Lesson-End Project

Updating Httpd Docker Images in a Kubernetes Cluster

Project agenda: To systematically update the Docker image versions of the httpd web server in a Kubernetes cluster, ensuring seamless and controlled rollouts to enhance server capabilities without disrupting service

Description: The project involves testing the rollout of different Docker images for the httpd web server within a Kubernetes cluster to ensure the cluster's efficient management of updates and versions of web server applications.

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

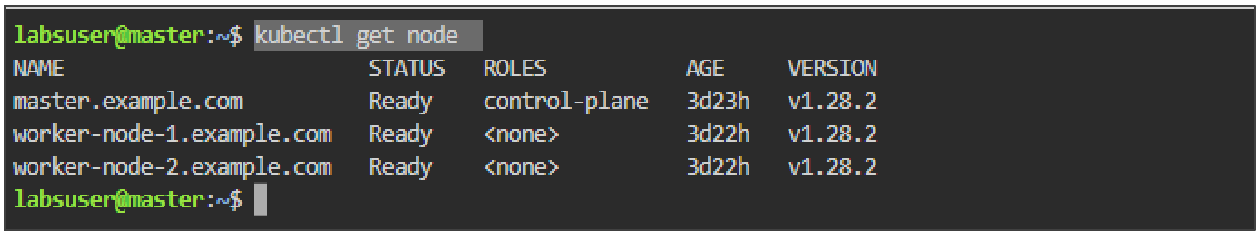
Expected deliverables: A Kubernetes cluster with the testing of httpd docker images

Steps to be followed:

1. Create the httpd deployment
2. Update the image version from httpd:2 to httpd:2.2
3. Update the image version from httpd:2.2 to httpd:2.4

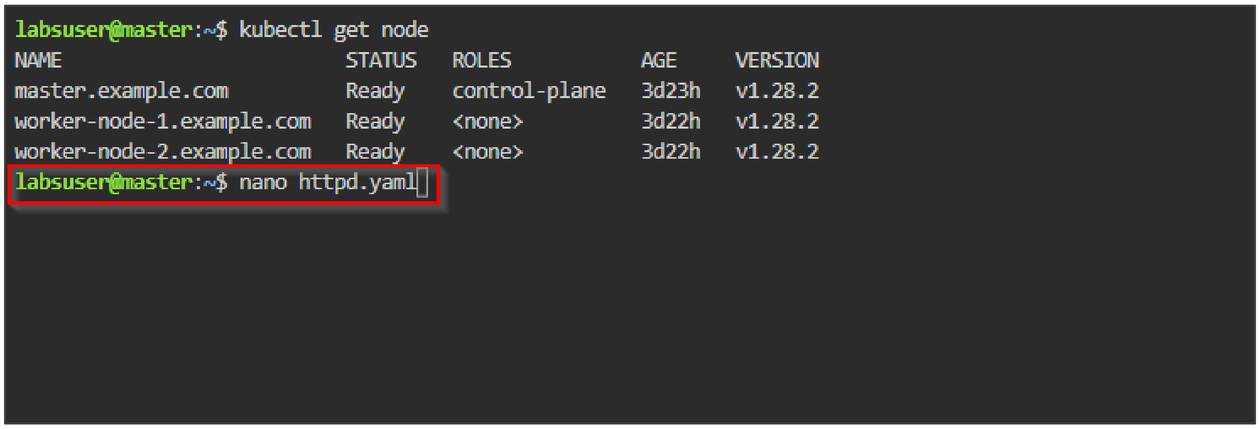
Step 1: Create the httpd deployment

1.1 Validate the connectivity between the master and worker nodes using the following command:
kubectl get node



```
labsuser@master:~$ kubectl get node
NAME                          STATUS   ROLES           AGE     VERSION
master.example.com            Ready    control-plane   3d23h   v1.28.2
worker-node-1.example.com     Ready    <none>          3d22h   v1.28.2
worker-node-2.example.com     Ready    <none>          3d22h   v1.28.2
labsuser@master:~$
```

