



KubeCon



CloudNativeCon

North America 2025

TikTok's IPv6 Journey to Cilium

Giri Kuncoro & Joseph Pallamidessi

#KubeCon #CloudNativeCon





Giri Kuncoro

 girikuncoro



Joseph Pallamidessi

 pallamidessi

Security Engineering, TikTok, SRE team

TikTok's Case Study as Cilium User



KubeCon



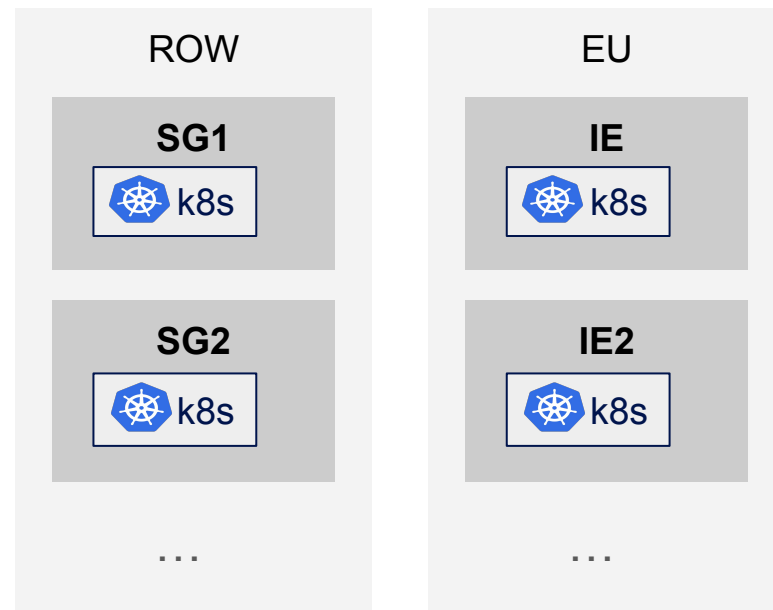
CloudNativeCon

North America 2025

Infrastructure Overview

Context

- Security engineering department
- Self-hosted K8s cluster per “vDC”
- Globally distributed:
 - >10 “regions”
 - >130 vDC



 *Mix of public and private clouds, VMs and bare-metal*

Infrastructure Overview



Application Ops



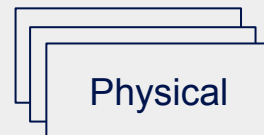
Cluster Ops



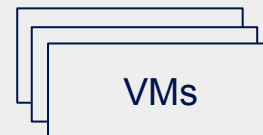
Kernel Ops



Datacenter Ops



Physical

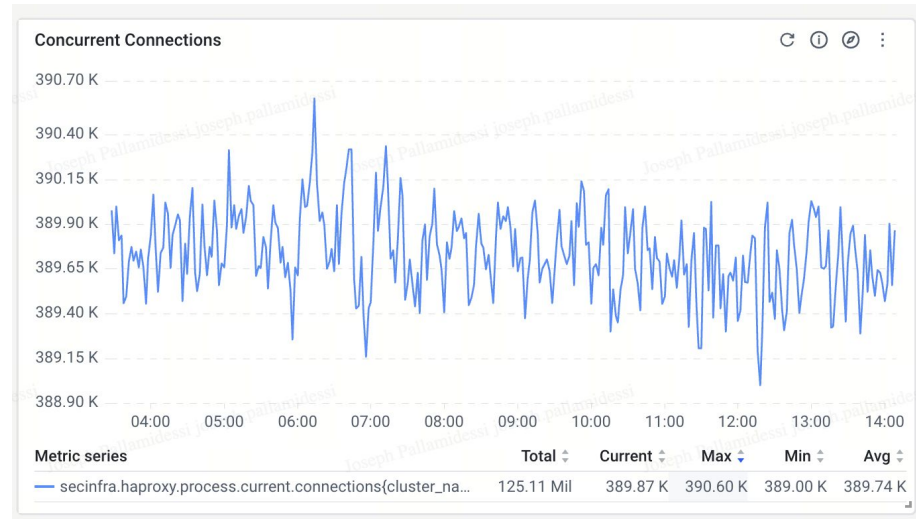
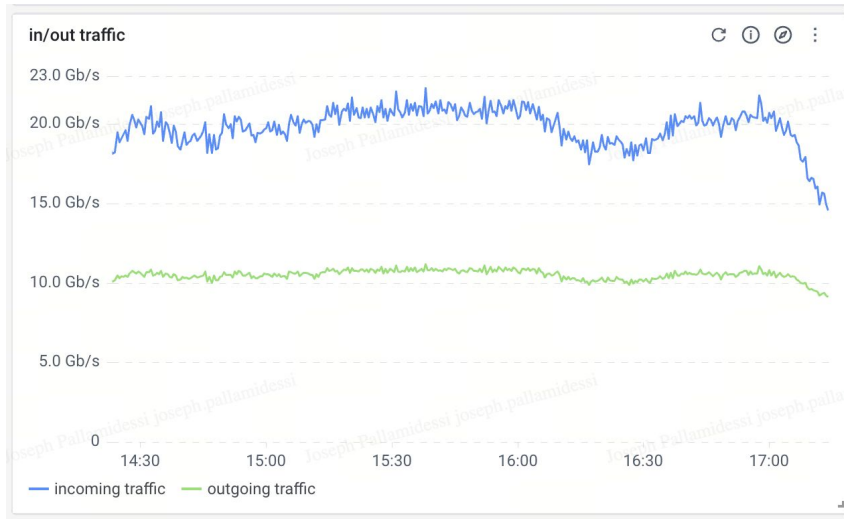


VMs

Infrastructure Overview

🔑 Authn/Authz/Key management/Signing applications

- gRPC/UDP only
- Dual-stack fronting NLB



Choosing the right CNI

We needed:

- ✨ Powerful CNI to **abstract** the underlying network
- 🚀 **High-Performance** Networking
- 🧰 Security feature and observability out-of-the-box
- 💕 Vibrant ecosystem

Choosing the right CNI

We choose  cilium

 Initial tests and rollout are great !

