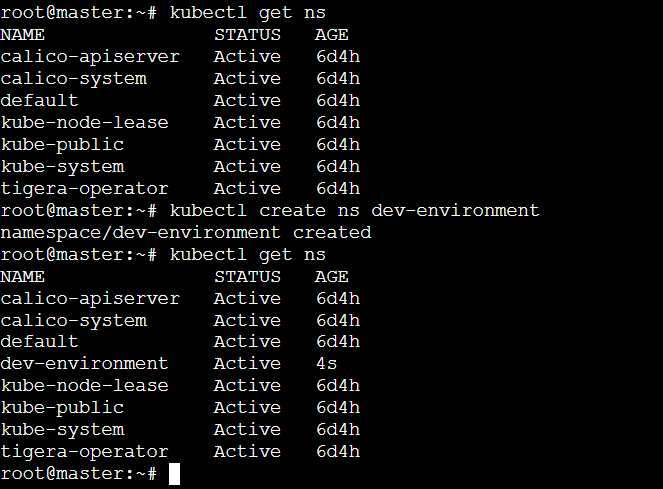
**Kubernetes Tasks - 5**

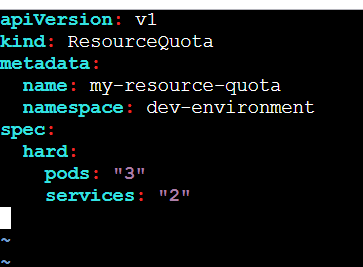
**1) Create a namespace dev-environment and apply a resource-based quota that restricts the number of pods to 3 and services to 2.**

**>> kubectl create ns dev-environment**



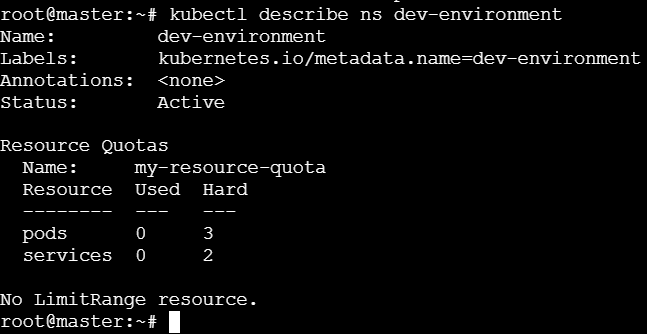
**>> vi rquota.yaml**

|  |
| --- |
| apiVersion: v1  kind: ResourceQuota  metadata:  name: my-resource-quota  namespace: dev-environment  spec:  hard:  pods: "3"  services: "2" |



