Life of a Packet in Kubernetes



A Network Engineer's Guide

Nico Vibert
Technical Marketing Engineering Director
Isovalent at Cisco

BRKCLD-3734

Webex App

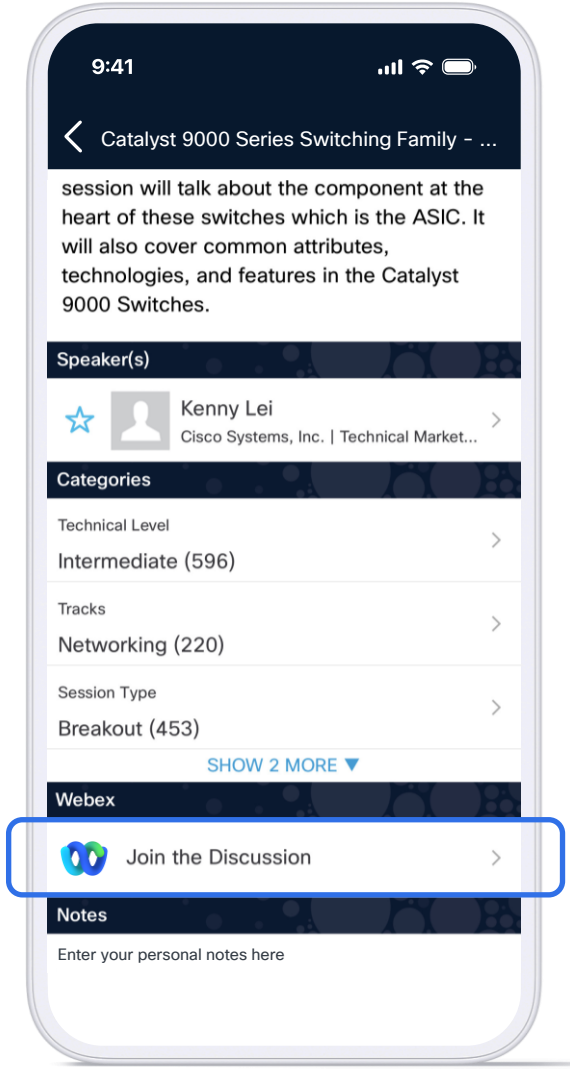
Questions?

Use Webex App to chat with the speaker after the session

How

1. Find this session in the Cisco Events App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 27, 2026.



© 2026 Cisco and/or its affiliates. All rights reserved.

Agenda

01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

07 Traffic Engineering

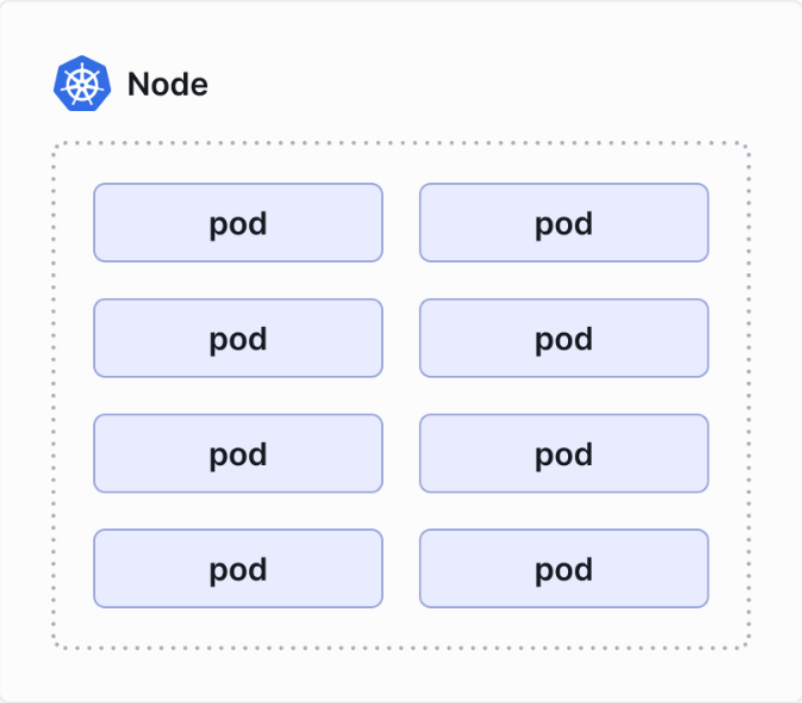
08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

07 Traffic Engineering

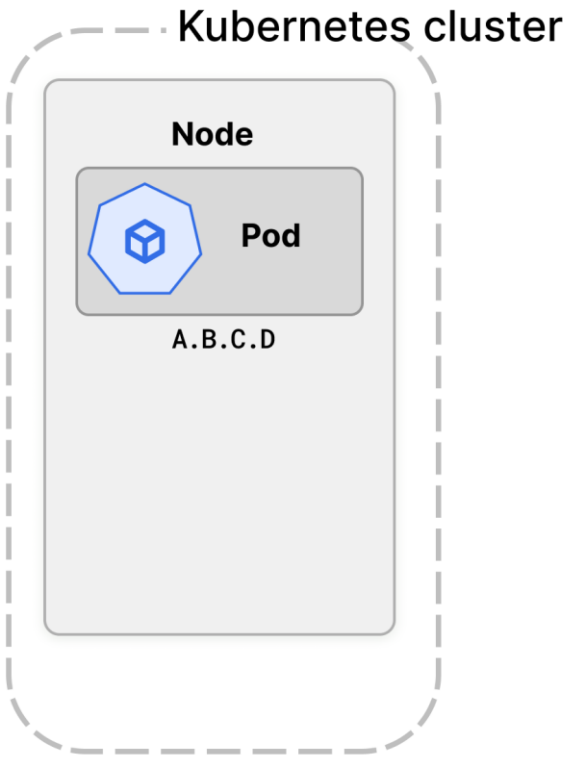
08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

07 Traffic Engineering

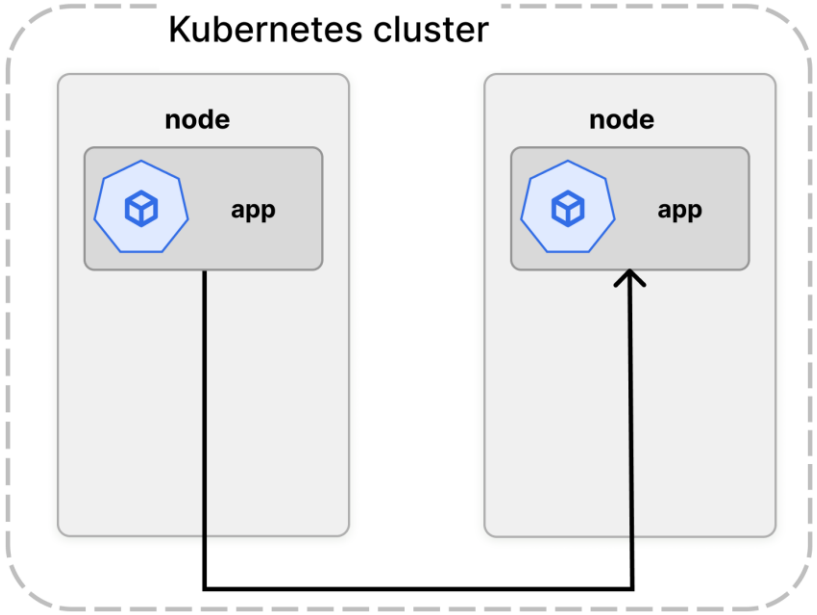
08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

07 Traffic Engineering

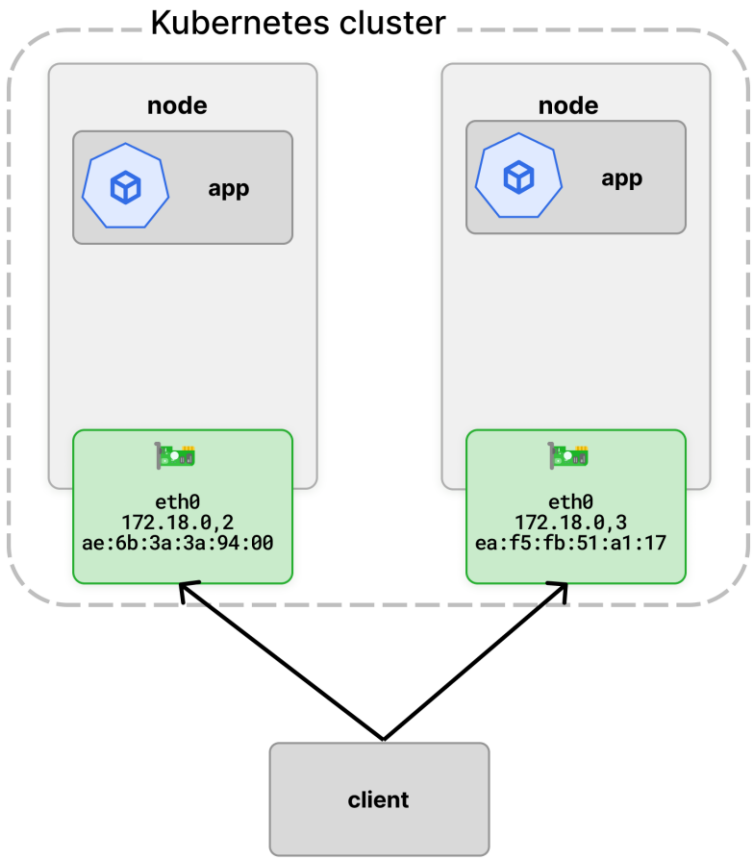
08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

07 Traffic Engineering

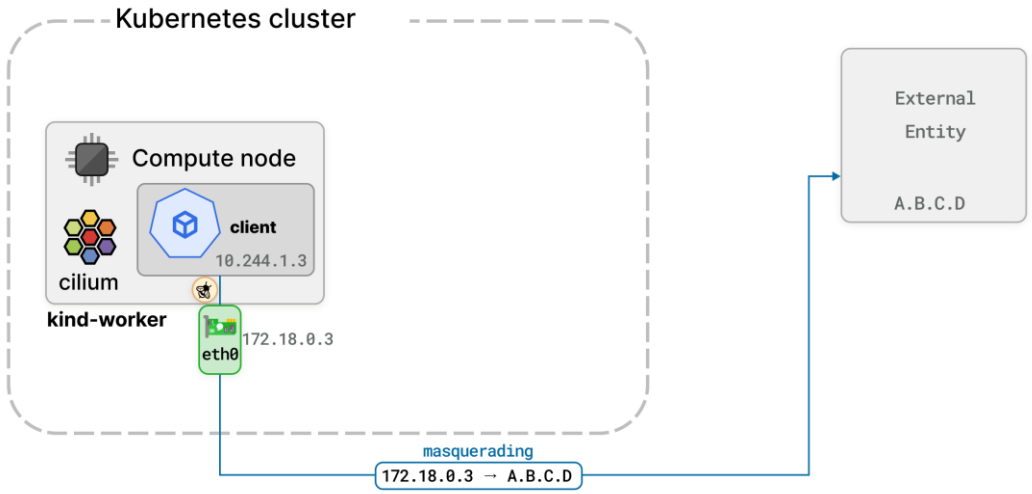
08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

07 Traffic Engineering

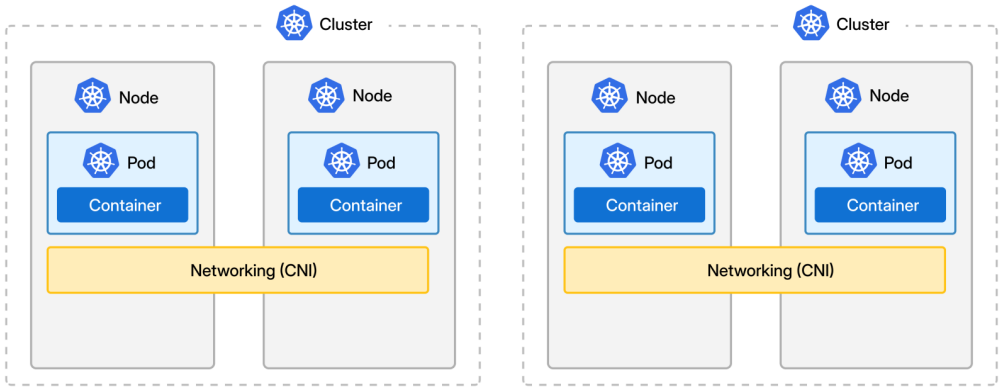
08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

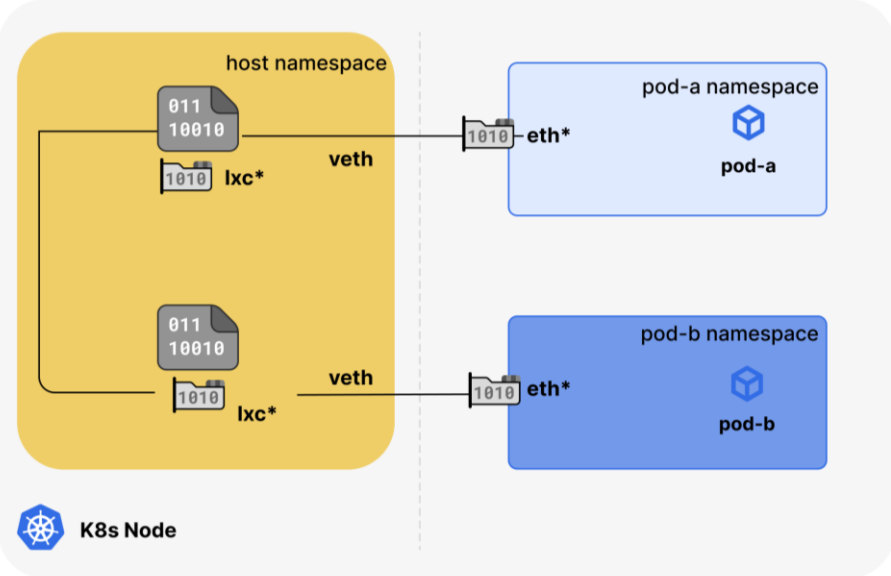
07 Traffic Engineering

08 Multi-Clustering

09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

Agenda



01 Kubernetes Networking 101

02 IP address management

03 Inter-node connectivity

04 Kubernetes Services

05 Cluster Ingress

06 Cluster Egress

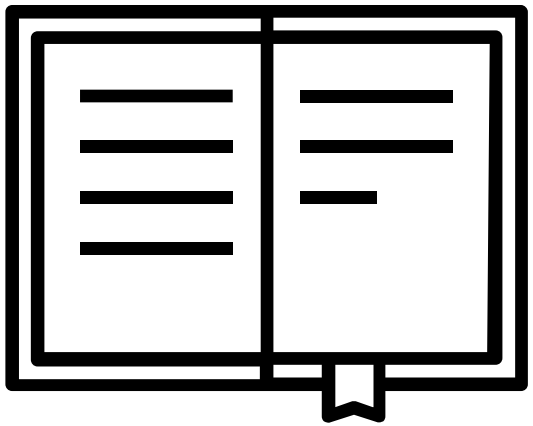
07 Traffic Engineering

08 Multi-Clustering

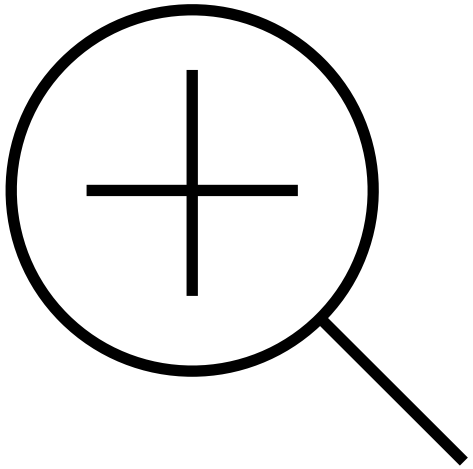
09 Intra-node connectivity

© 2026 Cisco and/or its affiliates. All rights reserved.

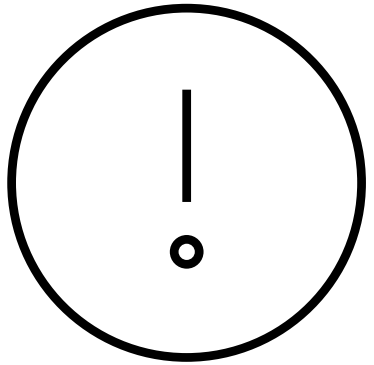
Keys



Reference Slide



Inside Packets

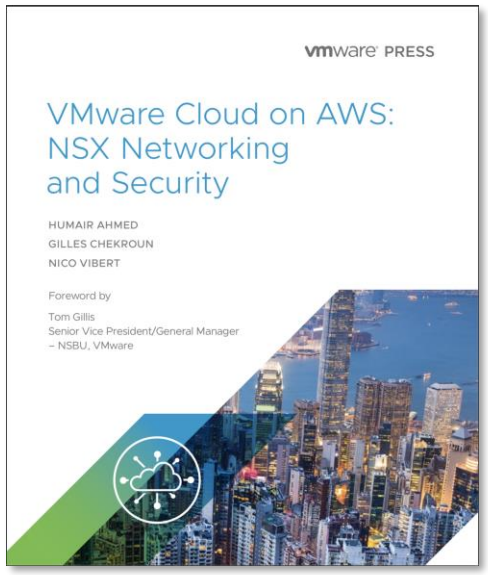


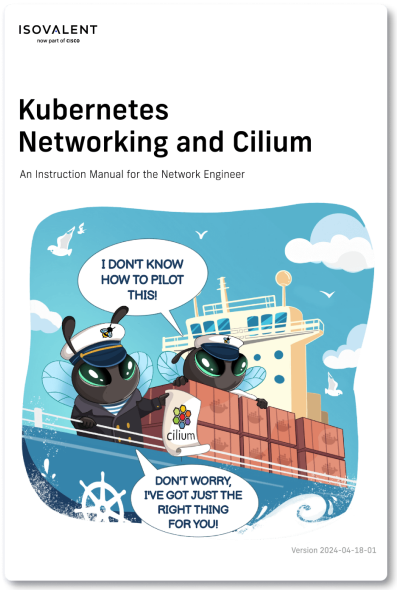
Warning

© 2026 Cisco and/or its affiliates. All rights reserved.

Hello, I'm Nico 👋🏾

- Technical Marketing Engineering Director at Isovalent
- Formerly at VMware, HashiCorp and...Cisco
- CCIE #22990 (expired...)
- Book(s) author







© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes Networking eBook

https://cs.co/k8s-for-network-engineers

- Written by a network engineer for network engineers
- Instruction manual focused on Kubernetes Networking and the de facto Kubernetes networking layer, Cilium.
- Over 10,000 downloads already!

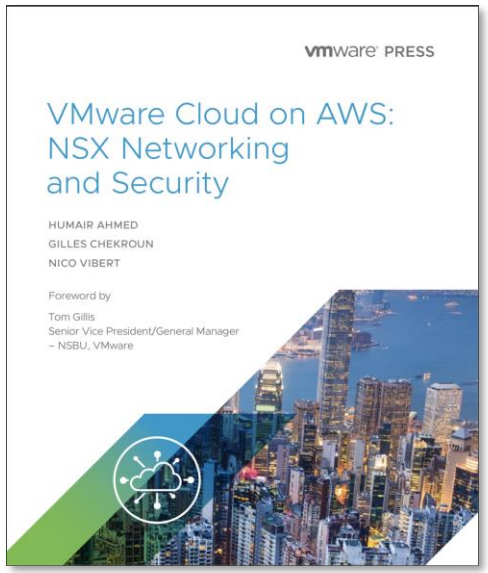


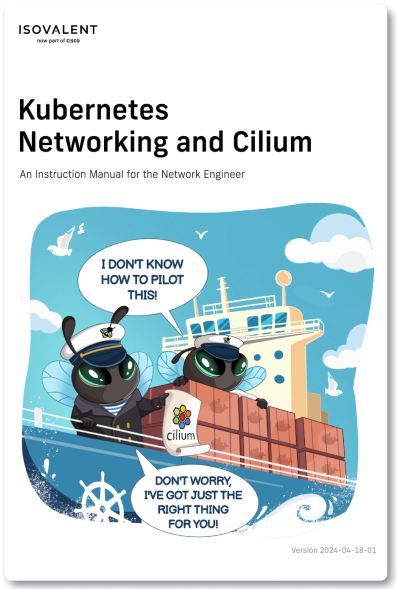
© 2026 Cisco and/or its affiliates. All rights reserved.

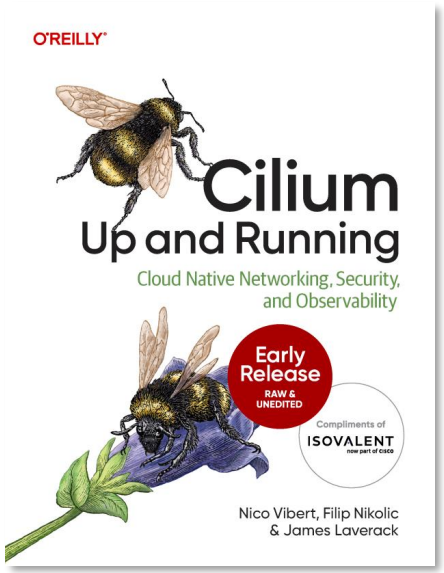
Hello, I'm Nico 👋

- Technical Marketing Engineering Director at Isovalent
- Formerly at VMware, HashiCorp and...Cisco
- CCIE #22990 (expired...)
- Book(s) author









© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium Up and Running



- Coming out next week!
- Latest O'Reilly book in the "Up and Running" series
- ~375 pages
- Covers all topics discussed in this presentation, in more depth, with detailed tutorials.
- Comes with an online lab for you try each feature in your time - at no cost.
- **Digital version available for free on Isovalent.com at release.**
- Physical edition available for free at KubeCon during the book signing!

Best Sellers Rank: #1,521,605 in Books (See Top 100 in Books)
#237 in Computer Networks
#992 in Cloud Computing (Books)
#1,125 in Computer Network Security

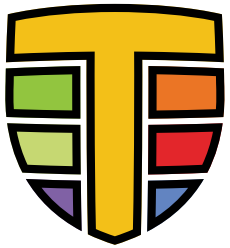
© 2026 Cisco and/or its affiliates. All rights reserved.

Isovalent & Cilium Recap





cilium



tetragon

- Open Source Projects
- Cilium and Tetragon use eBPF for networking and security
- Cilium = SDN for Kubernetes

ISOVALENT

- Leading Company behind Cilium, Tetragon and eBPF
- Provides Enterprise distributions of Cilium and Tetragon
- Cisco acquired Isovalent in 2024.

© 2026 Cisco and/or its affiliates. All rights reserved.

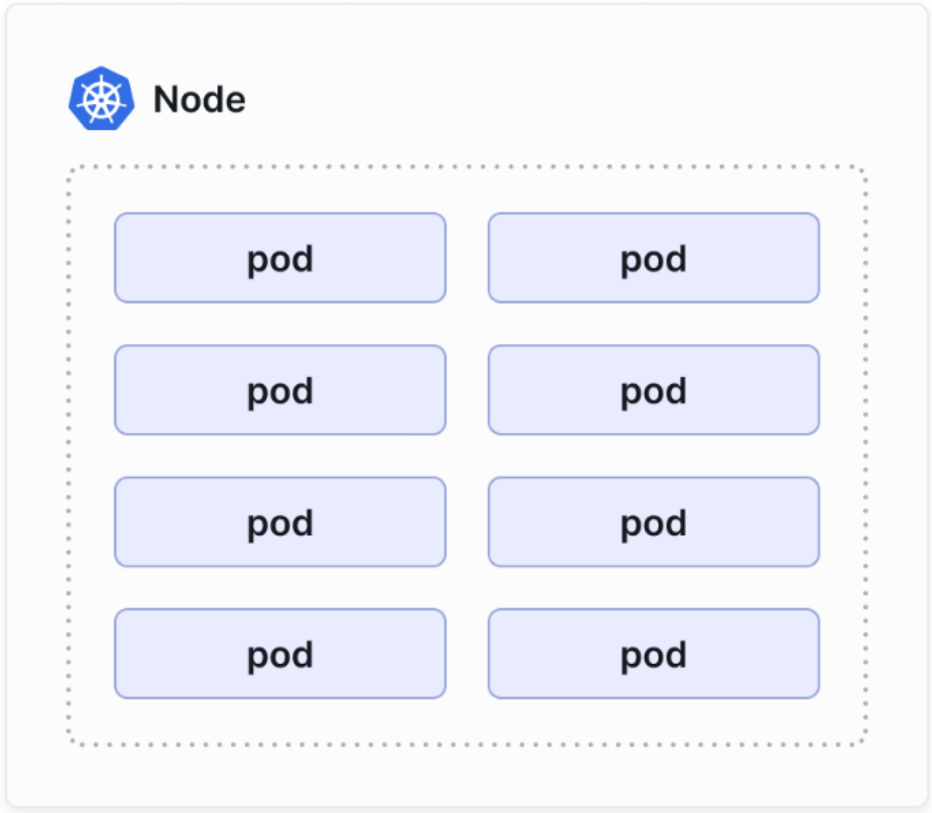
cisco

Kubernetes Networking Essentials

CISCO Live !

Kubernetes Networking Model

- Kubernetes is a container orchestration platform.
- A pod consists of one or more containers.
- A pod can represent an entire application, a single replica of a distributed application, or an individual service in a micro-service architecture.
- Kubernetes places pods onto nodes.



© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes Networking Model

- A group of nodes makes up a Kubernetes cluster.
- There are multiple models of cluster ownership:

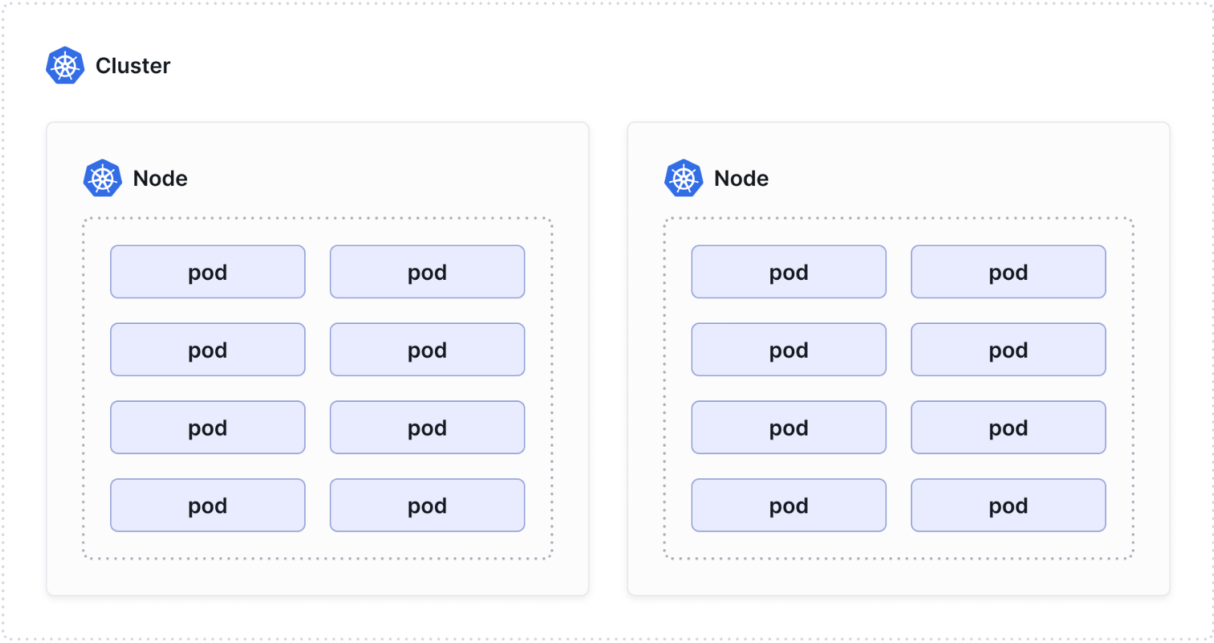
  Single shared cluster
  One cluster, many teams, isolated with namespaces and policies

  Cluster per environment
  Separate clusters for dev, staging, and production

  Cluster per team or workload
  Clusters aligned to teams or workload characteristics

  Cluster per tenant or customer
  Dedicated cluster per customer for maximum isolation

  Multi cluster platform
  Fleet of clusters operated as a single platform

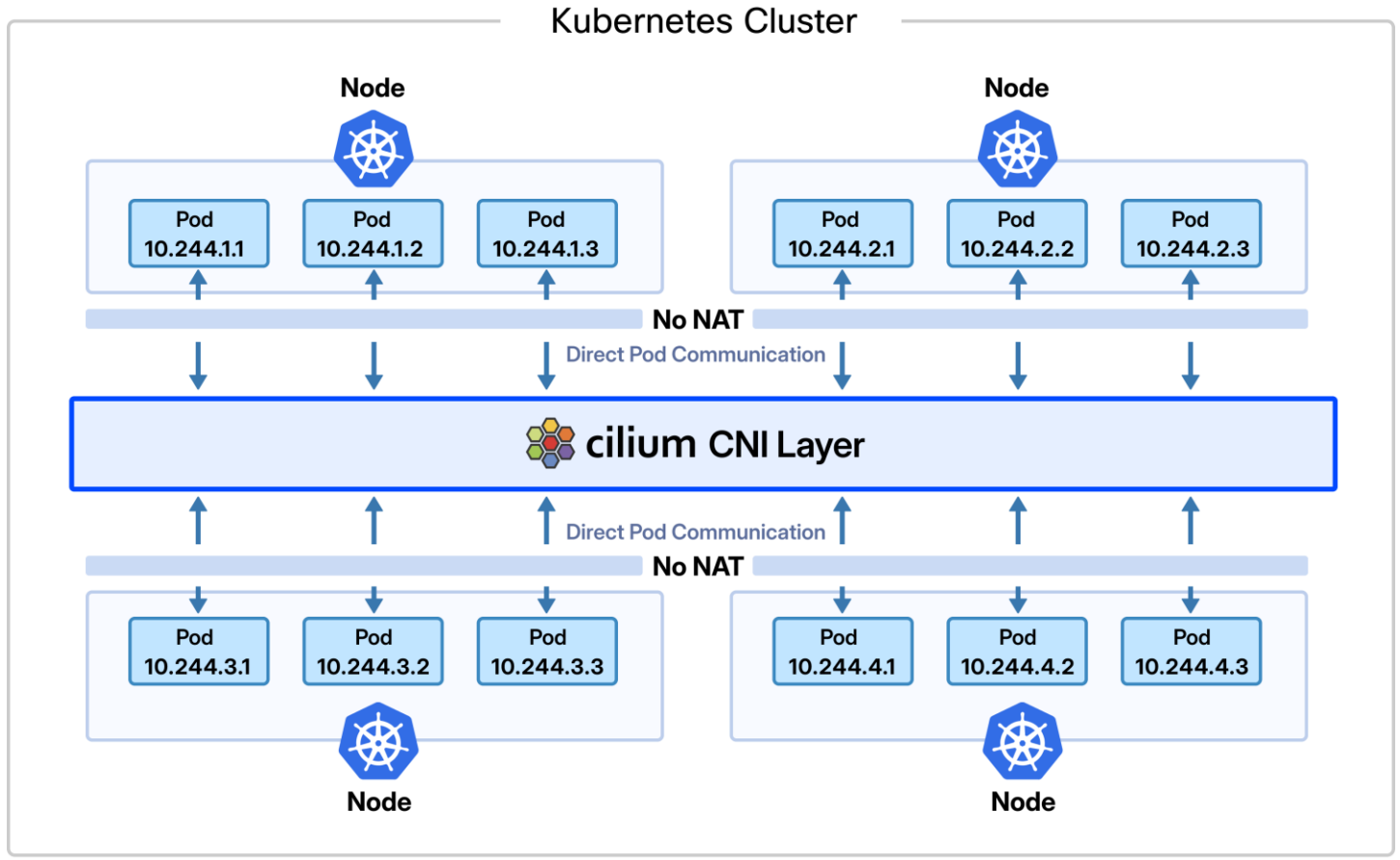


© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes Networking Model

The networking model in Kubernetes is designed to keep network infrastructure simple and predictable:

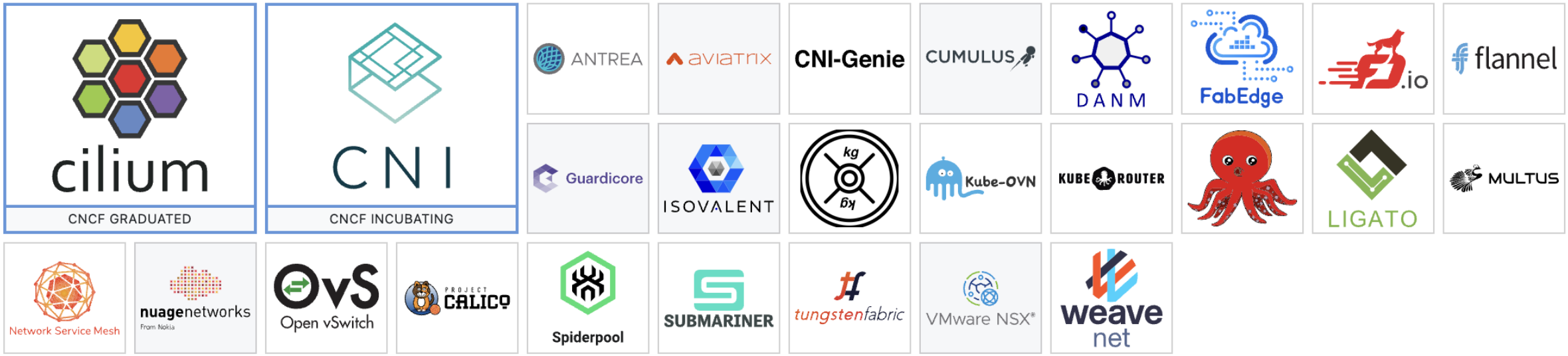
- Every pod in a cluster gets its unique cluster-wide IP address and can communicate with all other pods on any other node without using NAT.
- Pod-to-pod connectivity and IP address allocation requires a **Container Network Interface** (CNI) plugin to implement this model.



© 2026 Cisco and/or its affiliates. All rights reserved.

