KubeCon

CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon | CloudNativeCon

THE LINUX FOUNDATION OPEN SOURCE SUMMIT

AI_dev Open Source GenAI & ML Summit

China 2024

Understanding the Buzz Around Cilium: Introduction and in Production at Alibaba

Liyi Huang, Isovalent && Bokang Li, Alibaba Cloud

Agenda

- Cilium general introduction
- Shallow dive from network policy with ACK
- How CNI looks like at Alibaba Cloud
- Scalability on Alibaba Cloud
- What can you get from a full blown cilium on Alibaba
- Some highlights on cilium 1.16 release
- Q&A

cilium

eBPF-based:

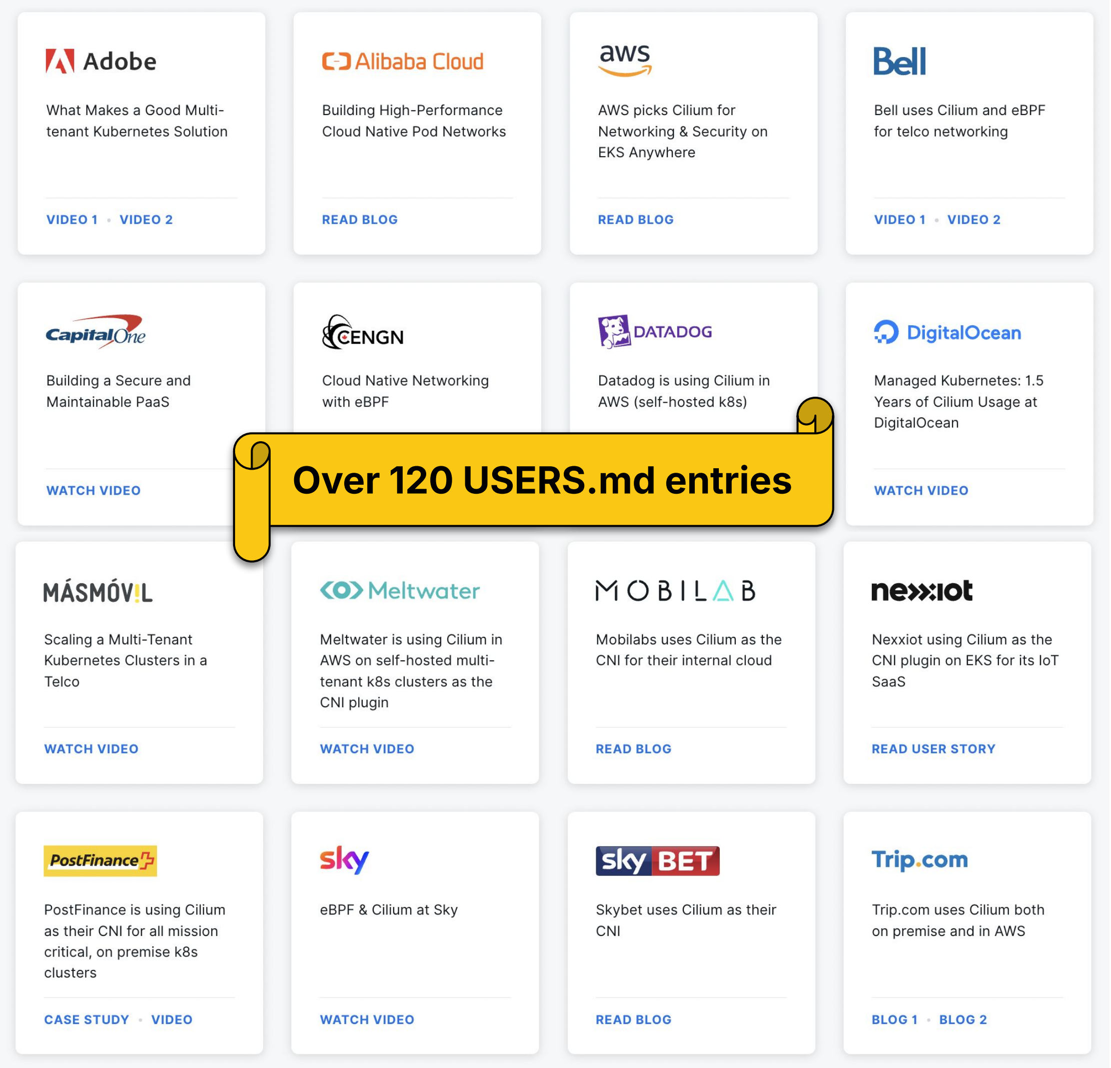
- Networking
- Security
- Observability
- Service Mesh & Ingress

Foundation

CLOUD NATIVE COMPUTING FOUNDATION

Technology

eBPF envoy



Adobe

What Makes a Good Multi-tenant Kubernetes Solution

VIDEO 1 · VIDEO 2

Alibaba Cloud

Building High-Performance Cloud Native Pod Networks

READ BLOG

aws

AWS picks Cilium for Networking & Security on EKS Anywhere

READ BLOG

Bell

Bell uses Cilium and eBPF for telco networking

VIDEO 1 · VIDEO 2

Capital One

Building a Secure and Maintainable PaaS

WATCH VIDEO

CENGN

Cloud Native Networking with eBPF

DATADOG

Datadog is using Cilium in AWS (self-hosted k8s)

DigitalOcean

Managed Kubernetes: 1.5 Years of Cilium Usage at DigitalOcean

WATCH VIDEO

MÁSMÓV!L

Scaling a Multi-Tenant Kubernetes Clusters in a Telco

WATCH VIDEO

Meltwater

Meltwater is using Cilium in AWS on self-hosted multi-tenant k8s clusters as the CNI plugin

WATCH VIDEO

MOBILAB

Mobilabs uses Cilium as the CNI for their internal cloud

READ BLOG

nexxiot

Nexxiot using Cilium as the CNI plugin on EKS for its IoT SaaS

READ USER STORY

PostFinance

PostFinance is using Cilium as their CNI for all mission critical, on premise k8s clusters

CASE STUDY · VIDEO

sky

eBPF & Cilium at Sky

WATCH VIDEO

sky BET

Skybet uses Cilium as their CNI

READ BLOG

Trip.com

Trip.com uses Cilium both on premise and in AWS

BLOG 1 · BLOG 2

Over 120 USERS.md entries

cilium

eBPF-based:

- Networking
- Security
- Observability
- Service Mesh & Ingress

Foundation

CLOUD NATIVE
COMPUTING FOUNDATION

Technology

eBPF envoy

Deploy on your prefered cloud

aws Sovereign Cloud Stack Tencent Cloud openstack CIVO nine cloud navigators

EQUINIX | METAL™ Alibaba Cloud Scaleway Google Cloud

DigitalOcean DaoCloud Azure EXOSCALE APPUiO SWISS CONTAINER PLATFORM HETZNER

Use your favorite Kubernetes distribution

Azure Kubernetes Service OPENSHIFT Amazon EKS RKE 2 Google Kubernetes Engine

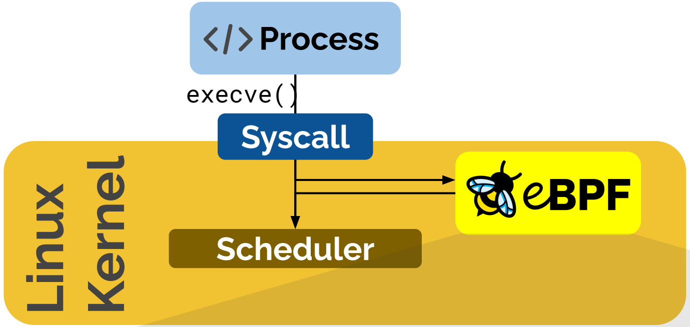
Gardener kind CRANE kops KUBEKEY k8e

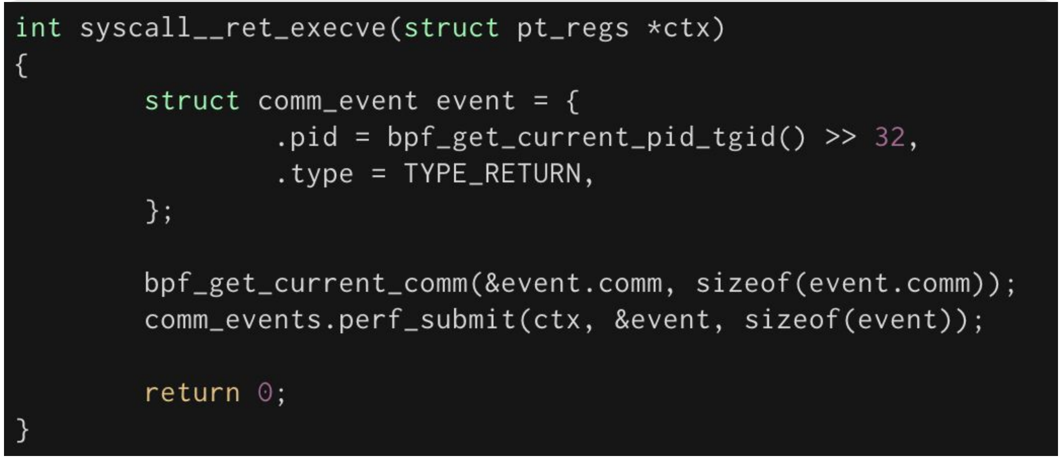
kubease KUBERMATIC KubeOne KUBESPRAY FLANT Deckhouse Kubernetes Platform

eBPF

Makes the Linux kernel programmable in a secure and efficient way.

