Kubernetes Topology Aware Routing

Topology Aware Routing is a Kubernetes feature that **routes service traffic to pods within the same zone** whenever possible, improving **latency, reliability, and reducing network costs**.

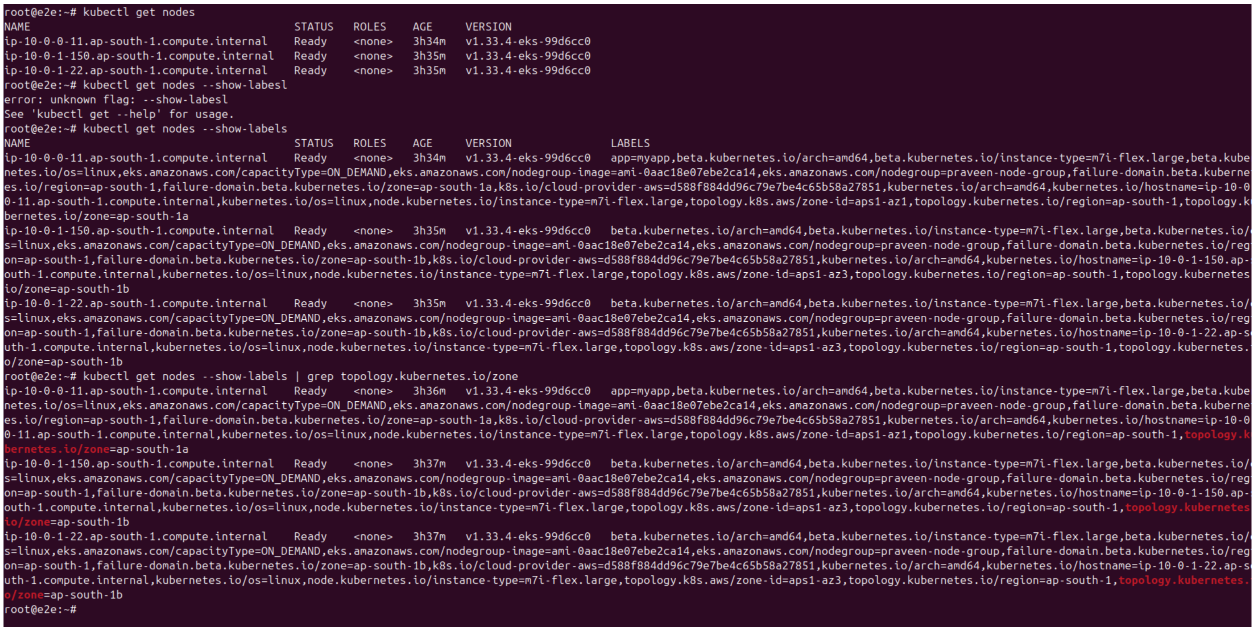
Real-life analogy:
Imagine a shopping app with servers in 3 zones (Zone A, Zone B, Zone C). A user connecting to the app from Zone A should ideally access backend servers in Zone A. This reduces **response time** and **cross-zone traffic charges**.

Prerequisites

- Kubernetes v1.23+ (v1.33 in this example)
- Multi-zone cluster (at least 2 zones) In my case i have two zones one is ap-south-1a,ap-south-1b.
- kubectl configured

Check if your cluster is multi-zone:

kubectl get nodes --show-labels



```
root@e2e:~# kubectl get nodes
NAME                                        STATUS   ROLES    AGE     VERSION
ip-10-0-0-11.ap-south-1.compute.internal    Ready    <none>   3h34m   v1.33.4-eks-99d6cc0
ip-10-0-1-150.ap-south-1.compute.internal   Ready    <none>   3h35m   v1.33.4-eks-99d6cc0
ip-10-0-1-22.ap-south-1.compute.internal    Ready    <none>   3h35m   v1.33.4-eks-99d6cc0
root@e2e:~# kubectl get nodes --show-labesl
error: unknown flag: --show-labesl
See 'kubectl get --help' for usage.
root@e2e:~# kubectl get nodes --show-labels
NAME                                        STATUS   ROLES    AGE     VERSION               LABELS
ip-10-0-0-11.ap-south-1.compute.internal    Ready    <none>   3h34m   v1.33.4-eks-99d6cc0   app=myapp,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kube
netes.io/os=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kuberne
es.io/region=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1a,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0
0-11.ap-south-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az1,topology.kubernetes.io/region=ap-south-1,topology.k
bernetes.io/zone=ap-south-1a
ip-10-0-1-150.ap-south-1.compute.internal   Ready    <none>   3h35m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/reg
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-150.ap-
outh-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes
io/zone=ap-south-1b
ip-10-0-1-22.ap-south-1.compute.internal    Ready    <none>   3h35m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/reg
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-22.ap-s
uth-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes.
o/zone=ap-south-1b
root@e2e:~# kubectl get nodes --show-labels | grep topology.kubernetes.io/zone
ip-10-0-0-11.ap-south-1.compute.internal    Ready    <none>   3h36m   v1.33.4-eks-99d6cc0   app=myapp,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kube
netes.io/os=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kuberne
es.io/region=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1a,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0
0-11.ap-south-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az1,topology.kubernetes.io/region=ap-south-1,topology.k
bernetes.io/zone=ap-south-1a
ip-10-0-1-150.ap-south-1.compute.internal   Ready    <none>   3h37m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/reg
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-150.ap-
outh-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes
io/zone=ap-south-1b
ip-10-0-1-22.ap-south-1.compute.internal    Ready    <none>   3h37m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/reg
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-22.ap-s
uth-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes.
o/zone=ap-south-1b
root@e2e:~#
```

