The inconspicuous role of **conntrack** in Kubernetes networking

Ricard Bejarano
Lead Site Reliability Engineer

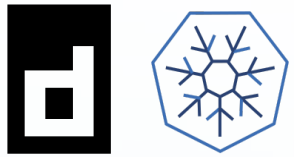
SREcon26 Americas


ЯДЕРНОЕ ДЕЛЕНИЕ
СТЕРЖНИ УПРАВЛЕНИЯ
ВОДА
НЕГАТИВНЫЙ КОЭФФИЦИЕНТ ТЕМПЕРАТУРЫ
ОТРАВЛЕНИЕ КСЕНОНОМ
ОТРАВЛЕНИЕ КСЕНОНОМ

Let's set a bit of a **system prompt**

Kubernetes
(upstream)

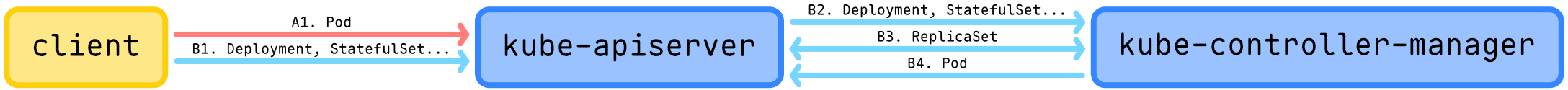
Linux
(5.x or above)



Common runtimes
(like containerd or CRI-O)

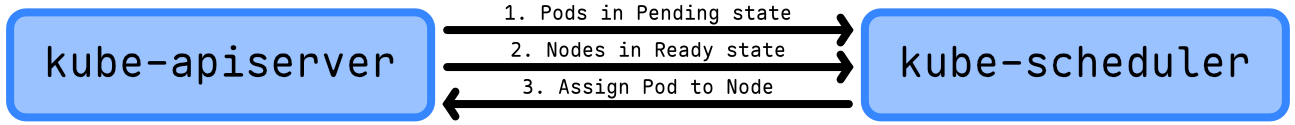
Before we begin...

Requesting a Pod



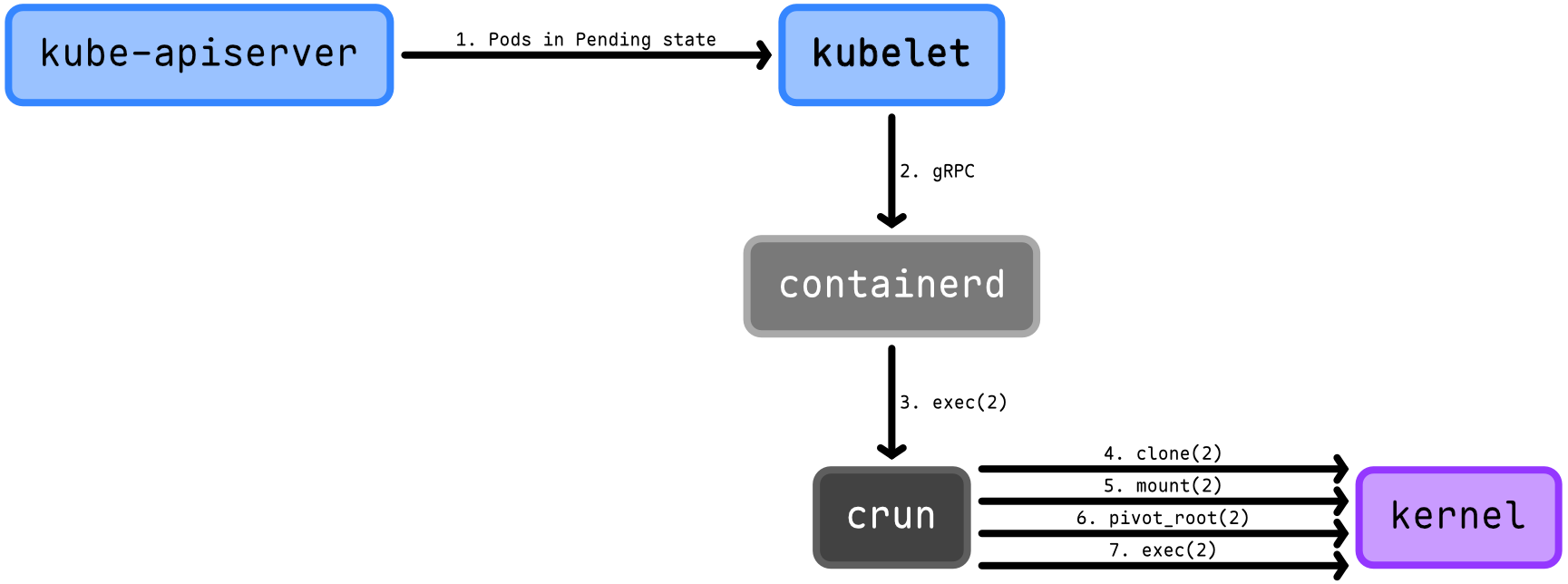
Before we begin...

Scheduling a Pod



Before we begin...