1.2 Create the **httpd.yaml** file using the following command:
nano httpd.yaml

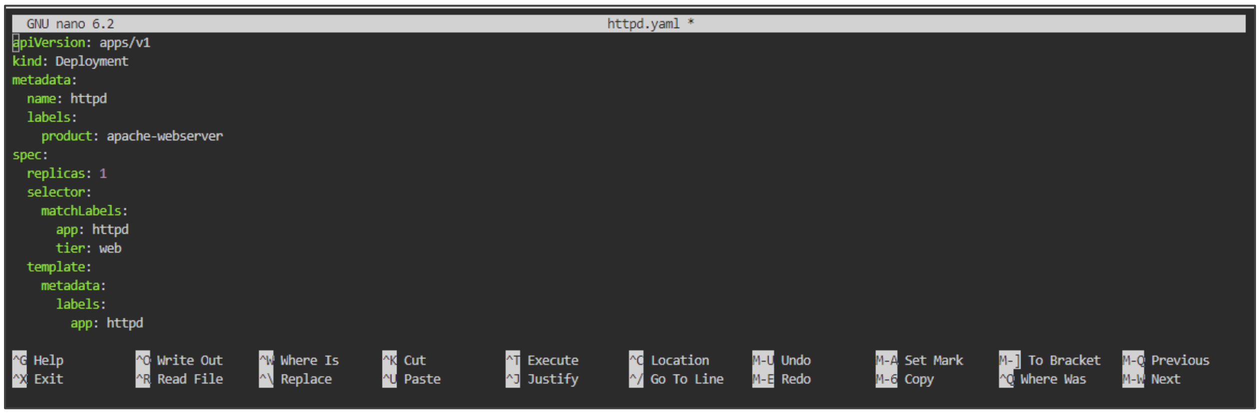


```
labsuser@master:~$ kubectl get node
NAME                         STATUS   ROLES           AGE     VERSION
master.example.com           Ready    control-plane   3d23h   v1.28.2
worker-node-1.example.com    Ready    <none>          3d22h   v1.28.2
worker-node-2.example.com    Ready    <none>          3d22h   v1.28.2
labsuser@master:~$ nano httpd.yaml
```

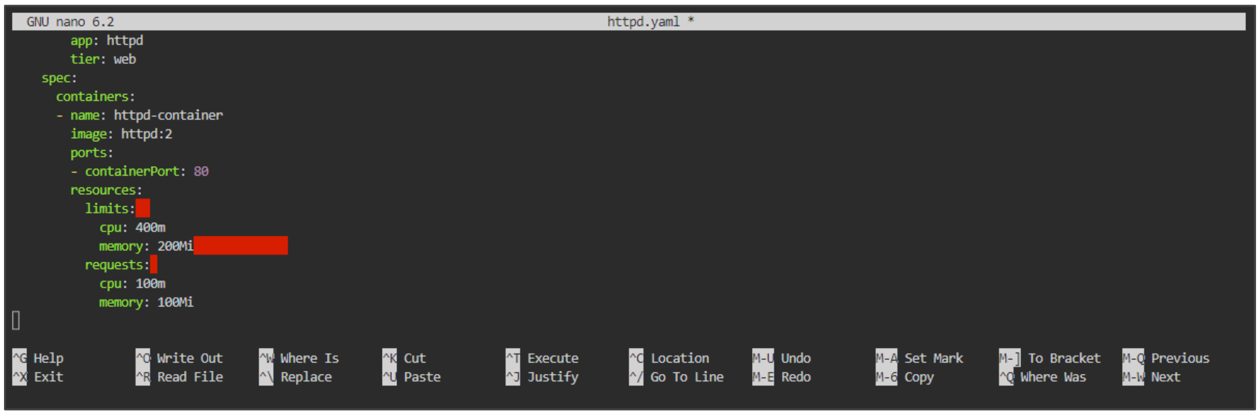
1.3 Add the following code in the **httpd.yaml** file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpd
  labels:
    product: apache-webserver
spec:
  replicas: 1
  selector:
    matchLabels:
      app: httpd
      tier: web
  template:
    metadata:
      labels:
        app: httpd
        tier: web
    spec:
      containers:
      - name: httpd-container
        image: httpd:2
        ports:
        - containerPort: 80
        resources:
          limits:
            cpu: 400m
            memory: 200Mi
```

requests:
  cpu: 100m
  memory: 100Mi

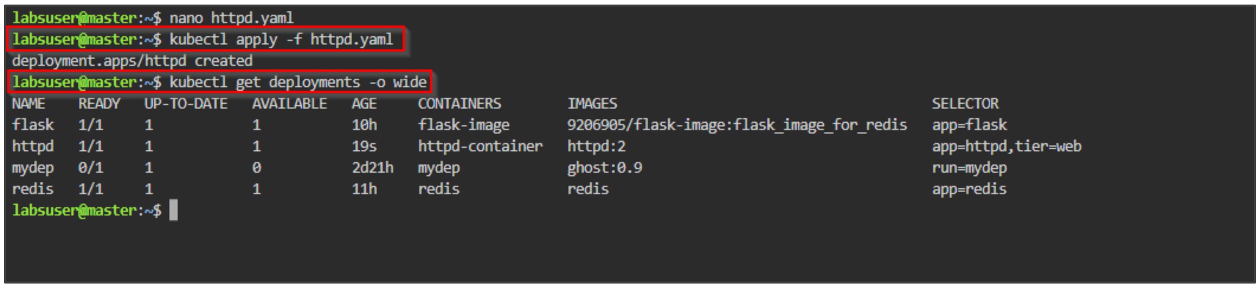