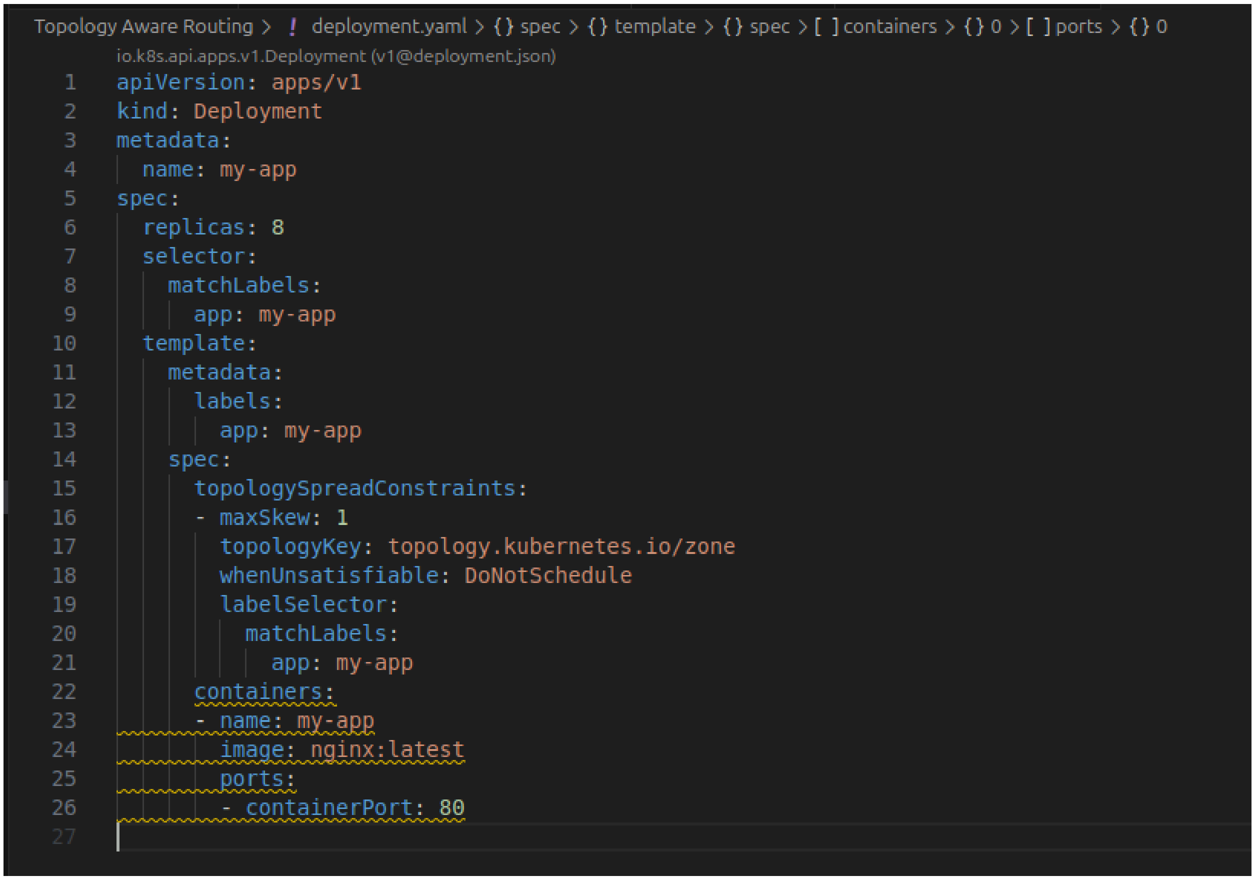
Look for labels like:

```
topology.kubernetes.io/zone
topology.kubernetes.io/region
```

Step 1: Create Deployment

```yaml
 apiVersion: apps/v1

kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 8
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      topologySpreadConstraints:
      - maxSkew: 1
        topologyKey: topology.kubernetes.io/zone
        whenUnsatisfiable: DoNotSchedule
        labelSelector:
          matchLabels:
            app: my-app
      containers:
      - name: my-app
        image: nginx:latest
        ports:
        - containerPort: 80
```


