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

Create a yaml file.

- **vi rollingupdate.yml**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app-deployment
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
```

app: my-app

spec:

containers:

- name: my-app-container

image: nginx:latest

ports:

- containerPort: 80

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app-deployment
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app-container
        image: nginx:latest
        ports:
        - containerPort: 80
```

- kubectl apply -f rollingupdate.yml
- kubectl rollout status deployment my-app-deployment
- kubectl get pods -l app=my-app

```
[root@master ~]# vi rollingupdate.yml
[root@master ~]# kubectl apply -f rollingupdate.yml
deployment.apps/my-app-deployment created
[root@master ~]# kubectl rollout status deployment my-app-deployment
deployment "my-app-deployment" successfully rolled out
[root@master ~]# kubectl get deployment my-app-deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
my-app-deployment   3/3     3            3           110s
[root@master ~]# kubectl get pods -l app=my-app
NAME                                 READY   STATUS    RESTARTS   AGE
my-app-deployment-9b8bb4c4-j7rhc   1/1     Running   0          2m9s
my-app-deployment-9b8bb4c4-xkt67   1/1     Running   0          2m9s
my-app-deployment-9b8bb4c4-xwv26   1/1     Running   0          2m9s
[root@master ~]#
```

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

- **vi nginx-recreate.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-recreate

spec:

replicas: 3

strategy:

type: Recreate

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.24

ports:

- containerPort: 80

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-recreate
spec:
  replicas: 3
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.24
        ports:
        - containerPort: 80
```

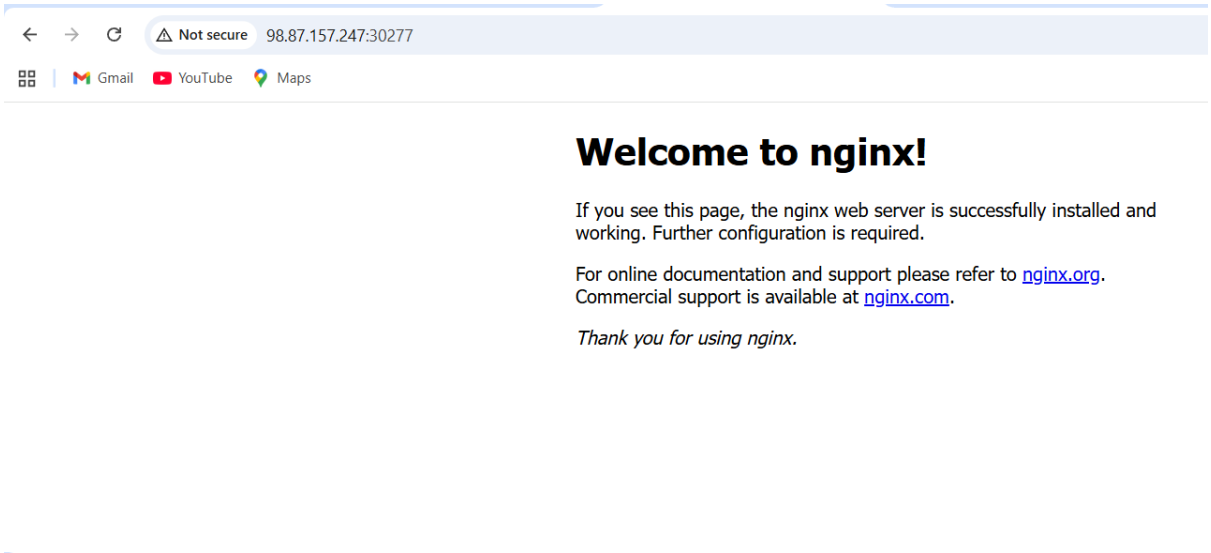
- **kubectl apply -f nginx-recreate.yaml**
- **kubectl get deploy**
- **kubectl get pods**

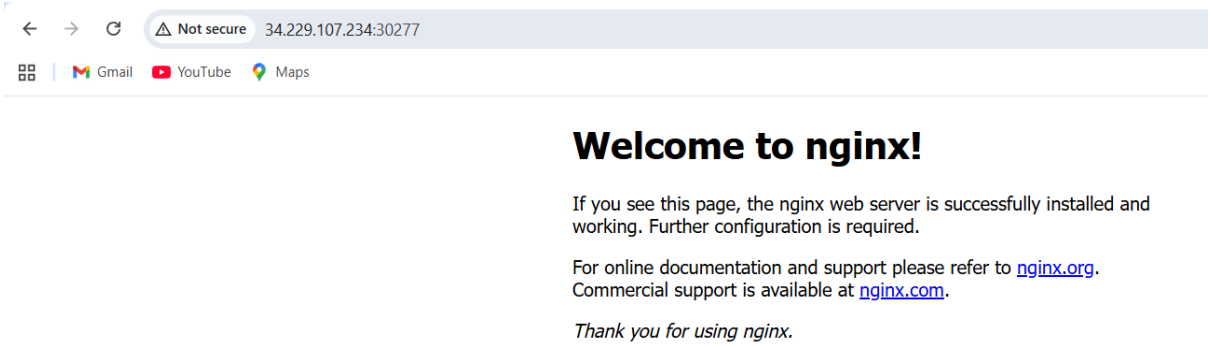
```
[root@master ~]# vi nginx-recreate.yaml
[root@master ~]# kubectl apply -f nginx-recreate.yaml
deployment.apps/nginx-recreate created
[root@master ~]# kubectl get deploy
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
nginx-recreate    3/3     3            3           12s
[root@master ~]# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-recreate-6d8cd7bf4b-4l8vv    1/1     Running   0          22s
nginx-recreate-6d8cd7bf4b-gm6x5    1/1     Running   0          22s
nginx-recreate-6d8cd7bf4b-ppv8z    1/1     Running   0          22s
[root@master ~]#
```

- **kubectl expose deployment nginx-recreate --type=NodePort --port=80**
- **kubectl get svc nginx-recreate**

