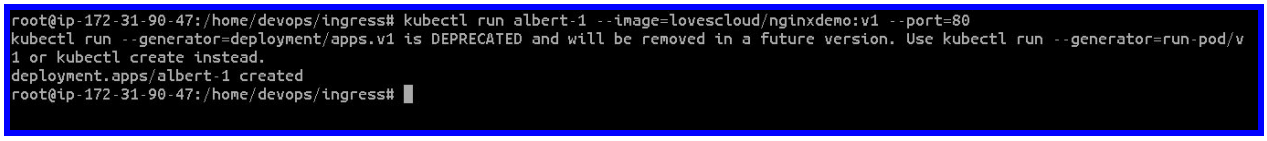
Deploying and Exposing Apps over Ingress

SSH to your AWS Workstation
ssh devops@<public-ip-addr> of your Workstation
Password is : **Dev0p$!!/**
Replace <your-name> with your name throughout the lab.

Create three Deployments by running the below commands.

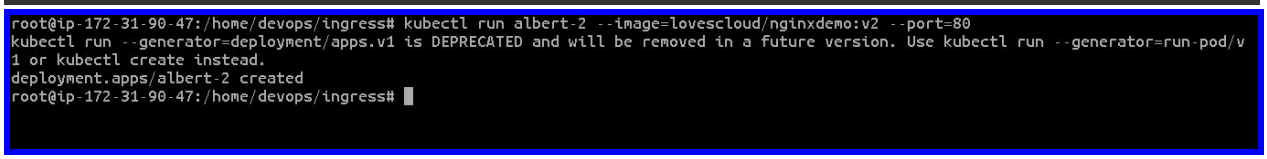
1. Deployment 1

```
$ sudo su
# mkdir ingress
# cd ingress/
# kubectl run <your-name>-1 --image=lovescloud/nginxdemo:v1 --port=80
```



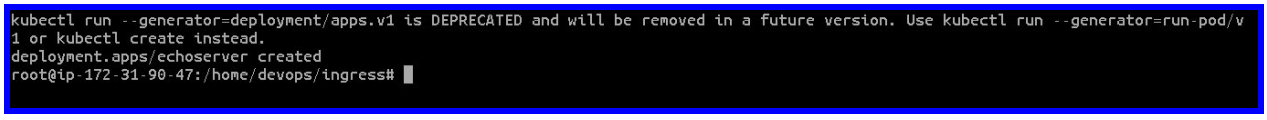
2. Deployment 2

```
# kubectl run <your-name>-2 --image=lovescloud/nginxdemo:v2 --port=80
```



3. Deployment 3 - The default Backend echo server

```
# kubectl run <your-name>-echoserver
--image=gcr.io/google_containers/echoserver:1.4 --port=8080
```



4. Check if your deployments are running

```
# kubectl get deploy
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
albert-1            1/1     1            1           5m1s
albert-2            1/1     1            1           4m33s
albert-echoserver   1/1     1            1           19s
root@ip-172-31-90-47:/home/devops/ingress#
```

5. Expose your deployments over NodePort

Deployment 1

```
# kubectl expose deployment <your-name>-1  --target-port=80 --type=NodePort
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl expose deployment albert-1  --target-port=80 --type=NodePort
service/albert-1 exposed
root@ip-172-31-90-47:/home/devops/ingress#
```

Deployment 2

```
# kubectl expose deployment <your-name>-2  --target-port=80 --type=NodePort
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl expose deployment albert-2  --target-port=80 --type=NodePort
service/albert-2 exposed
root@ip-172-31-90-47:/home/devops/ingress#
```

Deployment 3

```
# kubectl expose deployment <your-name>-echoserver --target-port=8080
--type=NodePort
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl expose deployment albert-echoserver --target-port=8080 --type=NodePort
service/albert-echoserver exposed
root@ip-172-31-90-47:/home/devops/ingress#
```

