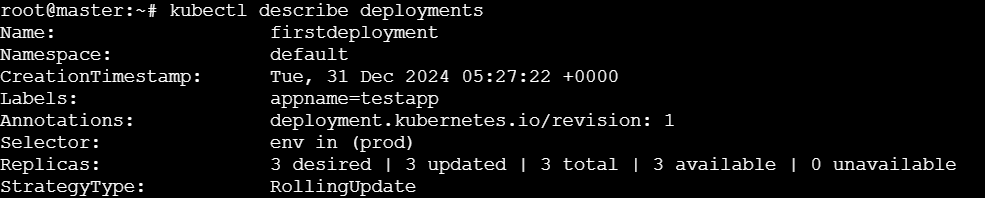
**Kubernetes Task-04:**

1. **Deploy an application using a Deployment with 3 replicas and a rolling update strategy.**

**>>> vi deployment.yaml**

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: firstdeployment  labels:  appname: testapp  spec:  replicas: 3  selector:  matchExpressions:  - key: env  operator: In  values:  - prod  template:  metadata:  name: firstpod  labels:  env: prod  spec:  containers:  - name: firstcontainer  image: nginx  env:  - name: myname |

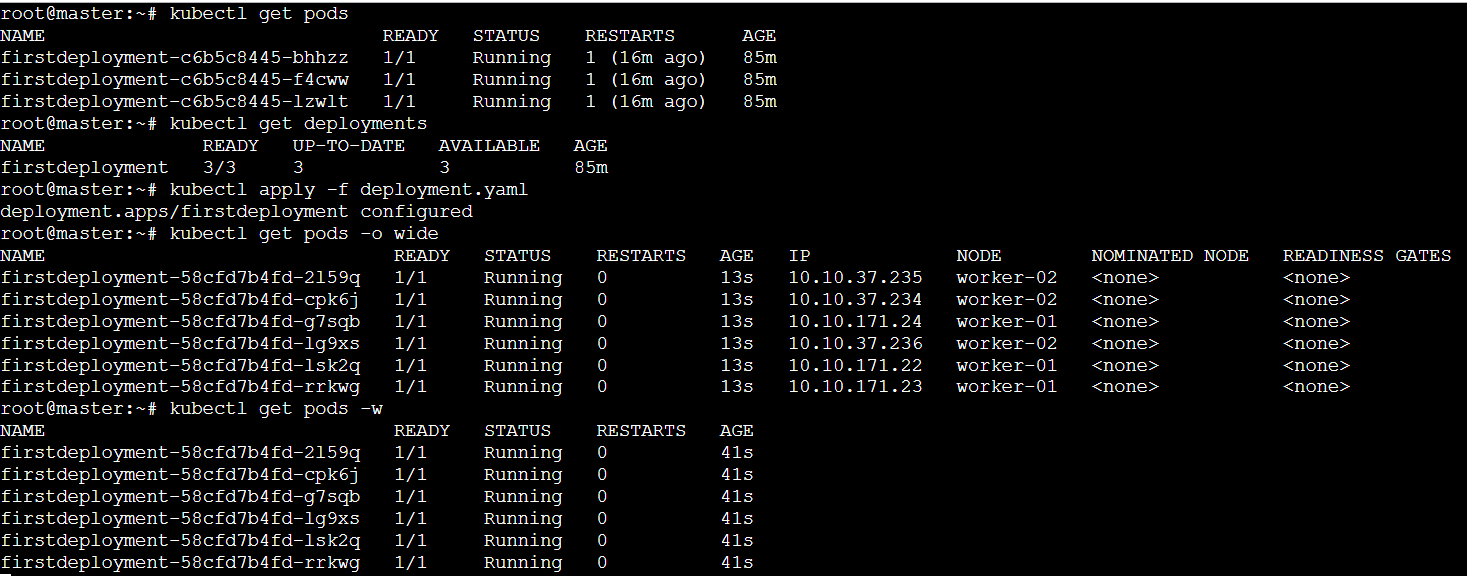
****

****

1. **Configure a Deployment with a Recreate strategy and observe the downtime.**

**>>vi deployment.yaml**

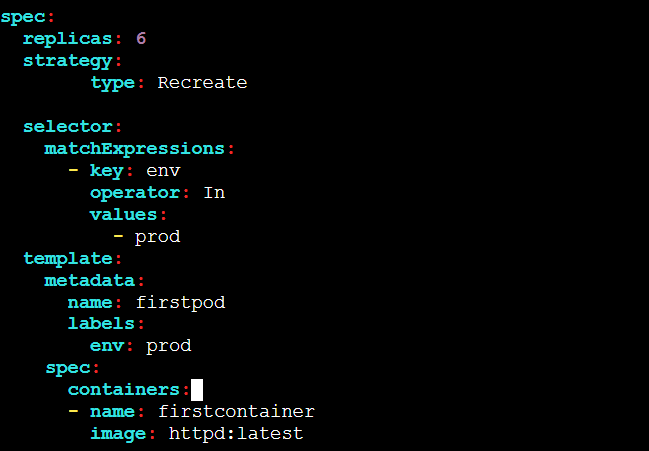