```
  GNU nano 6.2                                          httpd.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpd
  labels:
    product: apache-webserver
spec:
  replicas: 1
  selector:
    matchLabels:
      app: httpd
      tier: web
  template:
    metadata:
      labels:
        app: httpd

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket M-Q Previous
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was   M-W Next
```



```
  GNU nano 6.2                                          httpd.yaml *
        app: httpd
        tier: web
    spec:
      containers:
      - name: httpd-container
        image: httpd:2
        ports:
        - containerPort: 80
        resources:
          limits:
            cpu: 400m
            memory: 200Mi
          requests:
            cpu: 100m
            memory: 100Mi


^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket M-Q Previous
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was   M-W Next
```

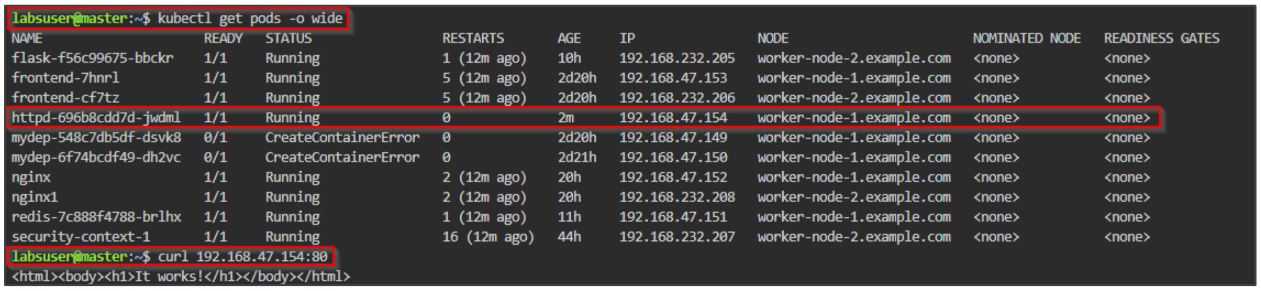
1.4 Create and validate the **httpd** deployment resource using the following commands:
kubectl apply -f httpd.yaml
kubectl get deployments -o wide



```
labsuser@master:~$ nano httpd.yaml
labsuser@master:~$ kubectl apply -f httpd.yaml
deployment.apps/httpd created
labsuser@master:~$ kubectl get deployments -o wide
NAME    READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS        IMAGES                                  SELECTOR
flask   1/1     1            1           10h    flask-image       9206905/flask-image:flask_image_for_redis   app=flask
httpd   1/1     1            1           19s    httpd-container   httpd:2                                 app=httpd,tier=web
mydep   0/1     1            0           2d21h  mydep             ghost:0.9                               run=mydep
redis   1/1     1            1           11h    redis             redis                                   app=redis
labsuser@master:~$
```

1.5 Validate if the httpd pod is working as expected using the following commands:

kubectl get pods -o wide
curl <pod_ip>:80



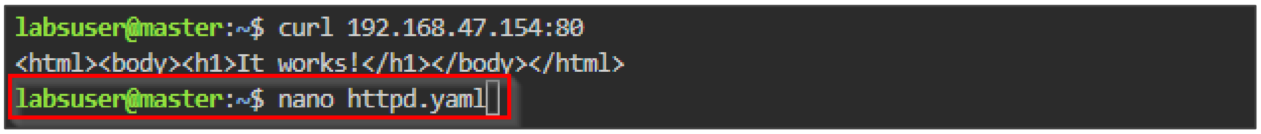
```
labsuser@master:~$ kubectl get pods -o wide
NAME                        READY   STATUS                 RESTARTS       AGE     IP                NODE                         NOMINATED NODE   READINESS GATES
flask-f56c99675-bbckr       1/1     Running                1 (12m ago)    10h     192.168.232.205   worker-node-2.example.com    <none>           <none>
frontend-7hnrl              1/1     Running                5 (12m ago)    2d20h   192.168.47.153    worker-node-1.example.com    <none>           <none>
frontend-cf7tz              1/1     Running                5 (12m ago)    2d20h   192.168.232.206   worker-node-2.example.com    <none>           <none>
httpd-696b8cdd7d-jwdml      1/1     Running                0              2m      192.168.47.154    worker-node-1.example.com    <none>           <none>
mydep-548c7db5df-dsvk8      0/1     CreateContainerError   0              2d20h   192.168.47.149    worker-node-1.example.com    <none>           <none>
mydep-6f74bcdf49-dh2vc      0/1     CreateContainerError   0              2d21h   192.168.47.150    worker-node-1.example.com    <none>           <none>
nginx                       1/1     Running                2 (12m ago)    20h     192.168.47.152    worker-node-1.example.com    <none>           <none>
nginx1                      1/1     Running                2 (12m ago)    20h     192.168.232.208   worker-node-2.example.com    <none>           <none>
redis-7c888f4788-brlhx      1/1     Running                1 (12m ago)    11h     192.168.47.151    worker-node-1.example.com    <none>           <none>
security-context-1          1/1     Running                16 (12m ago)   44h     192.168.232.207   worker-node-2.example.com    <none>           <none>
labsuser@master:~$ curl 192.168.47.154:80
<html><body><h1>It works!</h1></body></html>
```

Note: Replace **<pod_ip>** with the IP of the **httpd** pod as shown in the screenshot above

Step 2: Update the image version from httpd:2 to httpd:2.2

2.1 Open the **httpd.yaml** manifest file using the following command:

nano httpd.yaml



