KCD
GROWING CLOUD NATIVE TOGETHER
INDONESIA 2024

Google Cloud | NUTANIX | Red Hat | Alibaba Cloud | aws

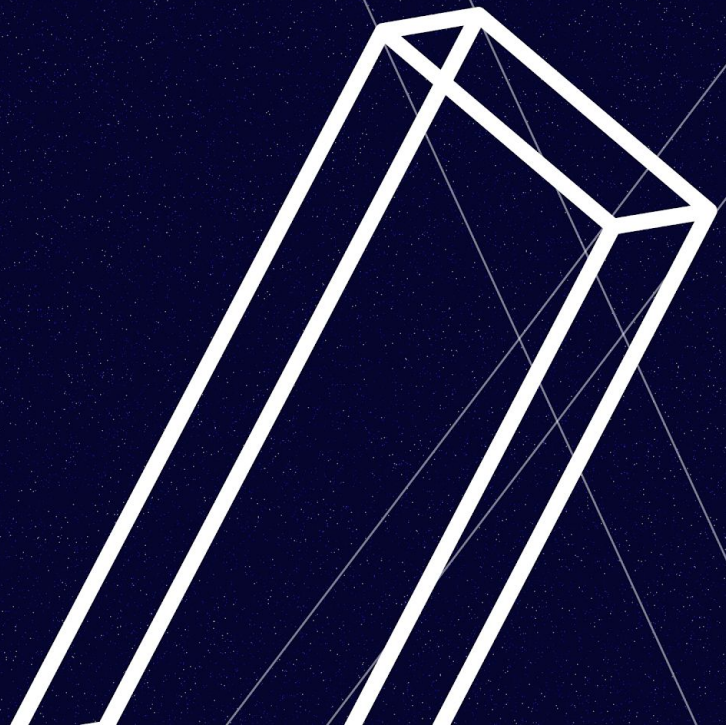
Breakdown the Black Magic of Cilium

Aldin Setiawan

Majutsu engineering

Disclaimer

- Open discussion.
- There is no vendor comparison.



AGENDA

Section	Part 1
Problem	00
Cilium CNI?	01
The Journey of netfilter	02
Breakdown the Black magic of Cilium	03

Problem

Our team have problem with High Latency in k8s

Research

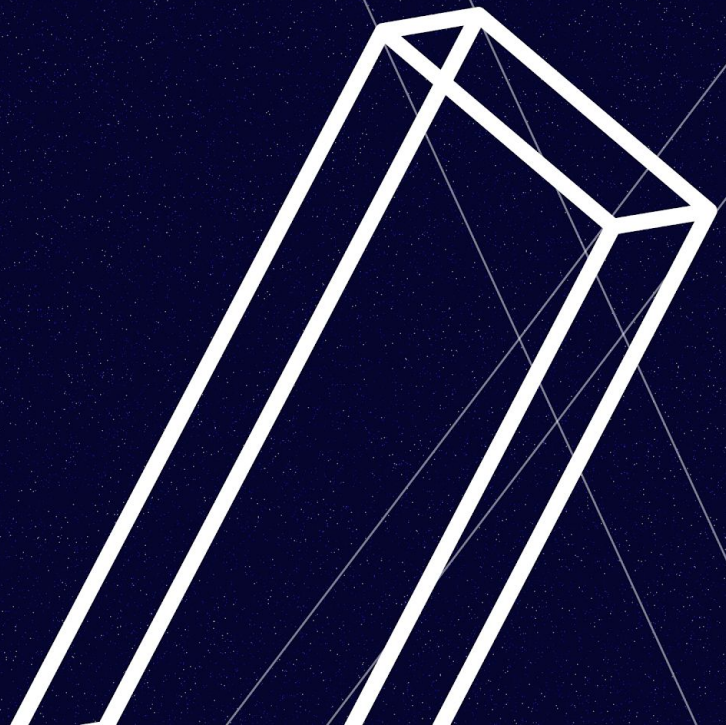
Find alternative CNI or Tuning mumbo jumbo(?)