|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: firstdeployment  labels:  appname: testapp  spec:  replicas: 6  strategy:  type: Recreate  selector:  matchExpressions:  - key: env  operator: In  values:  - prod  template:  metadata:  name: firstpod  labels:  env: prod  spec:  containers:  - name: firstcontainer  image: nginx  env:  - name: myname |

****

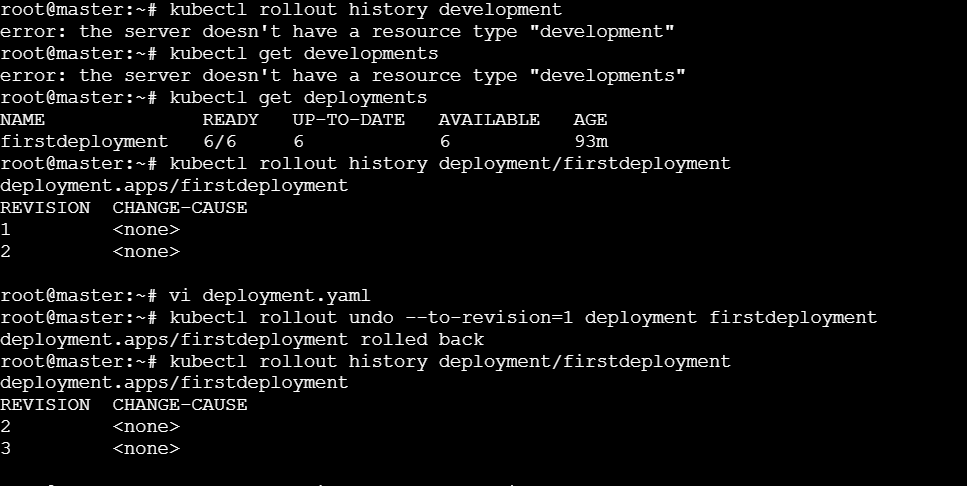
1. **Update an existing Deployment and perform a rollback to the previous version.**

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: firstdeployment  labels:  appname: testapp  spec:  replicas: 3  selector:  matchExpressions:  - key: env  operator: In  values:  - prod  template:  metadata:  name: firstpod  labels:  env: prod  spec:  containers:  - name: firstcontainer  image: nginx  env:  - name: myname |

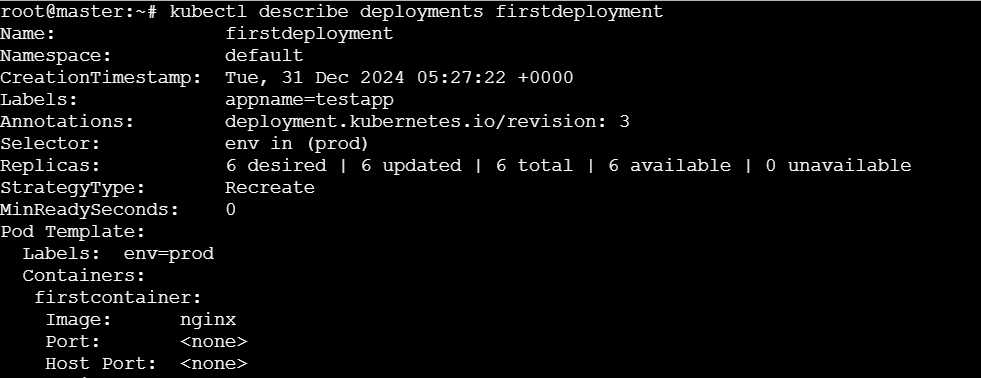
* **Change the revision by changing the image from nginx to httpd:**