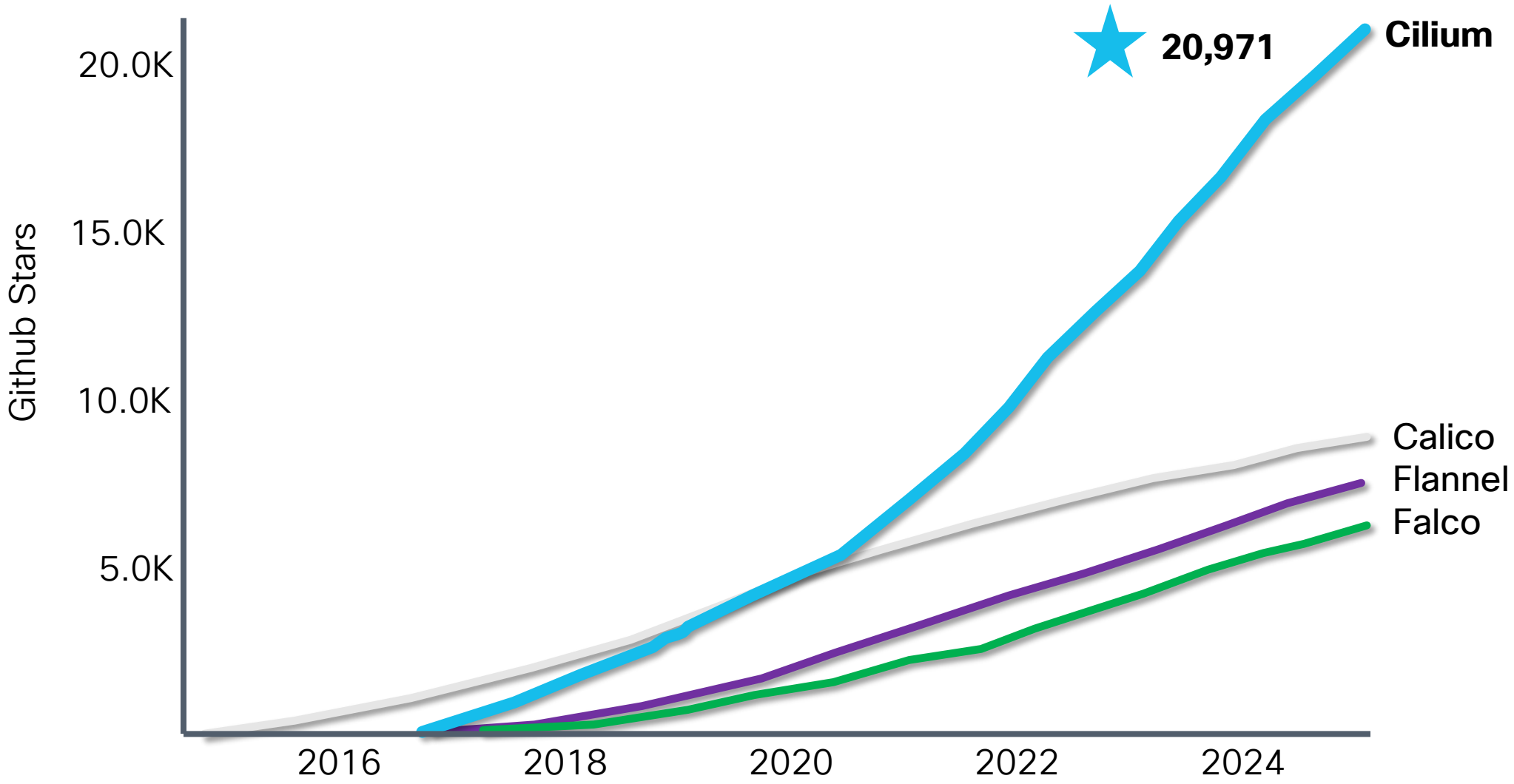
Kubernetes Networking Model

- Selecting a CNI for your Kubernetes deployment is like choosing a network vendor for a network refresh.
- It is one of the first crucial technical decisions in any Kubernetes project.
- Many options to choose from but one clear leader:



© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium: The Leading Kubernetes Networking & Security Project



© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Cilium: Chosen as Default By All Major Cloud Providers



2020 - Google announces Cilium & eBPF as the new networking data plane for GKE



2021 - AWS picks Cillium for Networking & Security on EKS Anywhere



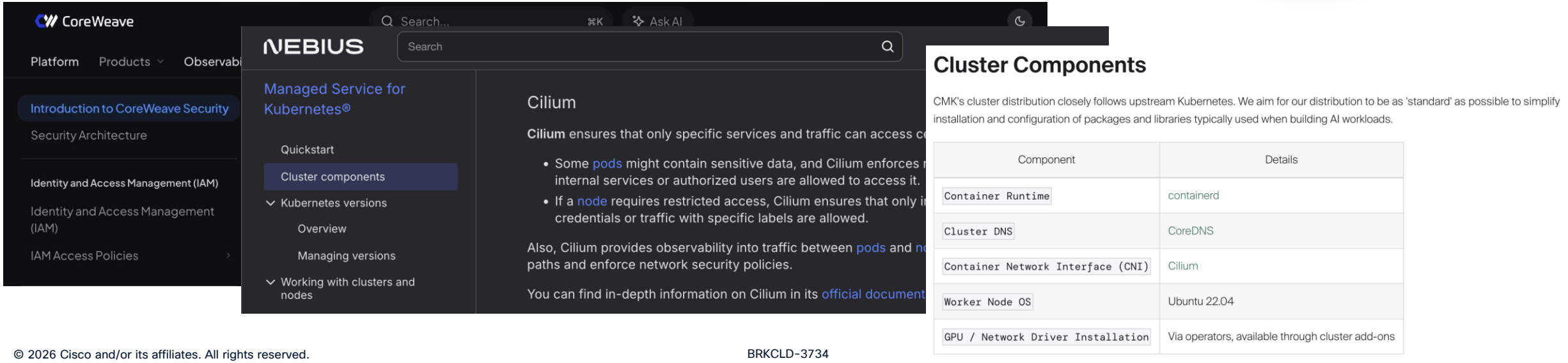
2022 - Announcing Azure CNI Powered by Cillium

2023 - Cloud Native Computing Foundation (CNCF) Announces Cilium Graduation



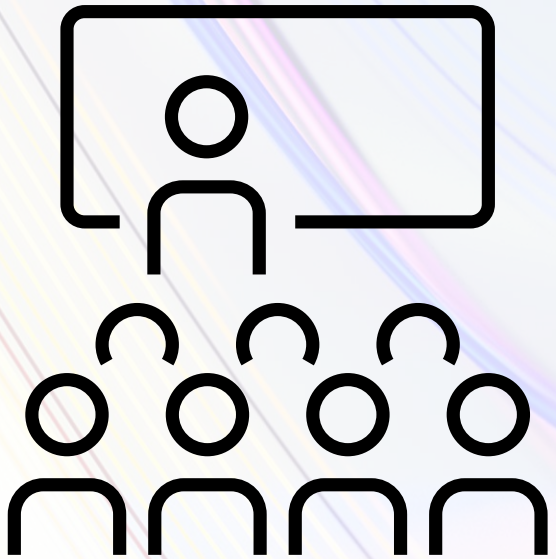
© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium: Chosen as Default By All Neo Cloud Providers

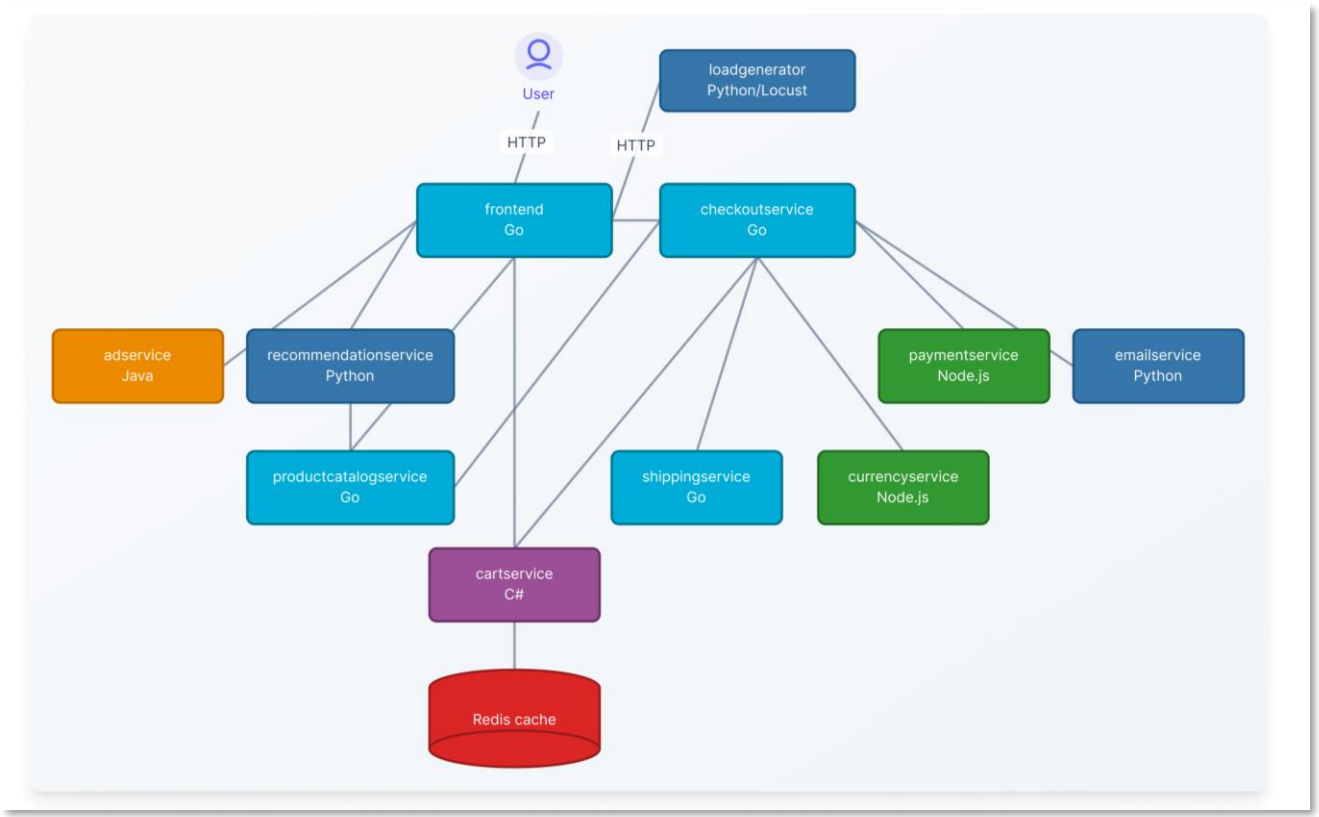


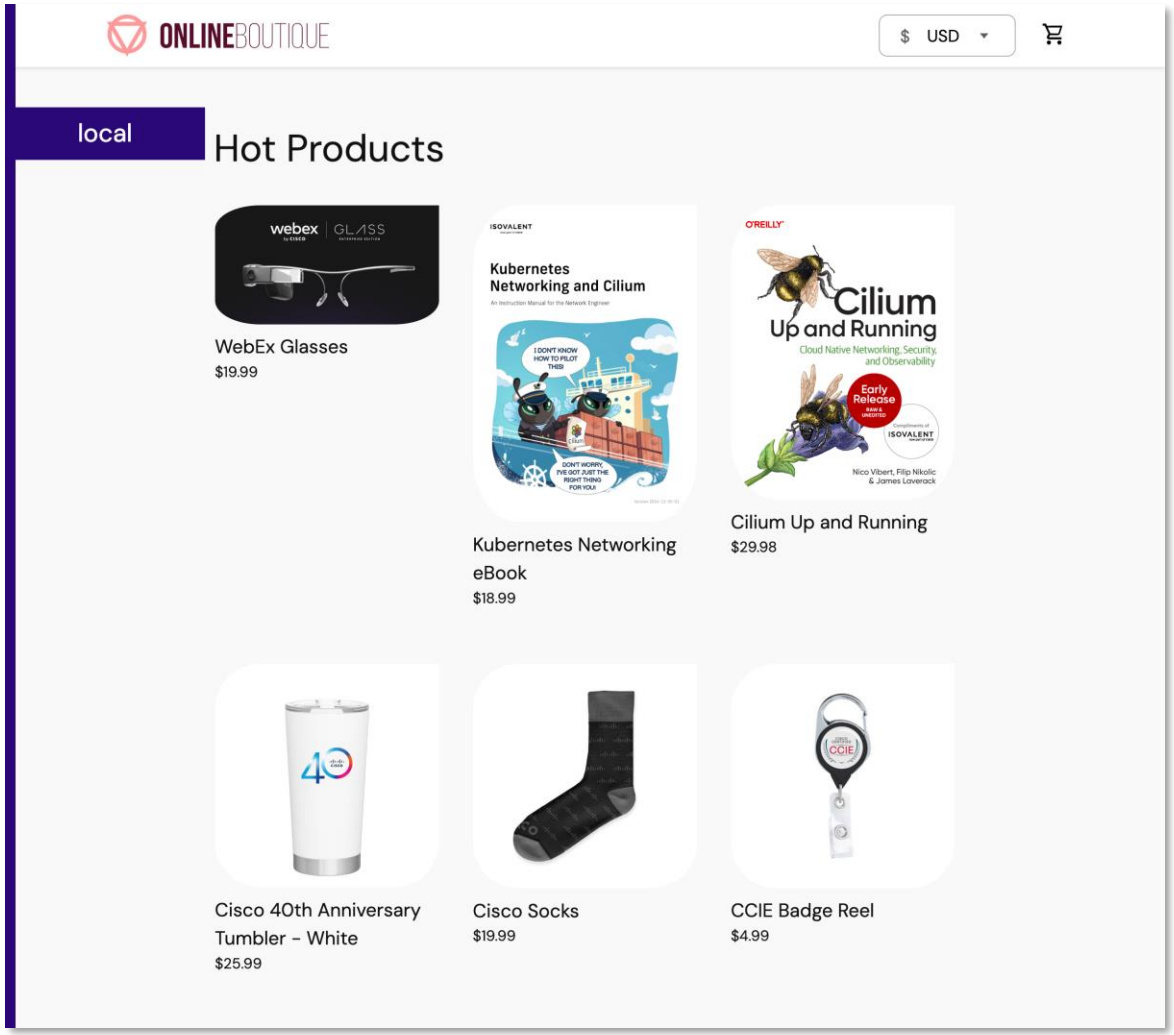
© 2026 Cisco and/or its affiliates. All rights reserved.

Demo



Micro Services Demo App





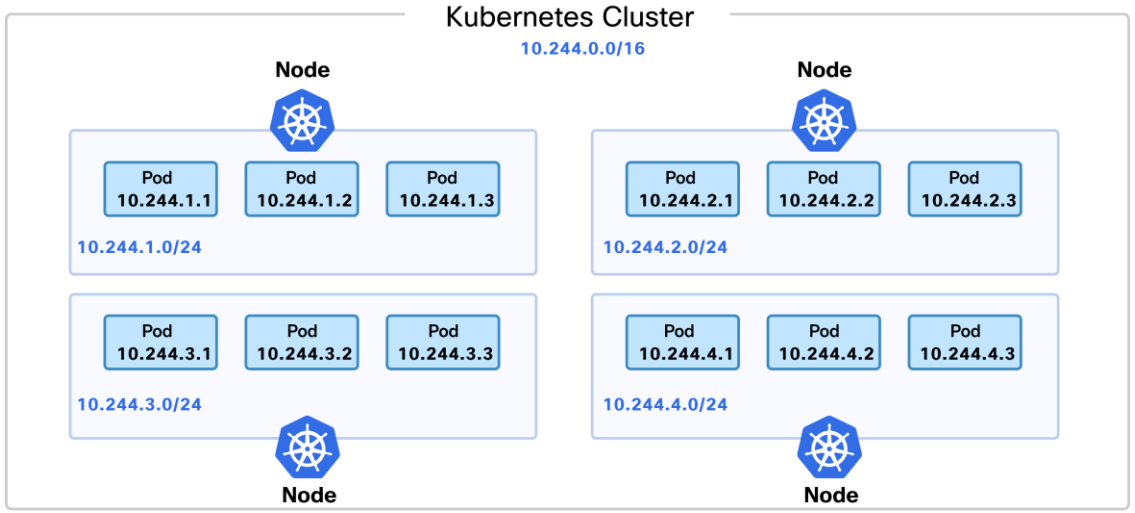
© 2026 Cisco and/or its affiliates. All rights reserved.

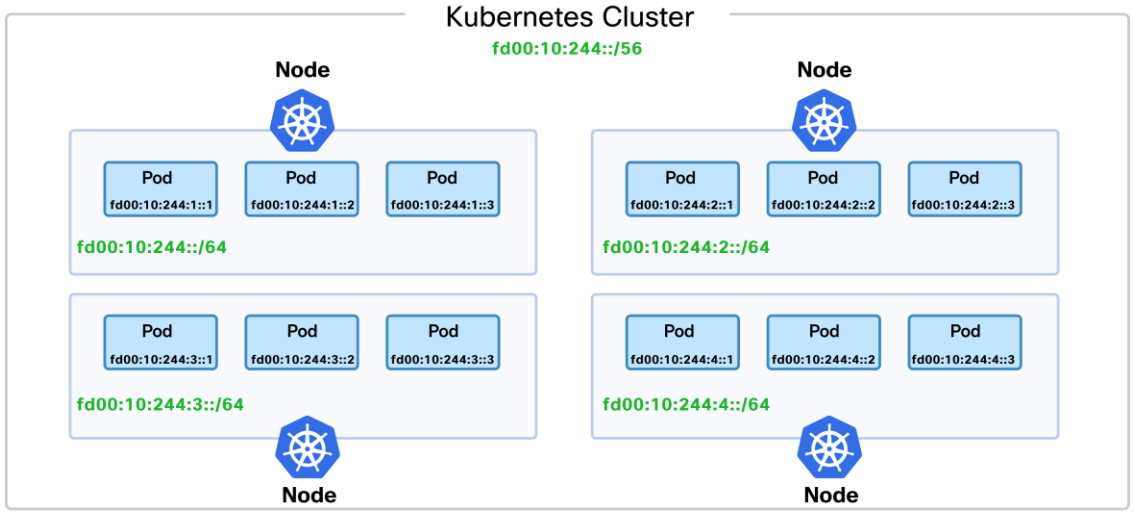
IPv6 or IPv4

IPv6 or IPv4?

Or Dual Stack?

- Kubernetes supports IPv4 single-stack, IPv6 single-stack, and dual-stack – depending on whether the CNI you choose to implement the required functionality.
- Cilium supports them all.
- There's near feature parity across IPv4/IPv6/Dual Stack.
- Sadly – the ecosystem isn't quite ready for IPv6-only meaning Dual Stack or IPv4 is often required.





© 2026 Cisco and/or its affiliates. All rights reserved.

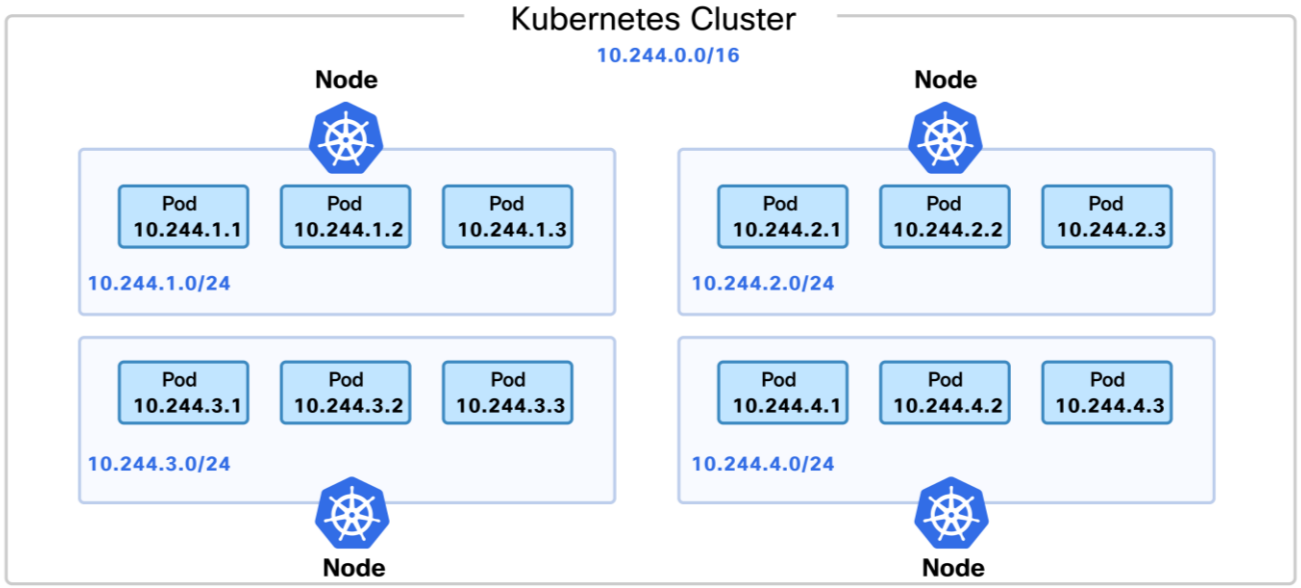
cisco

IP Address Management

IP Address Allocation

"Where's my DHCP server?"

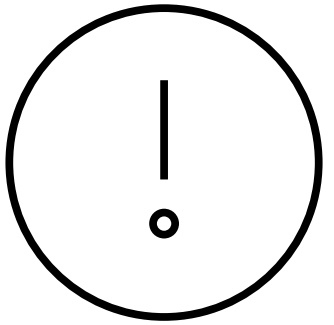
- **Nodes** receive their IP addresses from traditional DHCP or IPAM or via static addressing – just like any other server or VM on the network.
- For **pods**, the process is different.
- A cluster gets one or multiple CIDRs.
- Nodes receive one or multiple ranges (**PodCIDR**).
- Cilium will assign IPs from PodCIDR ranges as pods come online.
- *How* Cilium assigns IPs depends on the IPAM mode.
- More info on IPAM modes in Chapter 4 of Cilium Up and Running and in slides.



© 2026 Cisco and/or its affiliates. All rights reserved.

IP Addressing in Kubernetes

IP address != Identity



- IP addresses do not represent an identity in Kubernetes.
- Pods come and go. When pods are destroyed, they may be respawned with a different IP address.
- Network policies are to be defined based on business logic and metadata rather than IP addressing.

```
$ kubectl get pods -o wide -n ciscoboutique
NAME                                    READY   STATUS    RESTARTS   AGE     IP
adservice-764945b54b-dx9nf              1/1     Running   0          2m34s   10.244.1.140

$ kubectl delete -n ciscoboutique pod adservice-764945b54b-dx9nf
pod "adservice-764945b54b-dx9nf" deleted from ciscoboutique namespace

$ kubectl get pods -o wide -n ciscoboutique
NAME                                    READY   STATUS    RESTARTS   AGE     IP
adservice-764945b54b-sbl49              1/1     Running   0          5s      10.244.1.216
```

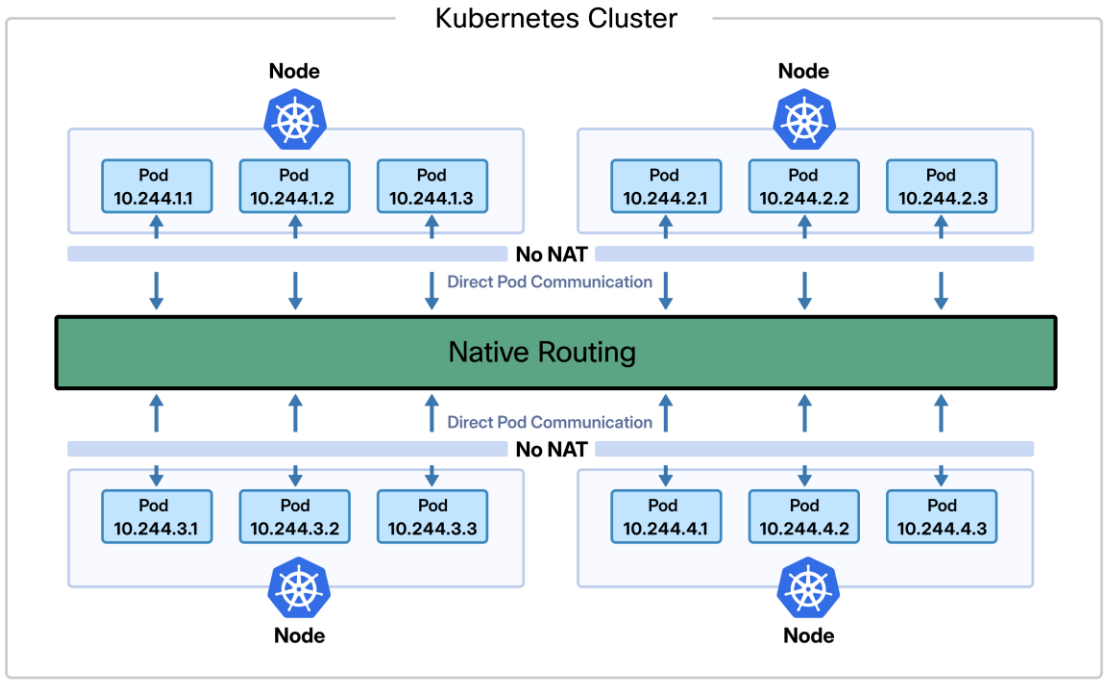
© 2026 Cisco and/or its affiliates. All rights reserved.

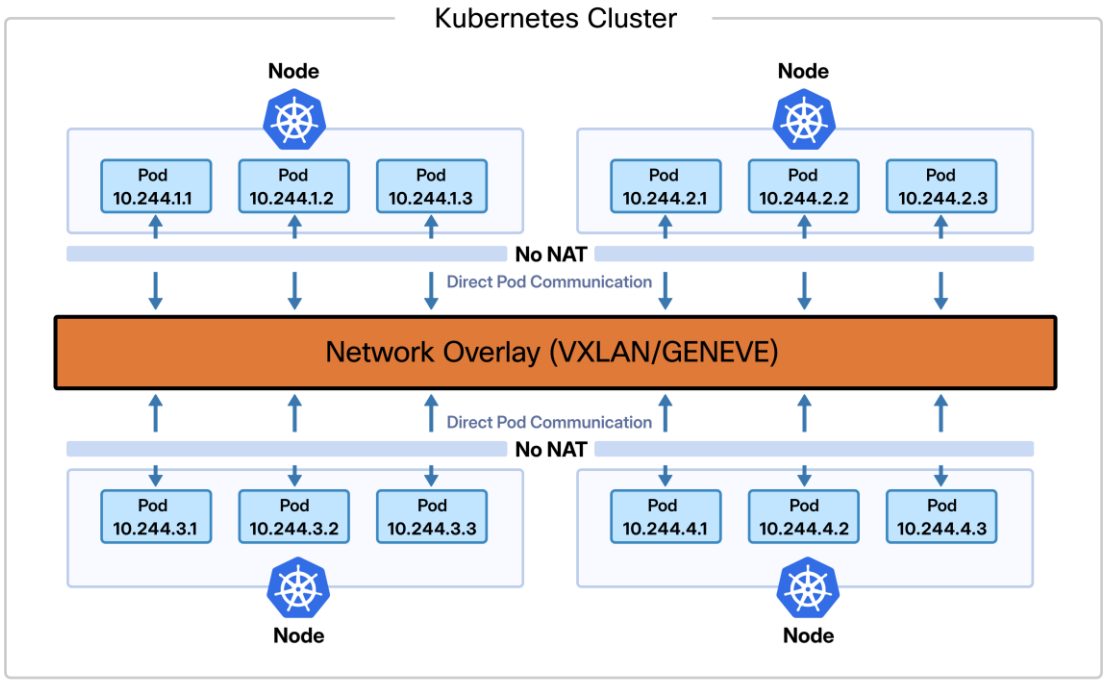
cisco

Inter-node connectivity

Inter-node Communication

- Kubernetes requires every pod to reach others directly without NAT, even if they are located on different nodes.
- Inter-node traffic crosses the underlying physical network so nodes must understand pod ranges.
- Cilium offers **native routing** or **encapsulation** to maintain full inter node reachability.

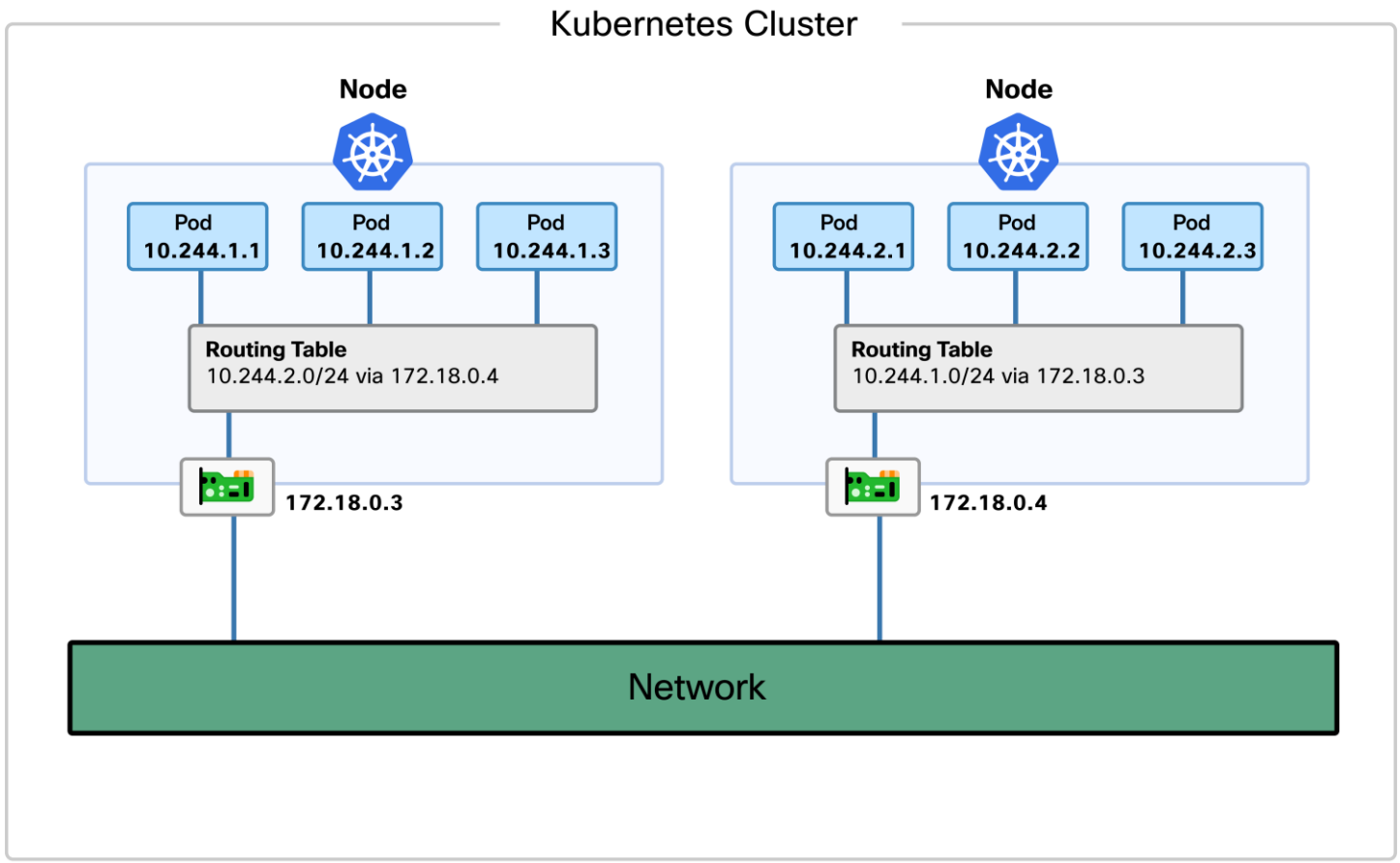




© 2026 Cisco and/or its affiliates. All rights reserved.

Native Routing

- In native routing, packets sent by the pods to other pods are routed through the host networking stack.
- Once a packet exits the pod and enters the host's networking namespace, it is routed based on the host's routing table.
- This mode requires either that each Kubernetes node knows the PodCIDRs assigned to all other nodes, or that the underlying network can route Pod IPs directly.



© 2026 Cisco and/or its affiliates. All rights reserved.

Native Routing – Route Distribution

How do we distribute routing information about PodCIDR?

Three Common Methods

Static Routing	Auto Route Injection	BGP

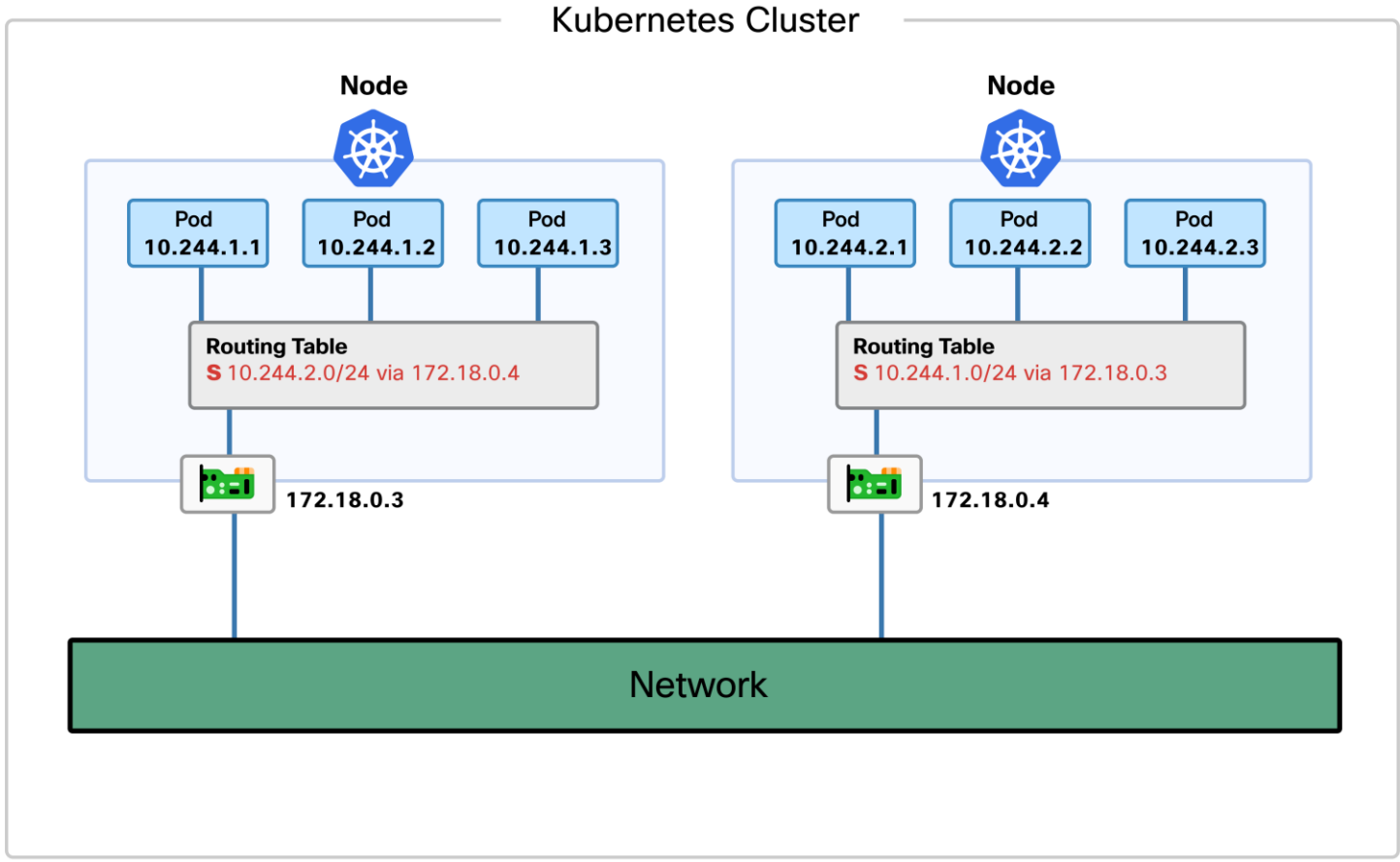
© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Static Routing

Administrators manually install routes on every node (and/or across the network) to reach PodCIDRs assigned to other nodes.

This works only for tiny stable clusters since any node change requires updating every route, making the method impractical at scale.

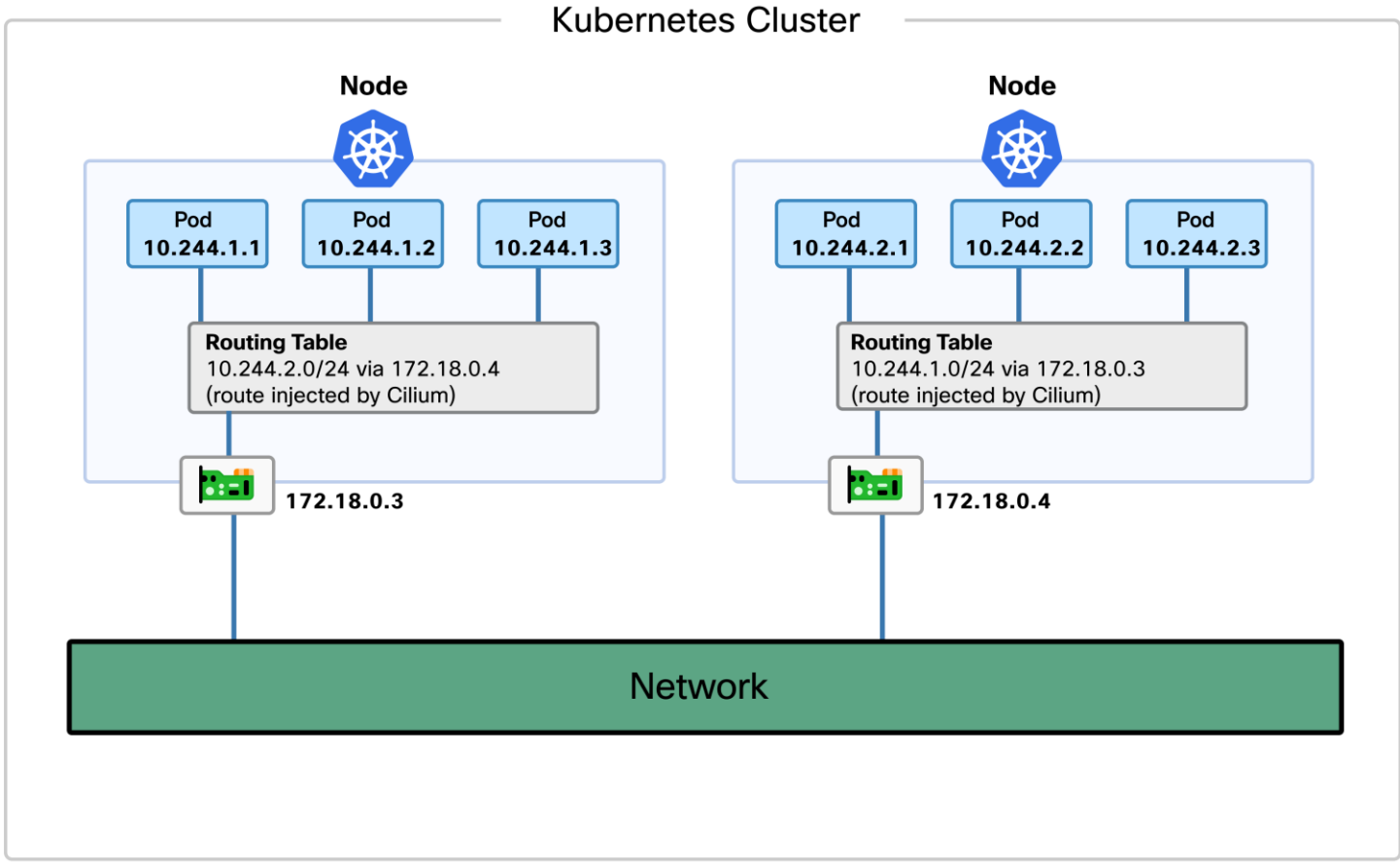


© 2026 Cisco and/or its affiliates. All rights reserved.

