KubeCon

CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon

CloudNativeCon

THE LINUX FOUNDATION

China 2024

深入了解Cilium背后的热潮：在阿里巴巴的实践

Liyi Huang, Isovalent && Bokang Li, Alibaba Cloud

议程

- Cilium 基本介绍
- 浅谈ACK上的network policy
- 阿里云上的CNI
- 阿里云上的规模化实践
- 在阿里云上完整的cilium有什么额外功能
- 1.16 新版本发布的亮眼特性
- 提问时间

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

Over 120 USERS.md entries

MÁSMÓVIL

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

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

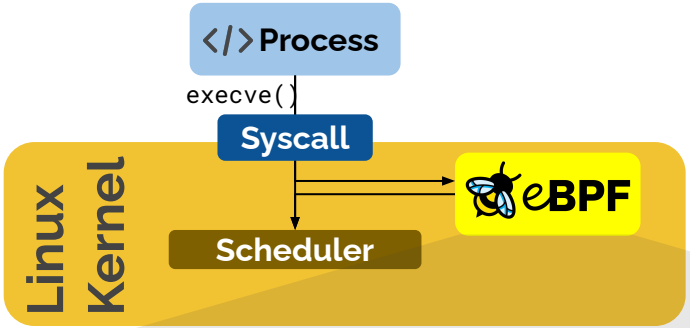
Use your favorite Kubernetes distribution

Azure Kubernetes Service, OPENSHIFT, Amazon EKS, RKE 2, Google Kubernetes Engine, Gardener, kind, CRANE, kops, KUBEKEY, k8e, kubeasz, KUBERMATIC KubeOne, KUBESPRAY, PLANT Deckhouse Kubernetes Platform

eBPF

使 Linux 内核可以安全高效地编程。

"JavaScript之于浏览器，eBPF之于Linux内核"



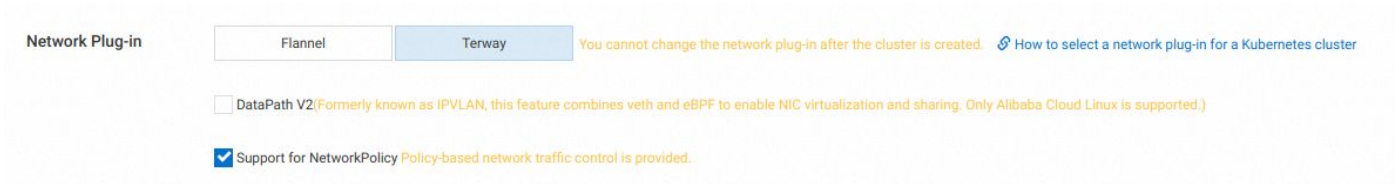

```c
int syscall__ret_execve(struct pt_regs *ctx)
{
        struct comm_event event = {
                .pid = bpf_get_current_pid_tgid() >> 32,
                .type = TYPE_RETURN,
        };

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

        return 0;
}
```

Kubernetes Network policy

- 同一 Kubernetes 集群中的 Pod 可以不受限制地相互通信。
- 如果要限制 Pod 之间的流量，则需要使用network policy。
- 您可以在创建 ACK 群集时勾选框来启用network policy，如下图所示。



Kubernetes Network policy

- Pod A和B是否可以通信？
- 一个基本的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

- 这对 Linux 主机到底意味着什么？

```
-A cali-pi-_otwv6_8NtgmJghT8l96 -m comment --comment "cali:1GqLxVx7Oeo1hWn7" -m comment
                                           --comment "Policy default/knp.default.allow-curl ingress" -m set
                                           --match-set cali40s:s33YkCe7jRY-julDezR1ydl src -j MARK --set-xmark 0x10000/0x10000
```

- 需要查看主机上的 iptables 表/规则和 ipset。我发现在我的 KIND 集群上，只有 2 个 pod 和 3 个节点（1 个controller和 2 个workers），而且没有用户定义的服务，就有大约 300 条规则（包括 kube-proxy 的规则）。
- 如何通过 iptable 规则知道是否有数据丢失？需要使用其他非标准network policy 实现将流量记录到文件中
- 查询的时间复杂度是 O(n)。

