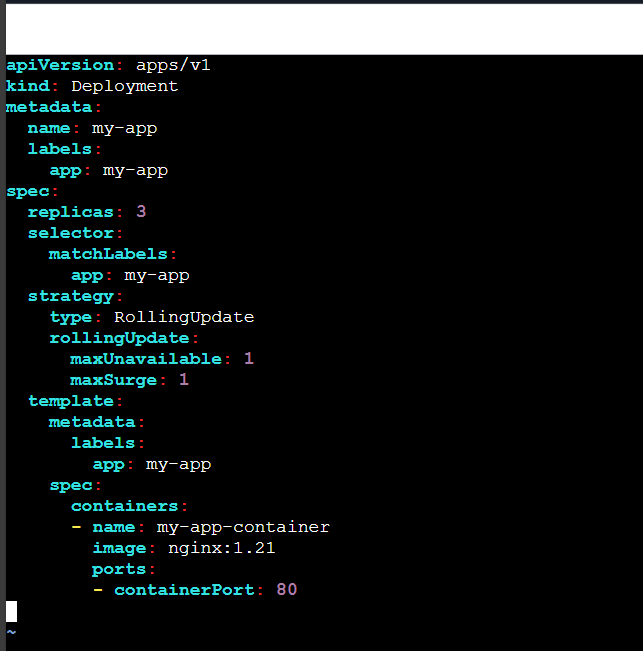
**KUBERNETES[TASK-4]**

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

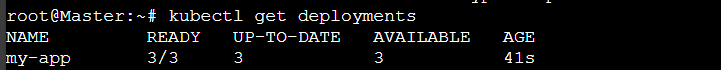
Created a file,deployment.yaml



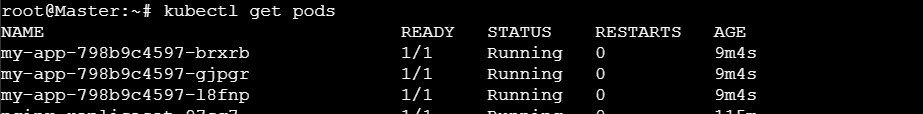
Now,using command

kubectl apply -f deployment.yaml



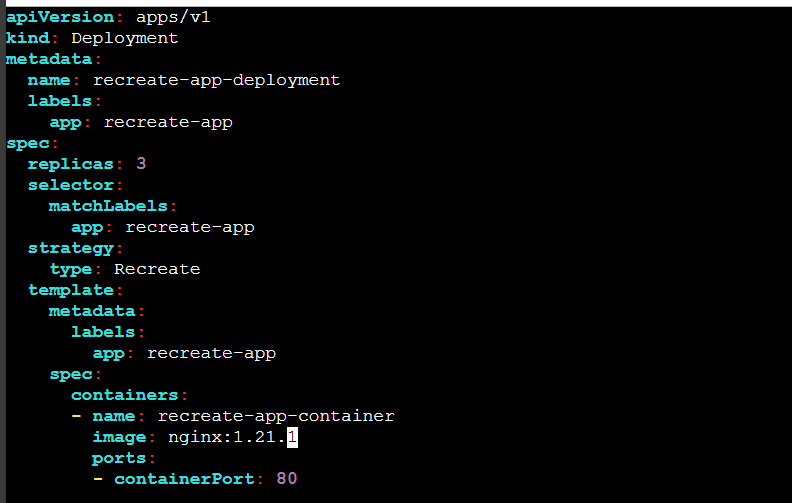


Checking the pods: kubectl get pods

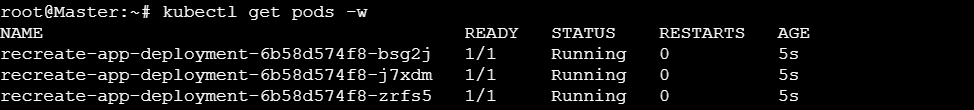




2) Configure a Deployment with a Recreate strategy and observe the downtime.



