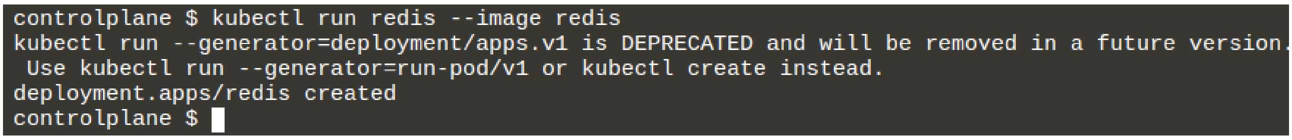
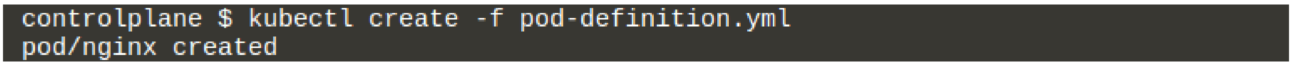
2- Create a pod with the name redis and with the image redis.

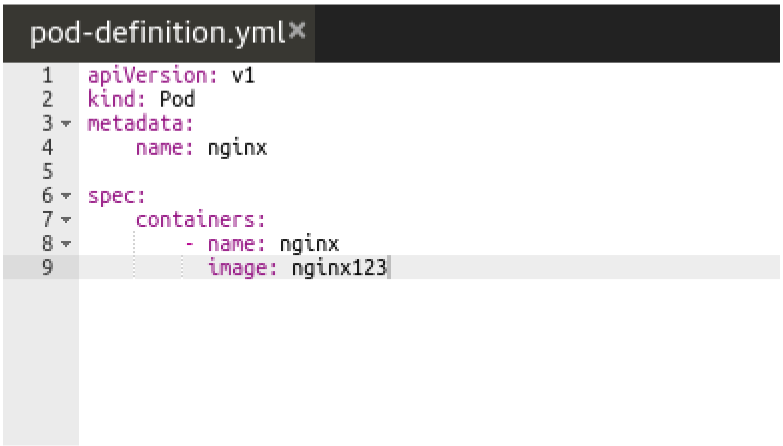


```
controlplane $ kubectl run redis --image redis
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version.
 Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/redis created
controlplane $ 
```

3- Create a pod with the name nginx and with the image nginx123.
Use a pod-definition YAML file. And yes the image name is wrong!



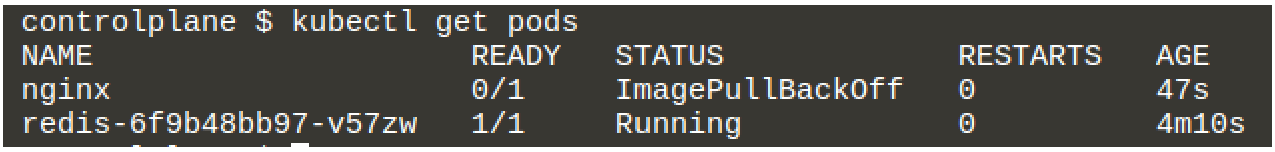
```
controlplane $ kubectl create -f pod-definition.yml
pod/nginx created
```



pod-definition.yml

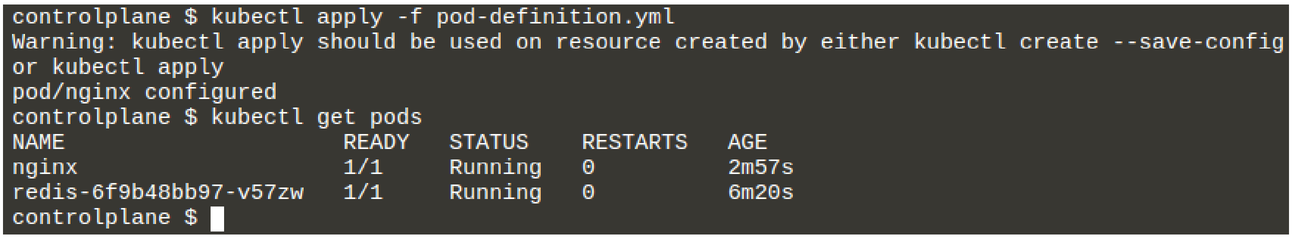
```
apiVersion: v1
kind: Pod
metadata:
    name: nginx

spec:
    containers:
        - name: nginx
          image: nginx123
```

4- What is the nginx pod status?



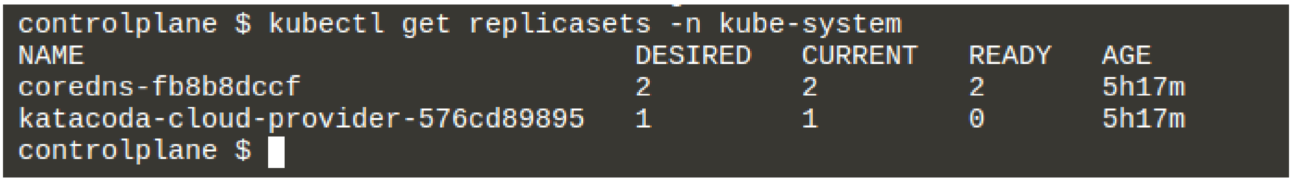
```
controlplane $ kubectl get pods
NAME                       READY    STATUS              RESTARTS    AGE
nginx                      0/1      ImagePullBackOff    0           47s
redis-6f9b48bb97-v57zw     1/1      Running             0           4m10s
```

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



```
controlplane $ kubectl apply -f pod-definition.yml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config
or kubectl apply
pod/nginx configured
controlplane $ kubectl get pods
NAME                      READY    STATUS    RESTARTS    AGE
nginx                     1/1      Running   0           2m57s
redis-6f9b48bb97-v57zw    1/1      Running   0           6m20s
controlplane $ 
```

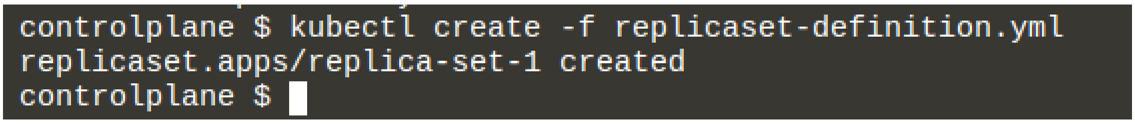
6- How many ReplicaSets exist on the system?