- Kube-proxy replacement
- CNP mTLS
- Network policy
- Hubble

TikTok's Global Private VPC Networks

- **Flat network** with strict compliance control
- **Unique IP** address globally

But:

- No control over the network configuration (BGP, Routing)
- Started to run out IPv4 addresses
- Building **IPv6 only datacenter**

IPv6-only Kubernetes Cluster

- ✗ Cilium doesn't seem to work
- ✗ IPv6 is a hard requirement

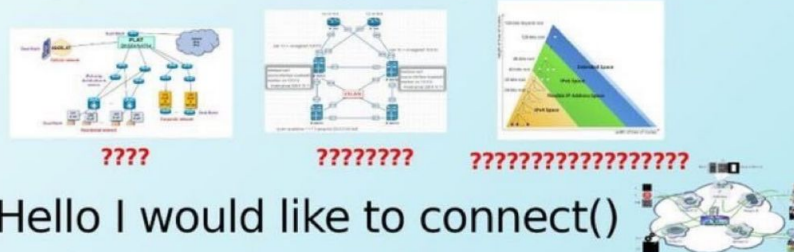
✨ We decide to invest in Cilium
(and use calico in the meantime)

STOP DOING IPv6

- ADDRESSES WERE NOT SUPPOSE TO HAVE LETTERS IN THEM
- YEARS OF WORK yet NO REAL-WORLD USE FOUND for going bigger than 10.0.0.0/8
- Wanted more hosts than that for a laugh? We had a tool for that: It was called "NAT"
- 'Yes please connect to 2001:500:2f::f. Just ping ff02::1%eth0' - Statements dreamed up by the utterly Deranged

LOOK at what networking tech have been demanding your Respect for all this time, with all the routers & switches we built for them

(This is REAL network, done by REAL networkers):



They have played us for absolute fools

Road to Cilium IPv6



KubeCon



CloudNativeCon

North America 2025

IPv6-only Kubernetes Cluster

```
$ kubectl get node -owide
NAME          STATUS  INTERNAL-IP
host-001     Ready  2001:db8:abcd:1::1
host-002     Ready  2001:db8:abcd:2::1
host-003     Ready  2001:db8:abcd:2::2
...
```

IPv6-only Kubernetes Cluster



Subnet A

Node 1

Node 2




Subnet B

Node 3

...



Cilium Rollout

- IPv4-only Kubernetes 
- Dual-stack Kubernetes 
- IPv6-only Kubernetes 

Cilium Rollout in IPv6-only

Cilium v1.12



2023

Problem #1: Cilium agent crashes



Problem #1: Cilium agent crashes

- Cilium agent exits immediately if it can't find IPv4 address
([#21538](#))

```
$ kubectl logs cilium
level=info msg="Validating configured node address ranges" subsys=daemon
level=fatal msg="postinit failed" error="external IPv4 node address could not be
derived, please configure via --ipv4-node" subsys=daemon
level=info msg="Starting IP identity watcher" subsys=ipcache
```

Cilium Routing Mode

Encapsulation

Native-Routing

Cilium Routing Mode

Encapsulation

- Simplicity
- Larger addressing space
- MTU Overhead

Native-Routing

Cilium Routing Mode

Encapsulation

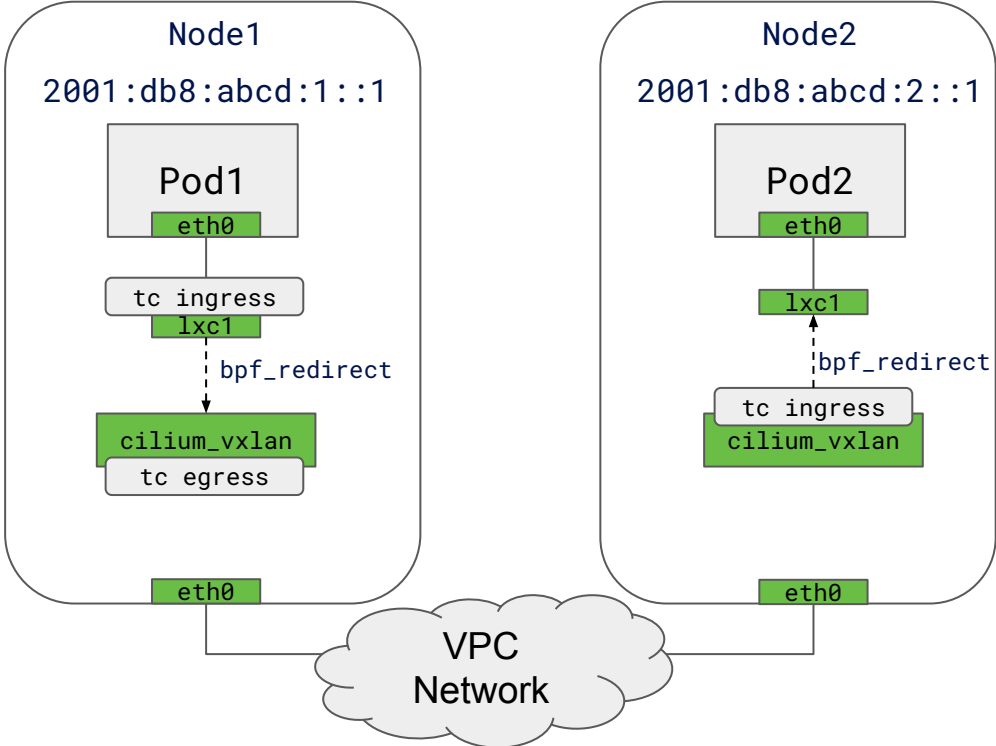
- Simplicity
- Larger addressing space
- MTU Overhead