Auto Route Injection

Cilium automatically adds routes for every PodCIDR to each node's routing table.

This requires a shared Layer 2 network but provides simple automatic distribution of pod networks without manual configuration or external routing systems.

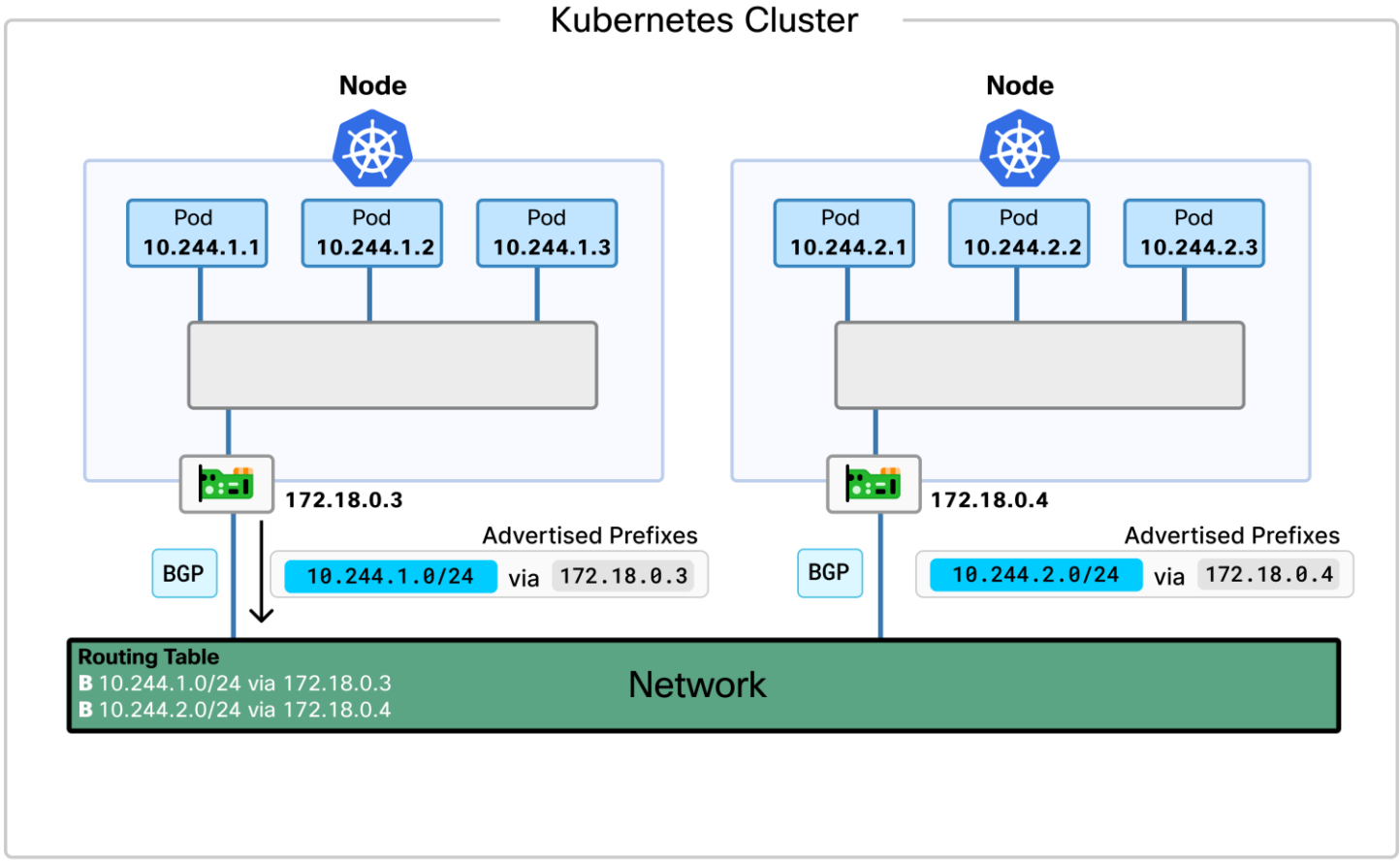


© 2026 Cisco and/or its affiliates. All rights reserved.

BGP

Nodes advertise their PodCIDRs to the wider network using BGP.

Routing devices learn pod prefixes dynamically, allowing scalable automatic updates when nodes join leave or move, making it the most robust option for production environments.

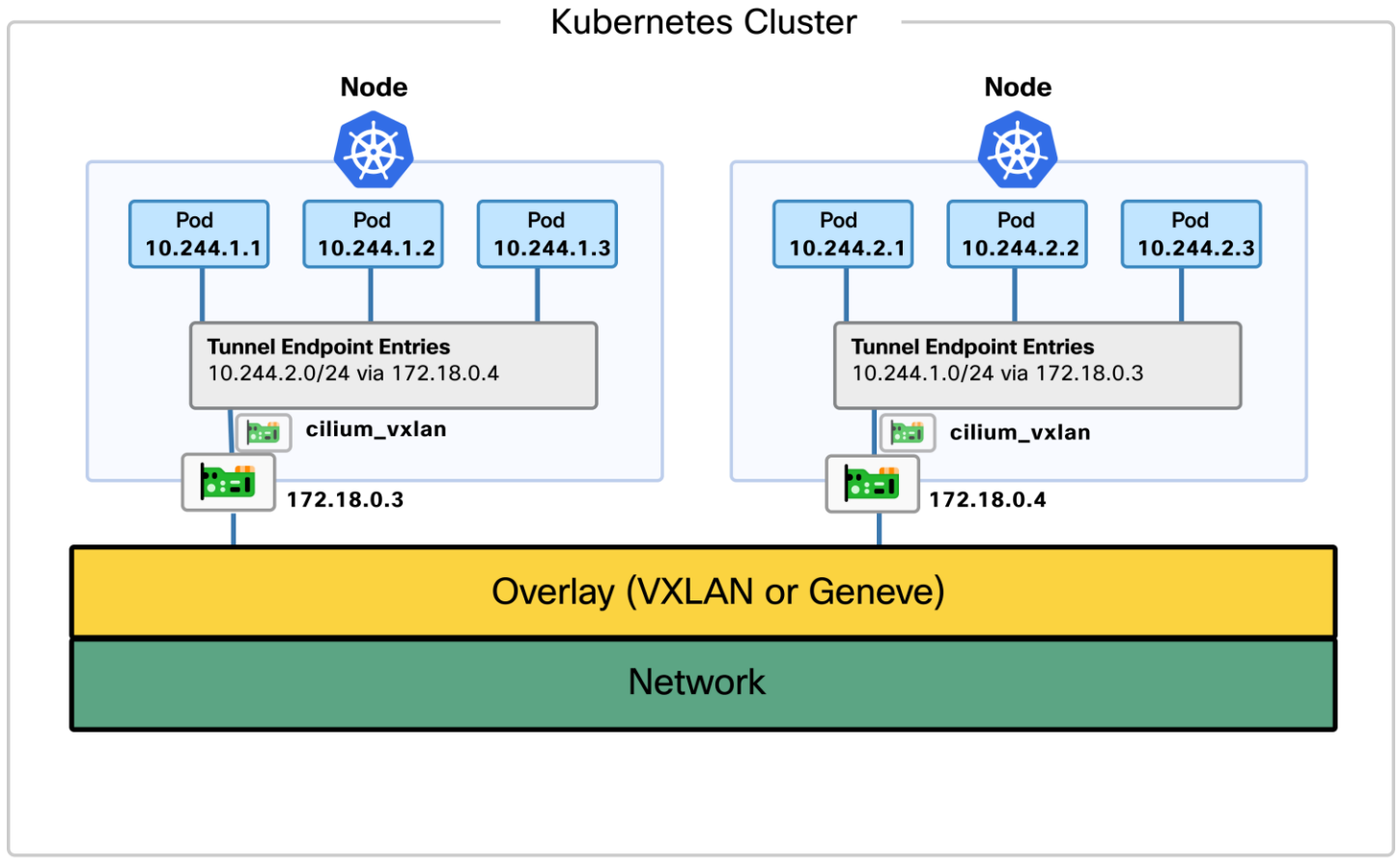


© 2026 Cisco and/or its affiliates. All rights reserved.

Encapsulation Mode

a.k.a. tunnel mode, a.k.a. overlay mode

- In this mode, all nodes form a full set of overlay tunnels using a UDP-based encapsulation protocol such as VXLAN o Geneve.
- This is Cilium's **default routing mode.**
- Encapsulation is most useful in environments where you cannot control the underlay routing.
- Instead of requiring the physical network to carry pod prefixes, Cilium hides pod addressing behind node IPs.
- From the underlay's perspective, all inter-pod traffic looks like node-to-node traffic between Kubernetes nodes.



© 2026 Cisco and/or its affiliates. All rights reserved.

VXLAN or Geneve?

Encapsulation Mode

VXLAN

Default mode.

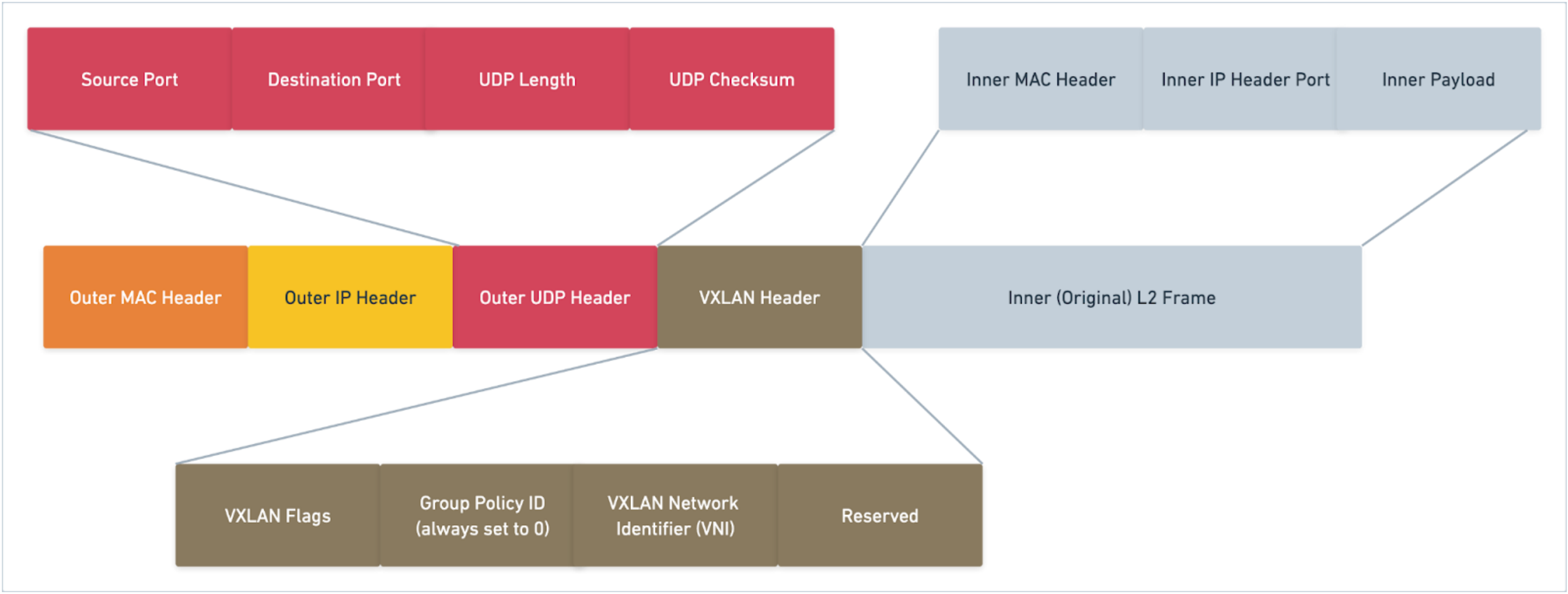
Geneve

Less commonly deployed.

Supports Direct Server Return, using Geneve's TLV.

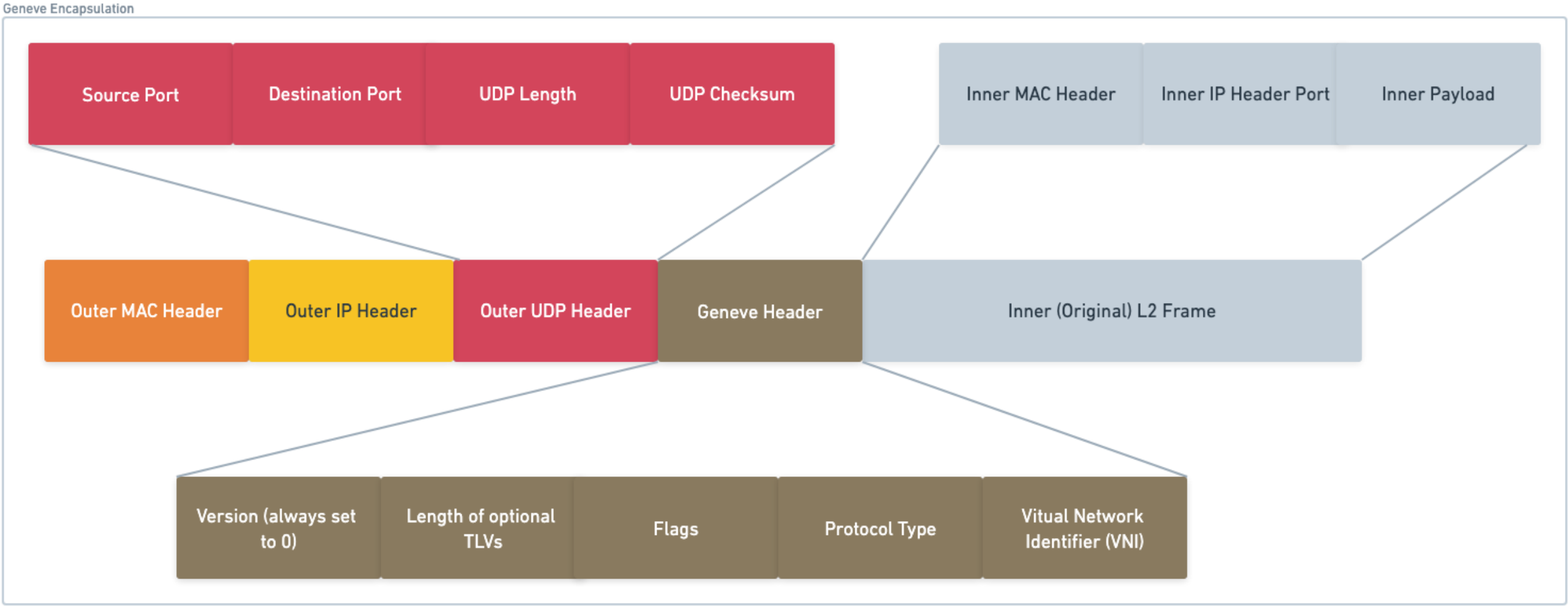
© 2026 Cisco and/or its affiliates. All rights reserved.

VXLAN Encapsulation Mode



© 2026 Cisco and/or its affiliates. All rights reserved.

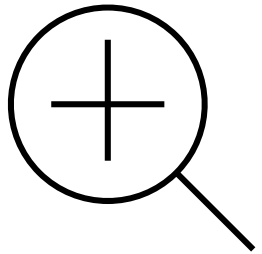
Geneve Encapsulation Mode



© 2026 Cisco and/or its affiliates. All rights reserved.

Looking into the Packet (VXLAN)

Request

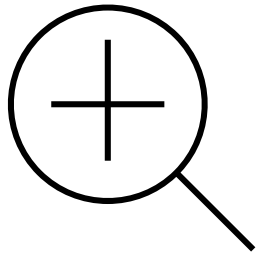


```
› Frame 28: Packet, 191 bytes on wire (1528 bits), 191 bytes captured (1528 bits)
› Ethernet II, Src: 02:42:ac:12:00:05 (02:42:ac:12:00:05), Dst: 02:42:ac:12:00:03 (02:42:ac:12:00:03)
› Internet Protocol Version 4, Src: 172.18.0.5, Dst: 172.18.0.3
› User Datagram Protocol, Src Port: 41914, Dst Port: 8472
⌄ Virtual eXtensible Local Area Network
  › Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 62651
    Reserved: 0
› Ethernet II, Src: 12:46:40:12:4e:45 (12:46:40:12:4e:45), Dst: fe:fc:e9:58:b4:01 (fe:fc:e9:58:b4:01)
› Internet Protocol Version 4, Src: 10.244.1.6, Dst: 10.244.3.193
› Transmission Control Protocol, Src Port: 52542, Dst Port: 80, Seq: 1, Ack: 1, Len: 75
⌄ Hypertext Transfer Protocol
  › GET / HTTP/1.1\r\n
    Host: 10.244.3.193\r\n
    User-Agent: curl/8.7.1\r\n
    Accept: */*\r\n
    \r\n
    [Response in frame: 32]
    [Full request URI: http://10.244.3.193/]
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Looking into the Packet (VXLAN)

Response

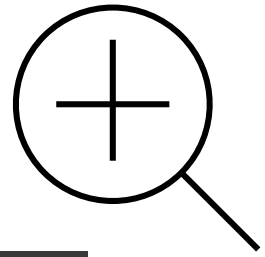


```
> Frame 28: Packet, 191 bytes on wire (1528 bits), 191 bytes captured (1528 bits)
> Ethernet II, Src: 02:42:ac:12:00:05 (02:42:ac:12:00:05), Dst: 02:42:ac:12:00:03 (02:42:ac:12:00:03)
> Internet Protocol Version 4, Src: 172.18.0.5, Dst: 172.18.0.3
v User Datagram Protocol, Src Port: 41914, Dst Port: 8472
    Source Port: 41914
    Destination Port: 8472
    Length: 157
  > Checksum: 0x0000 [zero-value ignored]
    [Stream index: 12]
    [Stream Packet Number: 3]
  > [Timestamps]
    UDP payload (149 bytes)
> Virtual eXtensible Local Area Network
> Ethernet II, Src: 12:46:40:12:4e:45 (12:46:40:12:4e:45), Dst: fe:fc:e9:58:b4:01 (fe:fc:e9:58:b4:01)
> Internet Protocol Version 4, Src: 10.244.1.6, Dst: 10.244.3.193
> Transmission Control Protocol, Src Port: 52542, Dst Port: 80, Seq: 1, Ack: 1, Len: 75
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: 10.244.3.193\r\n
    User-Agent: curl/8.7.1\r\n
    Accept: */*\r\n
    \r\n
    [Response in frame: 32]
    [Full request URI: http://10.244.3.193/]
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Looking into the Packet (Geneve)

Request

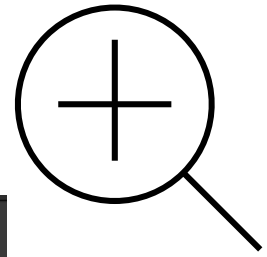


```
> Frame 128: Packet, 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits)
> Ethernet II, Src: Microsoft_66:44:1d (7c:ed:8d:66:44:1d), Dst: 12:34:56:78:9a:bc (12:34:56:78:9a:bc)
> Internet Protocol Version 4, Src: 192.168.10.5, Dst: 192.168.10.6
> User Datagram Protocol, Src Port: 40314, Dst Port: 6081
v Generic Network Virtualization Encapsulation, VNI: 0x000006
    Version: 0
    Length: 0 bytes
  > Flags: 0x00
    Protocol Type: Transparent Ethernet bridging (0x6558)
    Virtual Network Identifier (VNI): 0x000006 (6)
> Ethernet II, Src: 7e:97:26:a4:f0:9f (7e:97:26:a4:f0:9f), Dst: 42:68:28:e6:f5:9d (42:68:28:e6:f5:9d)
> Internet Protocol Version 4, Src: 10.0.0.187, Dst: 10.0.1.219
> Transmission Control Protocol, Src Port: 36470, Dst Port: 4240, Seq: 1, Ack: 1, Len: 101
v Hypertext Transfer Protocol
  > GET /hello HTTP/1.1\r\n
    Host: 10.0.1.219:4240\r\n
    User-Agent: Go-http-client/1.1\r\n
    Accept-Encoding: gzip\r\n
    \r\n
    [Response in frame: 130]
    [Full request URI: http://10.0.1.219:4240/hello]
```

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Looking into the Packet (Geneve)



Response

```
> Frame 130: Packet, 191 bytes on wire (1528 bits), 191 bytes captured (1528 bits)
> Ethernet II, Src: AristaNetwor_3b:c0:04 (d4:af:f7:3b:c0:04), Dst: Microsoft_66:44:1d (7c:ed:8d:66:44:1d)
> Internet Protocol Version 4, Src: 192.168.10.6, Dst: 192.168.10.5
> User Datagram Protocol, Src Port: 3024, Dst Port: 6081
v Generic Network Virtualization Encapsulation, VNI: 0x000004
    Version: 0
    Length: 0 bytes
  > Flags: 0x00
    Protocol Type: Transparent Ethernet bridging (0x6558)
    Virtual Network Identifier (VNI): 0x000004 (4)
> Ethernet II, Src: d6:2a:21:67:94:5c (d6:2a:21:67:94:5c), Dst: 12:97:f8:63:d9:88 (12:97:f8:63:d9:88)
> Internet Protocol Version 4, Src: 10.0.1.219, Dst: 10.0.0.187
> Transmission Control Protocol, Src Port: 4240, Dst Port: 36470, Seq: 1, Ack: 102, Len: 75
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Thu, 26 Jun 2025 14:48:14 GMT\r\n
  > Content-Length: 0\r\n
    \r\n
    [Request in frame: 128]
    [Time since request: 1.436000 milliseconds]
    [Request URI: /hello]
    [Full request URI: http://10.0.1.219:4240/hello]
```

© 2026 Cisco and/or its affiliates. All rights reserved.

VXLAN or Geneve?

Encapsulation Mode

VXLAN

Default routing mode.

Default encapsulation mode used when encapsulation protocol is not specified.

Geneve

Less commonly deployed.

Supports Direct Server Return, using Geneve's TLV.

© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes Services

Kubernetes Service Networking Model

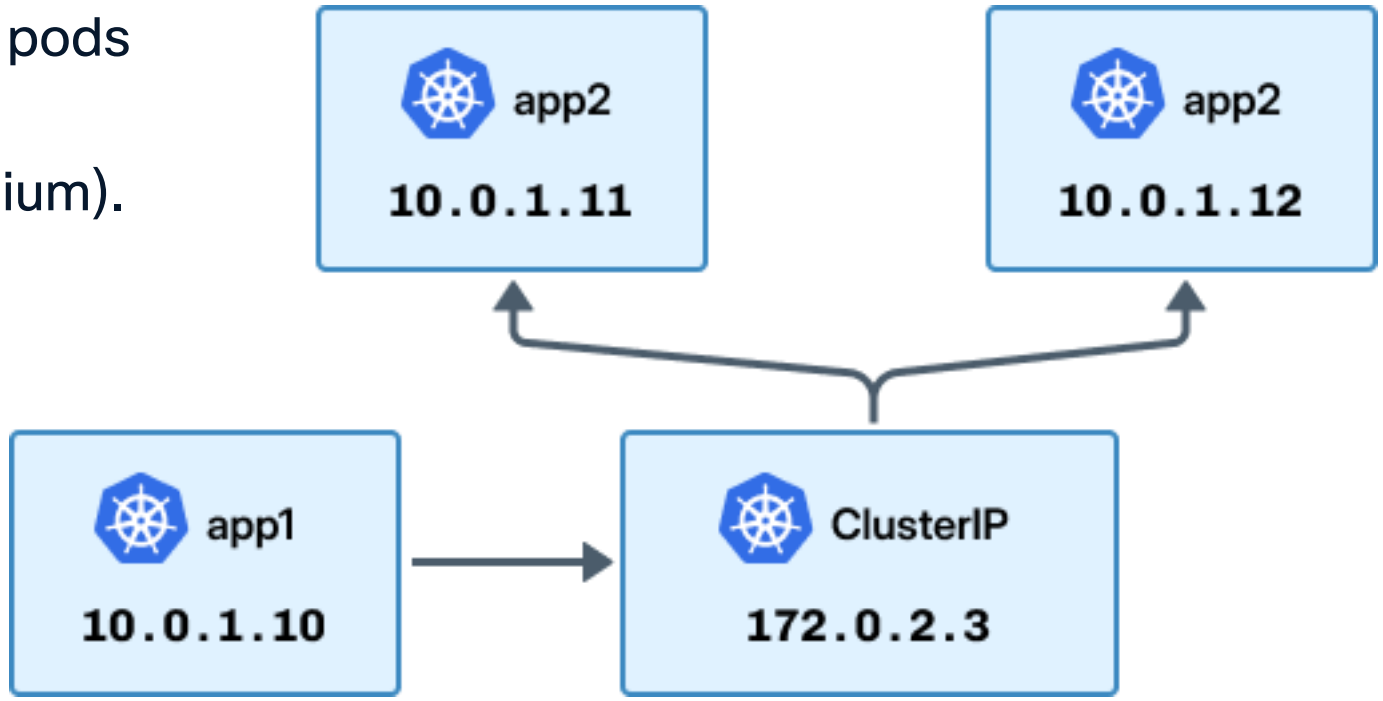
- Kubernetes workloads change often so pod IPs cannot provide stable connectivity.
- Services give consistent access using a virtual name and a deterministic cluster virtual address. By default, **kube proxy** manages service traffic inside clusters and has known limitations.
- Kubernetes supports several types of services for different use cases.
  - ClusterIP
  - NodePort
  - LoadBalancer

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Kubernetes ClusterIP Service

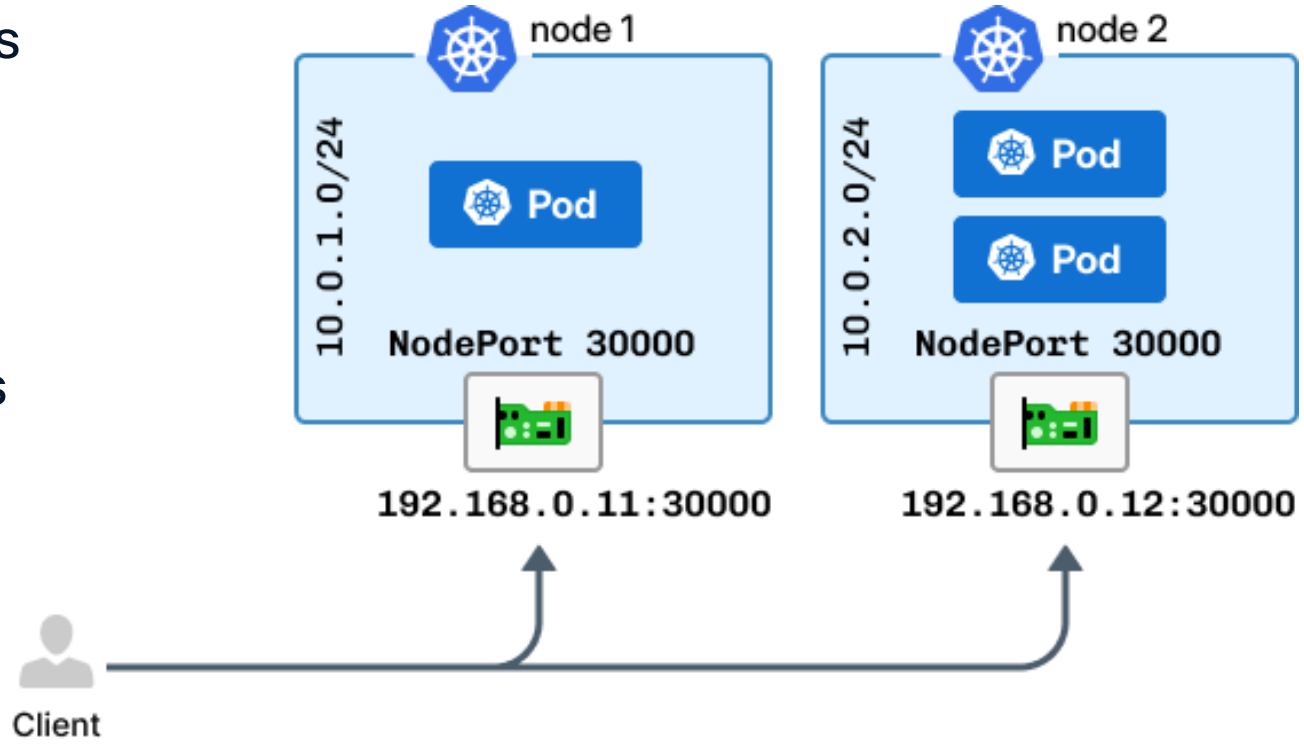
- Default Service
- Provides an in-cluster VIP so that other pods can reach the Service.
- IP is assigned by Kubernetes (not by Cilium).



© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes NodePort Service

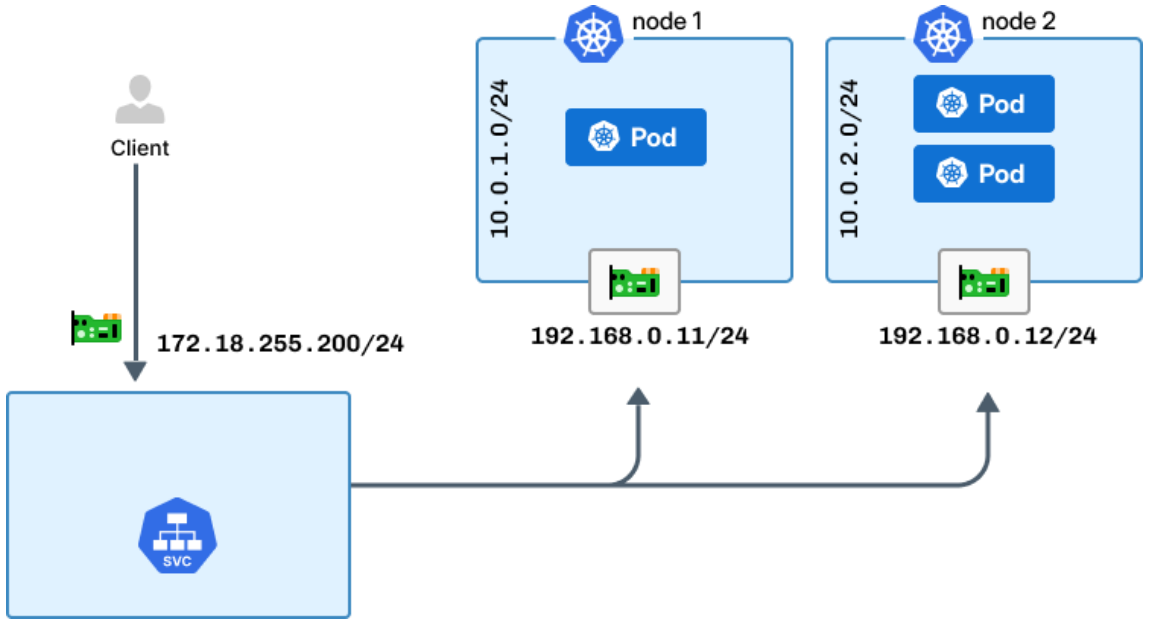
- NodePort exposes the Service on the same static port across all nodes, which allows basic external access.
- To make the node port available, Kubernetes sets up a cluster IP address, the same as if you had requested a ClusterIP Service. The difference is that NodePort also exposes a port on every node towards this IP.
- NodePort is simple but has limitations: users often run into issues with the constrained 30000- 32767 port range, and NodePort does not natively offer load-balancing capabilities.



© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes LoadBalancer Service

- LoadBalancer builds on ClusterIP and NodePort service behavior.
- Cilium can allocate external IPs to this type of services and forward traffic.
- In cloud environments, providers assign IPs via native load balancers.
- Kubernetes exposes NodePorts on all nodes for external access.
- On premises, Cilium provides load balancing with LB IPAM.



© 2026 Cisco and/or its affiliates. All rights reserved.

Kubernetes LoadBalancer IPAM

- Cilium also acts as an IPAM platform for Kubernetes LoadBalancer Service types
- Define the pool of your choice and assign IPs to services based on metadata.
- Feature enabled and dormant by default.
- Capability previously required a separate tool (MetalLB) but now natively built-in Cilium.
- IPv4 and IPv6 support.

```yaml
apiVersion: "cilium.io/v2"
kind: CiliumLoadBalancerIPPool
metadata:
  name: "production-pool"
spec:
  allowFirstLastIPs: "No"
  blocks:
  - cidr: "100.16.38.0/24"
  serviceSelector:
    matchLabels:
      environment: production
```

© 2026 Cisco and/or its affiliates. All rights reserved.

What did we learn?

3 common type of Kubernetes services

ClusterIP

- Used for internal access

NodePort

- Used for external access
- Leverages node IPs with fixed high port
- Simple but requires manual load-balancing and IP management

LoadBalancer

- Used for external access
- Slightly more complex than NodePort
- Provides a single stable external IP
- Recommended for production

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Cluster Ingress



Ingress Access

- Access to applications has traditionally been done over the **Ingress API**.
- Ingress is a Kubernetes-adjacent API that defines how to bring external HTTP traffic into services using defined routing rules.
- **Gateway API** is the successor to Ingress API.
- Gateway API and Ingress API provide similar capabilities to **Layer 7 Load Balancer / Reverse Proxy**.
- To use Ingress and Gateway APIs, you require a “controller”.
- Cilium natively supports both Ingress API and Gateway API.



© 2026 Cisco and/or its affiliates. All rights reserved.

Gateway API

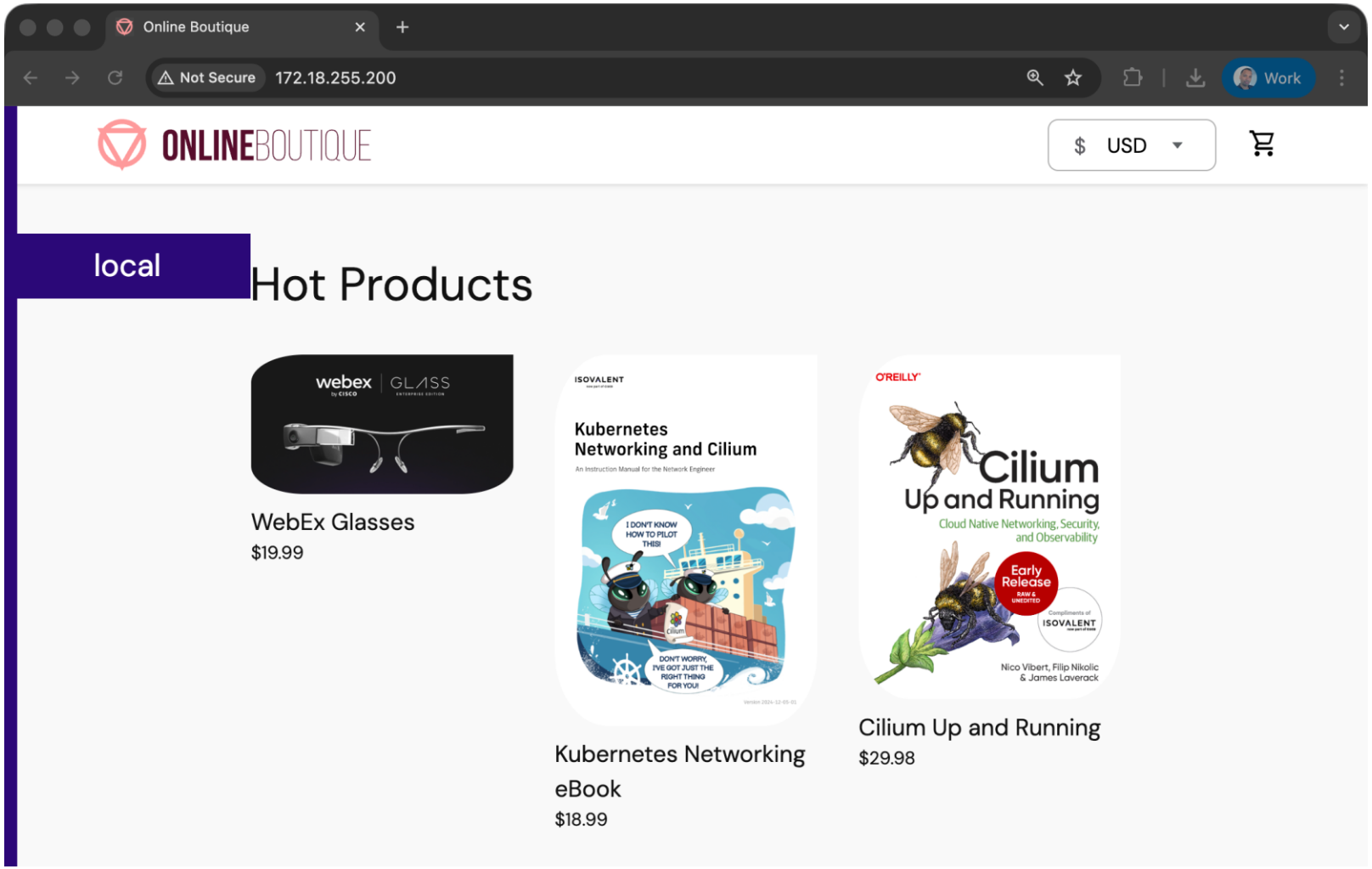
- When a Gateway API object is created, a LoadBalancer Service will be automatically deployed and will be assigned an IP address.

```
$ kubectl get svc -n ciscoboutique
NAMESPACE       NAME                      TYPE          EXTERNAL-IP
ciscoboutique   cilium-boutique-gateway   LoadBalancer  172.18.255.200
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: cisco-boutique-gateway
  namespace: ciscoboutique
spec:
  gatewayClassName: cilium
  listeners:
  - allowedRoutes:
      namespaces:
        from: Same
    name: http
    port: 80
    protocol: HTTP
```

© 2026 Cisco and/or its affiliates. All rights reserved.

