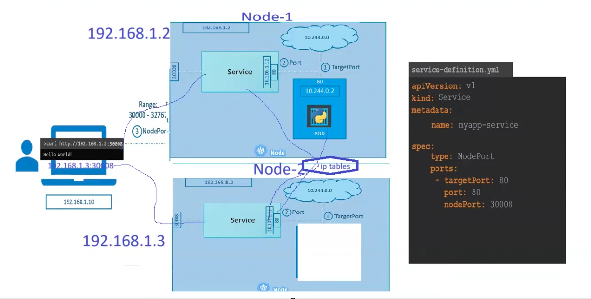
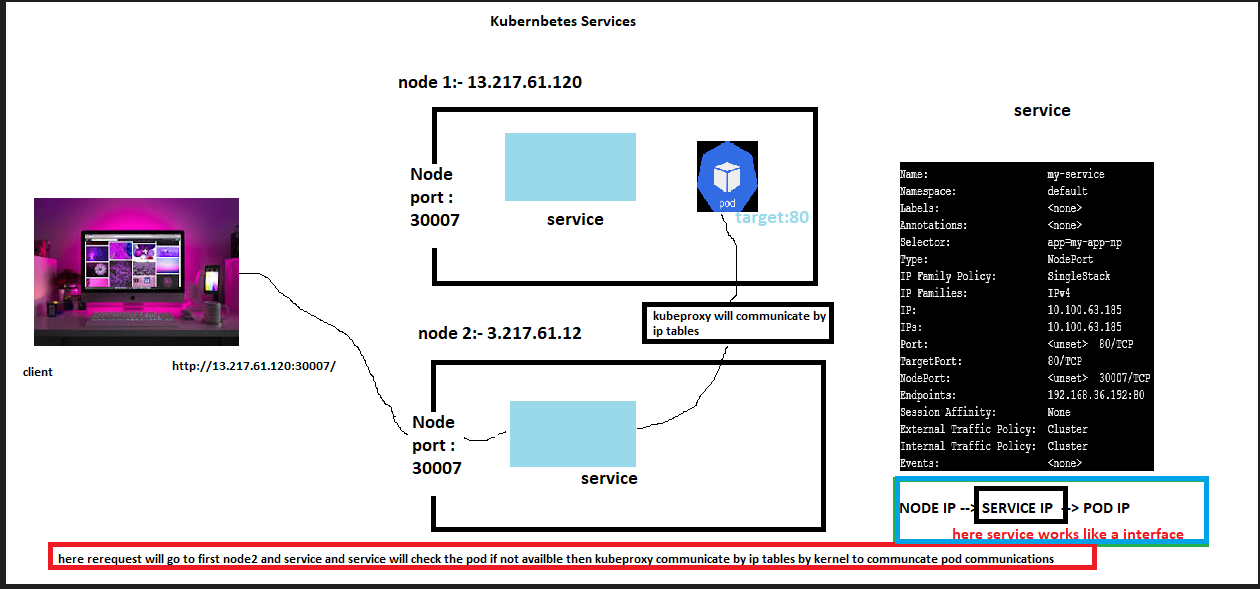


Node port represents here node communication and target ports represent target communication.





Nodeports : ---

===================

What is a NodePort?

A NodePort is a type of Kubernetes Service that exposes your application running in a Pod outside the cluster by opening a port on each Kubernetes node.

Kubernetes reserves the range 30000–32767 for NodePorts.

Traffic coming to <NodeIP>:<NodePort> will be forwarded to the underlying Pod(s).

Works with kubectl, load balancers, or directly hitting the node’s public IP.

