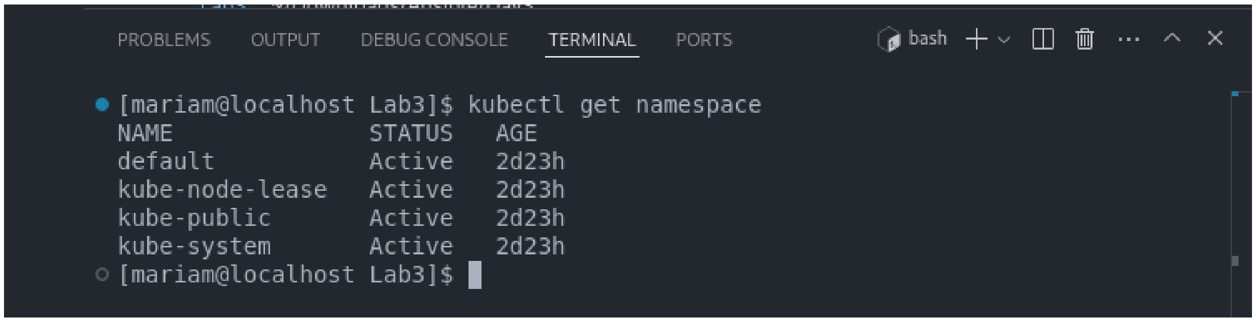
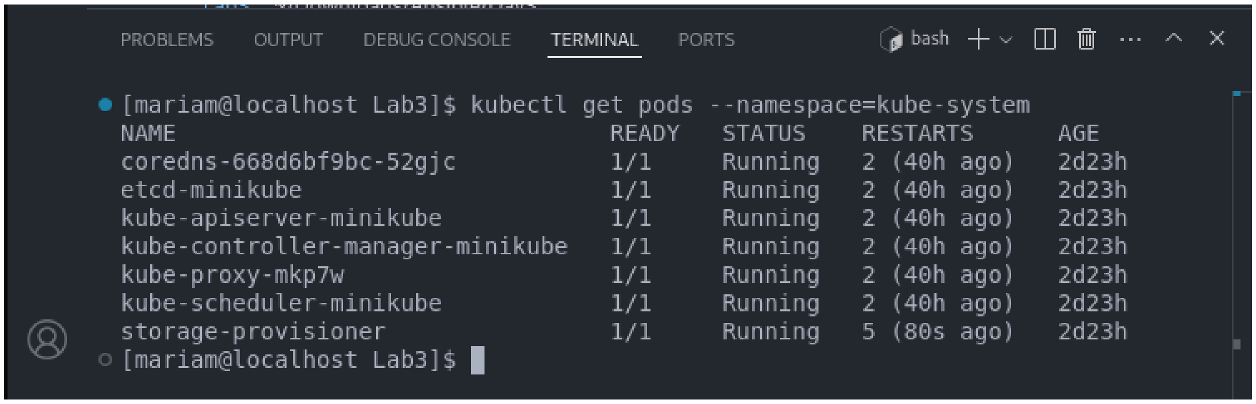
1. How many Namespaces exist on the system?



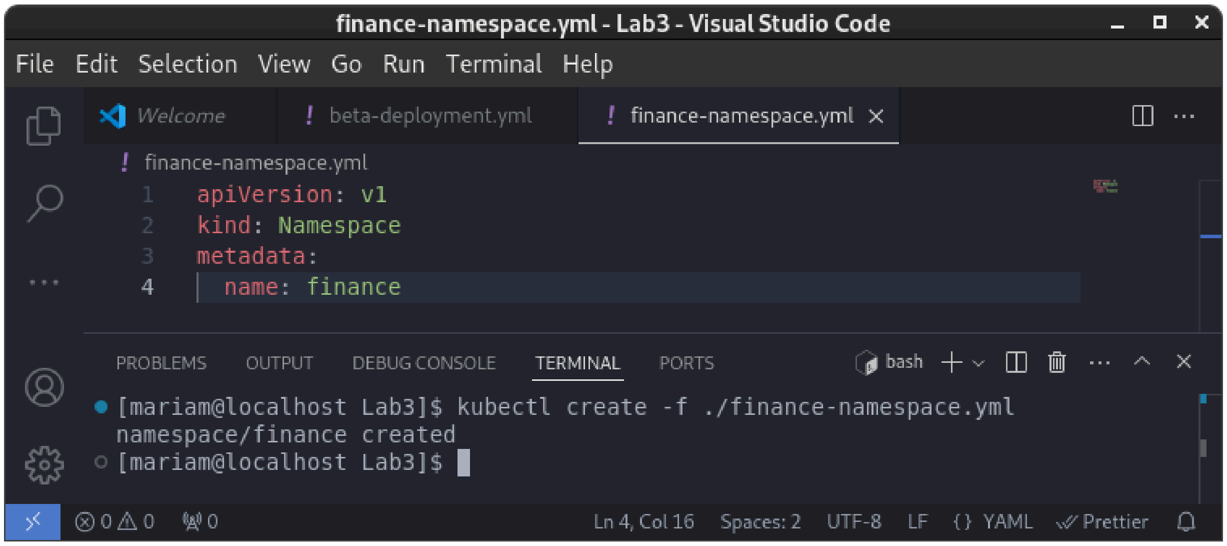
```
[mariam@localhost Lab3]$ kubectl get namespace
NAME              STATUS   AGE
default           Active   2d23h
kube-node-lease   Active   2d23h
kube-public       Active   2d23h
kube-system       Active   2d23h
[mariam@localhost Lab3]$
```