Native-Routing

- Rely on kernel routing
- Nodes must aware of Pod
CIDRs

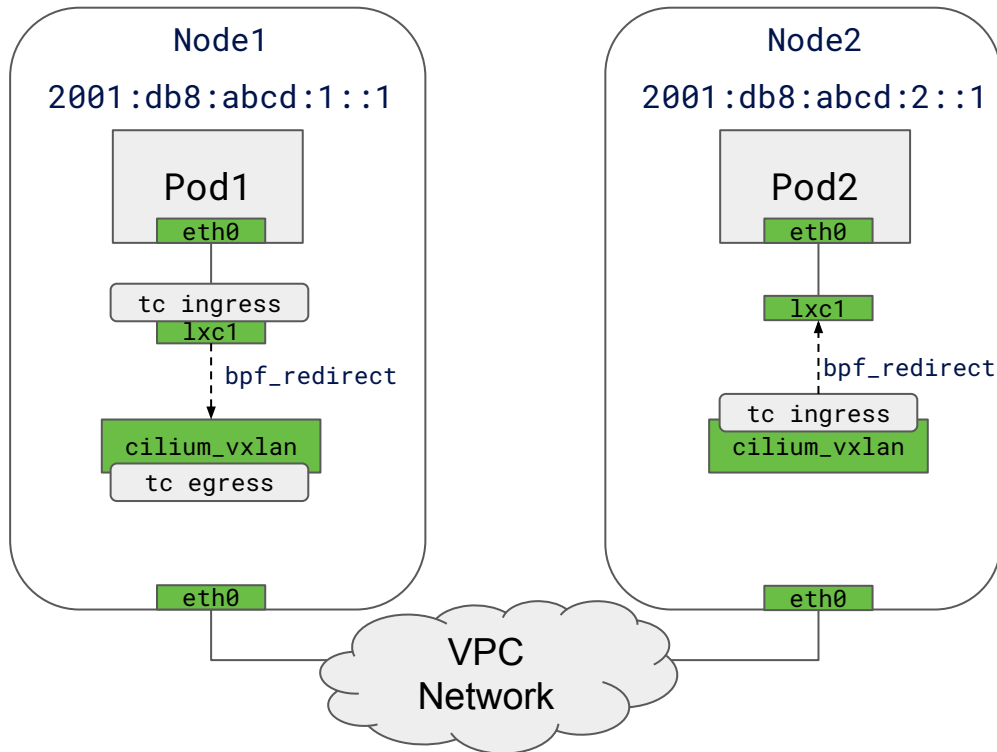
Cilium v1.12

- Encapsulation mode



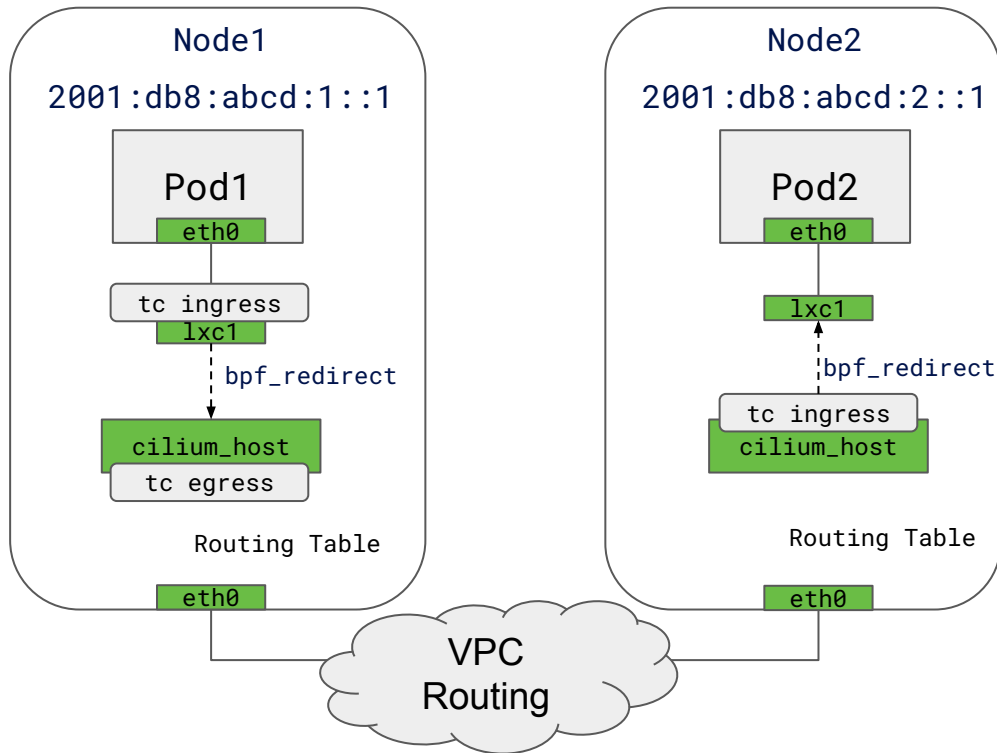
Cilium v1.12

- Encapsulation mode ❌
 - IPv6 encapsulation not supported ([#17240](#))



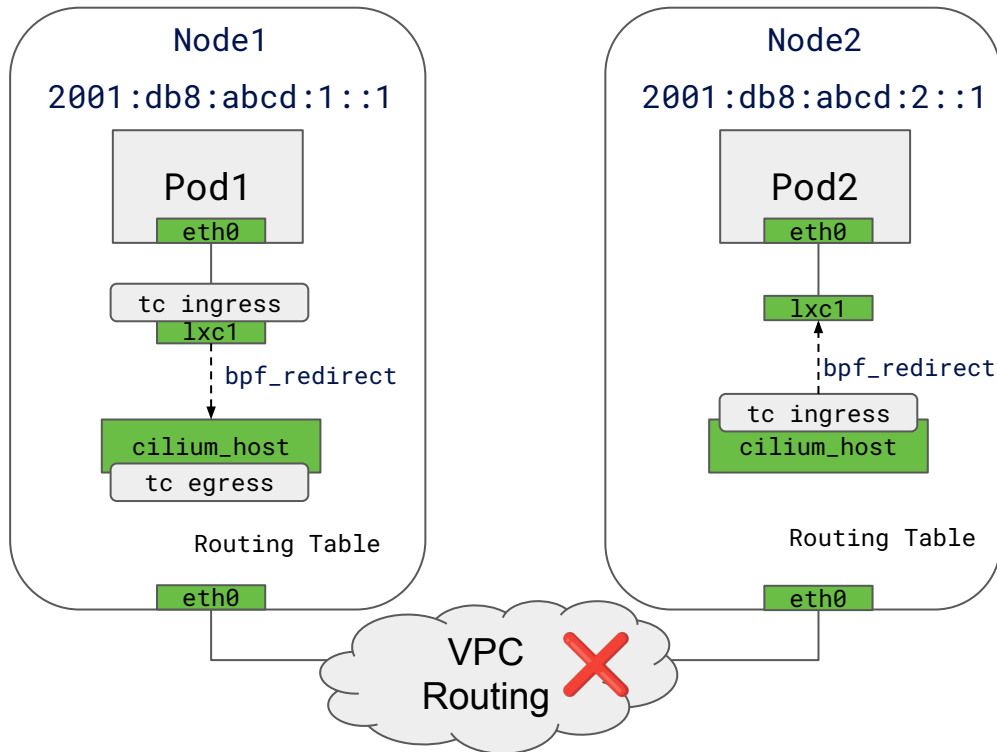
Cilium v1.12

- Native-routing mode
 - VPC routing



Cilium v1.12

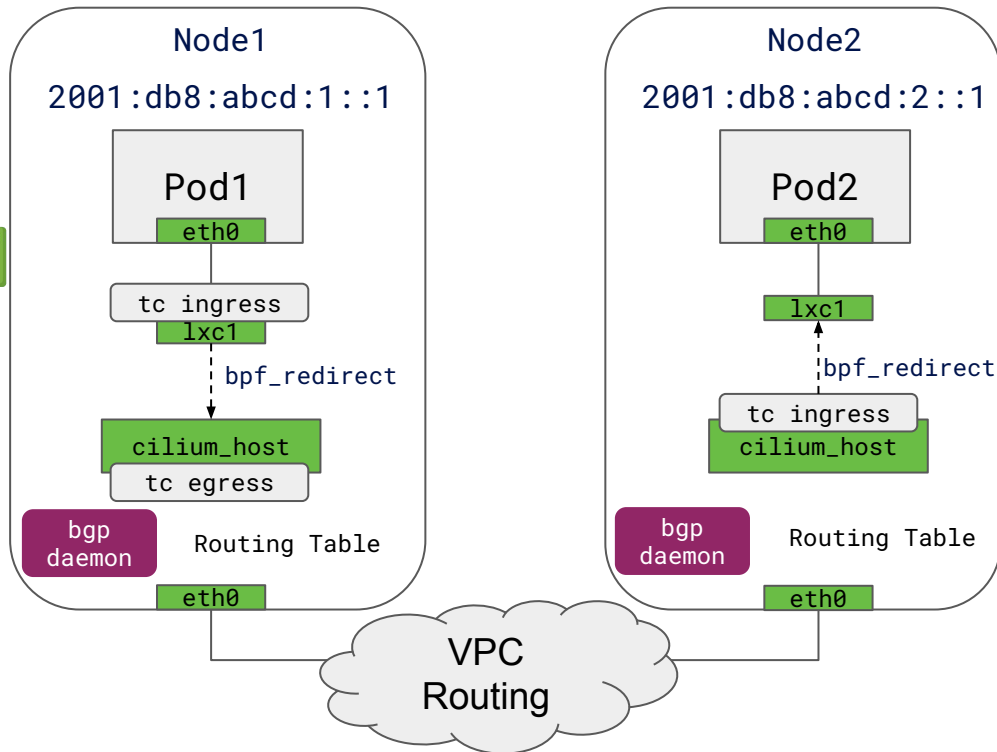
- Native-routing mode
 - VPC routing ❌



Cilium v1.12

- Native routing mode

- VPC routing ❌
- Cluster-level BGP ✅



- Native routing mode with BGP

- Using Kube-router

[🏠 / BGP / Using Kube-Router to Run BGP \(deprecated\)](#)