****

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

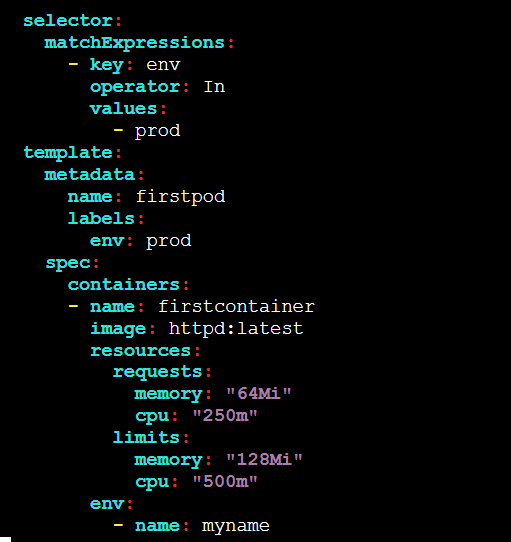
****

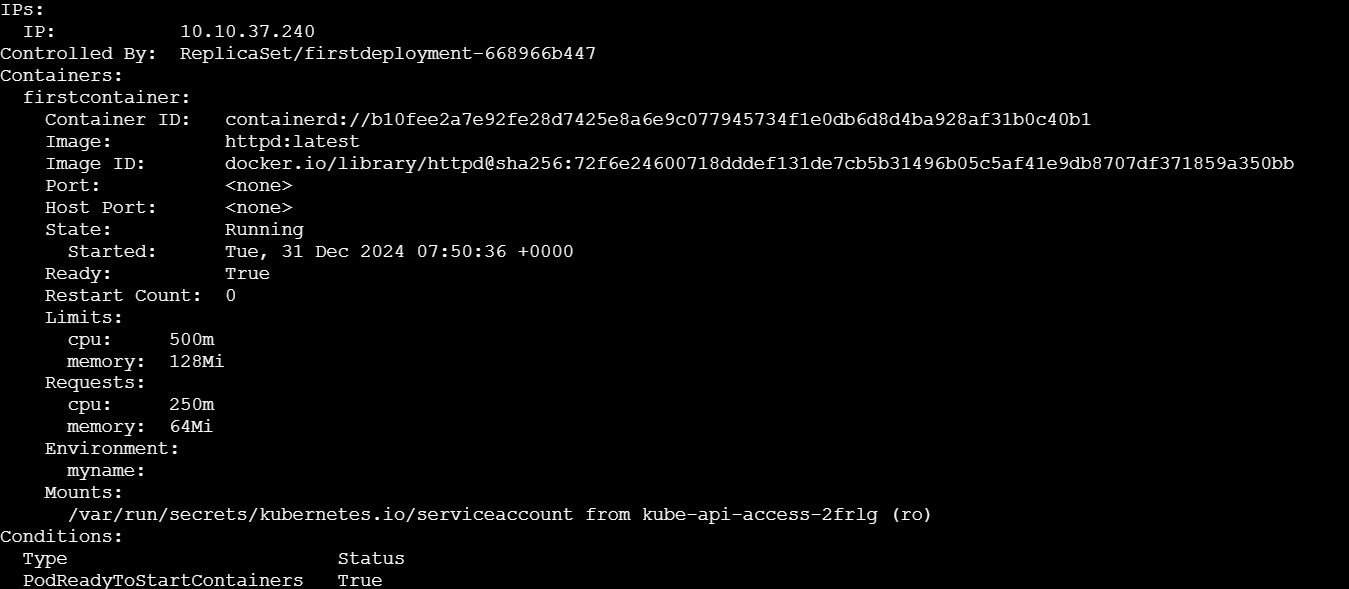
* **We can see the image: nginx**

****

1. **Modify a Deployment to add resource requests and limits for CPU and memory.**

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: firstdeployment  labels:  appname: testapp  spec:  replicas: 6  strategy:  type: Recreate  selector:  matchExpressions:  - key: env  operator: In  values:  - prod  template:  metadata:  name: firstpod  labels:  env: prod  spec:  containers:  - name: firstcontainer  image: httpd:latest  resources:  requests:  memory: "64Mi"  cpu: "250m"  limits:  memory: "128Mi"  cpu: "500m"  env:  - name: myname |

