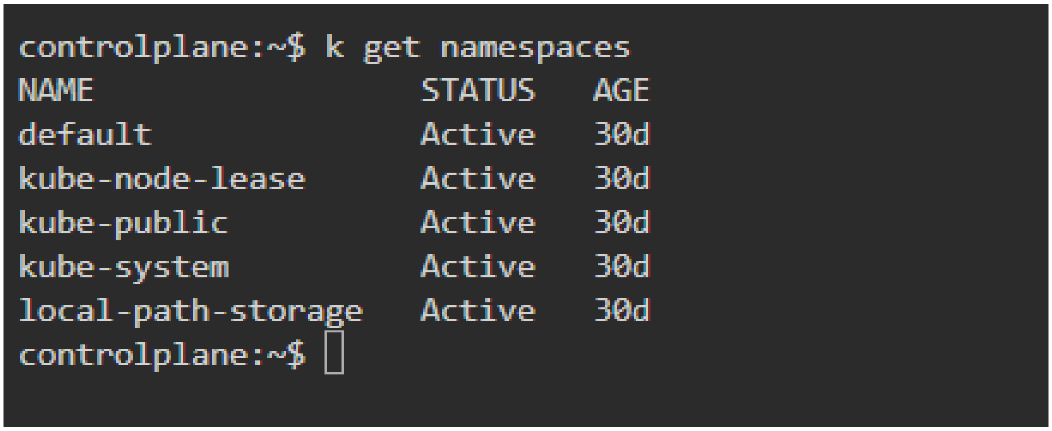
# Mohanad Mahmoud Sayed

# Lab2

## 1- How many Namespaces exist on the system?

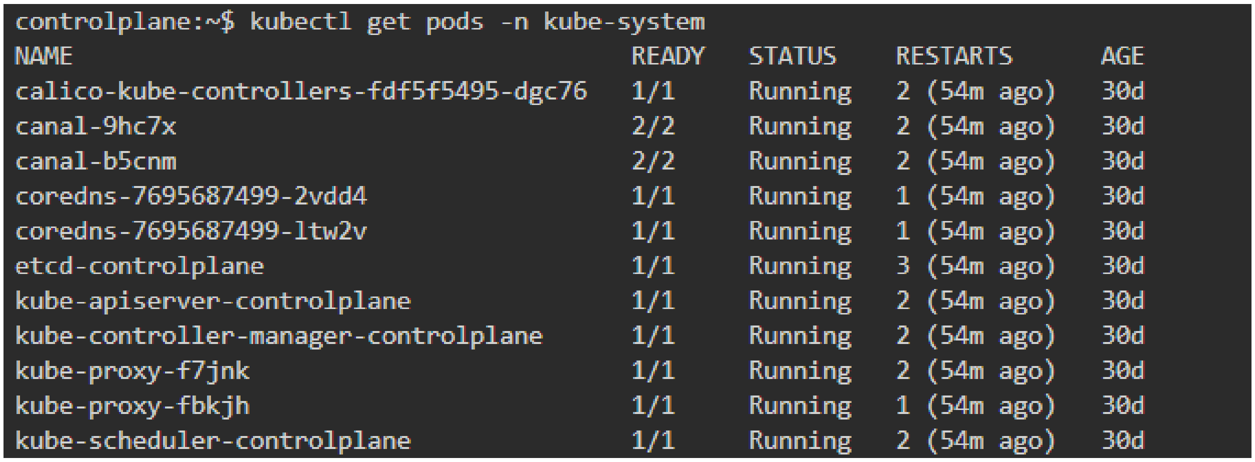
→There are 5 namespaces on the system



```
controlplane:~$ k get namespaces
NAME                 STATUS   AGE
default              Active   30d
kube-node-lease      Active   30d
kube-public          Active   30d
kube-system          Active   30d
local-path-storage   Active   30d
controlplane:~$ 
```