Apply the changes

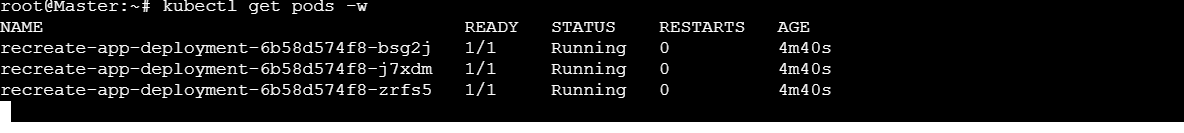


Now change the image version

And apply



Then we will observe some downtime



3) Update an existing Deployment and perform a rollback to the previous version.



4) Modify a Deployment to add resource requests and limits for CPU and memory.

apiVersion: apps/v1

kind: Deployment

metadata:

name: example-deployment

spec:

replicas: 3

selector:

matchLabels:

app: example-app

template:

metadata:

labels:

app: example-app

spec:

containers:

- name: example-container

image: nginx:latest

resources:

requests:

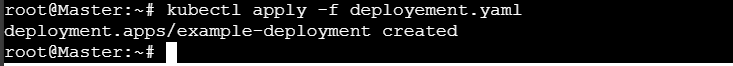
memory: "64Mi"

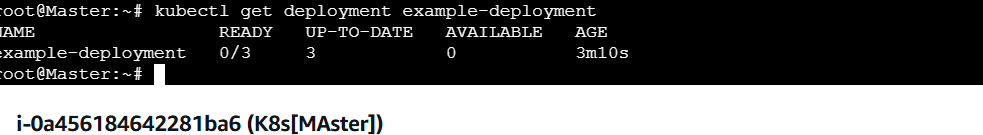
cpu: "250m"

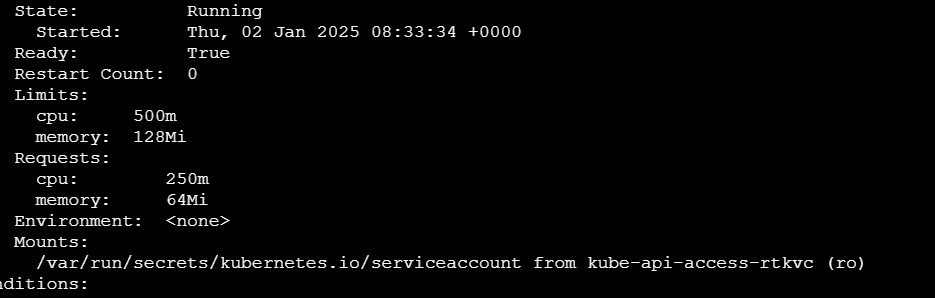
limits:

memory: "128Mi"

cpu: "500m"







5) Create a Deployment with MaxSurge and MaxUnavailable configurations.

apiVersion: apps/v1

kind: Deployment

metadata:

name: example-deployment

spec:

replicas: 5

selector:

matchLabels:

app: example-app

strategy:

type: RollingUpdate

rollingUpdate:

maxSurge: 2

maxUnavailable: 1

template:

metadata:

labels:

app: example-app

spec:

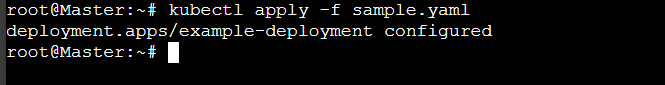
containers:

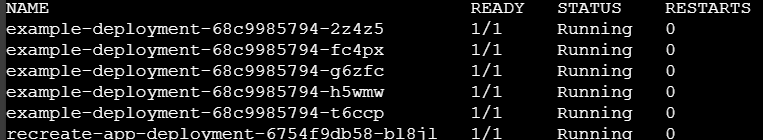
- name: example-container

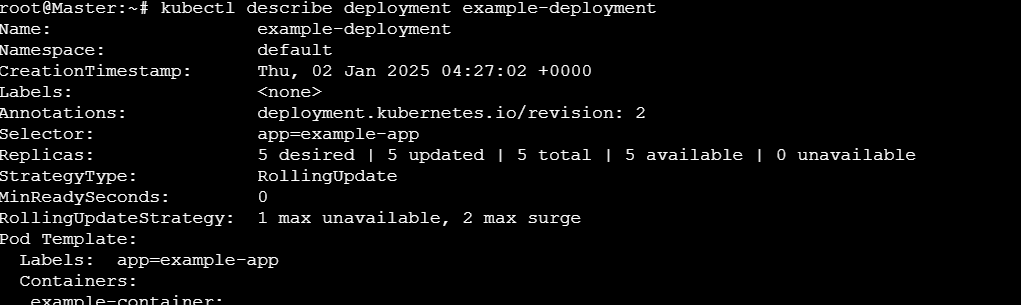
image: nginx:latest

ports:

