Exercise 3.5: Access from Outside the Cluster

You can access a Service from outside the cluster using a DNS add-on or environment variables. We will use environment variables to gain access to a Pod.

1. Begin by getting a list of the pods.

```
student@cp:~$ kubectl get po
```

```
NAME                       READY     STATUS    RESTARTS   AGE
nginx-1423793266-13p69     1/1       Running   0          4m10s
nginx-1423793266-8w2nk     1/1       Running   0          8m2s
nginx-1423793266-fbt4b     1/1       Running   0          8m2s
```

2. Choose one of the pods and use the exec command to run **printenv** inside the pod. The following example uses the first pod listed above.

```
student@cp:~$ kubectl exec nginx-1423793266-13p69 \
    -- printenv |grep KUBERNETES
```

```
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
<output_omitted>
```

3. Find and then delete the existing service for **nginx**.

```
student@cp:~$ kubectl get svc
```

```
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   4h
nginx        ClusterIP   10.100.61.122  <none>        80/TCP    17m
```

4. Delete the service.

```
student@cp:~$ kubectl delete svc nginx
```

```
service "nginx" deleted
```

5. Create the service again, but this time pass the `LoadBalancer` type. Check to see the status and note the external ports mentioned. The output will show the `External-IP` as `pending`. Unless a provider responds with a load balancer it will continue to show as pending.

```
student@cp:~$ kubectl expose deployment nginx --type=LoadBalancer
```

```
service/nginx exposed
```

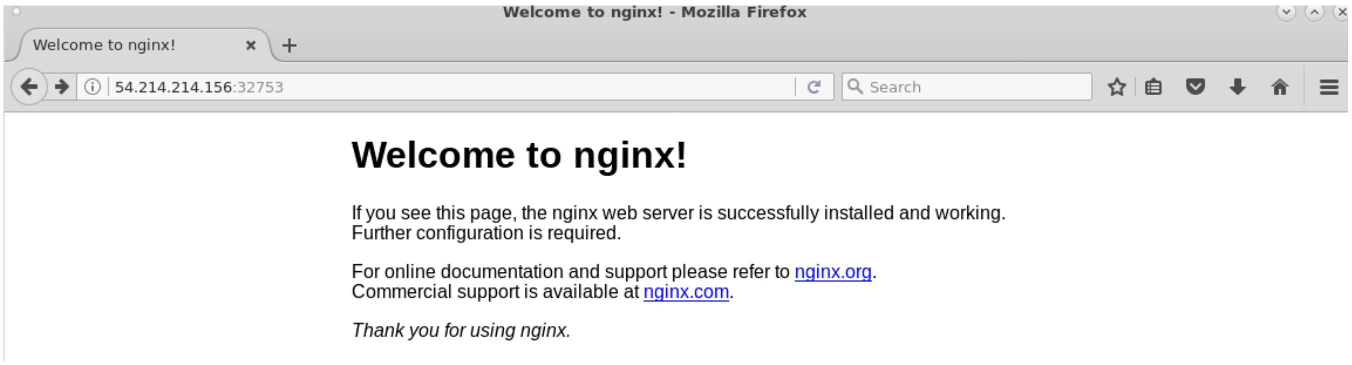
```
student@cp:~$ kubectl get svc
```

```
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP      10.96.0.1       <none>        443/TCP        4h
nginx        LoadBalancer   10.104.249.102  <pending>     80:32753/TCP   6s
```

6. Open a browser on your local system, not the lab exercise node, and use the public IP of your node and port 32753, shown in the output above. If running the labs on remote nodes like **AWS** or **GCE** use the public IP you used with PuTTY or SSH to gain access. You may be able to find the IP address using **curl**.

```
student@cp:˜$ curl ifconfig.io
```

```
54.214.214.156
```


Welcome to nginx! - Mozilla Firefox

54.214.214.156:32753

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Figure 3.1: **External Access via Browser**

7. Scale the deployment to zero replicas. Then test the web page again. Once all pods have finished terminating accessing the web page should fail.

```
student@cp:˜$ kubectl scale deployment nginx --replicas=0
```

```
deployment.apps/nginx scaled
```

```
student@cp:˜$ kubectl get po
```

```
No resources found in default namespace.
```

8. Scale the deployment up to two replicas. The web page should work again.

```
student@cp:˜$ kubectl scale deployment nginx --replicas=2
```

```
deployment.apps/nginx scaled
```

```
student@cp:˜$ kubectl get po
```

```
NAME                      READY    STATUS    RESTARTS   AGE
nginx-1423793266-7x181    1/1      Running   0          6s
nginx-1423793266-s6vcz    1/1      Running   0          6s
```

9. Delete the deployment to recover system resources. Note that deleting a deployment does not delete the endpoints or services.

```
student@cp:˜$ kubectl delete deployments nginx
```

```
deployment.apps "nginx" deleted
```

```
student@cp:˜$ kubectl delete ep nginx
```

```
endpoints "nginx" deleted
```

```
student@cp:˜$ kubectl delete svc nginx
```

```
service "nginx" deleted
```