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

➔ There are 11 pods exist



```
controlplane:~$ kubectl get pods -n kube-system
NAME                                       READY   STATUS    RESTARTS      AGE
calico-kube-controllers-fdf5f5495-dgc76    1/1     Running   2 (54m ago)   30d
canal-9hc7x                                2/2     Running   2 (54m ago)   30d
canal-b5cnm                                2/2     Running   2 (54m ago)   30d
coredns-7695687499-2vdd4                   1/1     Running   1 (54m ago)   30d
coredns-7695687499-ltw2v                   1/1     Running   1 (54m ago)   30d
etcd-controlplane                          1/1     Running   3 (54m ago)   30d
kube-apiserver-controlplane                1/1     Running   2 (54m ago)   30d
kube-controller-manager-controlplane       1/1     Running   2 (54m ago)   30d
kube-proxy-f7jnk                           1/1     Running   2 (54m ago)   30d
kube-proxy-fbkjh                           1/1     Running   1 (54m ago)   30d
kube-scheduler-controlplane                1/1     Running   2 (54m ago)   30d
```

## 3- Create a deployment with

Name: beta

Image: redis

Replicas: 2

Namespace: finance

Resources Requests:

CPU: .5 vcpu

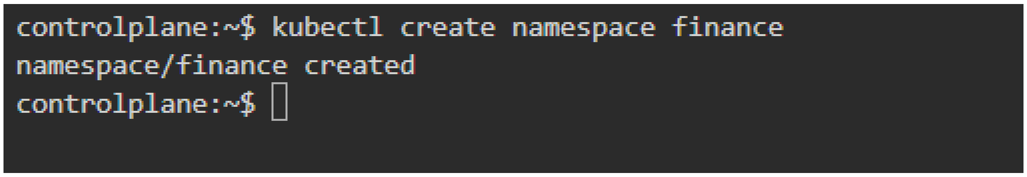
Mem: 1G

Resources Limits:

