

# Node port IP

A NodePort is a type of Kubernetes service that exposes an application running inside a cluster to the outside world by binding a port on every worker node.

When you create a NodePort service:

Kubernetes assigns a static port (default range: 30000-32767) on every node

# Created Nodeport for service.yml

```
C: > Users > Niree > ! service.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx-service
5  spec:
6    selector:
7      app: nginx # Corrected indentation
8    ports:
9      - protocol: TCP # Corrected "protocal" to "protocol" and added "TCP"
10      port: 80
11      targetPort: 80
12      nodeport: 30080
13    type: nodeport
14
15
```

```
PS C:\Users\Niree> kubectl apply -f service.yml
service/nginx-service configured
PS C:\Users\Niree> kubectl get svc
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
example-service     LoadBalancer  10.100.164.58 <pending>    8765:31040/TCP   7h
kubernetes           ClusterIP     10.96.0.1     <none>       443/TCP          3d20h
nginx-service       NodePort      10.111.92.27  <none>       80:30080/TCP     38m
PS C:\Users\Niree> kubectl get node -o wide
NAME                STATUS    ROLES          AGE      VERSION   INTERNAL-IP  EXTERNAL-IP  OS-IMAGE             KERNEL-VERSION
minikube            Ready     control-plane  3d21h    v1.32.0   192.168.49.2 <none>       Ubuntu 22.04.5 LTS   5.15.167.4-microsoft-standard-WSL2
docker://27.4.1
```

```
PS C:\Users\Niree> curl http://192.168.49.2:30080
curl : Unable to connect to the remote server
At line:1 char:1
+ curl http://192.168.49.2:30080
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
+ FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand

PS C:\Users\Niree> kubectl port-forward svc/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

---

# 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](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

- Create a service.yml file and define the service with NodePort.
- Set the NodePort value to 30080.
- Specify the service type as NodePort.
- Apply the service.yml file using `kubectl apply -f service.yml`.
- Retrieve the Node's IP address using `kubectl get nodes -o wide` or `minikube ip`.
- Access the service externally using `curl http://<NODE-IP>:30080`.
- If NodePort access fails, use port forwarding:
- Run `kubectl port-forward svc/nginx-service 8080:80`.
- Access the service locally via `curl http://localhost:8080`.

## Commands

- `kubectl apply -f service.yml`
- `kubectl apply -f service.yml`
- `kubectl get svc`
- `kubectl get node -o wide`
- `curl http://192.168.49.2:30080`
- `kubectl port-forward svc/nginx-service 8080:80`
- `Localhost:8080`

# Difference Between NodePort and TargetPort in Kubernetes

Feature	NodePort	TargetPort
Definition	The port on the node (host machine) that forwards traffic to the pod.	The port on the pod where the application is running.
Scope	Exposes the service externally on the node's IP at a specified port (e.g., 30080).	Exposes the service internally to other services in the cluster.
Access	Can be accessed from outside the cluster using NodeIP:NodePort.	Used for internal communication between services and pods.
Example	nodePort: 30080 (Accessible via http://<NodeIP>:30080)	targetPort: 80 (Traffic forwarded to the container running on port 80)

port: 80 # Service Port (ClusterIP)

targetPort: 80 # Pod Port (Container)

nodePort: 30080 # Exposed Port on Node

### How It Works:

- 1 A request to `http://<NodeIP>:30080` reaches the NodePort.
- 2 The request is forwarded to the port 80 on the service inside the cluster.
- 3 The service then routes the request to targetPort 80 on the pod running Nginx.