Provisioning a Pod



Our pod is **Running**

```
$ kubectl get pod

NAME                    READY   STATUS    RESTARTS   AGE
foo-6f9c7d8b6f-xyz12    1/1     Running   0          12s
```

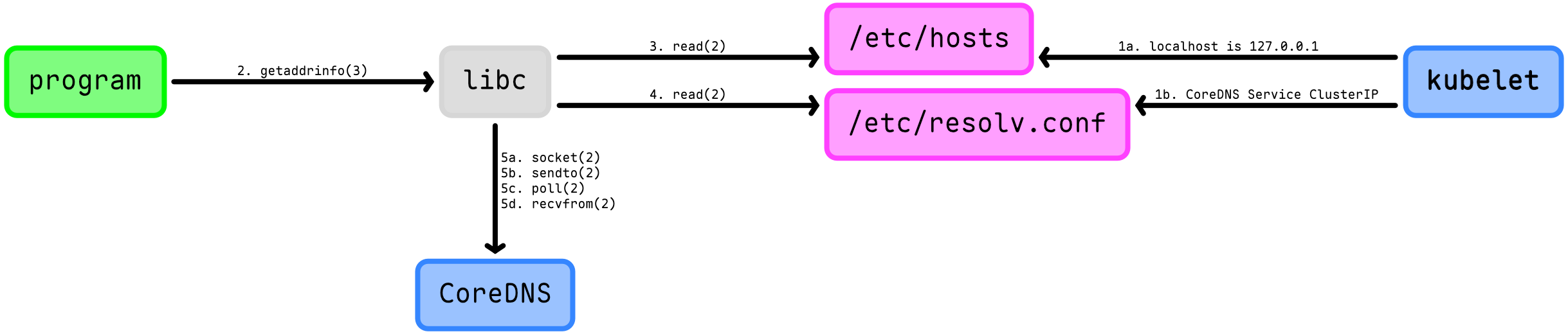

cisco

2025-09-04T14:12:01Z [ERROR] read udp REDACTED:42999->REDACTED:53: i/o timeout
2025-09-04T14:12:01Z [ERROR] read udp REDACTED:26046->REDACTED:53: i/o timeout
2025-09-04T14:12:01Z [ERROR] read udp REDACTED:59401->REDACTED:53: i/o timeout
2025-09-04T14:12:02Z [ERROR] read udp REDACTED:40757->REDACTED:53: i/o timeout
2025-09-04T14:12:03Z [ERROR] read udp REDACTED:18235->REDACTED:53: i/o timeout
2025-09-04T14:12:03Z [ERROR] read udp REDACTED:21225->REDACTED:53: i/o timeout
2025-09-04T14:12:03Z [ERROR] read udp REDACTED:43711->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:42821->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:43706->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:19688->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:48784->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:30583->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:25345->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:53407->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:41013->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:23084->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:40908->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:23099->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:11909->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:44784->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:38213->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:19995->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:55447->REDACTED:53: i/o timeout

September 4th, 2025

How does name resolution work?

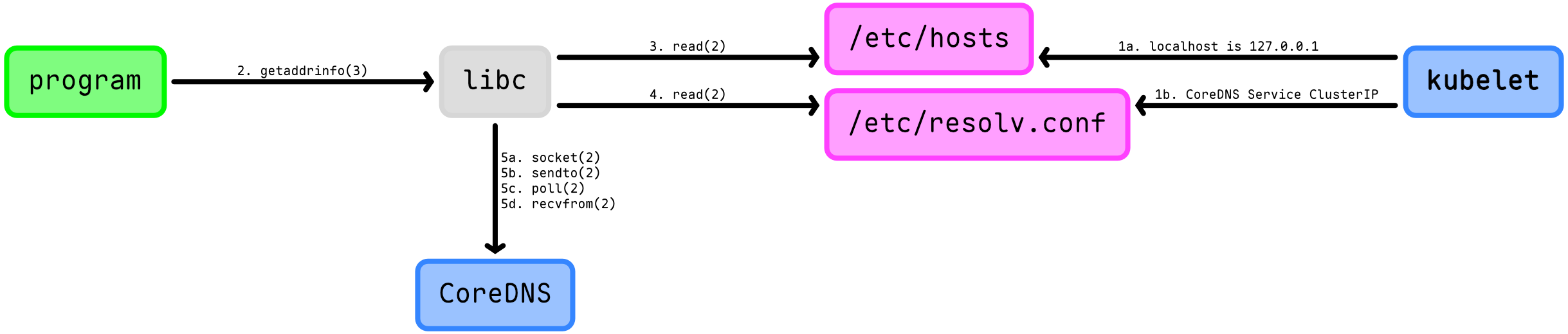
getaddrinfo(3)



Non-standard behavior **hurts my feelings**

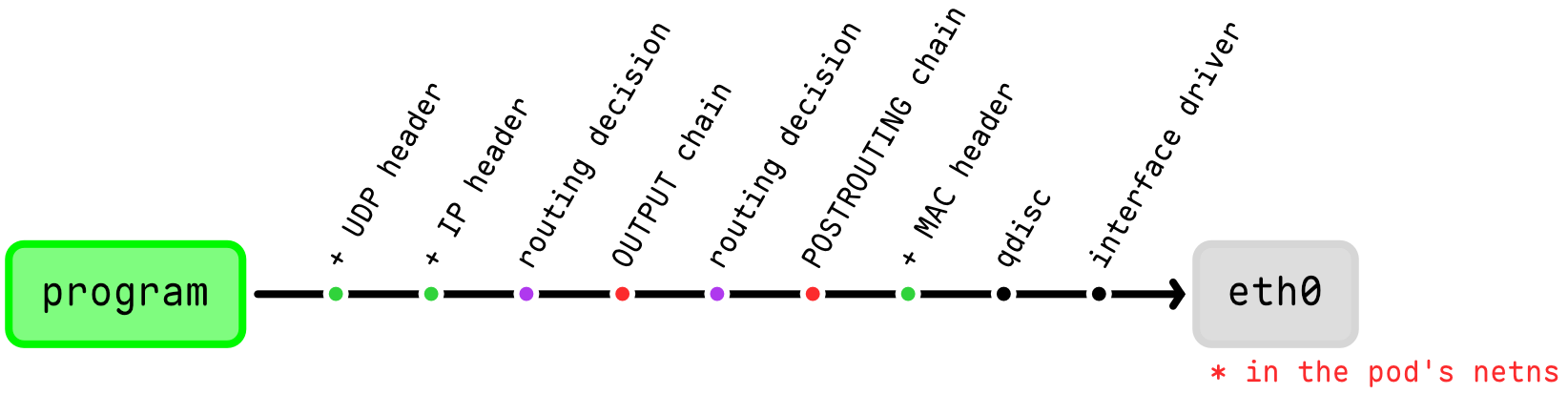
How does name resolution work?