6. Check the Deployment svc

```
# kubectl get svc
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
albert-1            NodePort    10.51.250.102   <none>        80:30010/TCP     3m34s
albert-2            NodePort    10.51.247.33    <none>        80:32676/TCP     3m7s
albert-echoserver   NodePort    10.51.253.87    <none>        8080:31133/TCP   40s
kubernetes          ClusterIP   10.51.240.1     <none>        443/TCP          132m
root@ip-172-31-90-47:/home/devops/ingress#
```

7. Deploy the Ingress resource.

```
# curl -k https://pastebin.com/raw/ckUZHEhi > ingress.yaml
```

8. Now, edit the yaml file and update the below fields.

```
# vim ingress.yaml
```

<your-name> update your name with your name
<your-service1-name> with your service name for the first deployment (check step 6)
<your-service2-name> with your service name for the second deployment (check step 6)
<path1> desired first path for redirection
<path1> desired second path for redirection

To save and exit once the changes have been made
Ctrl+x (yes) and press enter to exit

Example of the deployment ingress.yaml file

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-demo
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  backend:
    serviceName: default-http-backend
    servicePort: 80
  rules:
  - http:
      paths:
      - path: /
        backend:
          serviceName: echoserver
          servicePort: 8080
  - http:
      paths:
      - path: /path1
        backend:
          serviceName: albert-1
          servicePort: 80
      - path: /path2
        backend:
          serviceName: albert-2
          servicePort: 80
```

9. Deploy the Ingress.

```
# kubectl create -f ingress.yaml
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl apply -f ingress.yaml
ingress.extensions/ingress-albert created
root@ip-172-31-90-47:/home/devops/ingress#
```

10. Check the Deployment

```
# kubectl get ing
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl get ing
NAME              HOSTS   ADDRESS         PORTS   AGE
ingress-albert    *       35.244.17.37    80      37s
root@ip-172-31-90-47:/home/devops/ingress#
```

11. Describe the Ingress

```
# kubectl describe ing <your-name>-ingress
```

```
root@ip-172-31-90-47:/home/devops# kubectl describe ing ingress-demo
Name:             ingress-demo
Namespace:        default
Address:          35.244.17.37
Default backend:  default-http-backend:80 (<none>)
Rules:
  Host  Path  Backends
  ----  ----  --------
  *
        /   echoserver:8080 (10.48.1.7:8080)
  *
        /path1   albert-1:80 (10.48.0.11:80)
        /path2   albert-2:80 (10.48.1.9:80)
Annotations:
  nginx.ingress.kubernetes.io/rewrite-target:  /
Events:
  Type     Reason     Age                 From                      Message
  ----     ------     ----                ----                      -------
  Normal   CREATE     35m                 nginx-ingress-controller  Ingress default/ingress-demo
  Normal   ADD        35m                 loadbalancer-controller   default/ingress-demo
  Normal   UPDATE     34m                 nginx-ingress-controller  Ingress default/ingress-demo
  Warning  Translate  104s (x23 over 35m)  loadbalancer-controller   error while evaluating the ingress spec: could not find service "default/default-
http-backend"
root@ip-172-31-90-47:/home/devops#
```

12. Accessing the Deployments over Ingress.

```
# kubectl get svc -n ingress-nginx
```

```
root@ip-172-31-90-47:/home/devops/ingress# kubectl get svc -n ingress-nginx
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)                      AGE
ingress-nginx   LoadBalancer   10.51.242.222   35.244.17.37   80:32098/TCP,443:31196/TCP   102m
root@ip-172-31-90-47:/home/devops/ingress#
```

From the above screenshot we can see that in this example the ingress controller service is exposed over **31196 (https) port.**
Node down the **443:Port/TCP** (second one, as in our example it is **31196)** of your ingress controller service.

13. Check the Ingress controller Node Details