Using Kube-Router to Run BGP (deprecated)

This guide explains how to configure Cilium and kube-router to co-operate to use kube-router for BGP peering and route propagation and Cilium for policy enforcement and load-balancing.

- Using BIRD

[🏠 / BGP / Using BIRD to run BGP \(deprecated\)](#)

Using BIRD to run BGP (deprecated)

BIRD is an open-source implementation for routing Internet Protocol packets on Unix-like operating systems. If you are not familiar with it, you had best have a glance at the [User's Guide](#) first.

BIRD provides a way to advertise routes using traditional networking protocols to allow Cilium-managed endpoints to be accessible outside the cluster. This guide assumes that Cilium is already deployed in the cluster, and that the remaining piece is how to ensure that the pod CIDR ranges are externally routable.

Problem #2: Per-endpoint Routes Broken



Problem #2: Per-endpoint Routes Broken

- IPv6 NDP traffic was dropped by CiliumNetworkPolicy due to incorrect packet identification ([#23852](#) [#23910](#))
 - Fixed in v1.14

```
xx drop (Policy denied) flow 0x0 to endpoint 1096, ifindex 35, file 2:1604, ,  
identity world->54318: fd00:10:244:2::27b4 -> fe80::241e:5dff:fe66:6c4d  
NeighborAdvertisement
```

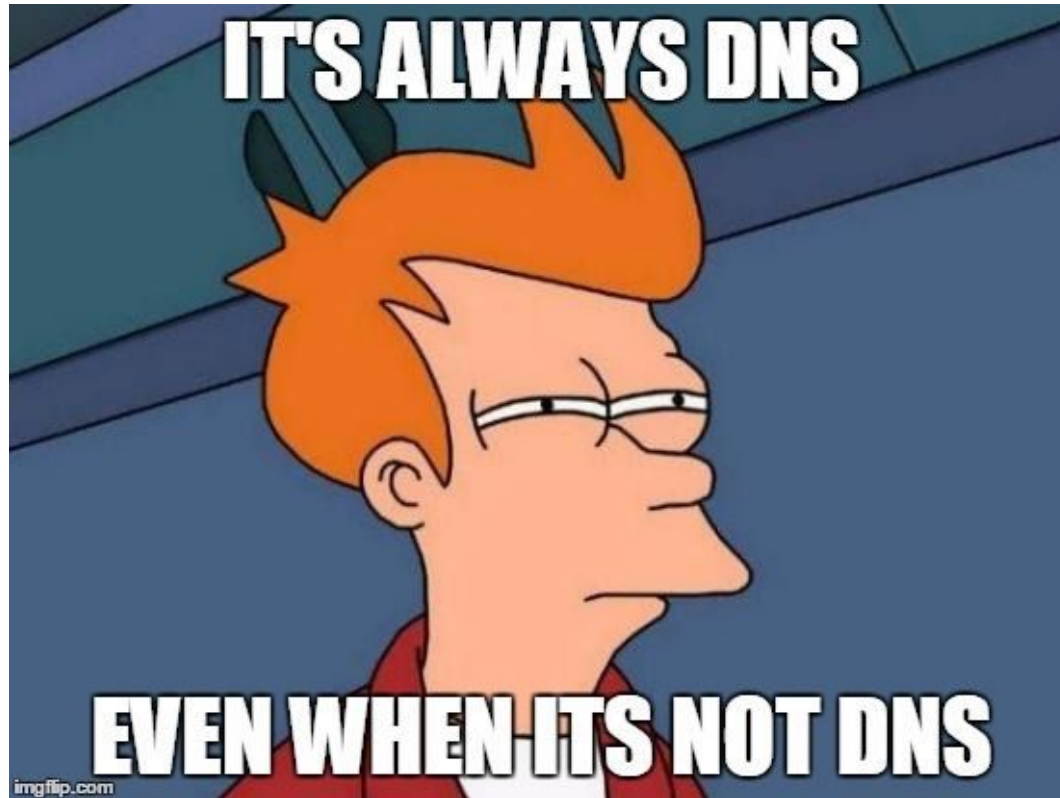
Cilium Rollout in IPv6-only

Cilium v1.14

●
2023

●
2024

Problem #3: L7 DNS Policy not working



Problem #3: L7 DNS Policy not working

- L7 DNS policy doesn't work with IPv6 DNS server ([#28678](#))
 - Fixed in v1.15
- The DNS proxy doesn't set checksum in UDP header

```
> Frame 4: 285 bytes on wire (2280 bits), 285 bytes captured (2280 bits)
> Ethernet II, Src: 72:bc:a1:37:10:59 (72:bc:a1:37:10:59), Dst: 52:14:78:79:7b:d5 (52:14:78:
> Internet Protocol Version 6, Src: 2001:db8:1234:5678:2::efa0, Dst: 2001:db8:1234:5678:4::f
v User Datagram Protocol, Src Port: 53, Dst Port: 34766
  Source Port: 53
  Destination Port: 34766
  Length: 231
  Checksum: 0x0000 [zero-value illegal]
    [Expert Info (Error/Checksum): Illegal checksum value (0)]
      [Illegal checksum value (0)]
      [Severity level: Error]
      [Group: Checksum]
      [Checksum Status: Illegal]
  [Stream index: 0]
  [Timestamps]
  UDP payload (223 bytes)
  Domain Name System (response)
0000  52 14 78 79 7b d5 72 bc a1 37 10 59 86 dd 60 00 R xy{r .7.Y.
0010  47 cb 00 e7 11 3f 20 01 0d b8 12 34 56 78 00 02 G . . . ? . . . .4Vx.
0020  00 00 00 00 ef a0 20 01 0d b8 12 34 56 78 00 04 . . . . . . . . . .4Vx.
0030  00 00 00 00 f8 9b 00 35 87 ce 00 e7 00 00 50 49 . . . . .5 . . . . .PI
0040  85 80 00 01 00 04 00 00 00 01 03 77 77 77 06 67 . . . . . . . . . .www.g
0050  6f 6f 67 6c 65 03 63 6f 6d 00 00 1c 00 01 03 77 oogle.co m . . . . .w
0060  77 77 06 67 6f 6f 67 6c 65 03 63 6f 6d 00 00 1c ww.googl e.com . . . . .
0070  00 01 00 00 00 1a 00 10 24 04 68 00 40 03 0c 05 . . . . . $ . h @ . . . . .
0080  00 00 00 00 00 00 00 93 03 77 77 77 06 67 6f 6f . . . . . . . . . .www.goo
0090  67 6c 65 03 63 6f 6d 00 00 1c 00 01 00 00 00 1a gle.com . . . . . . . . . .
00a0  00 10 24 04 68 00 40 03 0c 05 00 00 00 00 00 00 . . $ . h @ . . . . .
00b0  00 68 03 77 77 77 06 67 6f 6f 67 6c 65 03 63 6f . h . www . g oogle . co
00c0  6d 00 00 1c 00 01 00 00 00 1a 00 10 24 04 68 00 00 1a 00 10 24 04 68 00 m . . . . . $ . h .
00d0  40 03 0c 05 00 00 00 00 00 00 00 6a 03 77 77 77 @ . . . . . . . . . . j . www
00e0  06 67 6f 6f 67 6c 65 03 63 6f 6d 00 00 1c 00 01 . google . com . . . . .
00f0  00 00 00 1a 00 10 24 04 68 00 40 03 0c 05 00 00 . . . . . $ . h @ . . . . .
0100  00 00 00 00 00 63 00 00 29 10 00 00 00 00 00 00 . . . . . c . . . . .
0110  0c 00 0a 00 08 e0 29 51 23 fc ca 06 72 . . . . . ) Q # . . . . . r
```