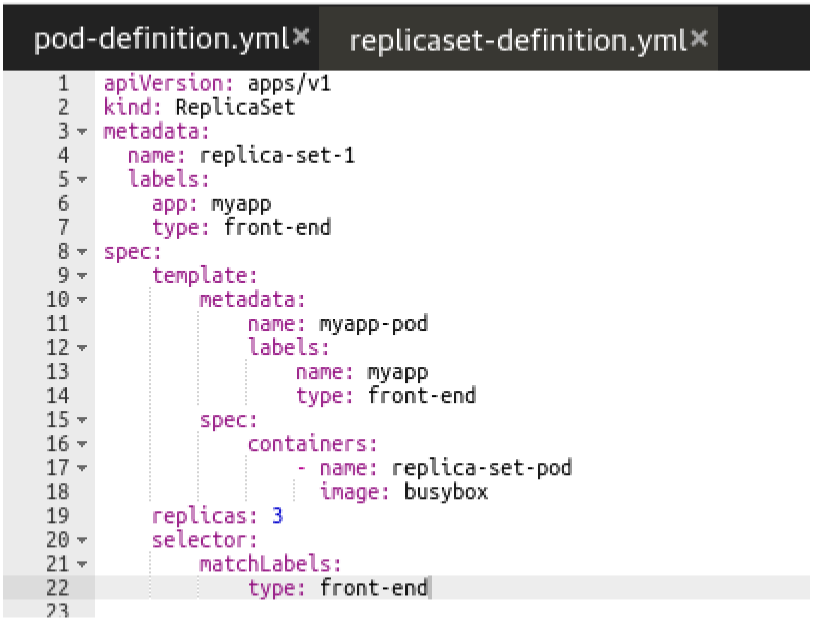
```
controlplane $ kubectl get replicasets -n kube-system
NAME                                  DESIRED    CURRENT    READY    AGE
coredns-fb8b8dccf                     2          2          2        5h17m
katacoda-cloud-provider-576cd89895    1          1          0        5h17m
controlplane $ 
```

7- create a ReplicaSet with

name= replica-set-1
image= busybox
replicas= 3

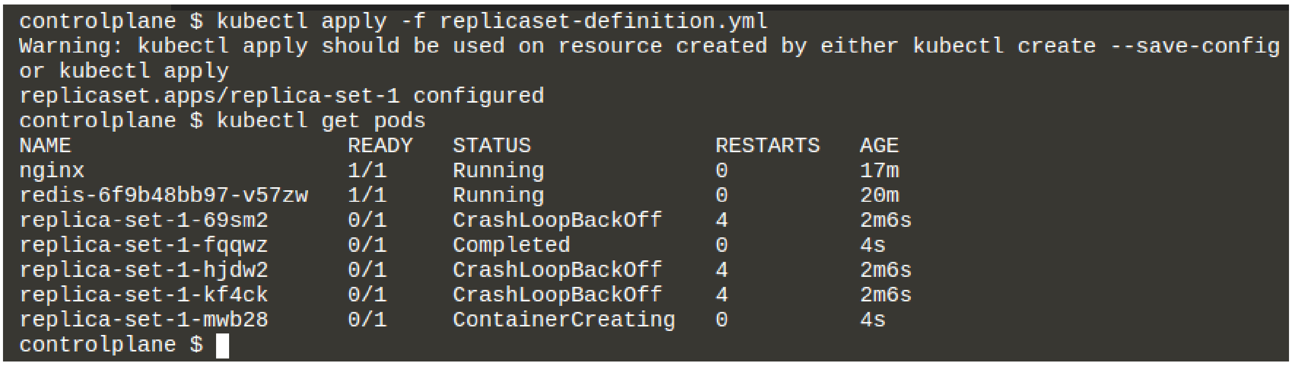


```
controlplane $ kubectl create -f replicaset-definition.yml
replicaset.apps/replica-set-1 created
controlplane $ 
```