- containerPort: 80







6) Set up a Deployment with a custom revision history limit.

apiVersion: apps/v1

kind: Deployment

metadata:

name: example-deployment

spec:

replicas: 3

selector:

matchLabels:

app: example-app

strategy:

type: RollingUpdate

rollingUpdate:

maxSurge: 1

maxUnavailable: 1

revisionHistoryLimit: 5 # Keep 5 previous revisions

template:

metadata:

labels:

app: example-app

spec:

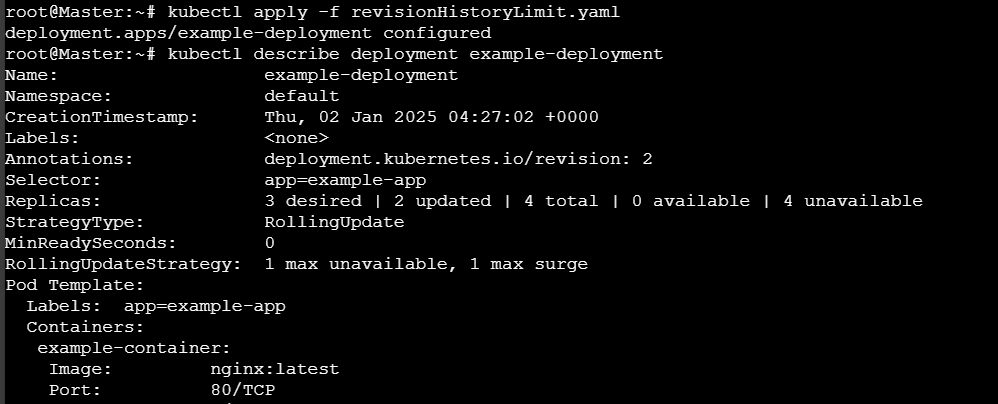
containers:

- name: example-container

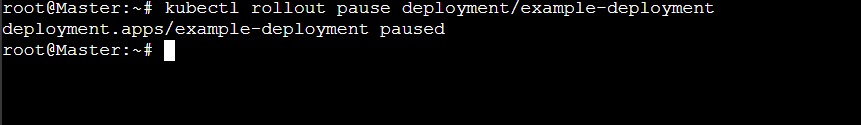
image: nginx:latest

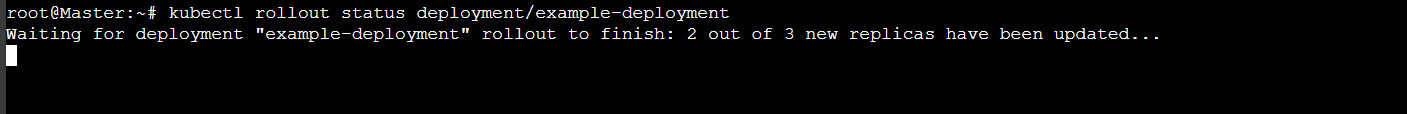
ports:

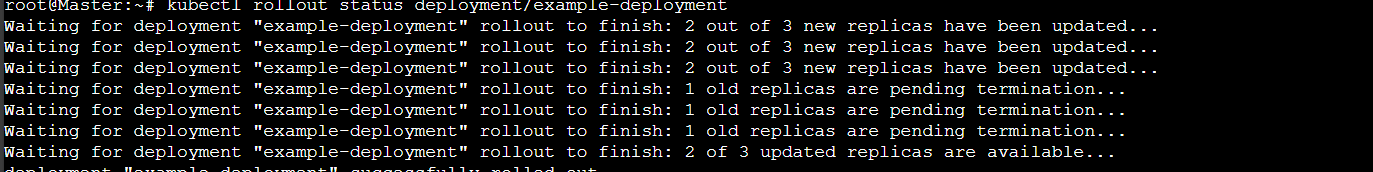
- containerPort: 80



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.