Topology Aware Routing > ! deployment.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] ports > {} 0
io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 8
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      topologySpreadConstraints:
      - maxSkew: 1
        topologyKey: topology.kubernetes.io/zone
        whenUnsatisfiable: DoNotSchedule
        labelSelector:
          matchLabels:
            app: my-app
      containers:
      - name: my-app
        image: nginx:latest
        ports:
        - containerPort: 80

```

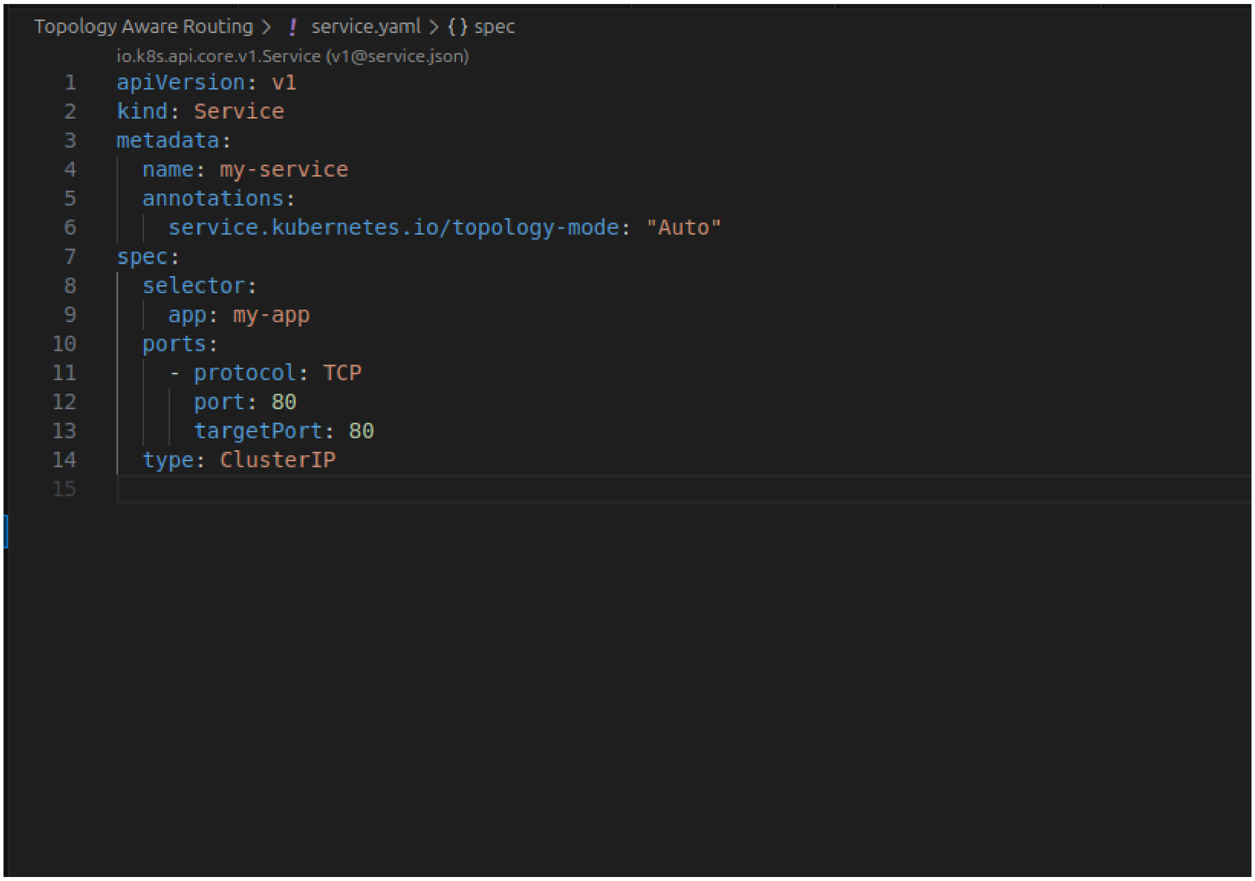
Apply deployment:

```
kubectl apply -f my-app-deployment.yaml
```

Step 2: Create Service with Topology Aware Routing

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
  annotations:
    service.kubernetes.io/topology-mode: Auto
spec:
  selector:
    app: my-app
  ports:
  - port: 80
```

```
    targetPort: 80
  type: ClusterIP
```


Topology Aware Routing > ! service.yaml > { } spec
io.k8s.api.core.v1.Service (v1@service.json)