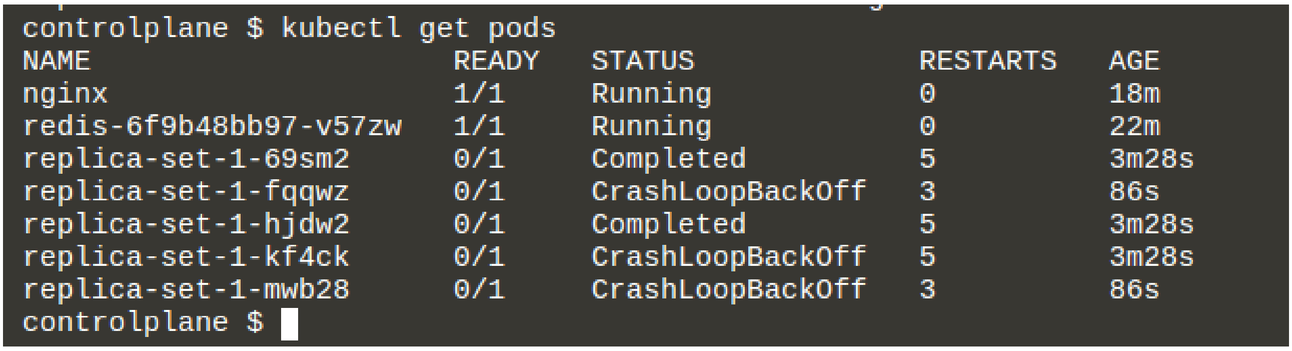
```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
  labels:
    app: myapp
    type: front-end
spec:
    template:
        metadata:
            name: myapp-pod
            labels:
                name: myapp
                type: front-end
        spec:
            containers: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

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



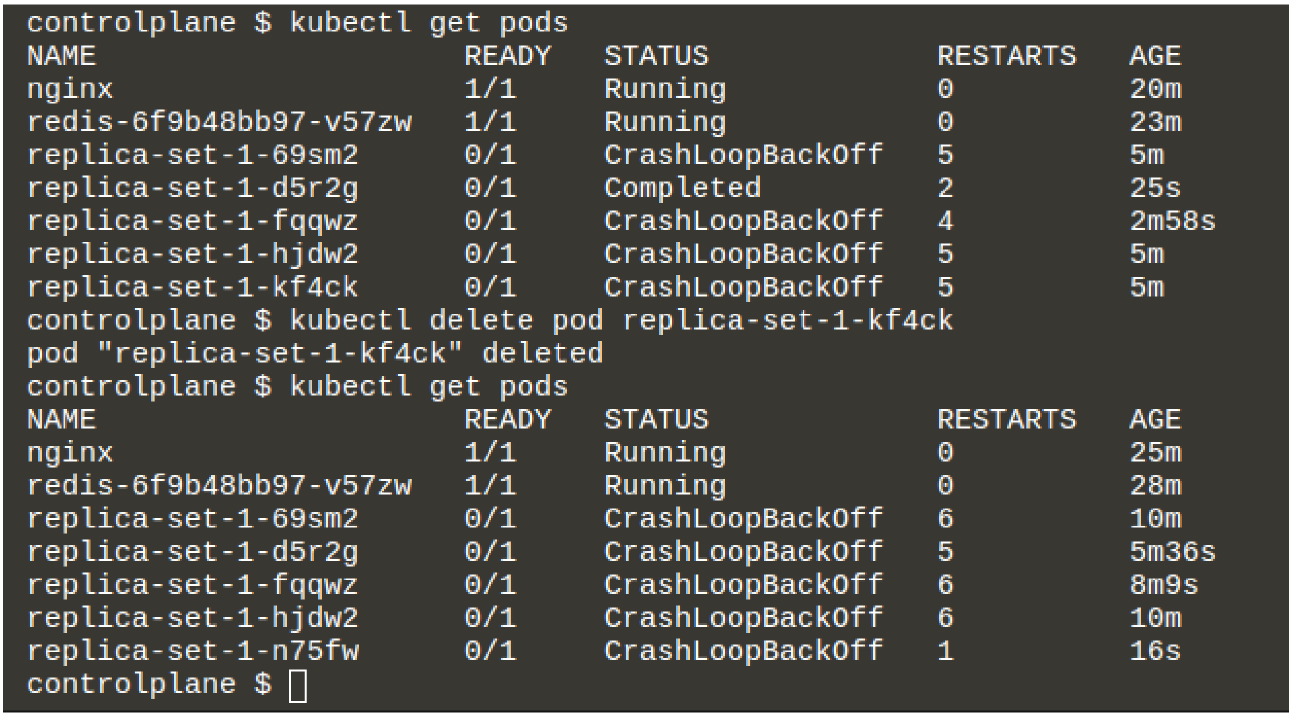
```
controlplane $ kubectl apply -f replicaset-definition.yml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config
or kubectl apply
replicaset.apps/replica-set-1 configured
controlplane $ kubectl get pods
NAME                     READY   STATUS              RESTARTS   AGE
nginx                    1/1     Running             0          17m
redis-6f9b48bb97-v57zw   1/1     Running             0          20m
replica-set-1-69sm2      0/1     CrashLoopBackOff    4          2m6s
replica-set-1-fqqwz      0/1     Completed           0          4s
replica-set-1-hjdw2      0/1     CrashLoopBackOff    4          2m6s
replica-set-1-kf4ck      0/1     CrashLoopBackOff    4          2m6s
replica-set-1-mwb28      0/1     ContainerCreating   0          4s
controlplane $ 
```

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



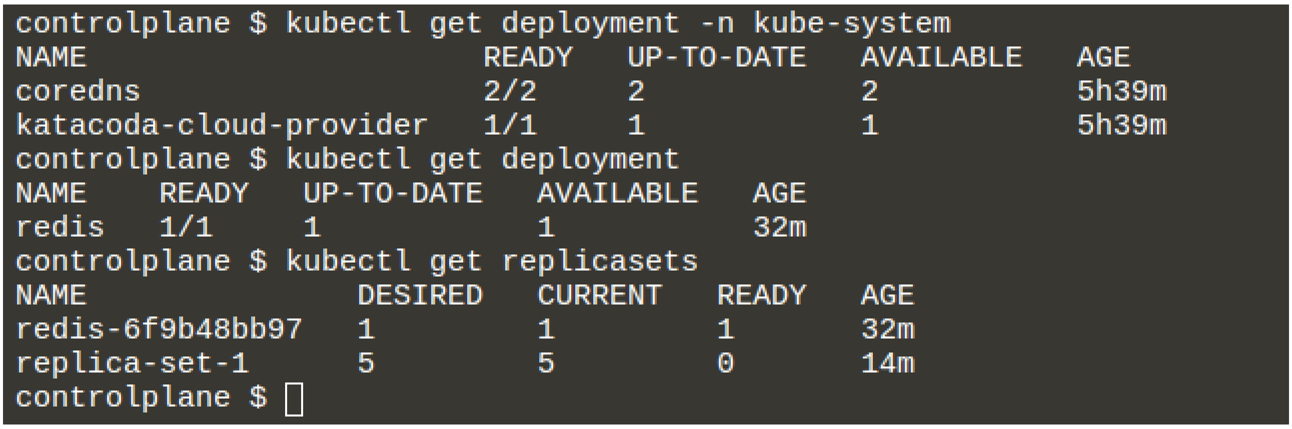
```
controlplane $ kubectl get pods
NAME                     READY   STATUS             RESTARTS   AGE
nginx                    1/1     Running            0          18m
redis-6f9b48bb97-v57zw   1/1     Running            0          22m
replica-set-1-69sm2      0/1     Completed          5          3m28s
replica-set-1-fqqwz      0/1     CrashLoopBackOff   3          86s
replica-set-1-hjdw2      0/1     Completed          5          3m28s
replica-set-1-kf4ck      0/1     CrashLoopBackOff   5          3m28s
replica-set-1-mwb28      0/1     CrashLoopBackOff   3          86s
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?