Cilium Rollout in IPv6-only





Cilium v1.15

2023





2024

Jan 2025

Test Results

- Pod-to-pod same node 
- Pod-to-pod different node 
- Pod-to-external 
- DNS resolution 

Test Results

- Pod-to-pod same node 
- Pod-to-pod different node 
- Pod-to-external 
- DNS resolution 

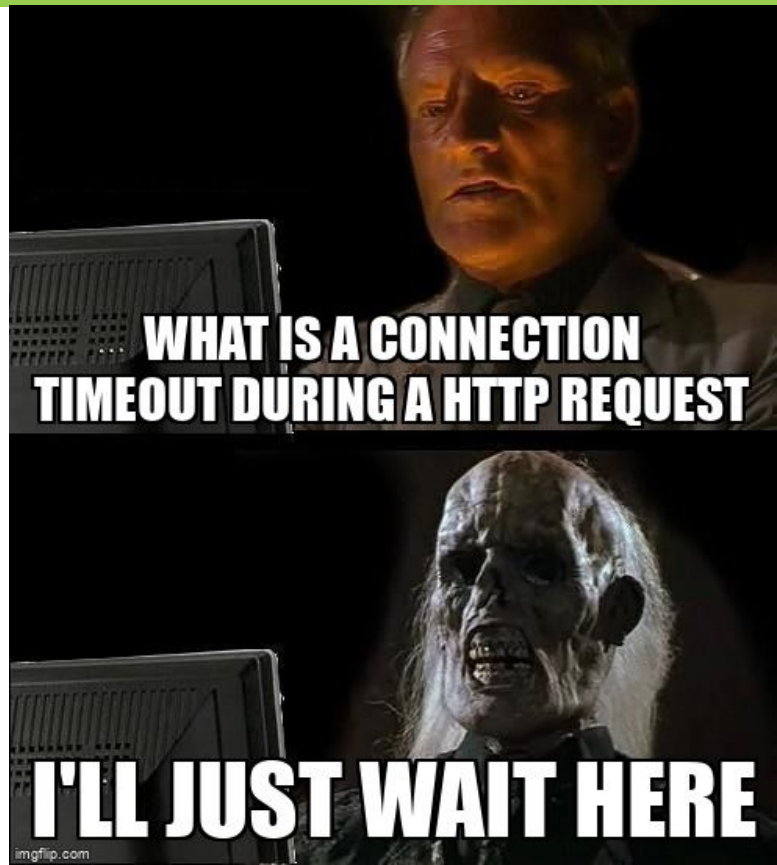
Set `KubeProxyReplacement=true`

Test Results

- Pod-to-pod same node ✓
- Pod-to-pod different node ✓
- Pod-to-external ✓
- DNS resolution ✓

Set `KubeProxyReplacement=true`

- External-to-pod (NodePort) ✗



Test Results

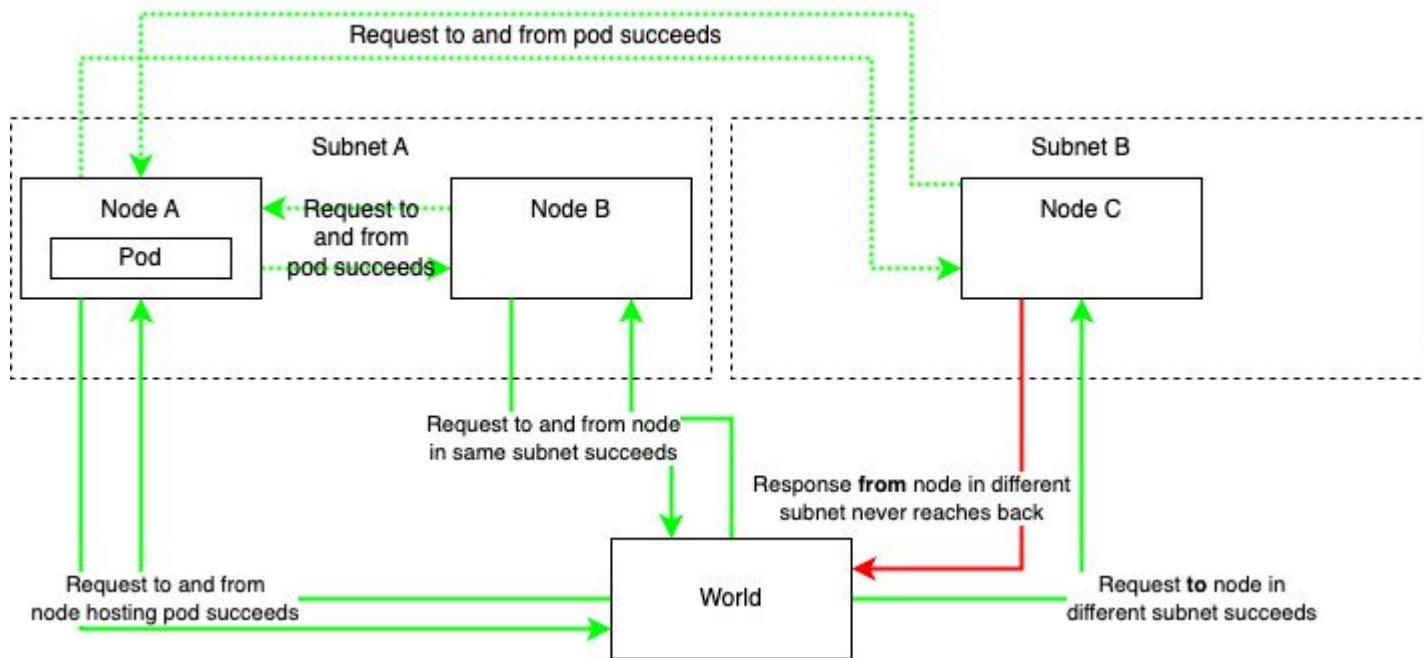
- End-to-end connectivity tests ❌

```
$ cilium connectivity test \  
--external-cidr '2606:4700:4700::/48' \  
--external-ip '2606:4700:4700::1111' \  
--external-other-ip '2606:4700:4700::1001' \  
--ip-families 'ipv6'
```