getaddrinfo(3)

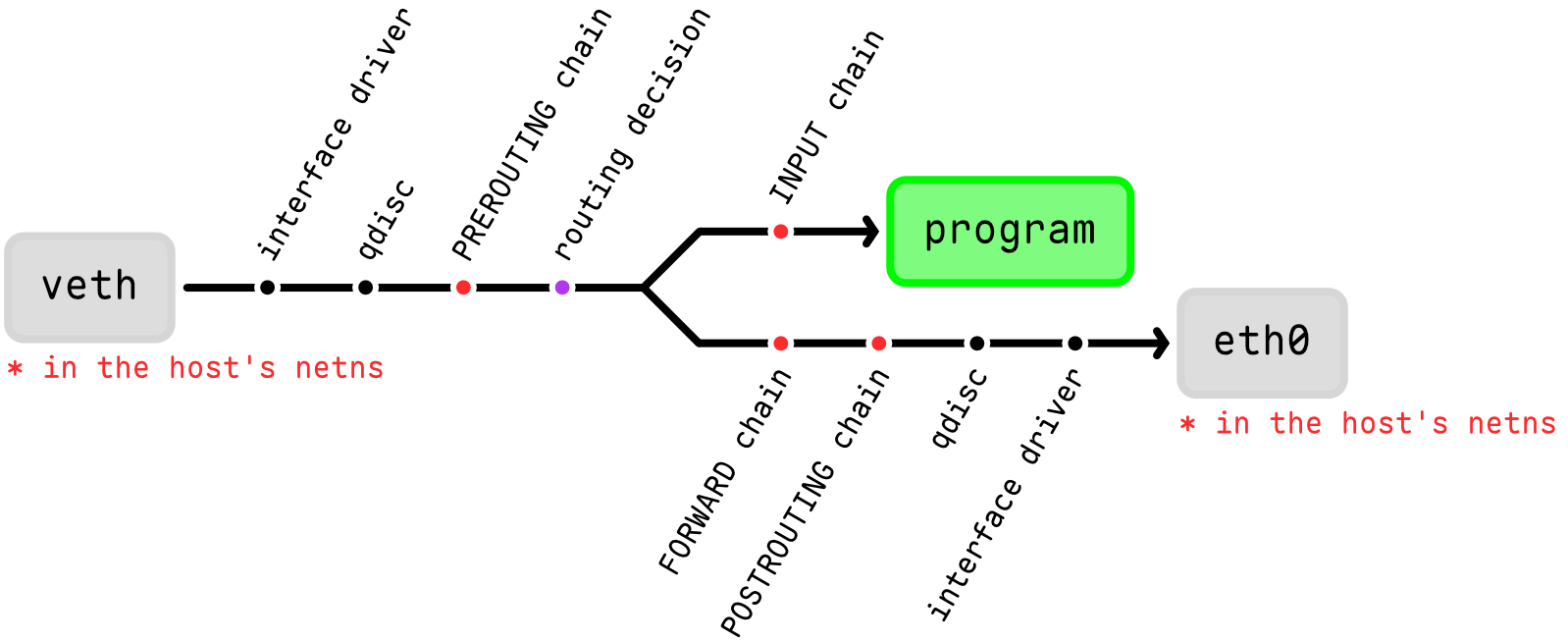


How does routing work?

Egress path



Ingress path



```
Chain KUBE-SERVICES
 target             prot  opt  in  out  source     destination
 KUBE-SVC-64172328  udp   --   *   *    0.0.0.0/0  10.96.0.10   udp dpt:53

Chain KUBE-SVC-64172328
 target             prot  opt  in  out  source     destination
 KUBE-SEP-04AD2884  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 0.3333
 KUBE-SEP-E8CE23BB  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 0.5000
 KUBE-SEP-152964BC  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 1.0000

Chain KUBE-SEP-04AD2884
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.12.34:53

Chain KUBE-SEP-E8CE23BB
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.51.16:53

Chain KUBE-SEP-152964BC
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.3.7:53
```

The KUBE-SERVICES iptables chain

cisco

```
Chain KUBE-SERVICES
 target             prot  opt  in  out  source     destination
 KUBE-SVC-64172328  udp   --   *   *    0.0.0.0/0  10.96.0.10   udp dpt:53

Chain KUBE-SVC-64172328
 target             prot  opt  in  out  source     destination
 KUBE-SEP-04AD2884  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 0.3333
 KUBE-SEP-E8CE23BB  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 0.5000
 KUBE-SEP-152964BC  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 1.0000

Chain KUBE-SEP-04AD2884
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.12.34:53

Chain KUBE-SEP-E8CE23BB
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.51.16:53

Chain KUBE-SEP-152964BC
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.3.7:53
```