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
  annotations:
    service.kubernetes.io/topology-mode: "Auto"
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: ClusterIP
```

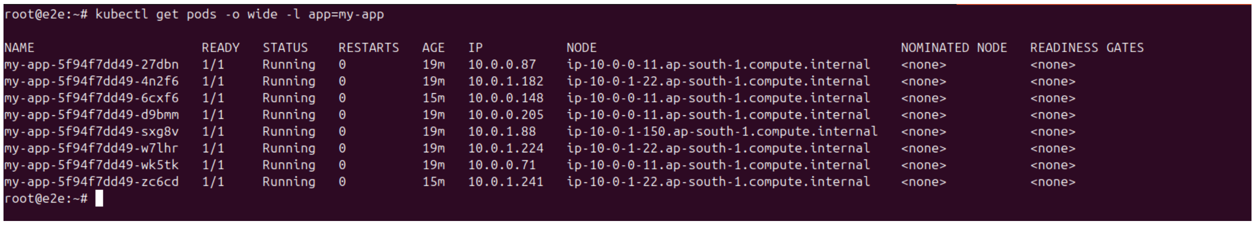
Apply service:

```
kubectl apply -f my-app-service.yaml
```

Step 3: Verify Pod Placement

Check pod distribution:

```
kubectl get pods -o wide -l app=my-app
```



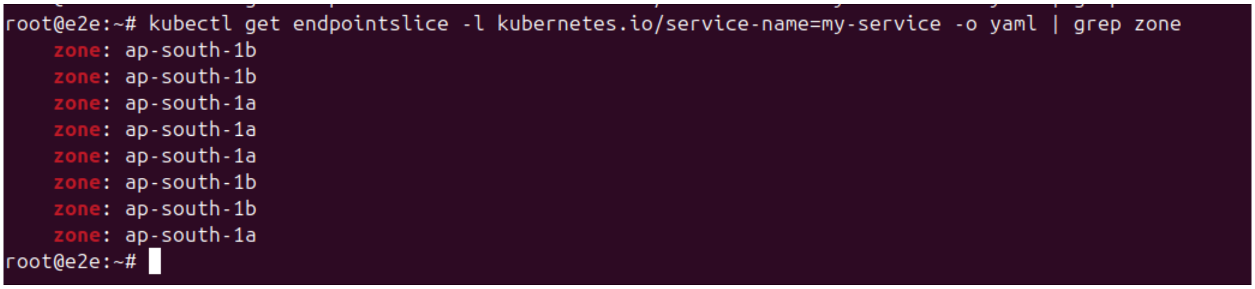
```
root@e2e:~# kubectl get pods -o wide -l app=my-app

NAME                       READY   STATUS    RESTARTS   AGE   IP           NODE                                         NOMINATED NODE   READINESS GATES
my-app-5f94f7dd49-27dbn    1/1     Running   0          19m   10.0.0.87    ip-10-0-0-11.ap-south-1.compute.internal     <none>           <none>
my-app-5f94f7dd49-4n2f6    1/1     Running   0          19m   10.0.1.182   ip-10-0-1-22.ap-south-1.compute.internal     <none>           <none>
my-app-5f94f7dd49-6cxf6    1/1     Running   0          15m   10.0.0.148   ip-10-0-0-11.ap-south-1.compute.internal     <none>           <none>
my-app-5f94f7dd49-d9bmm    1/1     Running   0          19m   10.0.0.205   ip-10-0-0-11.ap-south-1.compute.internal     <none>           <none>
my-app-5f94f7dd49-sxg8v    1/1     Running   0          19m   10.0.1.88    ip-10-0-1-150.ap-south-1.compute.internal    <none>           <none>
my-app-5f94f7dd49-w7lhr    1/1     Running   0          19m   10.0.1.224   ip-10-0-1-22.ap-south-1.compute.internal     <none>           <none>
my-app-5f94f7dd49-wk5tk    1/1     Running   0          19m   10.0.0.71    ip-10-0-0-11.ap-south-1.compute.internal     <none>           <none>
my-app-5f94f7dd49-zc6cd    1/1     Running   0          15m   10.0.1.241   ip-10-0-1-22.ap-south-1.compute.internal     <none>           <none>
root@e2e:~#
```

- Pods should be **spread across zones**.
- Check **which pod is on which node/zone**.

Step 4: Inspect EndpointSlices

kubectl get endpointslice -l kubernetes.io/service-name=my-service -o yaml