CPU: 1 vcpu

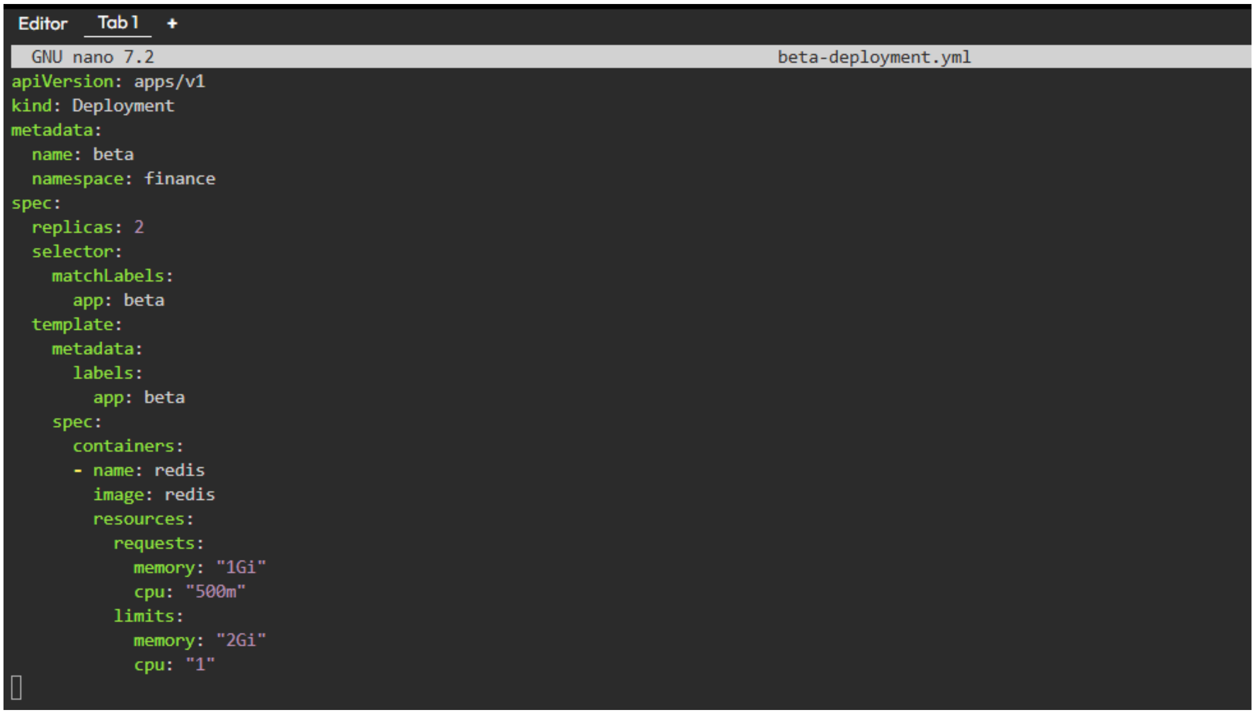
Mem: 2G

Firstly, I created a namespace its name finance



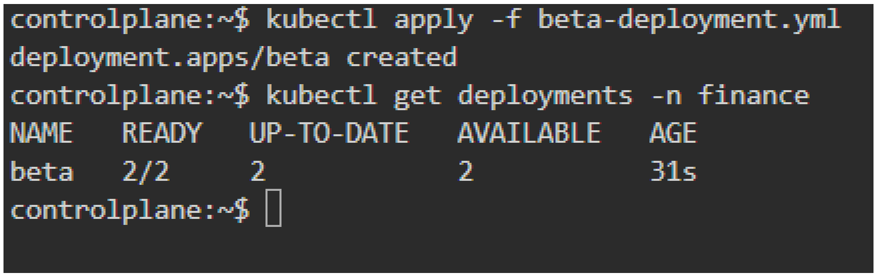
```
controlplane:~$ kubectl create namespace finance
namespace/finance created
controlplane:~$ 
```

Then, create beta-deployment.yml file:



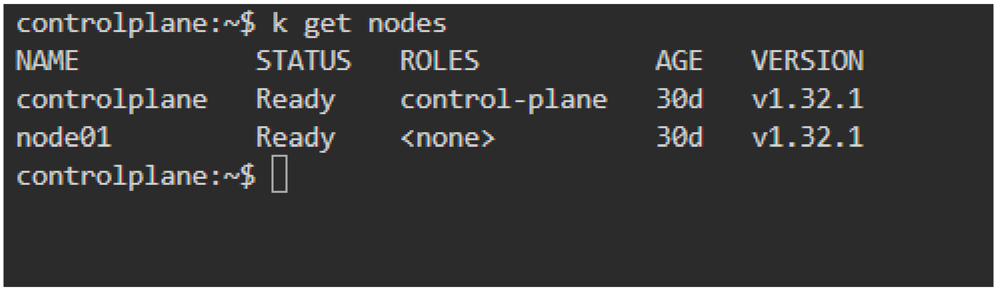
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: beta
  namespace: finance
spec:
  replicas: 2
  selector:
    matchLabels:
      app: beta
  template:
    metadata:
      labels:
        app: beta
    spec:
      containers:
      - name: redis
        image: redis
        resources:
          requests:
            memory: "1Gi"
            cpu: "500m"
          limits:
            memory: "2Gi"
            cpu: "1"
```

Finally, apply and check:



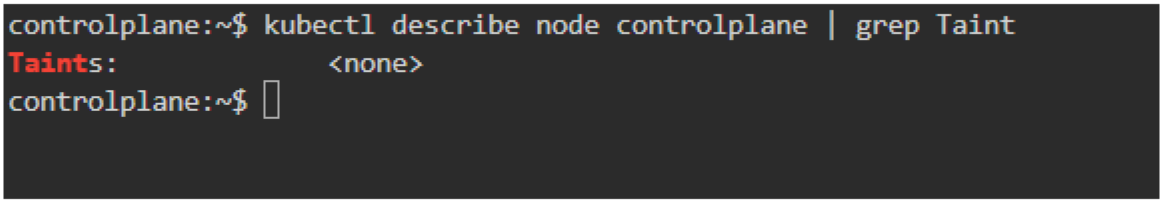
```
controlplane:~$ kubectl apply -f beta-deployment.yml
deployment.apps/beta created
controlplane:~$ kubectl get deployments -n finance
NAME   READY   UP-TO-DATE   AVAILABLE   AGE
beta   2/2     2            2           31s
controlplane:~$
```

## 4-How many Nodes exist on the system?