The KUBE-SVC iptables chain

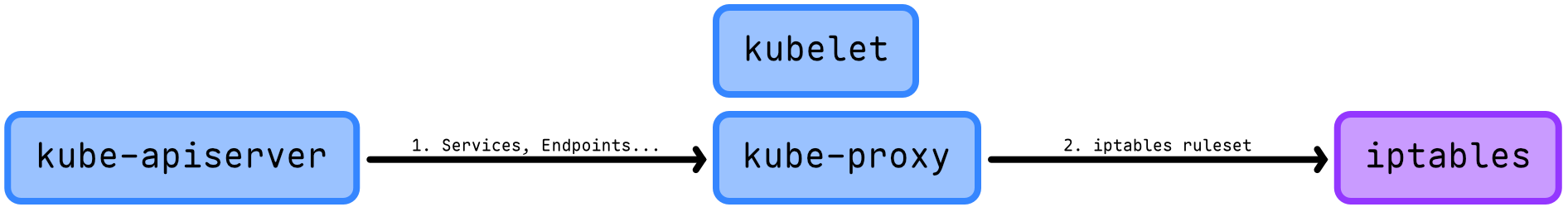
cisco

The KUBE-SEP iptables chains

```
Chain KUBE-SERVICES
 target             prot  opt  in  out  source     destination
 KUBE-SVC-64172328  udp   --   *   *    0.0.0.0/0  10.96.0.10   udp dpt:53

Chain KUBE-SVC-64172328
 target             prot  opt  in  out  source     destination
 KUBE-SEP-04AD2884  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 0.3333
 KUBE-SEP-E8CE23BB  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 0.5000
 KUBE-SEP-152964BC  udp   --   *   *    0.0.0.0/0  0.0.0.0/0    random probability 1.0000

Chain KUBE-SEP-04AD2884
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.12.34:53

Chain KUBE-SEP-E8CE23BB
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.51.16:53

Chain KUBE-SEP-152964BC
 target             prot  opt  in  out  source     destination
 DNAT               udp   --   *   *    0.0.0.0/0  0.0.0.0/0    to:10.0.3.7:53
```