```
[root@master ~]# kubectl expose deployment nginx-recreate --type=NodePort --port=80
service/nginx-recreate exposed
[root@master ~]# kubectl get svc nginx-recreate
NAME              TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
nginx-recreate    NodePort   10.96.179.144   <none>        80:30277/TCP   10s
[root@master ~]#
```

if you check with ip addresses and with portnumber 30277





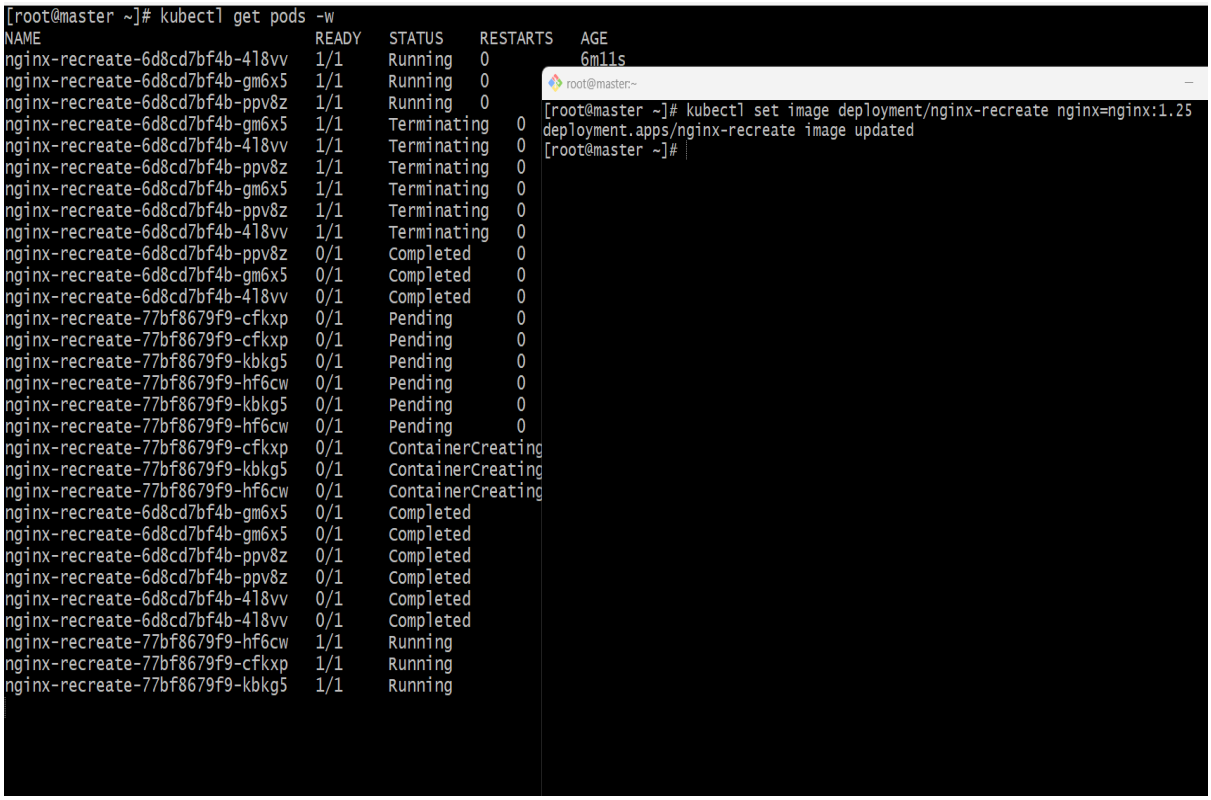
Keep as watching the pods running by using

- **kubectl get pods -w**

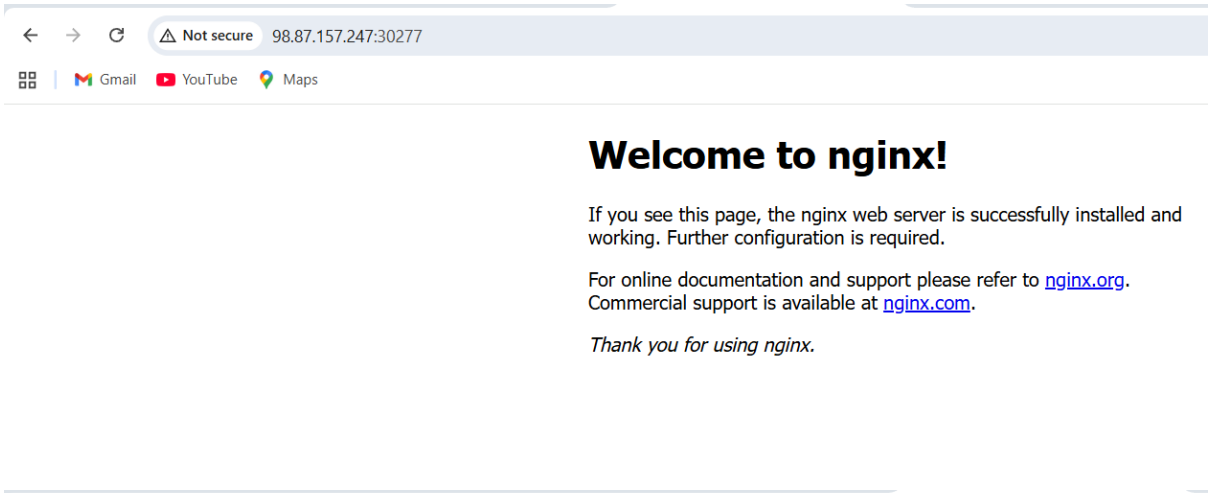
```
[root@master ~]# kubectl get pods -w
NAME                               READY   STATUS    RESTARTS   AGE
nginx-recreate-6d8cd7bf4b-4l8vv   1/1     Running   0          6m11s
nginx-recreate-6d8cd7bf4b-gm6x5   1/1     Running   0          6m11s
nginx-recreate-6d8cd7bf4b-ppv8z   1/1     Running   0          6m11s
```

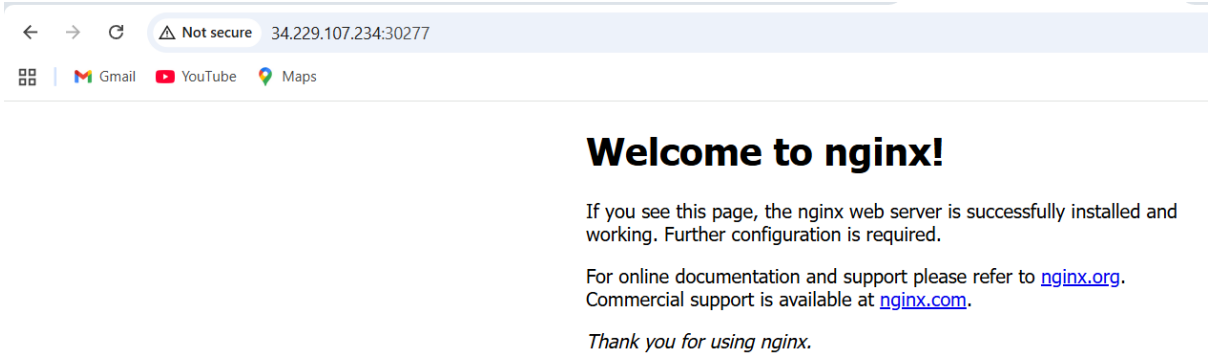
open another terminal and login same master machine and perform this.

- **kubectl set image deployment/nginx-recreate nginx=nginx:1.25**