[root@ip-172-31-26-247 ~]# vi nodeport.yml

[root@ip-172-31-26-247 ~]# kubectl apply -f nodeport.yml

deployment.apps/my-deployment-np created

service/my-service created

[root@ip-172-31-26-247 ~]# kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-192-168-27-63.ec2.internal Ready <none> 4h25m v1.32.7-eks-3abbec1

ip-192-168-37-174.ec2.internal Ready <none> 4h25m v1.32.7-eks-3abbec1

[root@ip-172-31-26-247 ~]# kubectl get pods

NAME READY STATUS RESTARTS AGE

my-deployment-np-77dcd85c54-b4hfd 1/1 Running 0 25s

my-deployment-np-77dcd85c54-p9fzx 1/1 Running 0 25s

my-deployment-np-77dcd85c54-t6pkt 1/1 Running 0 25s

[root@ip-172-31-26-247 ~]# kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 5h7m

my-service NodePort 10.100.102.177 <none> 80:30007/TCP 72s

[root@ip-172-31-26-247 ~]# kubectl describe svc

Name: kubernetes

Namespace: default

Labels: component=apiserver

provider=kubernetes

Annotations: <none>

Selector: <none>

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.0.1

IPs: 10.100.0.1

Port: https 443/TCP

TargetPort: 443/TCP

Endpoints: 192.168.122.193:443,192.168.87.159:443

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

Name: my-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=my-app-np

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.214.208

IPs: 10.100.214.208

Port: <unset> 80/TCP

TargetPort: 80/TCP

NodePort: <unset> 30007/TCP

Endpoints: 192.168.18.215:80,192.168.17.160:80,192.168.36.192:80

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

[root@ip-172-31-26-247 ~]#

[root@ip-172-31-26-247 ~]#

[root@ip-172-31-26-247 ~]#

[root@ip-172-31-26-247 ~]#

Node pods endpoints it doesnt matter which nodes pods are there.

Endpoints: 192.168.18.215:80,192.168.17.160:80,192.168.36.192:80

POD port : TargetPort: 80/TCP

Nodeport for request :

NodePort: <unset> 30007/TCP

[root@ip-172-31-26-247 ~]# kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

my-deployment-np-77dcd85c54-b4hfd 1/1 Running 0 21m 192.168.36.192 ip-192-168-37-174.ec2.internal <none> <none>

my-deployment-np-77dcd85c54-p9fzx 1/1 Running 0 21m 192.168.17.160 ip-192-168-27-63.ec2.internal <none> <none>

my-deployment-np-77dcd85c54-t6pkt 1/1 Running 0 21m 192.168.18.215 ip-192-168-27-63.ec2.internal <none> <none>

[root@ip-172-31-26-247 ~]#

Columns you get with -o wide:

NAME → Pod name

READY → How many containers in the pod are ready (1/1)

STATUS → Pod state (Running, Pending, CrashLoopBackOff, etc.)

RESTARTS → Number of times the pod has restarted

AGE → How long since the pod was created

IP → Pod’s internal cluster IP

NODE → Which Kubernetes worker node it is running on

NOMINATED NODE → Node chosen for scheduling (used in rescheduling)

READINESS GATES → Additional readiness conditions (rarely used)

=========================================================================

What is a LoadBalancer Service?

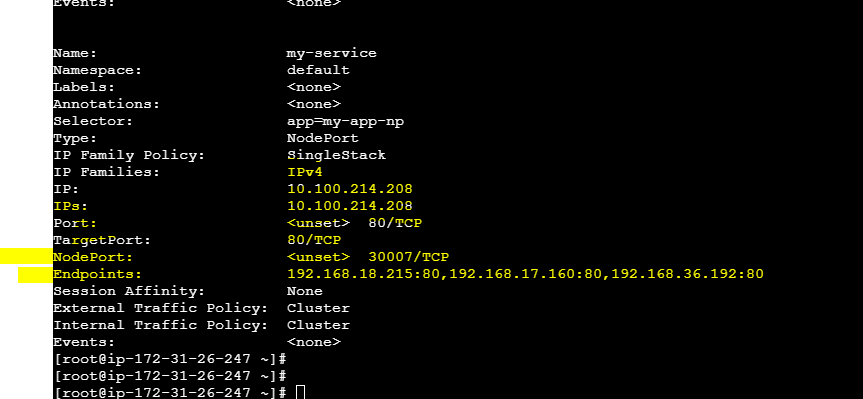
A LoadBalancer service is a Kubernetes Service type that exposes your app to the internet (or external network) using a cloud provider’s load balancer (AWS ELB/ALB, GCP LB, Azure LB, etc.).