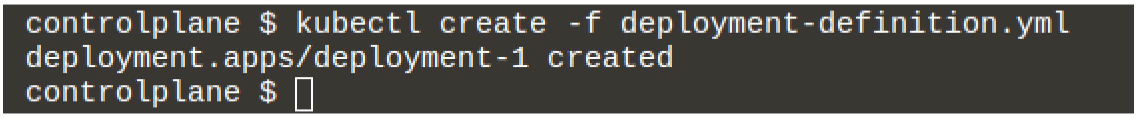
```
controlplane $ kubectl get pods
NAME                        READY   STATUS             RESTARTS   AGE
nginx                       1/1     Running            0          20m
redis-6f9b48bb97-v57zw      1/1     Running            0          23m
replica-set-1-69sm2         0/1     CrashLoopBackOff   5          5m
replica-set-1-d5r2g         0/1     Completed          2          25s
replica-set-1-fqqwz         0/1     CrashLoopBackOff   4          2m58s
replica-set-1-hjdw2         0/1     CrashLoopBackOff   5          5m
replica-set-1-kf4ck         0/1     CrashLoopBackOff   5          5m
controlplane $ kubectl delete pod replica-set-1-kf4ck
pod "replica-set-1-kf4ck" deleted
controlplane $ kubectl get pods
NAME                        READY   STATUS             RESTARTS   AGE
nginx                       1/1     Running            0          25m
redis-6f9b48bb97-v57zw      1/1     Running            0          28m
replica-set-1-69sm2         0/1     CrashLoopBackOff   6          10m
replica-set-1-d5r2g         0/1     CrashLoopBackOff   5          5m36s
replica-set-1-fqqwz         0/1     CrashLoopBackOff   6          8m9s
replica-set-1-hjdw2         0/1     CrashLoopBackOff   6          10m
replica-set-1-n75fw         0/1     CrashLoopBackOff   1          16s
controlplane $ 
```

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

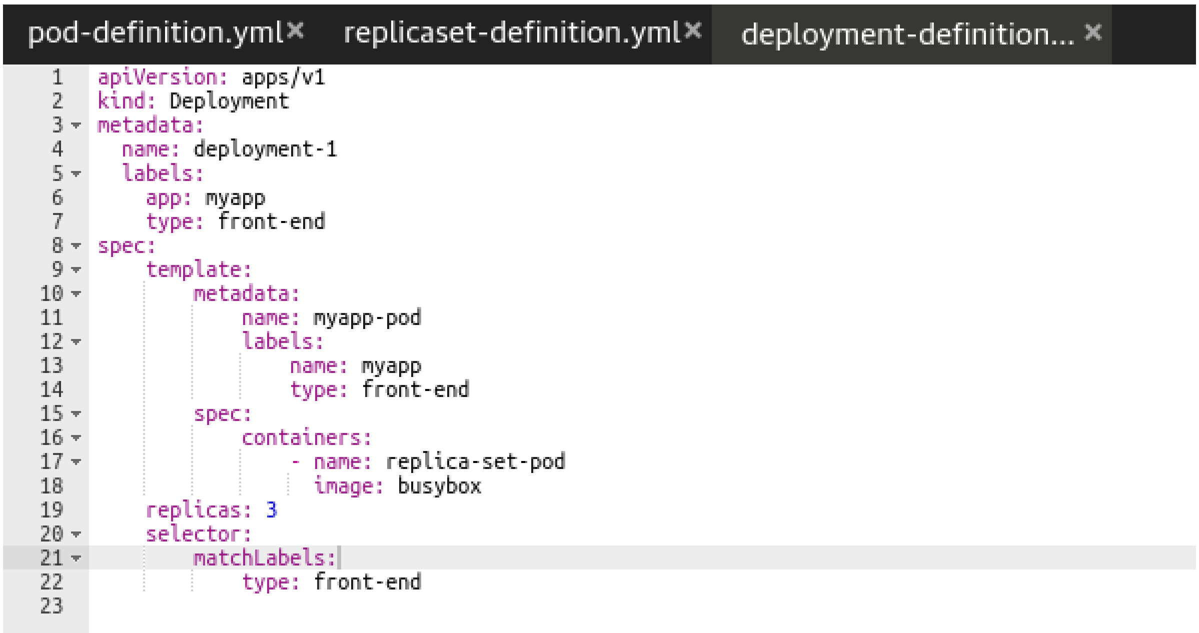


```
controlplane $ kubectl get deployment -n kube-system
NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
coredns                    2/2     2            2           5h39m
katacoda-cloud-provider    1/1     1            1           5h39m
controlplane $ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
redis   1/1     1            1           32m
controlplane $ kubectl get replicasets
NAME                 DESIRED   CURRENT   READY   AGE
redis-6f9b48bb97     1         1         1       32m
replica-set-1        5         5         0       14m
controlplane $ 
```

