTS       AGE
deployment-1-6d84448bdc-4wpgt   0/1     CrashLoopBackOff   6 (6s ago)     6m1s
deployment-1-6d84448bdc-4xjl8   0/1     CrashLoopBackOff   6 (11s ago)    6m1s
deployment-1-6d84448bdc-8ch86   0/1     Completed          6 (3m6s ago)   6m1s
controlplane $ 
```

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

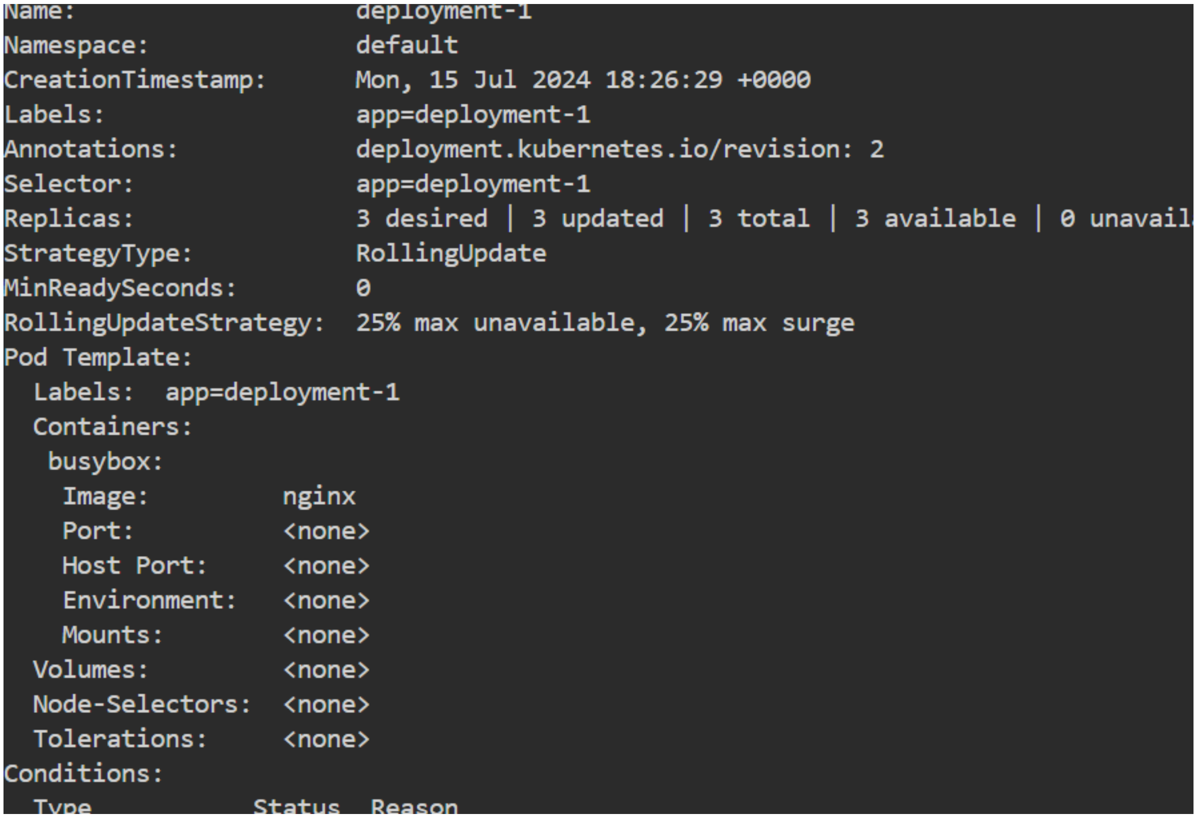


```
  uid: 22775c5f-490c-4a98-8d1b-b21016b6181b
spec:
  progressDeadlineSeconds: 600
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: deployment-1
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: deployment-1
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: busybox
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
"/tmp/kubectl-edit-366095869.yaml" 65L, 1787C written
```



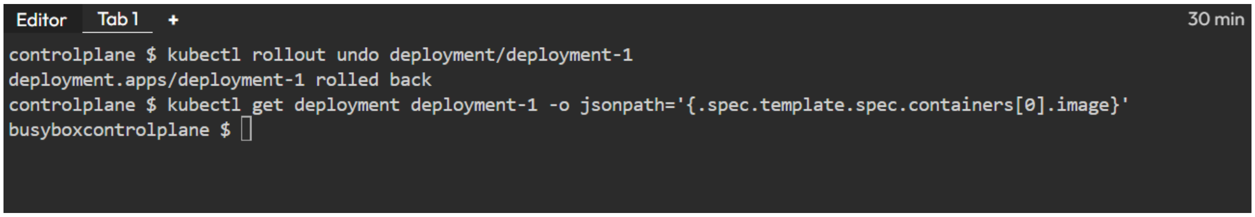
```
controlplane $ kubectl edit deployment deployment-1
deployment.apps/deployment-1 edited
controlplane $ kubectl get pod
NAME                            READY   STATUS              RESTARTS        AGE
deployment-1-5bbfbf4589-9lnmz   0/1     ContainerCreating   0               5s
deployment-1-6d84448bdc-4wpgt   0/1     CrashLoopBackOff    6 (2m19s ago)   8m14s
deployment-1-6d84448bdc-4xjl8   0/1     CrashLoopBackOff    6 (2m24s ago)   8m14s
deployment-1-6d84448bdc-8ch86   0/1     CrashLoopBackOff    6 (2m28s ago)   8m14s