```
controlplane:~$ k get nodes
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   30d   v1.32.1
node01         Ready    <none>          30d   v1.32.1
controlplane:~$
```

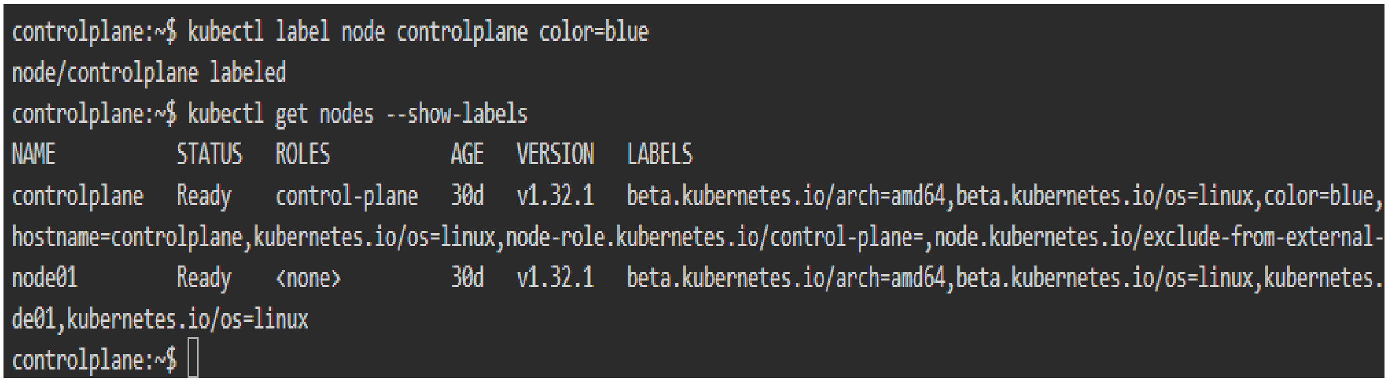
## 5- Do you see any taints on master?



```
controlplane:~$ kubectl describe node controlplane | grep Taint
Taints:             <none>
controlplane:~$ 
```

➔ master node has **no taints**

## 6- Apply a label color=blue to the master node