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

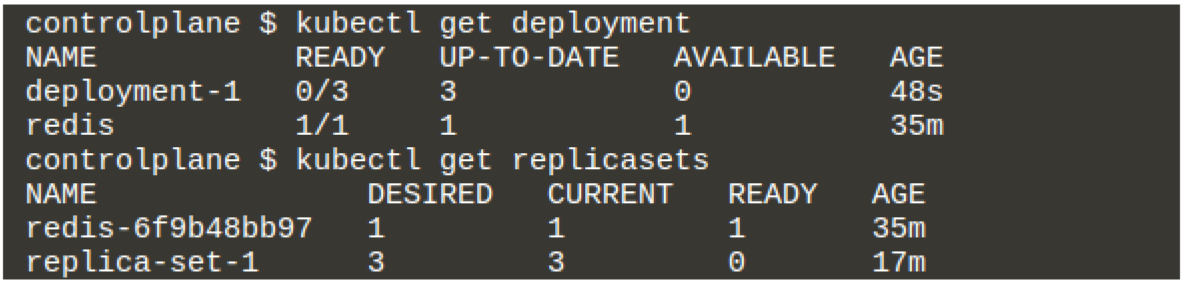


```
controlplane $ kubectl create -f deployment-definition.yml
deployment.apps/deployment-1 created
controlplane $ 
```



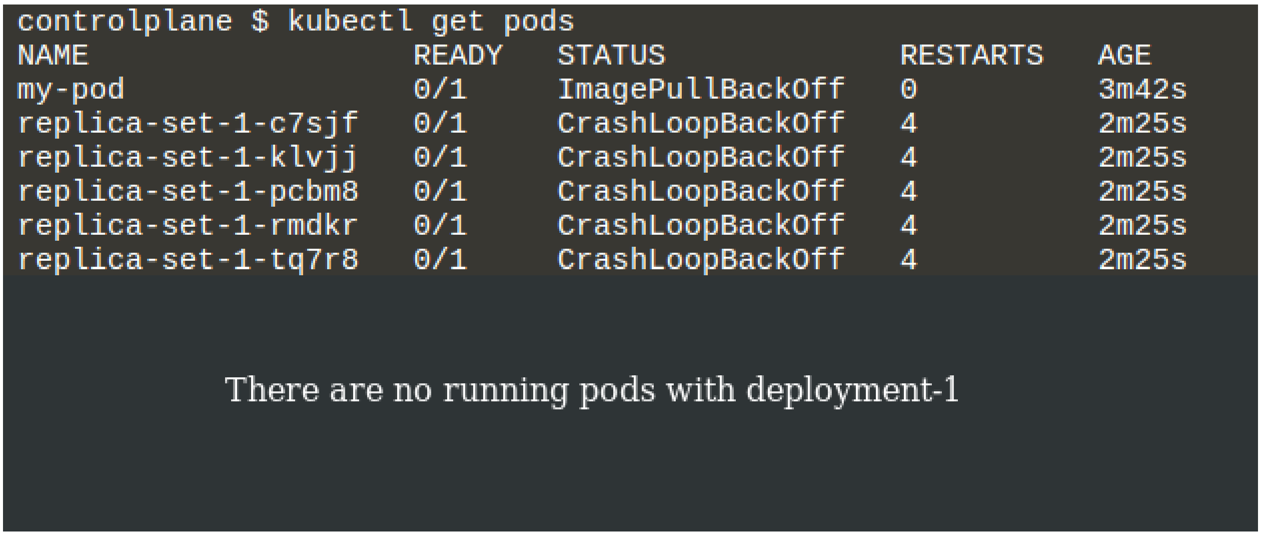
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
  labels:
    app: myapp
    type: front-end
spec:
    template:
        metadata:
            name: myapp-pod
            labels:
                name: myapp
                type: front-end
        spec:
            containers:
                - name: replica-set-pod
                  image: busybox
    replicas: 3
    selector:
        matchLabels:
            type: front-end

```

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



```
controlplane $ kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   0/3     3            0           48s
redis          1/1     1            1           35m
controlplane $ kubectl get replicasets
NAME               DESIRED   CURRENT   READY   AGE
redis-6f9b48bb97   1         1         1       35m
replica-set-1      3         3         0       17m
```

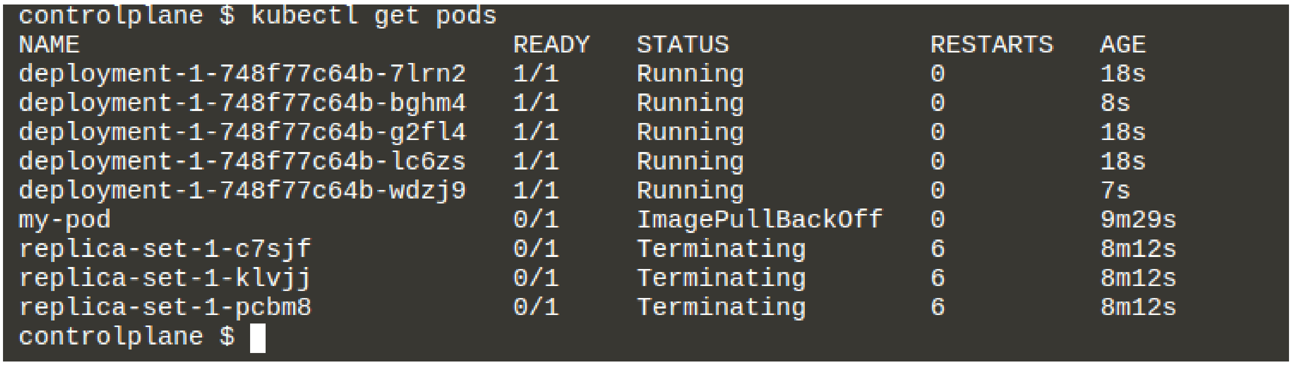
14- How many pods are ready with the deployment-1?



