Lesson 4 Demo 18: Scale an Application Up and Down Using Kubectl Commands

This section will guide you to:

* Scale an application up and down using kubectl commands

This lab has one sub-section, namely:

1. Scaling an application up and down using kubectl commands

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

**Step 1:** Scaling an application up and down using kubectl commands

* Start the kubernetes cluster in the lab
* As the first step, let’s create a deployment named **nginx-deployment**, indicated by the .metadata.namefield. The yaml file is as shown below:

*cat > nginx-deployment.yaml*

*apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2*

*kind: Deployment*

*metadata:*

*name: nginx-deployment*

*spec:*

*selector:*

*matchLabels:*

*app: nginx*

*replicas: 2 # tells deployment to run 2 pods matching the template*

*template:*

*metadata:*

*labels:*

*app: nginx*

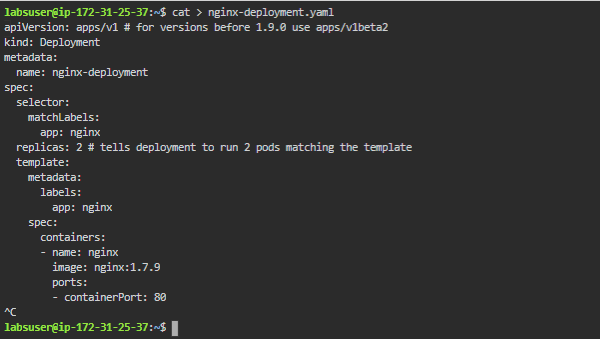
*spec:*

*containers:*

*- name: nginx*

*image: nginx:1.7.9*

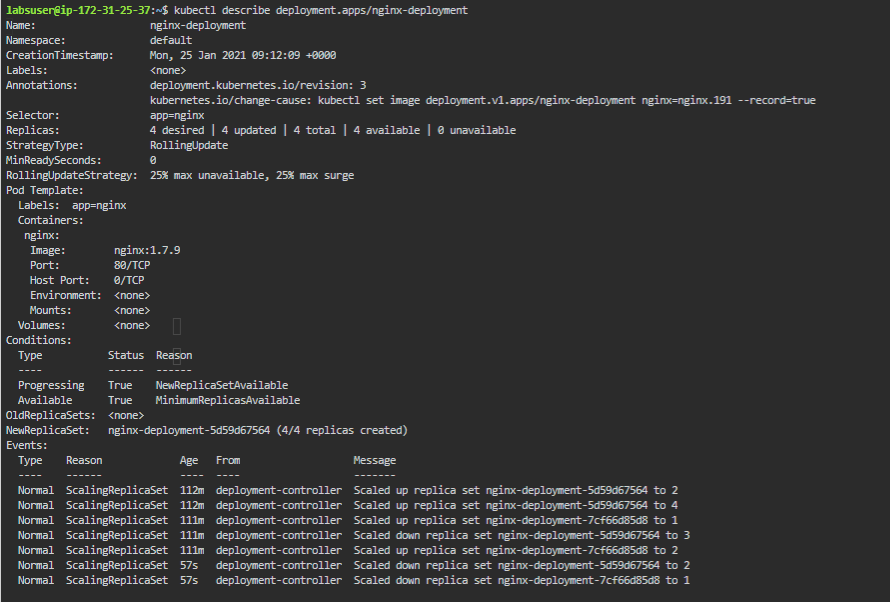
*ports:*

*- containerPort: 80*  
  


* Next, create the deployment by running the following kubectl command:

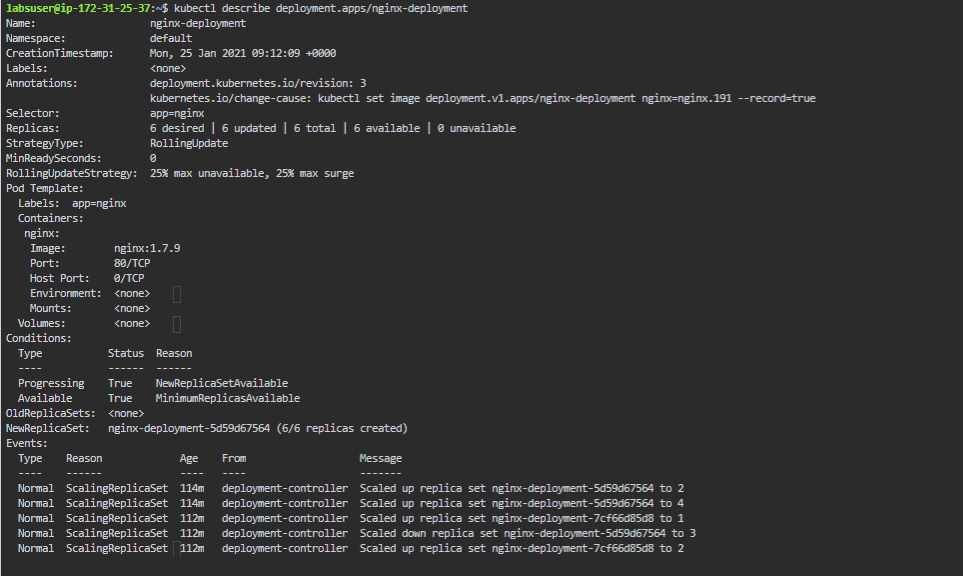
*kubectl apply -f nginx-deployment.yaml  
*

* The kubectl scale command enables you to scale one replicated deployment or service either up or down to the desired number of replicas. This command cannot be applied on services which are global mode. The command will return immediately, but the actual scaling of the deployment/service may take some time. To stop all replicas of a service while keeping the service active in the swarm, you can set the scale to 0
* Let's start with scaling up the deployment. Check the ReplicaSet of the existing deployment that was created previously, as shown below:

*kubectl describe deployment.apps/nginx-deployment*

* So here our deployment is with replica set 3
* Let’s now scale up the deployment as shown below to ReplicaSet 10:   
    
  *kubectl scale deployment.v1.apps/nginx-deployment --replicas=10  
  *
* You can verify the scaled ReplicaSet by using the kubectl describe command as shown below:

*kubectl describe deployment.apps/nginx-deployment*

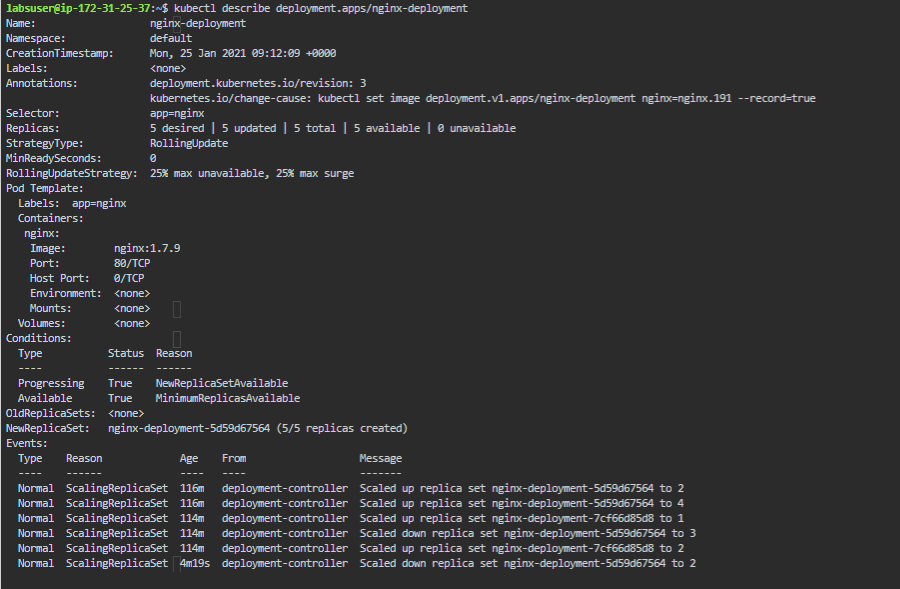
**

* The kubectl scale command also lets you scale down the ReplicaSet as shown below from 10 to 5:

*kubectl scale deployment.v1.apps/nginx-deployment --replicas=5  
*

* You can verify the services ReplicaSet using the describe command as shown below:

*kubectl describe deployment.apps/nginx-deployment*



* This is how an application can be scaled up and down using kubectl commands