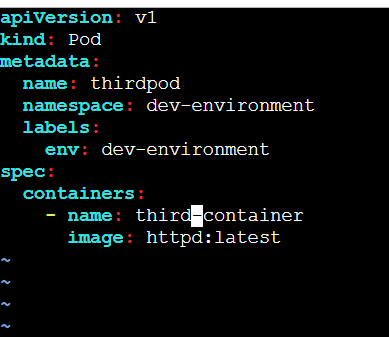
**>> kubectl apply -f rquota.yaml**

**>> kubectl describe ns dev-environment**

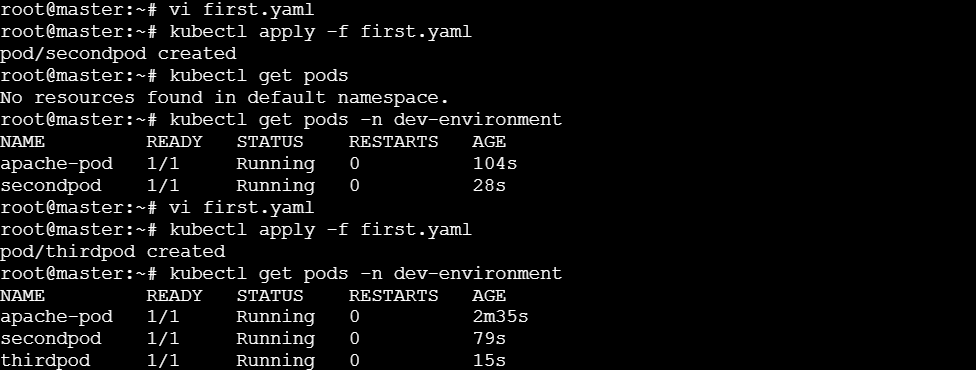


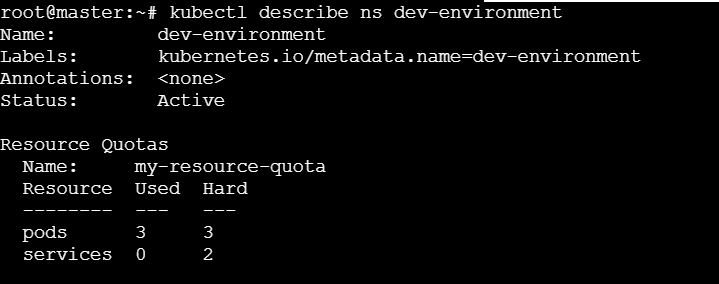
* **Create pods:**

**>> kubectl apply -f first.yaml**



* Similarly create 3 more pods so that it should through an error while creating 4th pod, as we have configured resource-based quota that restricts the number of pods to 3.

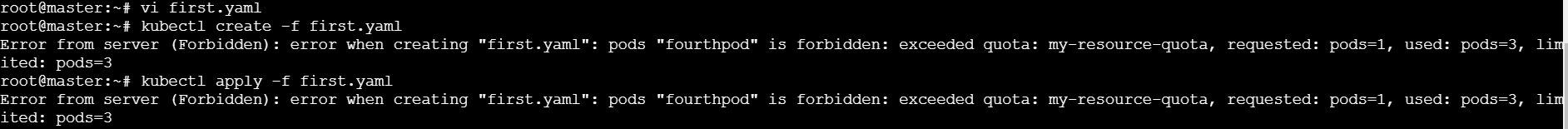




* **Creating the 4th pod:**



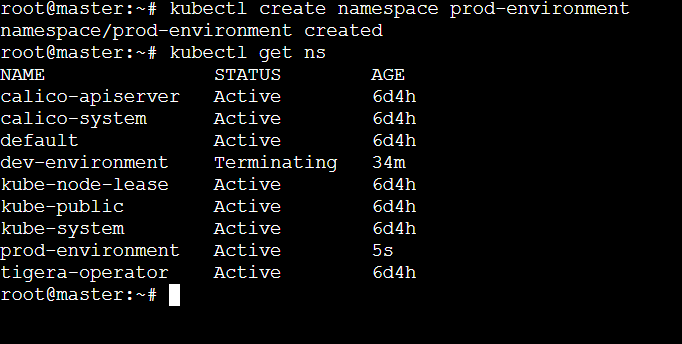
* **It throughs error while creating the 4th pod:**



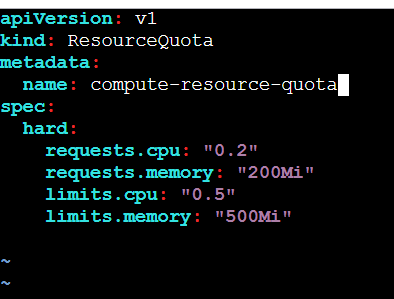
**2) Create a pod in the prod-environment namespace with 0.2 CPU and 200Mi memory requests, and 0.5 CPU and 500Mi memory limits.**

**>> kubectl create namespace prod-environment**

**>> kubectl get ns**

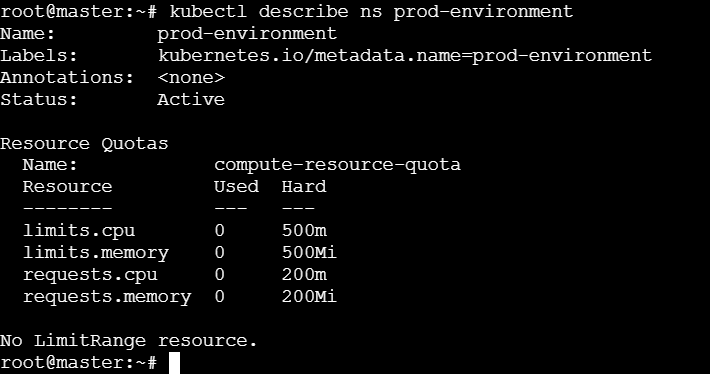
****

|  |
| --- |
| apiVersion: v1  kind: ResourceQuota  metadata:  name: compute-resource-quota  spec:  hard:  requests.cpu: "0.2"  requests.memory: "200Mi"  limits.cpu: "0.5"  limits.memory: "500Mi" |