```
controlplane $ kubectl get pods
NAME                  READY   STATUS             RESTARTS   AGE
my-pod                0/1     ImagePullBackOff   0          3m42s
replica-set-1-c7sjf   0/1     CrashLoopBackOff   4          2m25s
replica-set-1-klvjj   0/1     CrashLoopBackOff   4          2m25s
replica-set-1-pcbm8   0/1     CrashLoopBackOff   4          2m25s
replica-set-1-rmdkr   0/1     CrashLoopBackOff   4          2m25s
replica-set-1-tq7r8   0/1     CrashLoopBackOff   4          2m25s
```

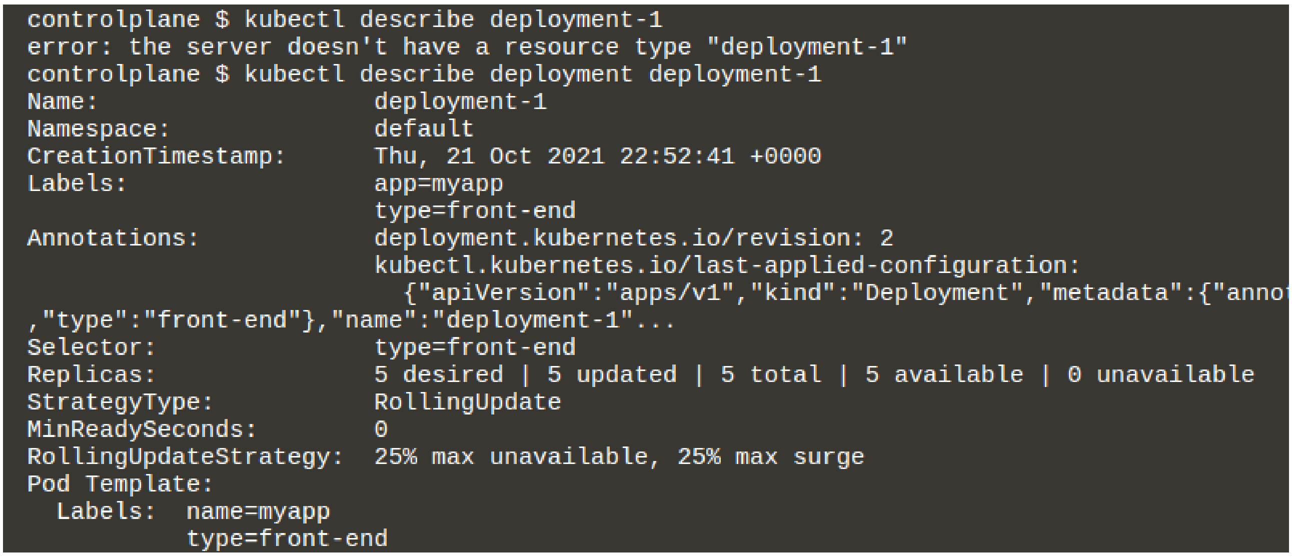
There are no running pods with deployment-1

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



```
controlplane $ kubectl get pods
NAME                            READY   STATUS             RESTARTS   AGE
deployment-1-748f77c64b-7lrn2   1/1     Running            0          18s
deployment-1-748f77c64b-bghm4   1/1     Running            0          8s
deployment-1-748f77c64b-g2fl4   1/1     Running            0          18s
deployment-1-748f77c64b-lc6zs   1/1     Running            0          18s
deployment-1-748f77c64b-wdzj9   1/1     Running            0          7s
my-pod                          0/1     ImagePullBackOff   0          9m29s
replica-set-1-c7sjf             0/1     Terminating        6          8m12s
replica-set-1-klvjj             0/1     Terminating        6          8m12s
replica-set-1-pcbm8             0/1     Terminating        6          8m12s
controlplane $ 
```

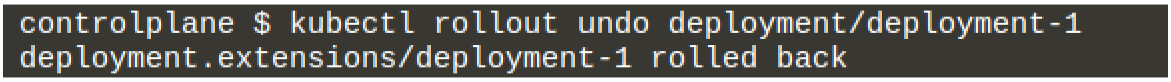
16- Run kubectl describe deployment deployment-1 and check events
What is the deployment strategy used to upgrade the deployment-1?



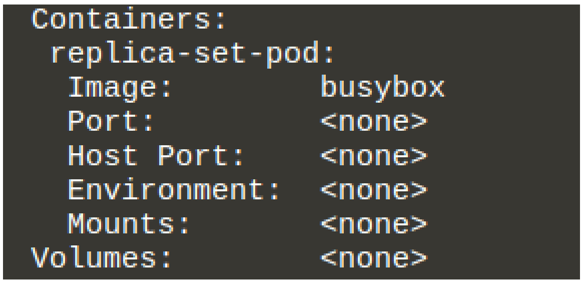
```
controlplane $ kubectl describe deployment-1
error: the server doesn't have a resource type "deployment-1"
controlplane $ kubectl describe deployment deployment-1
Name:                   deployment-1
Namespace:              default
CreationTimestamp:      Thu, 21 Oct 2021 22:52:41 +0000
Labels:                 app=myapp
                        type=front-end
Annotations:            deployment.kubernetes.io/revision: 2
                        kubectl.kubernetes.io/last-applied-configuration:
                          {"apiVersion":"apps/v1","kind":"Deployment","metadata":{"anno
,"type":"front-end"},"name":"deployment-1"...
Selector:               type=front-end
Replicas:               5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  name=myapp
           type=front-end
```

17- Rollback the deployment-1

What is the used image with the deployment-1?

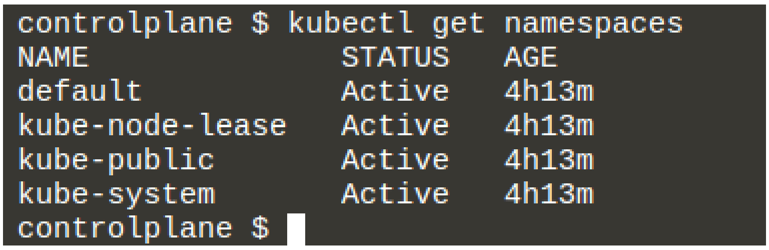