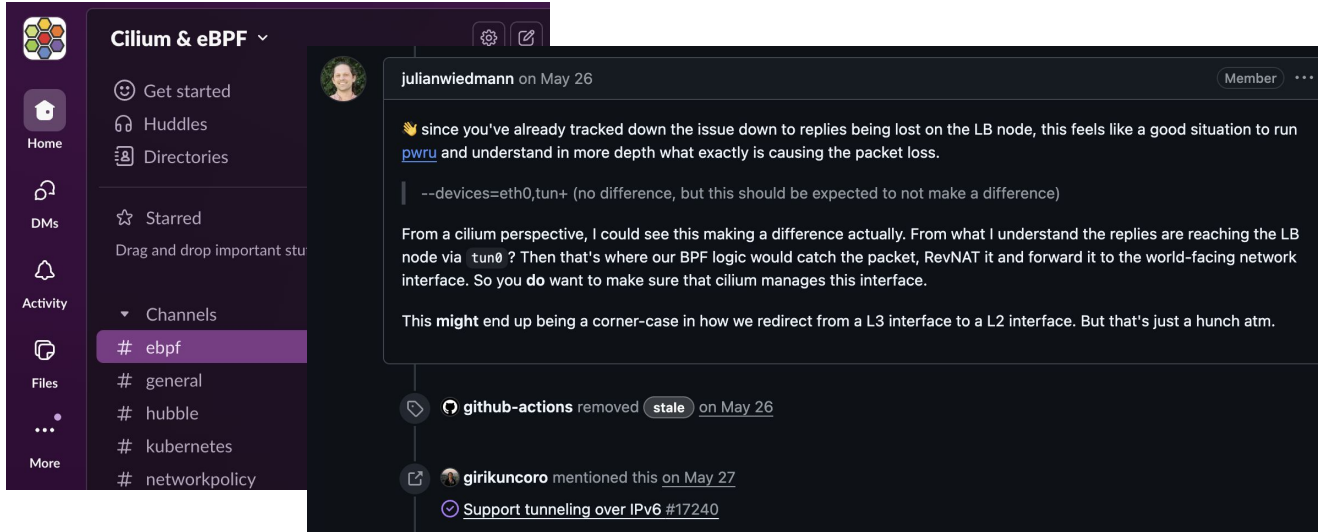
📄 Test Report [cilium-test-1]

❌ 33/64 tests failed ...

Problem #4: NodePort timeout



Discussions with Maintainers



Cilium & eBPF

Get started
Huddles
Directories
Starred
Channels
ebpf
general
hubble
kubernetes
networkpolicy

Member

julianwiedmann on May 26

👉 since you've already tracked down the issue down to replies being lost on the LB node, this feels like a good situation to run [pwru](#) and understand in more depth what exactly is causing the packet loss.

```
--devices=eth0,tun+ (no difference, but this should be expected to not make a difference)
```

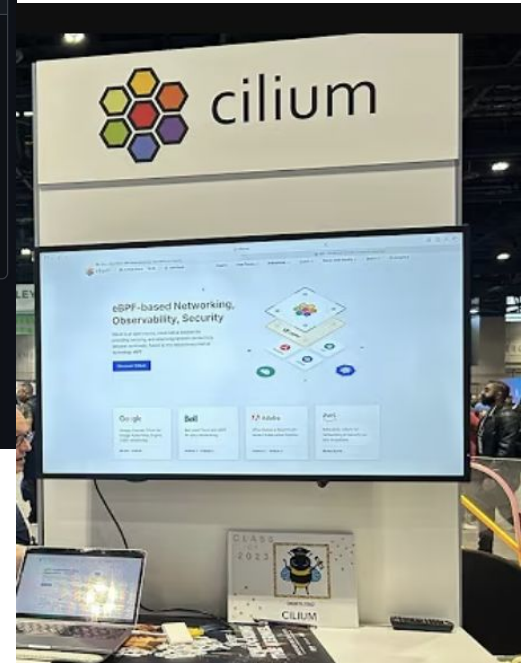
From a cilium perspective, I could see this making a difference actually. From what I understand the replies are reaching the LB node via `tun0` ? Then that's where our BPF logic would catch the packet, RevNAT it and forward it to the world-facing network interface. So you do want to make sure that cilium manages this interface.

This **might** end up being a corner-case in how we redirect from a L3 interface to a L2 interface. But that's just a hunch atm.

github-actions removed **stale** on May 26


girikuncoro mentioned this on May 27

👍 Support tunneling over IPv6 #17240




Cilium Support Tunneling over IPv6

- Support added in 1.18 ([#17240](#)) 🎉🎉🎉

 pchaigno on May 16 Member ...

Pull requests [#38523](#), [#38296](#), and [#39074](#) added support for an IPv6 underlay in IPv6-only clusters. Support for dual-stack clusters will come as a follow up. It should be part of the next minor release.

👍 6 ❤️ 4 🚀 4

 pchaigno on Jul 4 Member ...

Support for IPv6 underlays in dual-stack clusters has just been merged at [#40324](#) and will be available in Cilium v1.19.

👍 8 🚀 7

Cilium Support Tunneling over IPv6

- IPv6 Underlay for Encapsulation Mode
- IPv6 Underlay for Kube Proxy Replacement

```
--underlay-protocol
```

Set the IP family for the underlay. Defaults to `ipv4`. The underlying network must support that protocol. `ipv6` is only supported in IPv6-only clusters.

Cilium Rollout in IPv6-only

Cilium v1.18

2023

2024

Jan 2025

May 2025



Test Results

- Connectivity check tests ([connectivity-check.yaml](#)) 🎉🎉🎉

echo-a-64c5768bc9-16ggv	1/1	Running
echo-b-5c7cc5fc4f-tzwpz	1/1	Running
echo-b-host-684fc8c6d6-nmhfk	1/1	Running
host-to-b-multi-node-clusterip-54b75f998d-6vtpm	1/1	Running
host-to-b-multi-node-headless-78bcbc6f9b-92btk	1/1	Running
pod-to-a-68d9b9d5d-xwr8v	1/1	Running
pod-to-a-allowed-cnp-85b8f754bd-t2g5z	1/1	Running
pod-to-a-denied-cnp-5fdd4cdc58-b67cv	1/1	Running
pod-to-b-intra-node-nodeport-6db7f65984-76bld	1/1	Running
pod-to-b-multi-node-clusterip-857855dc74-hfzdh	1/1	Running
pod-to-b-multi-node-headless-788f886b4d-hjp25	1/1	Running
pod-to-b-multi-node-nodeport-668df7b68d-xt4f7	1/1	Running
pod-to-external-1111-8549d99f48-9cr4c	1/1	Running
pod-to-external-fqdn-allow-google-cnp-f6df4f66-8wx46	1/1	Running

- End-to-end connectivity tests 🎉🎉🎉

```
$ cilium connectivity test \  
--external-cidr '2606:4700:4700::/48' \  
--external-ip '2606:4700:4700::1111' \  
--external-other-ip '2606:4700:4700::1001' \  
--ip-families 'ipv6'
```

📄 Test Report [cilium-test-1]

❌ 1/77 tests failed (1/633 actions), 46 tests skipped, 0 scenarios skipped

Cilium Rollout in IPv6-only



2023

2024

Jan 2025

Sep 2025

Debugging Techniques



KubeCon



CloudNativeCon

North America 2025

hubble observe

Filter by resources, incoming/outgoing traffic, policy verdict, etc.