~~Deploying~~

Solve the step 1 or back to step 1

The Problem

- Big latency between service
- Often get “connection reset by peer”
- Another latency



Research

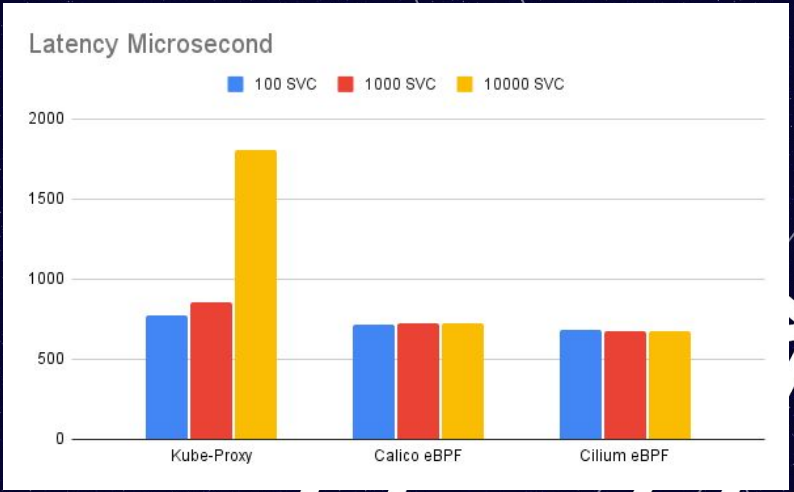
One of our wizard tech recommend us to try Cilium

No one ask “why cilium” because the wizard already spoken

Tbh I little skeptic about his whisper, as far as i know k8s CNI is only about Tunneling or Routing

Research Result

Higher is better



Lower is better

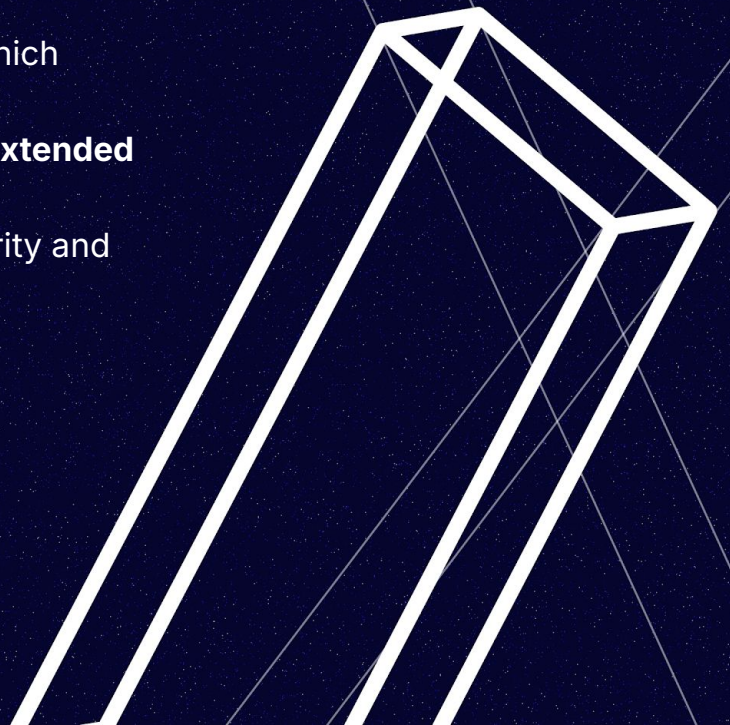
Credit research
@Jhody Sekardono

Research Result

My honest reaction

Cilium CNI

Cilium CNI is a powerful networking plugin for Kubernetes which provides enhanced security and networking capabilities for containerised applications. It leverages the power of **eBPF (extended Berkeley Packet Filter)**, a highly efficient and programmable kernel-level technology, to deliver transparent network security and traffic monitoring features.