```
controlplane:~$ kubectl label node controlplane color=blue
node/controlplane labeled
controlplane:~$ kubectl get nodes --show-labels
NAME           STATUS   ROLES           AGE   VERSION   LABELS
controlplane   Ready    control-plane   30d   v1.32.1   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,color=blue,
hostname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/exclude-from-external-
node01         Ready    <none>          30d   v1.32.1   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.
de01,kubernetes.io/os=linux
controlplane:~$ 
```

7- Create a new deployment named blue with the nginx image and 3 replicas

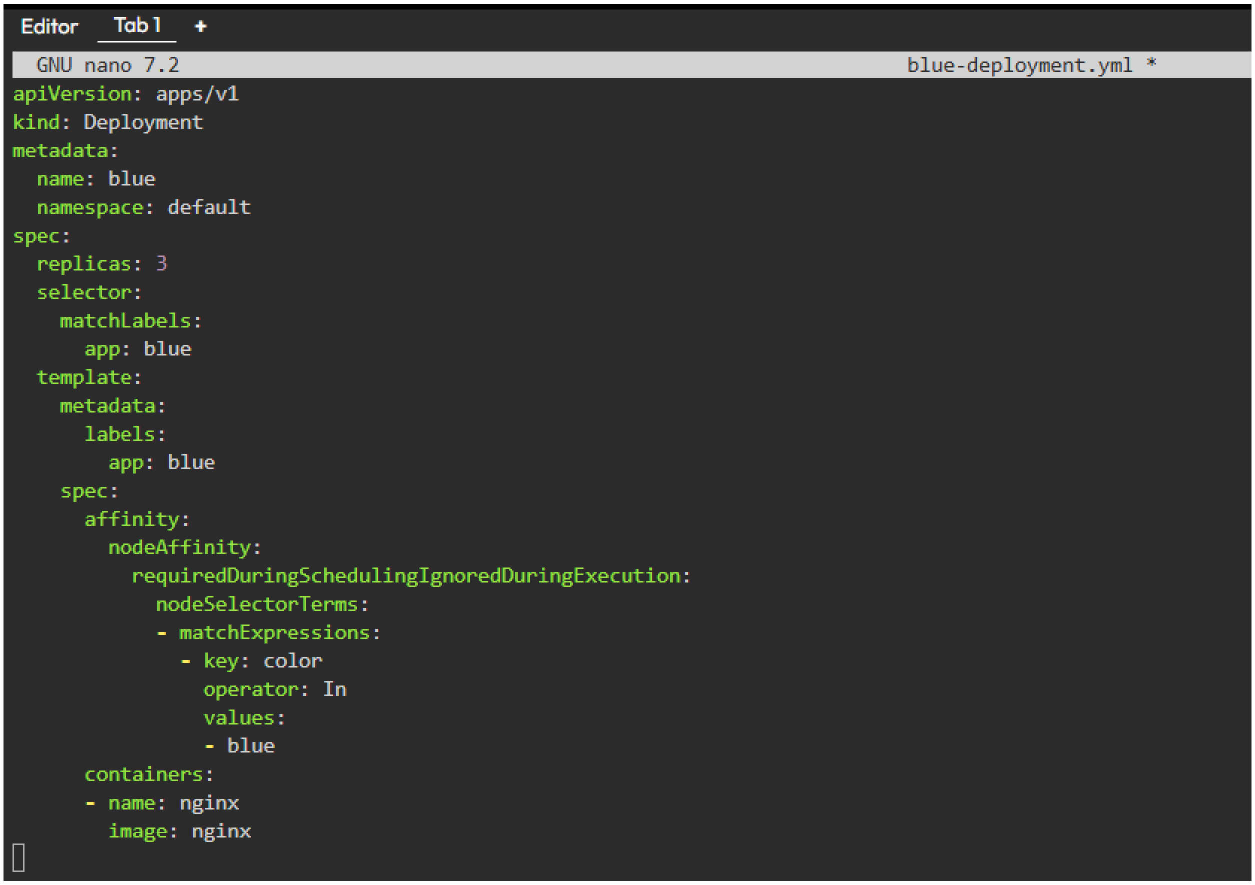
Set Node Affinity to the deployment to place the pods on master only

NodeAffinity: requiredDuringSchedulingIgnoredDuringExecution

Key: color

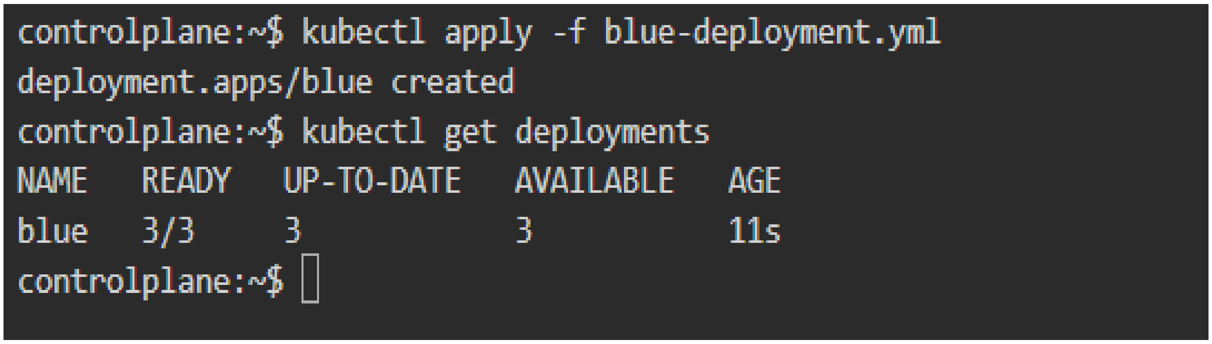
values: blue

Firstly, create blue-deployment.yml file:



```
  GNU nano 7.2                                                         blue-deployment.yml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue
  namespace: default
spec:
  replicas: 3
  selector:
    matchLabels:
      app: blue
  template:
    metadata:
      labels:
        app: blue
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: color
                operator: In
                values:
                - blue
      containers:
      - name: nginx
        image: nginx
```

Finally, apply and check:



```
controlplane:~$ kubectl apply -f blue-deployment.yml
deployment.apps/blue created
controlplane:~$ kubectl get deployments
NAME   READY   UP-TO-DATE   AVAILABLE   AGE
blue   3/3     3            3           11s
controlplane:~$ 
```