cisco

Introducing kube-proxy

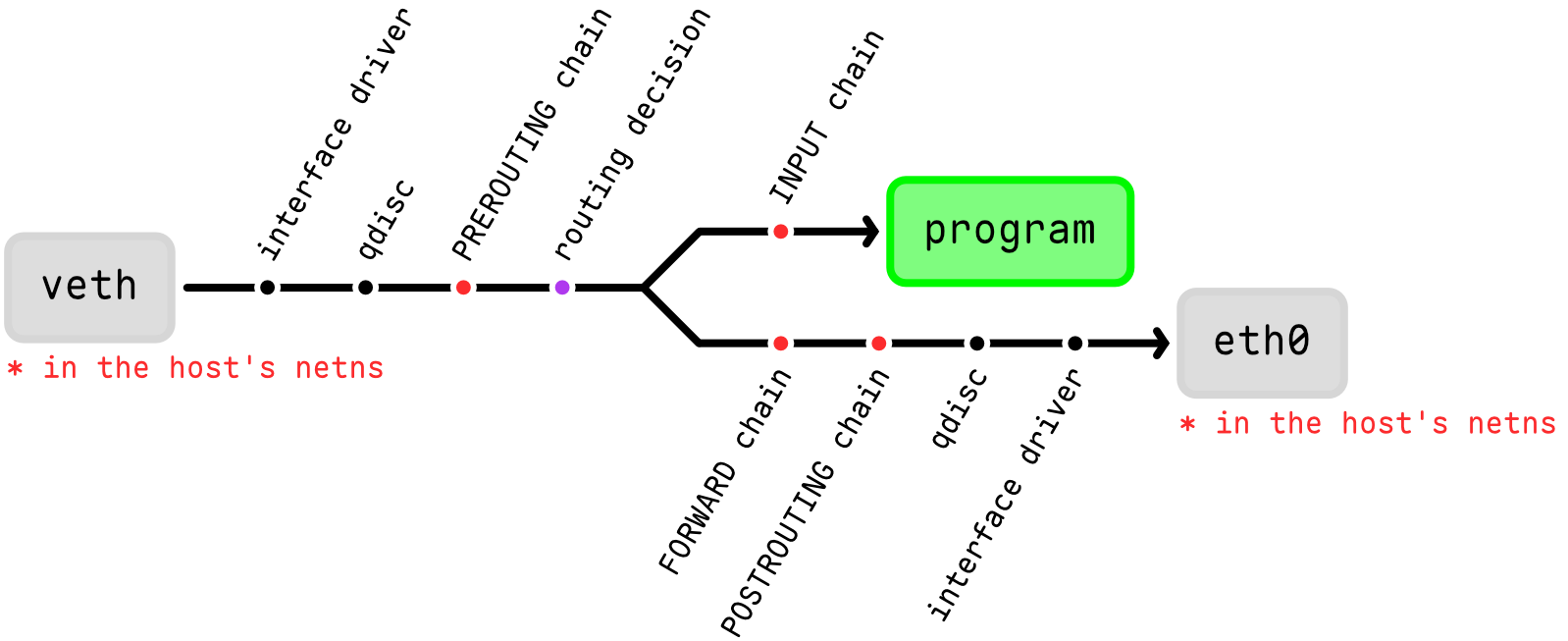


Cilium

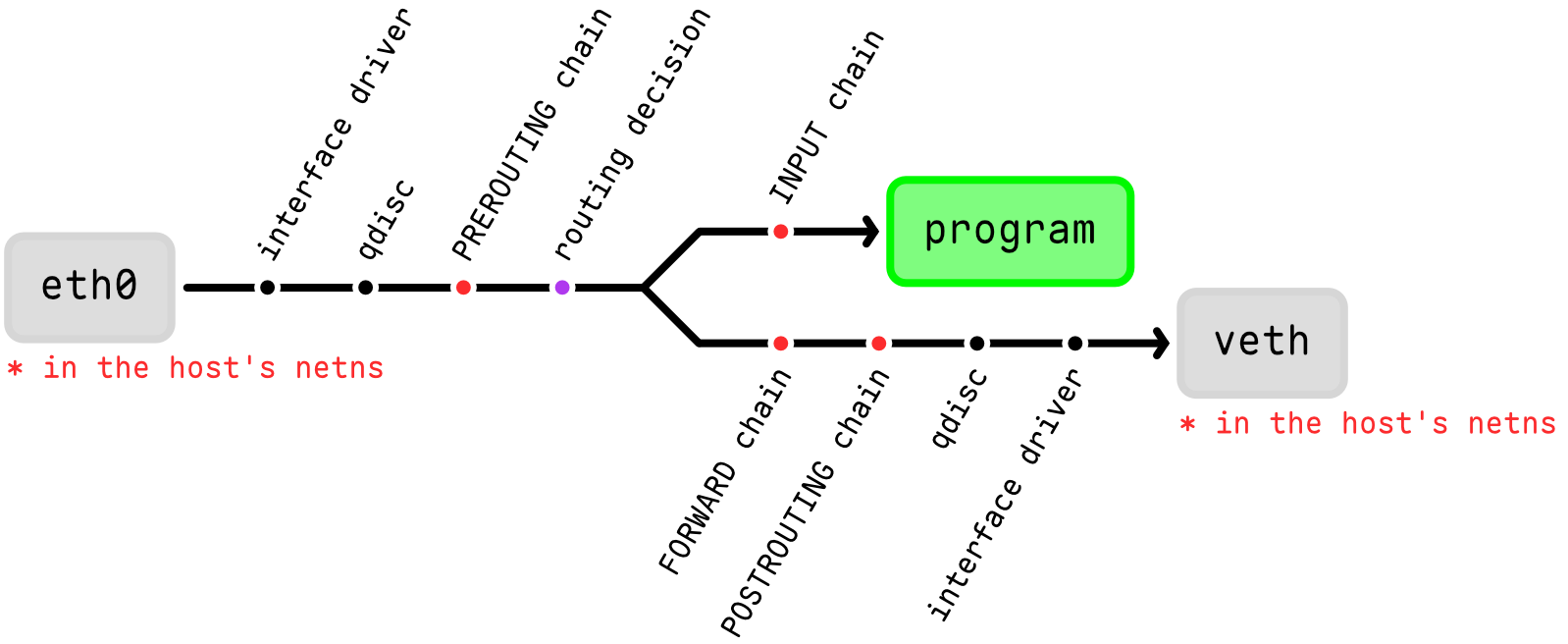
Map-based lookups. Instead of matching packets' destination IP addresses to a Service sequentially in the KUBE-SERVICES iptables chain.

eBPF data path. Can be configured to bypass iptables and replace kube-proxy completely.

Ingress path



Ingress path (again)



2025-09-04T14:12:01Z [ERROR] read udp REDACTED:42999->REDACTED:53: i/o timeout
2025-09-04T14:12:01Z [ERROR] read udp REDACTED:26046->REDACTED:53: i/o timeout
2025-09-04T14:12:01Z [ERROR] read udp REDACTED:59401->REDACTED:53: i/o timeout
2025-09-04T14:12:02Z [ERROR] read udp REDACTED:40757->REDACTED:53: i/o timeout
2025-09-04T14:12:03Z [ERROR] read udp REDACTED:18235->REDACTED:53: i/o timeout
2025-09-04T14:12:03Z [ERROR] read udp REDACTED:21225->REDACTED:53: i/o timeout
2025-09-04T14:12:03Z [ERROR] read udp REDACTED:43711->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:42821->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:43706->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:19688->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:48784->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:30583->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:25345->REDACTED:53: i/o timeout
2025-09-04T14:12:04Z [ERROR] read udp REDACTED:53407->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:41013->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:23084->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:40908->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:23099->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:11909->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:44784->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:38213->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:19995->REDACTED:53: i/o timeout
2025-09-04T14:12:05Z [ERROR] read udp REDACTED:55447->REDACTED:53: i/o timeout

September 4th, 2025

A CoreDNS pod was OOMKilled

```
$ kubectl describe pod coredns-6f9c7d8b6f-xyz12

Name:         coredns-6f9c7d8b6f-xyz12
Namespace:    kube-system
Node:         ip-10-0-12-34.ec2.internal/10.0.12.34
Start Time:   Thu, 4 Sep 2025 14:22:11 +0000
Labels:       app=coredns
              pod-template-hash=6f9c7d8b6f
Status:       Running
IP:           10.244.3.45
Controlled By:  ReplicaSet/coredns-6f9c7d8b6f

Containers:
  coredns:
    Container ID:   docker://a1b2c3d4e5f67890123456789abcdef
    Image:          docker.io/coredns:latest
    Port:           53/UDP
    State:          Running
      Started:      Thu, 4 Sep 2025 14:25:03 +0000
    Last State:     Terminated
      Reason:       OOMKilled
      Exit Code:    137
      Started:      Thu, 4 Sep 2025 14:23:10 +0000
      Finished:     Thu, 4 Sep 2025 14:24:58 +0000
    Ready:          True
```