```
[root@master ~]# kubectl get pods -w
NAME                               READY   STATUS              RESTARTS   AGE
nginx-recreate-6d8cd7bf4b-4l8vv    1/1     Running             0          6m11s
nginx-recreate-6d8cd7bf4b-gm6x5    1/1     Running             0          6m11s
nginx-recreate-6d8cd7bf4b-ppv8z    1/1     Running             0          6m11s
nginx-recreate-6d8cd7bf4b-gm6x5    1/1     Terminating         0          10m
nginx-recreate-6d8cd7bf4b-4l8vv    1/1     Terminating         0          10m
nginx-recreate-6d8cd7bf4b-ppv8z    1/1     Terminating         0          10m
nginx-recreate-6d8cd7bf4b-gm6x5    1/1     Terminating         0          10m
nginx-recreate-6d8cd7bf4b-ppv8z    1/1     Terminating         0          10m
nginx-recreate-6d8cd7bf4b-4l8vv    1/1     Terminating         0          10m
nginx-recreate-6d8cd7bf4b-ppv8z    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-gm6x5    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-4l8vv    0/1     Completed           0          10m
nginx-recreate-77bf8679f9-cfkxp    0/1     Pending             0          0s
nginx-recreate-77bf8679f9-cfkxp    0/1     Pending             0          0s
nginx-recreate-77bf8679f9-kbkg5    0/1     Pending             0          0s
nginx-recreate-77bf8679f9-hf6cw    0/1     Pending             0          0s
nginx-recreate-77bf8679f9-kbkg5    0/1     Pending             0          0s
nginx-recreate-77bf8679f9-hf6cw    0/1     Pending             0          0s
nginx-recreate-77bf8679f9-cfkxp    0/1     ContainerCreating   0          0s
nginx-recreate-77bf8679f9-kbkg5    0/1     ContainerCreating   0          0s
nginx-recreate-77bf8679f9-hf6cw    0/1     ContainerCreating   0          0s
nginx-recreate-6d8cd7bf4b-gm6x5    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-gm6x5    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-ppv8z    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-ppv8z    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-4l8vv    0/1     Completed           0          10m
nginx-recreate-6d8cd7bf4b-4l8vv    0/1     Completed           0          10m
nginx-recreate-77bf8679f9-hf6cw    1/1     Running             0          1s
nginx-recreate-77bf8679f9-cfkxp    1/1     Running             0          1s
nginx-recreate-77bf8679f9-kbkg5    1/1     Running             0          3s
^C[root@master ~]# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-recreate-77bf8679f9-cfkxp    1/1     Running   0          82s
nginx-recreate-77bf8679f9-hf6cw    1/1     Running   0          82s
nginx-recreate-77bf8679f9-kbkg5    1/1     Running   0          82s
[root@master ~]#
```





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

- **kubectl get deployments**
- **kubectl get deployments nginx-recreate -o wide**

