Step 1 - Creating a Pod with Label NGINX

Create a YAML File

vi pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: label-demo
  labels:
    environment: production
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
```

kubectl create -f pod.yaml

```
root@kubernetes-master:/home/ubuntu# kubectl create -f pod.yaml
pod/label-demo created
```

Valide if Pod is running correctly

Kubectl get pods

```
root@kubernetes-master:/home/ubuntu# kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
label-demo   1/1     Running   0          8s
```

Step 2 - Creating a Node Port Service

vi pod-node-selector.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
 type: NodePort
 ports:
 - port: 8080
   targetPort: 80
   nodePort: 30111
 selector:
   app: nginx
```

kubectl create -f pod-node-selector.yaml

```
root@kubernetes-master:/home/ubuntu# kubectl create -f nodeport-service.yaml
service/nginx-nodeport created
```

Validate Service is running NodePort

kubectl get svc

```
root@kubernetes-master:/home/ubuntu# kubectl get svc
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP          27h
nginx-nodeport   NodePort    10.108.28.234   <none>        8080:30111/TCP   3s
```

Step 3 - Test Node Port Service

kubectl get nodes -o wide

```
root@kubernetes-master:/home/ubuntu# kubectl get nodes -o wide
NAME                STATUS   ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
kubernetes-master   Ready    master   27h   v1.18.1   10.1.0.109    <none>        Ubuntu 18.04.3 LTS   4.15.0-1057-aws     docker://19.3.8
worker-1            Ready    <none>   3h15m   v1.18.1   10.1.0.188    <none>        Ubuntu 18.04.3 LTS   4.15.0-1057-aws     docker://19.3.8
worker-2            Ready    <none>   26h   v1.18.1   10.1.0.85     <none>        Ubuntu 18.04.3 LTS   4.15.0-1057-aws     docker://19.3.8
```

curl <Kubernetes-master_IP_Addr>:30111

curl <Worker-1_IP_Addr>:30111

curl <Worker-2_IP_Addr>:30111

```
root@kubernetes-master:/home/ubuntu# curl 10.1.0.85:30111
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Step 4 - Create a Service Cluster-IP

Use the same Pod created in the last steps

Create a Service Cluster-IP YAML File

vi cluster-ip-service.yaml

```
apiVersion: v1
kind: Service
metadata:
     name: nginx
spec:
   ports:
   - port: 8080
     targetPort: 80
   selector:
      app: nginx
```

kubectl create -f cluster-ip-service.yaml

```
root@kubernetes-master:/home/ubuntu# kubectl create -f cluster-ip-service.yaml
service/nginix created
```

Validate the Cluster-IP Service and the IP address

Nginx - Cluster IP - IP Addres

```
root@kubernetes-master:/home/ubuntu# kubectl get svc
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP          27h
nginix           ClusterIP   10.109.82.47    <none>        8080/TCP         10s
```

Test the access to the Pod with the Cluster-IP Address

Example:

curl 10.109.82.47:8080

```
root@kubernetes-master:/home/ubuntu# curl http://10.109.82.47
^C
root@kubernetes-master:/home/ubuntu# curl 10.109.82.47:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Step 5 - Delete both services NodePort and Cluster-IP

kubectl delete svc nginx-nodeport

kubectl delete svc nginx

```
root@kubernetes-master:/home/ubuntu# kubectl delete svc nginx
service "nginx" deleted
```

```
root@kubernetes-master:/home/ubuntu# kubectl delete svc nginx-nodeport
service "nginx-nodeport" deleted
```