“What JavaScript is to the browser, eBPF is to the Linux Kernel”





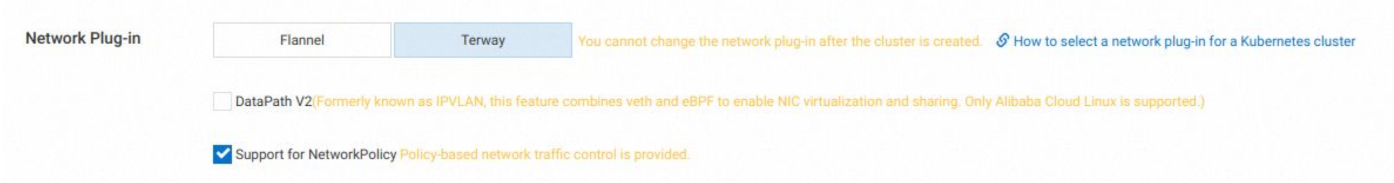
```
int syscall__ret_execve(struct pt_regs *ctx)
{
        struct comm_event event = {
                .pid = bpf_get_current_pid_tgid() >> 32,
                .type = TYPE_RETURN,
        };

        bpf_get_current_comm(&event.comm, sizeof(event.comm));
        comm_events.perf_submit(ctx, &event, sizeof(event));

        return 0;
}
```

Kubernetes Network policy

- Pods within the same Kubernetes cluster can communicate with each other without restriction.
- If you want to limit the traffic between pods, you will need to use a network policy.
- You can enable it by ticking the checkbox when creating the ACK cluster, as shown in the following picture.



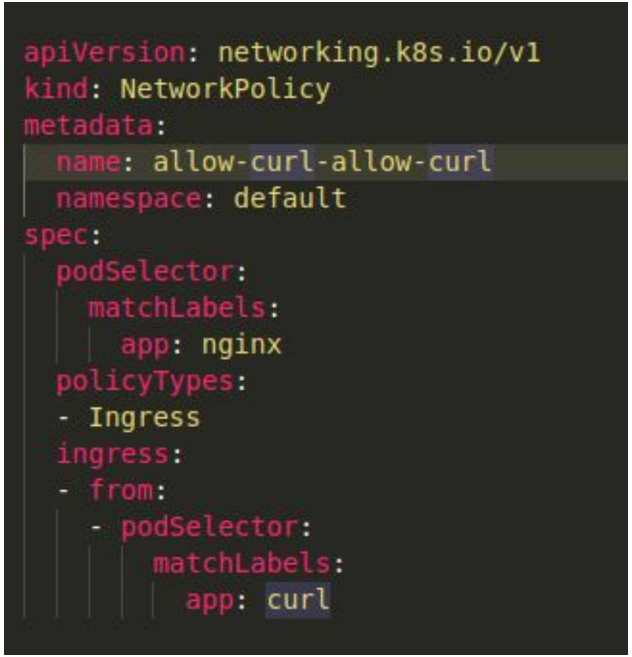
Network Plug-in [Flannel] [Terway] You cannot change the network plug-in after the cluster is created. How to select a network plug-in for a Kubernetes cluster

□ DataPath V2(Formerly known as IPVLAN, this feature combines veth and eBPF to enable NIC virtualization and sharing. Only Alibaba Cloud Linux is supported.)

☑ Support for NetworkPolicy Policy-based network traffic control is provided.

Kubernetes Network policy

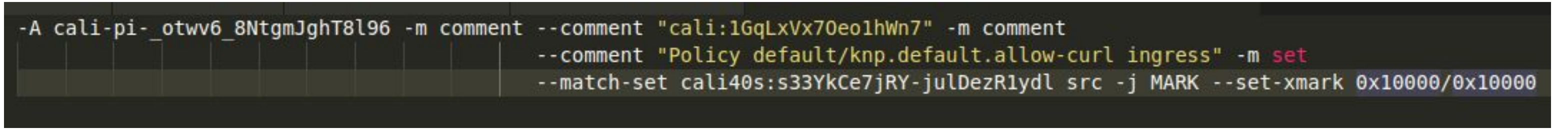
- Can pod A talk to pod B?
- A example of the basic network policy



```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-curl-allow-curl
  namespace: default
spec:
  podSelector:
    matchLabels:
      app: nginx
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: curl
```

Kubernetes Network policy

- What does this really mean to the Linux host?

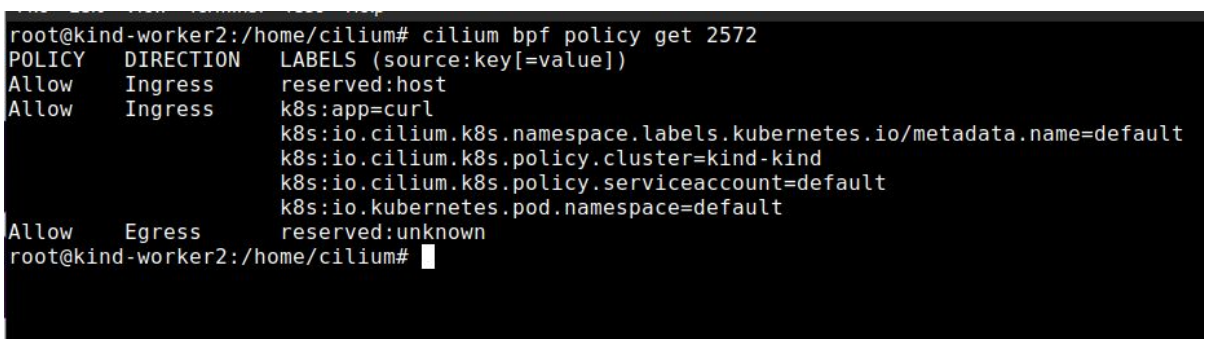


```
-A cali-pi-_otwv6_8NtgmJghT8l96 -m comment --comment "cali:1GqLxVx70eo1hWn7" -m comment
                                           --comment "Policy default/knp.default.allow-curl ingress" -m set
                                           --match-set cali40s:s33YkCe7jRY-julDezR1ydl src -j MARK --set-xmark 0x10000/0x10000
```

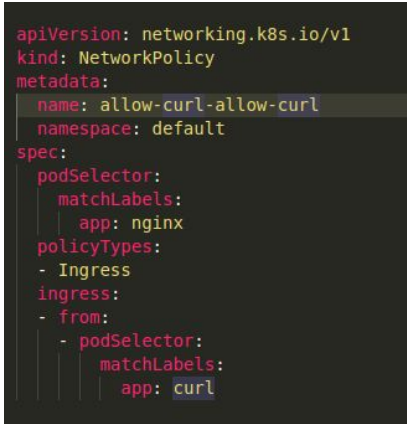
- You start to chase the iptables tables/rules and ipset on the host. I found there are around 300 rules(including rules for kube-proxy) with just 2 pods and 3 nodes(1 controller and 2 workers) and no user defined service on it on my KIND cluster.
- How do I know if there is a drop with iptables rules? You need to other non standard network policy implementation to log the flow to a file
- Iptables lookup performance is O(n).

Kubernetes Network policy

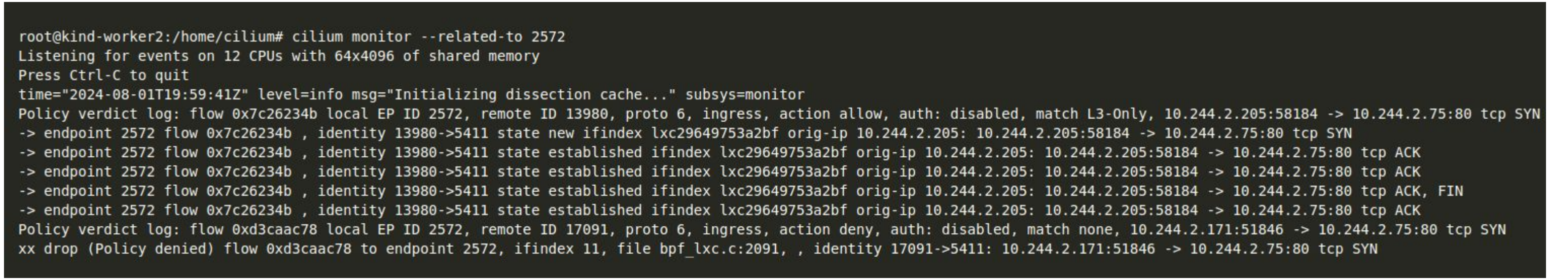
- Same policy for cilium



```
root@kind-worker2:/home/cilium# cilium bpf policy get 2572
POLICY   DIRECTION   LABELS (source:key[=value])
Allow    Ingress     reserved:host
Allow    Ingress     k8s:app=curl
                     k8s:io.cilium.k8s.namespace.labels.kubernetes.io/metadata.name=default
                     k8s:io.cilium.k8s.policy.cluster=kind-kind
                     k8s:io.cilium.k8s.policy.serviceaccount=default
                     k8s:io.kubernetes.pod.namespace=default
Allow    Egress      reserved:unknown
root@kind-worker2:/home/cilium#
```



```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-curl-allow-curl
  namespace: default
spec:
  podSelector:
    matchLabels:
      app: nginx
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: curl
```