eBPF

eBPF technology actually pretty old(1992) and already exists long time ago, the original of eBPF is Berkeley Packet Filter(BPF) or originally for packet filtering (tcpdump), eBPF has evolved into a versatile framework for observing and extending kernel behavior without requiring custom kernel modules

tl;dr

Think of eBPF as the "secret sauce" for Linux. It’s like that one app that can do literally everything, but instead of just running on your phone, it’s vibing straight in the kernel (the brain of the operating system). 💡🔥🔥🔥

“the packet should be filtered ‘in place’ (e.g.,where the network interface DMA engine put it)”

eBPF

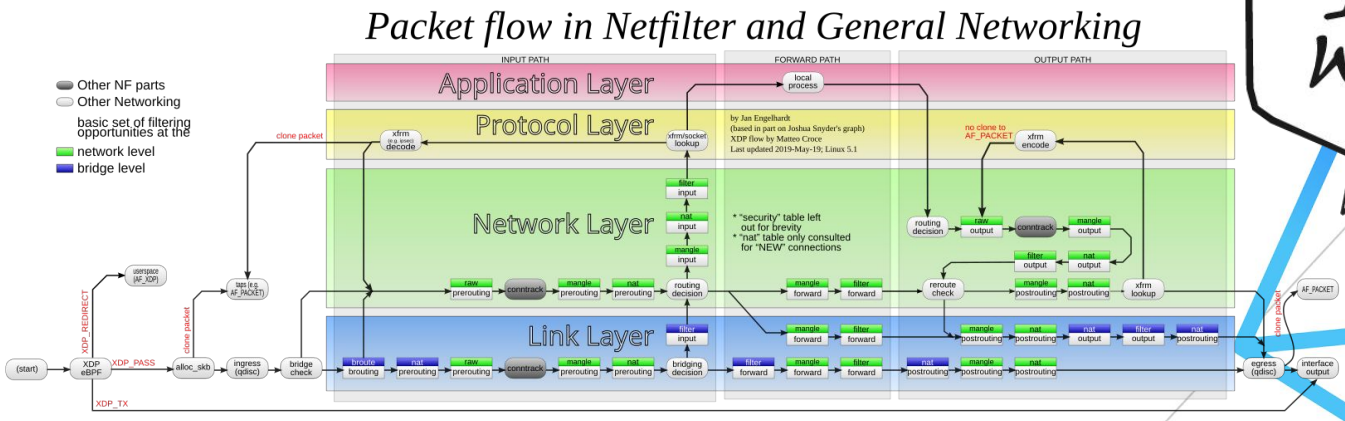
Why package filter technology can solve latency issues?

What the correlation between app inside kernel with cni?

Why other cni look more slow than cilium?

is mayonnaise an instrument?

