

SERVICES:

two types: 1> **cluster IP** ---- access only inside cluster. {fire curl from inside cluster }

2> **NodePort** ---- access from outside cluster { fire curl from outside cluster}

1.Cluster ip:

```
shubham@sp:~/study/k8$ cat mypod.yaml
apiVersion: v1
metadata:
  name: myapp2
  labels:
    type: app1
kind: Pod
spec:
  containers:
  - name: con1
    image: nginx
shubham@sp:~/study/k8$ kubectl apply -f mypod.yaml
pod/myapp2 created
shubham@sp:~/study/k8$ kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE     NOMINATED NODE   READINESS GATES
myapp2        0/1     ContainerCreating   0      10s   <none>        minikube <none>          <none>
shubham@sp:~/study/k8$ kubectl get pod --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
myapp2        1/1     Running   0          20s   type=app1
shubham@sp:~/study/k8$ kubectl get nodes -o wide
NAME          STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION   CONTAINER-RUNTIME
minikube      Ready     master   5h53m v1.18.3    172.17.0.3    <none>        Ubuntu 19.10      4.15.0-109-generic docker://19.3.2
shubham@sp:~/study/k8$ kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE     NOMINATED NODE   READINESS GATES
myapp2        1/1     Running   0          43s   172.18.0.3    minikube <none>          <none>
```

```
shubham@sp:~/study/k8$ kubectl expose pod myapp2 --port=8080 --target-port=80 --name=s1
service/s1 exposed
shubham@sp:~/study/k8$ kubectl get svc
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes    ClusterIP     10.96.0.1    <none>        443/TCP        13m
s1            ClusterIP     10.101.205.89 <none>        8080/TCP        7s
shubham@sp:~/study/k8$ kubectl get svc -o wide
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE   SELECTOR
kubernetes    ClusterIP     10.96.0.1    <none>        443/TCP        13m   <none>
s1            ClusterIP     10.101.205.89 <none>        8080/TCP        11s   type=app1
shubham@sp:~/study/k8$
```

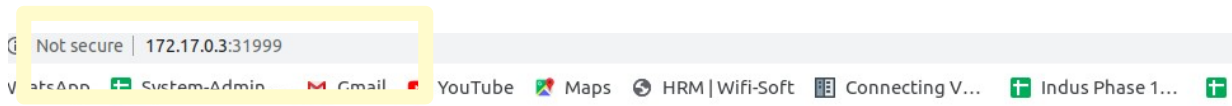
```
docker@minikube:~$ curl 10.103.216.100:8888
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
http://nginx.org/.<br/>
Commercial support is available at
http://nginx.com/.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
docker@minikube:~$
```

Inside Cluster / Node

- kubectl apply -f mypod.yaml
- kubectl get pod -show-labels
- kubectl expose pod myapp --port=8080 --target-port=80 -name=s1
- kubectl get svc

2: NodePort : access pod container from outside the cluster.

```
shubham@sp:~/study/k8$ vim nodeport.yaml
shubham@sp:~/study/k8$ kubectl apply -f nodeport.yaml
service/s1 create
shubham@sp:~/study/k8$ cat nodeport.yaml
apiVersion: v1
kind: Service
metadata:
  name: s1
spec:
  type: NodePort
  selector:
    type: app1
  ports:
    - targetPort: 80
      port: 7777
      nodePort: 31999
shubham@sp:~/study/k8$ kubectl get svc -o wide
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE    SELECTOR
kubernetes   ClusterIP    10.96.0.1      <none>          443/TCP          18m    <none>
s1           NodePort     10.98.91.118   <none>          7777:31999/TCP   14s    type=app1
shubham@sp:~/study/k8$ kubectl get node -o wide
NAME        STATUS    ROLES    AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE      KERNEL-VERSION    CONTAINER-RUNTIME
minikube    Ready    master   6h4m   v1.18.3    172.17.0.3     <none>         Ubuntu 19.10   4.15.0-109-generic docker://19.3.2
shubham@sp:~/study/k8$ curl 172.17.0.3:31999
<!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>
</body>
</html>
```



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.

```
shubham@sp:~/study/k8$ kubectl expose pod myapp2 --type=NodePort --target-port=80 --port=7777 --name=s1
service/s1 exposed
shubham@sp:~/study/k8$ kubectl get svc -o wide
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE    SELECTOR
kubernetes   ClusterIP    10.96.0.1      <none>          443/TCP          22m    <none>
s1           NodePort     10.111.135.198 <none>          7777:31534/TCP   8s     type=app1
shubham@sp:~/study/k8$ kubectl get nodes -o wide
NAME        STATUS    ROLES    AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE      KERNEL-VERSION    CONTAINER-RUNTIME
minikube    Ready    master   6h8m   v1.18.3    172.17.0.3     <none>         Ubuntu 19.10   4.15.0-109-generic docker://19.3.2
shubham@sp:~/study/k8$
```

By using command line:

`kubectl expose pod myapp2 --type=NodePort --target-port=80 --port=7777 --name=s1`
: **NodePort**

`kubectl expose pod myapp --port=8080 --target-port=80 --name=s1` : **ClusterIP**

```
shubham@sp:~/study/k8$ cat mypod.yaml
```

```
apiVersion: v1
```

```
metadata:
```

```
  name: myapp2
```

```
  labels:
```

```
    type: app1
```

```
kind: Pod
```

```
spec:
```

```
  containers:
```

```
    - name: con1
```

```
      image: nginx
```

```
shubham@sp:~/study/k8$ cat clusterip.yaml
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: s1
```

```
spec:
```

```
  selector:
```

```
    type: app1
```

```
  ports:
```

```
    - targetPort: 80
```

```
      port: 8888
```

```
shubham@sp:~/study/k8$ cat nodeport.yaml
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: s1
```

```
spec:
```

```
  type: NodePort
```

```
  selector:
```

```
    type: app1
```

```
  ports:
```

```
    - targetPort: 80
```

```
      port: 7777
```

```
      nodePort: 31999
```