K8s 04

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

→create a yaml file:

```
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
```

ubuntu@master: ~

```
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
```

→run yaml file:

```
ubuntu@master:~$ kubectl apply -f deployment.yaml
deployment.apps/firstdeployment created
```

→created Deployment with 3 replicas(pods):

```
ubuntu@master:~$ kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
firstdeployment   3/3     3            3           21s
ubuntu@master:~$ kubectl rollout status deployment/firstdeployment.yaml
Error from server (NotFound): deployments.apps "firstdeployment.yaml" not found
ubuntu@master:~$ kubectl rollout status deployment/firstdeployment
deployment "firstdeployment" successfully rolled out
ubuntu@master:~$ kubectl rollout history deployment/firstdeployment
deployment.apps/firstdeployment
REVISION  CHANGE-CAUSE
1         <none>
```

--creted 3 pods

```
root@master:~# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
firstdeployment-c6b5c8445-2z5pr    1/1     Running   0          11m
firstdeployment-c6b5c8445-6hv9v    1/1     Running   0          11m
firstdeployment-c6b5c8445-rjtmw    1/1     Running   0          11m
```

→ **Deployed an application using a Deployment with 3 replicas and a rolling update strategy:**

```
ubuntu@master:~$ kubectl describe deployments --recursive
Name:                   firstdeployment
Namespace:              default
CreationTimestamp:      Wed, 16 Apr 2025 09:03:03 +0000
Labels:                 appname=testapp
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               env in (prod)
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  env=prod
  Containers:
   firstcontainer:
    Image:      nginx
```

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

→create yaml file:

root@master: ~

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: firstdeployment
  labels:
    appname: testapp

spec:
  replicas: 3
  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
```

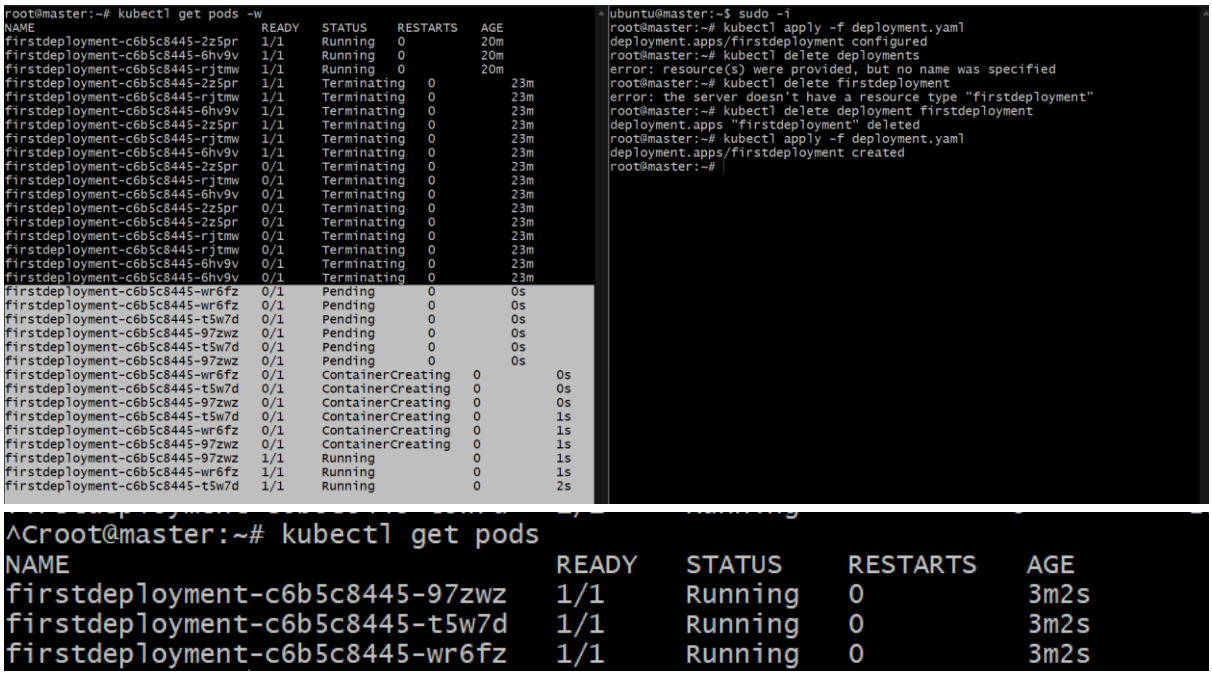
→run the yaml file:

```
root@master:~# kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
firstdeployment   3/3     3            3           15m
```

→ **Configured a Deployment with a Recreate strategy:**

```
root@master:~# kubectl describe deployments --recursive
Name:                   firstdeployment
Namespace:              default
CreationTimestamp:      Wed, 16 Apr 2025 09:26:35 +0000
Labels:                 appname=testapp
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               env in (prod)
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           Recreate
MinReadySeconds:        0
Pod Template:
  Labels:  env=prod
  Containers:
   firstcontainer:
    Image:      nginx
```

→ **observed the downtime:**




