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

```
sabry@sabry-vm:~/ingress-app$ kubectl get namespace
NAME                 STATUS   AGE
default              Active   39h
ingress-nginx        Active   39h
kube-node-lease      Active   39h
kube-public          Active   39h
kube-system          Active   39h
local-path-storage   Active   39h
sabry@sabry-vm:~/ingress-app$ kubectl get ns --no-headers | wc -l
6
```

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

```
sabry@sabry-vm:~/ingress-app$ kubectl get pods -n kube-system
NAME                                         READY   STATUS    RESTARTS   AGE
coredns-668d6bf9bc-qbvsb                     1/1     Running   0          39h
coredns-668d6bf9bc-qfldk                     1/1     Running   0          39h
etcd-kind-control-plane                      1/1     Running   0          39h
kindnet-5k7vh                                1/1     Running   0          39h
kube-apiserver-kind-control-plane            1/1     Running   0          39h
kube-controller-manager-kind-control-plane   1/1     Running   0          39h
kube-proxy-jd9tc                             1/1     Running   0          39h
kube-scheduler-kind-control-plane            1/1     Running   0          39h
sabry@sabry-vm:~/ingress-app$ kubectl get pods -n kube-system --no-headers | wc -l
8
```

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

```
sabry@sabry-vm:~/ingress-app$ vim deployment.yaml
sabry@sabry-vm:~/ingress-app$ kubectl create namespace finance
namespace/finance created
sabry@sabry-vm:~/ingress-app$ kubectl apply -f deployment.yaml
deployment.apps/beta created
```

```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:
              cpu: "500m"
              memory: "1Gi"
            limits:
              cpu: "1"
              memory: "2Gi"
```

4- How many Nodes exist on the system?

```
sabry@sabry-vm:~/ingress-app$ kubectl get nodes
NAME                 STATUS   ROLES           AGE   VERSION
kind-control-plane   Ready    control-plane   39h   v1.32.2
```

5- Do you see any taints on master?

```
sabry@sabry-vm:~/ingress-app$ kubectl describe node kind-control-plane | grep Taint
Taints:             <none>
```

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

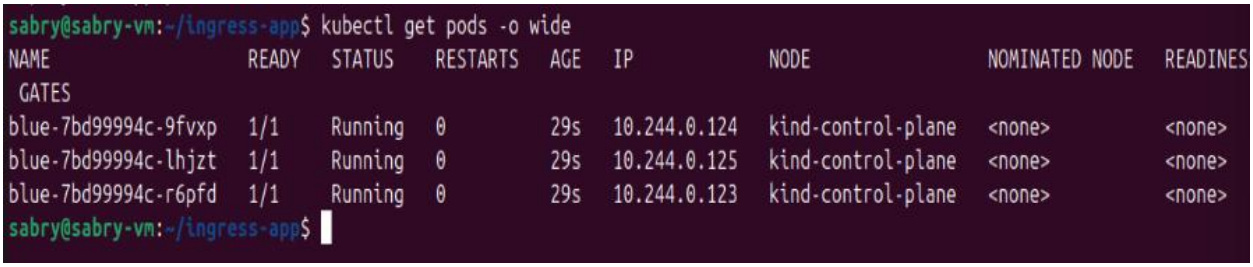
```
sabry@sabry-vm:~/ingress-app$ kubectl label node kind-control-plane color=blue
node/kind-control-plane labeled
```

```
sabry@sabry-vm:~/ingress-app$ kubectl get nodes --show-labels
NAME                 STATUS   ROLES           AGE   VERSION   LABELS
kind-control-plane   Ready    control-plane   39h   v1.32.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,
color=blue,kubernetes.io/arch=amd64,kubernetes.io/hostname=kind-control-plane,kubernetes.io/os=linux,node-role.kubernete
s.io/control-plane=
```

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

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue
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
```

```
sabry@sabry-vm:~/ingress-app$ vim blue-deployment.yaml
sabry@sabry-vm:~/ingress-app$ kubectl apply -f blue-deployment.yaml
deployment.apps/blue created
```


sabry@sabry-vm:~/ingress-app$ kubectl get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINES
GATES
blue-7bd99994c-9fvxp 1/1 Running 0 29s 10.244.0.124 kind-control-plane <none> <none>
blue-7bd99994c-lhjzt 1/1 Running 0 29s 10.244.0.125 kind-control-plane <none> <none>
blue-7bd99994c-r6pfd 1/1 Running 0 29s 10.244.0.123 kind-control-plane <none> <none>
sabry@sabry-vm:~/ingress-app$