```
root@e2e:~# kubectl get endpointslice -l kubernetes.io/service-name=my-service -o yaml | grep zone
    zone: ap-south-1b
    zone: ap-south-1b
    zone: ap-south-1a
    zone: ap-south-1a
    zone: ap-south-1a
    zone: ap-south-1b
    zone: ap-south-1b
    zone: ap-south-1a
root@e2e:~# 
```

Look for hints like:

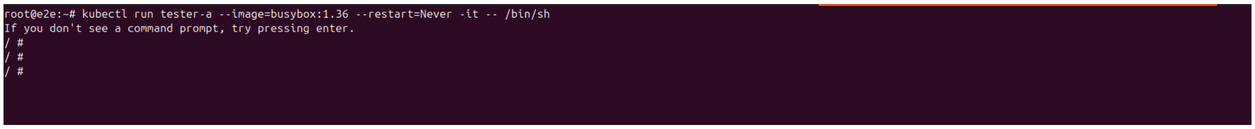
```
hints:
  forZones:
    - name: ap-south-1a
```

This confirms **kube-proxy prefers same-zone pods**.

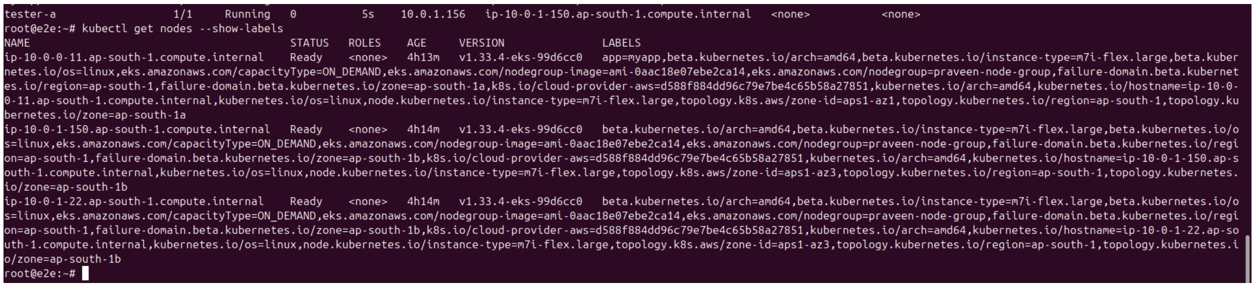
Step 5: Test Traffic Routing

5.1 Create test pods in each zone

kubectl run tester-a --image=busybox:1.36 --restart=Never -it -- /bin/sh



```
root@e2e:~# kubectl run tester-a --image=busybox:1.36 --restart=Never -it -- /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ #
```

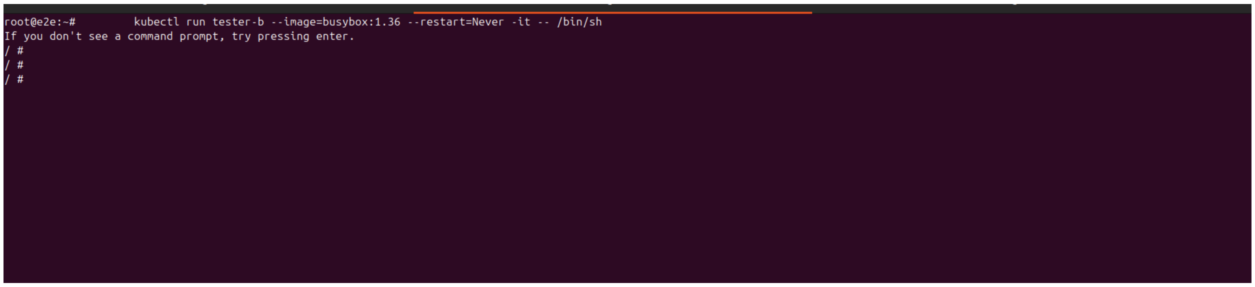


```
tester-a                  1/1     Running   0          5s    10.0.1.156   ip-10-0-1-150.ap-south-1.compute.internal   <none>           <none>
root@e2e:~# kubectl get nodes --show-labels
NAME                                        STATUS   ROLES    AGE     VERSION               LABELS
ip-10-0-0-11.ap-south-1.compute.internal    Ready    <none>   4h13m   v1.33.4-eks-99d6cc0   app=myapp,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kuber
netes.io/os=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernet
es.io/region=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1a,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-
0-11.ap-south-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az1,topology.kubernetes.io/region=ap-south-1,topology.ku
bernetes.io/zone=ap-south-1a
ip-10-0-1-150.ap-south-1.compute.internal   Ready    <none>   4h14m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/o
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/regi
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-150.ap-s
outh-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes.
io/zone=ap-south-1b
ip-10-0-1-22.ap-south-1.compute.internal    Ready    <none>   4h14m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/o
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/regi
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-22.ap-so
uth-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes.i
o/zone=ap-south-1b
root@e2e:~# 
```

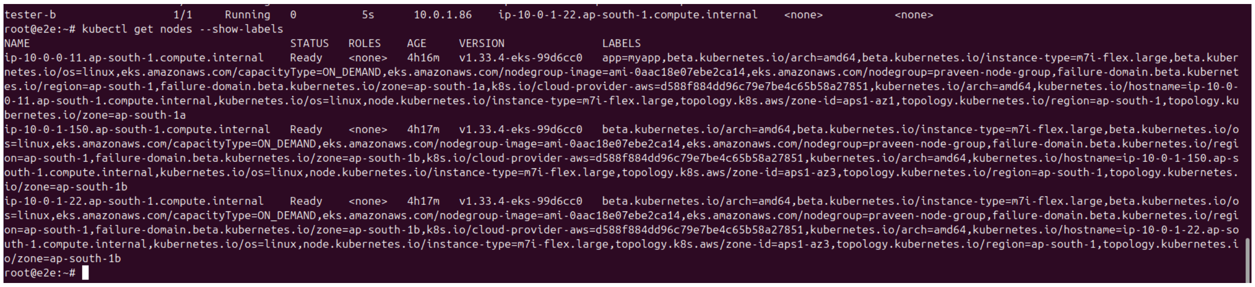
It was deployed on the node

ip-10-0-1-150.ap-south-1.compute.internal

kubectl run tester-b --image=busybox:1.36 --restart=Never -it -- /bin/sh