```
[root@master ~]# kubectl get deployments
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
nginx-recreate   3/3     3            3           20m
[root@master ~]# kubectl get deployments nginx-recreate -o wide
NAME            READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES       SELECTOR
nginx-recreate   3/3     3            3           20m   nginx        nginx:1.25   app=nginx
```

- **kubectl rollout history deployment nginx-recreate**

```
[root@master ~]# kubectl rollout history deployment nginx-recreate
deployment.apps/nginx-recreate
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

- **kubectl rollout undo deployment nginx-recreate**

```
[root@master ~]# kubectl rollout undo deployment nginx-recreate
deployment.apps/nginx-recreate rolled back
[root@master ~]# kubectl get pods -w
NAME                              READY   STATUS    RESTARTS   AGE
nginx-recreate-6d8cd7bf4b-7jlsr   1/1     Running   0          29s
nginx-recreate-6d8cd7bf4b-m5h87   1/1     Running   0          29s
nginx-recreate-6d8cd7bf4b-vdhvd   1/1     Running   0          29s
```

- **kubectl get deployment nginx-recreate -o wide**

```
[root@master ~]# kubectl get deployment nginx-recreate -o wide
NAME            READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES       SELEC
TOR
nginx-recreate   3/3     3            3           22m   nginx        nginx:1.24   app=n
ginx
[root@master ~]# kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
nginx-recreate-6d8cd7bf4b-7jlsr   1/1     Running   0          92s
nginx-recreate-6d8cd7bf4b-m5h87   1/1     Running   0          92s
nginx-recreate-6d8cd7bf4b-vdhvd   1/1     Running   0          92s
[root@master ~]# kubectl rollout history deployment nginx-recreate
deployment.apps/nginx-recreate
REVISION  CHANGE-CAUSE
2         <none>
3         <none>
```

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

- **vi nginx-resources.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-resources

spec:

replicas: 3

```
strategy:
  type: RollingUpdate
selector:
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - name: nginx
      image: nginx:1.25
      ports:
      - containerPort: 80
      resources:
        requests:
          cpu: "100m"
          memory: "128Mi"
        limits:
          cpu: "200m"
```

memory: "256Mi"