```
controlplane $ kubectl rollout undo deployment/deployment-1
deployment.extensions/deployment-1 rolled back
```



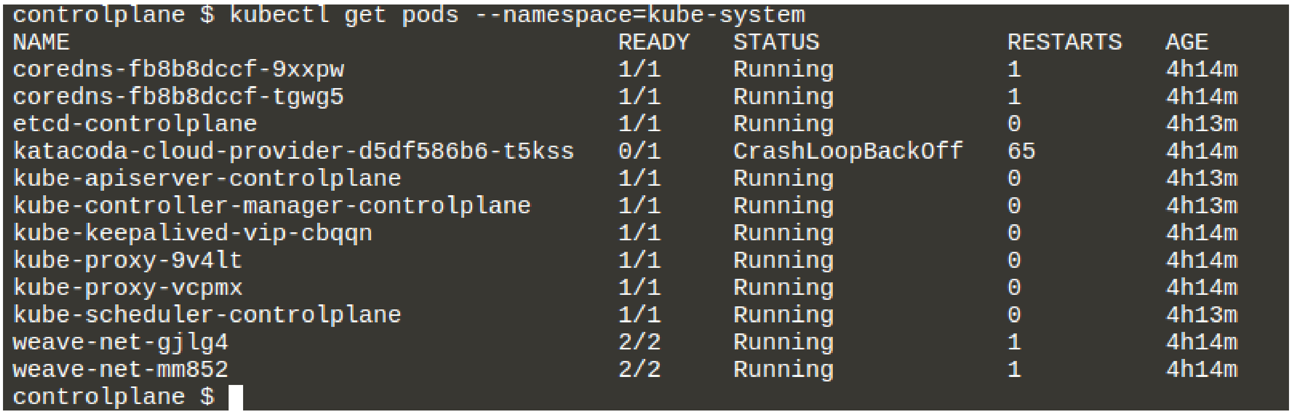
```
Containers:
 replica-set-pod:
  Image:        busybox
  Port:         <none>
  Host Port:    <none>
  Environment:  <none>
  Mounts:       <none>
Volumes:        <none>
```

18- How many Namespaces exist on the system?



