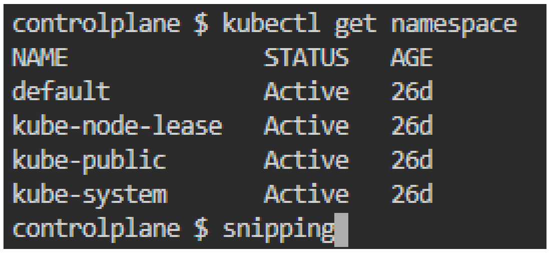
Kubernetes

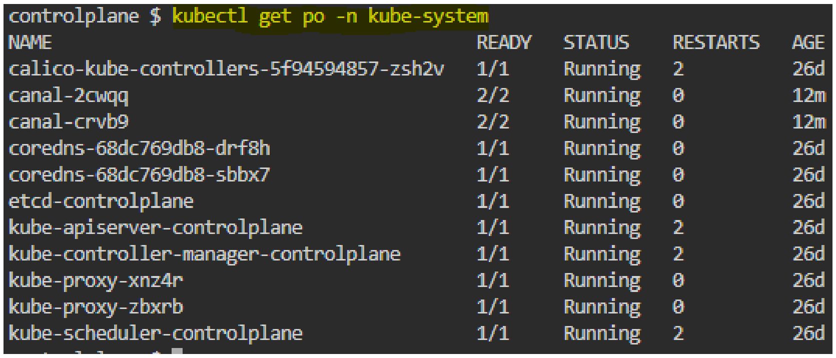
Lab_2

1- How many Namespaces exist on the system? *4 namespaces*



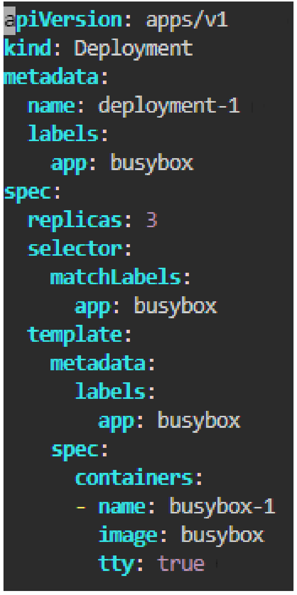
```
controlplane $ kubectl get namespace
NAME              STATUS   AGE
default           Active   26d
kube-node-lease   Active   26d
kube-public       Active   26d
kube-system       Active   26d
controlplane $ snipping
```

2-How many pods exist in the kube-system namespace? *11 pods*

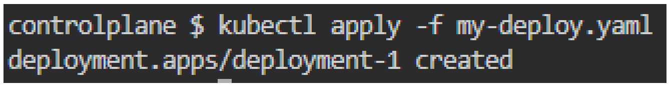


```
controlplane $ kubectl get po -n kube-system
NAME                                       READY   STATUS    RESTARTS   AGE
calico-kube-controllers-5f94594857-zsh2v   1/1     Running   2          26d
canal-2cwqq                                2/2     Running   0          12m
canal-crvb9                                2/2     Running   0          12m
coredns-68dc769db8-drf8h                   1/1     Running   0          26d
coredns-68dc769db8-sbbx7                   1/1     Running   0          26d
etcd-controlplane                          1/1     Running   0          26d
kube-apiserver-controlplane                1/1     Running   2          26d
kube-controller-manager-controlplane       1/1     Running   2          26d
kube-proxy-xnz4r                           1/1     Running   0          26d
kube-proxy-zbxrb                           1/1     Running   0          26d
kube-scheduler-controlplane                1/1     Running   2          26d
```