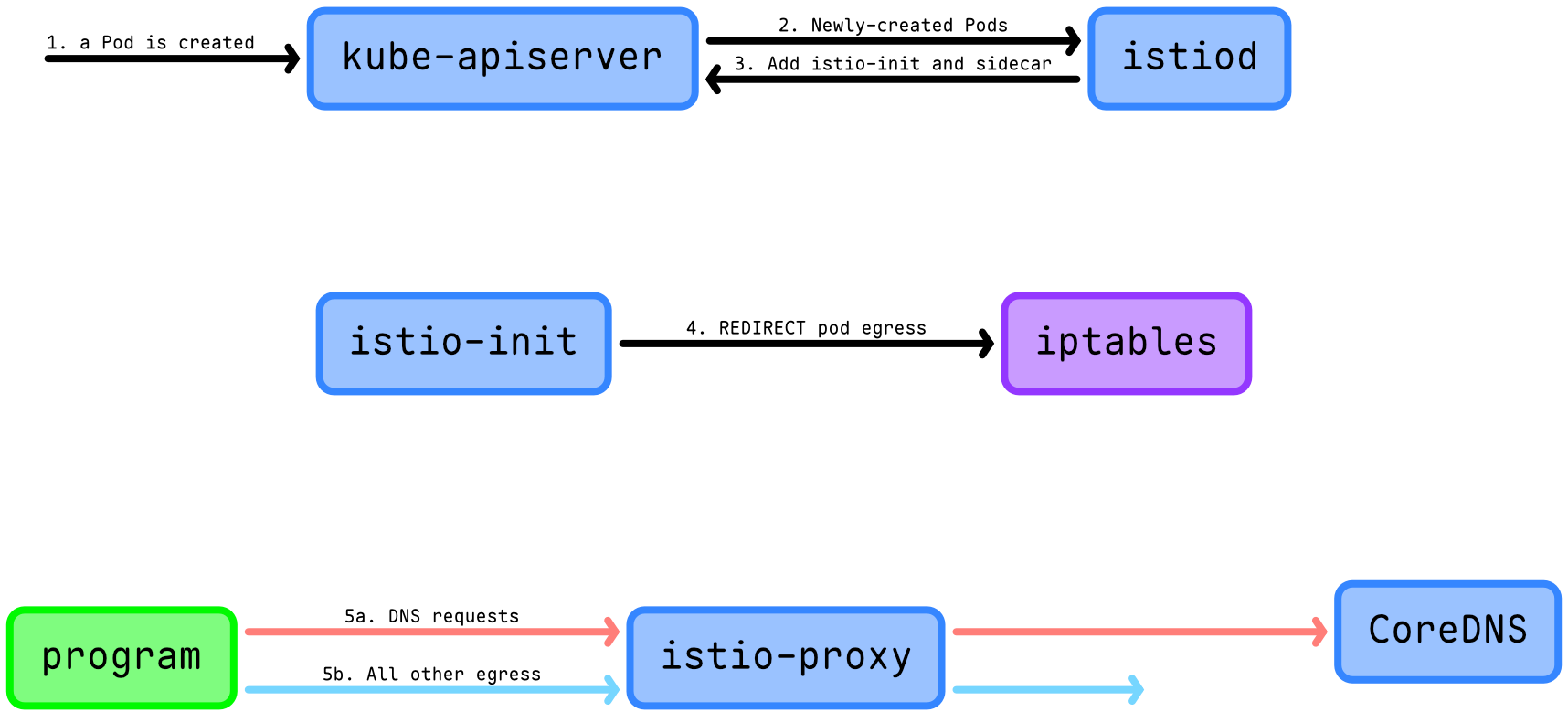
Our second agent of chaos

Istio

Our service mesh of choice. Makes some very complex things simple, but some very simple things complex.

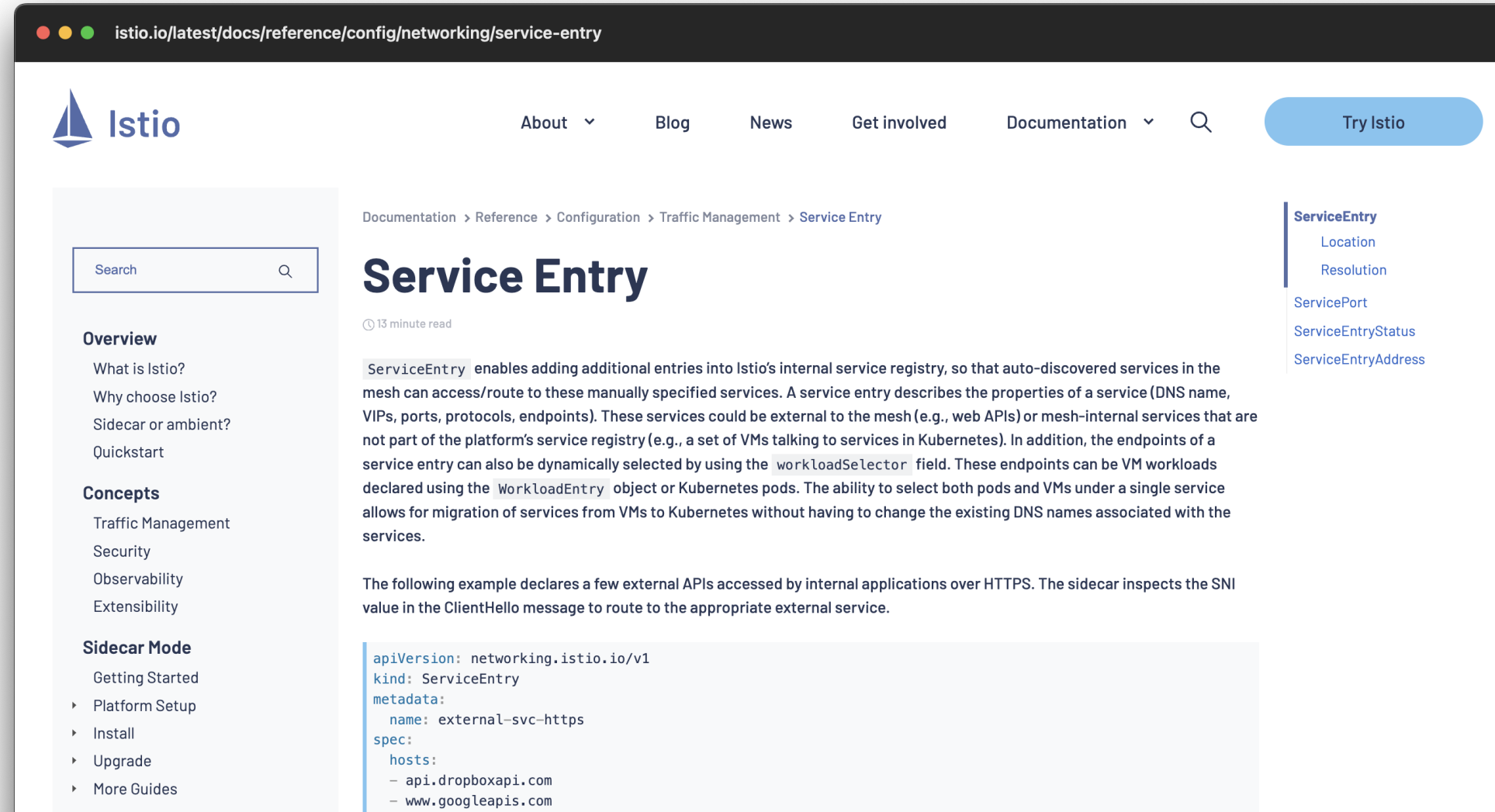
Sidecar mode. Pods get an Istio sidecar container run alongside them.

