Lesson 4 Demo 1: Deployment Rolling Update with kubectl Rollout

This section will guide you to:

* Update deployment rolling with kubectl rollout (command)

This lab has one sub-section, namely:

1. Update deployment rolling with kubectl rollout (command)

**Note:** If you don’t have an existing Kubernetes cluster, refer to the Demo 1.1 of Lesson 1.

**Step 1:** Update deployment rolling with kubectl rollout (command)

* Start the kubernetes cluster in the lab
* As the first step, let’s create a deployment named nginx-deployment, indicated by: metadata.namefield. The yaml file will look like the one shown below:  
    
  *cat > nginx-deployment.yaml*

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

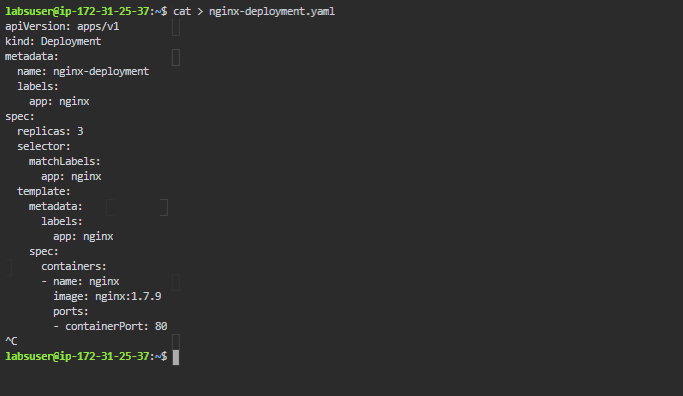
containers:

- name: nginx

image: nginx:1.7.9

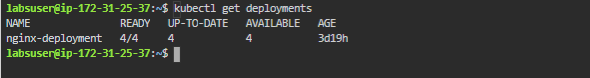
ports:

- containerPort: 80



* Next, create the deployment by running the following kubectl command:  
    
  *kubectl apply -f nginx-deployment.yaml*



* Run the **kubectl get deployments** command to check if the deployment was created. If the deployment is still being created, the output is like the one shown below:  
    
  
* To see the deployment rollout status, run **kubectl rollout status deployment.v1.apps/nginx-deployment**. The output is as shown below:
* Sometimes, you may want to rollback a deployment. For example, when the deployment is not stable, such as crash looping. Suppose you made a typo while updating the deployment, by putting the image name as nginx:1.91 instead of nginx:1.9.1, the output will be shown similar to that shown below:  
    
  *kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.91 --record=true*
* You can now check the rollout history as shown below:  
    
  *kubectl rollout history deployment.v1.apps/nginx-deployment*