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

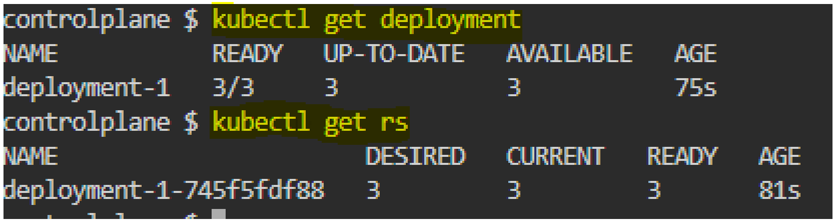


```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
  labels:
    app: busybox
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox-1
        image: busybox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



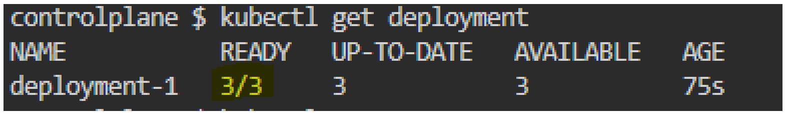
```
controlplane $ kubectl apply -f my-deploy.yaml
deployment.apps/deployment-1 created
```

4- How many Deployments and ReplicaSets exist on the system now? *1 deployment & 1 ReplicaSet*



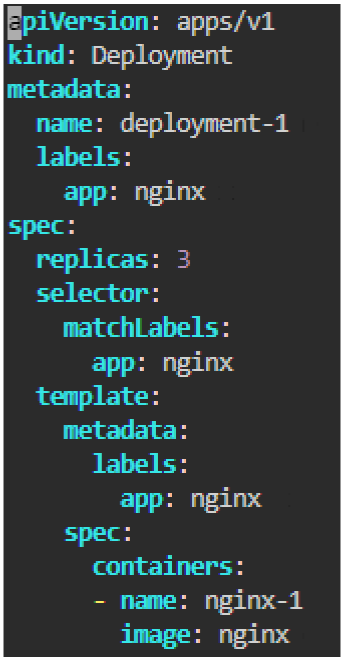
```
controlplane $ kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           75s
controlplane $ kubectl get rs
NAME                      DESIRED   CURRENT   READY   AGE
deployment-1-745f5fdf88   3         3         3       81s
```

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

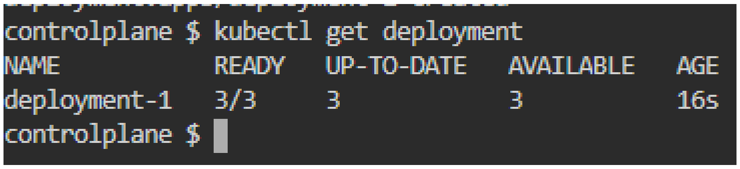


```
controlplane $ kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           75s
```

6- Update deployment-1 image to nginx then check the ready pods again? 3 pods

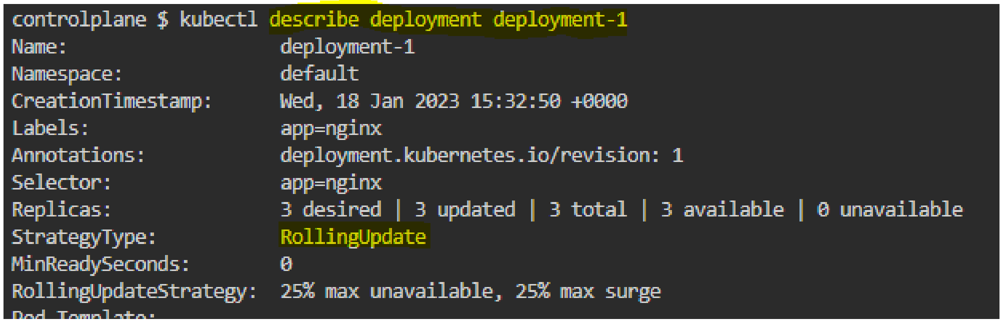


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-1
        image: nginx
```

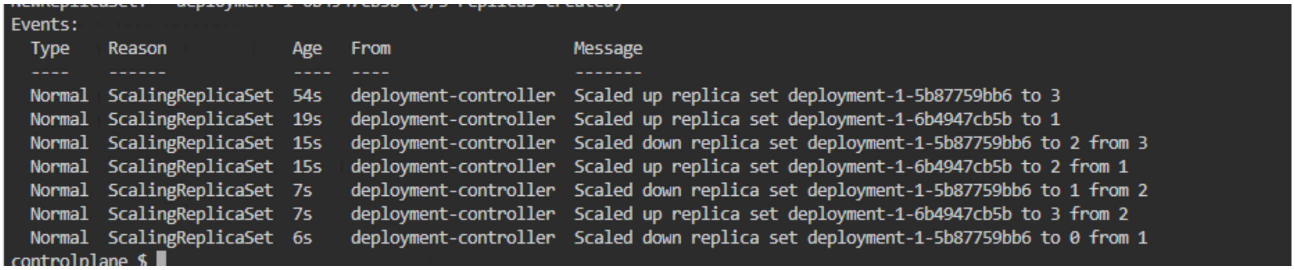


```
controlplane $ kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           16s
controlplane $ 
```