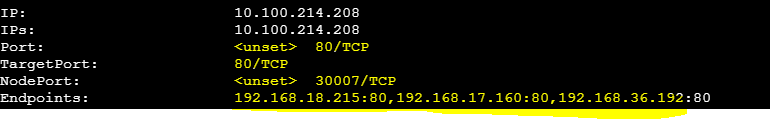
It automatically creates a cloud Load Balancer and routes traffic to your Pods.

Load balancer gets a public IP / DNS you can use to access the app.

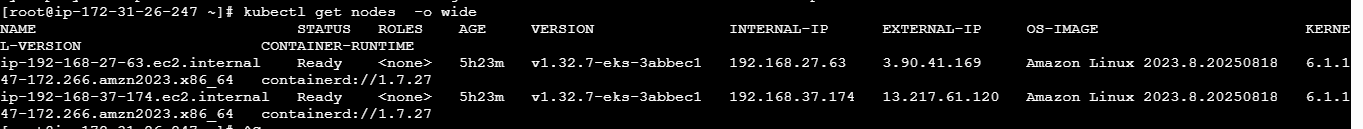
It balances traffic across all Pods behind it



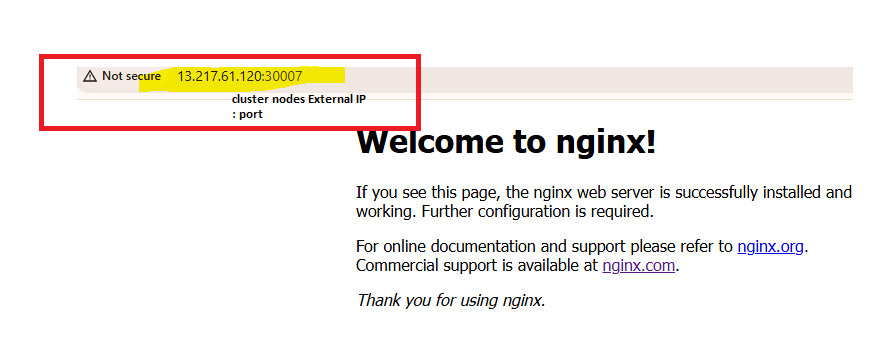




External IP is public ip clusters for nodes:



**We need to enable for SG port : 30007**



Load Balancer Service :--

Vi lb.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment-lb

spec:

replicas: 3

selector:

matchLabels:

app: my-app-lb

template:

metadata:

labels:

app: my-app-lb

spec:

containers:

- name: my-container

image: nginx:latest

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: my-app-lb

labels:

app: my-app-lb

spec:

type: LoadBalancer

ports:

- port: 80

targetPort: 80

protocol: TCP

selector:

app: my-app-lb

[root@ip-172-31-26-247 ~]# kubectl apply -f lb.yml

deployment.apps/my-deployment-lb created

service/my-app-lb created

[root@ip-172-31-26-247 ~]# kubectl get pods

NAME READY STATUS RESTARTS AGE

my-deployment-lb-589845c7bb-2kcgc 1/1 Running 0 2m48s

my-deployment-lb-589845c7bb-8b724 1/1 Running 0 2m48s

my-deployment-lb-589845c7bb-sjdgw 1/1 Running 0 2m48s

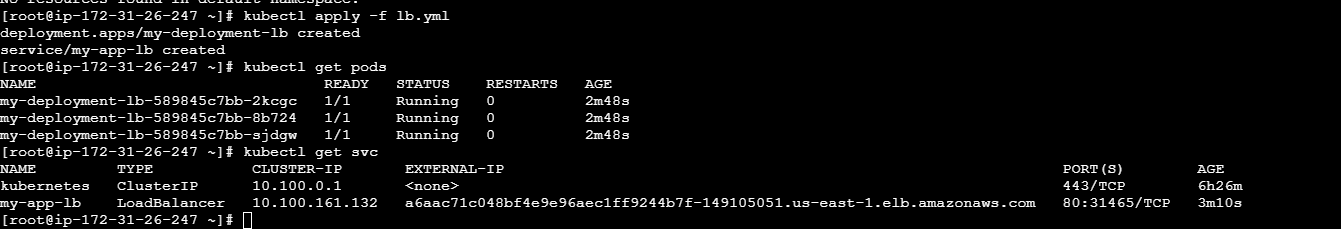
[root@ip-172-31-26-247 ~]# kubectl get svc

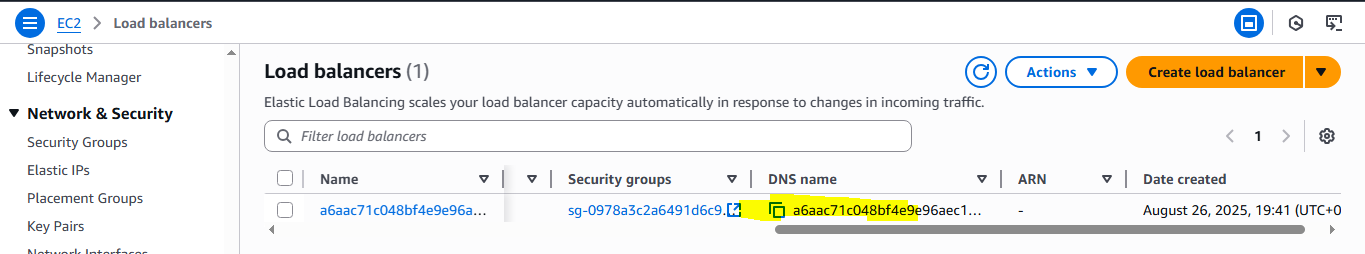
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

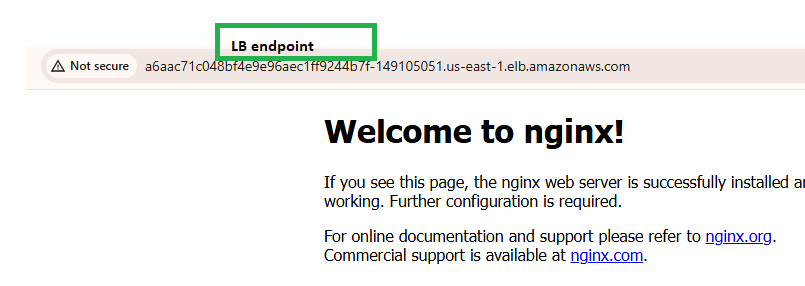
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 6h26m

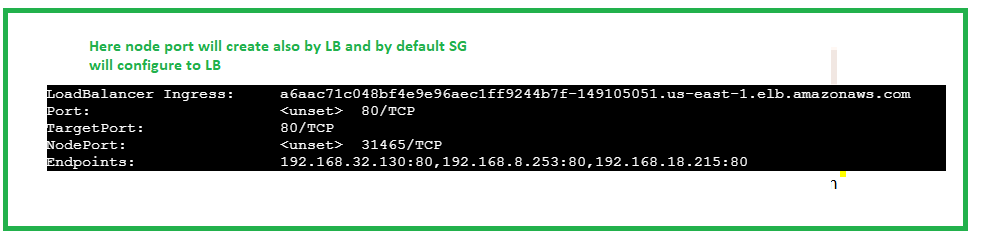
my-app-lb LoadBalancer 10.100.161.132 a6aac71c048bf4e9e96aec1ff9244b7f-149105051.us-east-1.elb.amazonaws.com 80:31465/TCP 3m10s

[root@ip-172-31-26-247 ~]#









**What is ClusterIP?**

* **ClusterIP** is the **default Service type** in Kubernetes.
* It exposes the service **inside the cluster only** (no external/public access).
* A **virtual IP (ClusterIP)** is assigned by Kubernetes, and traffic is load-balanced across matching Pods.