```
labsuser@master:~$ curl 192.168.47.154:80
<html><body><h1>It works!</h1></body></html>
labsuser@master:~$ nano httpd.yaml
```

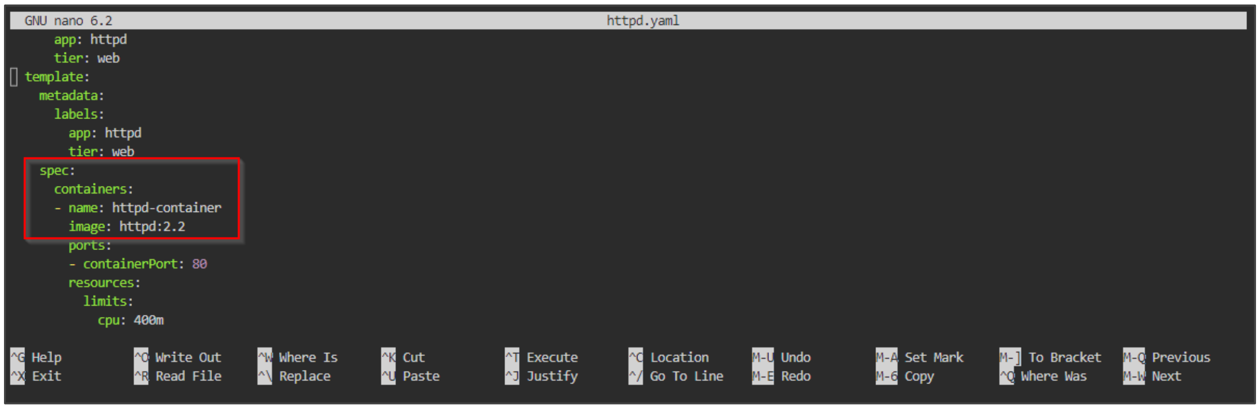
2.2 Use the following code to change the image value from **httpd:2** to **httpd:2.2**:

From:

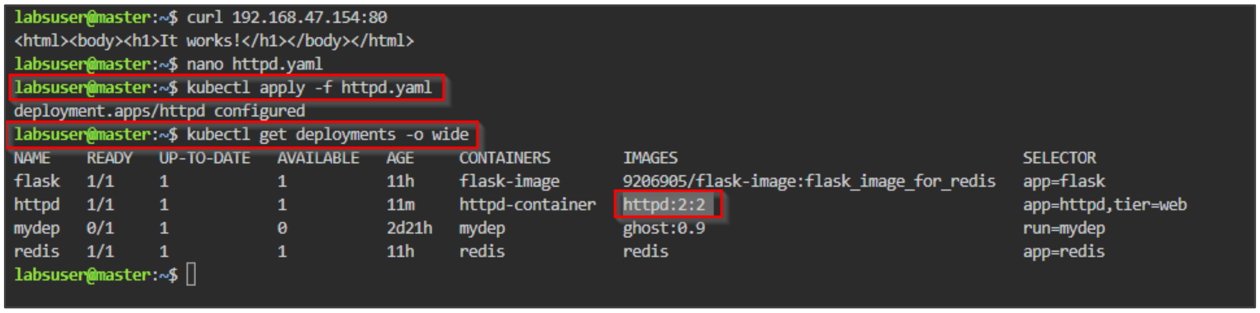
```
spec:
  containers:
  - name: httpd-container
    image: httpd:2
```

To:

```
spec:
  containers:
  - name: httpd-container
    image: httpd:2.2
```

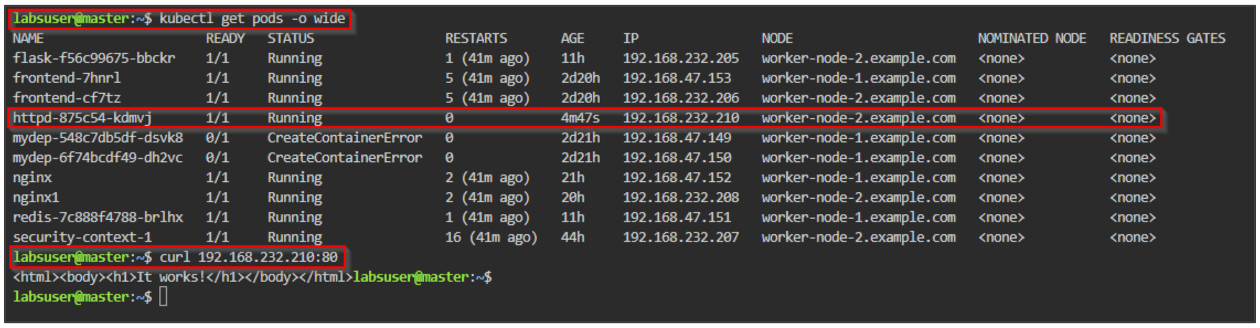

```
  GNU nano 6.2                                   httpd.yaml
      app: httpd
      tier: web
  template:
    metadata:
      labels:
        app: httpd
        tier: web
    spec:
      containers:
      - name: httpd-container
        image: httpd:2.2
        ports:
        - containerPort: 80
        resources:
          limits:
            cpu: 400m

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was   M-W Next
```

2.3 Apply and validate the changes made to the **httpd.yaml** file using the following commands:
kubectl apply -f httpd.yaml
kubectl get deployments -o wide


```
labsuser@master:~$ curl 192.168.47.154:80
<html><body><h1>It works!</h1></body></html>
labsuser@master:~$ nano httpd.yaml
labsuser@master:~$ kubectl apply -f httpd.yaml
deployment.apps/httpd configured
labsuser@master:~$ kubectl get deployments -o wide
NAME    READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS        IMAGES                                    SELECTOR
flask   1/1     1            1           11h     flask-image       9206905/flask-image:flask_image_for_redis   app=flask
httpd   1/1     1            1           11m     httpd-container   httpd:2:2                                 app=httpd,tier=web
mydep   0/1     1            0           2d21h   mydep             ghost:0.9                                 run=mydep
redis   1/1     1            1           11h     redis             redis                                     app=redis
labsuser@master:~$
```

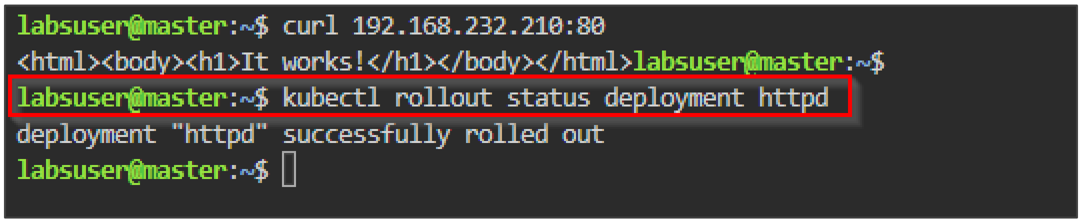
Note: The previous pod is deleted and a new pod is created with an updated Docker image.

2.4 Validate the Docker image again by fetching the updated pod IP and check for a response using the following commands:
kubectl get pods -o wide
curl <pod_ip>:80