- `hubble observe --pod foo --verdict DROPPED`
- `hubble observe --to-fqdn tiktok.com`
- `hubble observe --pod bar --protocol dns`

hubble cheatsheet

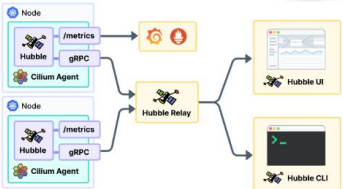
Hubble Cheat Sheet

Hubble can provide information on the following areas:

- Service dependencies and communication maps
- Network Monitoring and alerting
- Application Monitoring
- Security Observability



Components of Hubble



Cilium Agent - Runs the cilium-agent binary which acts as a CNI to manage connectivity, observability, and security for all CNI-managed Kubernetes pods.
Hubble Relay - Provides a cluster-wide API for querying Hubble flow data, which can be accessed directly or via the Hubble CLI and UI.
Hubble UI - Provides a graphical UI for visualizing network flow data, network policy, and security-related events.

Accessing Hubble

To access the CLI

```
$ cilium hubble port-forward&
Forwarding from 0.0.0.0:4245 -> 4245
Forwarding from [::]:4245 -> 4245
```

To access the UI

```
$ cilium hubble ui
Forwarding from 0.0.0.0:12000 -> 8081
Forwarding from [::]:12000 -> 8081
```

Checking Hubble Status

Seeing the current and max flows at 100% is expected, as the Hubble relay ring buffer fills, older events will automatically be dropped.

```
$ hubble status
Healthcheck (via localhost:4245): Ok
Current/Max Flows: 11917/12288 (96.98%)
Flows/s: 11.74
Connected Nodes: 3/3
```

Selecting which traffic flows to observe from the buffer

```
$ hubble observe
--all          Get all flows stored in Hubble's buffer
--first N     Get first N flows stored in Hubble's buffer
-f, --follow  Follow flows output
--last N     Get last N flows stored in Hubble's buffer (default 20)
```

OBSERVING AND FILTERING TRAFFIC EXAMPLES

Observe by Resource

With **hubble observe** you can filter by resources, either looking at incoming/outgoing or all traffic for that resource, below is a list of the filters available.

- from-label filter Show only flows originating in an endpoint with the given labels (e.g. "key1=value1")
- from-namespace filter Show all flows originating in the given Kubernetes namespace
- from-pod filter Show all flows originating in the given pod name prefix([namespace]/<pod-name>). If namespace is not provided, 'default' is used
- from-port filter Show only flows with the given source port (e.g. 8080)
- from-service filter Show flows where the source IP address matches the ClusterIP address of the given service name prefix([namespace]/<svc-name>).
- l, --label filter Show only flows related to an endpoint with the given labels (e.g. "key1=value1")
- n, --namespace filter Show all flows related to the given Kubernetes namespace
- node-name filter Show all flows which match the given node names (e.g. "k8s*", "test-cluster/*company.com")
- not filter[=true] Reverses the next filter to be blacklist i.e. --not --from-ip 2.2.2.2
- pod filter Show all flows related to the given pod name prefix ([namespace]/<pod-name>). If namespace is not provided, 'default' is used.
- service filter Show flows where either the source or destination IP address matches the ClusterIP address of the given service name prefix ([namespace]/<svc-name>)
- to-label filter Show only flows terminating in an endpoint with given labels (e.g. "key1=value1")
- to-namespace filter Show all flows terminating in the given Kubernetes namespace.
- to-pod filter Show all flows terminating in the given pod name prefix([namespace]/<pod-name>). If namespace is not provided, 'default' is used
- to-port filter Show only flows with the given destination port (e.g. 8080)

--to-service filter Show flows where the destination IP address matches the ClusterIP address of the given service name prefix ([namespace]/<svc-name>)

The following examples will show a mix of these filters in use

Observe by Protocol

--protocol filter Show only flows which match the given L4/L7 flow protocol (e.g. "udp", "http")

\$ hubble observe --pod deathstar --protocol http

```
May 4 13:23:40:501: default/tiefighter:42690 ->
default/deathstar-c74d84667-cx5kp:80 http-request FORWARDED
(HTTP/1.1 POST
http://deathstar.default.svc.cluster.local/v1/request-landing)
```

\$ hubble observe --namespace tenant-jobs --protocol dns

```
Aug 3 15:13:18.943: tenant-jobs/coreapi-767cf69fb8-cvqx1:53740
(ID:44253) -> kube-system/coredns-787d4945fb-6vfvfg:53 (ID:43153)
dns-request proxy FORWARDED (DNS Query
elasticsearch-master.tenant-jobs.svc.cluster.internal. AAAA)
```

Observe by Policy Verdict

--verdict filter Show only flows with this verdict [FORWARDED, DROPPED, AUDIT, REDIRECTED, ERROR, TRACED, TRANSLATED]

\$ hubble observe --pod deathstar --verdict DROPPED

```
May 4 13:23:47.852: default/xwing:42818 <-
default/deathstar-c74d84667-cx5kp:80 Policy denied DROPPED
(TCP Flags: SYN)
```

Observe by FQDN

--fqdn filter Show all flows related to the given fully qualified domain name (e.g. "*cilium.io").

--from-fqdn filter Show all flows originating at the given fully qualified domain name (e.g. "*ebpf.io").

--to-fqdn filter Show all flows terminating at the given fully qualified domain name (e.g. "*isovalent.com").