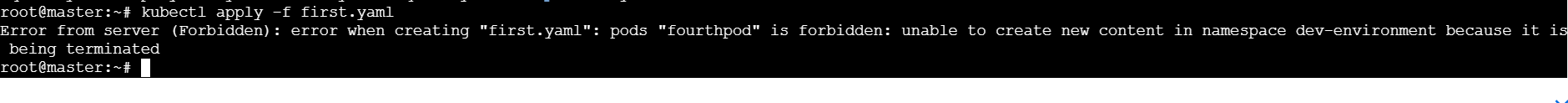
****

**>> kubectl apply -f cquota.yaml -n prod-environment**

**>> kubectl describe ns prod-environment**

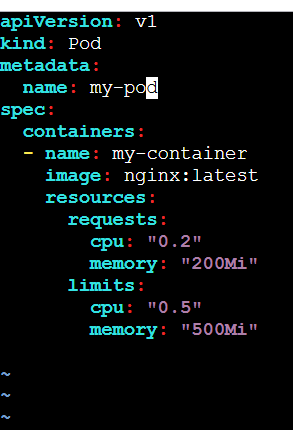
****

* **Creates error while creating a pod without setting the CPU limits, memory limits, cup request and memory requests;**

****

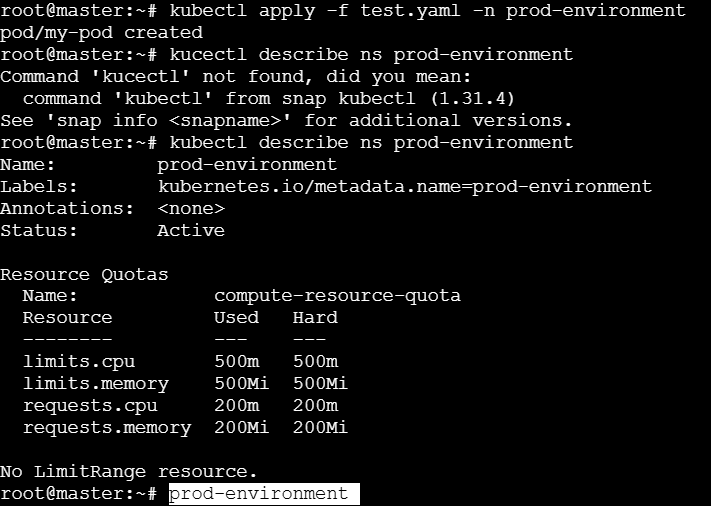
* **Let us create a pod with the specifications we configured in compute-based quota:**

|  |
| --- |
| apiVersion: v1  kind: Pod  metadata:  name: my-pod  spec:  containers:  - name: my-container  image: nginx:latest  resources:  requests:  cpu: "0.2"  memory: "200Mi"  limits:  cpu: "0.5"  memory: "500Mi" |

****

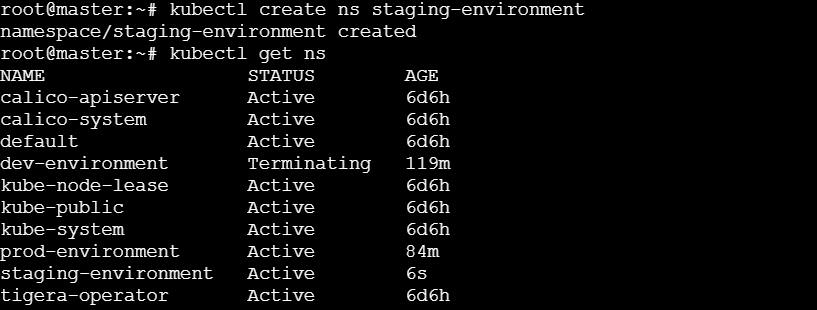
**>> kubectl apply -f test.yaml -n prod-environment**