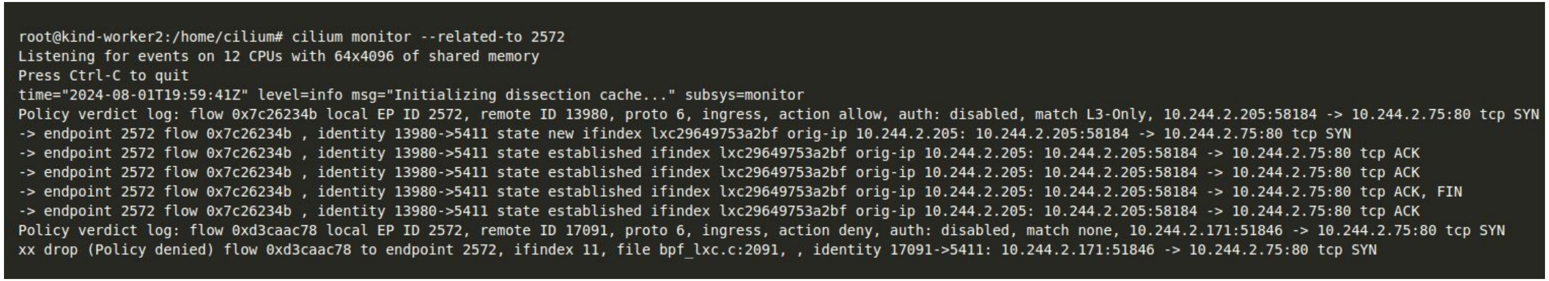
```
root@kind-worker2:/home/cilium# cilium monitor --related-to 2572
Listening for events on 12 CPUs with 64x4096 of shared memory
Press Ctrl-C to quit
time="2024-08-01T19:59:41Z" level=info msg="Initializing dissection cache..." subsys=monitor
Policy verdict log: flow 0x7c26234b local EP ID 2572, remote ID 13980, proto 6, ingress, action allow, auth: disabled, match L3-Only, 10.244.2.205:58184 -> 10.244.2.75:80 tcp SYN
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state new ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp SYN
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK, FIN
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK
Policy verdict log: flow 0xd3caac78 local EP ID 2572, remote ID 17091, proto 6, ingress, action deny, auth: disabled, match none, 10.244.2.171:51846 -> 10.244.2.75:80 tcp SYN
xx drop (Policy denied) flow 0xd3caac78 to endpoint 2572, ifindex 11, file bpf_lxc.c:2091, , identity 17091->5411: 10.244.2.171:51846 -> 10.244.2.75:80 tcp SYN
```

Kubernetes Network policy

- Network policy lookup from cilium is O(1).
- You can easily observe it with cilium tool
- More advanced Cilium network policy will be discussed later

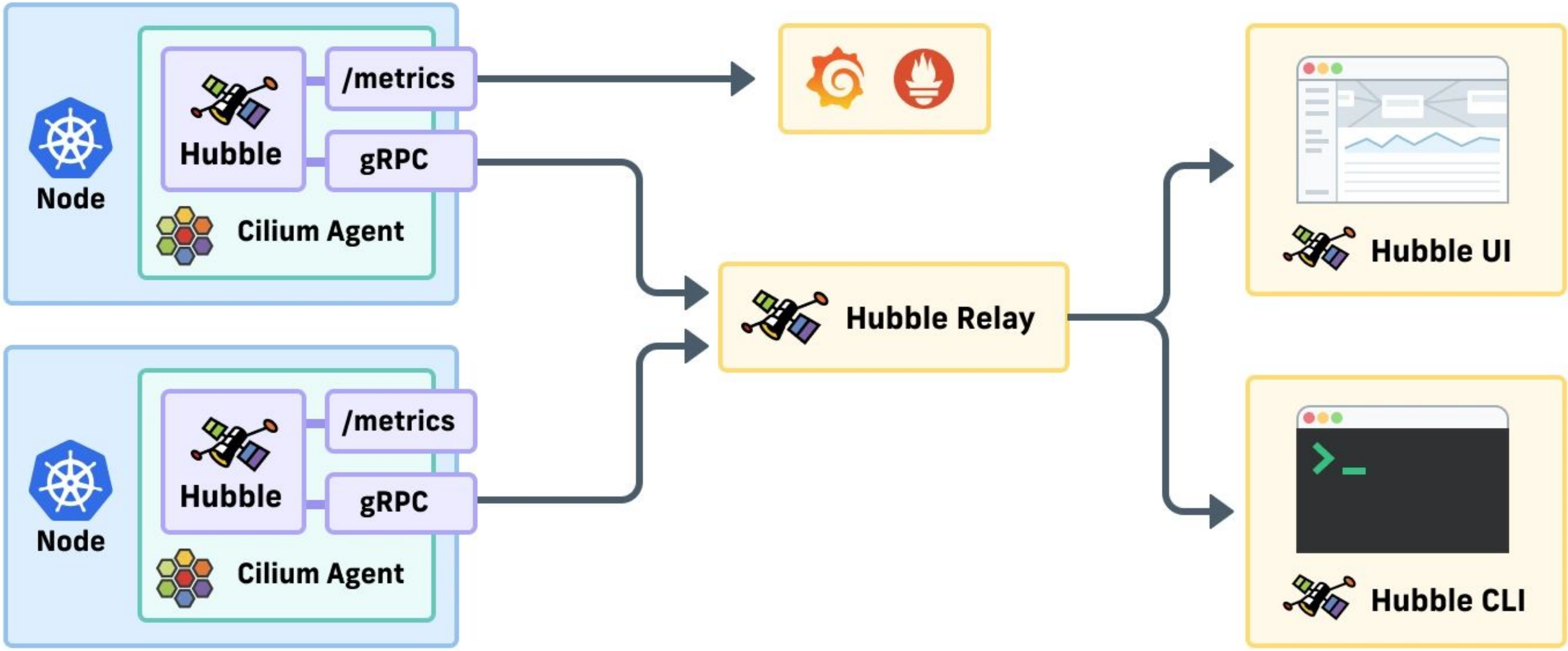
Hubble Intro

- We still need to get to the cilium container to run cilium monitor command.
- What if we have a tool to see all the flow logs on a cluster even more cluster?
- Can we export all the logs to SIEM?
- Can we generate the network policy based on flow data?



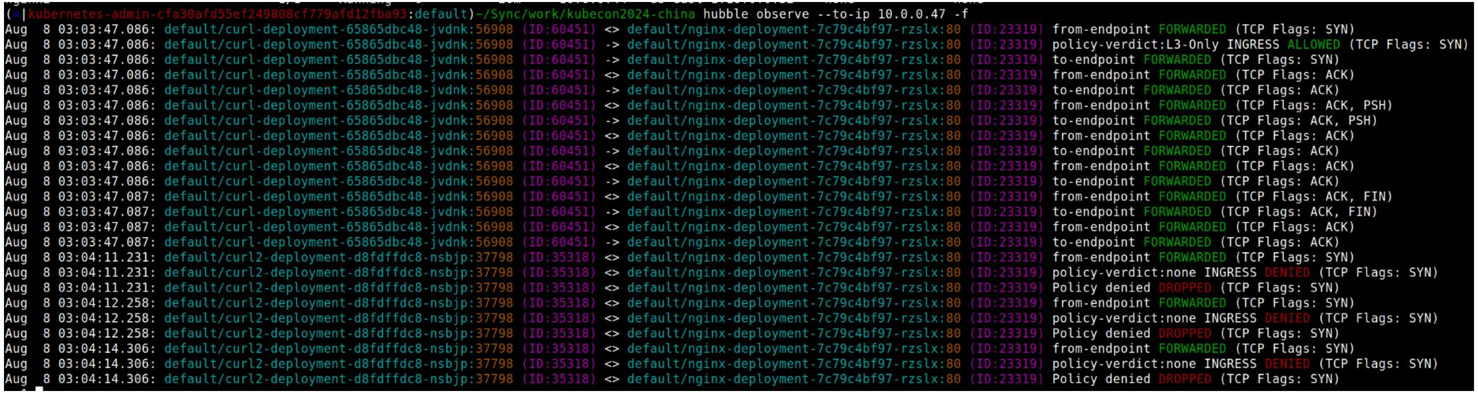
```
root@kind-worker2:/home/cilium# cilium monitor --related-to 2572
Listening for events on 12 CPUs with 64x4096 of shared memory
Press Ctrl-C to quit
time="2024-08-01T19:59:41Z" level=info msg="Initializing dissection cache..." subsys=monitor
Policy verdict log: flow 0x7c26234b local EP ID 2572, remote ID 13980, proto 6, ingress, action allow, auth: disabled, match L3-Only, 10.244.2.205:58184 -> 10.244.2.75:80 tcp SYN
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state new ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp SYN
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK, FIN
-> endpoint 2572 flow 0x7c26234b , identity 13980->5411 state established ifindex lxc29649753a2bf orig-ip 10.244.2.205: 10.244.2.205:58184 -> 10.244.2.75:80 tcp ACK
Policy verdict log: flow 0xd3caac78 local EP ID 2572, remote ID 17091, proto 6, ingress, action deny, auth: disabled, match none, 10.244.2.171:51846 -> 10.244.2.75:80 tcp SYN
xx drop (Policy denied) flow 0xd3caac78 to endpoint 2572, ifindex 11, file bpf_lxc.c:2091, , identity 17091->5411: 10.244.2.171:51846 -> 10.244.2.75:80 tcp SYN
```