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



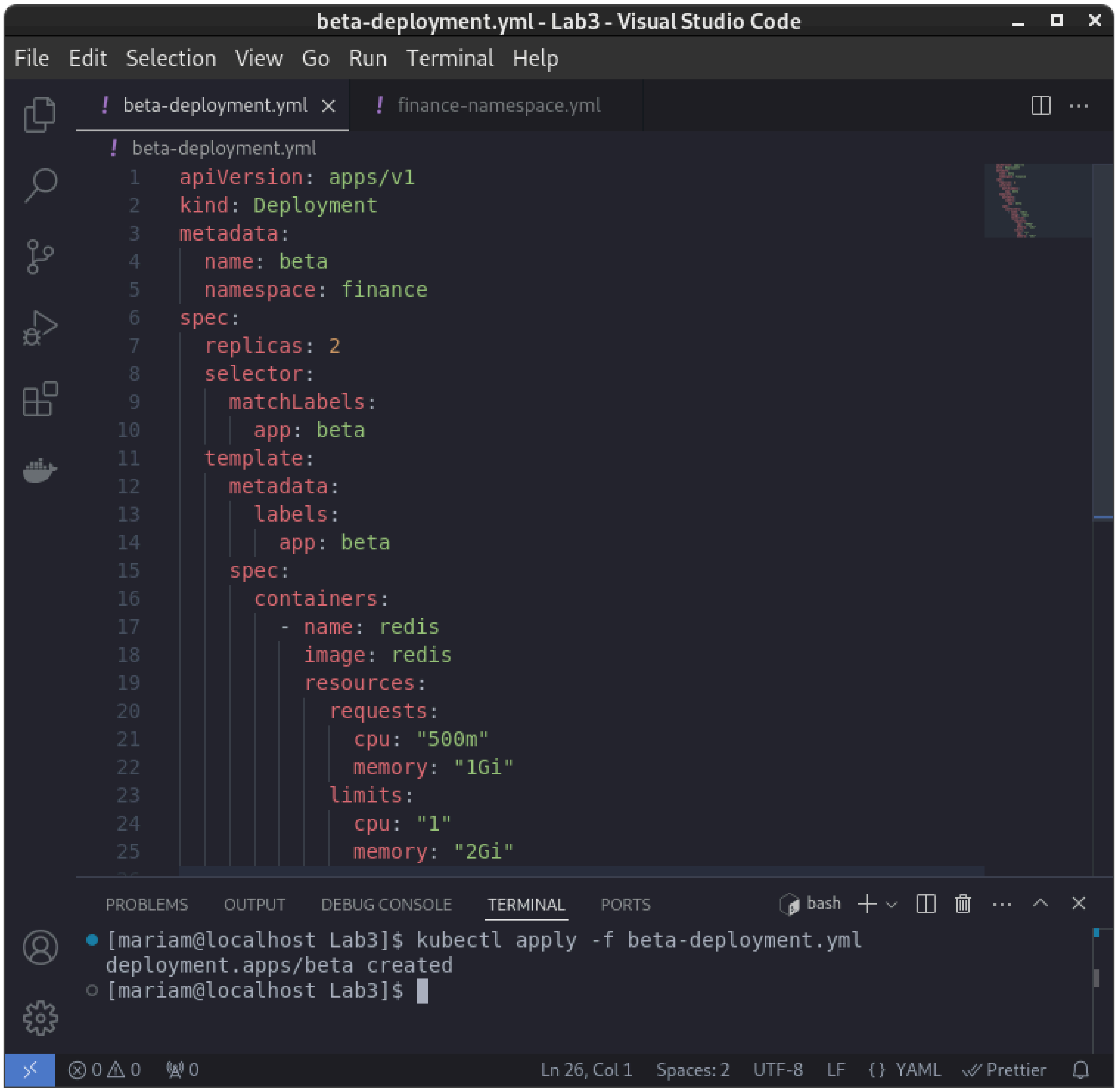
```
[mariam@localhost Lab3]$ kubectl get pods --namespace=kube-system
NAME                               READY   STATUS    RESTARTS      AGE
coredns-668d6bf9bc-52gjc           1/1     Running   2 (40h ago)   2d23h
etcd-minikube                      1/1     Running   2 (40h ago)   2d23h
kube-apiserver-minikube            1/1     Running   2 (40h ago)   2d23h
kube-controller-manager-minikube   1/1     Running   2 (40h ago)   2d23h
kube-proxy-mkp7w                   1/1     Running   2 (40h ago)   2d23h
kube-scheduler-minikube            1/1     Running   2 (40h ago)   2d23h
storage-provisioner                1/1     Running   5 (80s ago)   2d23h
[mariam@localhost Lab3]$
```

3. Create a deployment with:
➔ Name: beta
➔ Image: redis
➔ Replicas: 2
➔ Namespace: finance
➔ Resources Requests:
➔ CPU: 500m
➔ Mem: 1G
➔ Resources Limits:
➔ CPU: 1
➔ Mem: 2G


finance-namespace.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Welcome
beta-deployment.yml
finance-namespace.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 4, Col 16 Spaces: 2 UTF-8 LF YAML Prettier

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

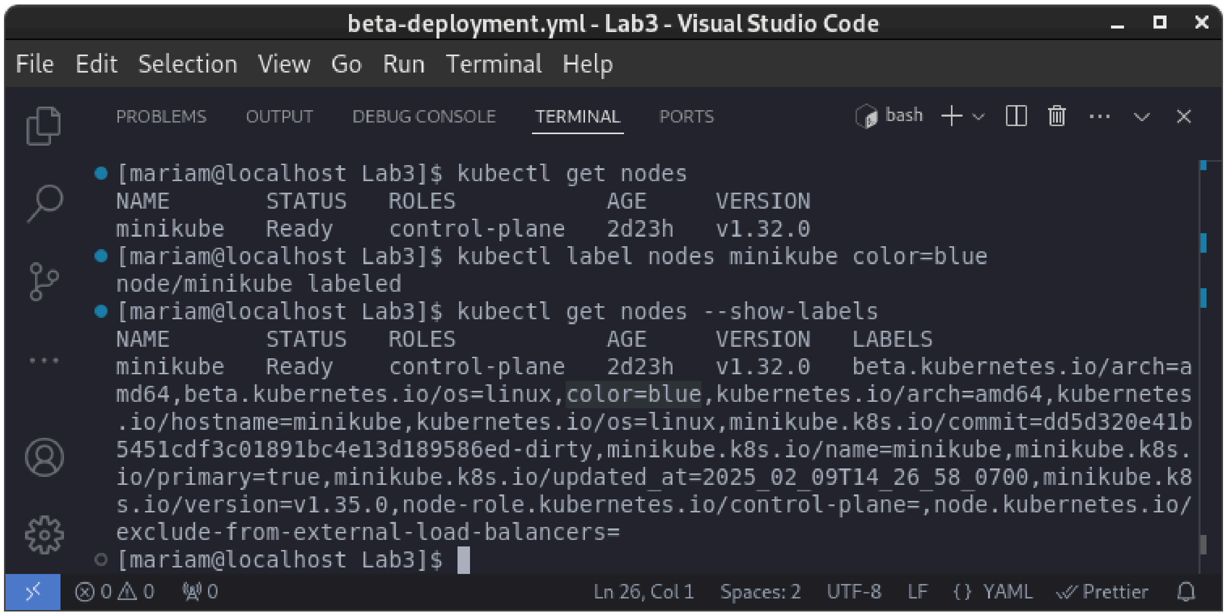
```
[mariam@localhost Lab3]$ kubectl create -f ./finance-namespace.yml
namespace/finance created
[mariam@localhost Lab3]$
```


beta-deployment.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
beta-deployment.yml
finance-namespace.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 26, Col 1 Spaces: 2 UTF-8 LF { } YAML Prettier

```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"
```

```
[mariam@localhost Lab3]$ kubectl apply -f beta-deployment.yml
deployment.apps/beta created
[mariam@localhost Lab3]$
```