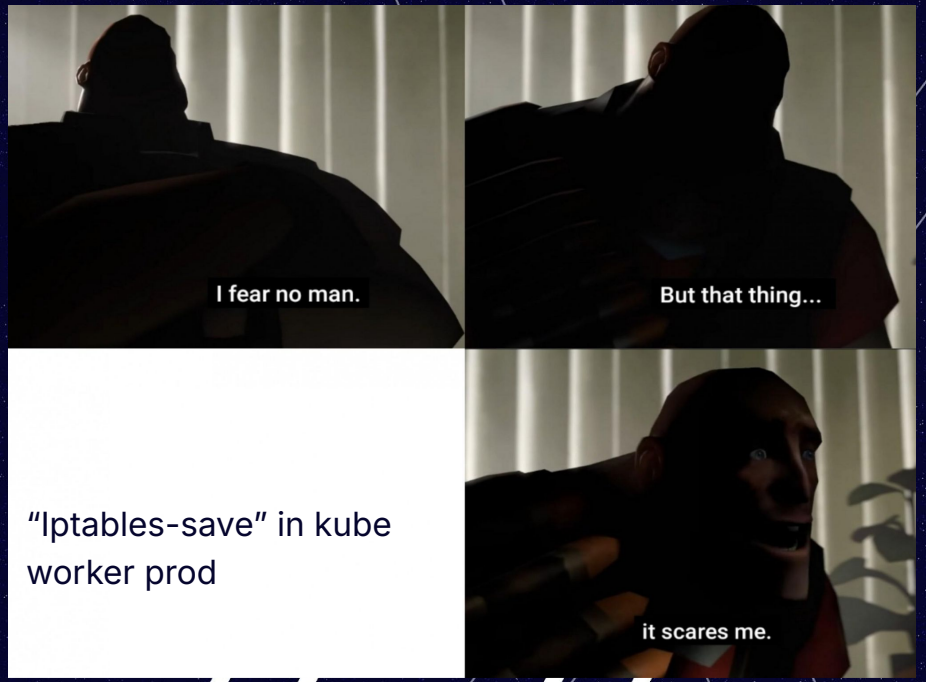
Iptables, our lovely tools



Over the years, iptables has been a blessing and a curse: a blessing for its flexibility and quick fixes. A curse during times debugging a 5K rules iptables setup in an environment where multiple system components are fighting over who gets to install what iptables rules. (cilium blog)

Iptables, our lovely tools

Iptables incompatible with many rule, i.e when doing scale up/down pod iptables will **rewrite** entire rule or when pod communication to svc they will create many conntrack



Too much yapping, show the rizz

Let’s tap the veth-peer interface
with tcpdump to see the pkt

Count the linux conntrack

Too much yapping, show the rizz

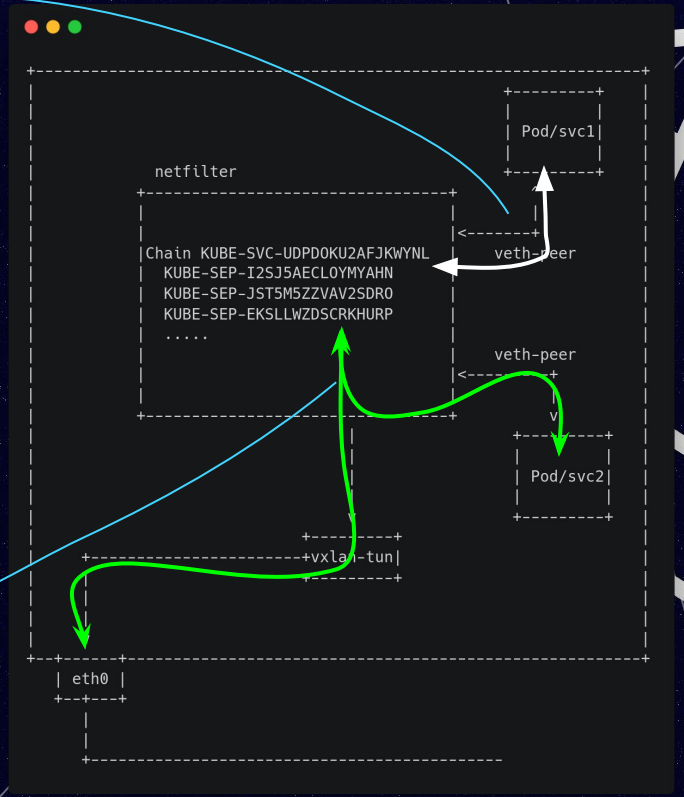
🗣️🗣️🗣️

```
curl 'https://asciinema.humanz.moe/a/MqiMLpfEdYW0hsIgnqXNOwkMF.cast?dl=1'\
-o /tmp/cast1;asciinema play /tmp/cast1
```

Too much yapping, show the rizz

First pkt was send through veth peer interface into netfiter with dst cluster ip

When pkt arrive in netfilter pkt will be load balancing(dnat) and trigger the conntrack



Ready to Breakdown the Black magic of Cilium?