Hubble Overview



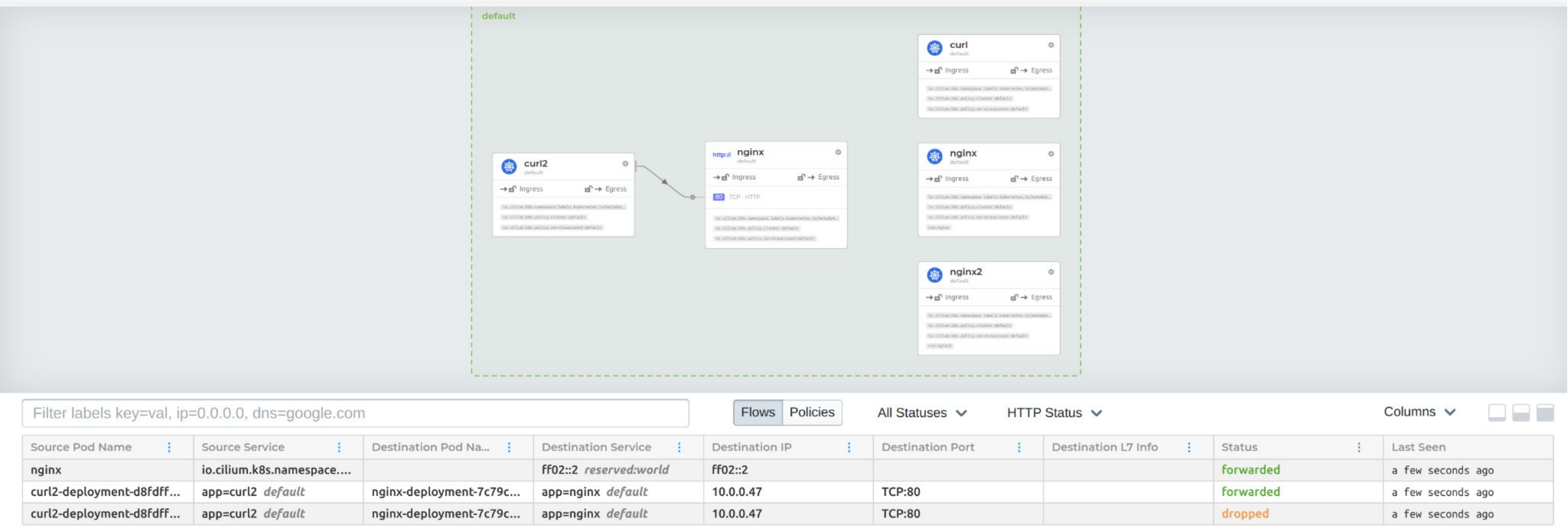
Hubble CLI

- Observe the traffic for the whole cluster
- Filter with ip/pod/svc/namespace/fqdn/http/type etc..



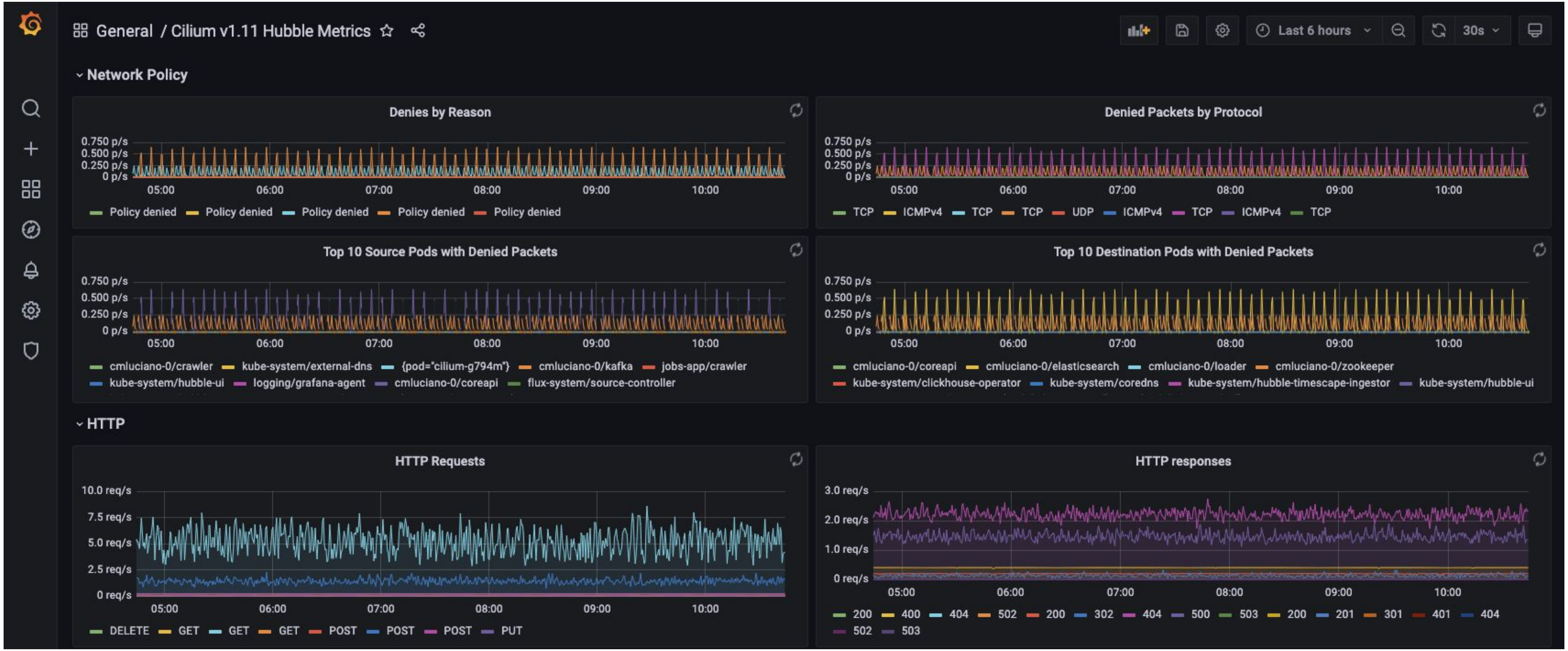
```
(⎈|kubernetes-admin-cfa30afd55ef249808cf779afd12fba93:default)~/Sync/work/kubecon2024-china hubble observe --to-ip 10.0.0.47 -f
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: SYN)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) policy-verdict:L3-Only INGRESS ALLOWED (TCP Flags: SYN)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: SYN)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: ACK, PSH)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: ACK, PSH)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.086: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.087: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: ACK, FIN)
Aug  8 03:03:47.087: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: ACK, FIN)
Aug  8 03:03:47.087: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:03:47.087: default/curl-deployment-65865dbc48-jvdnk:56908 (ID:60451) -> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) to-endpoint FORWARDED (TCP Flags: ACK)
Aug  8 03:04:11.231: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: SYN)
Aug  8 03:04:11.231: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) policy-verdict:none INGRESS DENIED (TCP Flags: SYN)
Aug  8 03:04:11.231: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) Policy denied DROPPED (TCP Flags: SYN)
Aug  8 03:04:12.258: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: SYN)
Aug  8 03:04:12.258: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) policy-verdict:none INGRESS DENIED (TCP Flags: SYN)
Aug  8 03:04:12.258: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) Policy denied DROPPED (TCP Flags: SYN)
Aug  8 03:04:14.306: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) from-endpoint FORWARDED (TCP Flags: SYN)
Aug  8 03:04:14.306: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) policy-verdict:none INGRESS DENIED (TCP Flags: SYN)
Aug  8 03:04:14.306: default/curl2-deployment-d8fdffdc8-nsbjp:37798 (ID:35318) <> default/nginx-deployment-7c79c4bf97-rzslx:80 (ID:23319) Policy denied DROPPED (TCP Flags: SYN)
```

Hubble GUI


default
curl2
default
Ingress
Egress
io.cilium.k8s.namespace.labels.kubernetes.io/metadat…
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
http://
nginx
default
Ingress
Egress
80
TCP · HTTP
io.cilium.k8s.namespace.labels.kubernetes.io/metadat…
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
curl
default
Ingress
Egress
io.cilium.k8s.namespace.labels.kubernetes.io/metadat…
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
nginx
default
Ingress
Egress
io.cilium.k8s.namespace.labels.kubernetes.io/metadat…
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
run:nginx
nginx2
default
Ingress
Egress
io.cilium.k8s.namespace.labels.kubernetes.io/metadat…
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
run:nginx2
Filter labels key=val, ip=0.0.0.0, dns=google.com
Flows
Policies
All Statuses
HTTP Status
Columns

Source Pod Name	Source Service	Destination Pod Na...	Destination Service	Destination IP	Destination Port	Destination L7 Info	Status	Last Seen
nginx	io.cilium.k8s.namespace....		ff02::2 *reserved:world*	ff02::2			forwarded	a few seconds ago
curl2-deployment-d8fdff...	app=curl2 *default*	nginx-deployment-7c79c...	app=nginx *default*	10.0.0.47	TCP:80		forwarded	a few seconds ago
curl2-deployment-d8fdff...	app=curl2 *default*	nginx-deployment-7c79c...	app=nginx *default*	10.0.0.47	TCP:80		dropped	a few seconds ago