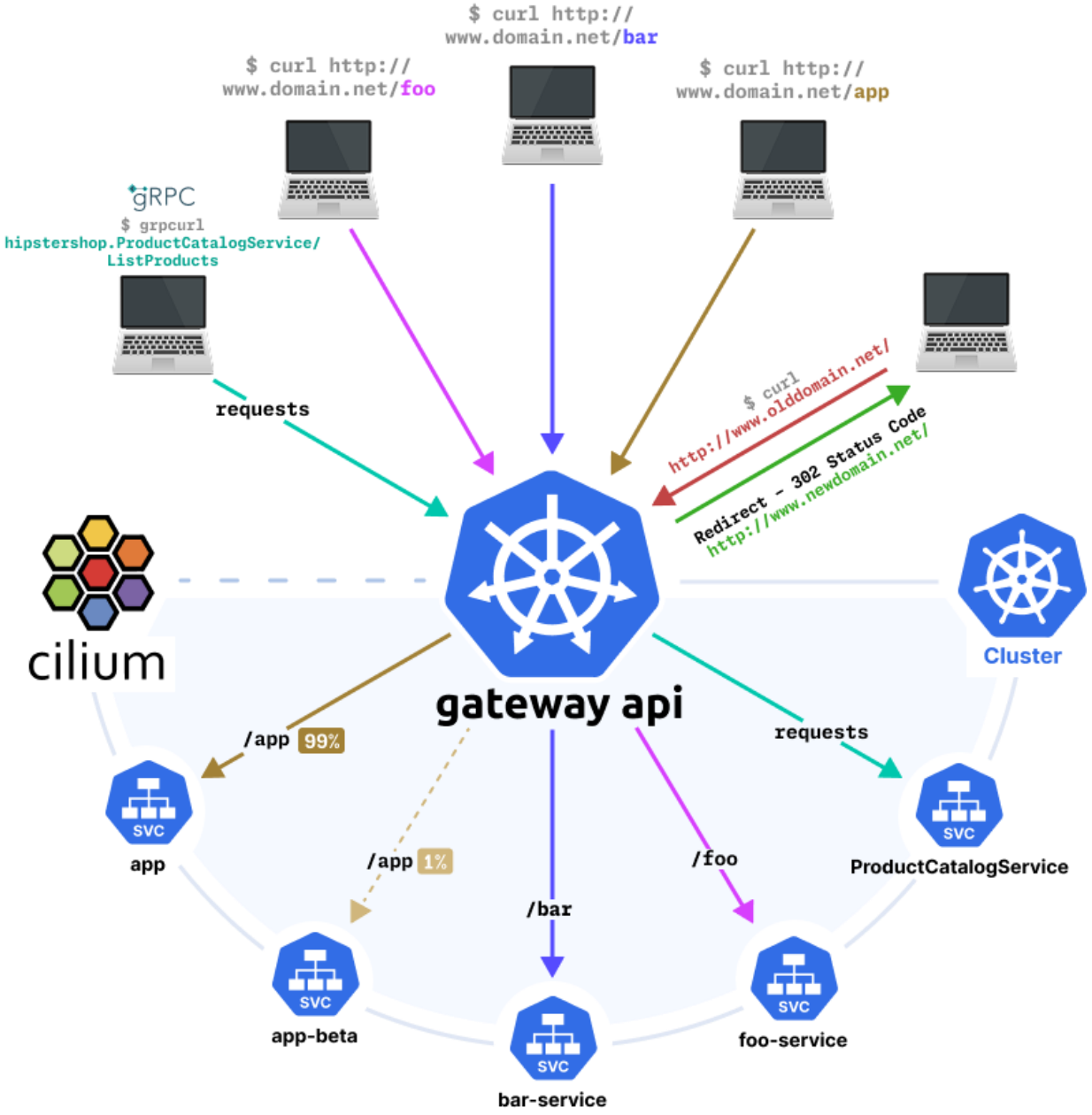
Gateway API



© 2026 Cisco and/or its affiliates. All rights reserved.

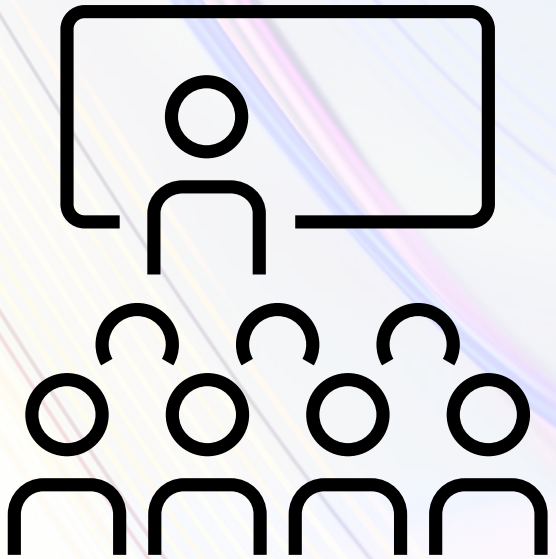
Gateway API

- Gateway API.supports a multitude of common use cases:
  - Path- and header-based routing
  - Load Balancing (traffic splitting)
  - Header manipulation (request and response)
  - Canary testing
  - gRPC routing
  - Redirection
  - Etc...



© 2026 Cisco and/or its affiliates. All rights reserved.

Gateway API Demo



Gateway API

Manifest Examples – Gateway Configuration

- Dual protocol support (HTTP & HTTPS)
- TLS termination with wildcard certificate
- Certificate management via Kubernetes secrets

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: cisco-boutique-gateway
  namespace: ciscoboutique
spec:
  gatewayClassName: cilium
  listeners:
  - name: http
    port: 80
    protocol: HTTP
  - name: https
    port: 443
    protocol: HTTPS
    tls:
      mode: Terminate
      certificateRefs:
      - kind: Secret
        name: demo-cert
        namespace: ciscoboutique
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Gateway API

Manifest Examples – Gateway Configuration

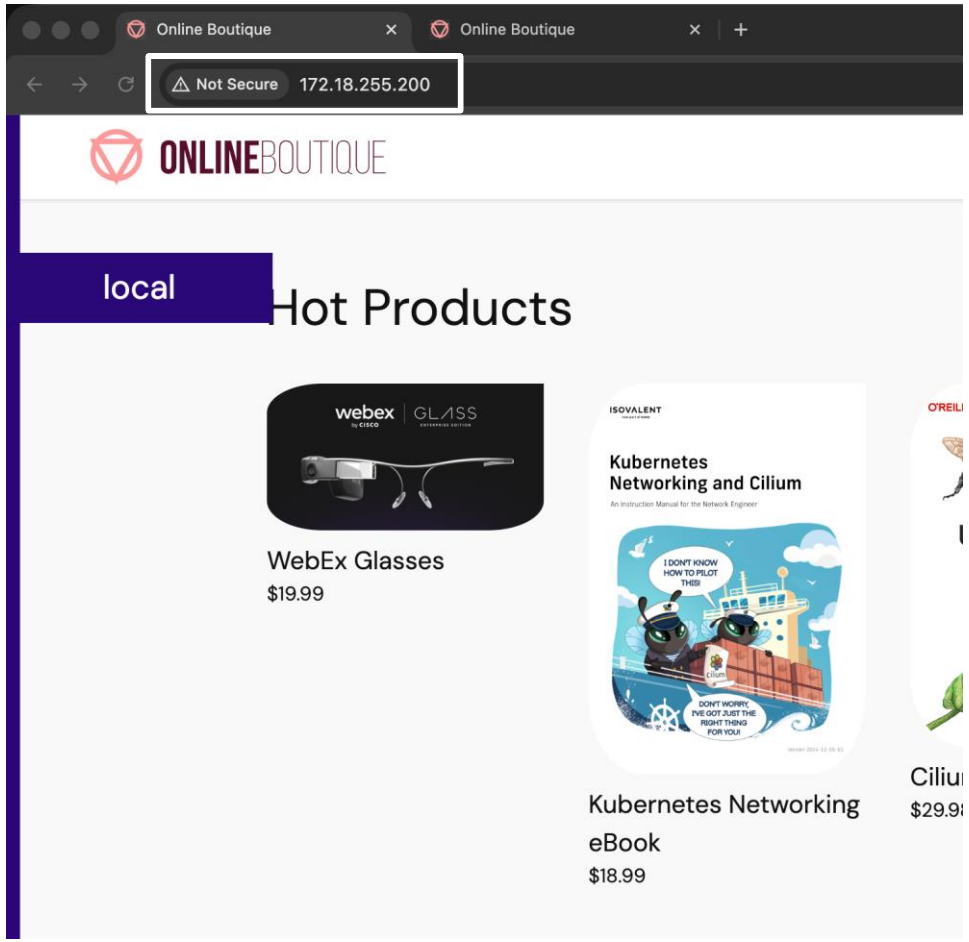
- Dual protocol support (HTTP & HTTPS)
- TLS termination with wildcard certificate
- Certificate management via Kubernetes secrets

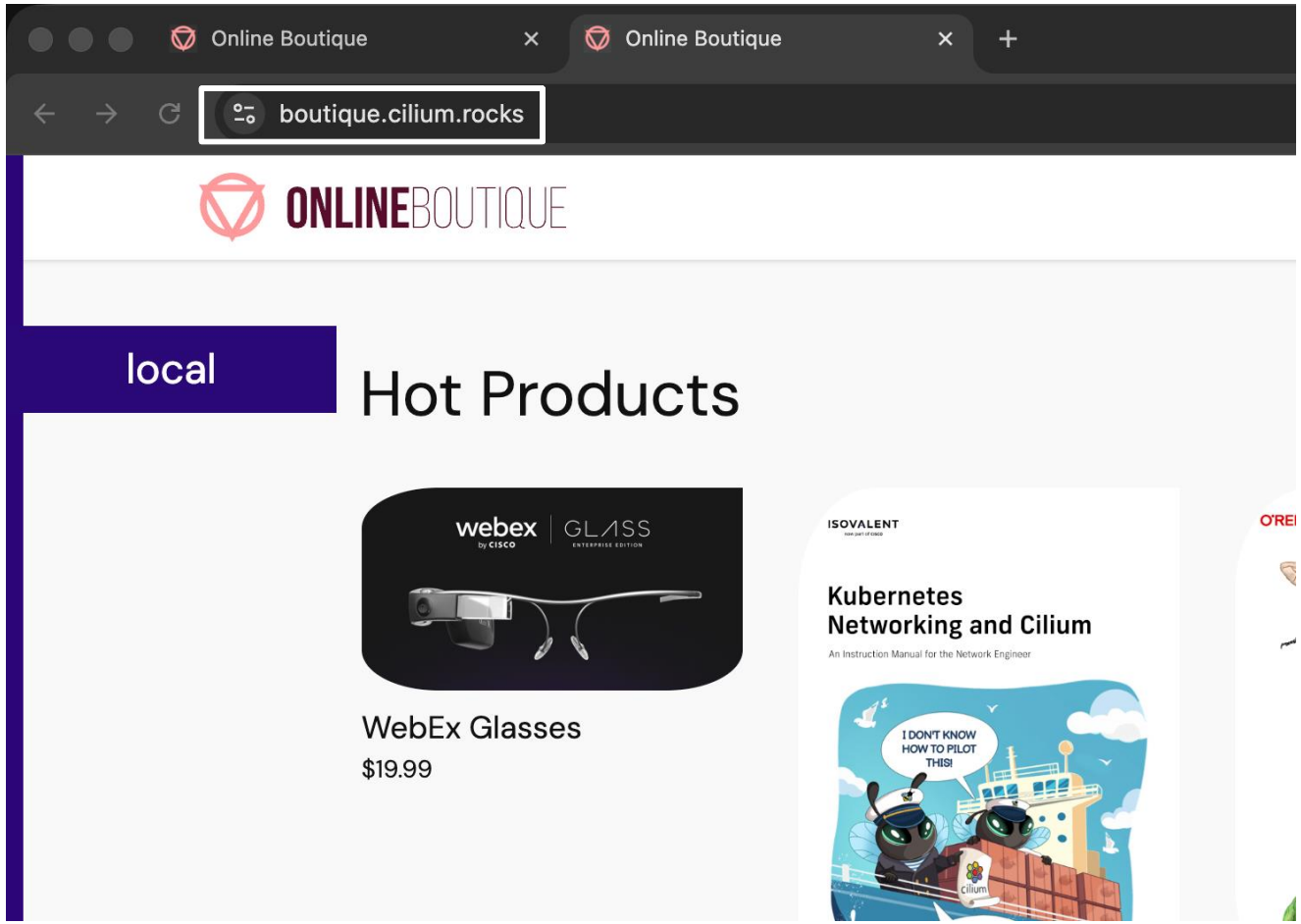
```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: boutique-frontend
  namespace: ciscoboutique
spec:
  parentRefs:
  - name: cisco-boutique-gateway
  hostnames:
  - "boutique.cilium.rocks"
  - "www.boutique.cilium.rocks"
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /
    backendRefs:
    - name: frontend
      port: 80
```

© 2026 Cisco and/or its affiliates. All rights reserved.

TLS Termination

Example with Online Boutique Demo





© 2026 Cisco and/or its affiliates. All rights reserved.

Gateway API

Manifest Examples – A/B testing

- Load balancing between two echo services
- 99/1 traffic split using weight-based routing
- Path-based routing on /a-b-testing prefix
- Gateway API attachment to cilium-gateway

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: a-b-testing
spec:
  parentRefs:
  - name: cilium-gateway
    namespace: default
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /a-b-testing
    backendRefs:
    - kind: Service
      name: app
      port: 8080
      weight: 99
    - kind: Service
      name: app-beta
      port: 8090
      weight: 1
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Gateway API

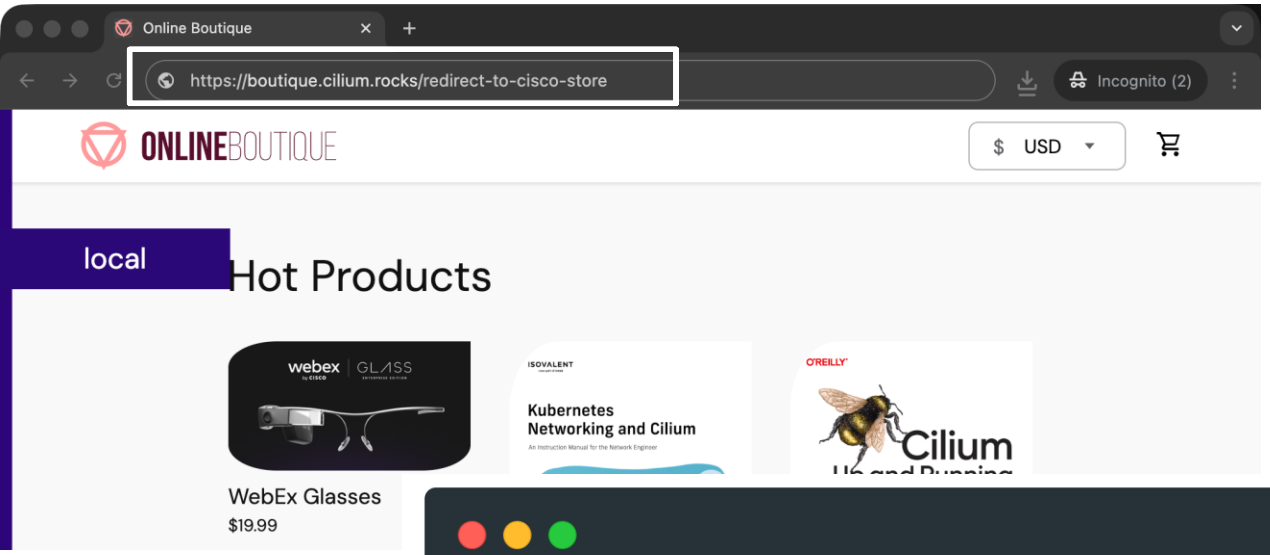
Manifest Examples - Redirection

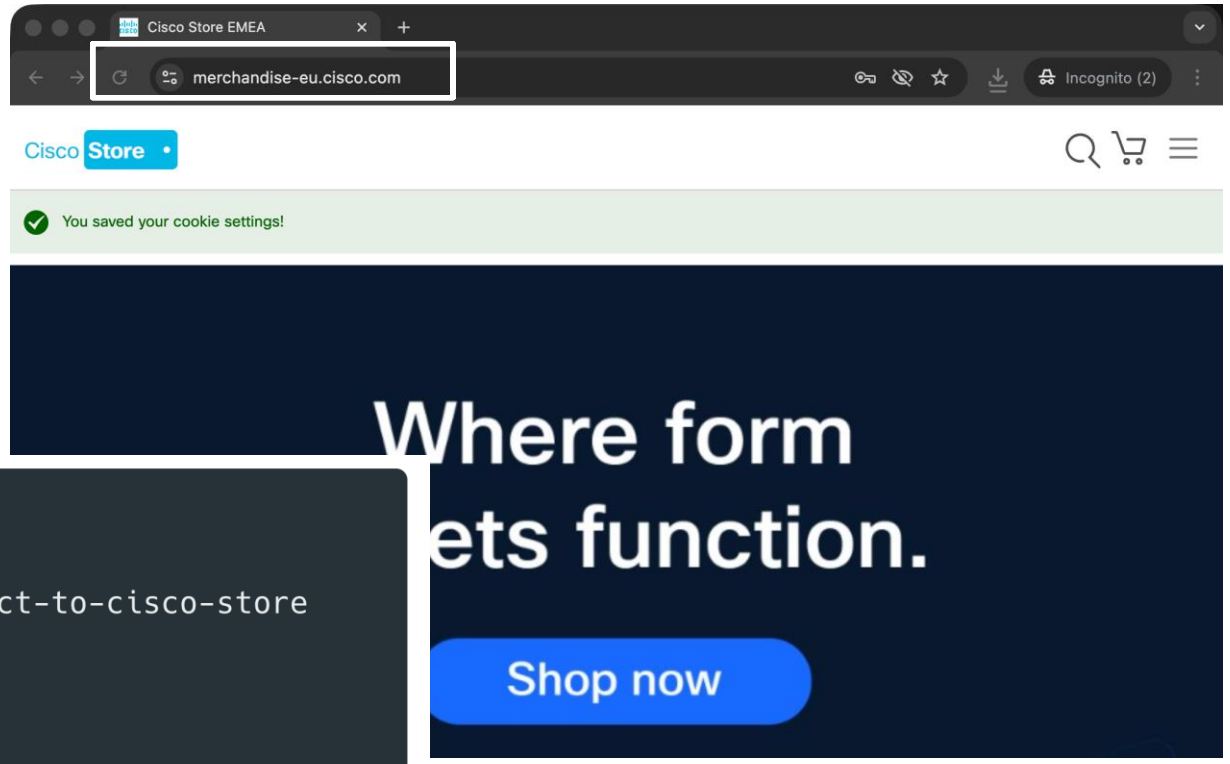
- External redirect to Cisco merchandise store
- 301 permanent redirect with HTTPS enforcement
- Path replacement (/redirect-to-cisco-store → /)

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: boutique-redirect
  namespace: ciscoboutique
spec:
  parentRefs:
  - name: cisco-boutique-gateway
  hostnames:
  - "boutique.cilium.rocks"
  - "www.boutique.cilium.rocks"
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /redirect-to-cisco-store
    filters:
    - type: RequestRedirect
      requestRedirect:
        hostname: merchandise-eu.cisco.com
        scheme: "https"
        path:
          type: ReplaceFullPath
          replaceFullPath: "/"
        statusCode: 301
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Redirection



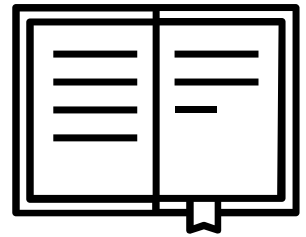



```
$ curl -v https://boutique.cilium.rocks/redirect-to-cisco-store
> GET /redirect-to-cisco-store HTTP/1.1
> Host: boutique.cilium.rocks
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 301 Moved Permanently
< location: https://merchandise-eu.cisco.com:443/
< date: Thu, 05 Feb 2026 18:00:22 GMT
< server: envoy
< content-length: 0
<
* Connection #0 to host boutique.cilium.rocks left intact
```

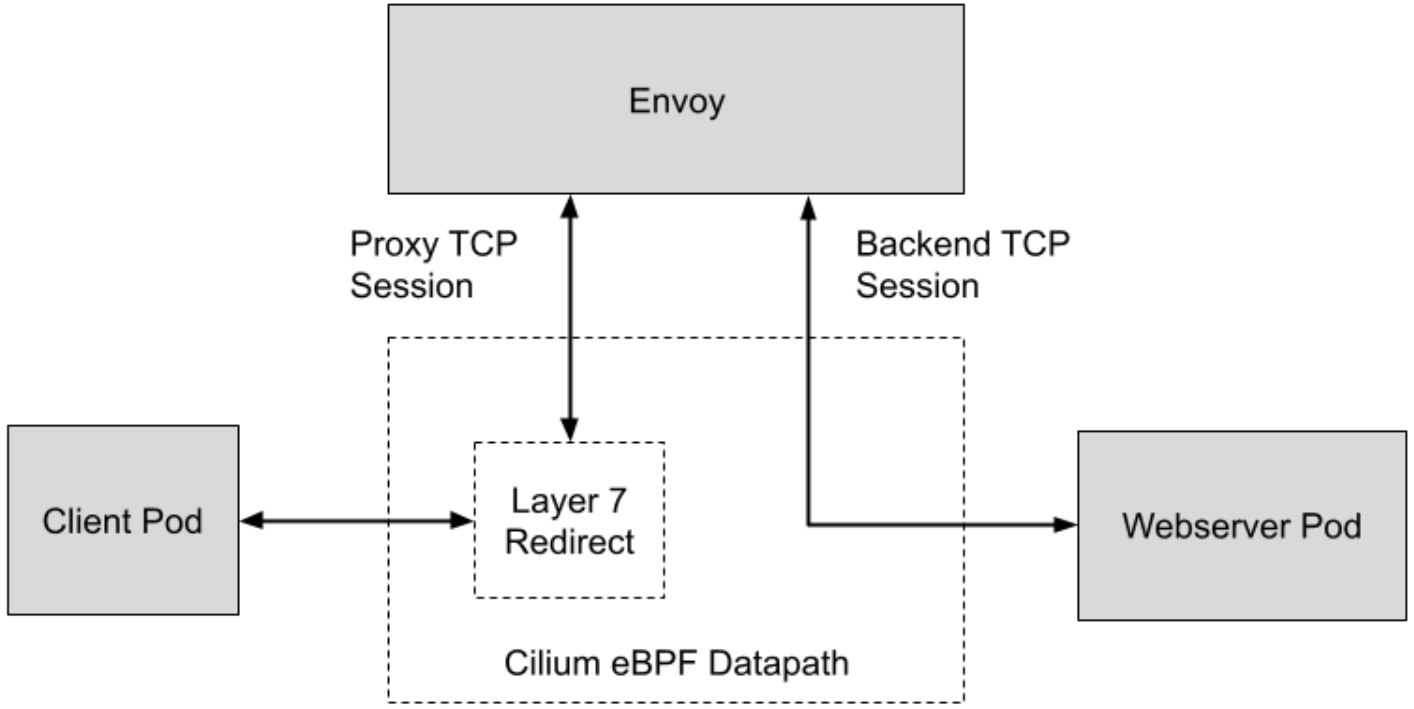
© 2026 Cisco and/or its affiliates. All rights reserved.

Envoy

Why Cilium Uses Envoy for Layer 7



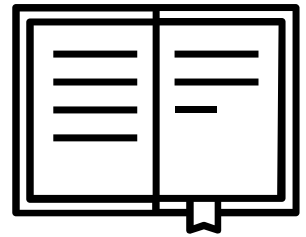
- eBPF delivers high speed decisions at L3/L4 but cannot parse L7 protocols (yet)
- Cilium uses Envoy to enforce HTTP and other application aware rules when L7 filtering is required.
- Envoy is a high-performance layer 7 proxy designed to route, filter, and observe application traffic.
- Traffic is redirected to Envoy only for ports that have configured L7 policies.
- Envoy also powers Gateway API and Ingress.
- Observability features rely on Envoy to extract application metadata unavailable in the kernel datapath.



© 2026 Cisco and/or its affiliates. All rights reserved.

Envoy

Operational Behaviour and Added Latency



- This userspace hop adds small latency and can slightly reduce throughput on busy services.
- Most workloads remain unaffected but latency sensitive applications should test layer 7 policy impact.

This testing below was performed on a kind cluster inside a Linux virtual machine running on a Mac. This is far from an ideal case for Cilium's deployment and performance. The results here, including the scale of the difference, will be much worse than in real deployments. The key point is that there is a difference in performance when using layer 7 policy, not the absolute numbers.

Without Envoy:

```
$ kubectl exec perf -- siege --benchmark --concurrent 25 --reps 100 webserver/api
** SIEGE 4.1.6
** Preparing 25 concurrent users for battle.
The server is now under siege...
Transactions:                   2500 hits
Availability:                 100.00 %
Elapsed time:                  10.21 secs
Data transferred:               0.01 MB
Response time:                  0.03 secs
Transaction rate:             244.86 trans/sec
Throughput:                     0.00 MB/sec
Concurrency:                    6.25
Successful transactions:        2500
Failed transactions:               0
Longest transaction:            5.01
Shortest transaction:           0.00
```

With Envoy:

```
$ kubectl exec perf -- siege --benchmark --concurrent 25 --reps 100 webserver/api
** SIEGE 4.1.6
** Preparing 25 concurrent users for battle.
The server is now under siege...
Transactions:                   2500 hits
Availability:                 100.00 %
Elapsed time:                  20.35 secs
Data transferred:               0.01 MB
Response time:                  0.02 secs
Transaction rate:             122.85 trans/sec
Throughput:                     0.00 MB/sec
Concurrency:                    2.75
Successful transactions:        2500
Failed transactions:               0
Longest transaction:            5.01
Shortest transaction:           0.00
```

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

How do we make the Ingress/GW API IP known to external clients?

Two main options

L2 Announcement

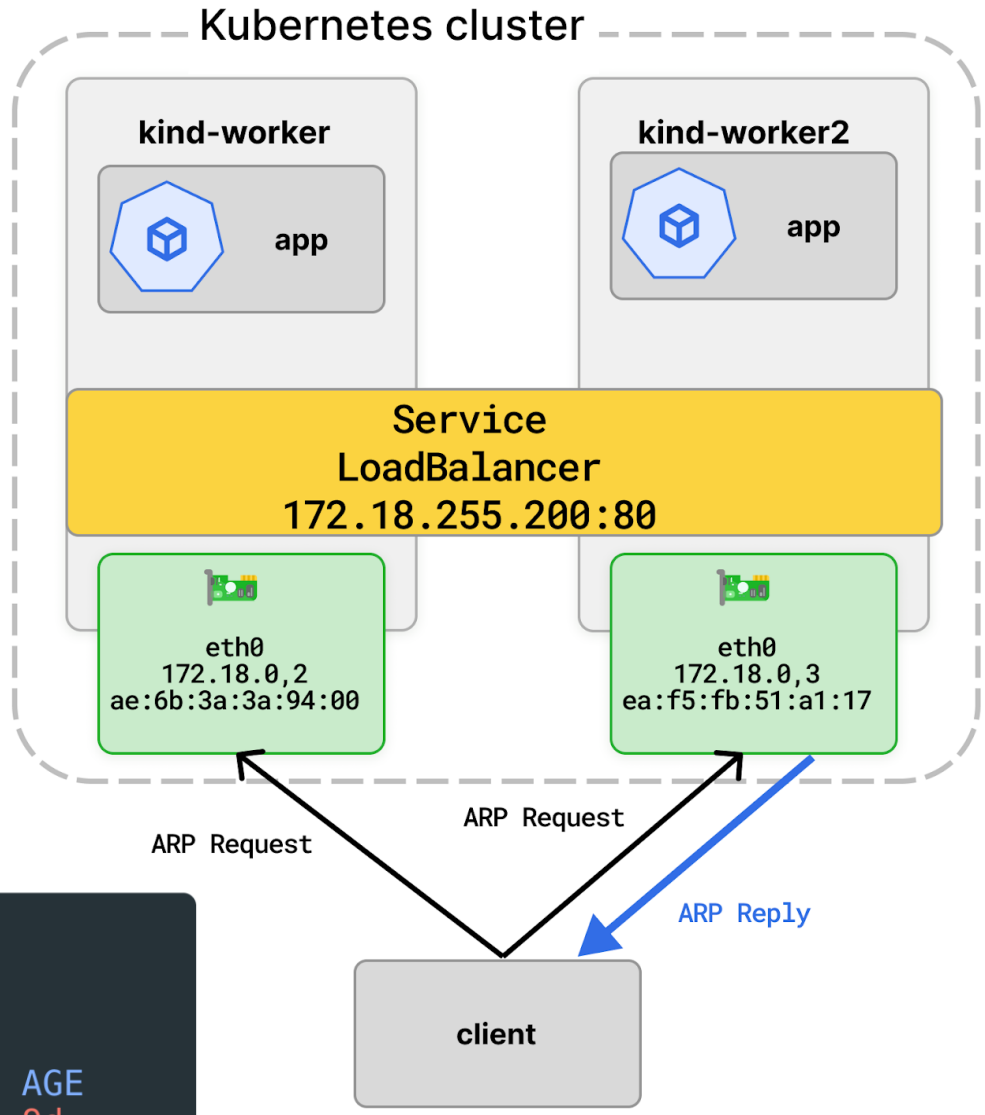
BGP

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Cilium L2 Announcement

- L2 announcements expose services by advertising IPs using ARP.
- Traffic is directed to a single active node at a time.
- Works when clients and nodes share the same subnet.
- Requires minimal configuration and no routing protocol expertise.
- Failover depends on ARP cache behavior and client refresh timing.



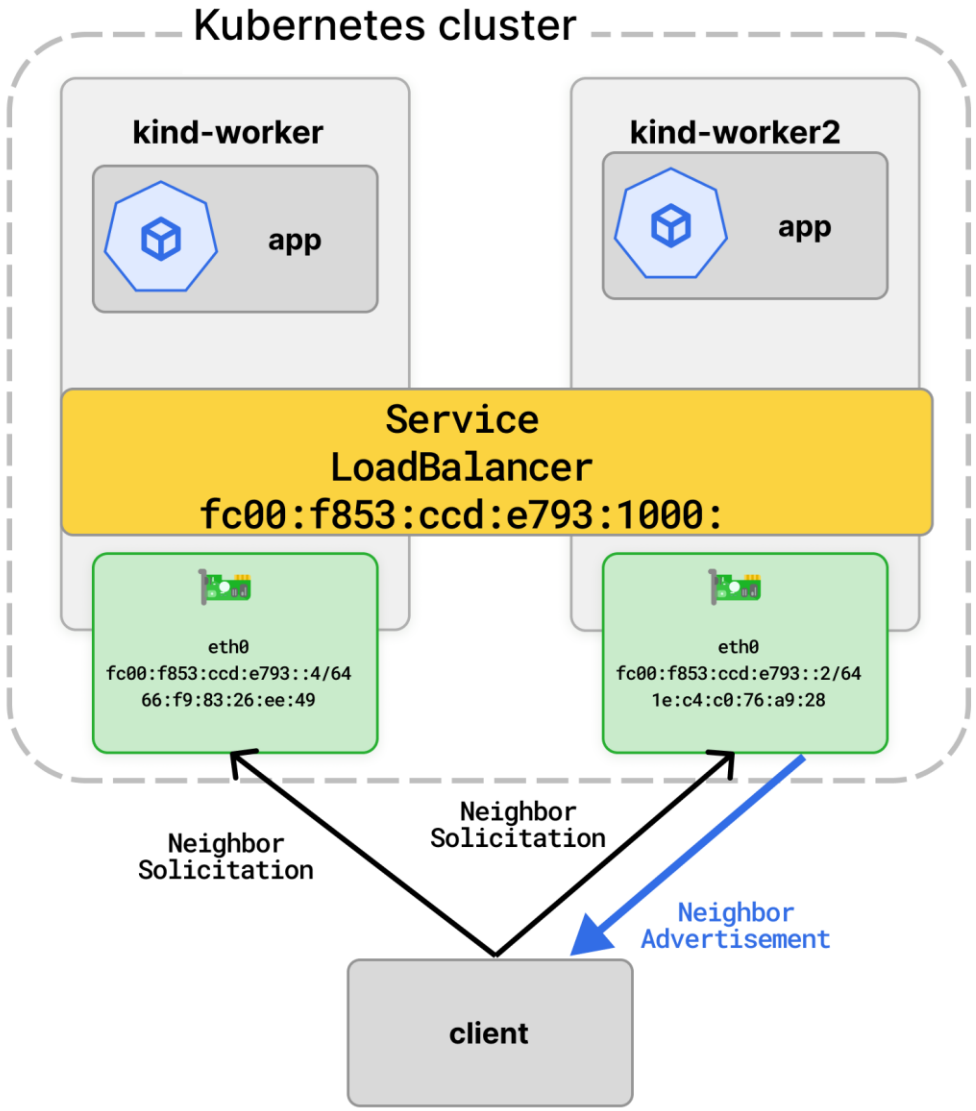

```
$ kubectl -n kube-system get lease
NAME                                                HOLDER          AGE
cilium-l2announce-ciscoboutique-cilium-gateway      kind-worker2    9d
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium L2 Announcement

IPv6 support

- IPv6 support for service announcements was introduced in Cilium 1.19.
- It enables exposing services using native IPv6 addressing.
- ARP is replaced by the IPv6 Neighbor Discovery Protocol.

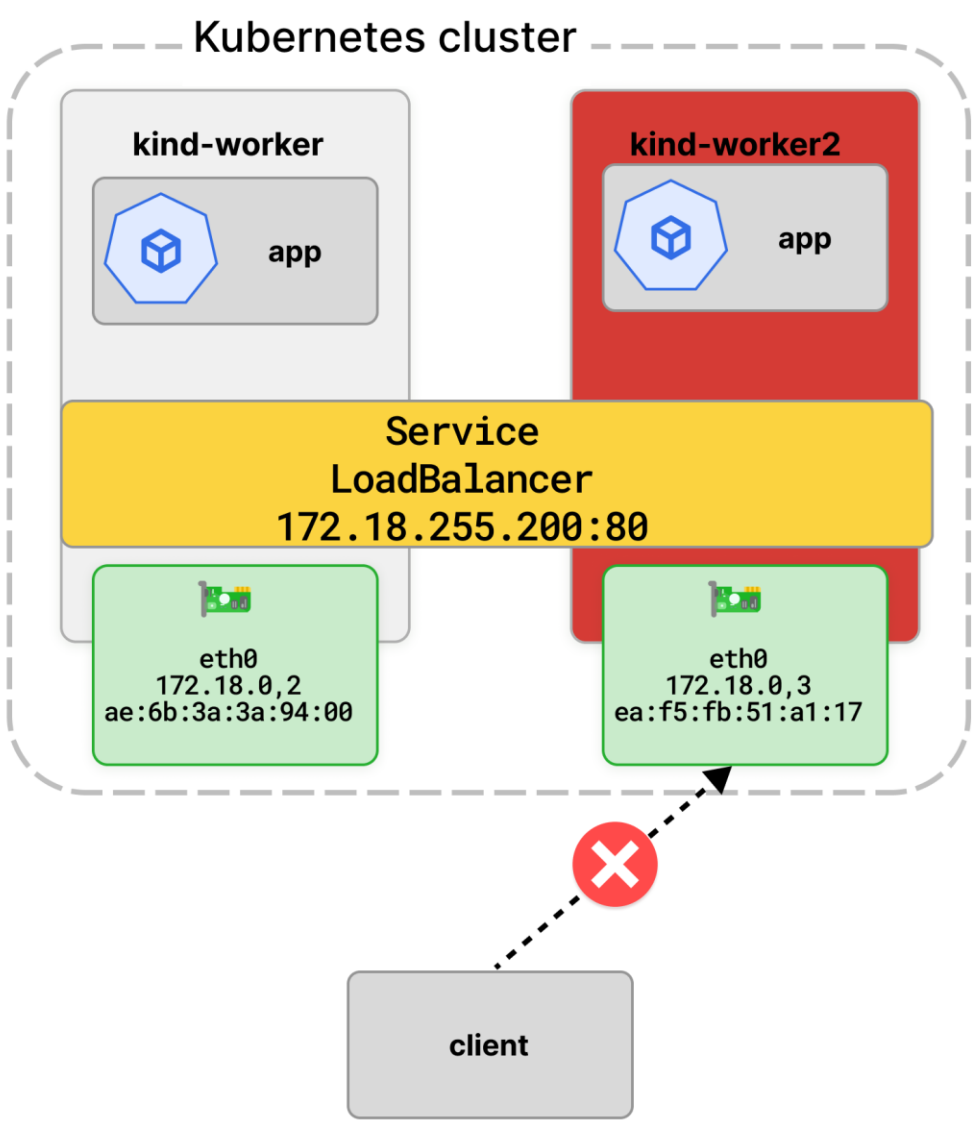


© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium L2 Announcement

Limitations

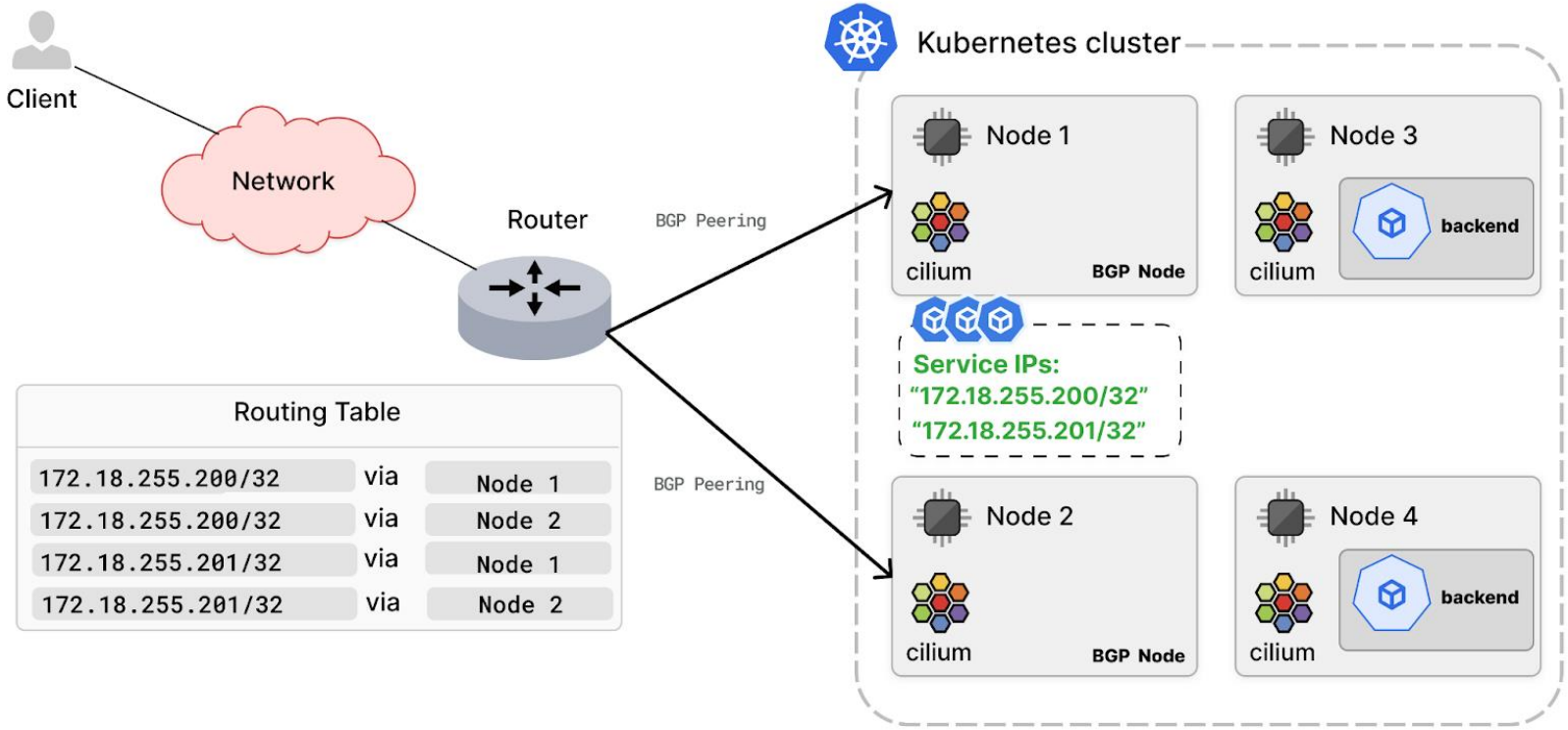
- Only one node serves each virtual IP, preventing load balancing across multiple nodes.
- Failover depends on ARP cache expiry, and faster recovery using gratuitous ARP is unreliable.
- For production environments, BGP offers faster failover, scalability, and stronger operational guarantees.



© 2026 Cisco and/or its affiliates. All rights reserved.

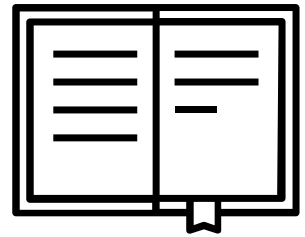
Cilium BGP

- BGP integrates Kubernetes networking with existing routed network infrastructure.
- It allows clusters to participate directly in layer 3 routing fabrics.
- 2 core use cases:
  - Inter-node communications (nodes can advertise PodCIDRs)
  - LoadBalancer and Service IPs can be advertised directly to external routers.



© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium BGP

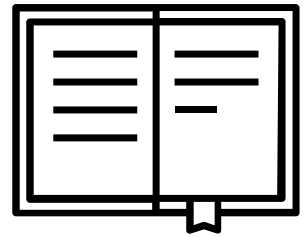


- Cilium manages BGP using Kubernetes custom resources for declarative configuration.
- These resources define peers routing behavior and which routes are advertised.
- *CiliumBGPClusterConfig* selects nodes and defines BGP instances and peers.
- *CiliumBGPPeerConfig* sets reusable peer parameters like timers authentication and families.
- *CiliumBGPAdvertisement* controls advertising of PodCIDRs and Service IPs.