So, **only Pods within the same cluster** can access it

Example YAML for ClusterIP

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment

spec:

replicas: 3

selector:

matchLabels:

app: my-app

template:

metadata:

labels:

app: my-app

spec:

containers:

- name: my-container

image: nginx:latest

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: my-app-svc

labels:

app: my-app

spec:

type: ClusterIP

ports:

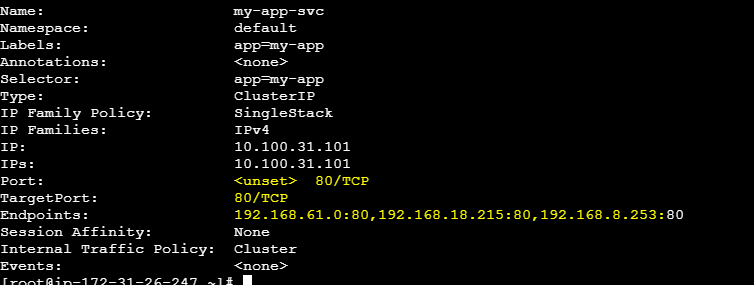
- port: 80

targetPort: 80

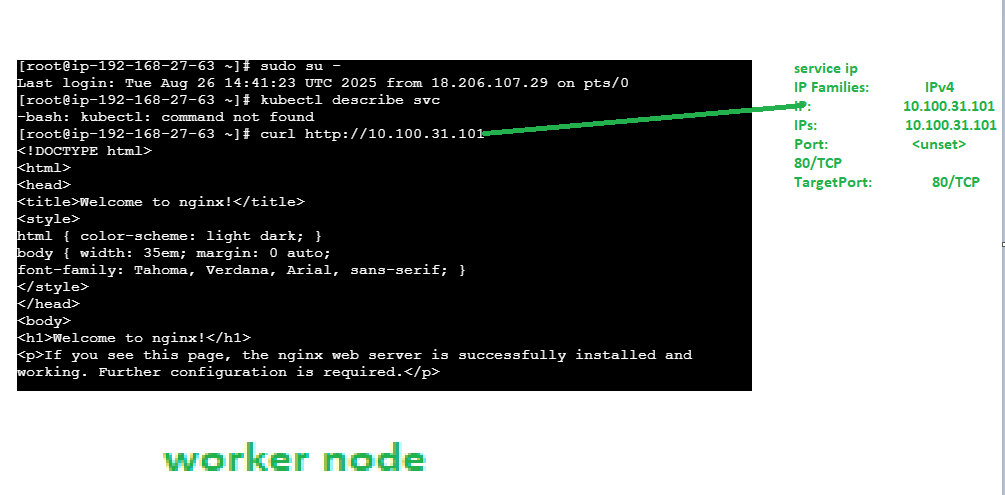
protocol: TCP

selector:

app: my-app



**To test cluster Ip we need to connect with worker node because pod is running internally in worker node so we need connect any worker node**



Last login: Tue Aug 26 14:41:23 UTC 2025 from 18.206.107.29 on pts/0

[root@ip-192-168-27-63 ~]# kubectl describe svc

-bash: kubectl: command not found

[root@ip-192-168-27-63 ~]# curl http://10.100.31.101

<!DOCTYPE html>

**Note :--**

Here FE to BE communication through cluster IP on and FE will access by LB only.