```
^Croot@master:~# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
firstdeployment-c6b5c8445-97zwz     1/1     Running   0          3m2s
firstdeployment-c6b5c8445-t5w7d     1/1     Running   0          3m2s
firstdeployment-c6b5c8445-wr6fz     1/1     Running   0          3m2s
```

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

→**first check revisions:**

```
root@master:~# kubectl rollout history deployment firstdeployment
deployment.apps/firstdeployment
REVISION  CHANGE-CAUSE
1         <none>
2         nginx changed to httpd
```

→**use this command to rollback to the previous version:**

kubectl rollout undo --to-revision=2 deployment <deployment_name>

```
root@master:~# kubectl rollout undo --to-revision=1 deployment firstdeployment
deployment.apps/firstdeployment rolled back
```

→**now check history:**

```
root@master:~# kubectl rollout history deployment firstdeployment
deployment.apps/firstdeployment
REVISION  CHANGE-CAUSE
2         nginx changed to httpd
3         <none>
```

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

→ Modified a Deployment to add resource requests and limits for CPU and memory:

root@master: ~

```
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
              value: "your_name_here"
          resources:
            requests:
              memory: "128Mi"
              cpu: "250m"
            limits:
              memory: "256Mi"
              cpu: "500m"
```

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
              value: "your_name_here"
          resources:
            requests:
              memory: "128Mi"
              cpu: "250m"
            limits:
              memory: "256Mi"
              cpu: "500m"
```

→**run the yaml file:**

```
root@master:~# kubectl apply -f deployment.yaml
deployment.apps/firstdeployment configured
```

→check with describe command:

kubectl describe pod firstdeployment-6798b47df7-75phj

```
    Restart Count:  0
    Limits:
      cpu:     500m
      memory:  256Mi
    Requests:
      cpu:     250m
      memory:  128Mi
```

5) Create a Deployment with MaxSurge and MaxUnavailable configurations.

→yaml file to Create a Deployment with MaxSurge and MaxUnavailable configurations:

root@master: ~

```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
~
```

```
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
```

→run the yaml file:

```
root@master:~# kubectl apply -f deployment.yaml
deployment.apps/firstdeployment configured
```

→check MaxSurge and MaxUnavailable configurations:

```
root@master:~# kubectl describe deployments
Name:                   firstdeployment
Namespace:              default
CreationTimestamp:      Wed, 16 Apr 2025 09:26:35 +0000
Labels:                 appname=testapp
Annotations:            deployment.kubernetes.io/revision: 5
Selector:               env in (prod)
Replicas:               6 desired | 6 updated | 8 total | 5 available | 3 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        30
RollingUpdateStrategy:  1 max unavailable, 2 max surge
```

--as MaxSurge 2 first created 8 pods then 2stoped finally 6 pods only created:

```
root@master:~# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
firstdeployment-6798b47df7-nxzr7    1/1     Running   0          17m
firstdeployment-6798b47df7-xg8rd    1/1     Running   0          17m
firstdeployment-c6b5c8445-6pzlr     1/1     Running   0          54s
firstdeployment-c6b5c8445-bmfpn     1/1     Running   0          22s
firstdeployment-c6b5c8445-c8chc     1/1     Running   0          54s
firstdeployment-c6b5c8445-cz2l6     1/1     Running   0          54s
firstdeployment-c6b5c8445-gnxx5     1/1     Running   0          22s
firstdeployment-c6b5c8445-gxw9t     1/1     Running   0          22s
root@master:~# kubectl get pods
NAME                                READY   STATUS        RESTARTS   AGE
firstdeployment-6798b47df7-nxzr7    0/1     Terminating   0          17m
firstdeployment-6798b47df7-xg8rd    0/1     Terminating   0          17m
firstdeployment-c6b5c8445-6pzlr     1/1     Running       0          65s
firstdeployment-c6b5c8445-bmfpn     1/1     Running       0          33s
firstdeployment-c6b5c8445-c8chc     1/1     Running       0          65s
firstdeployment-c6b5c8445-cz2l6     1/1     Running       0          65s
firstdeployment-c6b5c8445-gnxx5     1/1     Running       0          33s
firstdeployment-c6b5c8445-gxw9t     1/1     Running       0          33s
```