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-resources
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.25
        ports:
        - containerPort: 80
        resources:
          requests:
            cpu: "100m"
            memory: "128Mi"
          limits:
            cpu: "200m"
            memory: "256Mi"
```

- **kubectl apply -f nginx-resources.yaml**
- **kubectl get deployment nginx-resources**
- **kubectl get pods**

```
[root@master ~]# vi nginx-resources.yaml
[root@master ~]# kubectl apply -f nginx-resources.yaml
deployment.apps/nginx-resources created
[root@master ~]# kubectl get deployment nginx-resources
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
nginx-resources   3/3     3            3           17s
[root@master ~]# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-resources-fb5cb79f5-2zvbr    1/1     Running   0          31s
nginx-resources-fb5cb79f5-rzsxk    1/1     Running   0          31s
nginx-resources-fb5cb79f5-sv481    1/1     Running   0          31s
```

- **kubectl describe pod nginx-resources-fb5cb79f5-2zvbr**

```
[root@master ~]# kubectl describe pod nginx-resources-fb5cb79f5-2zvbr
Name:             nginx-resources-fb5cb79f5-2zvbr
Namespace:        default
Priority:         0
Service Account:  default
Node:             worker-01/172.31.27.50
Start Time:       Sat, 20 Dec 2025 16:30:22 +0000
Labels:           app=nginx
                  pod-template-hash=fb5cb79f5
Annotations:      <none>
Status:           Running
IP:               10.244.1.32
IPs:
  IP:           10.244.1.32
Controlled By:  ReplicaSet/nginx-resources-fb5cb79f5
Containers:
  nginx:
    Container ID:   containerd://60c0ecce9198c38620b608fe536221dd3952c0285f0eea8e1
    Image:          nginx:1.25
    Image ID:       docker.io/library/nginx@sha256:a484819eb60211f5299034ac80f6a68
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sat, 20 Dec 2025 16:30:23 +0000
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     200m
      memory:  256Mi
    Requests:
      cpu:        100m
      memory:     128Mi
    Environment:  <none>
    Mounts:
```

5. Create a Deployment with MaxSurge and MaxUnavailable configurations.

- **Vi nginx-rolling.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-rolling

spec:

```
replicas: 4
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
selector:
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - name: nginx
      image: nginx:1.25
      ports:
      - containerPort: 80
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-rolling
spec:
  replicas: 4
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.25
        ports:
        - containerPort: 80
```

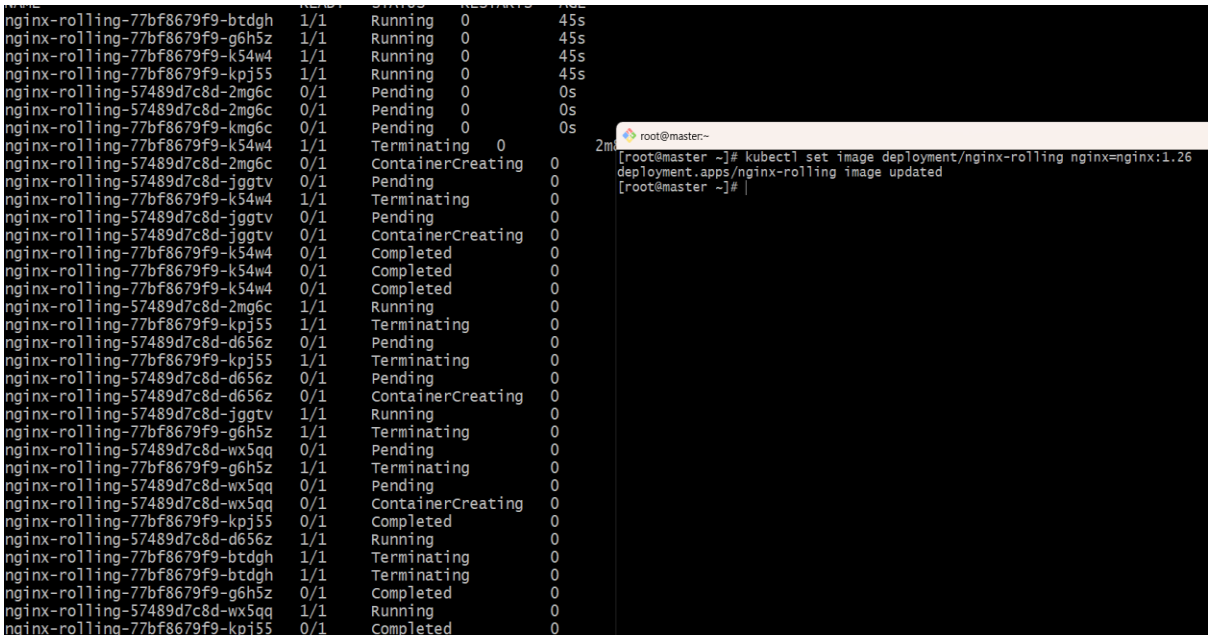
- **kubectl apply -f nginx-rolling.yaml**
- **kubectl get deployment nginx-rolling**
- **kubectl describe deployment nginx-rolling**

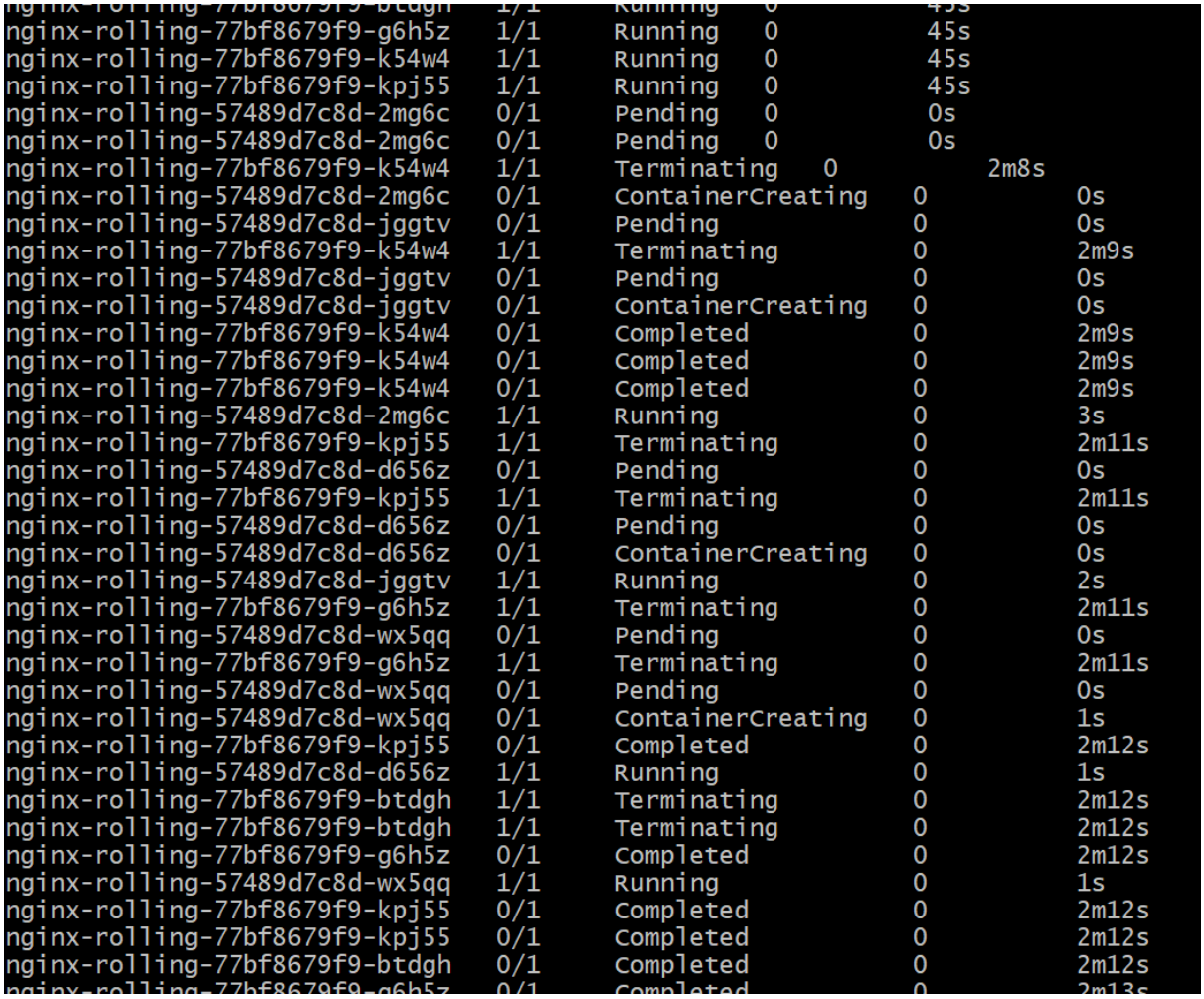
```
[root@master ~]# vi nginx-rolling.yaml
[root@master ~]# kubectl apply -f nginx-rolling.yaml
deployment.apps/nginx-rolling created
[root@master ~]# kubectl get deployment nginx-rolling
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
nginx-rolling   4/4     4            4           9s
[root@master ~]# kubectl describe deployment nginx-rolling
Name:                   nginx-rolling
Namespace:              default
CreationTimestamp:      Sat, 20 Dec 2025 16:40:30 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               4 desired | 4 updated | 4 total | 4 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  1 max unavailable, 1 max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:         nginx:1.25
    Port:          80/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
```

- **kubectl get pods -w**

observe the behaviour it will terminate one and create one.

- **kubectl set image deployment/nginx-rolling nginx=nginx:1.26**





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

- **vi nginx-revision-limit.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-revision-limit

spec:

replicas: 3