Hubble metrics



General / Cilium v1.11 Hubble Metrics

Last 6 hours

30s

Network Policy

Denies by Reason

Policy denied, Policy denied, Policy denied, Policy denied, Policy denied

Denied Packets by Protocol

TCP, ICMPv4, TCP, TCP, UDP, ICMPv4, TCP, ICMPv4, TCP

Top 10 Source Pods with Denied Packets

cmluciano-0/crawler, kube-system/external-dns, {pod="cilium-g794m"}, cmluciano-0/kafka, jobs-app/crawler, kube-system/hubble-ui, logging/grafana-agent, cmluciano-0/coreapi, flux-system/source-controller

Top 10 Destination Pods with Denied Packets

cmluciano-0/coreapi, cmluciano-0/elasticsearch, cmluciano-0/loader, cmluciano-0/zookeeper, kube-system/clickhouse-operator, kube-system/coredns, kube-system/hubble-timescape-ingestor, kube-system/hubble-ui

HTTP

HTTP Requests

DELETE, GET, GET, GET, POST, POST, POST, PUT

HTTP responses

200, 400, 404, 502, 200, 302, 404, 500, 503, 200, 201, 301, 401, 404, 502, 503

About Alibaba Cloud

No.1

Market Share in the Asia Pacific

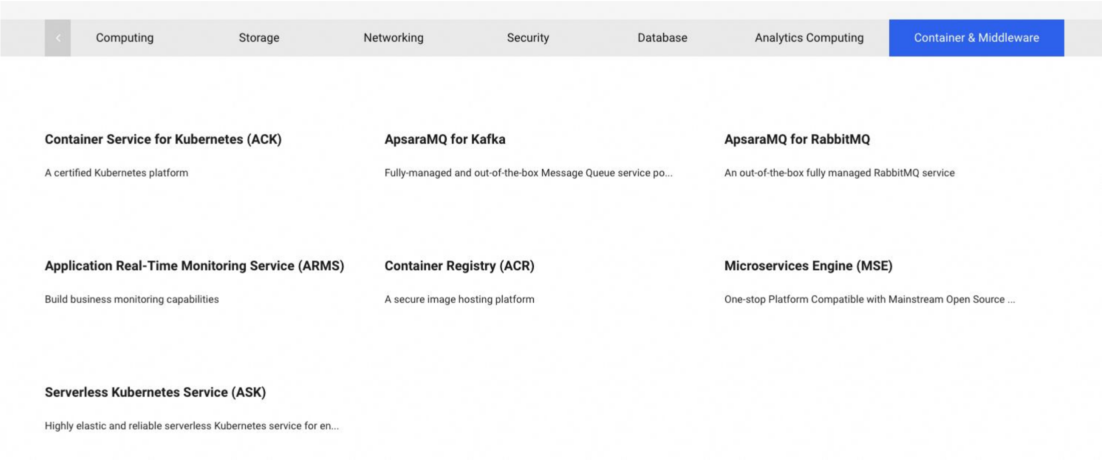
89

Availability Zones

30

Regions

Alibaba Cloud



Container Service for Kubernetes (ACK)

A certified Kubernetes platform

ApsaraMQ for Kafka

Fully-managed and out-of-the-box Message Queue service po...

ApsaraMQ for RabbitMQ

An out-of-the-box fully managed RabbitMQ service

Application Real-Time Monitoring Service (ARMS)

Build business monitoring capabilities

Container Registry (ACR)

A secure image hosting platform

Microservices Engine (MSE)

One-stop Platform Compatible with Mainstream Open Source ...

Serverless Kubernetes Service (ASK)

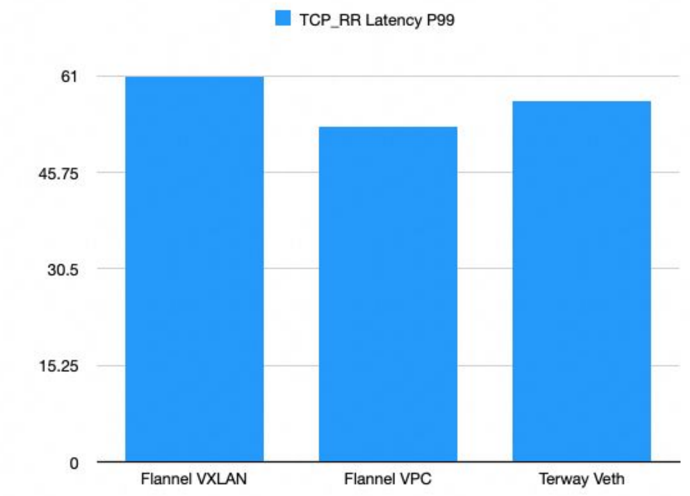
Highly elastic and reliable serverless Kubernetes service for en...

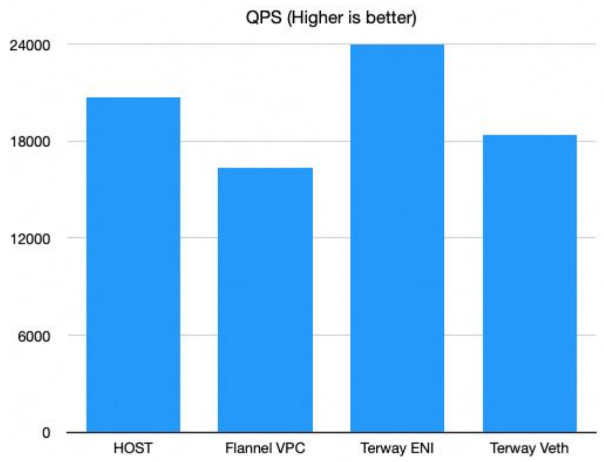
Container Service @alibabacloud

How CNI looks like at Alibaba Cloud

	ACK Flannel	Terway
Network Type	VPC Route	ECS ENI
Accelerated Networking	None	ipvlan & eBPF,datapath V2
Scale	200 nodes (up to 1000 nodes)	5000 nodes (up to 15000 nodes)
Security	None	Pod Security Group, NetworkPolicy, ACK GlobalNetworkPolicy
IPAM	Fixed size for every node	Elastic, can enlarge any time. Support fixed IP.
NFV	None	RDMA,eRDMA,SMC-R,SRIOV,DPDK...
Pod N/S Communication	None	EIP,DNAT Gateway, IPv6 Gateway (with ack-extend-netwokr-controller)
Loadbalancer Backend	NodePort	Pod IP

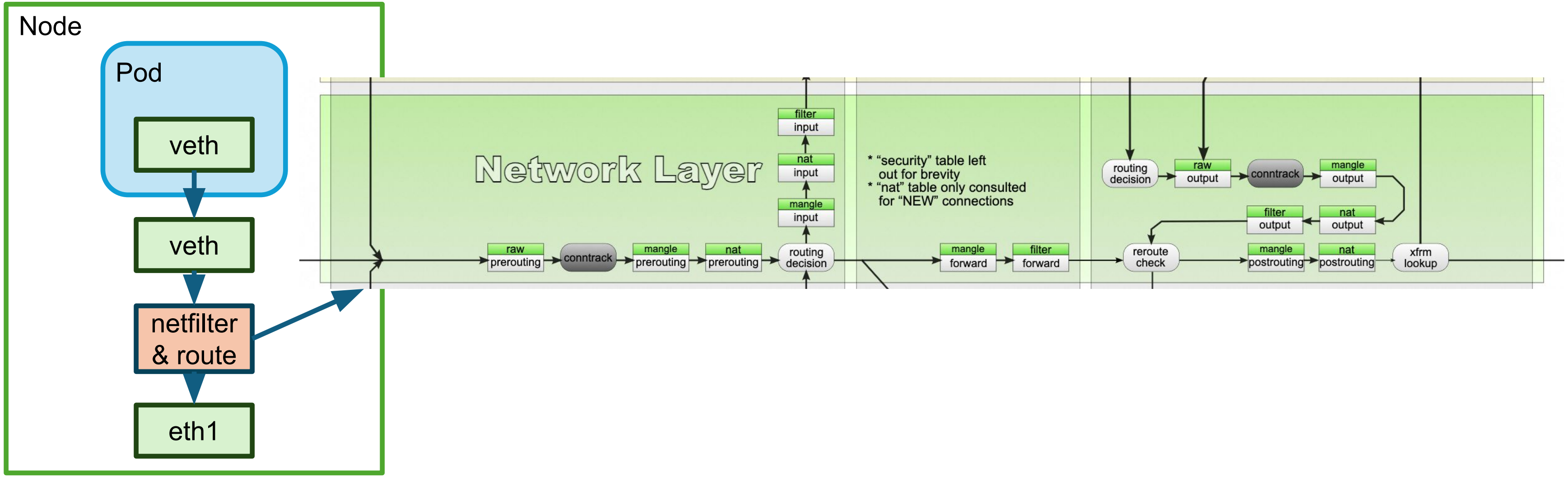
Overhead on stander datapath





The reason for the overhead is the packetization and the length of the kernel path