Since we already know the downside of iptables now let's move into Cilium

Bro is Cooking 🗣️🔥🔥🔥

```
curl 'https://asciinema.humanz.moe/a/bC0KPiKnfNpDKu4QShHdfLTic.cast?dl=1'\
-o /tmp/cast2;asciinema play /tmp/cast2
```

Bro is Cooking 🗣️🔥🔥🔥

Here was the ebpf action, when the pkt move into netfilter the dst already changed into pod ip

```
get pods -o wide
                                                  READY   STATUS      RESTARTS   AGE     IP              NODE                                                NOMINATED NODE   READINES
d564fc-4m7xj                                      1/1     Running     0          30d     10.0.2.246      kube-44ae8-default-worker-klxsw-t9swq-9rbrw         <none>           <none>
d564fc-ncdgg                                      1/1     Running     0          30d     10.0.1.102      kube-44ae8-default-worker-klxsw-t9swq-zkndh         <none>           <none>
d564fc-x4qkh                                      1/1     Running     0          30d     10.0.2.127      kube-44ae8-default-worker-klxsw-t9swq-9rbrw         <none>           <none>
                                                  1/1     Running     0          30d     10.0.2.205      kube-44ae8-default-worker-klxsw-t9swq-9rbrw         <none>           <none>
e-44ae8-default-worker-klxsw-t9swq-9rbrw-z66sv    1/1     Running     0          7m51s   172.16.0.150    kube-44ae8-default-worker-klxsw-t9swq-9rbrw         <none>           <none>
e-44ae8-default-worker-klxsw-t9swq-zkndh-rb7j4    1/1     Running     0          14h     172.16.0.214    kube-44ae8-default-worker-klxsw-t9swq-zkndh         <none>           <none>
e-44ae8-sbfkm-51m72-bfzgd                         0/1     Completed   0          14h     172.16.0.133    kube-44ae8-sbfkm-51m72                              <none>           <none>

 humanz in /mnt/Data/Kube on (master)xxx
 get svc
        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
terIP   10.254.224.248   <none>        80/TCP    30d
terIP   10.254.0.1       <none>        443/TCP   30d

 humanz in /mnt/Data/Kube on (master)xxx
```

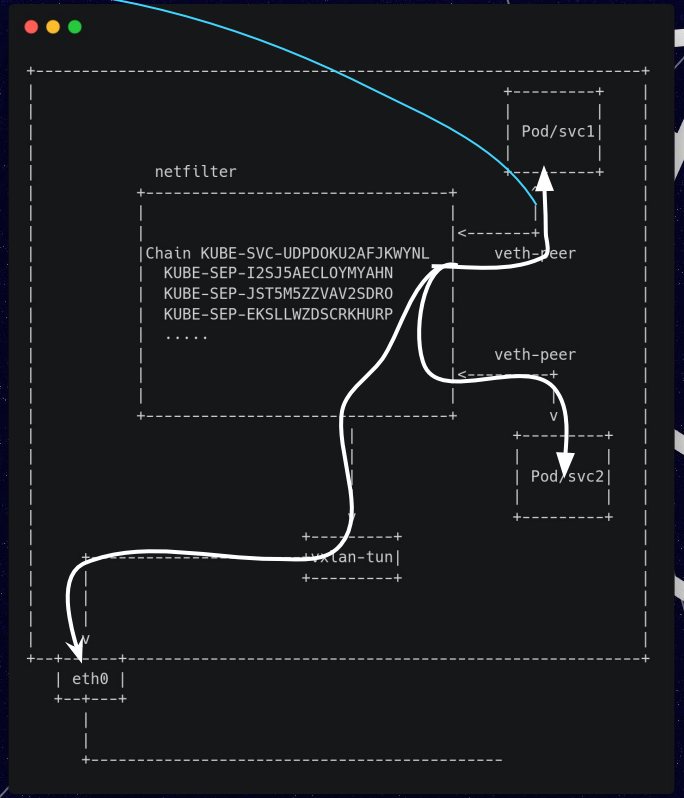
```
efault-worker-klxsw-t9swq-9rbrw:/# ip link | grep 13:
b9@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8905 qdisc noqueue state UP mode DEFAULT group default qlen 1000
efault-worker-klxsw-t9swq-9rbrw:/# tcpdump -ni lxc938b27f034b9 tcp -c 2
output suppressed, use -v[v]... for full protocol decode
38b27f034b9, link-type EN10MB (Ethernet), snapshot length 262144 bytes
P 10.0.2.205.59114 > 10.0.1.102.80: Flags [S], seq 1042711228, win 61705, options [mss 8815,sackOK,TS val 1924115373 ecr
 length 0
P 10.0.1.102.80 > 10.0.2.205.59114: Flags [S.], seq 3586449470, ack 1042711229, win 61621, options [mss 8815,sackOK,TS v
 1924115373,nop,wscale 7], length 0
d
ed by filter
 by kernel
efault-worker-klxsw-t9swq-9rbrw:/#
```

```
netshoot-1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueu
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:0
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
    link/ether ee:87:35:1c:8e:df brd ff:ff:ff:ff:ff:
    inet 10.0.2.205/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::ec87:35ff:fe1c:8edf/64 scope link
       valid_lft forever preferred_lft forever
netshoot-1:~# curl -I 10.254.224.248
HTTP/1.1 200 OK
Server: nginx/1.27.2
Date: Fri, 22 Nov 2024 09:53:05 GMT
Content-Type: text/html
```