\$ hubble observe --to-fqdn api.github.com

```
Aug 3 15:12:13.929: tenant-jobs/crawler-56545d68f4-qch8:47180
(ID:21867) -> api.github.com:80 (world) policy-verdict:all
EGRESS ALLOWED (TCP Flags: SYN)
```

Observe by HTTP Method, Path and Status

--http-method filter Show only flows which match this HTTP method (e.g. "GET", "POST")

--http-path filter Show only flows which match this HTTP path regular expressions (e.g. "/page/{id}*")

--http-status filter Show only flows which match this HTTP status code prefix (e.g. "404", "5*")

\$ hubble observe --namespace tenant-jobs --http-path /applicants

```
Aug 3 15:16:18.351: tenant-jobs/resumes-86bbf46b88-n6mcn:51768
(ID:20880) <- tenant-jobs/coreapi-767cf69fb8-cvqx1:9080
(ID:44253) http-request FORWARDED (HTTP/1.1 POST
http://coreapi:9080/applicants)
```

\$ hubble observe --label app-resumes --http-method POST

```
Aug 3 15:16:28.591: tenant-jobs/resumes-86bbf46b88-n6mcn:51768
(ID:20880) <- tenant-jobs/coreapi-767cf69fb8-cvqx1:9080
(ID:44253) http-response FORWARDED (HTTP/1.1 200 33ms (POST
http://coreapi:9080/applicants))
```

A more complex example

Filters can be combined, too, the below example filters for flows of HTTP requests any pod with the label "app-core-api", where the HTTP path is "/applicants" and the HTTP method is "PUT"

```
$ hubble observe --namespace tenant-jobs --from-label
'app-coreapi' --protocol http --http-path /applicants
--http-method PUT
```

```
Aug 3 15:26:41.563: tenant-jobs/coreapi-767cf69fb8-cvqx1:49662
(ID:44253) -> tenant-jobs/elasticsearch-master-0:9200 (ID:16821)
http-request FORWARDED (HTTP/1.1 PUT
http://elasticsearch-master.tenant-jobs.svc.cluster.local:9200/
applicants/_create/827)
```

You can use the following argument to exclude data from results:

--not filter[=true] Reverses the next filter to be blacklist i.e. --not --from-ip 2.2.2.2

This example command ensures no flows from anything with a specific label are returned when viewing all flows from a namespace

```
$ hubble observe -n tenant-jobs --not --label app-coreapi
```

Formatting the output

--color string Colorize the output when the output format is one of 'compact' or 'dict'. The value is one of 'auto' (default), 'always' or 'never' (default "auto")

-o, --output string Specify the output format, one of: compact: Compact output dict: Each flow is shown as KEY=VALUE pair jsonpb: JSON encoded GetFlowResponse according to proto3's JSON mapping json: Alias for jsonpb table: Tab-aligned columns (default "compact")

--color string Colorize the output when the output format is one of 'compact' or 'dict'. The value is one of 'auto' (default), 'always' or 'never' (default "auto")

--ip-translation Translate IP addresses to logical names such as pod name, FQDN, ... (default true)

- `cilium-dbg bpf ct`
- `cilium-dbg bpf nat`
- `cilium-dbg bpf lb`

Direct access to local BPF maps

- `cilium-dbg sysdump`

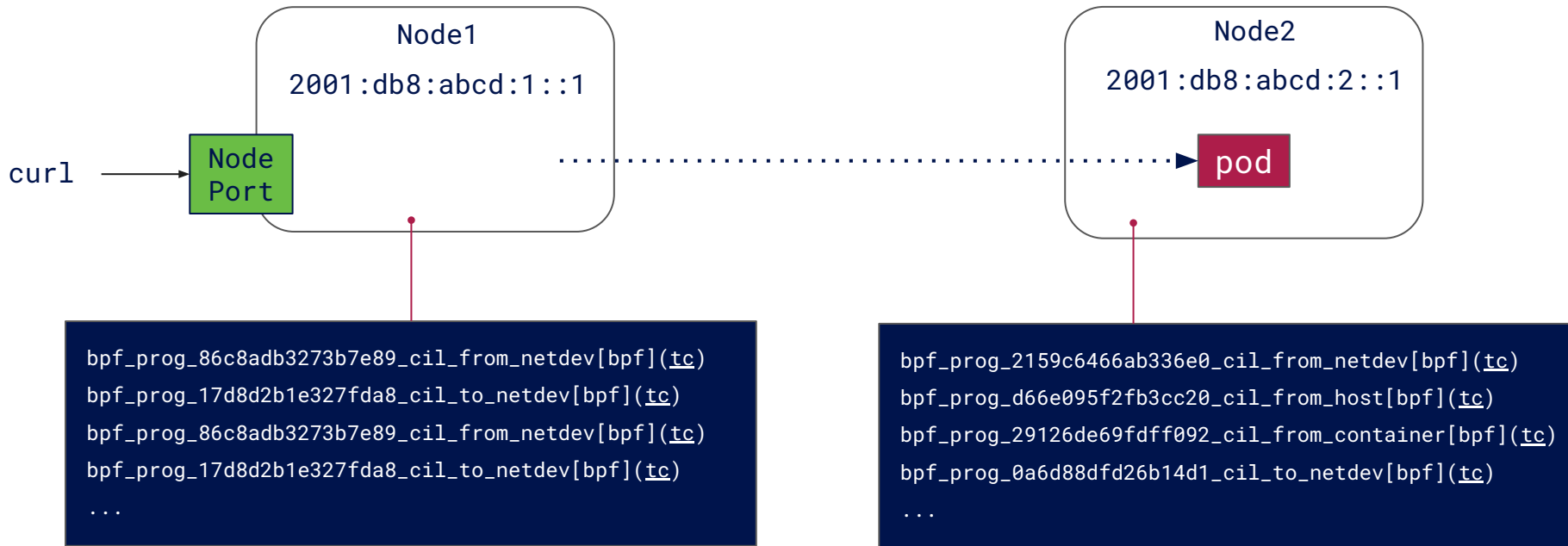
Useful for creating Github issue in Cilium repo

Problem #5: hardcoded IPv4 cilium-dbg

- `cilium-dbg bpf nat`

```
Error: Unable to open
/sys/fs/bpf/tc/globals/cilium_snat_v4_external: loading pinned
map /sys/fs/bpf/tc/globals/cilium_snat_v4_external: no such file
or directory
> Error while running 'cilium-dbg bpf nat list': exit status 1
```

pwru (packet, where are you?)



Takeaways And next steps



KubeCon



CloudNativeCon

North America 2025

Takeaways



As of Cilium 1.18, IPv6 only cluster with tunnelling is supported !

- Multiple production deployment at TikTok with this setup






Working with the cilium team and the open-source community is awesome !



More love for ipv6 by making more tools work flawlessly out-of-the-box

Takeaways

-  Started Cilium 1.12 in 2023, production ready in 1.18 in 2025
Some work is worth pursuing !
-  Encapsulation/Tunnel mode is a lot simpler to run (and still provide good perf!)
-  Debugging (in order): hubble (and its UI), connectivity end-to-end test, cilium-dbg and pwru

Next steps



Use more Cilium features:

- Transparent encryption
- Cluster mesh
- ingress



Investigate [Cilium BGP Control Plane](#) to enable Native routing

Acknowledgements



Shubham Jain

shubham@sjain.dev

sjain.dev



Paul Chaignon

 pchaigno



Bill Mulligan

 xmulligan



Julian Wiedmann

 julianwiedmann

Kapil Agrawal's CiliumCon talk about ipv6-only with native routing
[IPv6 First, Not Just Ready: Kubernetes Without IPv4 Using Cilium at ESnet - Kapil Agrawal, ESnet](#)



KubeCon



CloudNativeCon

North America 2025



Questions or feedbacks ?

Please ask us !