4. Apply a label color=blue to the master node


beta-deployment.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 26, Col 1 Spaces: 2 UTF-8 LF { } YAML Prettier

```
[mariam@localhost Lab3]$ kubectl get nodes
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   2d23h   v1.32.0
[mariam@localhost Lab3]$ kubectl label nodes minikube color=blue
node/minikube labeled
[mariam@localhost Lab3]$ kubectl get nodes --show-labels
NAME       STATUS   ROLES           AGE     VERSION   LABELS
minikube   Ready    control-plane   2d23h   v1.32.0   beta.kubernetes.io/arch=a
md64,beta.kubernetes.io/os=linux,color=blue,kubernetes.io/arch=amd64,kubernetes
.io/hostname=minikube,kubernetes.io/os=linux,minikube.k8s.io/commit=dd5d320e41b
5451cdf3c01891bc4e13d189586ed-dirty,minikube.k8s.io/name=minikube,minikube.k8s.
io/primary=true,minikube.k8s.io/updated_at=2025_02_09T14_26_58_0700,minikube.k8
s.io/version=v1.35.0,node-role.kubernetes.io/control-plane=,node.kubernetes.io/
exclude-from-external-load-balancers=
[mariam@localhost Lab3]$
```

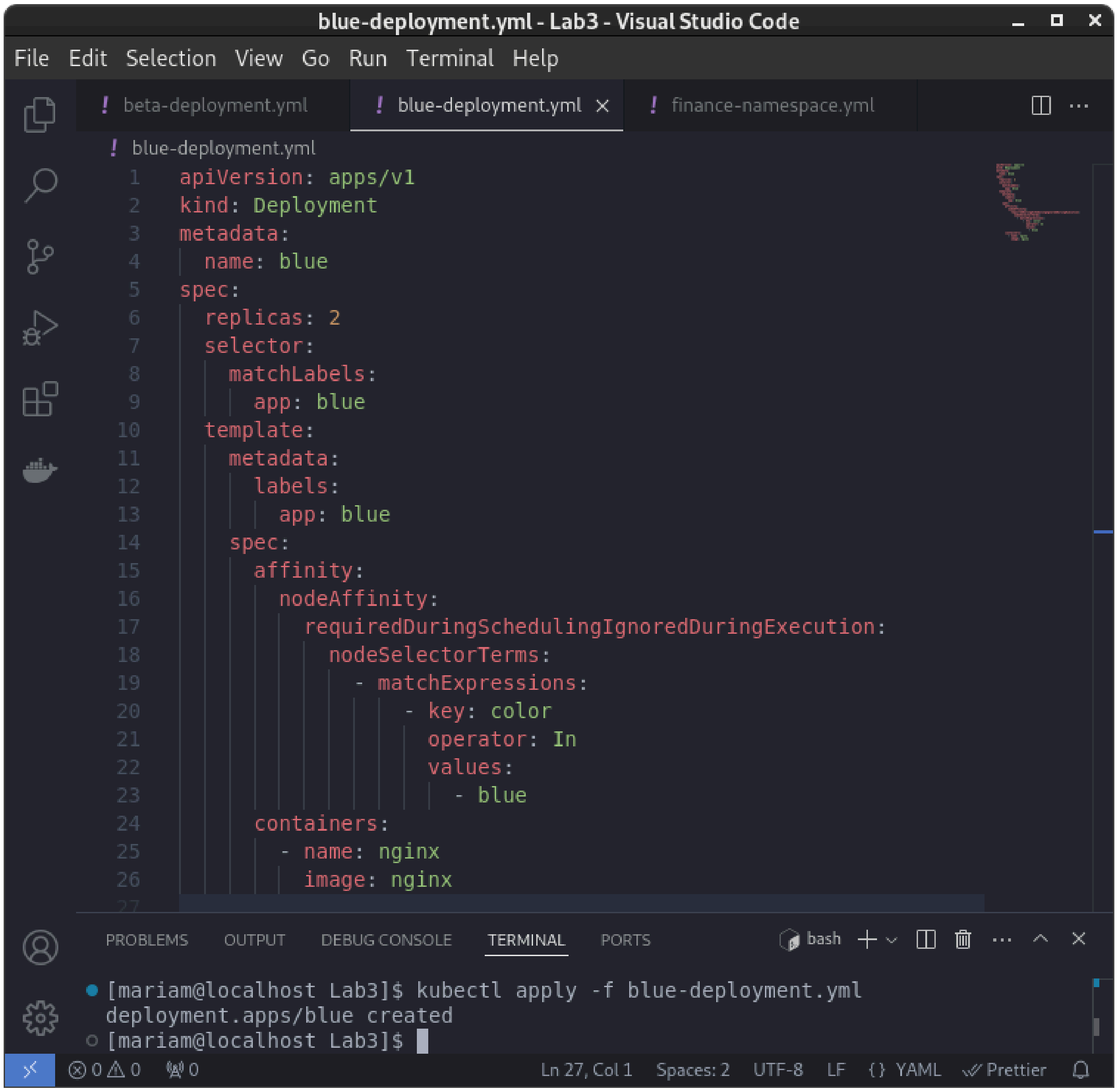
5. Create a new deployment named blue with the nginx image and 2 replicas