```
# kubectl get po -n ingress-nginx -o wide
```

```
root@ip-172-31-90-47:/home/devops# kubectl get po -n ingress-nginx -o wide
NAME                                        READY   STATUS    RESTARTS   AGE   IP          NODE                                   NOMINATED NODE
nginx-ingress-controller-76c86d76c4-dnzr7   1/1     Running   0          28m   10.48.0.8   gke-demo-default-pool-f95dcb2e-j9jz   <none>
root@ip-172-31-90-47:/home/devops#
root@ip-172-31-90-47:/home/devops#
```

From the above screenshot we can see that in this example the ingress controller POD is running on the NODE **gke-demo-default-pool-f95dcb2e-j9jz**.

14. To get the Public IP of the **gke-demo-default-pool-f95dcb2e-j9jz** NODE, run the below command.

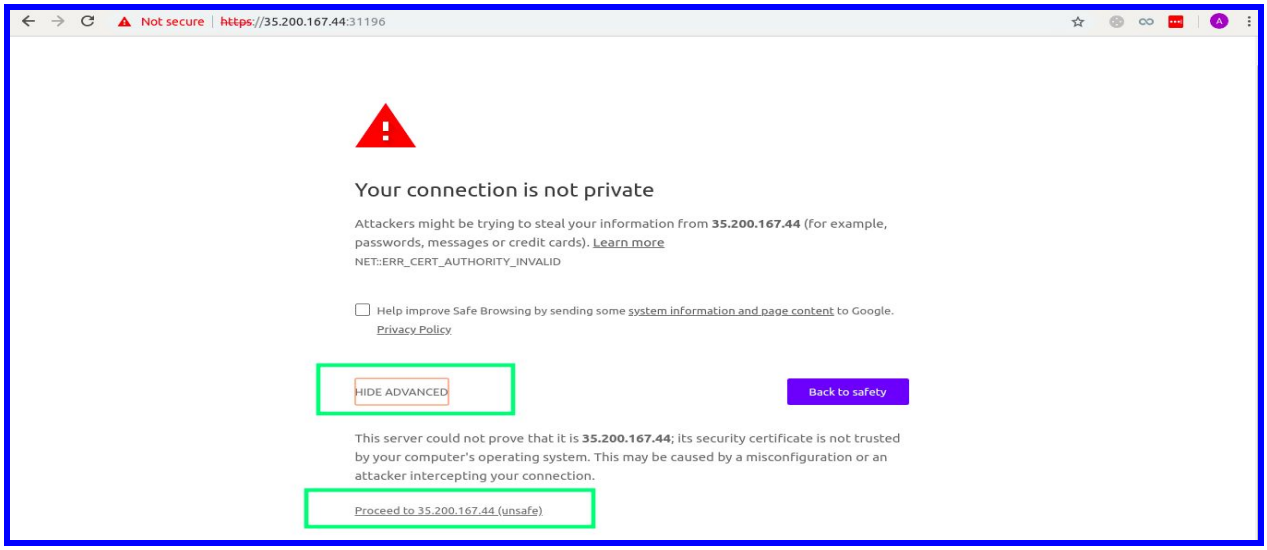
```
# kubectl get nodes -o wide
```