```yaml
apiVersion: cilium.io/v2
kind: CiliumBGPClusterConfig
metadata:
  name: frr
spec:
  nodeSelector:
    matchExpressions:
    - key: node-role.kubernetes.io/control-plane
      operator: DoesNotExist
  bgpInstances:
  - name: instance-64512
    localASN: 64512
    localPort: 179
    peers:
    - name: peer-64512-frr
      peerASN: 64512
      peerAddress: 172.18.0.7
      peerConfigRef:
        name: peer-config
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium BGP



```yaml
apiVersion: cilium.io/v2
kind: CiliumBGPPeerConfig
metadata:
  name: peer-config
spec:
  timers:
    connectRetryTimeSeconds: 30
    keepAliveTimeSeconds: 20
    holdTimeSeconds: 60
  gracefulRestart:
    enabled: true
    restartTimeSeconds: 15
  families:
    - afi: ipv4
      safi: unicast
      advertisements:
        matchLabels:
          advertise: "bgp"
```

```yaml
apiVersion: cilium.io/v2
kind: CiliumBGPAdvertisement
metadata:
  name: bgp-advertisement
  labels:
    advertise: bgp
spec:
  advertisements:
    - advertisementType: "Service"
      service:
        addresses:
        - LoadBalancerIP
      selector:
          - "64512:99"
        localPreference: 200
```

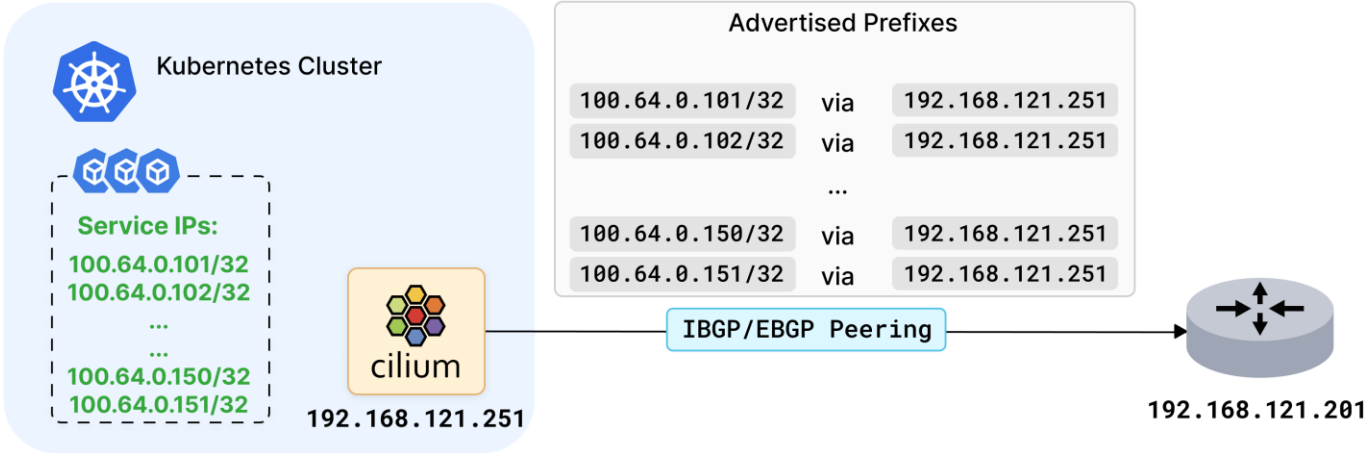
© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

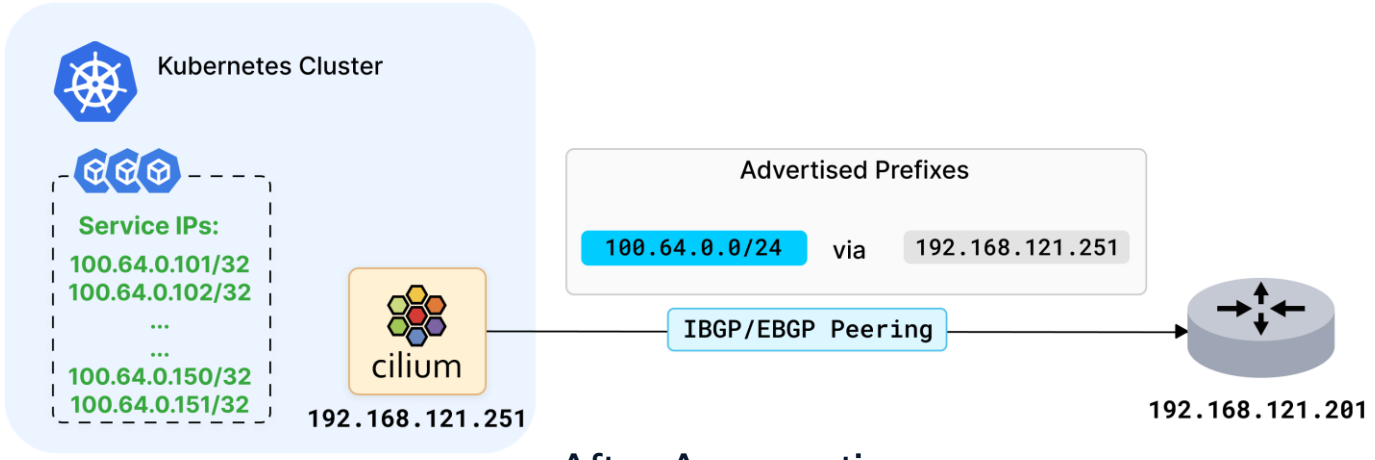
Cilium BGP

- Cilium's BGP engine is based on the open-source project GoBGP.
- Supports wealth of BGP features, including:
  - Custom timers
  - MD5 Authentication
  - eBGP Multihop
  - BGP Attributes:
    - Local Preference
    - Communities

- Isovalent Enterprise support these and:
  - BGP Unnumbered
  - BFD
  - **BGP Route Summarizaton a.k.a prefix aggregation**
  - Route Reflector
  - EVPN (new!)



Before Aggregation



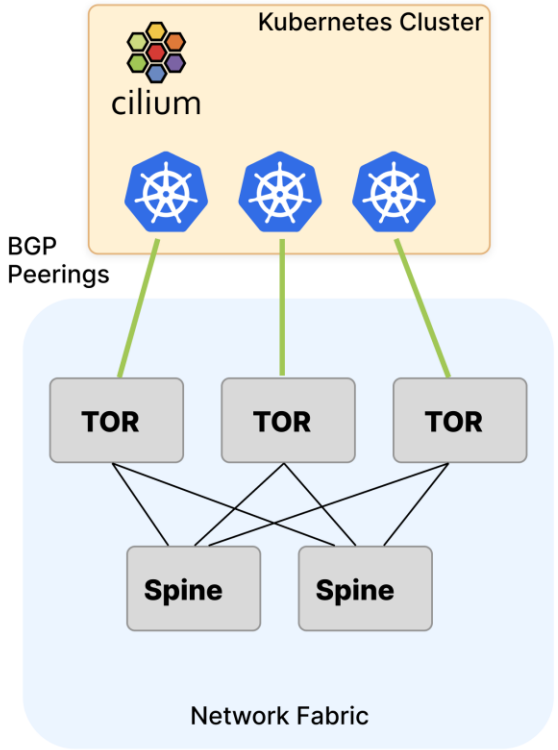
After Aggregation

© 2026 Cisco and/or its affiliates. All rights reserved.

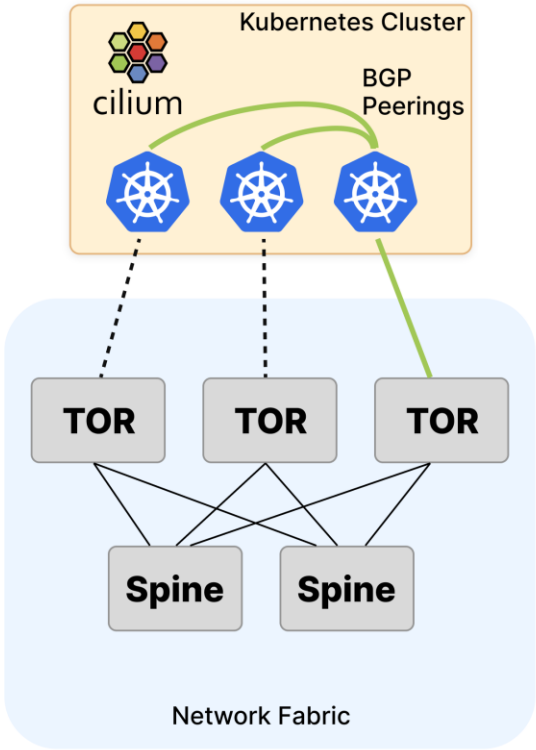
Cilium BGP

Route Reflector

- BGP route reflectors remove the need for a full mesh of node peering.
- Selected nodes act as reflectors and redistribute routes for others
- This reduces the total number of BGP sessions in the cluster.
- Routing control is separated from data plane workloads.
- Clusters peer with a small set of BGP speakers.
- The model matches established data centre and cloud routing practices.



Without BGP Route Reflector

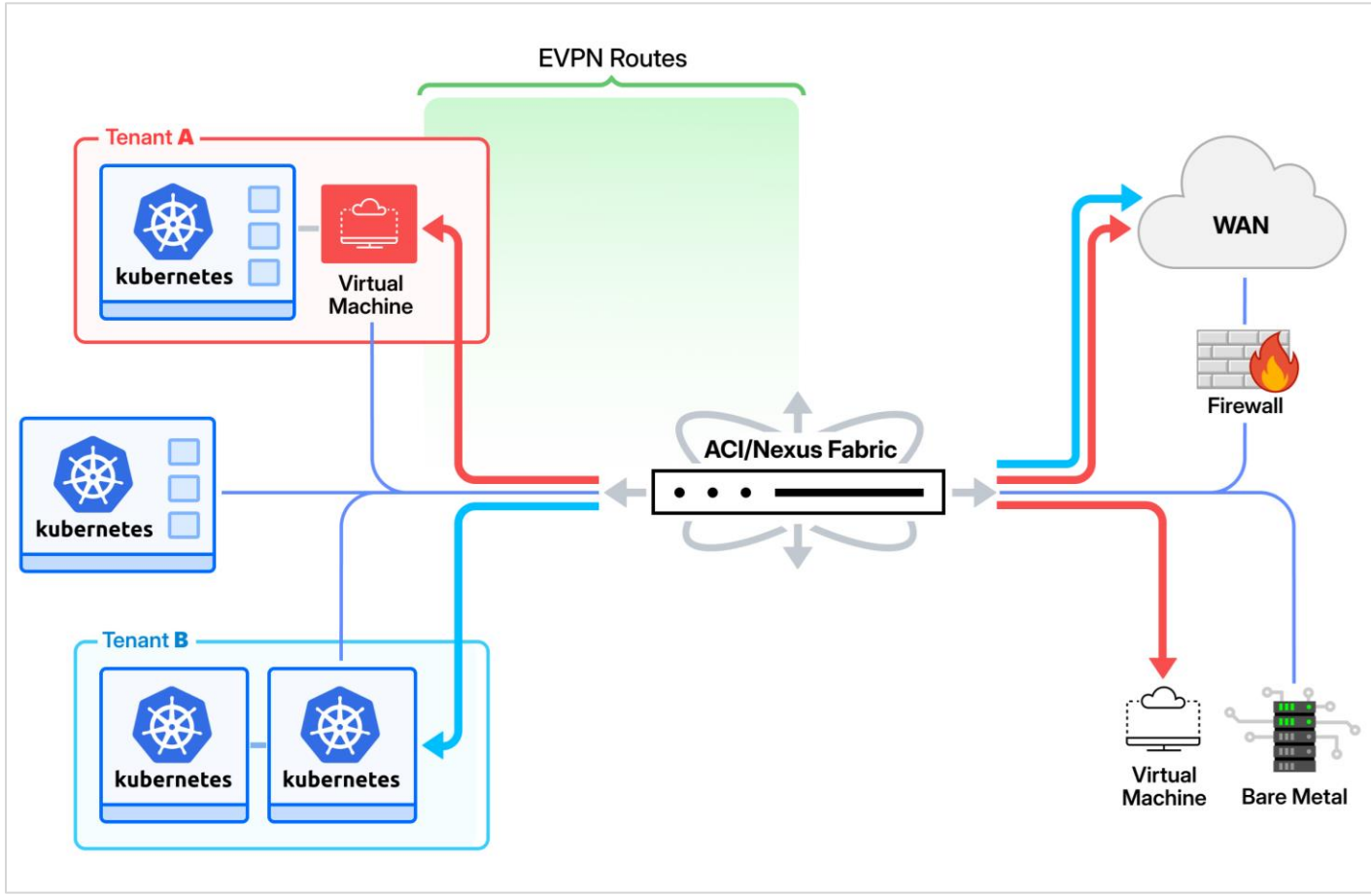


With BGP Route Reflector

© 2026 Cisco and/or its affiliates. All rights reserved.

New: EVPN Support

- **Fabric Integration**: Cilium nodes peer directly with DC Fabric Border Gateways via BGP EVPN.
- **Transport**: Native VXLAN EVPN transport between Kubernetes clusters and the fabric.
- **Routing**: Automatic route propagation through the EVPN control plane and support for multiple VRFs in both Isovalent and DC Fabrics.
- **BGP Efficiency**: A single BGP EVPN peering session manages multiple VRFs, reducing architectural complexity.
- **Security:** Identity-based network policies enforced at the edge, enabling microsegmentation across cluster and fabric boundaries.

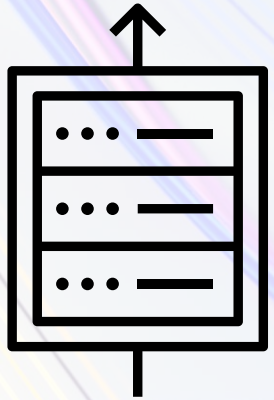


© 2026 Cisco and/or its affiliates. All rights reserved.

Cilium BGP Demo

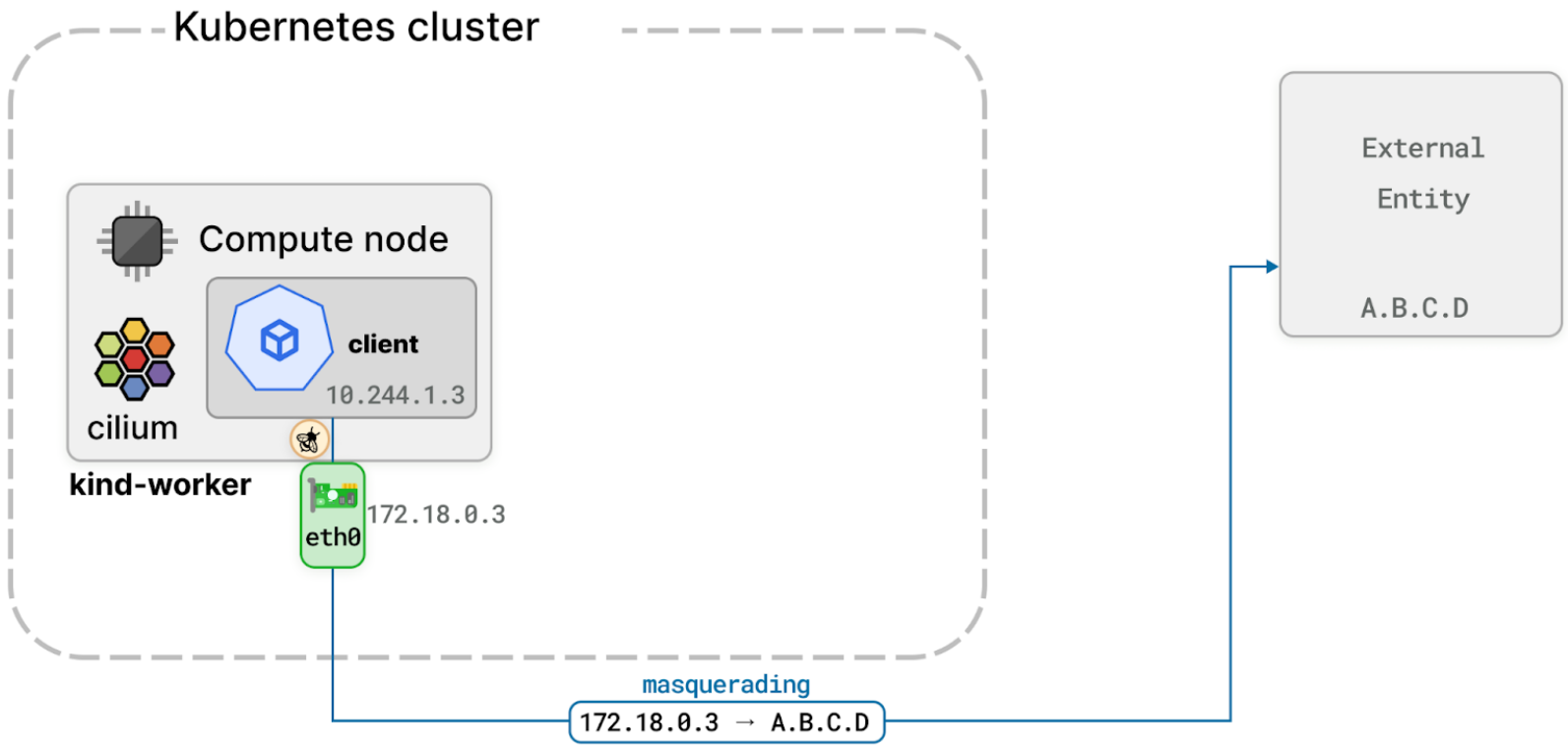


Cluster Egress



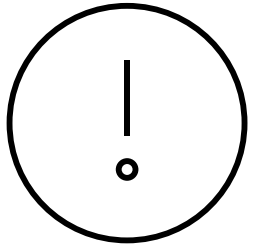
How do pods access external resources?

- Kubernetes isolates workloads using internal pod networks that external systems cannot reach.
- PodCIDR ranges are private so external entities usually lack return paths.
- Nodes use routable addresses so external systems can always reach them reliably.
- **Cilium therefore masquerades outbound pod traffic using the node address by default.**
- Masquerading performs SNAT (many to one) that hides pod sources behind nodes.

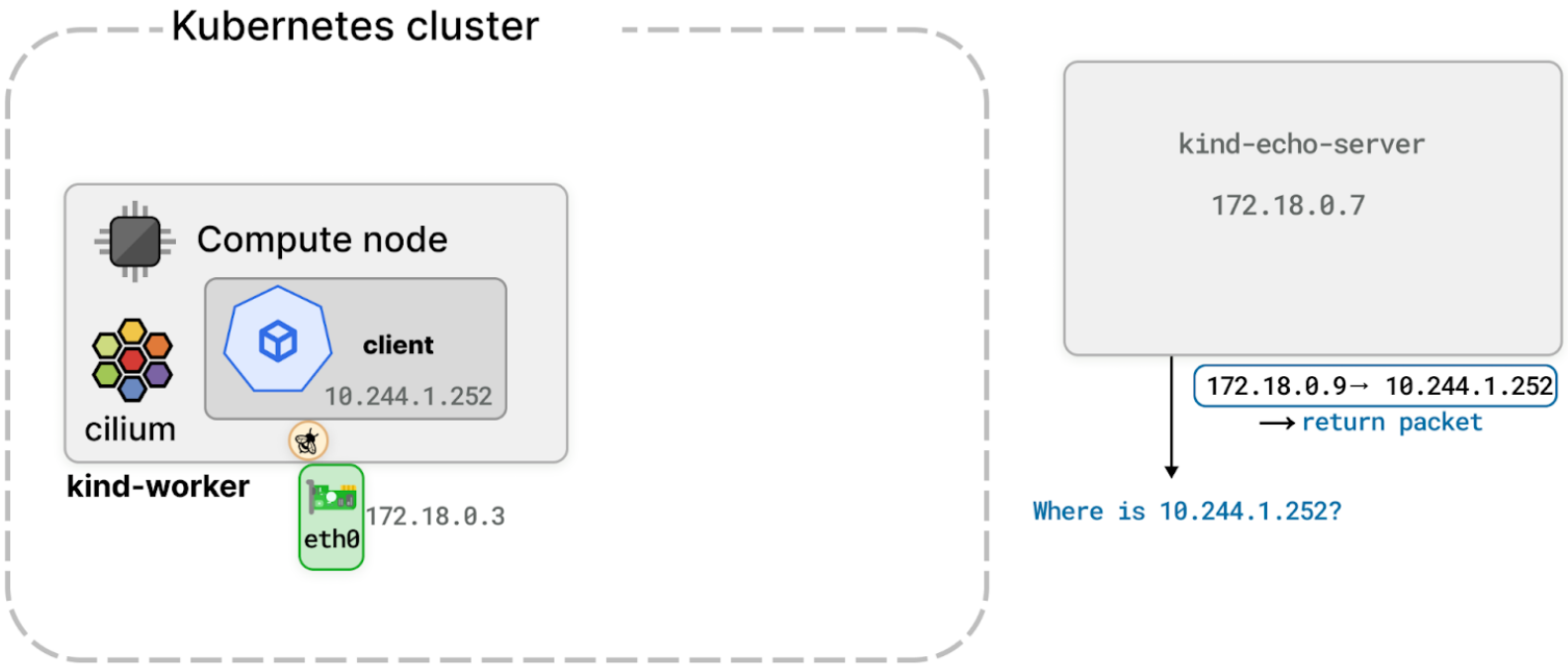


© 2026 Cisco and/or its affiliates. All rights reserved.

What if we disable Masquerading?



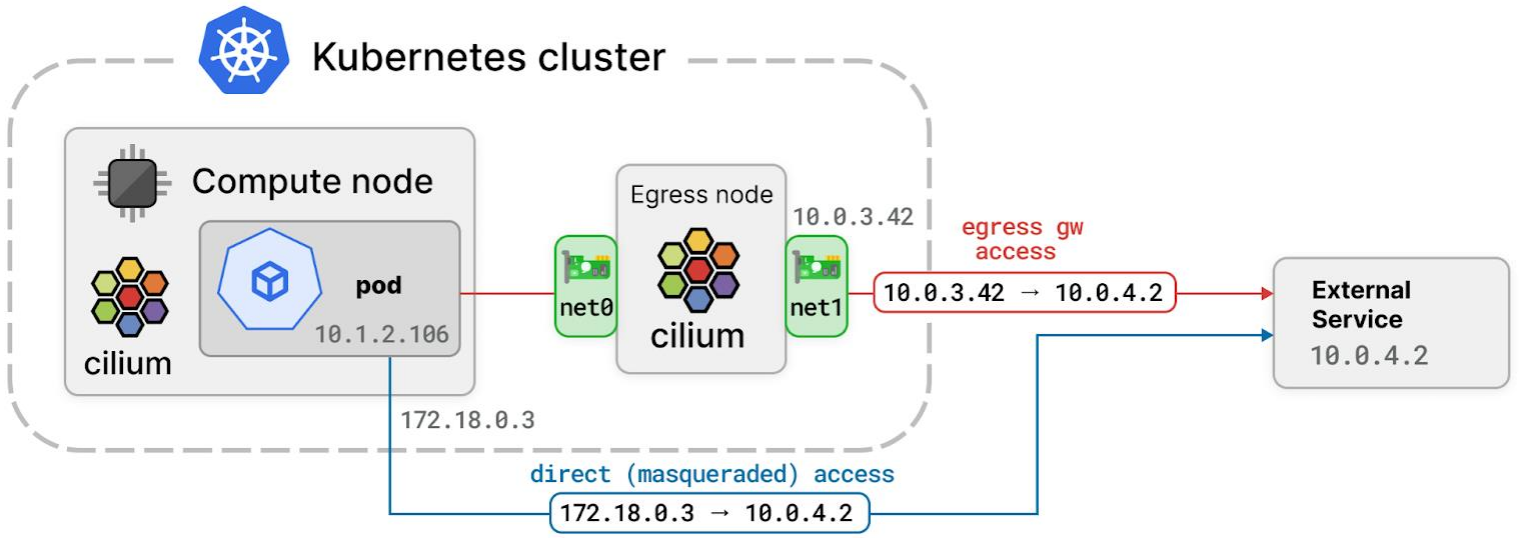
- Without masquerading replies fail because external systems lack routes to pods.
- Disabling masquerading works only when pod networks are fully routable externally.
- Many environments require masquerading so return traffic reliably reaches the cluster.



© 2026 Cisco and/or its affiliates. All rights reserved.

Egress Gateway

- Egress Gateway assigns predictable fixed source addresses to selected pod traffic leaving the cluster.
- Pods are matched by labels or namespaces and routed through designated gateway nodes.
- Matching outbound packets are masqueraded with stable egress IPs instead of pod or node addresses.
- This preserves identity across node changes and avoids unpredictable pod or node source behaviour.
- External systems see consistent trusted sources which improves auditing tracing and external policy control.

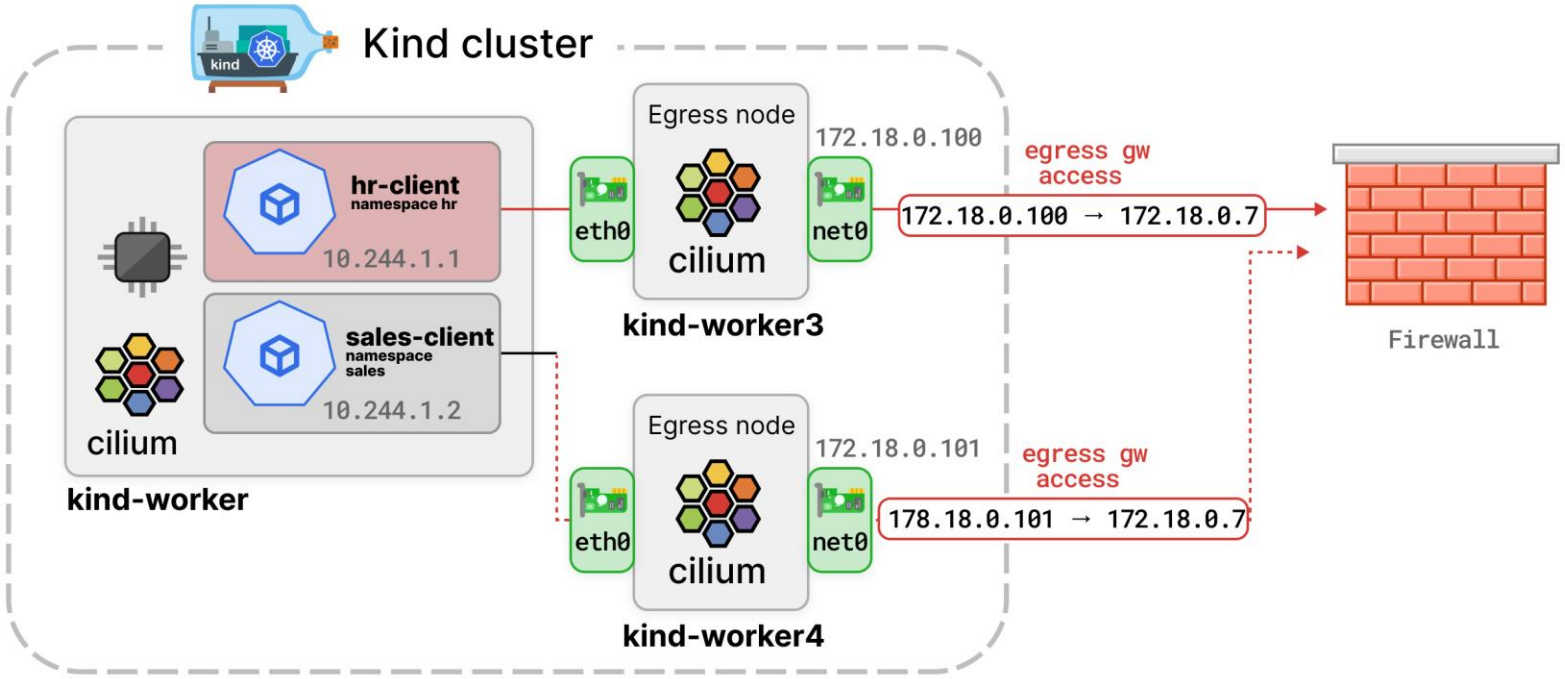


© 2026 Cisco and/or its affiliates. All rights reserved.

Egress Gateway Use Case

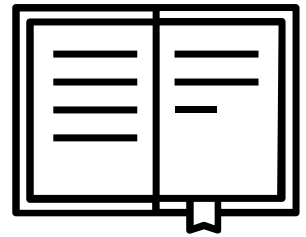
Firewall Integration

- Without EGW, external firewalls see incoming traffic originated from the cluster and cannot tell which workload/tenant was the source.
- **Firewalls lack Kubernetes context.**
- With EGW, traffic will leave the cluster with predictable per-tenant IPs and firewall operators can set up FW rules accordingly.



© 2026 Cisco and/or its affiliates. All rights reserved.

Egress Gateway Policy



- HR namespace egress control via label selector
- Destination CIDR filtering (172.18.0.0/16)
- Fixed egress IP (172.18.0.100) assignment
- Node-specific gateway with label matching

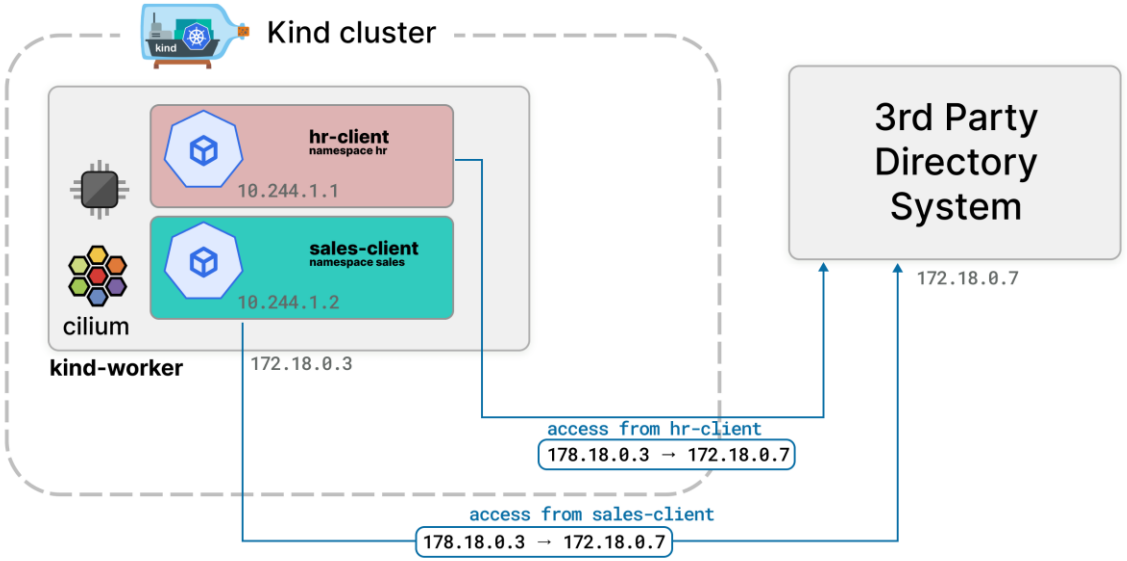
```yaml
apiVersion: cilium.io/v2
kind: CiliumEgressGatewayPolicy
metadata:
  name: hr-egress
spec:
  destinationCIDRs:
  - "172.18.0.0/16"
  selectors:
  - namespaceSelector:
      matchLabels:
        name: hr
  egressGateway:
    nodeSelector:
      matchLabels:
        egress-gw : "hr"
    egressIP: 172.18.0.100
```

© 2026 Cisco and/or its affiliates. All rights reserved.