**>> kucectl describe ns prod-environment**

****

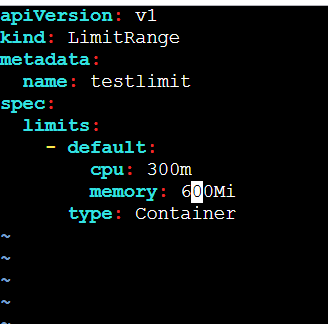
**3) In the staging-environment namespace, set a LimitRange that assigns default CPU and memory limits (300m CPU, 600Mi memory) and applies a minimum and maximum CPU.**

**>> kubectl create ns staging-environment**

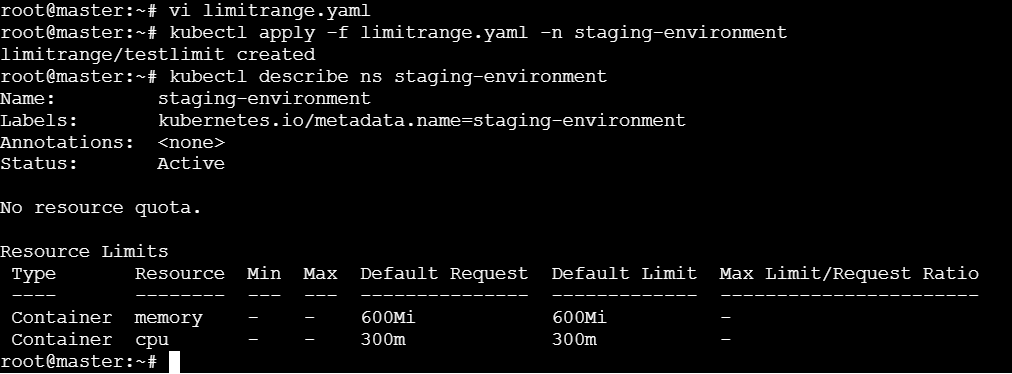
****

**>> vi limitrange.yaml**

|  |
| --- |
| apiVersion: v1  kind: LimitRange  metadata:  name: testlimit  spec:  limits:  - default:  cpu: 300m  memory: 600Mi  type: Container |

****

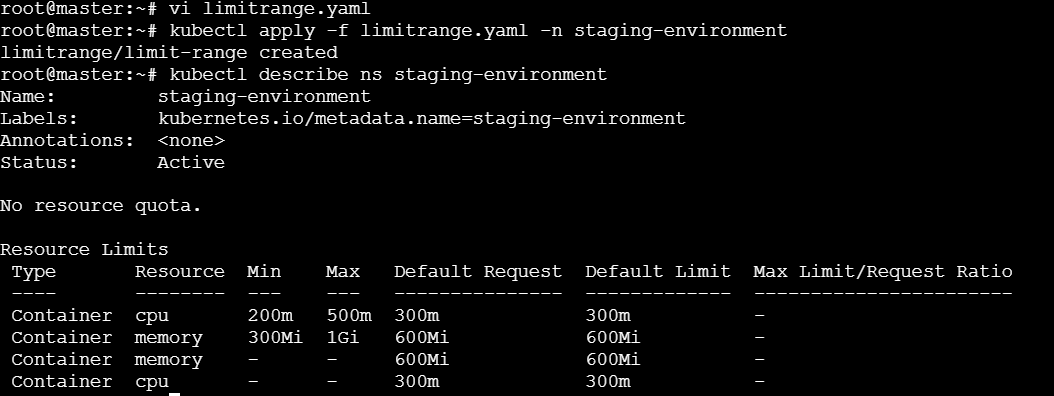
**>> kubectl apply -f limitrange.yaml -n staging-environment**