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

➔ NodeAffinity: requiredDuringSchedulingIgnoredDuringExecution

➔ Key: color

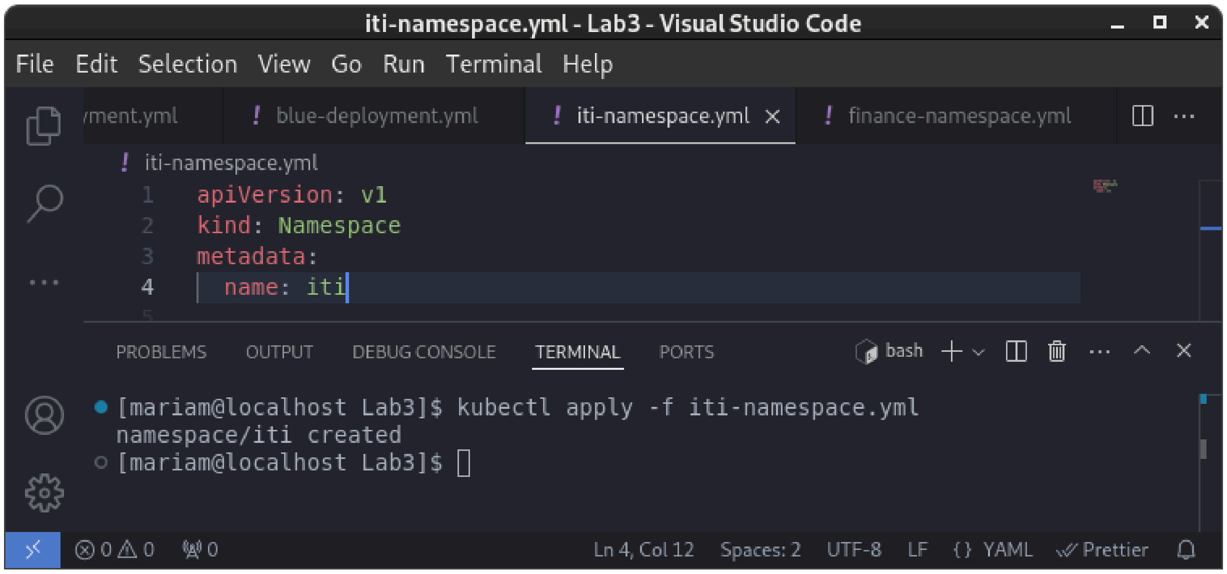
➔ values: blue


blue-deployment.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
beta-deployment.yml
blue-deployment.yml
finance-namespace.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 27, Col 1 Spaces: 2 UTF-8 LF {} YAML Prettier

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue
spec:
  replicas: 2
  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
```

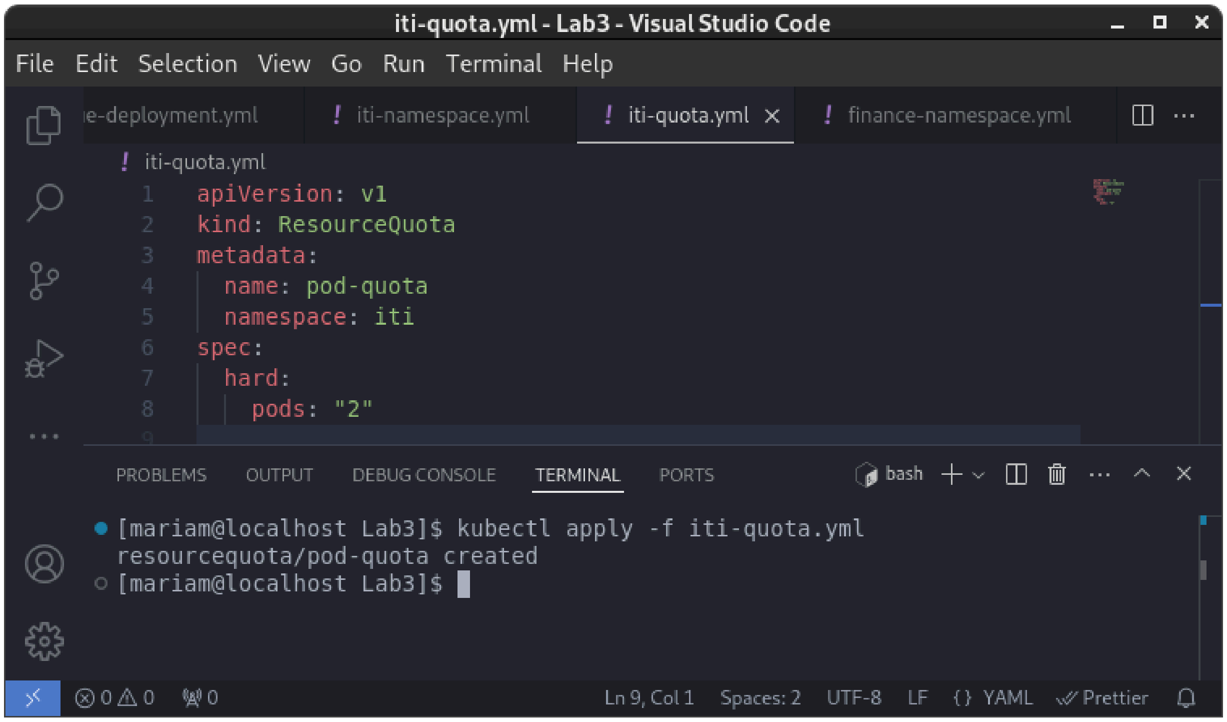
```
[mariam@localhost Lab3]$ kubectl apply -f blue-deployment.yml
deployment.apps/blue created
[mariam@localhost Lab3]$
```

6. Create a namespace named “iti” with a resource quota on pods “2”


iti-namespace.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ment.yml
blue-deployment.yml
iti-namespace.yml
finance-namespace.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 4, Col 12 Spaces: 2 UTF-8 LF {} YAML Prettier

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

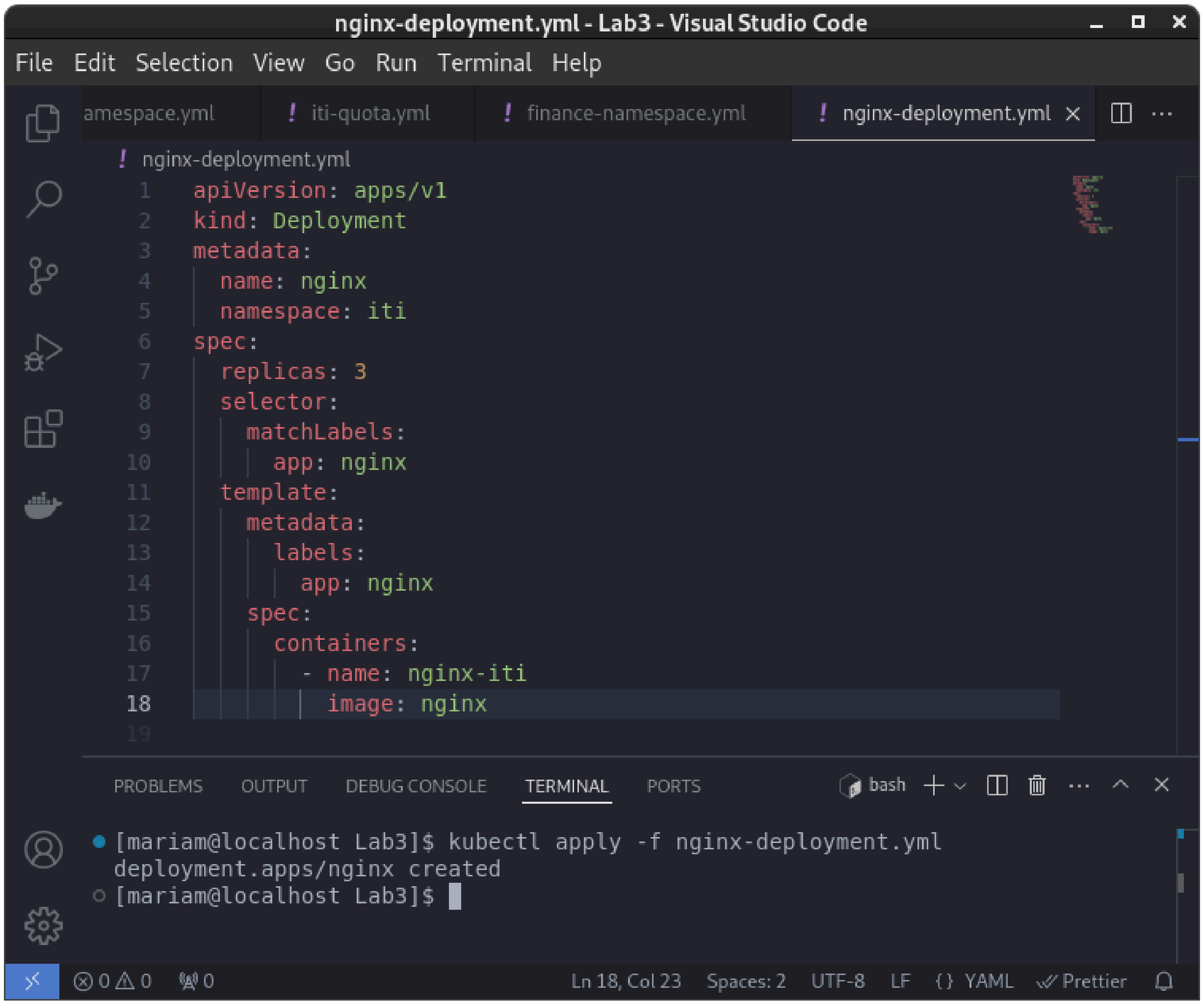
```
[mariam@localhost Lab3]$ kubectl apply -f iti-namespace.yml
namespace/iti created
[mariam@localhost Lab3]$
```


iti-quota.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
e-deployment.yml
iti-namespace.yml
iti-quota.yml
finance-namespace.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 9, Col 1 Spaces: 2 UTF-8 LF {} YAML Prettier