controlplane $ kubectl get pod
NAME                            READY   STATUS    RESTARTS   AGE
deployment-1-5bbfbf4589-5lt2p   1/1     Running   0          3s
deployment-1-5bbfbf4589-9lnmz   1/1     Running   0          11s
deployment-1-5bbfbf4589-zvhxm   1/1     Running   0          5s
controlplane $ 
```

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



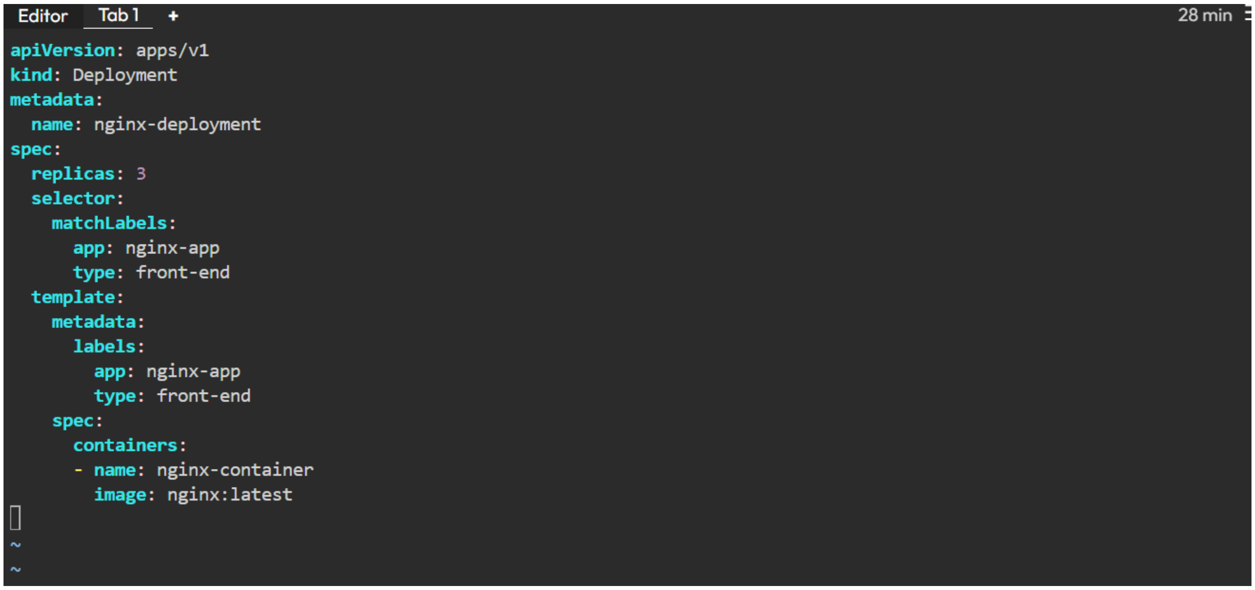
```
Name:                   deployment-1
Namespace:              default
CreationTimestamp:      Mon, 15 Jul 2024 18:26:29 +0000
Labels:                 app=deployment-1
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=deployment-1
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavail
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=deployment-1
  Containers:
   busybox:
    Image:         nginx
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
```

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



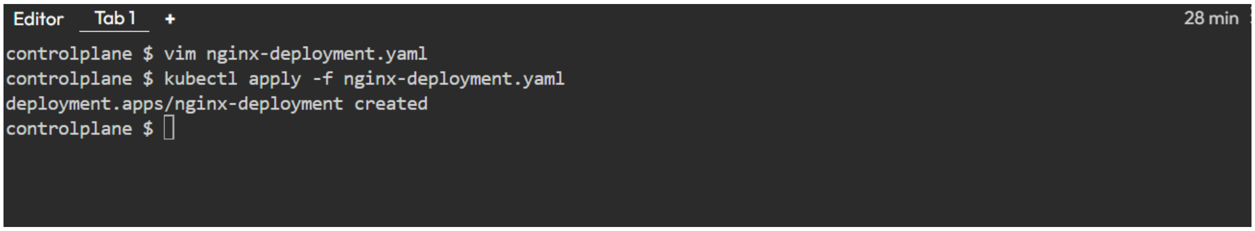
```
controlplane $ kubectl rollout undo deployment/deployment-1
deployment.apps/deployment-1 rolled back
controlplane $ kubectl get deployment deployment-1 -o jsonpath='{.spec.template.spec.containers[0].image}'
busyboxcontrolplane $ 
```

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.



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