****

* **For min & max CPU**

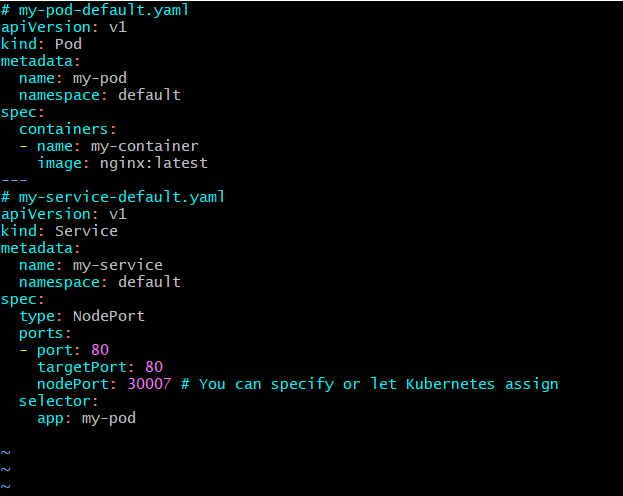
**>> vi limitrange.yaml**

|  |
| --- |
| apiVersion: v1  kind: LimitRange  metadata:  name: limit-range  spec:  limits:  - default:  cpu: 300m  memory: 600Mi  defaultRequest:  cpu: 300m  memory: 600Mi  max:  cpu: 500m  memory: 1Gi  min:  cpu: 200m  memory: 300Mi  type: Container |

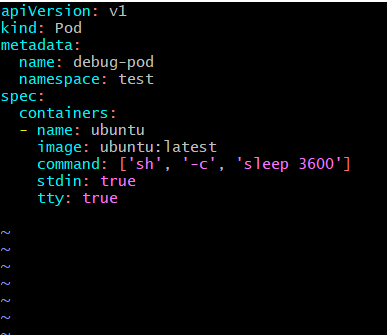
****

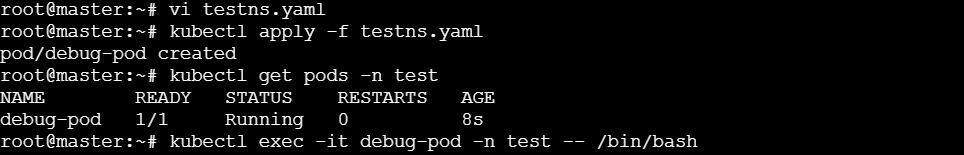
**4) Create a pod and a NodePort service in the default namespace, then create another pod in the test namespace and communicate between them using Service DNS.**

* **Create a pod in the default namespace and create the nodeport service and attach it to the pod in default namespace.**

****

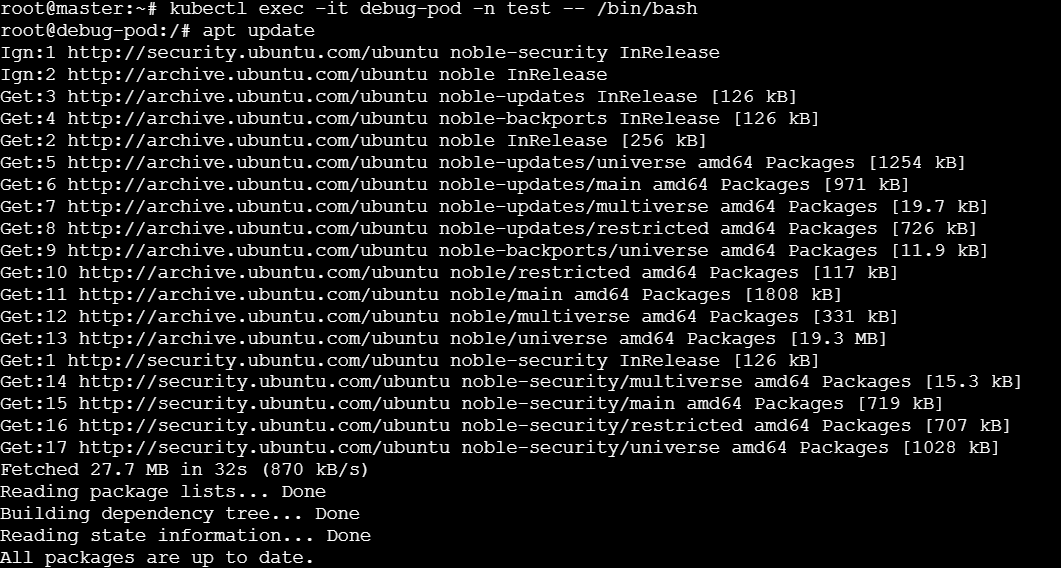
* **Create a pod in namespace called test.**