How does Istio work?

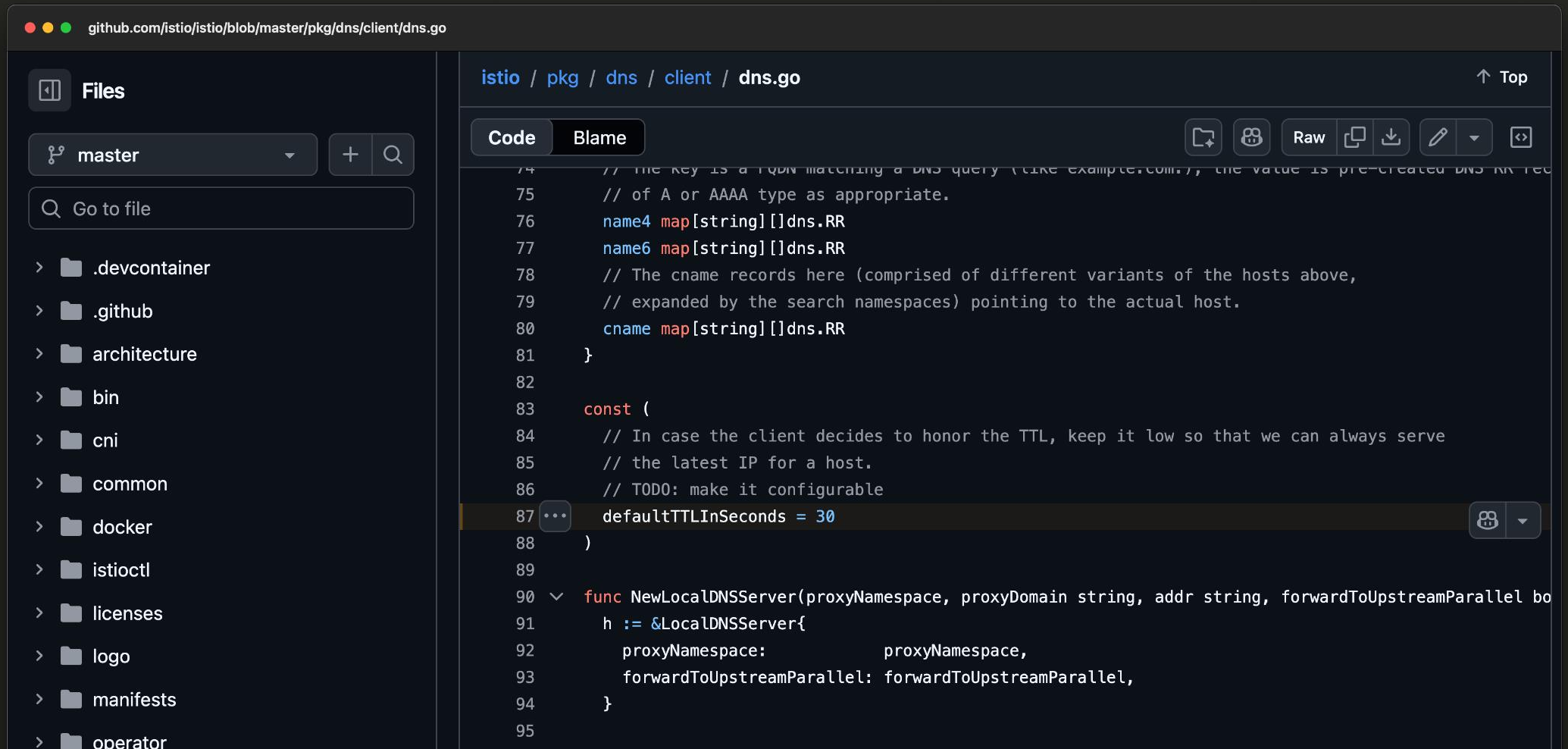


Our second agent of chaos: Istio

Istio Service Entries



This interval is **hard-coded**



Our third agent of chaos

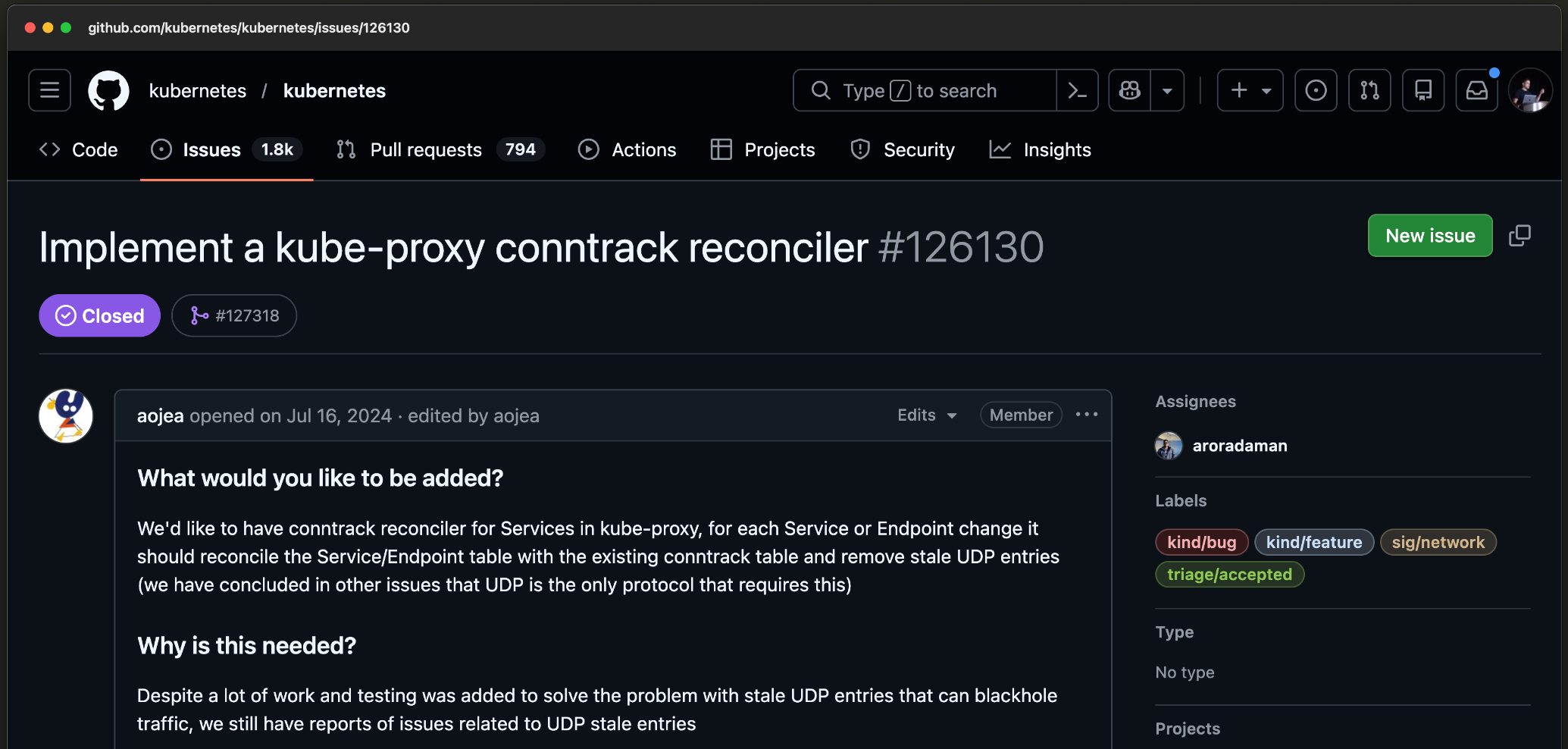
conntrack

```
$ sudo conntrack -L
tcp  6 431999 ESTABLISHED src=192.168.1.10 dst=1
tcp  6 120 ESTABLISHED src=192.168.1.11 dst=172.
tcp  6 98 TIME_WAIT src=192.168.1.12 dst=151.101
udp  17 27 src=192.168.1.13 dst=8.8.8.8 sport=53
tcp  6 300 ESTABLISHED src=10.0.0.5 dst=93.184.2
icmp 1 25 src=192.168.1.14 dst=1.1.1.1 type=8 co
tcp  6 250 ESTABLISHED src=192.168.1.15 dst=104.
tcp  6 200 ESTABLISHED src=192.168.1.16 dst=151.
udp  17 28 src=192.168.1.17 dst=8.8.4.4 sport=53
tcp  6 180 TIME_WAIT src=192.168.1.18 dst=151.10
tcp  6 175 ESTABLISHED src=192.168.1.19 dst=172.
udp  17 30 src=192.168.1.20 dst=1.1.1.1 sport=60
tcp  6 160 ESTABLISHED src=192.168.1.21 dst=93.1
tcp  6 155 ESTABLISHED src=192.168.1.22 dst=104.
udp  17 29 src=192.168.1.23 dst=9.9.9.9 sport=53
tcp  6 150 TIME_WAIT src=192.168.1.24 dst=151.10
tcp  6 145 ESTABLISHED src=192.168.1.25 dst=172.