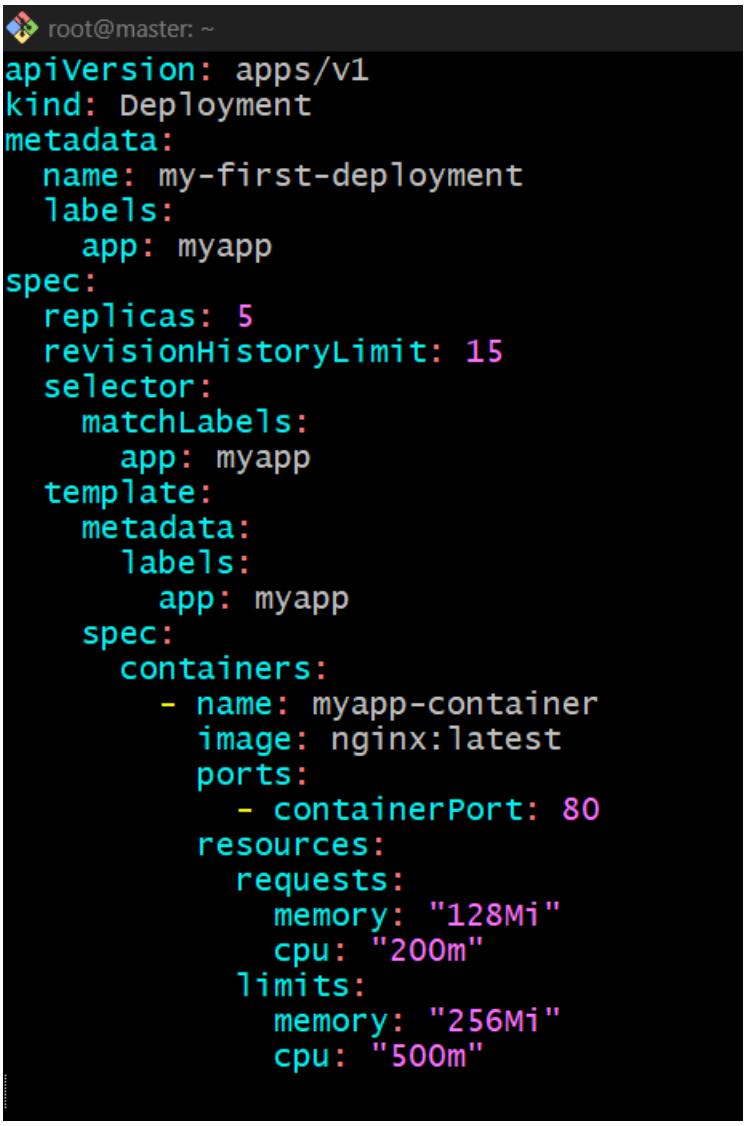
```
root@master:~# kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
firstdeployment   6/6     6            6           62m
```

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

→yaml file Set up a Deployment with a custom revision history limit:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-first-deployment
  labels:
    app: myapp
spec:
  replicas: 5
  revisionHistoryLimit: 15
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
```

```
  labels:
    app: myapp
spec:
  containers:
    - name: myapp-container
      image: nginx:latest
      ports:
        - containerPort: 80
      resources:
        requests:
          memory: "128Mi"
          cpu: "200m"
        limits:
          memory: "256Mi"
          cpu: "500m"
```



→run the yaml file:

```
root@master:~# kubectl apply -f deployment.yaml
deployment.apps/my-first-deployment created
```

→created deployment:

```
root@master:~# kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
firstdeployment       6/6     6            6           65m
my-first-deployment   5/5     5            5           13s
```

→check custom revision history limit with below command:

kubectl get deployment my-first-deployment -o yaml

```
spec:
  progressDeadlineSeconds: 600
  replicas: 5
  revisionHistoryLimit: 15
  selector:
    matchLabels:
      app: myapp
```

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

→to Pause Deployment name "my-first-deployment" use below command:

kubectl rollout pause deployment my-first-deployment

```
root@master:~# kubectl rollout pause deployment my-first-deployment
deployment.apps/my-first-deployment paused
```

→to resume a deployment use below command:

kubectl rollout resume deployment my-first-deployment

```
root@master:~# kubectl rollout resume deployment my-first-deployment
deployment.apps/my-first-deployment resumed
```

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

→ create yaml file to Create a pod using resource requests for memory and CPU:

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

```
    resources:
      requests:
        memory: "100Mi"
        cpu: "200m"
      limits:
        memory: "200Mi"
        cpu: "500m"
```

→run yaml file:

--created a pod

```
root@master:~# kubectl apply -f resource-pod.yaml
pod/resource-pod created
```

→check pods:

kubectl get pods -o wide

--Now we can see which node it was placed on by the scheduler

```
root@master:~# kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP              NODE               NOMINATED NODE   READINESS GATES
resource-pod   1/1     Running   0          18s   192.168.201.249   ip-172-31-13-158   <none>           <none>
```

→Look under the Events section we'll see lines like:

```
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  2m14s  default-scheduler  Successfully assigned default/resource-pod to ip-172-31-13-158
  Normal  Pulling    2m14s  kubelet            Pulling image "nginx"
  Normal  Pulled     2m14s  kubelet            Successfully pulled image "nginx" in 230ms (230ms including waiti
  Normal  Created    2m14s  kubelet            Created container: nginx
  Normal  Started    2m14s  kubelet            Started container nginx
root@master:~# kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
```

→Check Node Allocatable Resources with this command:

Kubectl describe node allocated node

```
root@master:~# kubectl describe node ip-172-31-13-158
Name:               ip-172-31-13-158
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ip-172-31-13-158
                    kubernetes.io/os=linux
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containe
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 172.31.13.158/20
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.201.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
```

--based on below resource-limits and requests the scheduler assigns pod to a node…

```
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource           Requests    Limits
  --------           --------    ------
  cpu                600m (30%)  500m (25%)
  memory             134Mi (3%)  298Mi (7%)
  ephemeral-storage  0 (0%)      0 (0%)
  hugepages-2Mi      0 (0%)      0 (0%)
```