**Lesson 5 Lesson-End Project**

**Rollout Test of Httpd Docker Images in Kubernetes Cluster**

**Project agenda:** To test different Docker images of httpd web server using the rollout process

**Description:** Verify if your organization’s images given below are working using the rollout scenarios:

httpd:2.4, httpd:2.2, httpd:2.0

**Tools required:** kubeadm, kubectl, kubelet, and Docker

**Prerequisites:** kubeadm, kubectl, kubelet, and Docker must be installed

**Expected Deliverables:**  A Kubernetes cluster with the testing of httpd docker images.

Steps to be followed:

1. Setting up the cluster
2. Creating an httpd Deployment
3. Updating the image version from httpd:2 to httpd:2.2
4. Updating the image version from httpd:2.2 to httpd:2.4
5. Verifying the rollout status

**Step** **1**: **Setting up the cluster**

|  |
| --- |
| **Note:** Refer Demo 1 of Lesson 2 to set up the Kubernetes cluster |

**Step 2: Creating an httpd Deployment**

1. Validate the connectivity between the master and worker nodes using the following command:

**kubectl get node**

Text

Description automatically generated

|  |
| --- |
| **Note:** The Kubernetes cluster has been successfully configured and is ready to use. |

1. Create and open the **httpd.yaml** file with the following command:

**vi httpd.yaml**

1. 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**

**Text

Description automatically generated**

**Text

Description automatically generated with medium confidence**

1. Using the commands below, create and validate the **httpd** Deployment:

**kubectl apply -f httpd.yaml**

**kubectl get deployments -o wide**

**Text

Description automatically generated**

1. Once the Deployment is hosted on a Kubernetes cluster, validate that the httpd Pod is working as expected using the below commands to hit on the private IP of the Pod:

**kubectl get pods -o wide**

**curl <pod\_ip>:80**

**Graphical user interface, text

Description automatically generated with medium confidence**

**Step 3: Updating the image version from httpd:2 to httpd:2.2**

1. Open the httpd manifest file, and 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**

**Text

Description automatically generated**

1. Using the command below, apply the changes made to the **httpd.yaml** file:

**kubectl apply -f httpd.yaml**

**kubectl get deployments -o wide**

***Graphical user interface, text

Description automatically generated***

|  |
| --- |
| **Note:** The previous Pod will be deleted, and a new Pod will be created with an updated Docker image |

1. Validate the Docker image once again by fetching the updated Pod IP and checking for a response using the commands below:

**kubectl get pods -o wide**

**curl <pod\_ip>:80**

**Graphical user interface

Description automatically generated with medium confidence**

1. Check the rollout status using the below command:

**kubectl rollout status deployment httpd**

**Text

Description automatically generated**

**Step 4: Updating the image version from httpd:2.2 to httpd:2.4**

1. Open the httpd manifest file, and 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**

**Text

Description automatically generated**

1. Using the command given below, apply the changes made to the **httpd.yaml** file:

**kubectl apply -f httpd.yaml**

**kubectl get deployments -o wide**

***Graphical user interface, text

Description automatically generated***

|  |
| --- |
| **Note:** The previous Pod will be deleted, and a new Pod will be created with an updated Docker image |

1. Validate the Docker image once more by fetching the updated Pod IP and checking for a response using the commands below:

**kubectl get pods -o wide**

**curl <pod\_ip>:80**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Step 5: Verifying the rollout status**

1. Check the rollout status using the command given below:

**kubectl rollout status deployment httpd**

**Text

Description automatically generated**

The different Docker images of **httpd** web server using the rollout process has been tested successfully.