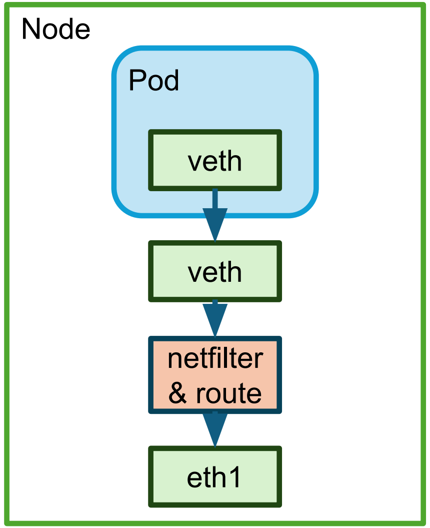
Overhead on stander datapath



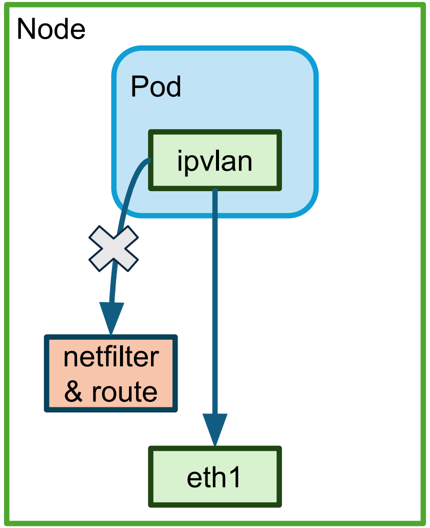
* "security" table left out for brevity
* "nat" table only consulted for "NEW" connections

Standard datapath

Datapath IPvlan



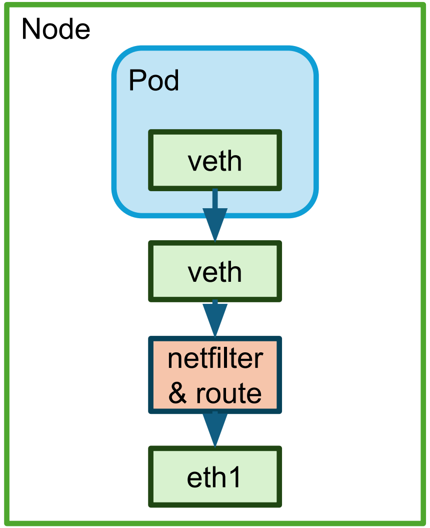
Standard datapath



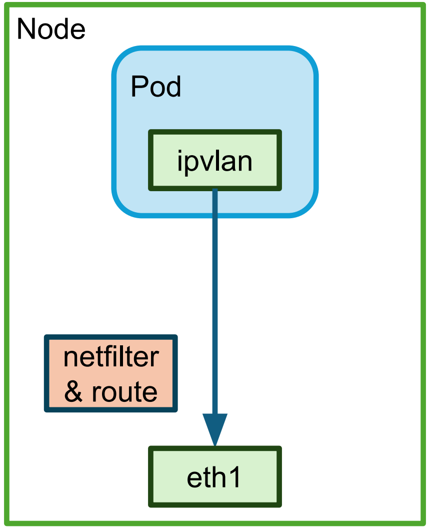
IPvlan datapath

IPvlan allows bypassing the host's network stack, but it poses challenges with Service functionality.

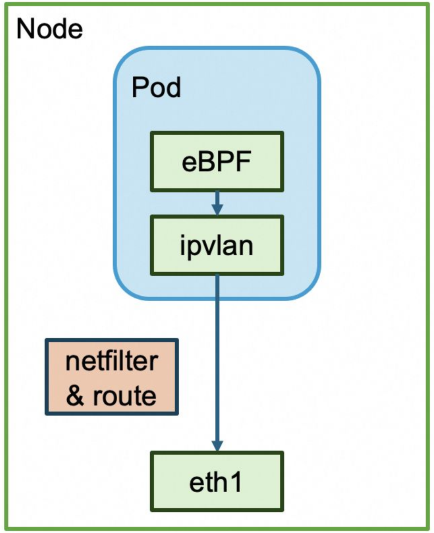
Datapath IPvlan + eBPF



Standard datapath

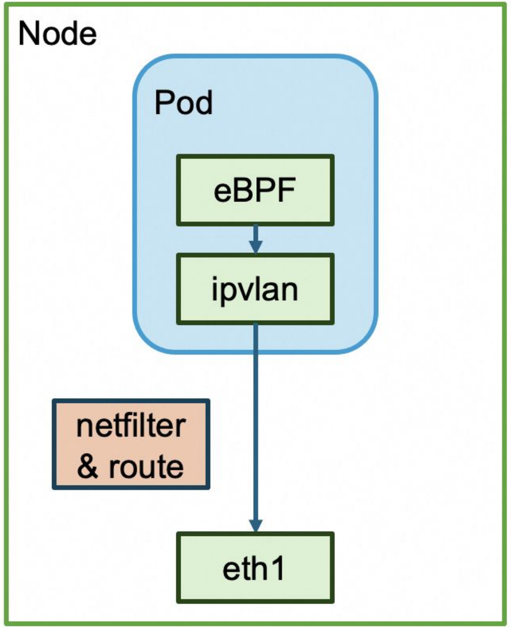


IPvlan datapath



IPvlan + eBPF datapath

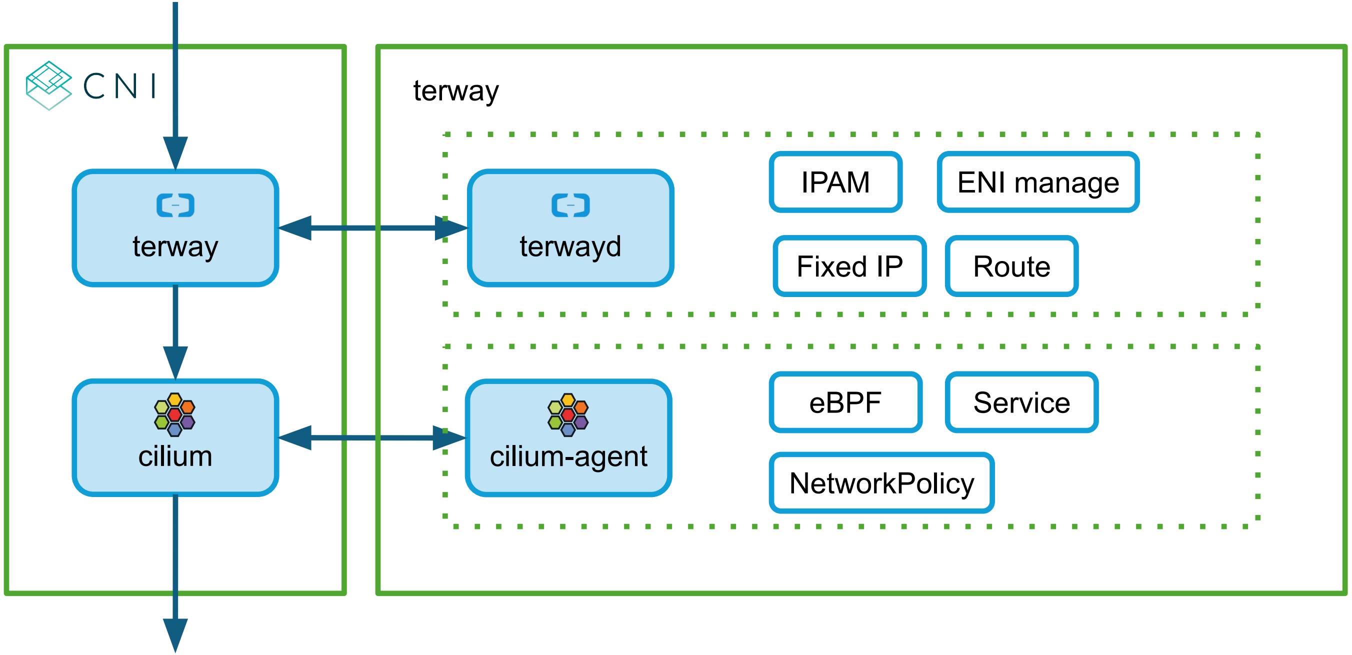
Features we used



IPvlan + eBPF datapath

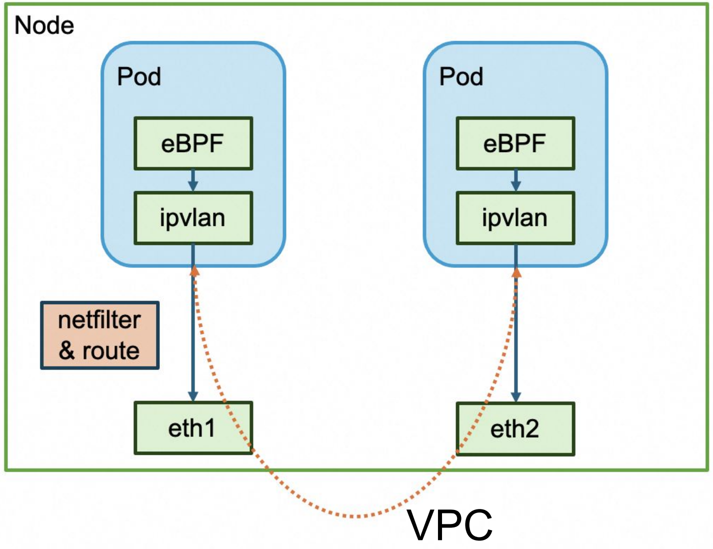
- KPR
  - partial, only for containers
- NetworkPolicy
  - K8s NetworkPolicy
  - CiliumClusterWideNetworkPolicy
    - Use ACK GlobalNetworkPolicy - Container Service for Kubernetes - Alibaba Cloud Documentation Center
- BandwidthManager
  - Egress side, EDT at kernel 5.10
- Hubble
  - Implement network observability by using ACK Terway and Cilium Hubble - Container - Alibaba Cloud