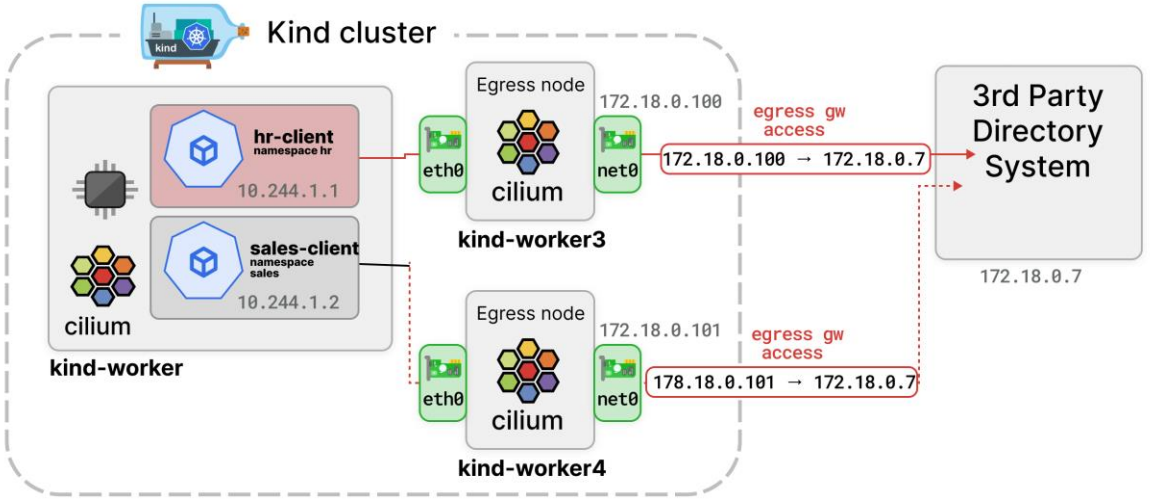
cisco

Egress Gateway

Recap



Default Masquerading



With Egress Gateway

© 2026 Cisco and/or its affiliates. All rights reserved.

Egress Bandwidth Control

- Egress control also requires preventing pods from overwhelming limited outbound network capacity.
- Cilium **Bandwidth Manager** applies per pod limits to avoid saturation and reduce packet loss.
- In multi tenant clusters it complements Egress Gateway by stopping noisy neighbour behaviour.
- It operates at the node level using Kubernetes annotations:

```
$ kubectl annotate pod/netperf-server kubernetes.io/egress-bandwidth="10M"
pod/netperf-server annotated
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Bandwidth Manager

Without Bandwidth Manager:

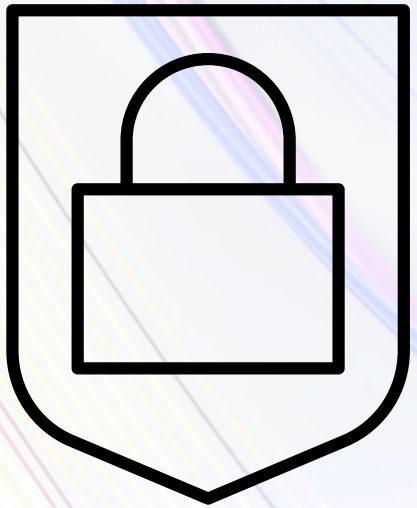
```
$ kubectl exec netperf-client -- netperf -t TCP_MAERTS -H "${NETPERF_SERVER_IP}"
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

131072  16384  16384    10.00    8827.69
```

With Bandwidth Manager, when applying a 10M egress limitation to a pod:

```
$ kubectl exec netperf-client -- netperf -t TCP_MAERTS -H "${NETPERF_SERVER_IP}"
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

131072  16384  16384    10.00       9.54
```

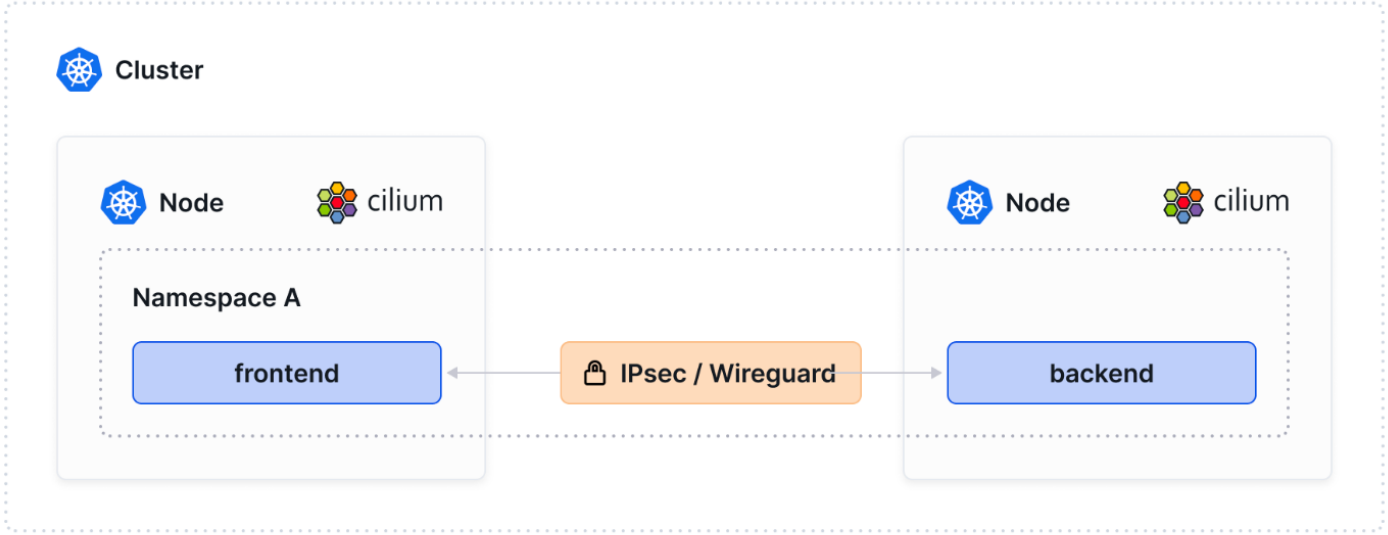
© 2026 Cisco and/or its affiliates. All rights reserved.

Encryption



Cilium Built-in Encryption

- Cilium provides transparent encryption for inter-node traffic.
- Workloads require no changes and remain unaware entirely.
- Encryption uses WireGuard or IPsec.
- Especially useful in:
  - Cloud/shared environments
  - Private DCs without hardware-based encryption (MACsec) enabled
  - Any environment where regulatory or compliance requirements demand in transit encryption.



© 2026 Cisco and/or its affiliates. All rights reserved.

Encryption Options

Encryption Options

IPsec

Cilium IPsec enables transparent encryption for all inter-node traffic.

IPsec requires manual key setup and algorithm choices yet suits environments needing strict cryptographic controls mandated by compliance.

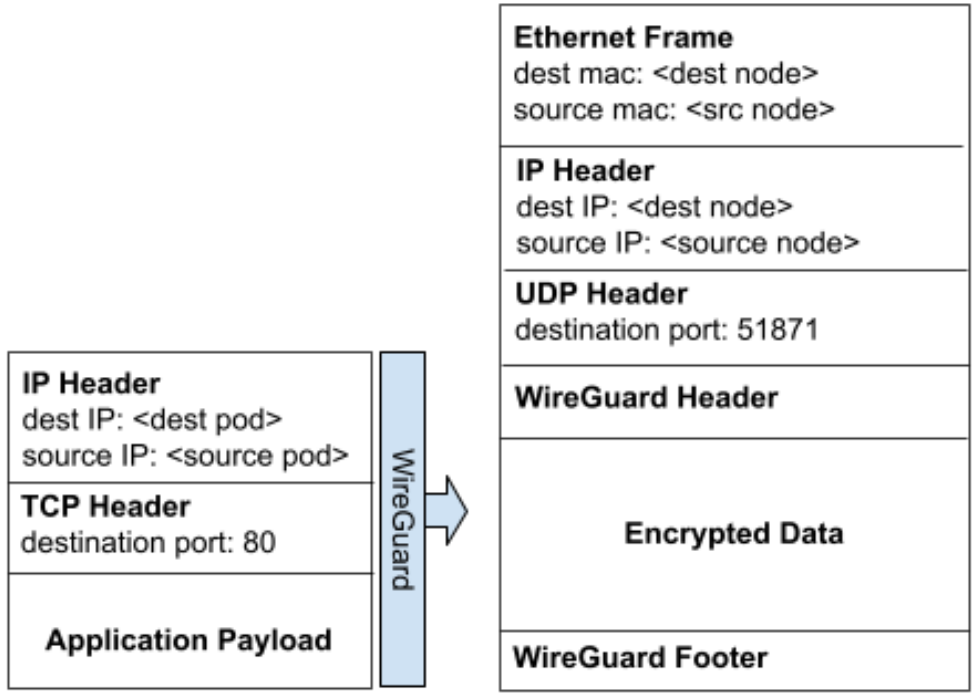
WireGuard

Cilium WireGuard provides transparent encryption using streamlined automatic key exchange and modern cryptography.

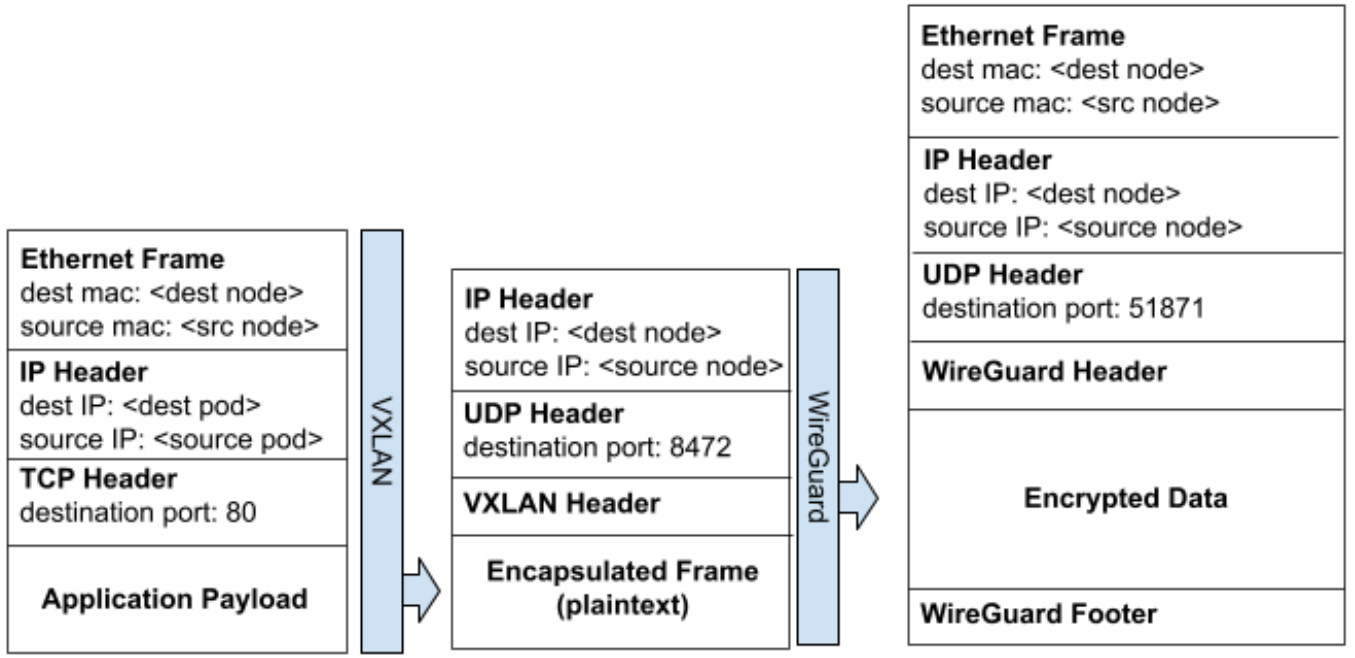
© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Encryption & Routing Modes



Encryption and encapsulation of a packet using native routing

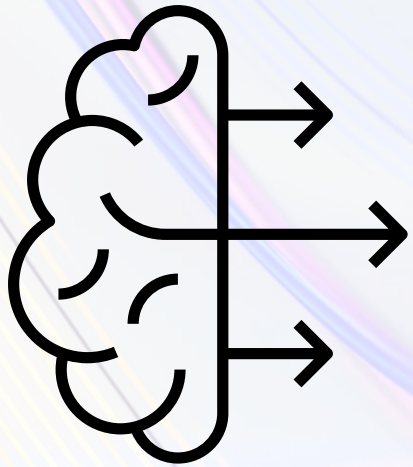


Encryption and encapsulation of a packet using VXLAN tunneling

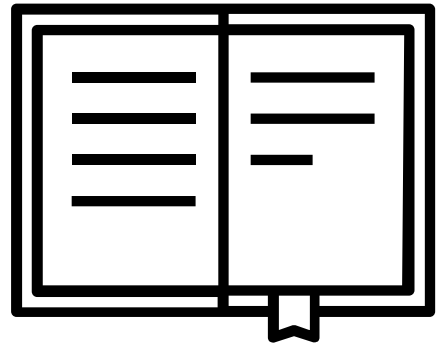
© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

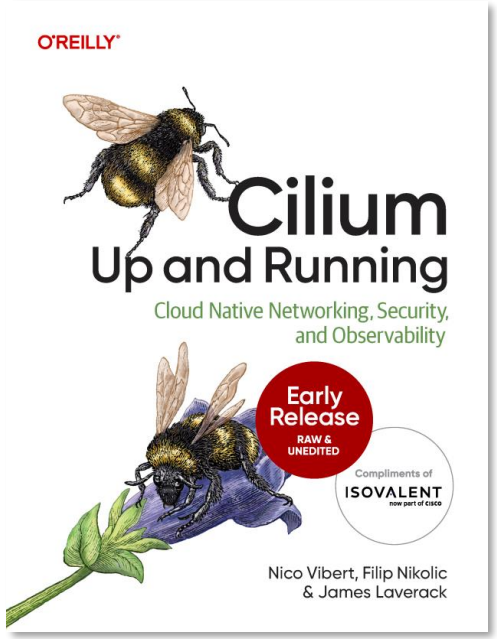
Traffic Engineering



Traffic Engineering & Performance Features



- **Service Traffic Distribution** keeps traffic inside preferred areas reducing latency and network costs.
- **Local Redirect Policy** selects backends on the same node as the source to avoid cross node hops entirely.
- **Direct Server Return** preserves client sources and returns responses directly for improved efficiency.
- **XDP** processes packets early in the kernel and lowers overhead for busy nodes.
- **Maglev** and **netkit** improve stability and throughput through consistent hashing and faster container paths.

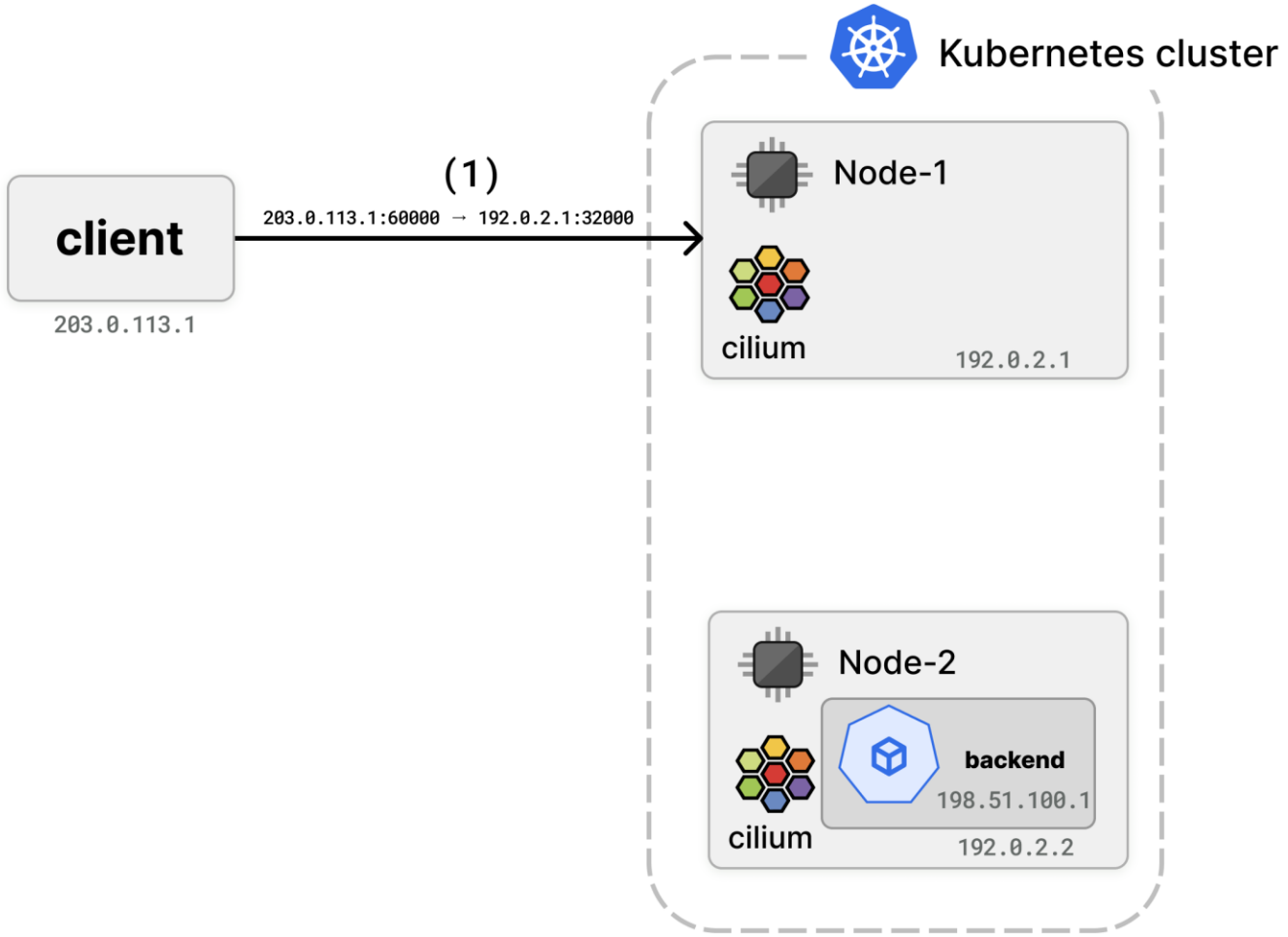


To learn more about performance features, **read Chapter 8.**

© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

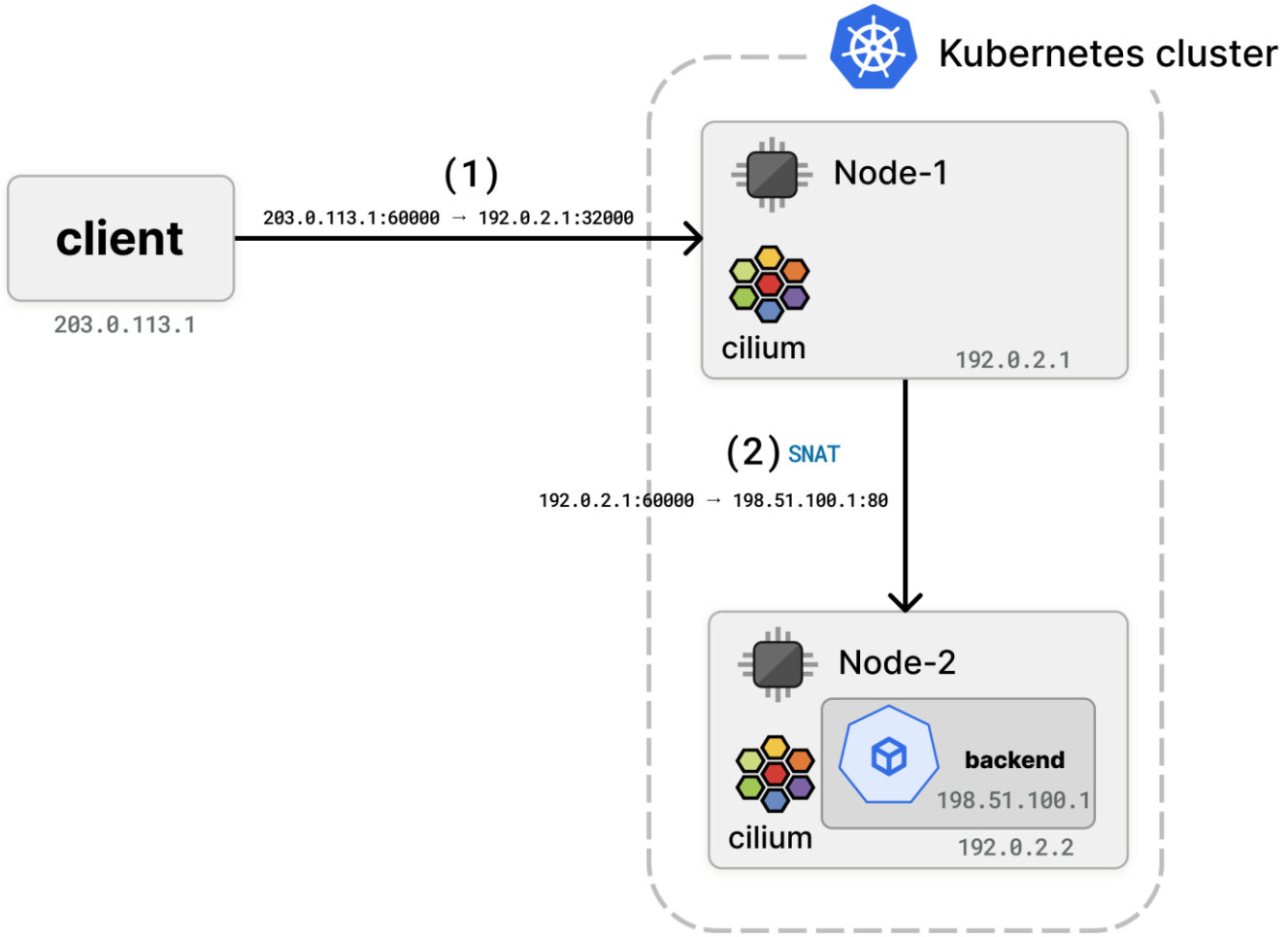
Indirect Server Return



© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

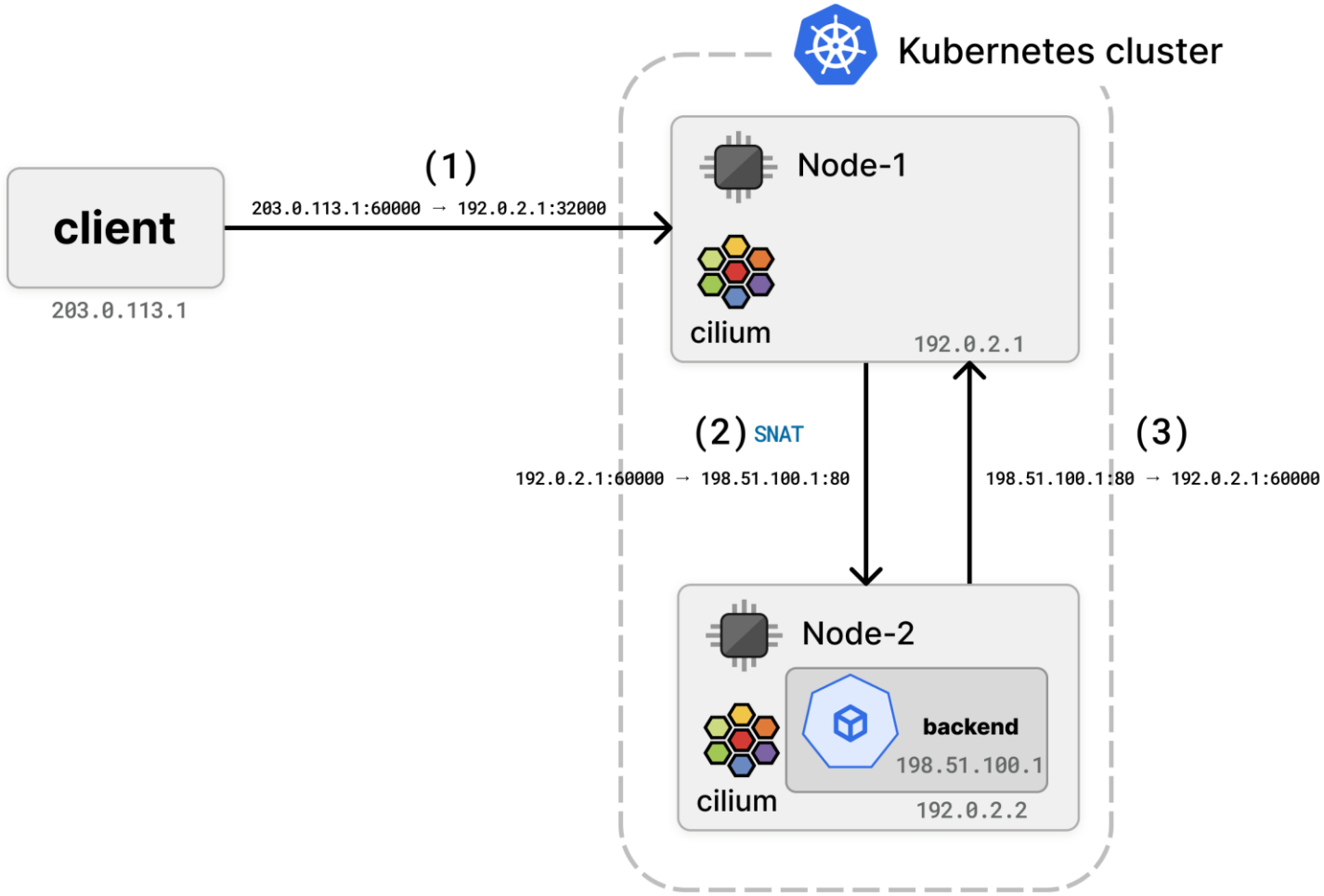
Indirect Server Return



© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

Indirect Server Return

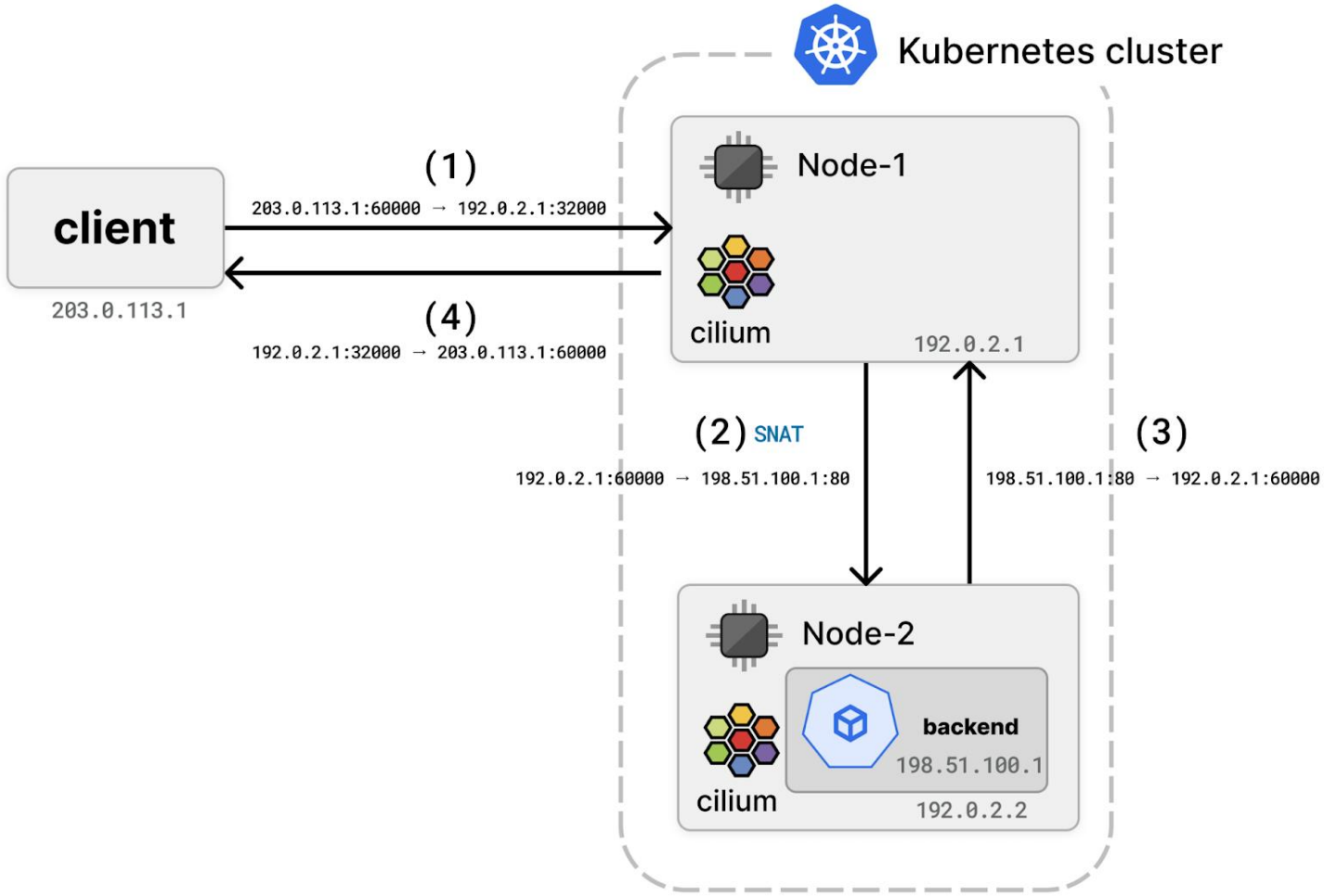


© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

Indirect Server Return

- By default, when a service is exposed externally (through NodePort or LoadBalancer) and where the service endpoint runs on a different node (Node-2) than the node the client connects to (Node-1), return traffic to the client is **indirect**.
- When Node-1 forwards the client request to Node-2, it would Source-NAT traffic, replacing the client source IP address with its internal IP from its own PodCIDR.
- This has two negative implications:
  - **Loss of observability**: the backend receiving the traffic sees the packet as originating from Node-1, instead of the actual client.
  - **Indirect return path**: the backend traffic is sent to Node-1 and then onto the client, instead of directly to the client.

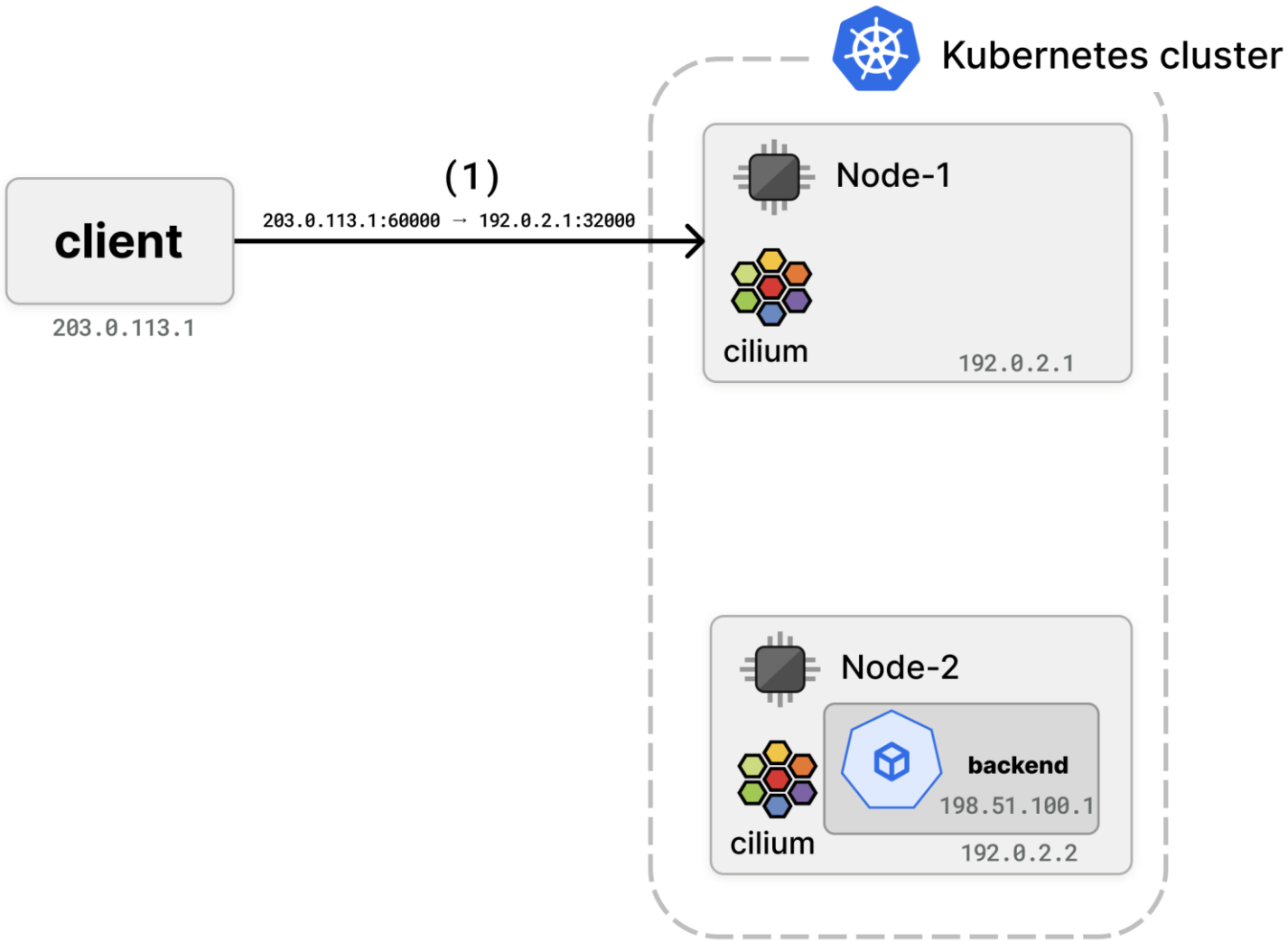


© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

DSR improves the performance of traffic returning back to the client.

Cilium's Direct Server Return leverages eBPF to preserve the source IP address and to ensure traffic is returned directly to the client.

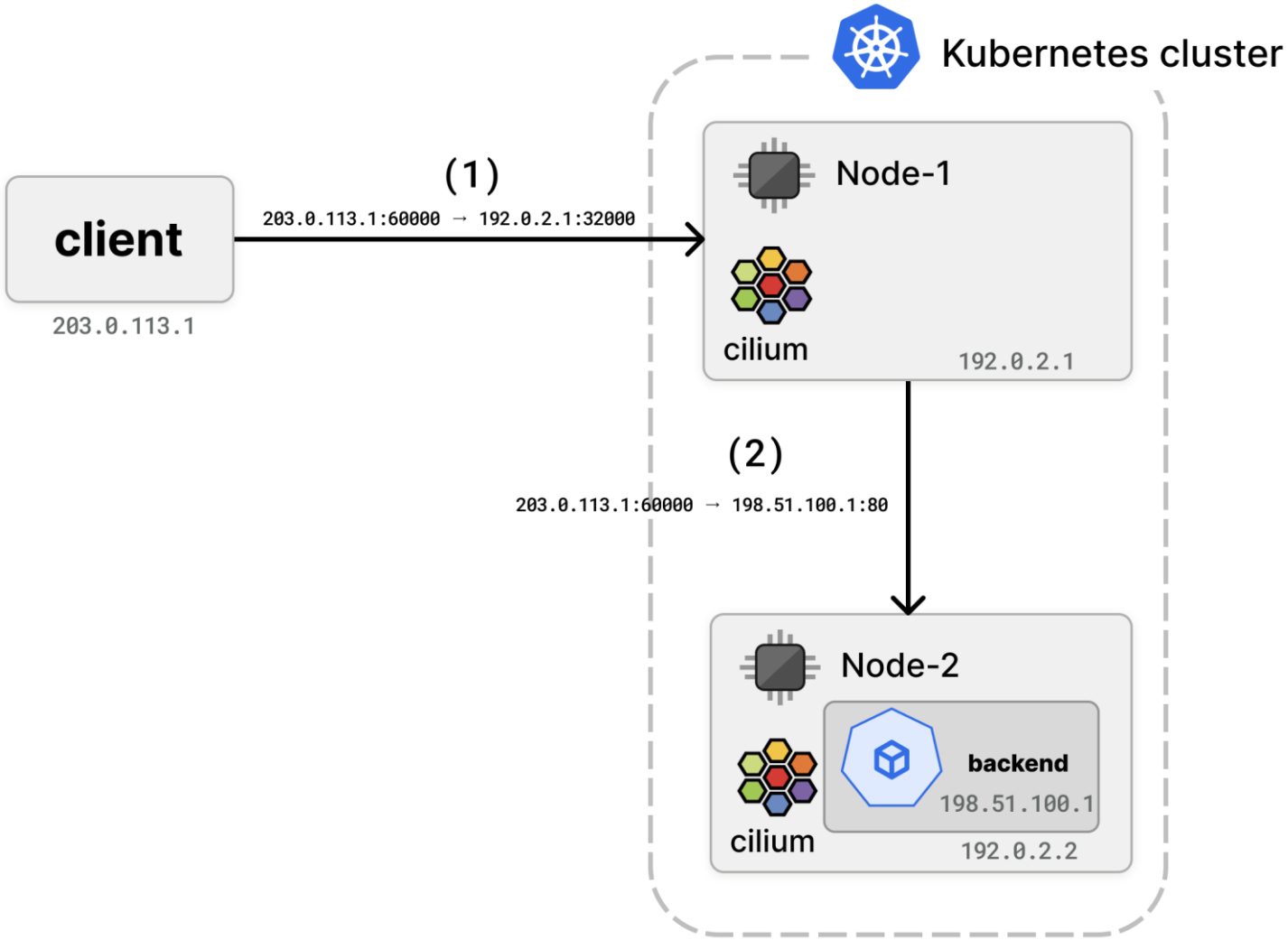


© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

DSR improves the performance of traffic returning back to the client.

Cilium's Direct Server Return leverages eBPF to preserve the source IP address and to ensure traffic is returned directly to the client.

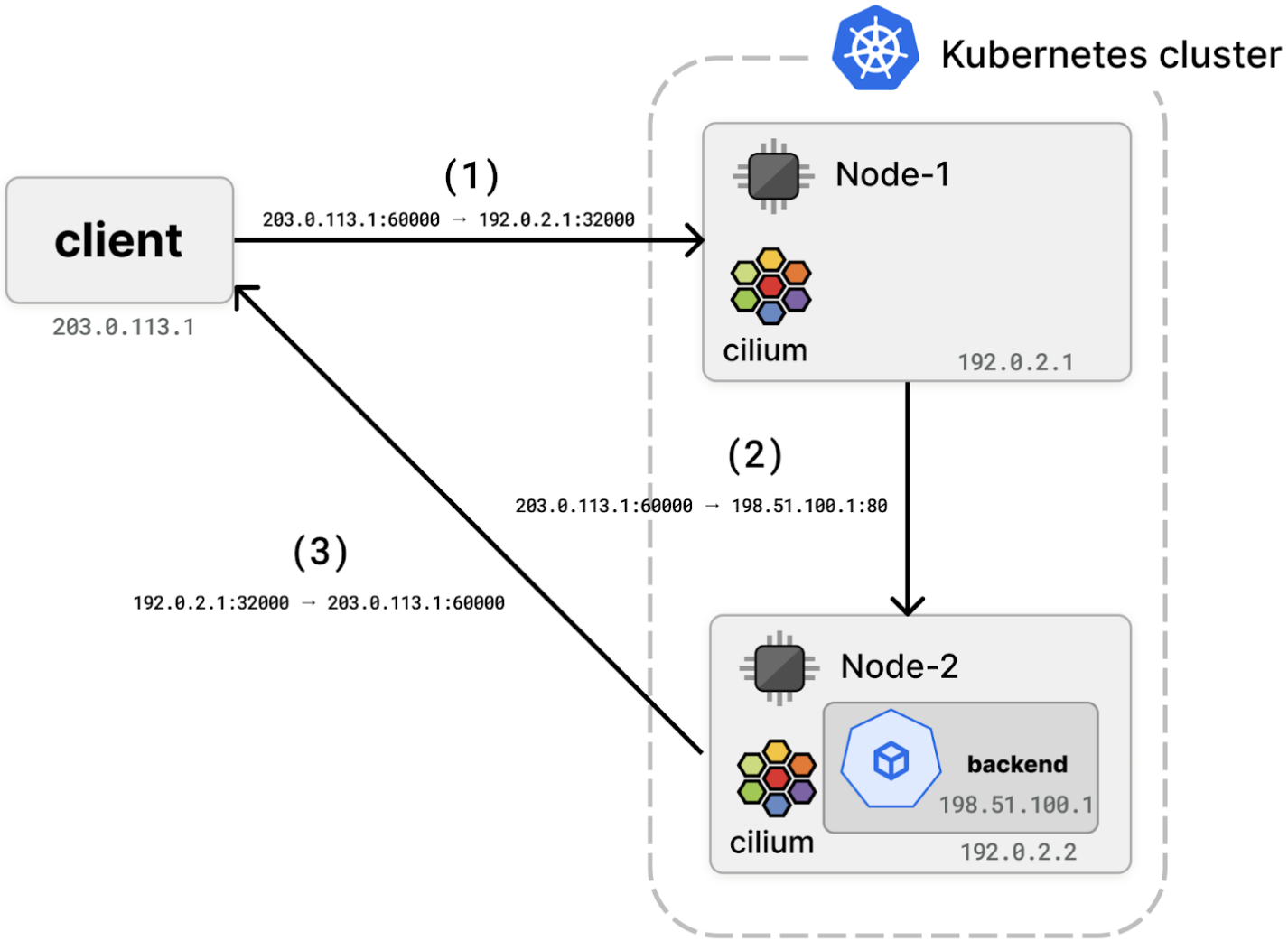


© 2026 Cisco and/or its affiliates. All rights reserved.

Direct Server Return (DSR)

DSR improves the performance of traffic returning back to the client.

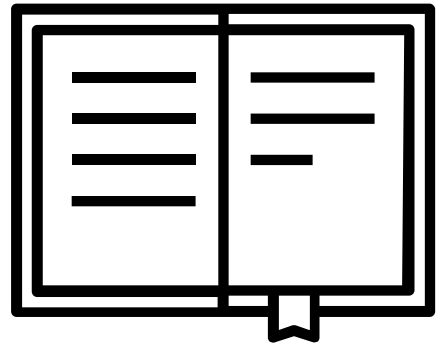
Cilium's Direct Server Return leverages eBPF to preserve the source IP address and to ensure traffic is returned directly to the client.



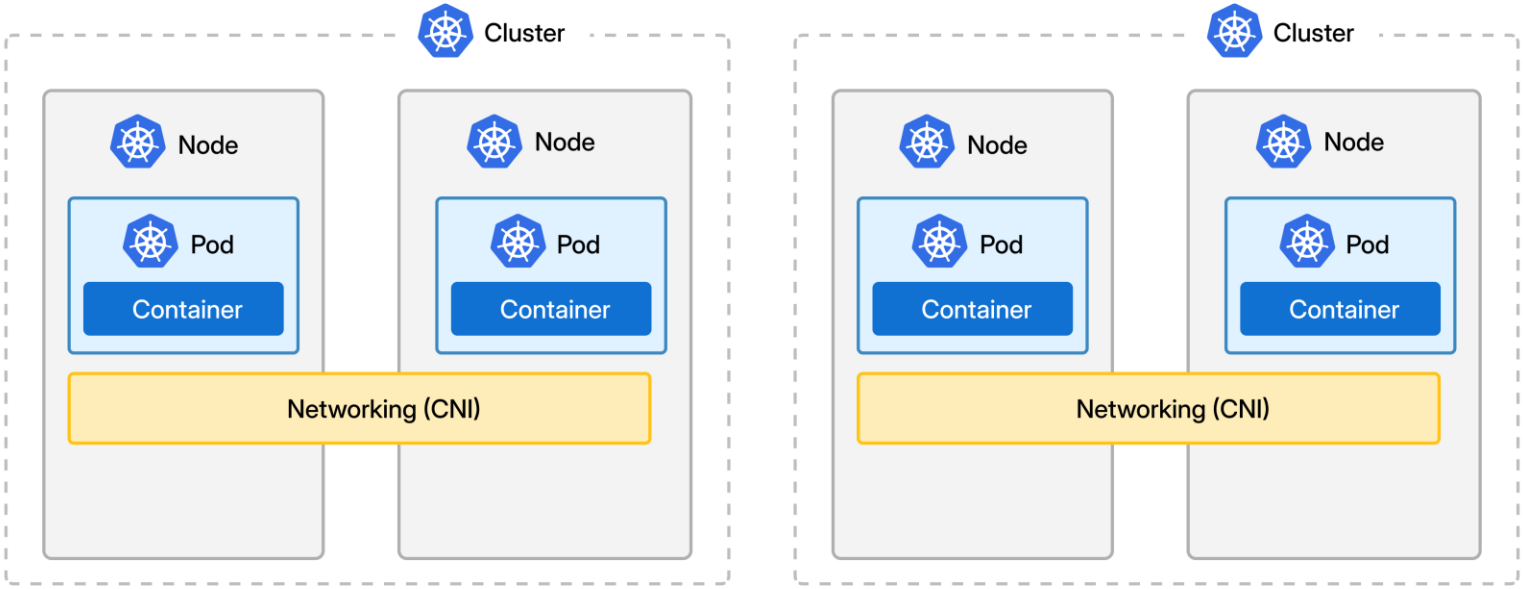
© 2026 Cisco and/or its affiliates. All rights reserved.

Multi Cluster

Multi-Clustering

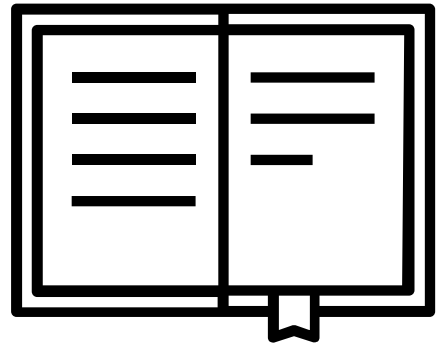


- Organizations adopt multiple clusters for scale resilience organizational boundaries and regulatory requirements.
- Large environments hit Kubernetes scale limits making single clusters impractical.
- Multi region or multi cloud deployments improve availability and fault tolerance.
- Workloads still need secure reliable communication across cluster boundaries.
- Networking security and observability become harder once clusters are separated.



© 2026 Cisco and/or its affiliates. All rights reserved.

Multi-Clustering Challenges

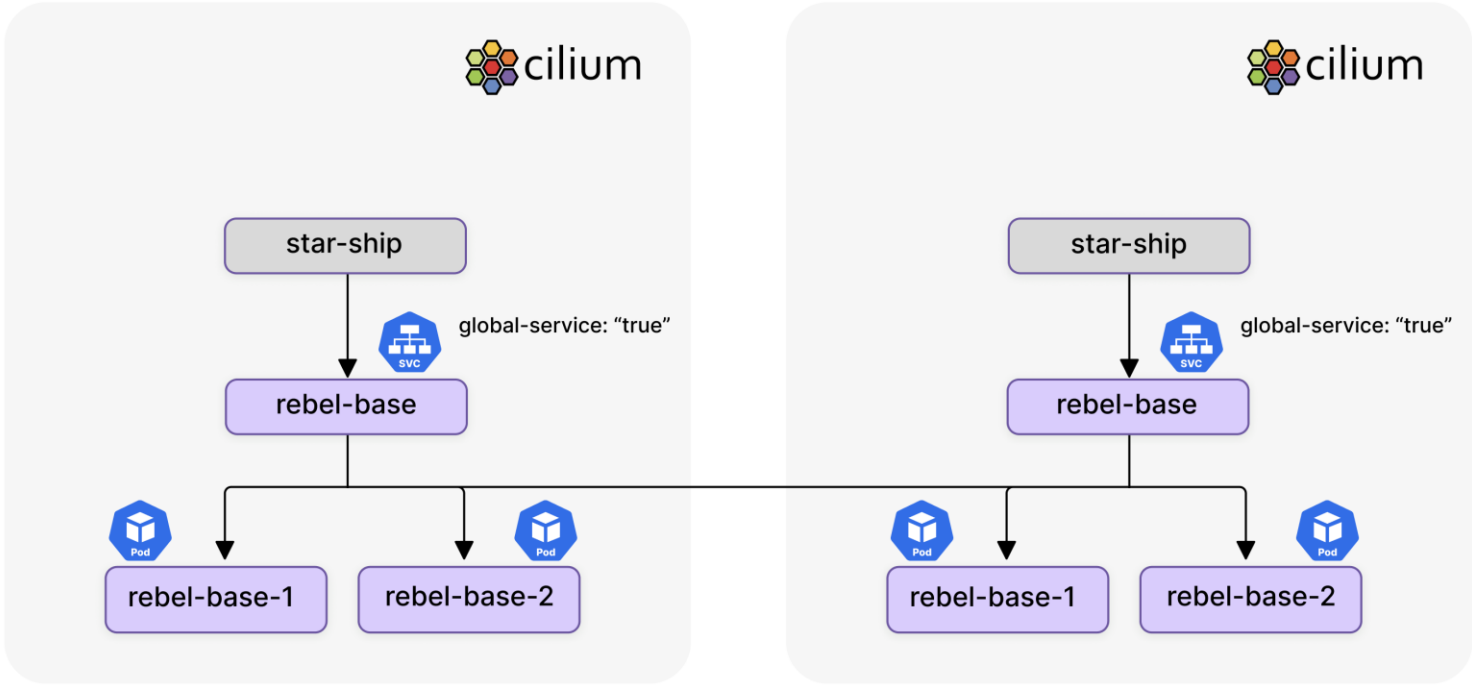


- Clusters are treated as independent and services exposed using load balancers or gateways.
- Remote services appear external, so identity-based policy and observability no longer apply.
- Policies require manual CIDR management which increases complexity and error risk.
- DNS based failover depends on client behaviour and time to live expiration.
- External load balancers add cost operational overhead and additional failure points.

© 2026 Cisco and/or its affiliates. All rights reserved.

Multi-Clustering with Cilium Cluster Mesh

- Mesh clusters and confirm interconnectivity between all participating nodes.
- Configure certificates and trust so clusters authenticate and share identities.
- Enable Cluster Mesh and verify shared service discovery is functioning.
- Annotate the Service as global so endpoints publish across clusters.
- Confirm DNS resolves the Service and traffic fails over between clusters.

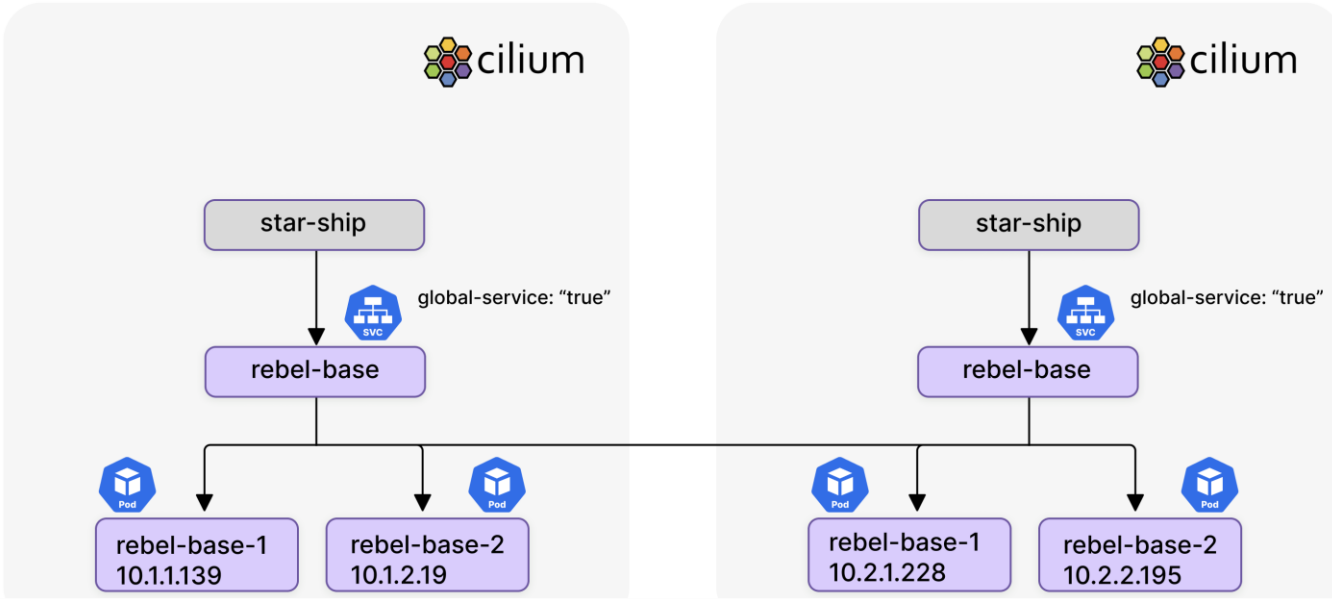


© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Multi-Clustering with Cilium Cluster Mesh

- By default, once a service is shared across all mesh, all backends are served randomly across both clusters.

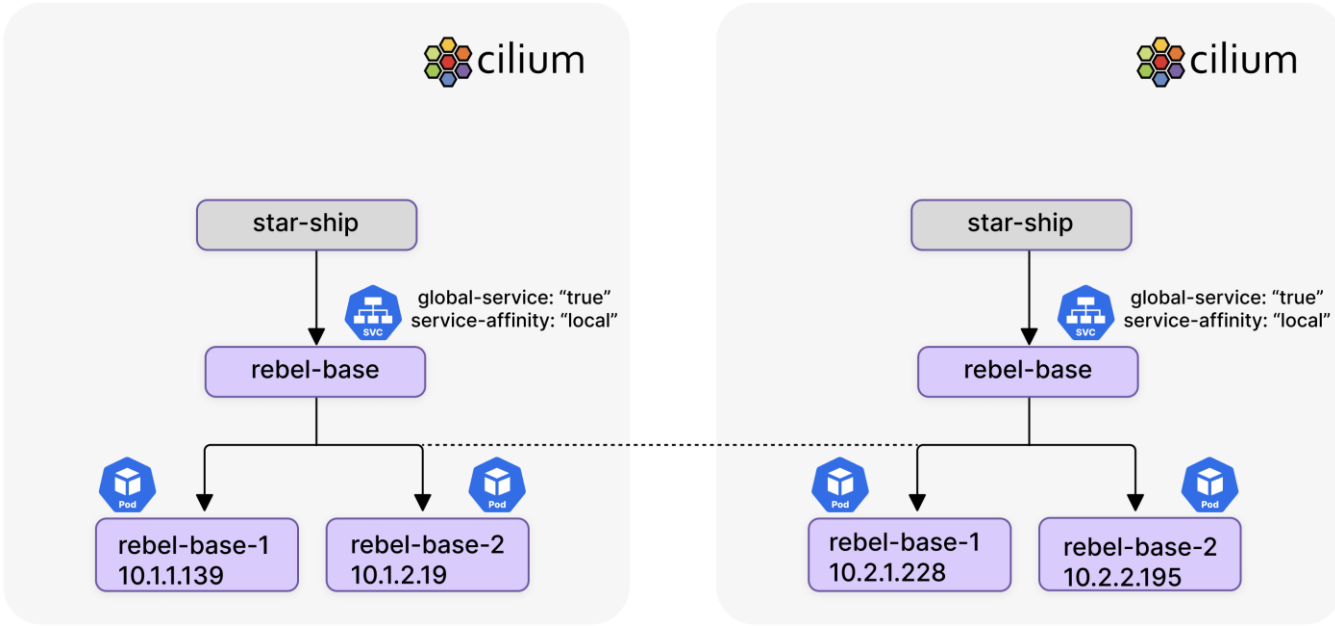