```
controlplane $ kubectl get namespaces
NAME              STATUS   AGE
default           Active   4h13m
kube-node-lease   Active   4h13m
kube-public       Active   4h13m
kube-system       Active   4h13m
controlplane $
```

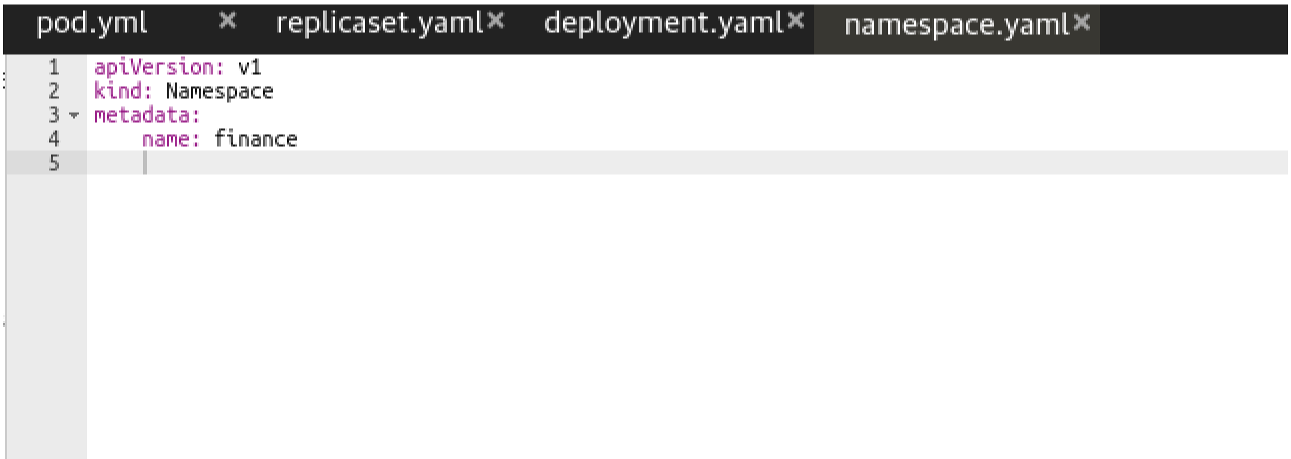
19- How many pods exist in the kube-system namespace?



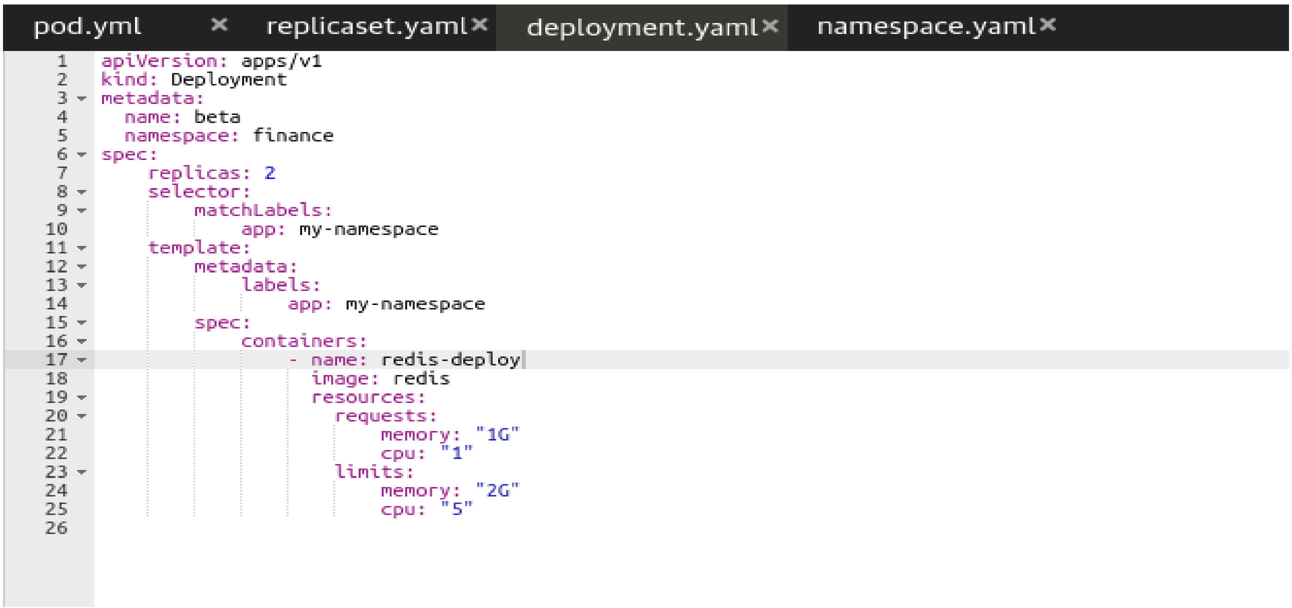
```
controlplane $ kubectl get pods --namespace=kube-system
NAME                                    READY   STATUS             RESTARTS   AGE
coredns-fb8b8dccf-9xxpw                 1/1     Running            1          4h14m
coredns-fb8b8dccf-tgwg5                 1/1     Running            1          4h14m
etcd-controlplane                       1/1     Running            0          4h13m
katacoda-cloud-provider-d5df586b6-t5kss 0/1     CrashLoopBackOff   65         4h14m
kube-apiserver-controlplane             1/1     Running            0          4h13m
kube-controller-manager-controlplane    1/1     Running            0          4h13m
kube-keepalived-vip-cbqqn               1/1     Running            0          4h14m
kube-proxy-9v4lt                        1/1     Running            0          4h14m
kube-proxy-vcpmx                        1/1     Running            0          4h14m
kube-scheduler-controlplane             1/1     Running            0          4h13m
weave-net-gjlg4                         2/2     Running            1          4h14m
weave-net-mm852                         2/2     Running            1          4h14m
controlplane $
```

20- Create a deployment with

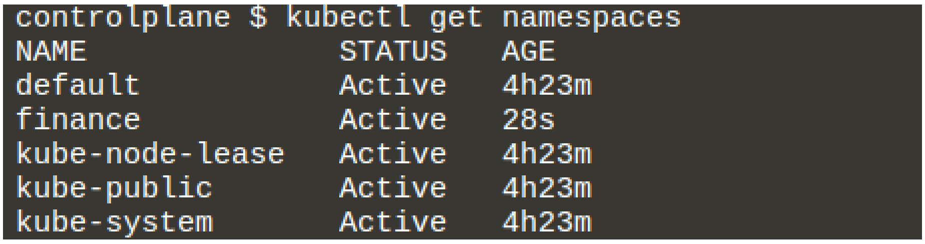
Name: beta
Image: redis
Replicas: 2
Namespace: finance
Resources Requests:
CPU: .5 vcpu
Mem: 1G
Resources Limits:
CPU: 1 vcpu
Mem: 2G



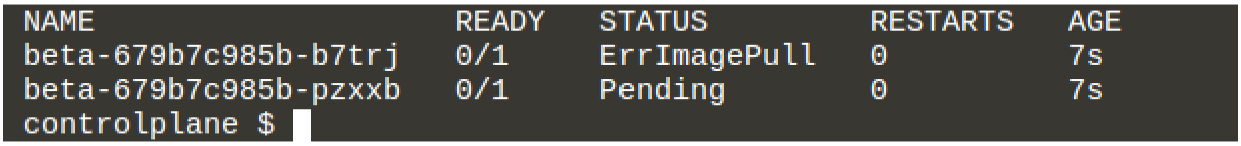
```yaml
apiVersion: v1
kind: Namespace
metadata:
    name: finance

```



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: beta
  namespace: finance
spec:
    replicas: 2
    selector:
        matchLabels:
            app: my-namespace
    template:
        metadata:
            labels:
                app: my-namespace
        spec:
            containers:
                - name: redis-deploy
                  image: redis
                  resources:
                    requests:
                        memory: "1G"
                        cpu: "1"
                    limits:
                        memory: "2G"
                        cpu: "5"

```



```
controlplane $ kubectl get namespaces
NAME                 STATUS   AGE
default              Active   4h23m
finance              Active   28s
kube-node-lease      Active   4h23m
kube-public          Active   4h23m
kube-system          Active   4h23m
```



```
NAME                    READY   STATUS         RESTARTS   AGE
beta-679b7c985b-b7trj   0/1     ErrImagePull   0          7s
beta-679b7c985b-pzxxb   0/1     Pending        0          7s
controlplane $ 
```