CNI Chaining



Limitation on IPvlan+eBPF

- Traffic will not go through node
  - Need additional redirect rule for traffic like nodelocal dns
  - Monitor may need addition adapt for the datapath
- Connectivity issue
  - NodePort may not be reachable
- Performance
  - Traffic between pods on the same node may route through the VPC

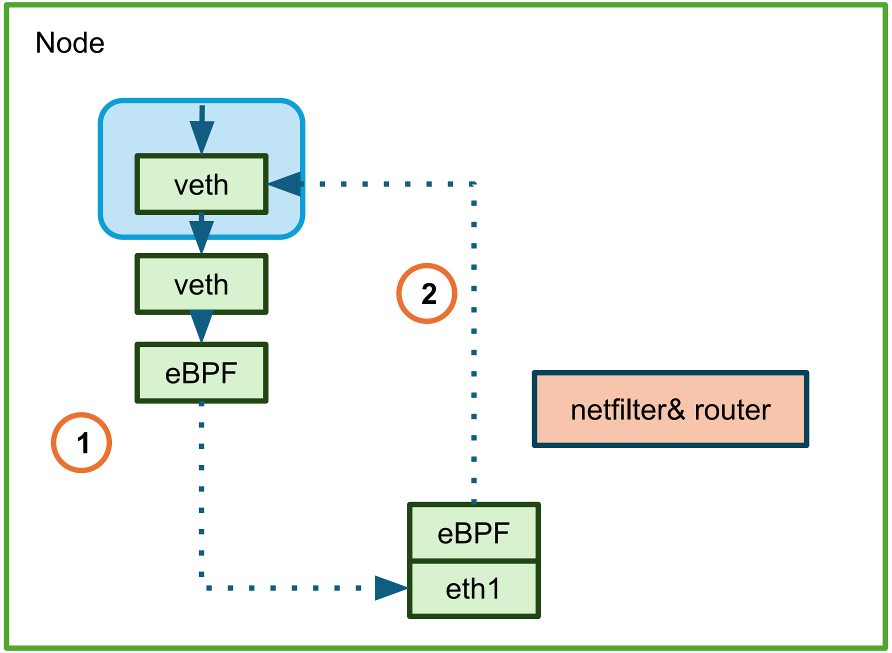


IPvlan + eBPF datapath

Datapath V2

Just like stander datapath, but enhanced with eBPF

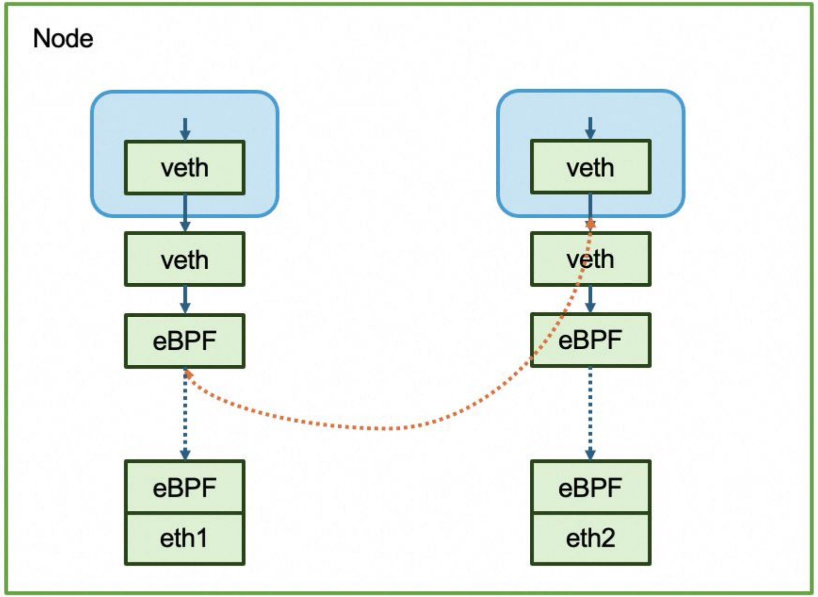
- Pod to world
  - bpf_redirect_neigh
- Reverse package
  - bpf_redirect_peer



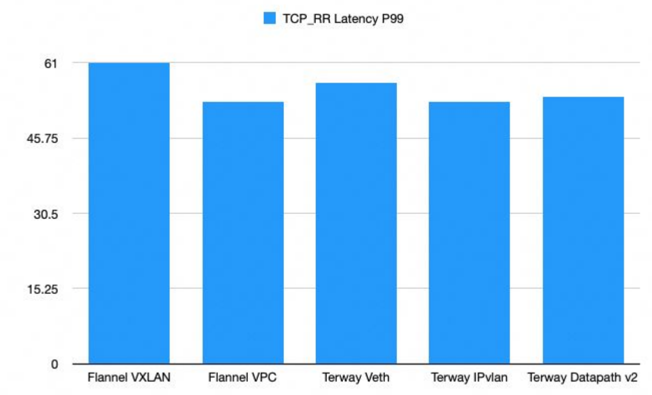
Datapath V2

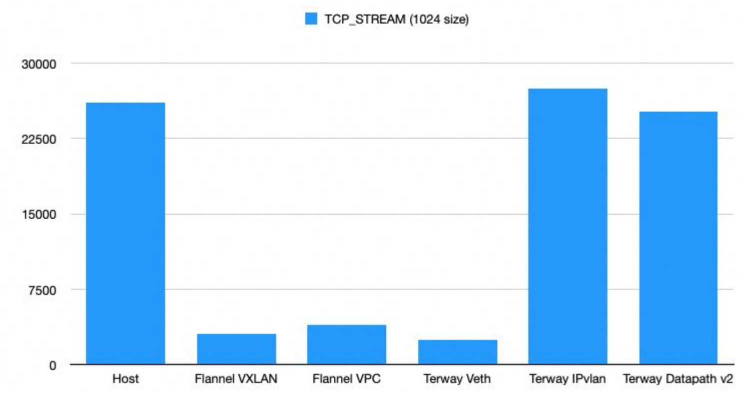
Enhanced compatibility

- Pod traffic can be tracked on host
- Pod to Pod on same node, will no longer go through VPC



Result





- The IPvlan mode provides the best pod-to-pod performance
- Datapath V2 performance significantly outperforms veth mode, coming very close to IPvlan mode.
- In certain scenarios, Datapath V2 performs better than IPvlan mode.

Cilium case study with Alibaba cloud

- https://www.cncf.io/case-studies/alibaba/

What’s next

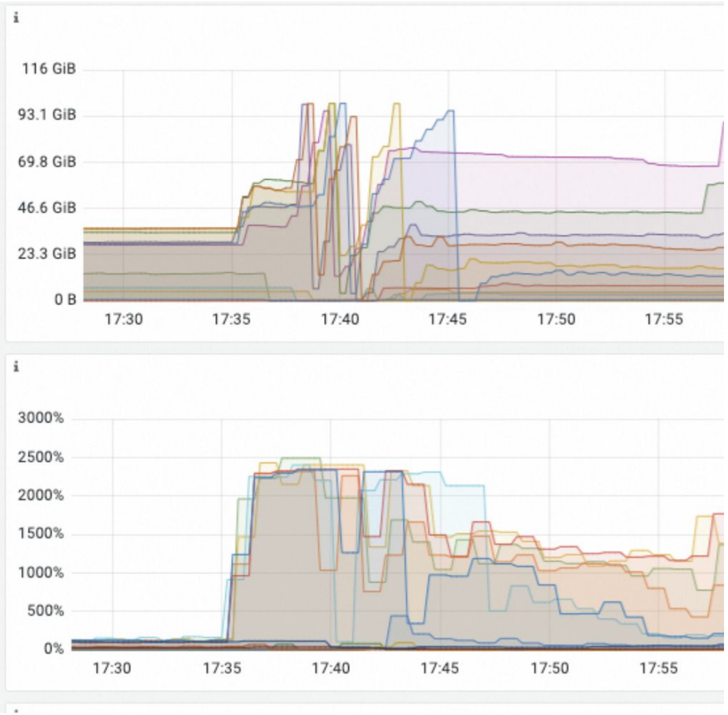
- netkit provide a faster network namespace switch for off-node traffic
- Full kube-proxy replacement can simplify the deployment configuration
- Network function offloading may be the final form

Real world disasters

Chang single namespace label.
Looks harmless...

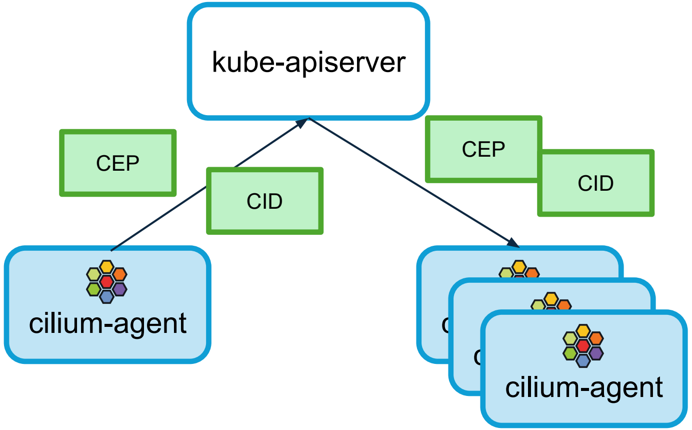
Result in to massive pressure in kube-apiserver.