Bro is Cooking 🗣️🔥🔥🔥

The dst of pkt already
translate into pod ip

But how that possible?



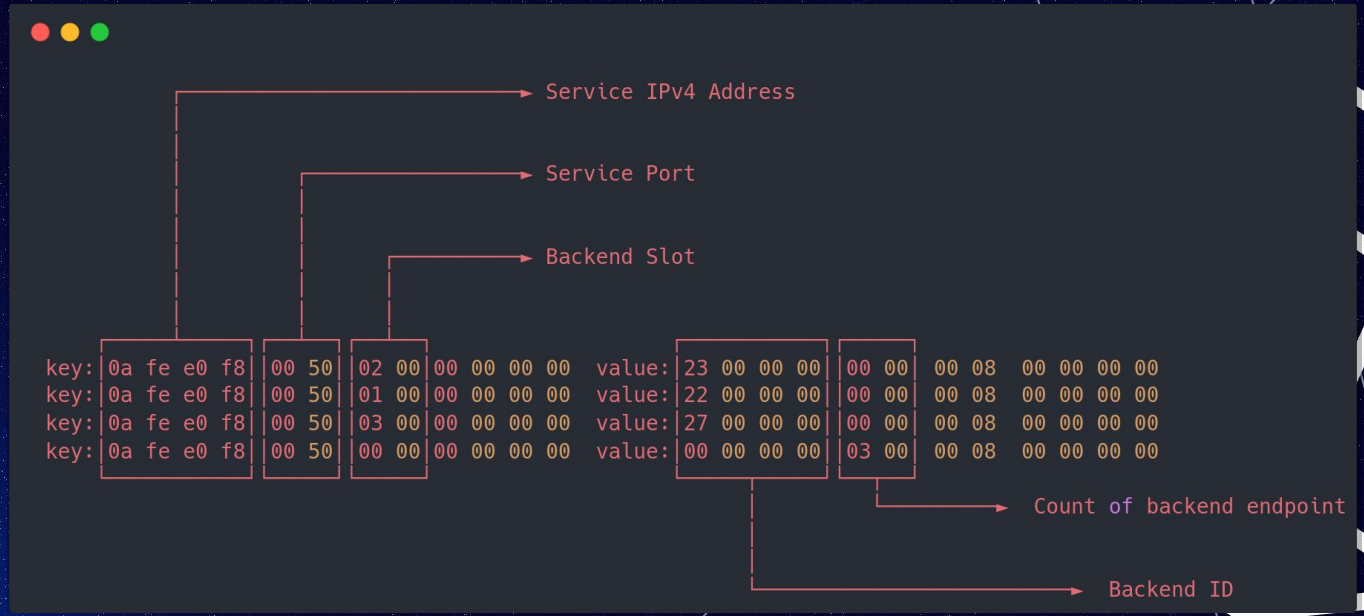
Bro is Cooking 🗣️🔥🔥🔥

All key value in eBPF was written in hex format so we need to convert svc ip into hex

```
root@kube-44ae8-default-worker-klxsw-t9swq-9rbrw:~# python3
Python 3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> for i in [10,254,224,248]:
...     print(f'{i:x}',end=" ")
...
a fe e0 f8 >>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
root@kube-44ae8-default-worker-klxsw-t9swq-9rbrw:~# bpftool map dump pinned /sys/fs/bpf/tc/globals/cilium_lb4_services_v2 | grep "0a fe e0 f8"
key: 0a fe e0 f8 00 50 02 00  00 00 00 00  value: 23 00 00 00 00 00 00 08  00 00 00 00
key: 0a fe e0 f8 00 50 01 00  00 00 00 00  value: 22 00 00 00 00 00 00 08  00 00 00 00
key: 0a fe e0 f8 00 50 03 00  00 00 00 00  value: 27 00 00 00 00 00 00 08  00 00 00 00
key: 0a fe e0 f8 00 50 00 00  00 00 00 00  value: 00 00 00 00 03 00 00 08  00 00 00 00
root@kube-44ae8-default-worker-klxsw-t9swq-9rbrw:~# _
```

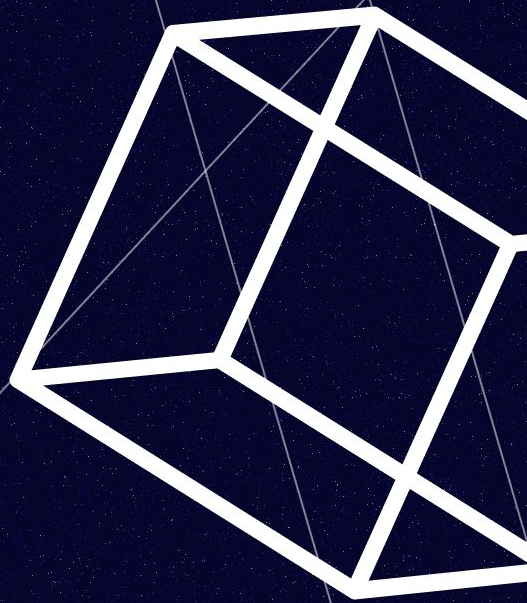
Bro is Cooking 🗣️🔥🔥🔥

According ~~judgment of Oratrice Mecanique d'Analyse Cardinale~~ to cilium source code you can see the ebpf map have struct like this