```
labsuser@master:~$ kubectl get pods -o wide
NAME                      READY   STATUS                 RESTARTS       AGE     IP                NODE                       NOMINATED NODE   READINESS GATES
flask-f56c99675-bbckr     1/1     Running                1 (41m ago)    11h     192.168.232.205   worker-node-2.example.com  <none>           <none>
frontend-7hnrl            1/1     Running                5 (41m ago)    2d20h   192.168.47.153    worker-node-1.example.com  <none>           <none>
frontend-cf7tz            1/1     Running                5 (41m ago)    2d20h   192.168.232.206   worker-node-2.example.com  <none>           <none>
httpd-875c54-kdmvj        1/1     Running                0              4m47s   192.168.232.210   worker-node-2.example.com  <none>           <none>
mydep-548c7db5df-dsvk8    0/1     CreateContainerError   0              2d21h   192.168.47.149    worker-node-1.example.com  <none>           <none>
mydep-6f74bcdf49-dh2vc    0/1     CreateContainerError   0              2d21h   192.168.47.150    worker-node-1.example.com  <none>           <none>
nginx                     1/1     Running                2 (41m ago)    21h     192.168.47.152    worker-node-1.example.com  <none>           <none>
nginx1                    1/1     Running                2 (41m ago)    20h     192.168.232.208   worker-node-2.example.com  <none>           <none>
redis-7c888f4788-brlhx    1/1     Running                1 (41m ago)    11h     192.168.47.151    worker-node-1.example.com  <none>           <none>
security-context-1        1/1     Running                16 (41m ago)   44h     192.168.232.207   worker-node-2.example.com  <none>           <none>
labsuser@master:~$ curl 192.168.232.210:80
<html><body><h1>It works!</h1></body></html>labsuser@master:~$
labsuser@master:~$
```

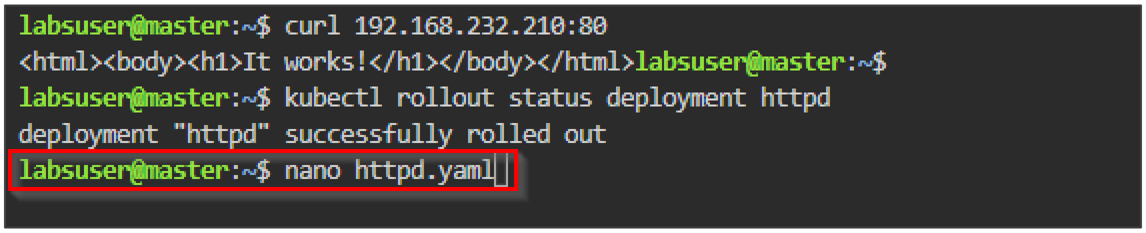
2.5 Check the rollout status using the following command:
kubectl rollout status deployment httpd



```
labsuser@master:~$ curl 192.168.232.210:80
<html><body><h1>It works!</h1></body></html>labsuser@master:~$
labsuser@master:~$ kubectl rollout status deployment httpd
deployment "httpd" successfully rolled out
labsuser@master:~$
```

Step 3: Update the image version from httpd:2.2 to httpd:2.4

3.1 Open the **httpd.yaml** manifest file using the following command:
nano httpd.yaml



```
labsuser@master:~$ curl 192.168.232.210:80
<html><body><h1>It works!</h1></body></html>labsuser@master:~$
labsuser@master:~$ kubectl rollout status deployment httpd
deployment "httpd" successfully rolled out
labsuser@master:~$ nano httpd.yaml
```

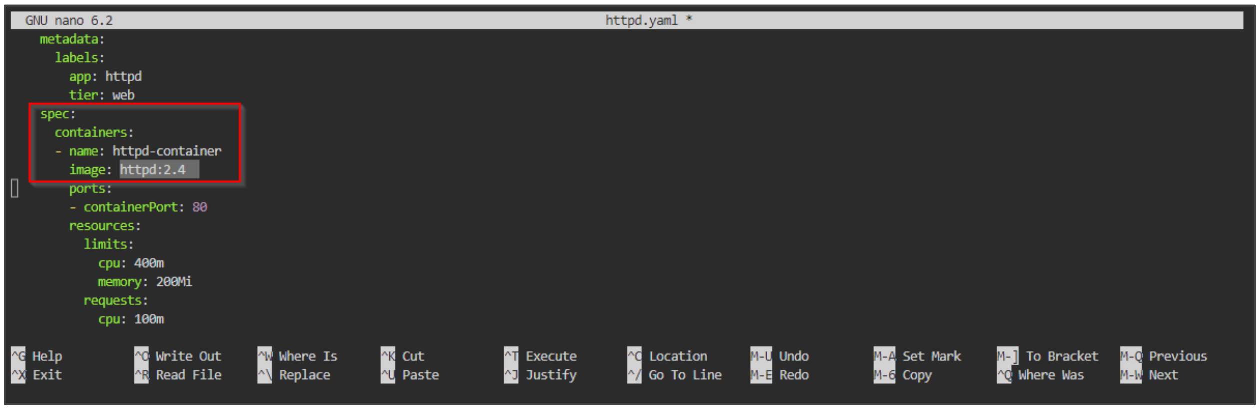
3.2 Use the following code to change the image value from **httpd:2.2** to **httpd:2.4**:

From:

```
  spec:
    containers:
    - name: httpd-container
      image: httpd:2.2
```

To:

```
  spec:
    containers:
    - name: httpd-container
      image: httpd:2.4
```

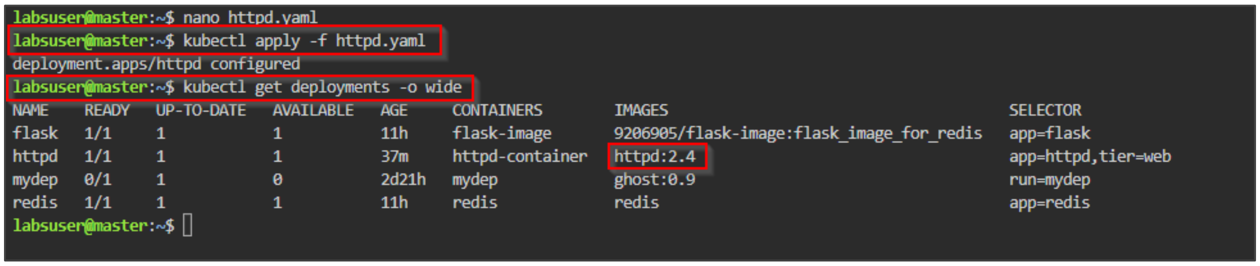


```
GNU nano 6.2                                   httpd.yaml *
  metadata:
    labels:
      app: httpd
      tier: web
  spec:
    containers:
    - name: httpd-container
      image: httpd:2.4
      ports:
      - containerPort: 80
      resources:
        limits:
          cpu: 400m
          memory: 200Mi
        requests:
          cpu: 100m
```

3.3 Apply and validate the changes made to the **httpd.yaml** file using the following commands:

kubectl apply -f httpd.yaml

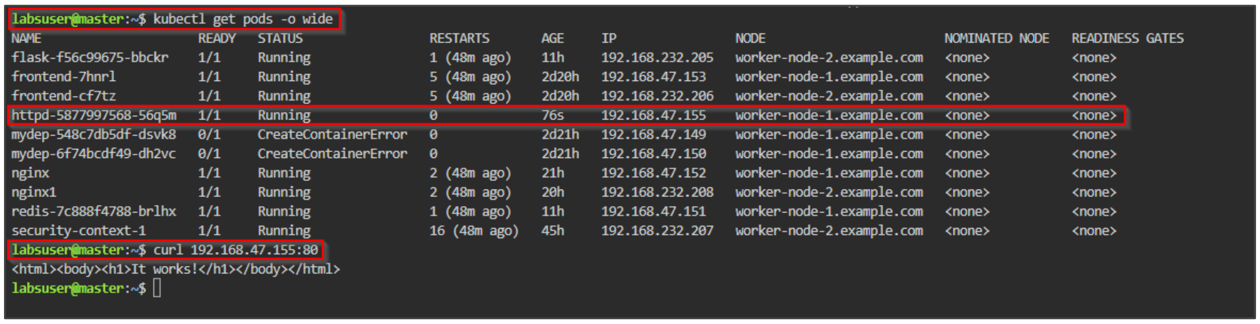
kubectl get deployments -o wide



```
labsuser@master:~$ nano httpd.yaml
labsuser@master:~$ kubectl apply -f httpd.yaml
deployment.apps/httpd configured
labsuser@master:~$ kubectl get deployments -o wide
NAME    READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS        IMAGES                                   SELECTOR
flask   1/1     1            1           11h     flask-image       9206905/flask-image:flask_image_for_redis   app=flask
httpd   1/1     1            1           37m     httpd-container   httpd:2.4                                app=httpd,tier=web
mydep   0/1     1            0           2d21h   mydep             ghost:0.9                                run=mydep
redis   1/1     1            1           11h     redis             redis                                    app=redis
labsuser@master:~$
```

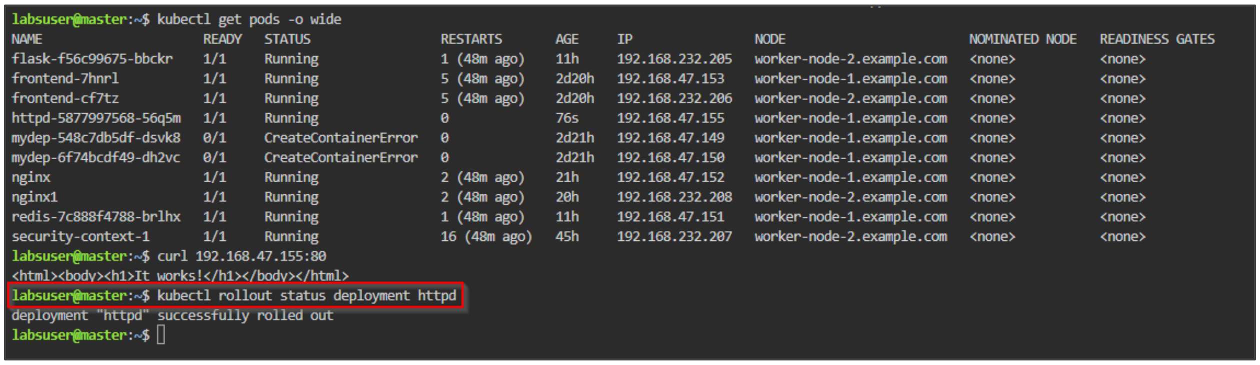
Note: The previous pod is deleted and a new pod is created with an updated Docker image.

3.4 Validate the Docker image again by fetching the updated pod IP and check for a response using the following commands:
kubectl get pods -o wide
curl <pod_ip>:80