```
root@e2e:~#         kubectl run tester-b --image=busybox:1.36 --restart=Never -it -- /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ #
```



```
tester-b                 1/1     Running   0          5s      10.0.1.86    ip-10-0-1-22.ap-south-1.compute.internal    <none>           <none>
root@e2e:~# kubectl get nodes --show-labels
NAME                                         STATUS   ROLES    AGE     VERSION               LABELS
ip-10-0-0-11.ap-south-1.compute.internal    Ready    <none>   4h16m   v1.33.4-eks-99d6cc0   app=myapp,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kuber
netes.io/os=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernet
es.io/region=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1a,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-
0-11.ap-south-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az1,topology.kubernetes.io/region=ap-south-1,topology.ku
bernetes.io/zone=ap-south-1a
ip-10-0-1-150.ap-south-1.compute.internal   Ready    <none>   4h17m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/o
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/regi
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-150.ap-s
outh-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes.
io/zone=ap-south-1b
ip-10-0-1-22.ap-south-1.compute.internal    Ready    <none>   4h17m   v1.33.4-eks-99d6cc0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=m7i-flex.large,beta.kubernetes.io/o
s=linux,eks.amazonaws.com/capacityType=ON_DEMAND,eks.amazonaws.com/nodegroup-image=ami-0aac18e07ebe2ca14,eks.amazonaws.com/nodegroup=praveen-node-group,failure-domain.beta.kubernetes.io/regi
on=ap-south-1,failure-domain.beta.kubernetes.io/zone=ap-south-1b,k8s.io/cloud-provider-aws=d588f884dd96c79e7be4c65b58a27851,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-10-0-1-22.ap-so
uth-1.compute.internal,kubernetes.io/os=linux,node.kubernetes.io/instance-type=m7i-flex.large,topology.k8s.aws/zone-id=aps1-az3,topology.kubernetes.io/region=ap-south-1,topology.kubernetes.i
o/zone=ap-south-1b
root@e2e:~#
```

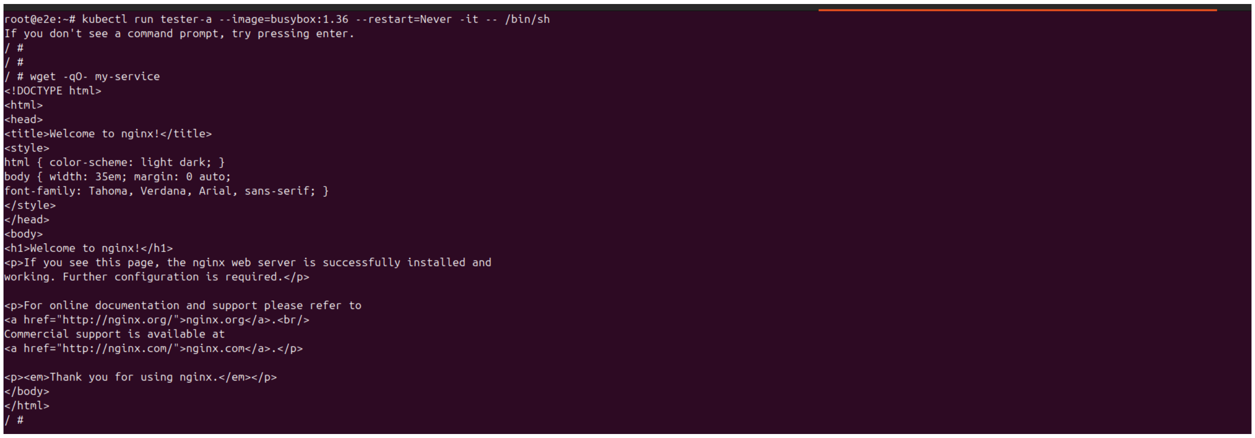
It was deployed on the node:

ip-10-0-1-22.ap-south-1.compute.internal

5.2 Send requests

Inside tester-a (zone a):

wget -qO- my-service



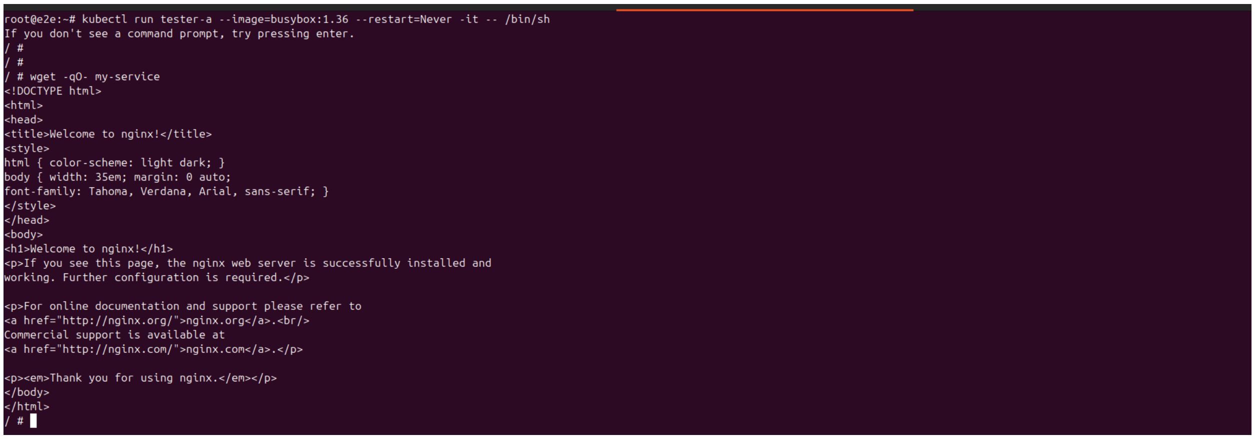
```
root@e2e:~# kubectl run tester-a --image=busybox:1.36 --restart=Never -it -- /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ # wget -qO- my-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; 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>
/ #
```

Inside tester-b (zone b):

wget -qO- my-service-svc



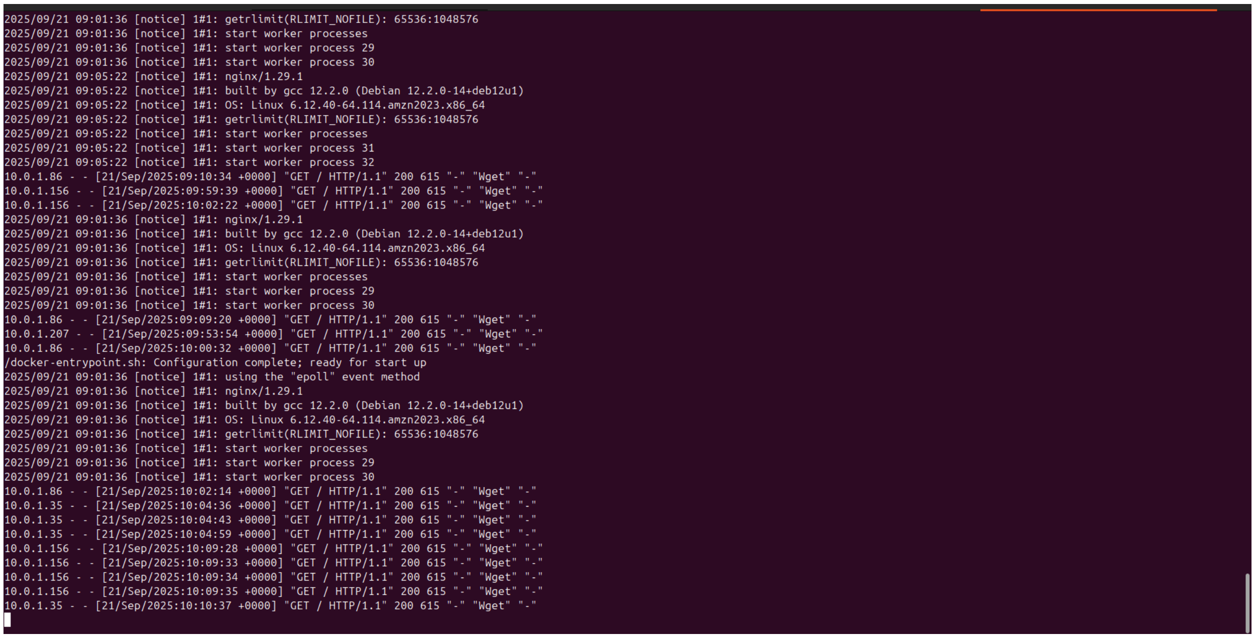
```
root@e2e:~# kubectl run tester-a --image=busybox:1.36 --restart=Never -it -- /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ # wget -qO- my-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; 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>
/ #
```

Check which pod served the requests

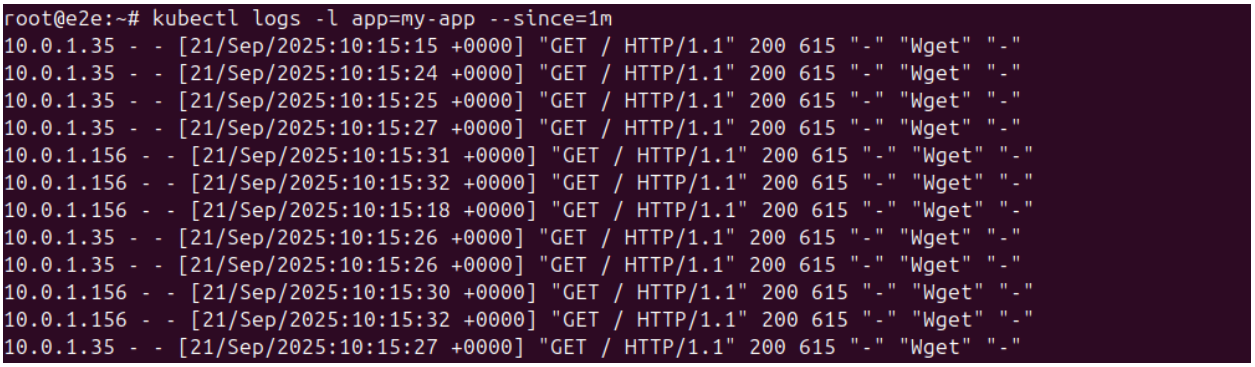
kubectl logs -l app=my-app --max-log-requests=10