```
Cilium service list from cluster1:
ID   Frontend                Service Type   Backend                                 7
10.11.20.25:80/TCP      ClusterIP      1 => 10.1.1.139:80/TCP (active) (preferred)
                                      2 => 10.1.2.19:80/TCP (active) (preferred)
                                      3 => 10.2.1.228:80/TCP (active) (preferred)
                                      4 => 10.2.2.195:80/TCP (active) (preferred)

Cilium service list from cluster2:
ID   Frontend                Service Type   Backend
7    10.22.147.226:80/TCP    ClusterIP      1 => 10.1.1.139:80/TCP (active) (preferred)
                                            2 => 10.1.2.19:80/TCP (active) (preferred)
                                            3 => 10.2.1.228:80/TCP (active) (preferred)
                                            4 => 10.2.2.195:80/TCP (active) (preferred)
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Multi-Clustering with Cilium Cluster Mesh

- Affinities can be set to direct traffic to either the **local** backends or to the **remote** ones.

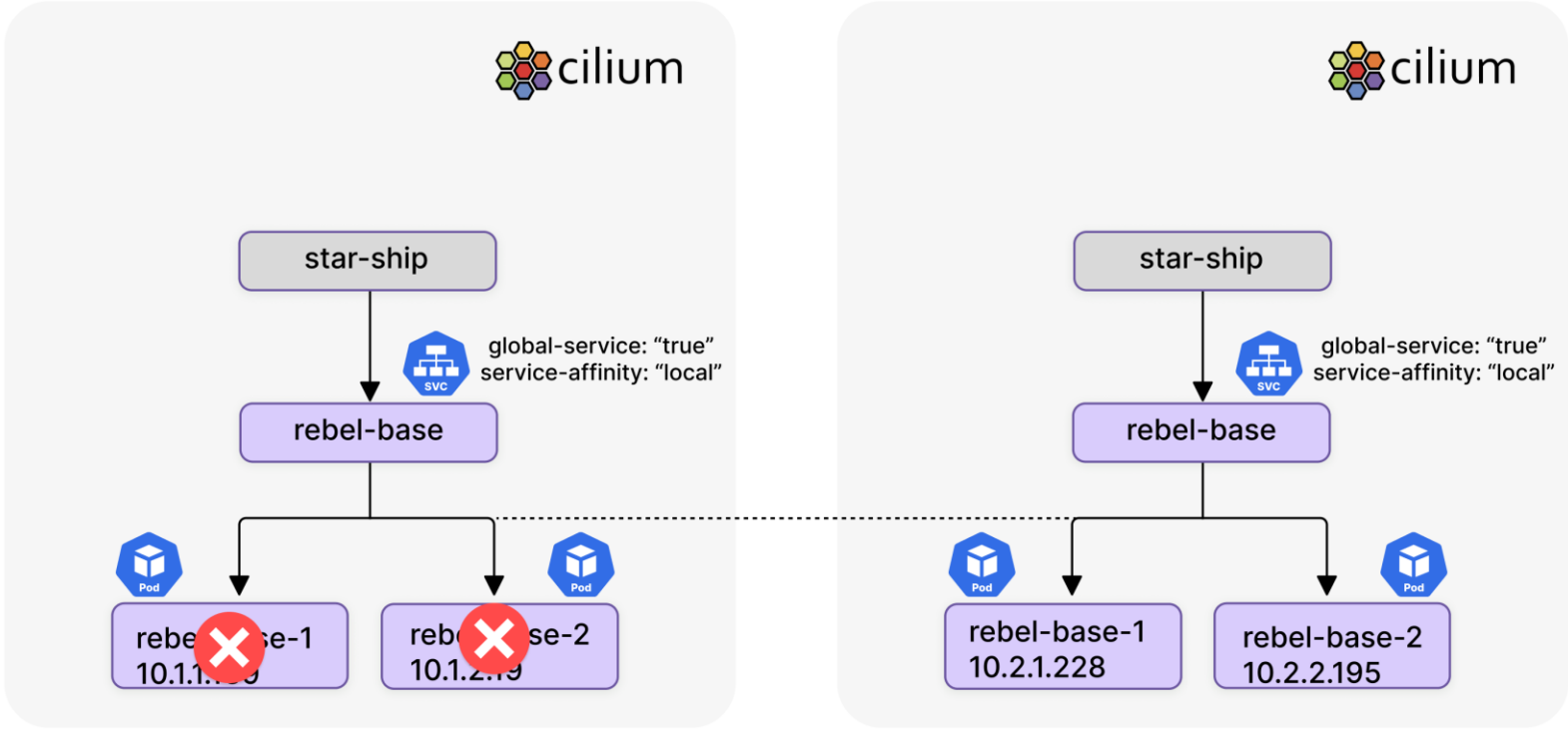



```
Cilium service list from cluster1 (only show LOCAL affinity):
ID   Frontend              Service Type   Backend
7    10.11.20.25:80/TCP    ClusterIP      1 => 10.1.1.139:80/TCP (active) (preferred)
                                          2 => 10.1.2.19:80/TCP (active) (preferred)
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Multi-Clustering with Cilium Cluster Mesh

- If an outage affects all local/remote backends, Cluster Mesh will direct the traffic to the “less-preferred” site.

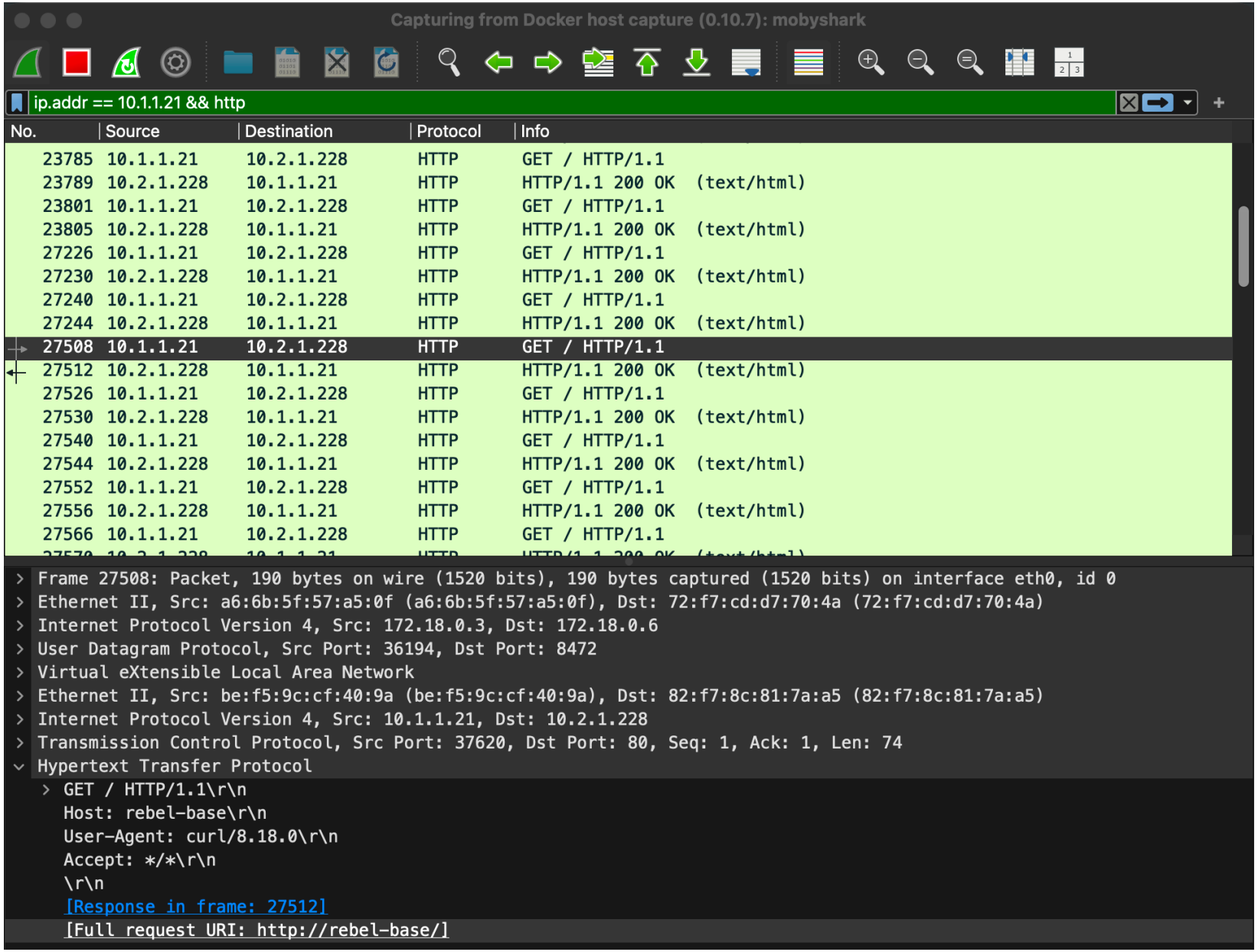


© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Life of a Cluster Mesh Packet

- **star-ship** client pod initiates curl request to global service **rebel-base**.
- Client pod IP 10.1.1.21 runs in cluster-1.
- Destination pod IP 10.2.1.228 lives in cluster 2.
- Cluster Mesh resolves the global service to a remote endpoint.
- Traffic is encapsulated and sent node to node across clusters.
- Outer headers show node addresses while inner pod addresses remain unchanged.
- Application sees simple HTTP despite crossing cluster boundaries transparently.



© 2026 Cisco and/or its affiliates. All rights reserved.

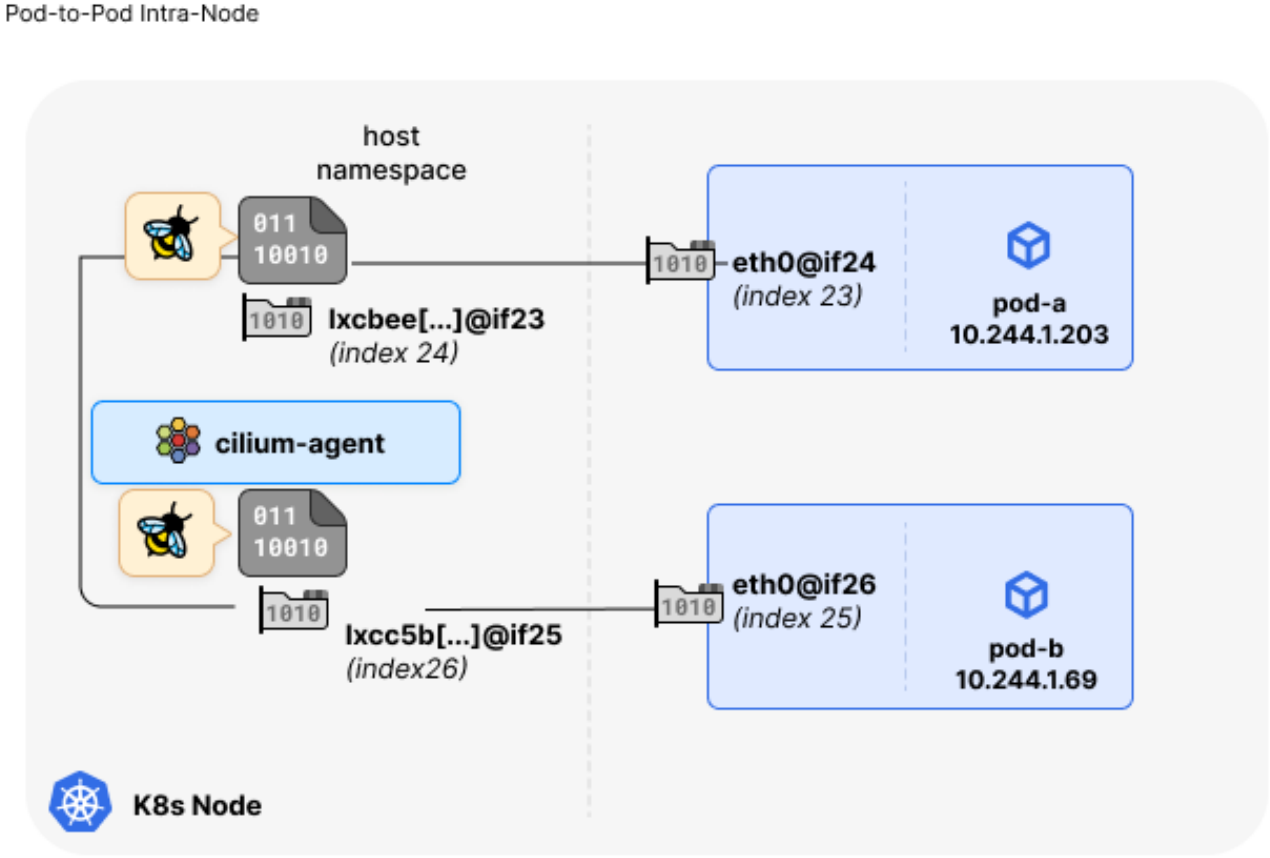
Intra-node connectivity

Intra-node pod-to-pod

Virtual Ethernet devices come in pairs.

Packets sent into one end of the pair appear on the other end.

Most CNIs use veth pairs to connect pod namespaces to the host's network namespace. Inside the pod, you see an interface such as `eth0`; on the host, you see its peer, usually named `lxc*`.

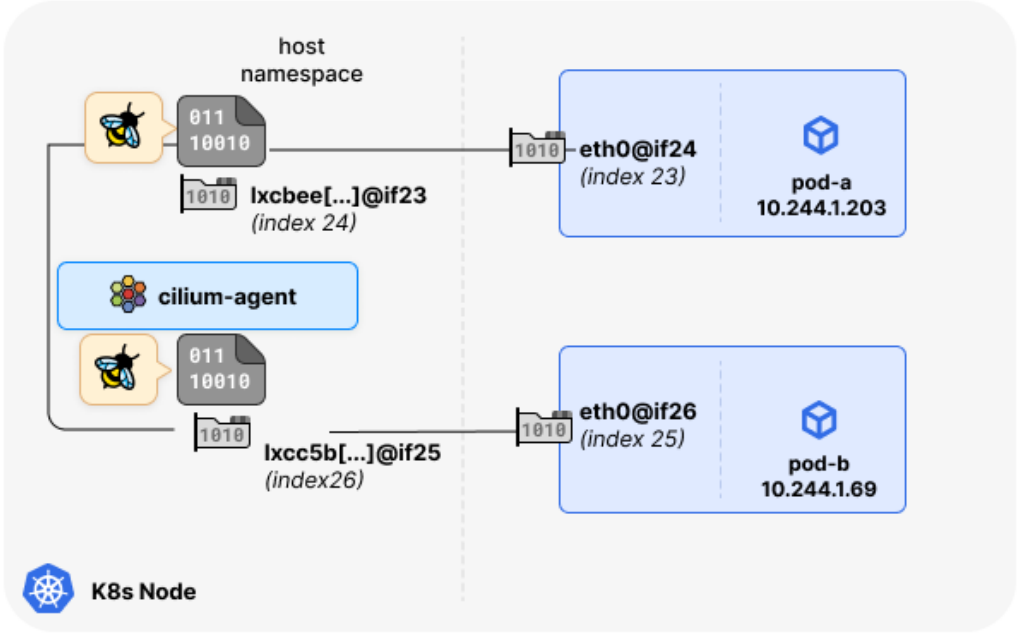


© 2026 Cisco and/or its affiliates. All rights reserved.

Intra-node pod-to-pod

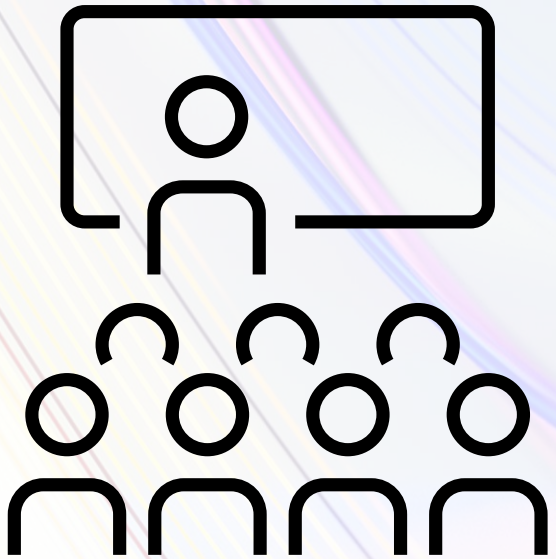
```
## on pod-b
pod-b:~# ip address show eth0
25: eth0@if26: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc noqueue
    state UP group default qlen 1000
    link/ether 46:9c:c0:5f:ec:1c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.244.1.69/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::449c:c0ff:fe5f:ec1c/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever

## on the host

root@kind-worker:/# ip address | grep -A 3 if25
26: lxcc5b0a1b7ddfb@if25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520
    qdisc noqueue state UP group default qlen 1000
    link/ether fa:a2:c6:90:65:f7 brd ff:ff:ff:ff:ff:ff
      link-netns cni-f4803bdf-3eb6-a209-831e-b93beac138a3
    inet6 fe80::f8a2:c6ff:fe90:65f7/64 scope link
       valid_lft forever preferred_lft forever
```

Pod-to-Pod Intra-Node



© 2026 Cisco and/or its affiliates. All rights reserved.

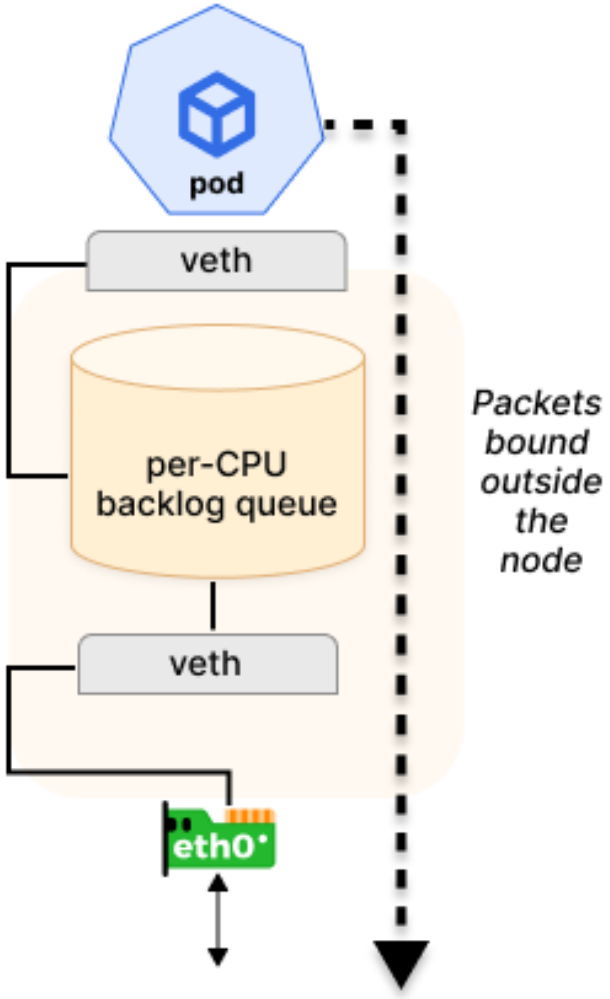
Demo



Virtual Ethernet (veth)

Limitations

- Veth has been around for 25+ years
- Has some inner limitations which causes performance issues:
  - Veth relies on L2 and performs ARP resolutions
  - For traffic bound outside the node, traffic switch to host network namespaces, is processed and then sent on to network interface for transmission.

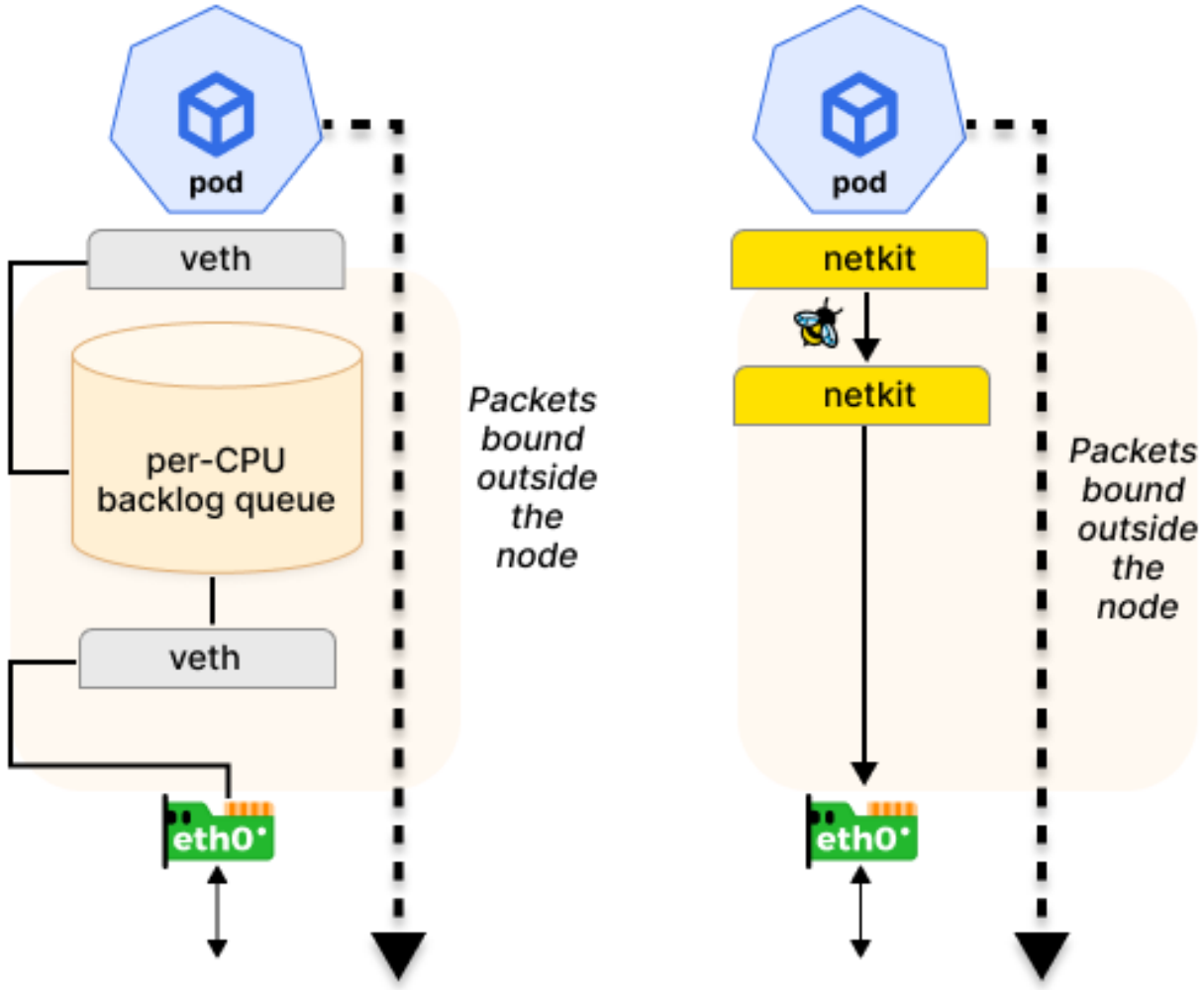


© 2026 Cisco and/or its affiliates. All rights reserved.

netkit

The veth successor

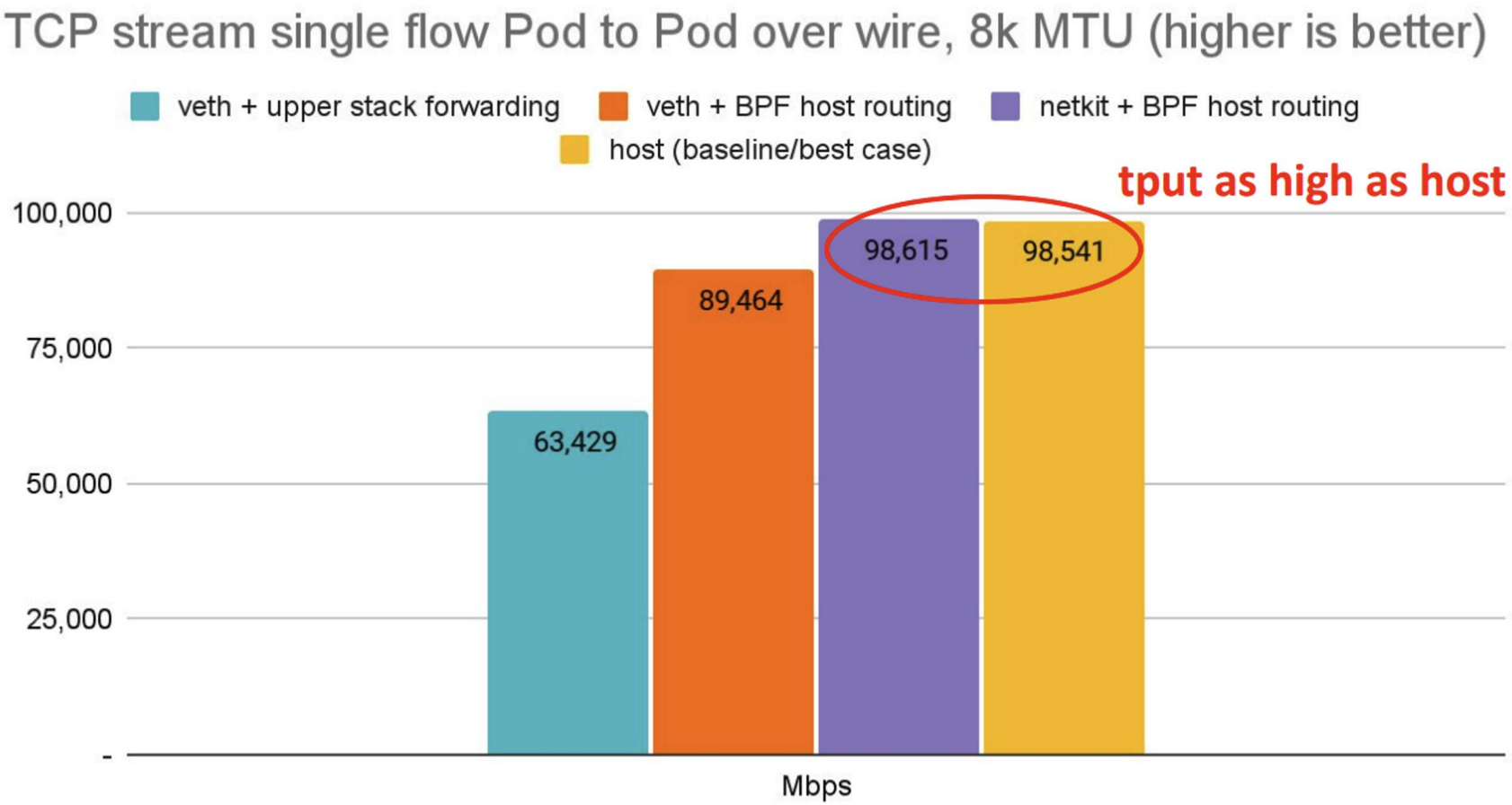
- The veth successor was launched in 2024 – ***netkit***
- Only available with Cilium.
- Runs eBPF programs *inside* the pod
- No unnecessary ARP
- For pod egress traffic bound for a workload outside of the node, netkit would redirect to the physical device without going via the per-CPU backlog queue.



© 2026 Cisco and/or its affiliates. All rights reserved.

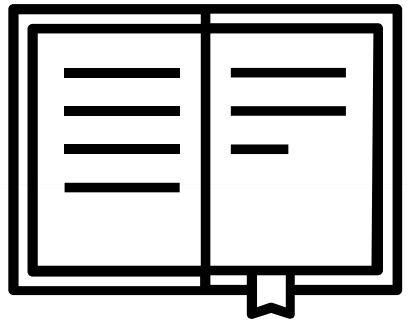
netkit

The results? Network performance parity between container and host



© 2026 Cisco and/or its affiliates. All rights reserved.

netkit



Cilium netkit: The Final Frontier in Container Networking Performance



Nico Vibert | Published: July 9, 2024 | Updated: July 31, 2024 | eBPF, Networking for Kubernetes



Table of Contents

Previously in container networking – a series of compromises

Roadblock #1: Kube-Proxy and iptables

Shortcut #1: eBPF-based Kube-Proxy replacement

Roadblock #2: upper stack forwarding

Shortcut #2: eBPF-based host routing

Roadblock #3 : undersized TCP packets

Shortcut #3: BIG TCP

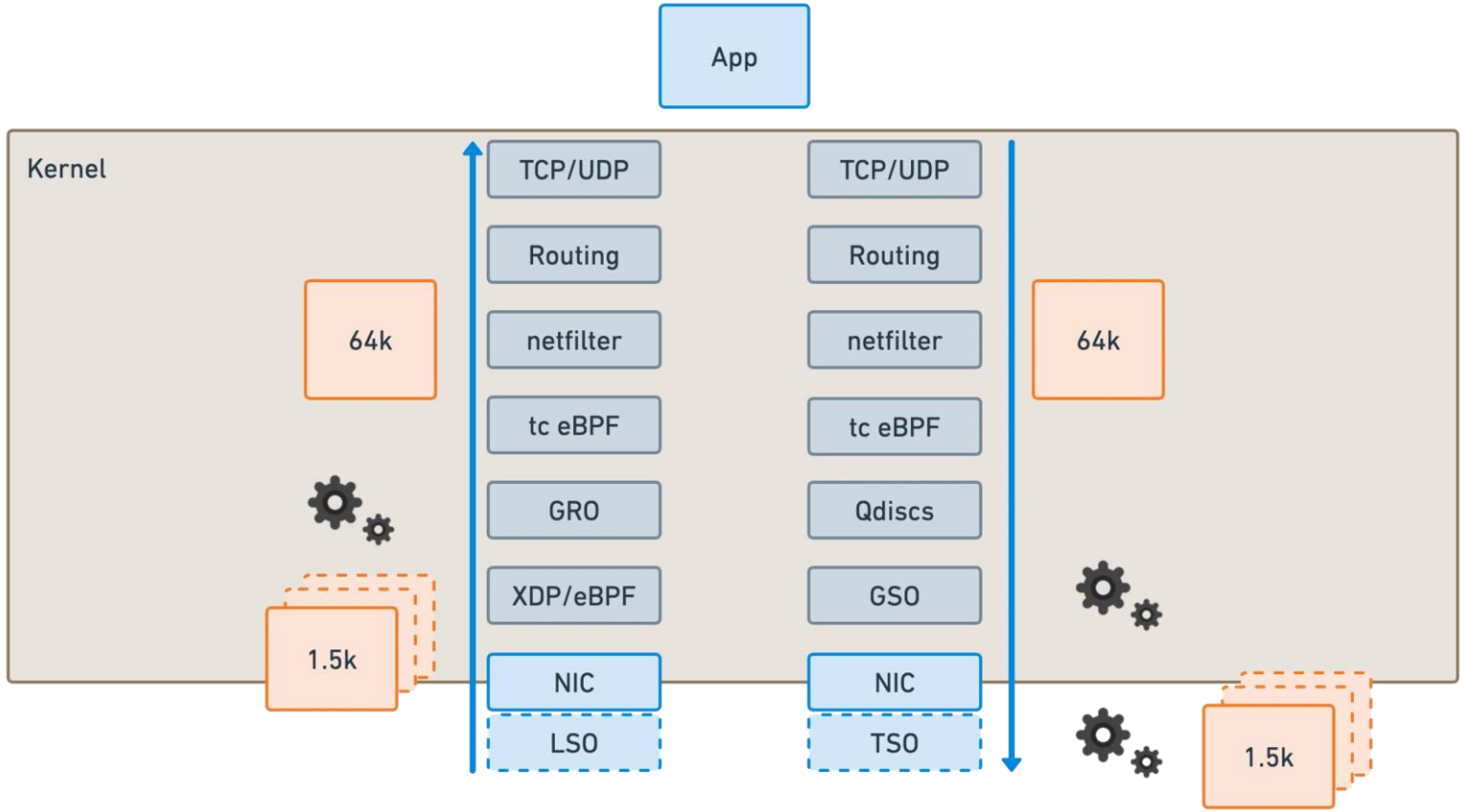
Roadblock #4: A legacy virtual cable

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

BIG TCP

- Invented by Google engineers
- First supported on IPv6, with IPv4 support coming later.
- With adoption of 100Gbps adapters comes a challenge: how can a CPU deal with eight million PPS?
- Would leave only 120 nanoseconds per packet for the CPU to handle – impossible.

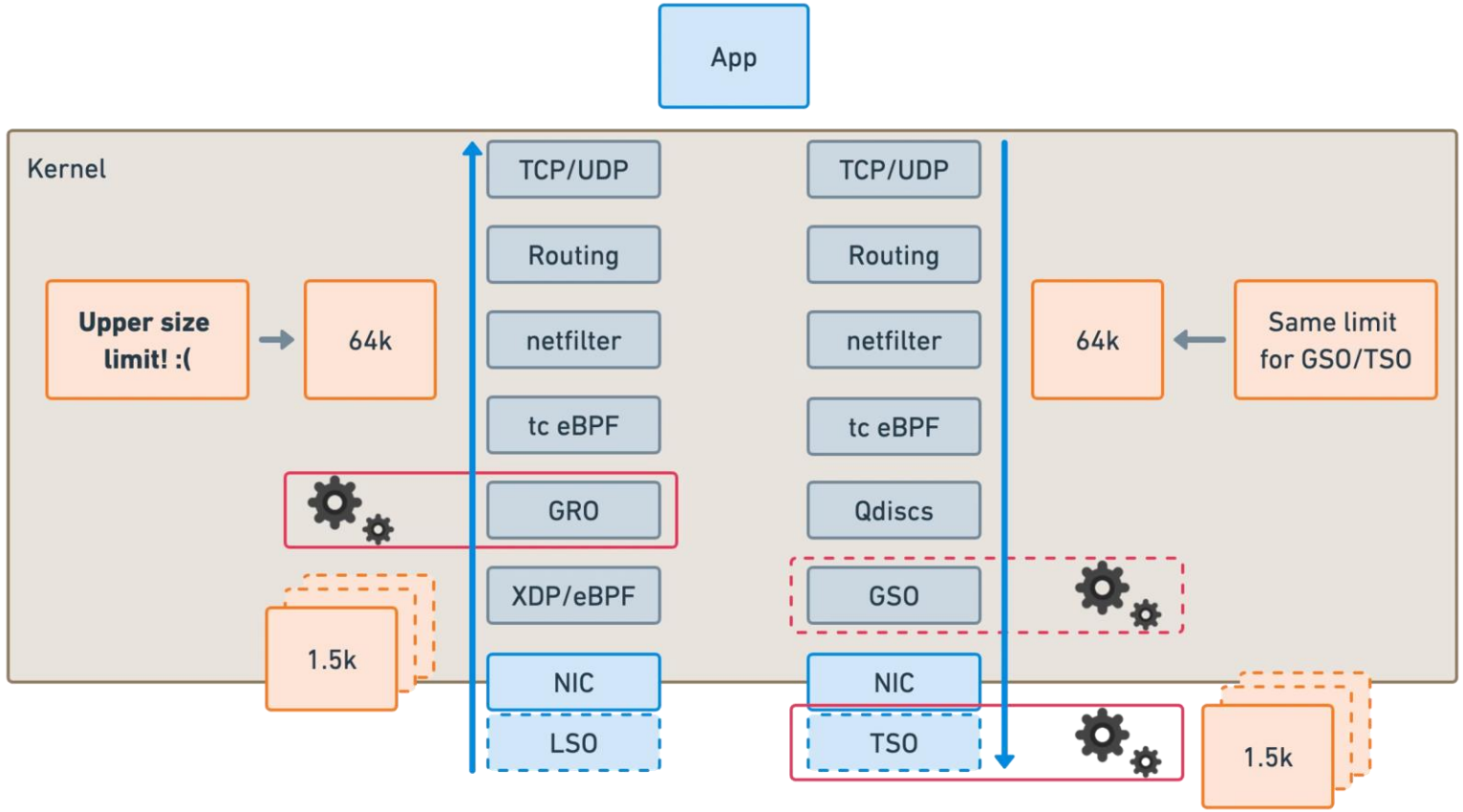


© 2026 Cisco and/or its affiliates. All rights reserved.

BIG TCP

Packet batching is needed to overcome packet limitation

- Linux uses GRO and TSO to batch packets efficiently.
- These features cap packet size around **64KB** due to header limits.
- The IPv4 and IPv6 headers restrict maximum payload size.
- Modern CPUs can handle far larger aggregated packets.

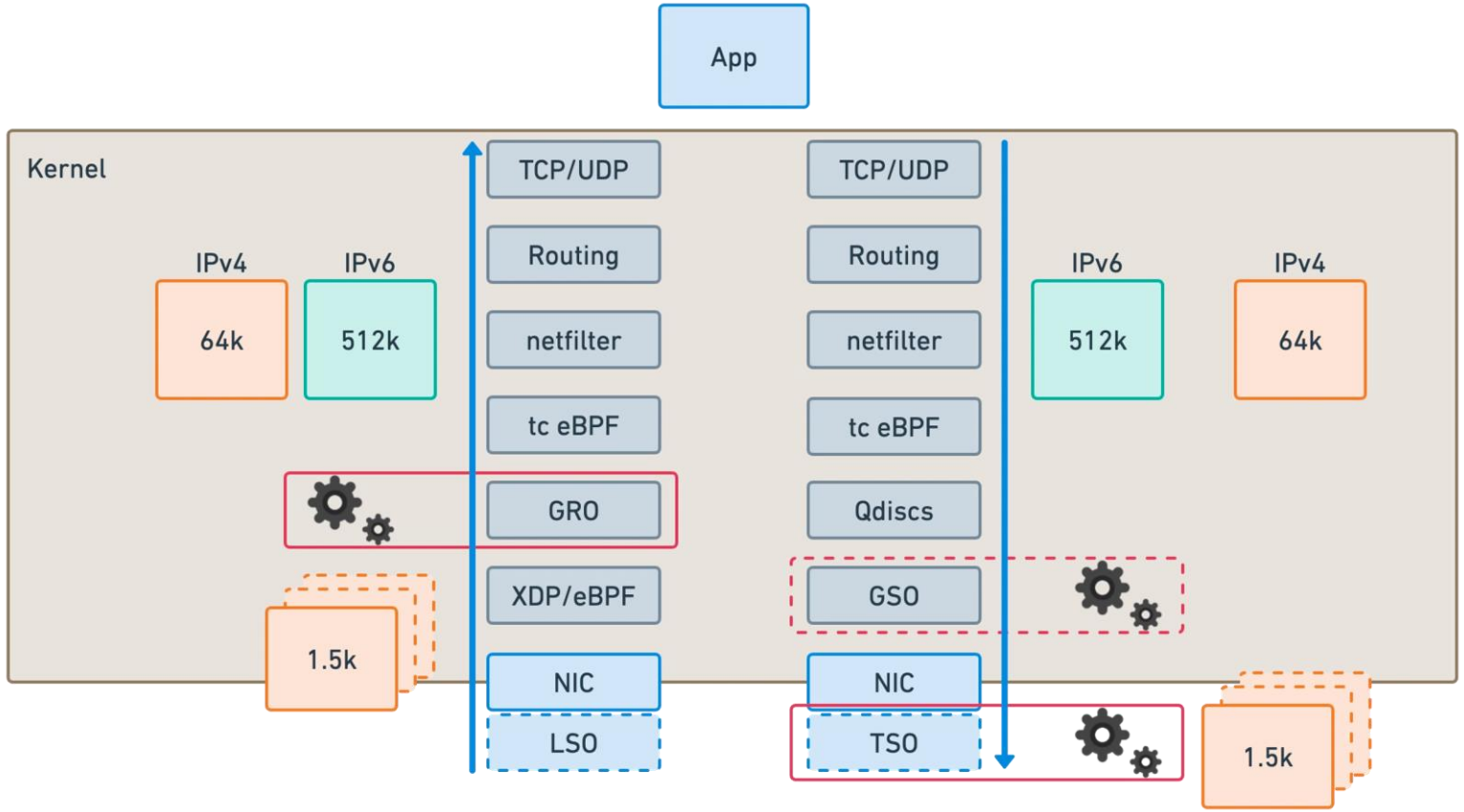


© 2026 Cisco and/or its affiliates. All rights reserved.

BIG TCP

Jumbograms!

- With BIG TCP, we can create **512K jumbograms** instead of 64K packets.
- Lead to 30-40% improvements in throughput.



© 2026 Cisco and/or its affiliates. All rights reserved.

BIG TCP

Results without and with BIG TCP

```
# kubectl exec netperf-client -- netperf  -t TCP_RR -H ${NETPERF_SERVER} -- -r80000:80000 -O
MIN_LATENCY,P90_LATENCY,P99_LATENCY,THROUGHPUT
Minimum      90th         99th         Throughput
Latency      Percentile   Percentile
Microseconds Latency      Latency
             Microseconds Microseconds
73           201          357          6587.63
```

Without BIG TCP

```
# kubectl exec netperf-client -- netperf  -t TCP_RR -H ${NETPERF_SERVER} -- -r80000:80000 -O
MIN_LATENCY,P90_LATENCY,P99_LATENCY,THROUGHPUT
Minimum      90th         99th         Throughput
Latency      Percentile   Percentile
Microseconds Latency      Latency
             Microseconds Microseconds
59           148          290          8663.99
```

With BIG TCP

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

BIG TCP

BIG Performances with BIG TCP on Cilium



Nico Vibert | Published: February 15, 2023 | Updated: August 24, 2023 | Networking for Kubernetes

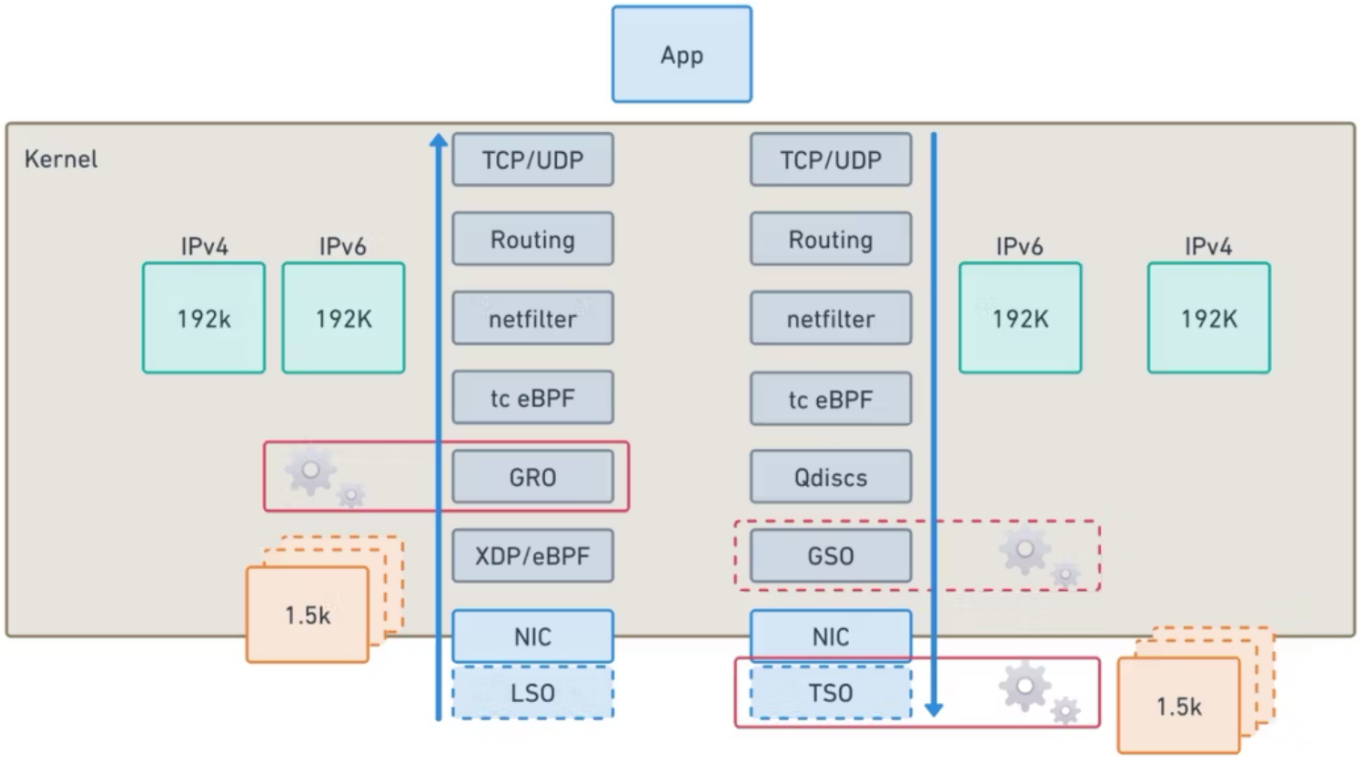


Table of Contents

100Gbps and beyond

Jumbo Frames vs BIG TCP

BIG TCP on Cilium

Learn More

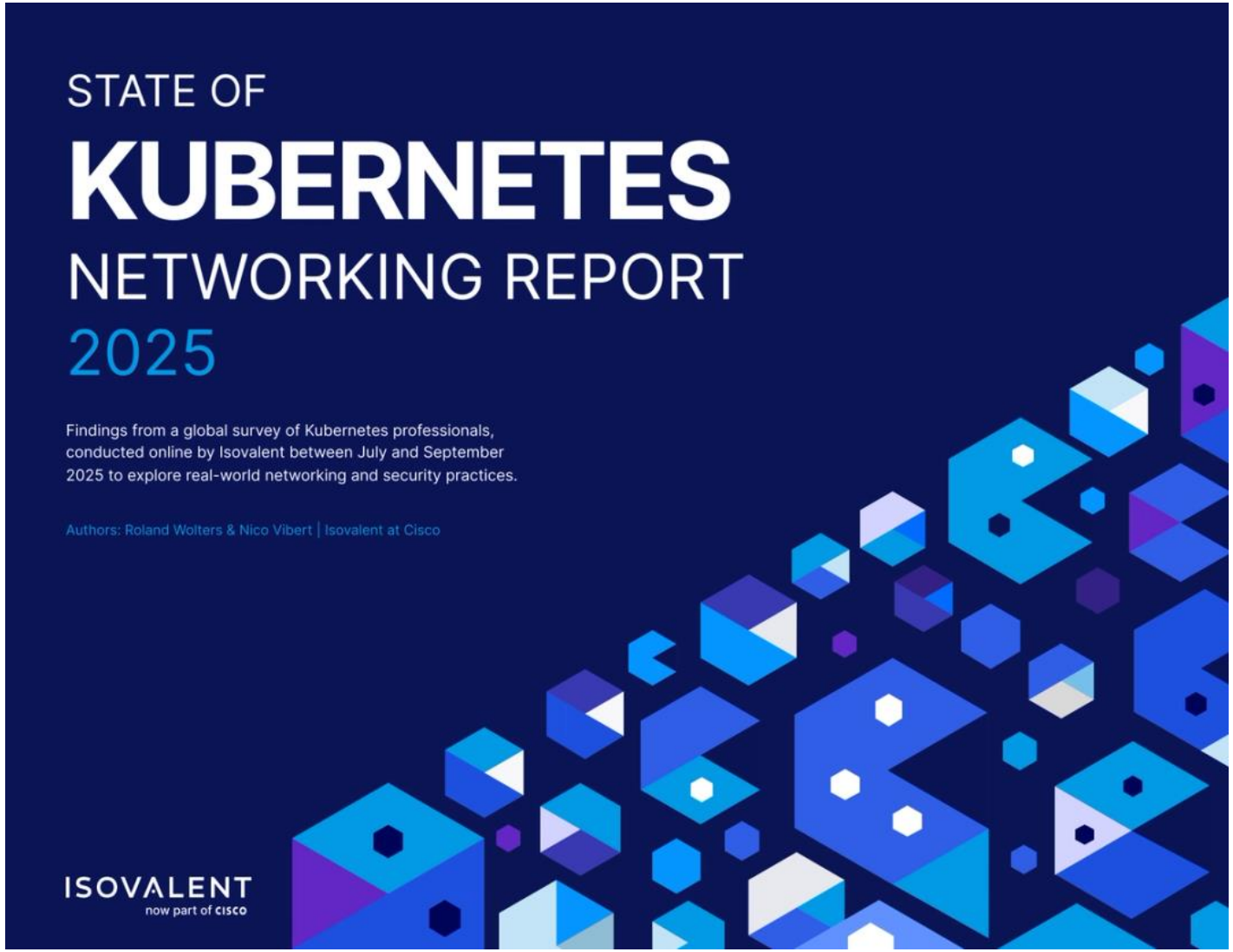
Back to top ↑

Share on social media

© 2026 Cisco and/or its affiliates. All rights reserved.

Final words

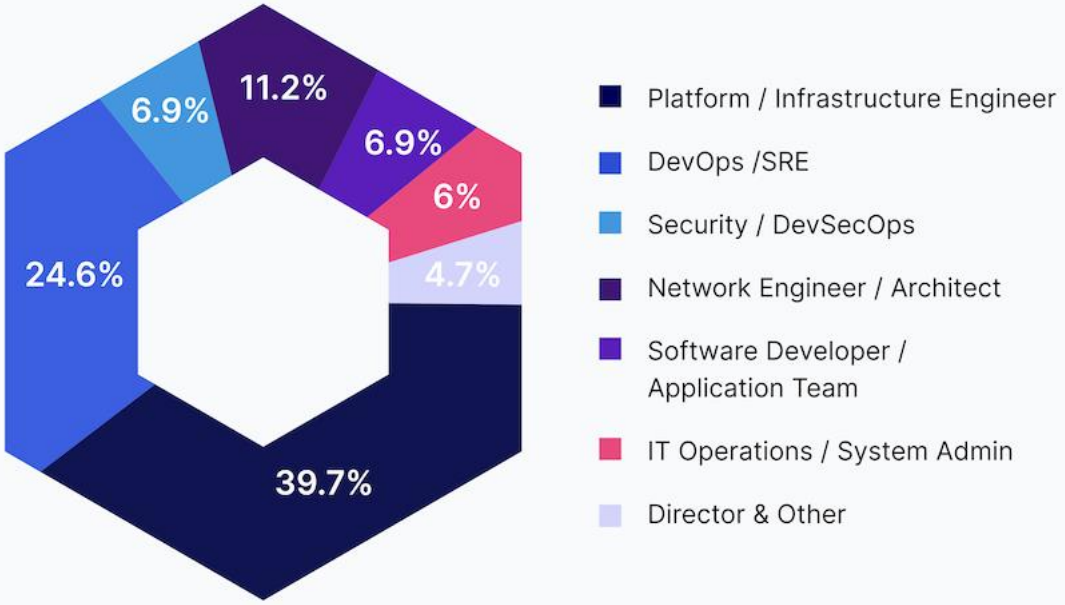
State of Kubernetes Networking Report 2025



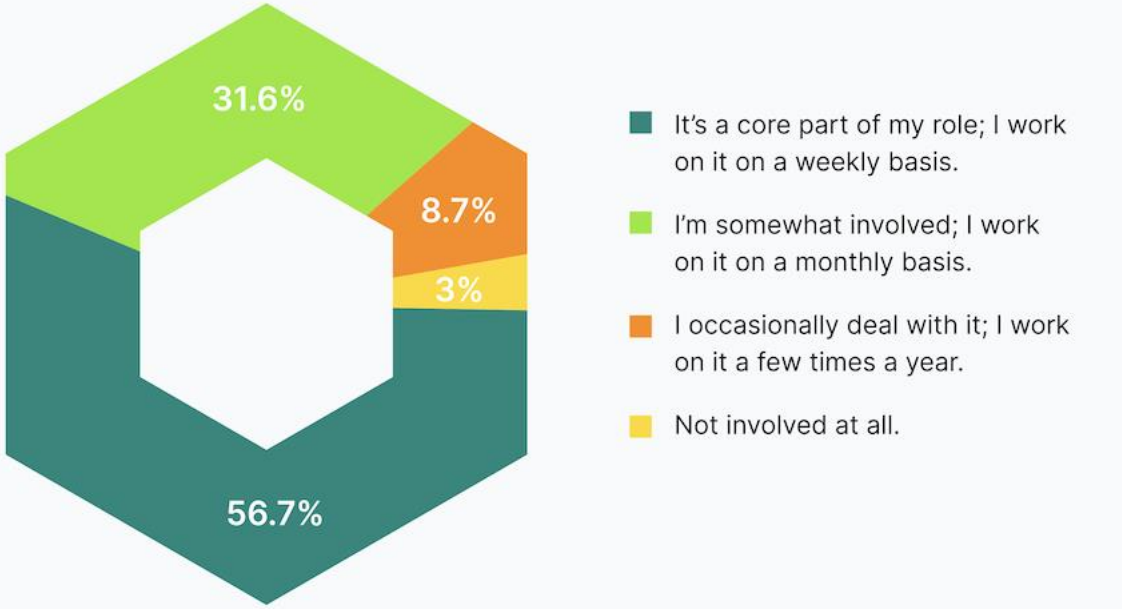
© 2026 Cisco and/or its affiliates. All rights reserved.

State of Kubernetes Networking Report 2025

How would you best describe your role?



How involved are you in managing networking and/or security?



© 2026 Cisco and/or its affiliates. All rights reserved.

- Today, Kubernetes networking is owned by platform teams.

- Network engineers are often excluded, intentionally or not.

- Will you shape the story, or just watch it?

© 2026 Cisco and/or its affiliates. All rights reserved.

Thank you

cisco Live !

Complete your session surveys



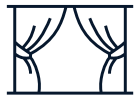
Complete your surveys in the Cisco Events App.



Complete a minimum of 4 session surveys and the overall event survey to receive a unique Cisco Live t-shirt.
(from 11:30 on Thursday, while supplies last)

© 2026 Cisco and/or its affiliates. All rights reserved.

Continue your education



Visit the Cisco Showcase for related demos



Book your one-on-one Meet the Engineer meeting

Visit the Technical Solutions Clinics to discuss your technical questions



Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs



Visit the On-Demand Library for more sessions at CiscoLive.com/On-Demand

© 2026 Cisco and/or its affiliates. All rights reserved.

Isovalent Cisco Live Select Sessions

Bring your favourite session back to the rest of your team!

- Tailored to help magnify the value & impact for your specific business needs.
- Enable deeper internal discussions around the session topic
- Give colleagues direct access to Isovalent expertise through live Q&A

© 2026 Cisco and/or its affiliates. All rights reserved.

Contact me



https://www.linkedin.com/in/nicolasvibert/

© 2026 Cisco and/or its affiliates. All rights reserved.

Thank you

cisco Live !


CISCO