udp  17 28 src=192.168.1.26 dst=8.26.56.26 sport
tcp  6 140 ESTABLISHED src=192.168.1.27 dst=151.
tcp  6 135 ESTABLISHED src=192.168.1.28 dst=13.1
udp  17 27 src=192.168.1.29 dst=208.67.222.222 s
tcp  6 130 TIME_WAIT src=192.168.1.30 dst=151.10
tcp  6 125 ESTABLISHED src=192.168.1.31 dst=104.
```

Sets a *connection state* attribute. Matches packets to active connections based on other packets it has seen.

Prevents full iptables re-evaluation for packets of known established connections, saving CPU time waste.

Kubernetes issue #126130



Istio

Extremely **aggressive** cache refresh mechanism.

Reuse of the same client port for all DNS requests.

kube-proxy

Lack of a **cleanup mechanism** for **stale conntrack entries** of terminated pods.

Fixed in v1.32.

conntrack

Default bidirectional UDP stream **timeout** setting of **2 minutes**.

The inconspicuous role of conntrack in Kubernetes networking

Ricard Bejarano

Lead Site Reliability Engineer
Cisco ThousandEyes

I'm **writing a book** about homelabbing!

O'REILLY®

bejarano.io/the-homelab-handbook

Meet me if you have a homelab!

cisco