```yaml
apiVersion: v1
kind: ResourceQuota
metadata:
  name: pod-quota
  namespace: iti
spec:
  hard:
    pods: "2"
```

```
[mariam@localhost Lab3]$ kubectl apply -f iti-quota.yml
resourcequota/pod-quota created
[mariam@localhost Lab3]$
```

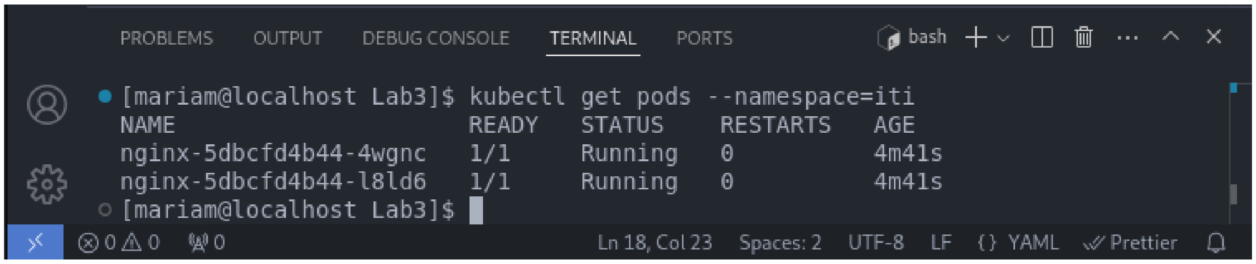
7. Create a deployment named “nginx” with image “nginx”, replicas 3 on the “iti” namespace


nginx-deployment.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
amespace.yml
iti-quota.yml
finance-namespace.yml
nginx-deployment.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 18, Col 23 Spaces: 2 UTF-8 LF {} YAML Prettier

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

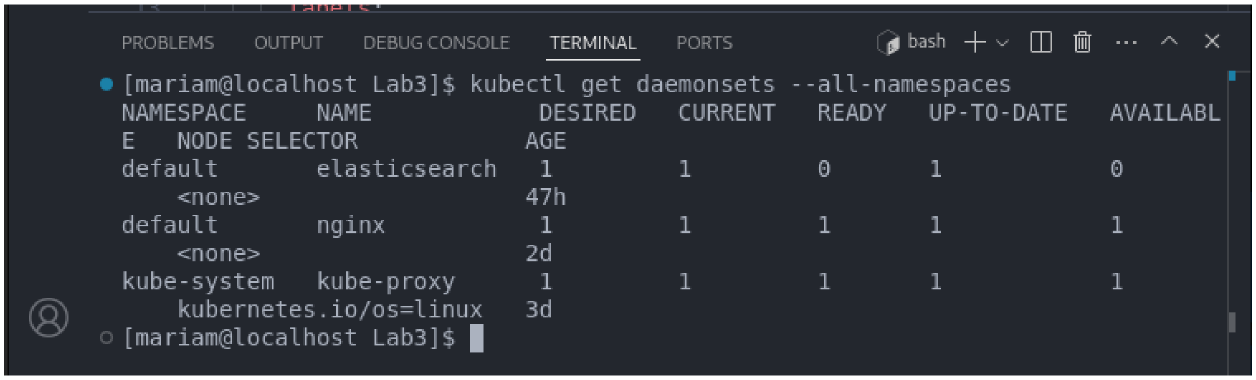
```
[mariam@localhost Lab3]$ kubectl apply -f nginx-deployment.yml
deployment.apps/nginx created
[mariam@localhost Lab3]$
```

8. How many pods have been created within the nginx deployment and why? => 2 because the “iti”namespace has a resource quota applied which limits the number of pods to 2


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 18, Col 23 Spaces: 2 UTF-8 LF {} YAML Prettier

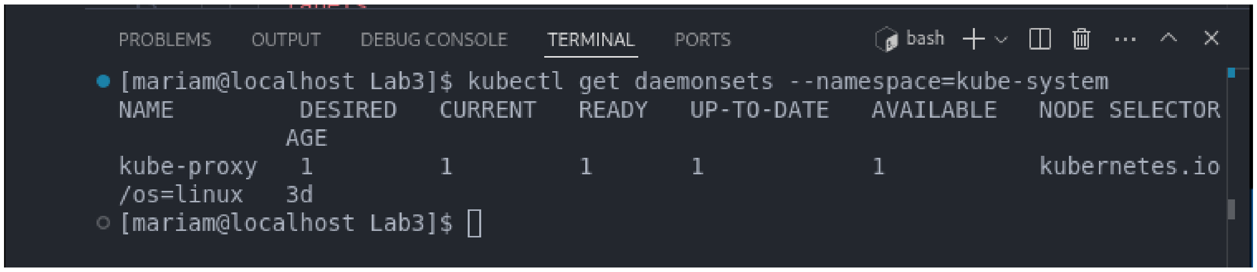
```
[mariam@localhost Lab3]$ kubectl get pods --namespace=iti
NAME                     READY   STATUS    RESTARTS   AGE
nginx-5dbcfd4b44-4wgnc   1/1     Running   0          4m41s
nginx-5dbcfd4b44-l8ld6   1/1     Running   0          4m41s
[mariam@localhost Lab3]$
```

9. How many DaemonSets are created in the cluster in all namespaces?



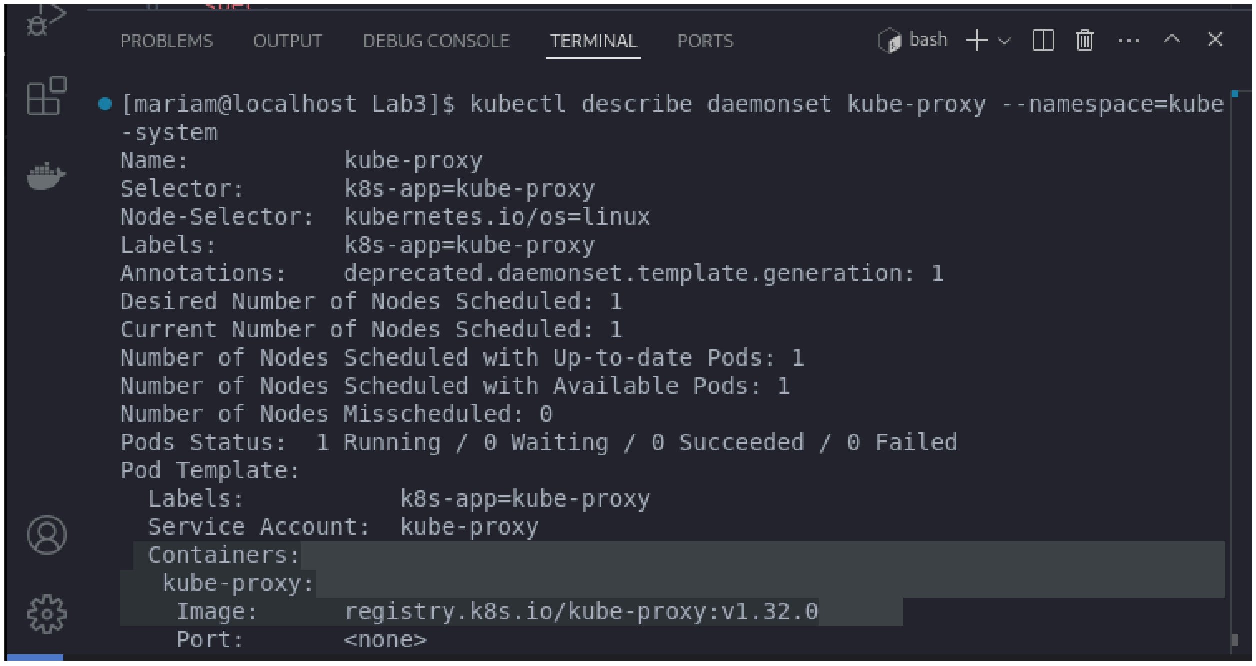
```
[mariam@localhost Lab3]$ kubectl get daemonsets --all-namespaces
NAMESPACE     NAME            DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABL
E   NODE SELECTOR            AGE
default       elasticsearch    1         1         0       1            0
    <none>                   47h
default       nginx            1         1         1       1            1
    <none>                   2d
kube-system   kube-proxy       1         1         1       1            1
    kubernetes.io/os=linux   3d
[mariam@localhost Lab3]$ 
```

10.what DaemonSets exist on the kube-system namespace?



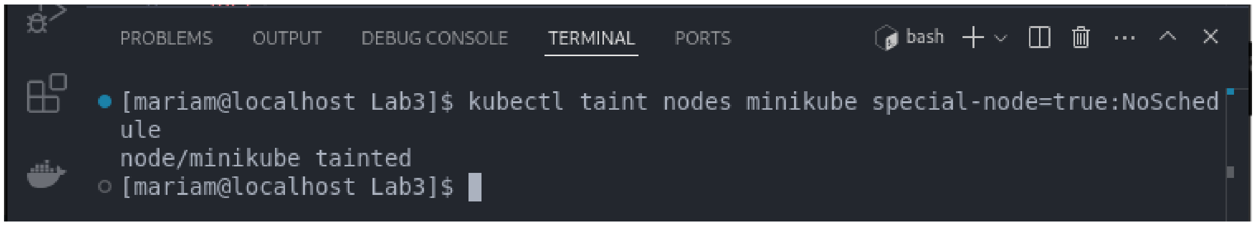
```
[mariam@localhost Lab3]$ kubectl get daemonsets --namespace=kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR
            AGE
kube-proxy   1         1         1       1            1           kubernetes.io
/os=linux   3d
[mariam@localhost Lab3]$ 
```

11.What is the image used by the POD deployed by the kube-proxy DaemonSet?



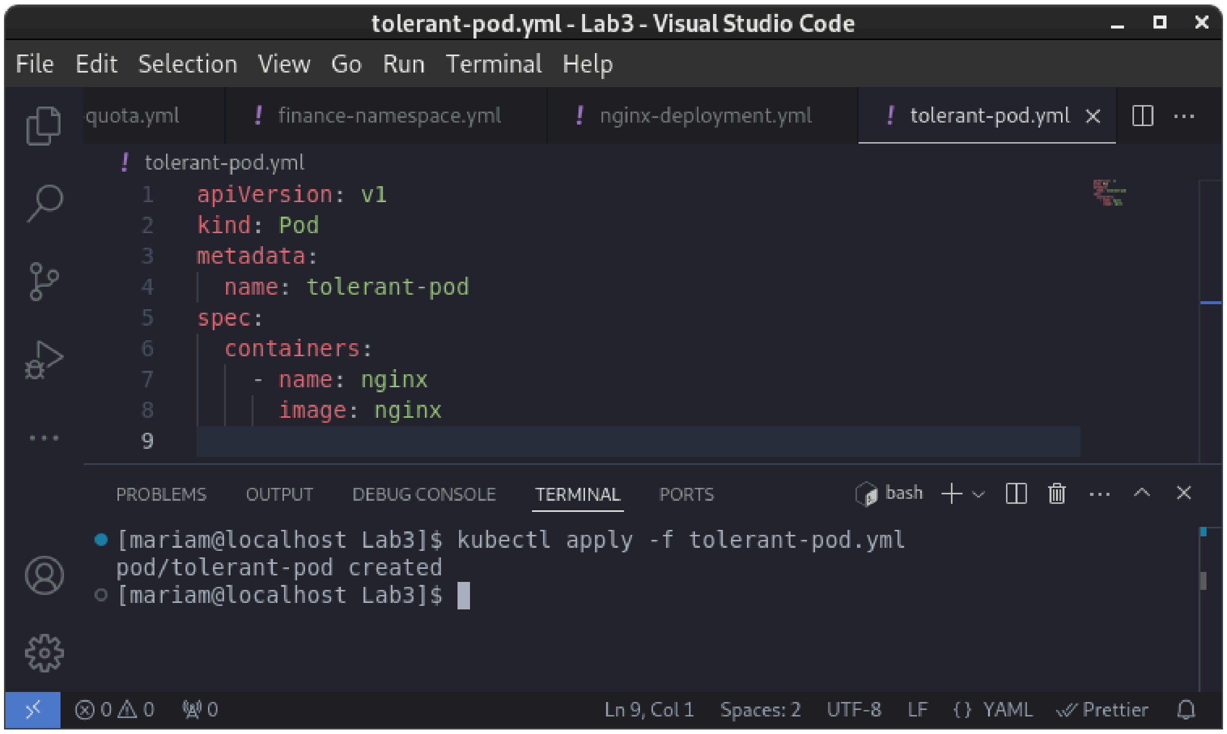
```
[mariam@localhost Lab3]$ kubectl describe daemonset kube-proxy --namespace=kube
-system
Name:           kube-proxy
Selector:       k8s-app=kube-proxy
Node-Selector:  kubernetes.io/os=linux
Labels:         k8s-app=kube-proxy
Annotations:    deprecated.daemonset.template.generation: 1
Desired Number of Nodes Scheduled: 1
Current Number of Nodes Scheduled: 1
Number of Nodes Scheduled with Up-to-date Pods: 1
Number of Nodes Scheduled with Available Pods: 1
Number of Nodes Misscheduled: 0
Pods Status:  1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:           k8s-app=kube-proxy
  Service Account:  kube-proxy
  Containers:
   kube-proxy:
    Image:      registry.k8s.io/kube-proxy:v1.32.0
    Port:       <none>
```

12.Taint node01, the taint should have:
➔ Key: special-node
➔ Value: true
➔ Effect: NoSchedule