****

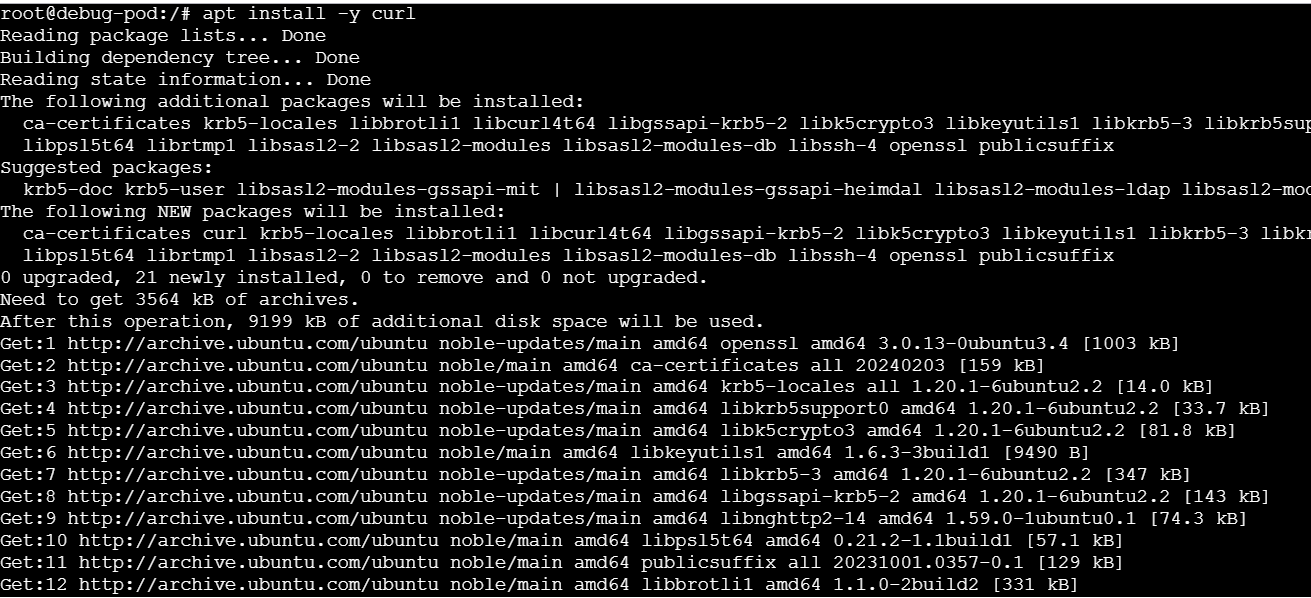
****

* **Login to the test-pod in the test namespace**

**>> kubectl exec -it test-pod -n test -- /bin/sh**

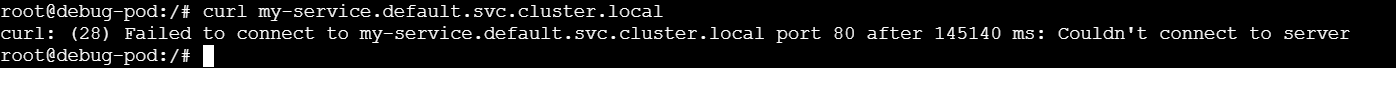
****

**>>** **apt install -y curl**

****

* **Testing the connection using curl**

**>> curl my-service.default.svc.cluster.local**

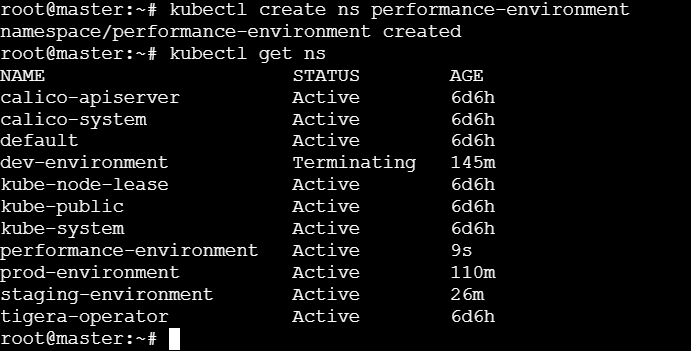
****

**5) Apply a LimitRange with a max limit/request ratio of 2 for memory in the performance-environment namespace, and test by creating a pod with mismatched resource requests and limits.**