7- Run kubectl describe deployment deployment-1 and check events. What is the deployment strategy used to upgrade deployment-1? *Rolling update*

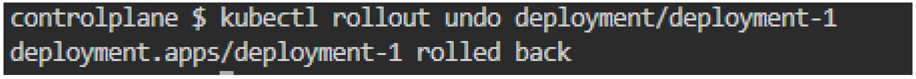


```
controlplane $ kubectl describe deployment deployment-1
Name:                   deployment-1
Namespace:              default
CreationTimestamp:      Wed, 18 Jan 2023 15:32:50 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
```



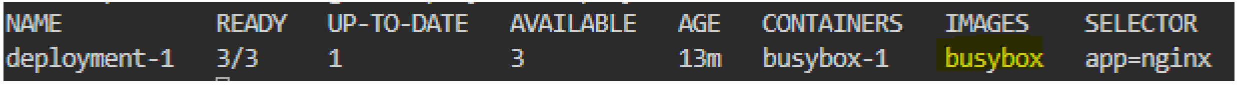
```
NewReplicaSet:   deployment-1-6b4947cb5b (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  54s   deployment-controller  Scaled up replica set deployment-1-5b87759bb6 to 3
  Normal  ScalingReplicaSet  19s   deployment-controller  Scaled up replica set deployment-1-6b4947cb5b to 1
  Normal  ScalingReplicaSet  15s   deployment-controller  Scaled down replica set deployment-1-5b87759bb6 to 2 from 3
  Normal  ScalingReplicaSet  15s   deployment-controller  Scaled up replica set deployment-1-6b4947cb5b to 2 from 1
  Normal  ScalingReplicaSet  7s    deployment-controller  Scaled down replica set deployment-1-5b87759bb6 to 1 from 2
  Normal  ScalingReplicaSet  7s    deployment-controller  Scaled up replica set deployment-1-6b4947cb5b to 3 from 2
  Normal  ScalingReplicaSet  6s    deployment-controller  Scaled down replica set deployment-1-5b87759bb6 to 0 from 1
controlplane $
```

8- Rollback the deployment-1



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

9- What is the used image with the deployment-1?



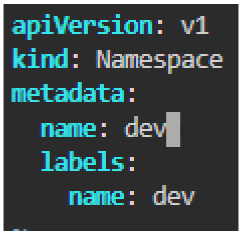
```
NAME           READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES    SELECTOR
deployment-1   3/3     1            3           13m   busybox-1    busybox   app=nginx
```

10- Create a deployment with

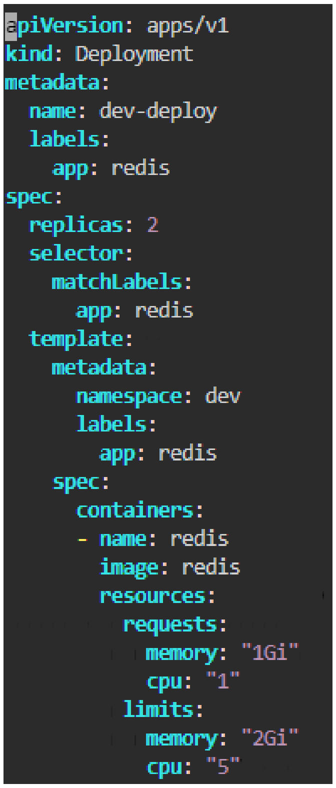
Name: dev-deploy Image: redis Replicas: 2 Namespace: dev

Resources Requests: CPU: .5 vcpu, Mem: 1G

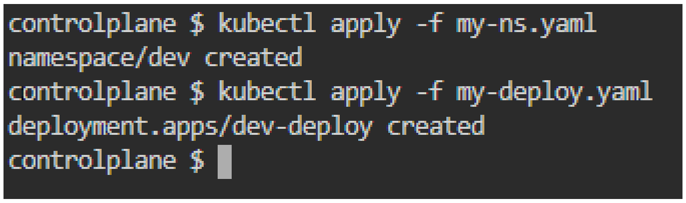
Resources Limits: CPU: 1 vcpu, Mem: 2G



```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: dev
  labels:
    name: dev
```



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



```
controlplane $ kubectl apply -f my-ns.yaml
namespace/dev created
controlplane $ kubectl apply -f my-deploy.yaml
deployment.apps/dev-deploy created
controlplane $
```