```
[mariam@localhost Lab3]$ kubectl taint nodes minikube special-node=true:NoSched
ule
node/minikube tainted
[mariam@localhost Lab3]$
```

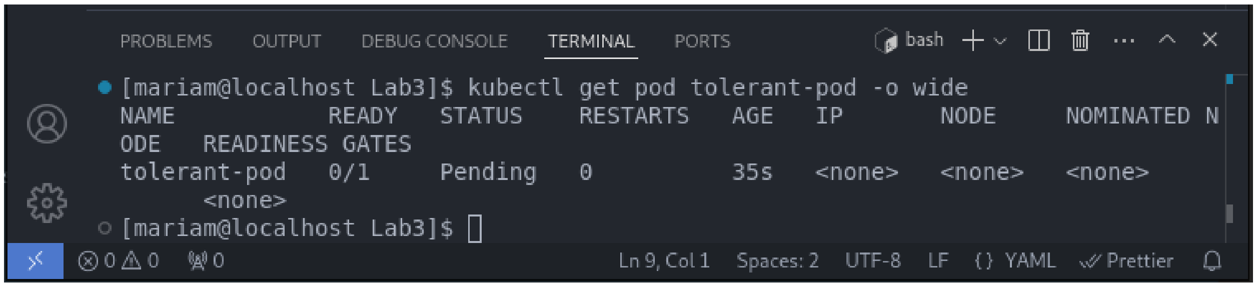
13.Create a pod named tolerant-pod that runs nginx.



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

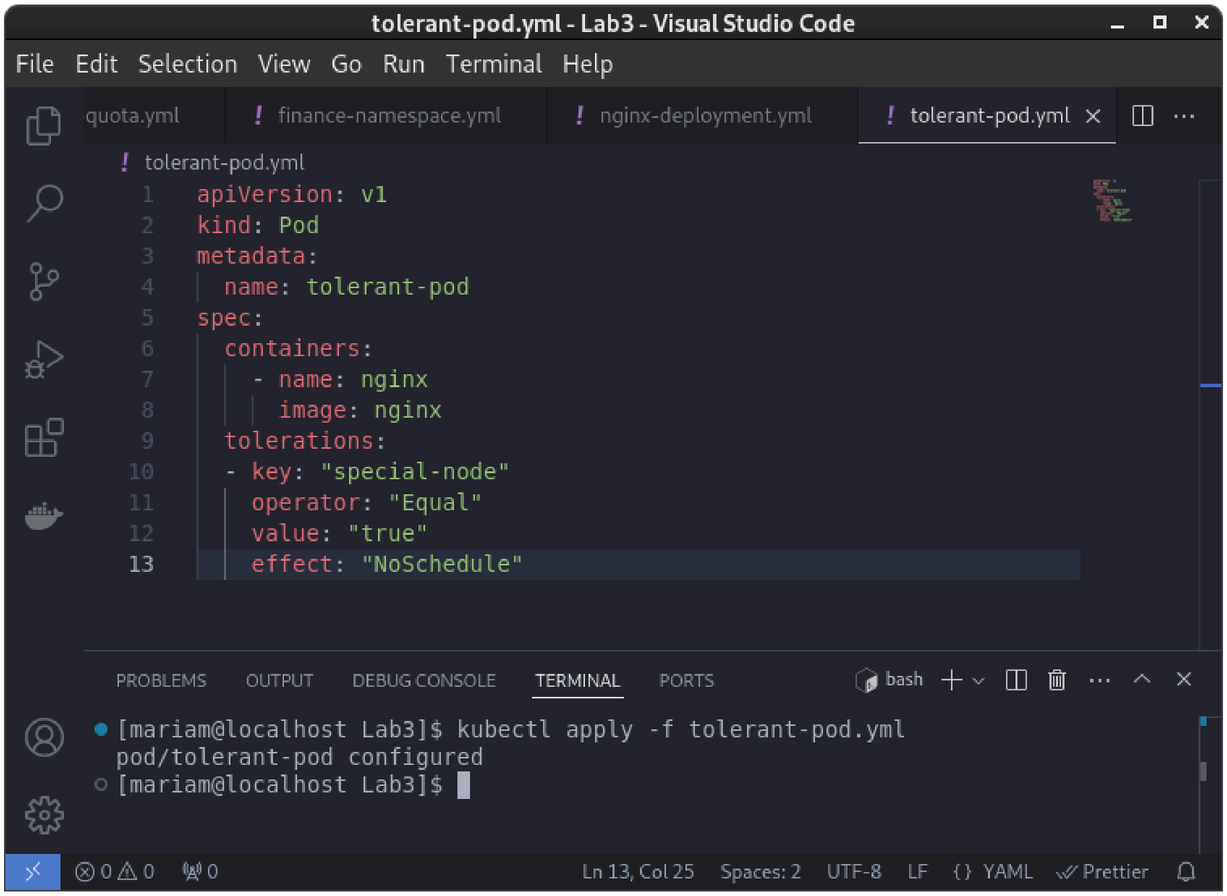
```
[mariam@localhost Lab3]$ kubectl apply -f tolerant-pod.yml
pod/tolerant-pod created
[mariam@localhost Lab3]$
```

14.On which node this pode scheduled & why? => it's pending because thd pod doesn't tolerant any taints yet, it's just named as tolerant-pod not more.