* **Create a namespace as performance-environment**

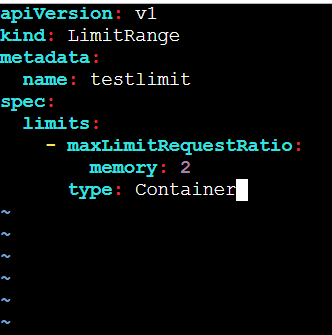
**>> kubectl create ns performance-environment**

**>> kubectl get ns**

****

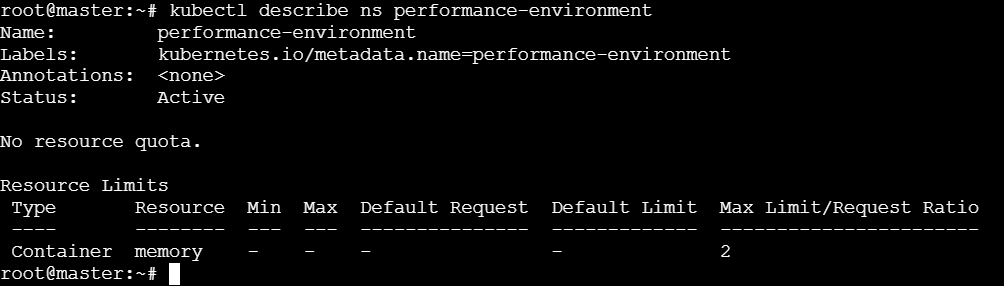
**>> vi ratio.yaml**

|  |
| --- |
| apiVersion: v1  kind: LimitRange  metadata:  name: testlimit  spec:  limits:  - maxLimitRequestRatio:  memory: 2  type: Container |

****

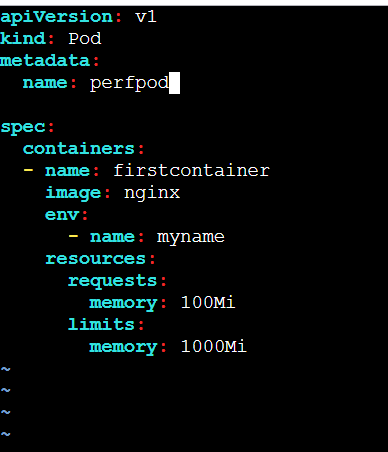
**>> kubectl apply -f ratio.yaml -n performance-environment**

**>> kubectl describe ns performance-environment**

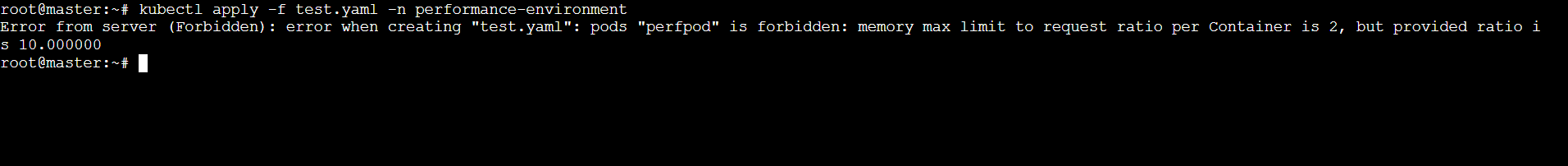
****

* **Let us create a pod which violates the above configured Max limit/Request Ratio:**

|  |
| --- |
| apiVersion: v1  kind: Pod  metadata:  name: perfpod  spec:  containers:  - name: firstcontainer  image: nginx  env:  - name: myname  resources:  requests:  memory: 100Mi  limits:  memory: 1000Mi  ~ |

****

**>> kubectl apply -f test.yaml -n performance-environment**

****