```
revisionHistoryLimit: 2
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
selector:
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - name: nginx
      image: nginx:1.25
      ports:
      - containerPort: 80
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-revision-limit
spec:
  replicas: 3
  revisionHistoryLimit: 2
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.25
        ports:
        - containerPort: 80
```

- **kubectl apply -f nginx-revision-limit.yaml**
- **kubectl describe deployment nginx-revision-limit**

```
[root@master ~]# vi nginx-revision-limit.yaml
[root@master ~]# kubectl apply -f nginx-revision-limit.yaml
deployment.apps/nginx-revision-limit created
[root@master ~]# kubectl describe deployment nginx-revision-limit
Name:                   nginx-revision-limit
Namespace:              default
CreationTimestamp:      Sat, 20 Dec 2025 17:04:16 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  1 max unavailable, 1 max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:         nginx:1.25
    Port:          80/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
```

Update the image multiple times

- **kubectl set image deployment/nginx-revision-limit nginx=nginx:1.26**
- **kubectl set image deployment/nginx-revision-limit nginx=nginx:1.27**
- **kubectl set image deployment/nginx-revision-limit nginx=nginx:1.28**
- **kubectl get rs**

```
[root@master ~]# kubectl set image deployment/nginx-revision-limit nginx=nginx:1.26
deployment.apps/nginx-revision-limit image updated
[root@master ~]# kubectl set image deployment/nginx-revision-limit nginx=nginx:1.27
deployment.apps/nginx-revision-limit image updated
[root@master ~]# kubectl set image deployment/nginx-revision-limit nginx=nginx:1.28
deployment.apps/nginx-revision-limit image updated
[root@master ~]# kubectl get rs
NAME                                DESIRED   CURRENT   READY   AGE
nginx-revision-limit-57489d7c8d     0         0         0       36s
nginx-revision-limit-7489fb554f     0         0         0       20s
nginx-revision-limit-7f9c8bf8cd     3         3         3       9s
[root@master ~]#
```

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

Use nginx-rolling yamlfile and update the image to 1.25

- **kubectl set image deployment/nginx-rolling nginx=nginx:1.25**

```
[root@master ~]# kubectl set image deployment/nginx-rolling nginx=nginx:1.25
deployment.apps/nginx-rolling image updated
```

- **kubectl rollout pause deployment nginx-rolling**

```
[root@master ~]# kubectl rollout pause deployment nginx-rolling
error: deployments.apps "nginx-rolling" is already paused
```

- **kubectl rollout resume deployment nginx-rolling**

```
[root@master ~]# kubectl rollout resume deployment nginx-rolling
deployment.apps/nginx-rolling resumed
```

- **kubectl rollout status deployment nginx-rolling**
- **kubectl get pods -w**

```
[root@master ~]# kubectl rollout status deployment nginx-rolling
deployment "nginx-rolling" successfully rolled out
[root@master ~]# kubectl get pods -w
NAME                             READY   STATUS    RESTARTS   AGE
nginx-rolling-77bf8679f9-h6dfk   1/1     Running   0          23s
nginx-rolling-77bf8679f9-hkxnc   1/1     Running   0          22s
nginx-rolling-77bf8679f9-mjxjq   1/1     Running   0          21s
nginx-rolling-77bf8679f9-vcmsf   1/1     Running   0          23s
^C[root@master ~]kubectl get deployment nginx-rolling -o widede
NAME            READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES        SELECTOR
nginx-rolling   4/4     4            4           9m43s   nginx        nginx:1.25    app=nginx
[root@master ~]#
```

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

- **vi resource-pod.yaml**

apiVersion: v1

kind: Pod

metadata:

name: resource-pod

spec:

containers:

- name: nginx

image: nginx

resources:

requests:

cpu: "500m"

memory: "256Mi"

limits:

cpu: "1"

memory: "512Mi"

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: resource-pod
spec:
  containers:
  - name: nginx
    image: nginx
    resources:
      requests:
        cpu: "500m"
        memory: "256Mi"
      limits:
        cpu: "1"
        memory: "512Mi"
```