****

****

1. **Create a Deployment with MaxSurge and MaxUnavailable configurations.**

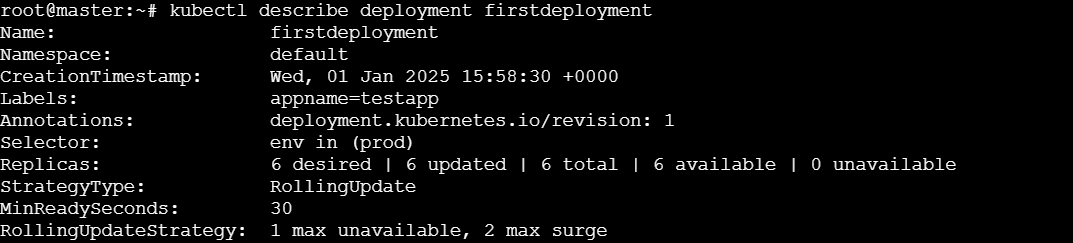
**>>> vi deployment.yaml**

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: firstdeployment  labels:  appname: testapp  spec:  replicas: 6  minReadySeconds: 30  strategy:  rollingUpdate:  maxSurge: 2  maxUnavailable: 1  selector:  matchExpressions:  - key: env  operator: In  values:  - prod  template:  metadata:  name: firstpod  labels:  env: prod  spec:  containers:  - name: firstcontainer  image: nginx  env:  - name: myname |