```
2025/09/21 09:01:36 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 65536:1048576
2025/09/21 09:01:36 [notice] 1#1: start worker processes
2025/09/21 09:01:36 [notice] 1#1: start worker process 29
2025/09/21 09:01:36 [notice] 1#1: start worker process 30
2025/09/21 09:05:22 [notice] 1#1: nginx/1.29.1
2025/09/21 09:05:22 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/09/21 09:05:22 [notice] 1#1: OS: Linux 6.12.40-64.114.amzn2023.x86_64
2025/09/21 09:05:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 65536:1048576
2025/09/21 09:05:22 [notice] 1#1: start worker processes
2025/09/21 09:05:22 [notice] 1#1: start worker process 31
2025/09/21 09:05:22 [notice] 1#1: start worker process 32
10.0.1.86 - - [21/Sep/2025:09:10:34 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:09:59:39 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:02:22 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
2025/09/21 09:01:36 [notice] 1#1: nginx/1.29.1
2025/09/21 09:01:36 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/09/21 09:01:36 [notice] 1#1: OS: Linux 6.12.40-64.114.amzn2023.x86_64
2025/09/21 09:01:36 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 65536:1048576
2025/09/21 09:01:36 [notice] 1#1: start worker processes
2025/09/21 09:01:36 [notice] 1#1: start worker process 29
2025/09/21 09:01:36 [notice] 1#1: start worker process 30
10.0.1.86 - - [21/Sep/2025:09:09:20 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.207 - - [21/Sep/2025:09:53:54 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.86 - - [21/Sep/2025:10:00:32 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/09/21 09:01:36 [notice] 1#1: using the "epoll" event method
2025/09/21 09:01:36 [notice] 1#1: nginx/1.29.1
2025/09/21 09:01:36 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/09/21 09:01:36 [notice] 1#1: OS: Linux 6.12.40-64.114.amzn2023.x86_64
2025/09/21 09:01:36 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 65536:1048576
2025/09/21 09:01:36 [notice] 1#1: start worker processes
2025/09/21 09:01:36 [notice] 1#1: start worker process 29
2025/09/21 09:01:36 [notice] 1#1: start worker process 30
10.0.1.86 - - [21/Sep/2025:10:02:14 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:04:36 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:04:43 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:04:59 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:09:28 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:09:33 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:09:34 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:09:35 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:10:37 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
```

Expected:

- Traffic from a pod in zone A hits pods in **zone A**.
- Traffic from a pod in zone B hits pods in **zone B**.



```
root@e2e:~# kubectl logs -l app=my-app --since=1m
10.0.1.35 - - [21/Sep/2025:10:15:15 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:15:24 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:15:25 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:15:27 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:15:31 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:15:32 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:15:18 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:15:26 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:15:26 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:15:30 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.156 - - [21/Sep/2025:10:15:32 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.0.1.35 - - [21/Sep/2025:10:15:27 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
```

Result

- Pods are **spread across zones**.

- Service routes traffic **preferentially to same-zone pods**.
- Logs and EndpointSlices confirm **topology-aware routing** is active.