- **kubectl apply -f resource-pod.yaml**
- **kubectl get pod resource-pod -o wide**

```
[root@master ~]# vi resource-pod.yaml
[root@master ~]# kubectl apply -f resource-pod.yaml
pod/resource-pod created
[root@master ~]# kubectl get pod resource-pod -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP           NODE        NOMINATED NODE   READINESS GATES
resource-pod   1/1     Running   0          12s   10.244.2.51   worker-02   <none>           <none>
```

- **kubectl describe pod resource-pod**

```
[root@master ~]# kubectl describe pod resource-pod
Name:             resource-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             worker-02/172.31.72.26
Start Time:       Sat, 20 Dec 2025 17:36:51 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.2.51
IPs:
  IP:  10.244.2.51
Containers:
  nginx:
    Container ID:   containerd://e208dc5a0a1e325942f8ec25bcc5a0902a155f3a8df58ed11a1
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:fb01117203ff38c2f9af91db1a7409459
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sat, 20 Dec 2025 17:36:52 +0000
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     1
      memory:  512Mi
    Requests:
      cpu:        500m
      memory:     256Mi
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-g855m (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
```

It was assigned on worker-02

- **kubectl describe node worker-02**

```
[root@master ~]# kubectl describe node worker-02
Name:               worker-02
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=worker-02
                    kubernetes.io/os=linux
Annotations:        flannel.alpha.coreos.com/backend-data: {"VNI":1,"VtepMAC":"02:29:ad:52:c0:48"}
                    flannel.alpha.coreos.com/backend-type: vxlan
                    flannel.alpha.coreos.com/kube-subnet-manager: true
                    flannel.alpha.coreos.com/public-ip: 172.31.72.26
                    node.alpha.kubernetes.io/ttl: 0
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Tue, 16 Dec 2025 16:32:11 +0000
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  worker-02
  AcquireTime:     <unset>
  RenewTime:       Sat, 20 Dec 2025 17:38:03 +0000
Conditions:
  Type                 Status  LastHeartbeatTime                 LastTransitionTime                Reason
  ----                 ------  -----------------                 ------------------                ------
  NetworkUnavailable   False   Sat, 20 Dec 2025 15:35:03 +0000   Sat, 20 Dec 2025 15:35:03 +0000   Flannel
g on this node
  MemoryPressure       False   Sat, 20 Dec 2025 17:36:37 +0000   Fri, 19 Dec 2025 15:00:46 +0000   Kubelet
cient memory available
  DiskPressure         False   Sat, 20 Dec 2025 17:36:37 +0000   Fri, 19 Dec 2025 15:00:46 +0000   Kubelet
sk pressure
  PIDPressure          False   Sat, 20 Dec 2025 17:36:37 +0000   Fri, 19 Dec 2025 15:00:46 +0000   Kubelet
cient PID available
  Ready                True    Sat, 20 Dec 2025 17:36:37 +0000   Fri, 19 Dec 2025 15:00:46 +0000   Kubelet
```

```
  pods:               110
Allocatable:
  cpu:                2
  ephemeral-storage:  19255611770
  hugepages-1Gi:      0
  hugepages-2Mi:      0
  memory:             3808856Ki
  pods:               110
System Info:
  Machine ID:                 ec20e49319d0cb0218d9174bf8e17800
  System UUID:                ec20e493-19d0-cb02-18d9-174bf8e17800
  Boot ID:                    7a470162-2749-45b3-823e-a73942549657
  Kernel Version:             6.1.158-180.294.amzn2023.x86_64
  OS Image:                   Amazon Linux 2023.9.20251208
  Operating System:           linux
  Architecture:               amd64
  Container Runtime Version:  containerd://2.1.5
  Kubelet Version:            v1.34.3
  Kube-Proxy Version:
PodCIDR:                      10.244.2.0/24
PodCIDRs:                     10.244.2.0/24
Non-terminated Pods:          (3 in total)
  Namespace                   Name                    CPU Requests  CPU Limits  Memory Requests  Memory Limits  Age
  ---------                   ----                    ------------  ----------  ---------------  -------------  ---
  default                     resource-pod            500m (25%)    1 (50%)     256Mi (6%)       512Mi (13%)    81s
  kube-flannel                kube-flannel-ds-chq7f   100m (5%)     0 (0%)      50Mi (1%)        0 (0%)         4d1h
  kube-system                 kube-proxy-9v7j7        0 (0%)        0 (0%)      0 (0%)           0 (0%)         4d1h
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource           Requests    Limits
  --------           --------    ------
  cpu                600m (30%)  1 (50%)
  memory             306Mi (8%)  512Mi (13%)
  ephemeral-storage  0 (0%)      0 (0%)
  hugepages-1Gi      0 (0%)      0 (0%)
  hugepages-2Mi      0 (0%)      0 (0%)
Events:              <none>
[root@master ~]#
```