2K+ Nodes
80K+ Pods



Cilium architecture

- CiliumEndpoint(CEP)
  - Track a pod
  - Every container network pod
  - Contains pod label, CiliumIdentity
- CiliumIdentity(CID)
  - Generated by pod label or CIDR (defined in NetworkPolicy)
  - Used in NetworkPolicy



Optimize in Alibaba Cloud

Changed

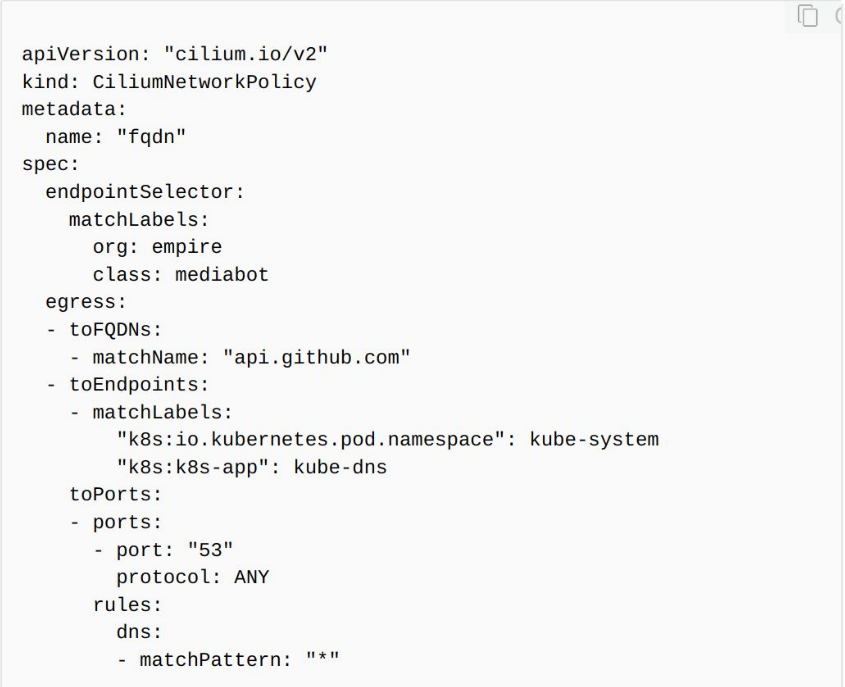
- Watch on demand
  - Added a node label for CEP. Default only watch the CEPs related to this node
- Limit the lables used in CID
- Simplify the fields in the CEP definition
- Do not sync pod labels to CEP labels
- Implement rate limiting on Kubernetes API for Cilium resources

Result

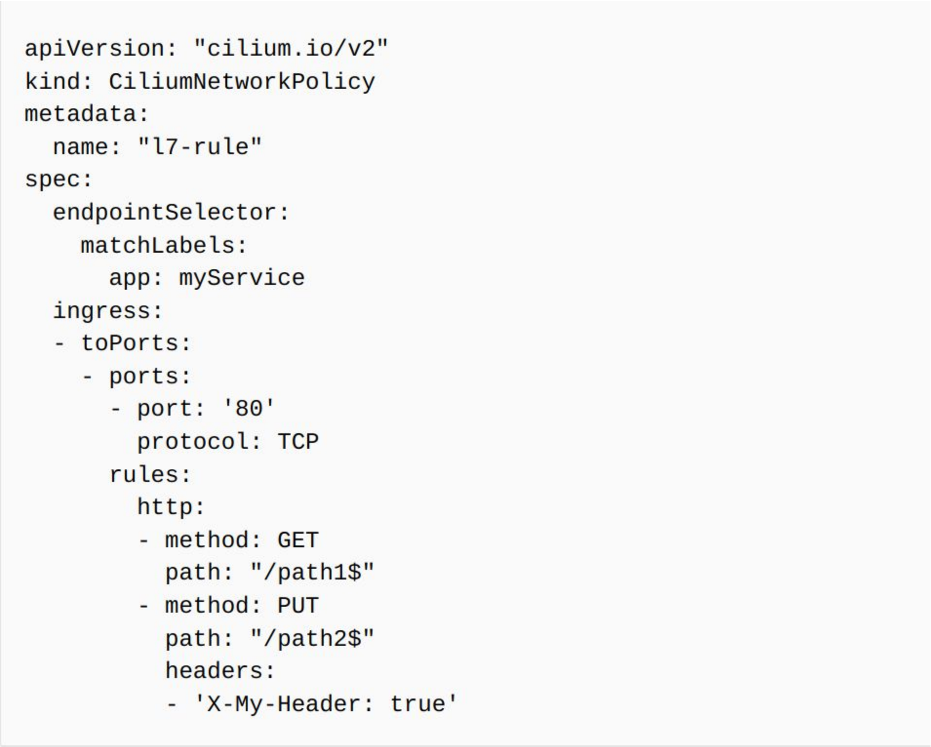
- Memory consumption has decreased by 82.5%
- The convergence time affected by the change has decreased by 95%

A full blown cilium on Alibaba

- Cilium network policy with FQDN or HTTP info



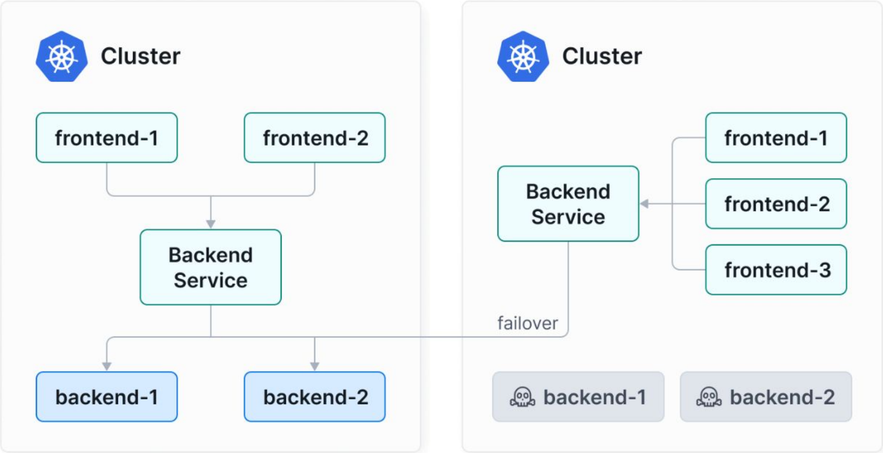
```yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "fqdn"
spec:
  endpointSelector:
    matchLabels:
      org: empire
      class: mediabot
  egress:
  - toFQDNs:
    - matchName: "api.github.com"
  - toEndpoints:
    - matchLabels:
        "k8s:io.kubernetes.pod.namespace": kube-system
        "k8s:k8s-app": kube-dns
    toPorts:
    - ports:
      - port: "53"
        protocol: ANY
      rules:
        dns:
        - matchPattern: "*"
```

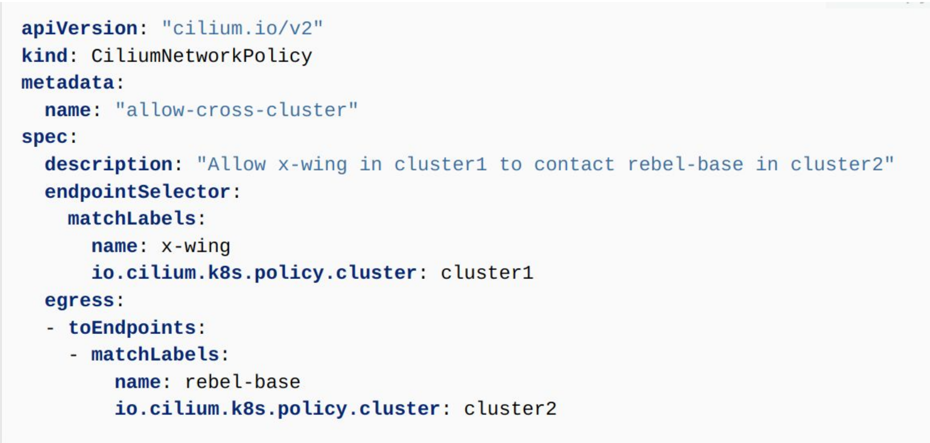


```yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "l7-rule"
spec:
  endpointSelector:
    matchLabels:
      app: myService
  ingress:
  - toPorts:
    - ports:
      - port: '80'
        protocol: TCP
      rules:
        http:
        - method: GET
          path: "/path1$"
        - method: PUT
          path: "/path2$"
          headers:
          - 'X-My-Header: true'
```

A full blown cilium on Alibaba

- Cluster mesh





```yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "allow-cross-cluster"
spec:
  description: "Allow x-wing in cluster1 to contact rebel-base in cluster2"
  endpointSelector:
    matchLabels:
      name: x-wing
      io.cilium.k8s.policy.cluster: cluster1
  egress:
  - toEndpoints:
    - matchLabels:
        name: rebel-base
        io.cilium.k8s.policy.cluster: cluster2
```

Cilium 1.16 release

Networking

- Cilium netkit: container-network throughput and latency as fast as host-network

Service mesh & Ingress/Gateway API

- Gateway API GAMMA support: East-west traffic management for the cluster via Gateway API
- Gateway API 1.1 support: Cilium now supports Gateway API 1.1

Security

- All kinds of enhancements for the Cilium Network Policy

More details on https://isovalent.com/blog/post/cilium-1-16/