```
labsuser@master:~$ kubectl get pods -o wide
NAME                        READY   STATUS                 RESTARTS       AGE     IP                NODE                        NOMINATED NODE   READINESS GATES
flask-f56c99675-bbckr       1/1     Running                1 (48m ago)    11h     192.168.232.205   worker-node-2.example.com   <none>           <none>
frontend-7hnrl              1/1     Running                5 (48m ago)    2d20h   192.168.47.153    worker-node-1.example.com   <none>           <none>
frontend-cf7tz              1/1     Running                5 (48m ago)    2d20h   192.168.232.206   worker-node-2.example.com   <none>           <none>
httpd-5877997568-56q5m      1/1     Running                0              76s     192.168.47.155    worker-node-1.example.com   <none>           <none>
mydep-548c7db5df-dsvk8      0/1     CreateContainerError   0              2d21h   192.168.47.149    worker-node-1.example.com   <none>           <none>
mydep-6f74bcdf49-dh2vc      0/1     CreateContainerError   0              2d21h   192.168.47.150    worker-node-1.example.com   <none>           <none>
nginx                       1/1     Running                2 (48m ago)    21h     192.168.47.152    worker-node-1.example.com   <none>           <none>
nginx1                      1/1     Running                2 (48m ago)    20h     192.168.232.208   worker-node-2.example.com   <none>           <none>
redis-7c888f4788-brlhx      1/1     Running                1 (48m ago)    11h     192.168.47.151    worker-node-1.example.com   <none>           <none>
security-context-1          1/1     Running                16 (48m ago)   45h     192.168.232.207   worker-node-2.example.com   <none>           <none>
labsuser@master:~$ curl 192.168.47.155:80
<html><body><h1>It works!</h1></body></html>
labsuser@master:~$
```

3.5 Check the rollout status using the following command:
kubectl rollout status deployment httpd



```
labsuser@master:~$ kubectl get pods -o wide
NAME                        READY   STATUS                 RESTARTS       AGE     IP                NODE                        NOMINATED NODE   READINESS GATES
flask-f56c99675-bbckr       1/1     Running                1 (48m ago)    11h     192.168.232.205   worker-node-2.example.com   <none>           <none>
frontend-7hnrl              1/1     Running                5 (48m ago)    2d20h   192.168.47.153    worker-node-1.example.com   <none>           <none>
frontend-cf7tz              1/1     Running                5 (48m ago)    2d20h   192.168.232.206   worker-node-2.example.com   <none>           <none>
httpd-5877997568-56q5m      1/1     Running                0              76s     192.168.47.155    worker-node-1.example.com   <none>           <none>
mydep-548c7db5df-dsvk8      0/1     CreateContainerError   0              2d21h   192.168.47.149    worker-node-1.example.com   <none>           <none>
mydep-6f74bcdf49-dh2vc      0/1     CreateContainerError   0              2d21h   192.168.47.150    worker-node-1.example.com   <none>           <none>
nginx                       1/1     Running                2 (48m ago)    21h     192.168.47.152    worker-node-1.example.com   <none>           <none>
nginx1                      1/1     Running                2 (48m ago)    20h     192.168.232.208   worker-node-2.example.com   <none>           <none>
redis-7c888f4788-brlhx      1/1     Running                1 (48m ago)    11h     192.168.47.151    worker-node-1.example.com   <none>           <none>
security-context-1          1/1     Running                16 (48m ago)   45h     192.168.232.207   worker-node-2.example.com   <none>           <none>
labsuser@master:~$ curl 192.168.47.155:80
<html><body><h1>It works!</h1></body></html>
labsuser@master:~$ kubectl rollout status deployment httpd
deployment "httpd" successfully rolled out
labsuser@master:~$
```

By following these steps, you have successfully updated the Docker image versions of the httpd web server in a Kubernetes cluster using a controlled rollout, ensuring enhanced server capabilities without disrupting service.