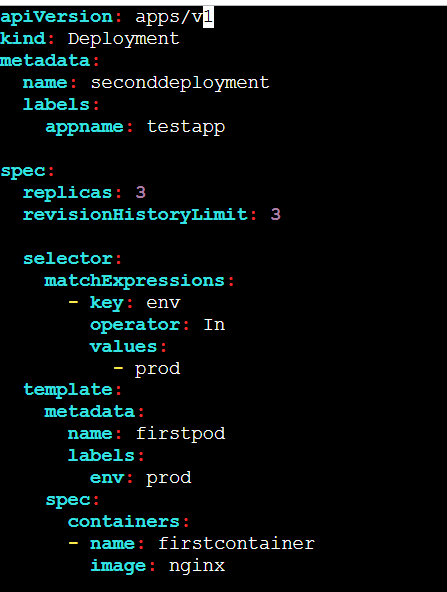
**>>>kubectl apply -f deployment.yaml**

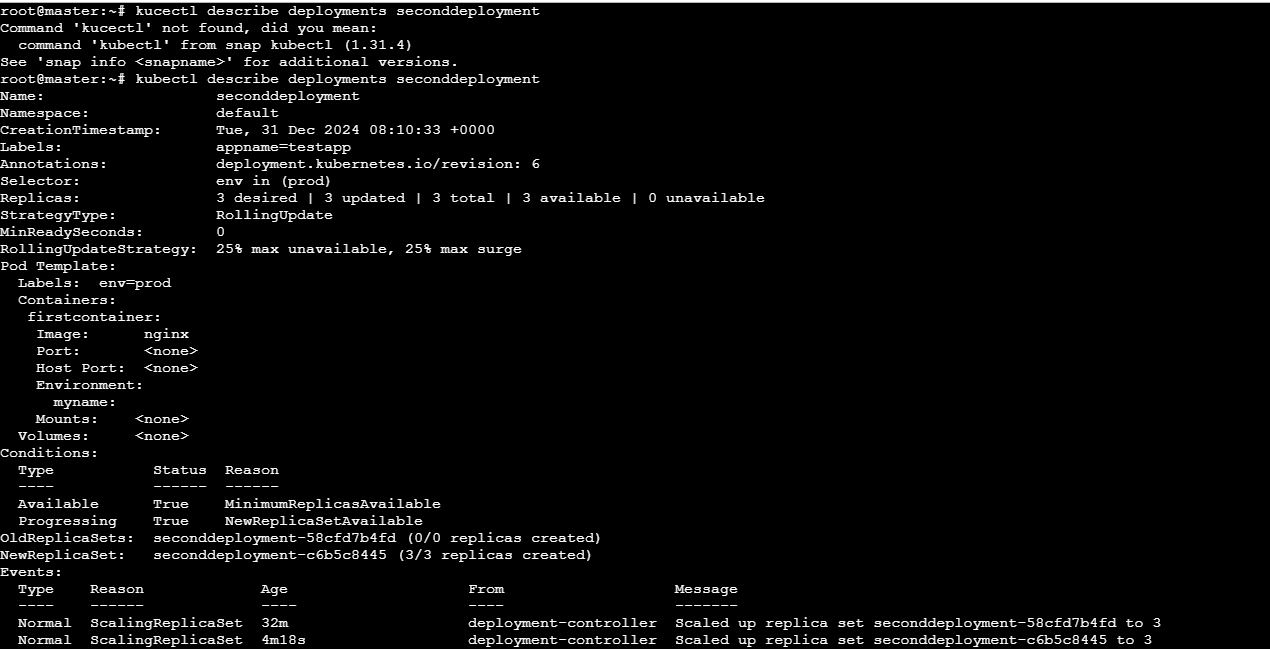
****

**>>> kubectl describe deployments firstdeployment**

****

1. **Set up a Deployment with a custom revision history limit.**

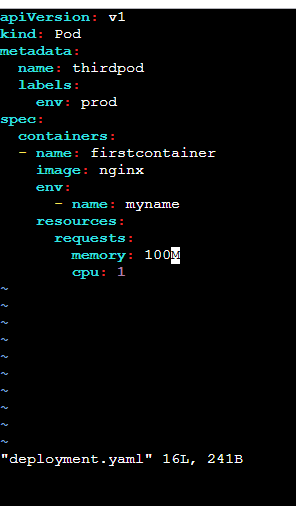
****

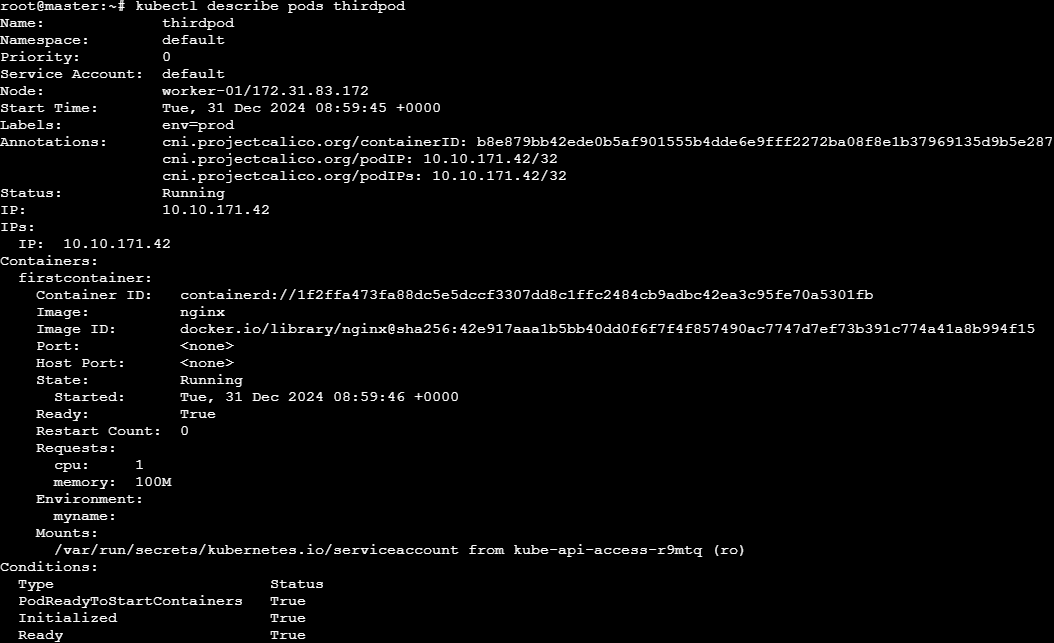
****

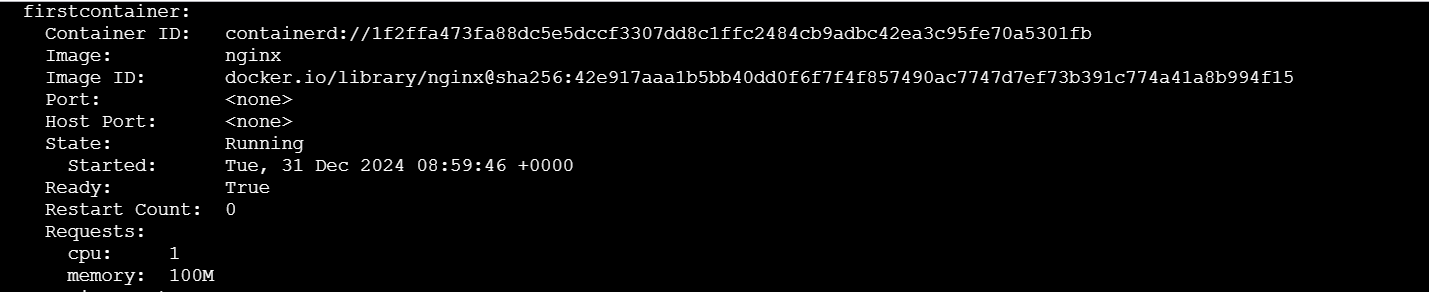
**7) Pause a Deployment during an update, and then resume it.**

****

**8) Create a pod using resource requests for memory and CPU, and observe how the scheduler assigns it to a node.**

****

****

****