```
[mariam@localhost Lab3]$ kubectl get pod tolerant-pod -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP       NODE     NOMINATED N
ODE    READINESS GATES
tolerant-pod   0/1     Pending   0          35s   <none>   <none>   <none>
      <none>
[mariam@localhost Lab3]$
```

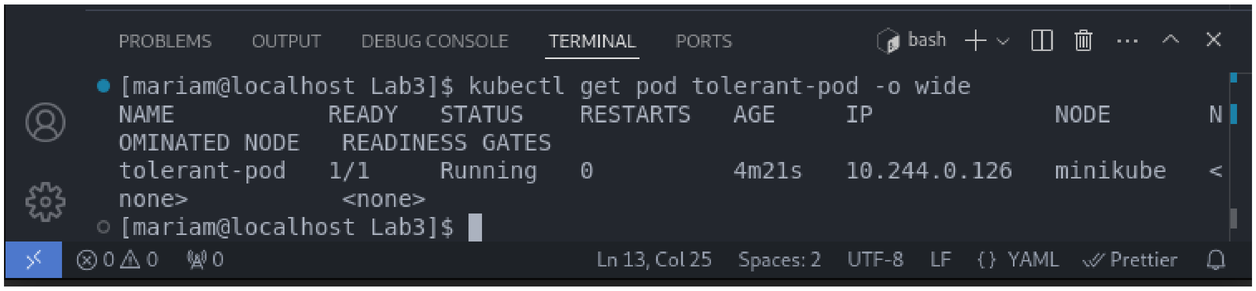
15.Tolerate pod tolerant-pod with the same taint that is on node01


tolerant-pod.yml - Lab3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
quota.yml
finance-namespace.yml
nginx-deployment.yml
tolerant-pod.yml
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash
Ln 13, Col 25   Spaces: 2   UTF-8   LF   {} YAML   Prettier

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tolerant-pod
spec:
  containers:
    - name: nginx
      image: nginx
  tolerations:
  - key: "special-node"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"
```

```
[mariam@localhost Lab3]$ kubectl apply -f tolerant-pod.yml
pod/tolerant-pod configured
[mariam@localhost Lab3]$
```

16.Now, on which node this pode scheduled & why? => it’s on minikube node because it has a toleration matching the taint on the node.



```
● [mariam@localhost Lab3]$ kubectl get pod tolerant-pod -o wide
  NAME           READY   STATUS    RESTARTS   AGE     IP             NODE       N
  OMINATED NODE   READINESS GATES
  tolerant-pod   1/1     Running   0          4m21s   10.244.0.126   minikube   <
  none>           <none>
○ [mariam@localhost Lab3]$ 
```