```
root@ip-172-31-90-47:/home/devops# kubectl get nodes -o wide
NAME                                   STATUS   ROLES    AGE   VERSION          INTERNAL-IP   EXTERNAL-IP     OS-IMAGE                             KERN
EL-VERSION   CONTAINER-RUNTIME
gke-demo-default-pool-f95dcb2e-j9jz   Ready    <none>   91m   v1.12.6-gke.10   10.160.0.23   35.200.167.44   Container-Optimized OS from Google   4.14
.91+         docker://17.3.2
gke-demo-default-pool-f95dcb2e-qvct   Ready    <none>   91m   v1.12.6-gke.10   10.160.0.24   35.244.57.29    Container-Optimized OS from Google   4.14
.91+         docker://17.3.2
root@ip-172-31-90-47:/home/devops#
```

From the above screenshot we can see that the Public IP associated with Node **gke-demo-default-pool-f95dcb2e-j9jz** is **35.200.167.44**

15. Access the Ingress Controller

https://<Node-Public-IP>:NodePort of ingress controller
Example
https://35.200.167.44:31196
Click on the Advanced button and click on Proceed to (unsafe)



16. The default backend i.e the echoserver

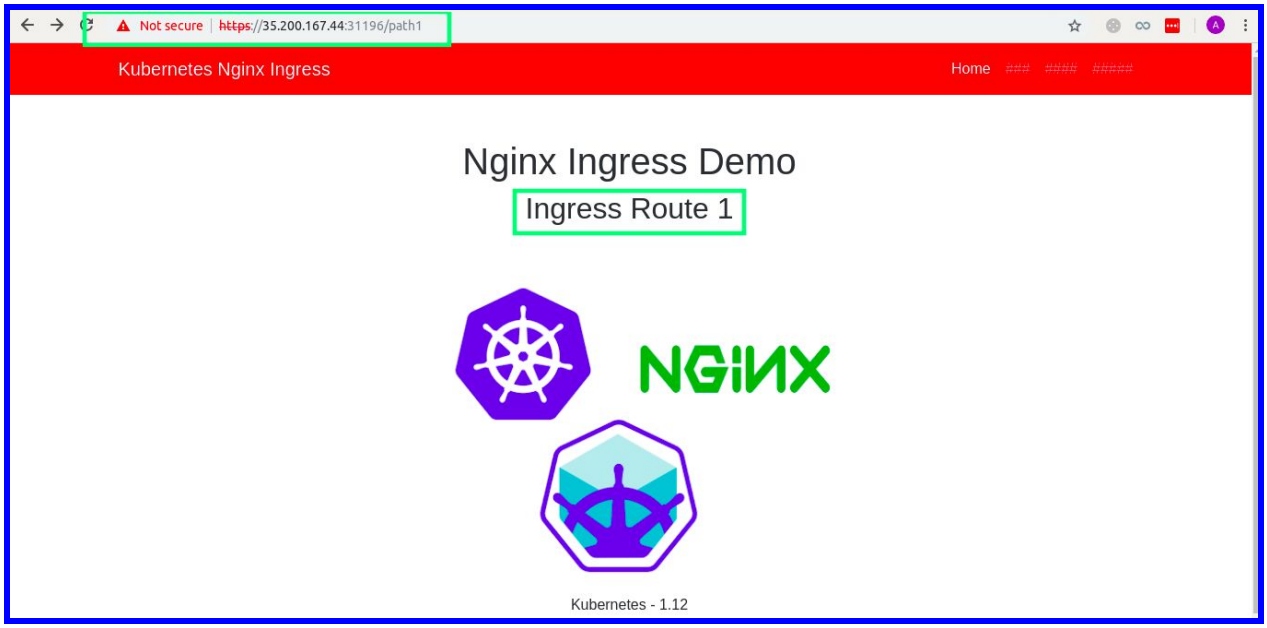
The url above will redirect the user to the default backend as we defined in our ingress deployment. The default backend for this demo is the echo server.

17. Now try to access the ingress controller through the **/paths** you specified in the **ingress.yaml**

https://<worker-node-public-ip>:NodePort/path1

Example

https://35.200.167.44:31196/path1



https://<worker-node-public-ip>:NodePort/path2
Example
https://35.200.167.44:31196/path2

18. Cleaning Up
Delete the deployments and services before moving on to the next labs in order to reduce the cluster load.

```
# kubectl get deployment
# kubectl delete deployment arg1 arg2 arg3 ..argN
```

Where arg1 arg2 arg3 are deployment names

And to delete the service we created

```
# kubectl get svc
# kubectl delete svc arg1 arg2 arg3 ..argN
```

Where arg1 arg2 arg3 are service names