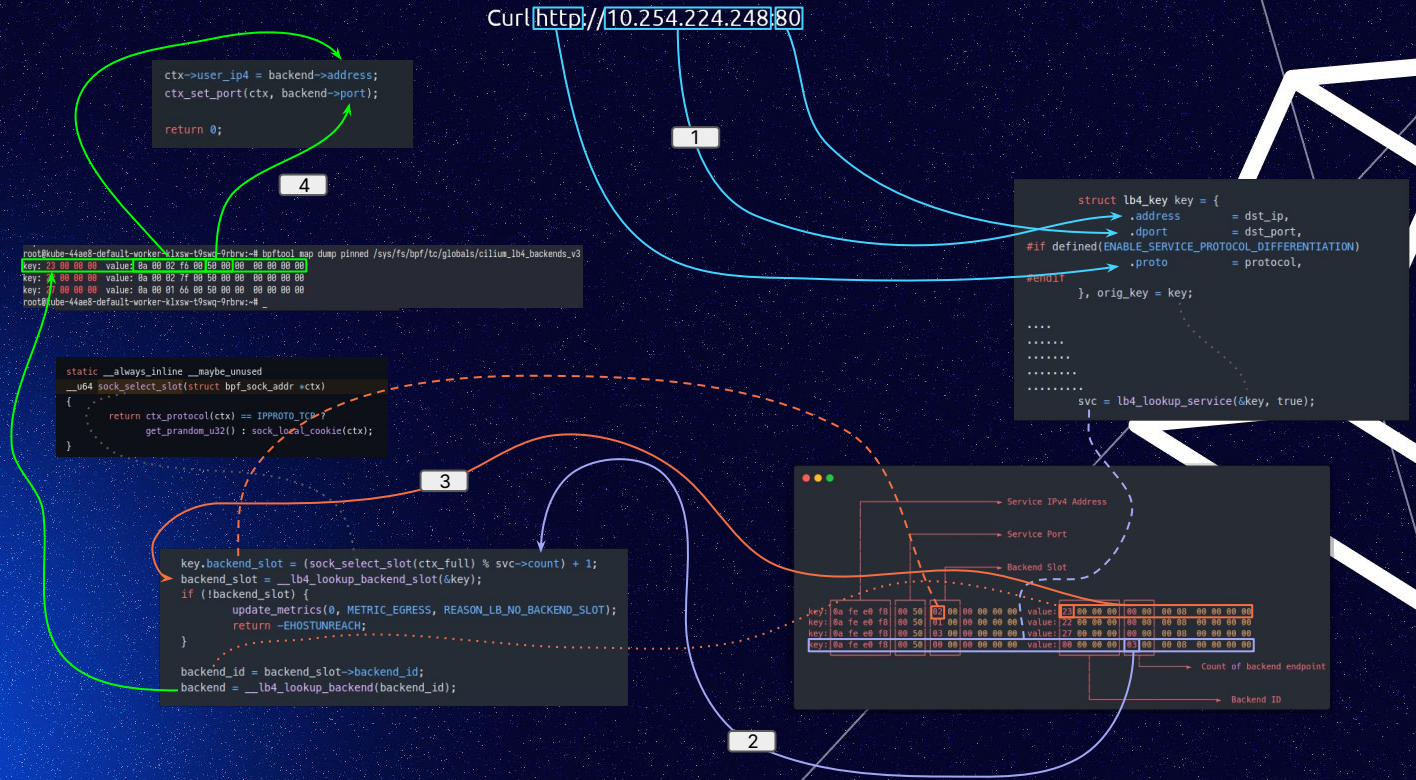


Bro is Cooking 🗣️🔥🔥🔥

Now the last part is to understand how cilium create a load balancer only from ebpf map



Bro is Cooking 🗣️🔥🔥🔥



Bro is Cooking 🗣️🔥🔥🔥

When and Where the eBPF change the pkt?

Bro is Cooking 🗣️🔥🔥🔥

If it on kernel then syscall manage it

```
__section("cgroup/connect4")
int cil_sock4_connect(struct bpf_sock_addr *ctx)
{
        int err;
```

```
netshoot-1:~# strace -e trace=network curl 10.254.224.248
socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 5
setsockopt(5, SOL_TCP, TCP_NODELAY, [1], 4) = 0
setsockopt(5, SOL_SOCKET, SO_KEEPALIVE, [1], 4) = 0
setsockopt(5, SOL_TCP, TCP_KEEPIDLE, [60], 4) = 0
setsockopt(5, SOL_TCP, TCP_KEEPINTVL, [60], 4) = 0
connect(5, {sa_family=AF_INET, sin_port=htons(80), sin_addr=inet_addr("10.254.224.248")}, 16) = -1
EINPROGRESS (Operation in progress)
getsockname(5, {sa_family=AF_INET, sin_port=htons(59256), sin_addr=inet_addr("10.0.2.205")}, [128 => 16])
= 0
getsockopt(5, SOL_SOCKET, SO_ERROR, [0], [4]) = 0
getsockname(5, {sa_family=AF_INET, sin_port=htons(59256), sin_addr=inet_addr("10.0.2.205")}, [128 => 16])
= 0
getsockname(5, {sa_family=AF_INET, sin_port=htons(59256), sin_addr=inet_addr("10.0.2.205")}, [128 => 16])
= 0
getsockname(5, {sa_family=AF_INET, sin_port=htons(59256), sin_addr=inet_addr("10.0.2.205")}, [128 => 16])
= 0
sendto(5, "GET / HTTP/1.1\r\nHost: 10.254.224"..., 77, MSG_NOSIGNAL, NULL, 0) = 77
recvfrom(5, "HTTP/1.1 200 OK\r\nServer: nginx/1"..., 102400, 0, NULL, NULL) = 853
```

Phew that’s a lot, any question?

THANK YOU

NUTANIX