Kubernetes Network policy

- 同样的策略对于cilium来说意味着什么

```
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

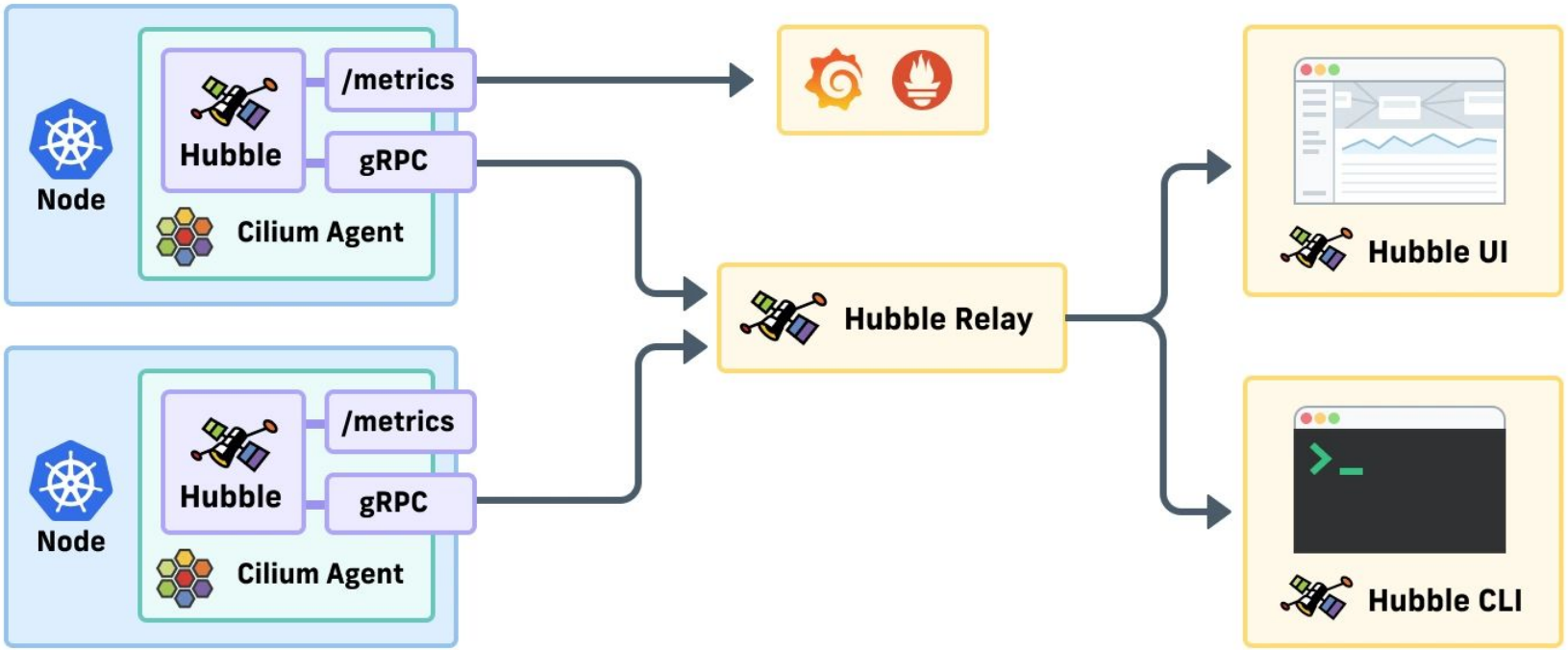
- cilium 的network policy 时间复杂度为 O(1)。
- 使用 cilium 工具可以轻松观测。
- 我们稍后会介绍更高级的cilium network policy。

Hubble 介绍

- 我们仍然需要进入 cilium 容器运行 cilium monitor 命令。
- 如果我们有一个工具可以查看一个集群甚至更多集群上的所有流量日志呢？
- 我们能将所有日志导出到 SIEM 吗？
- 我们能否根据流量数据生成network policy？

```
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 总览

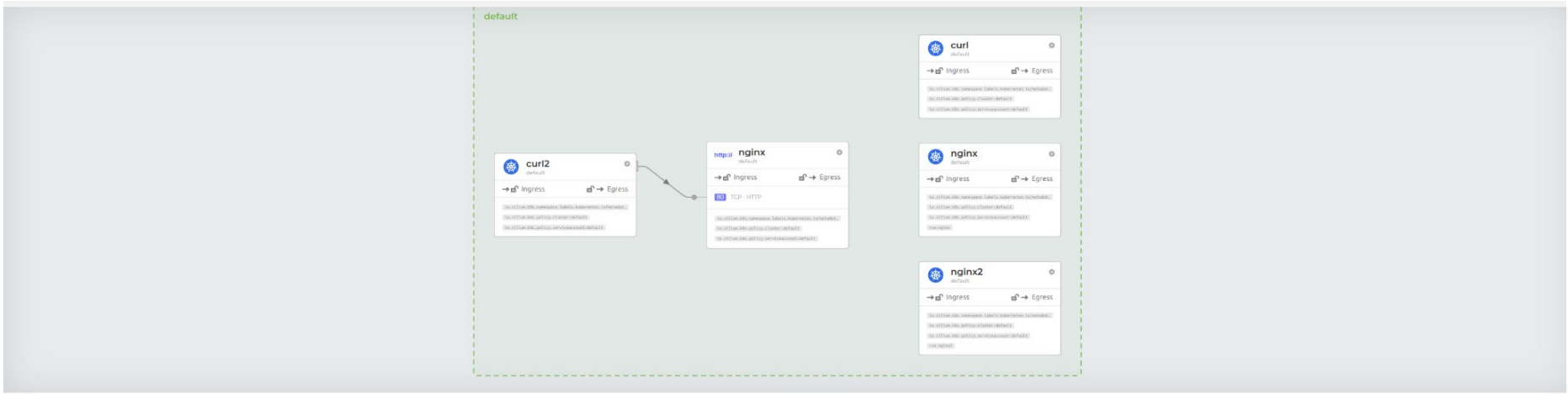

Node
Hubble
/metrics
gRPC
Cilium Agent
Node
Hubble
/metrics
gRPC
Cilium Agent
Hubble Relay
Hubble UI
Hubble CLI

Hubble CLI

- 观测整个集群的流量
- 过滤方式 ip/pod/svc/namespace/fqdn/http/type 等

```
(⎈|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 图形化界面



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

关于阿里云

No.1

Market Share in the Asia Pacific

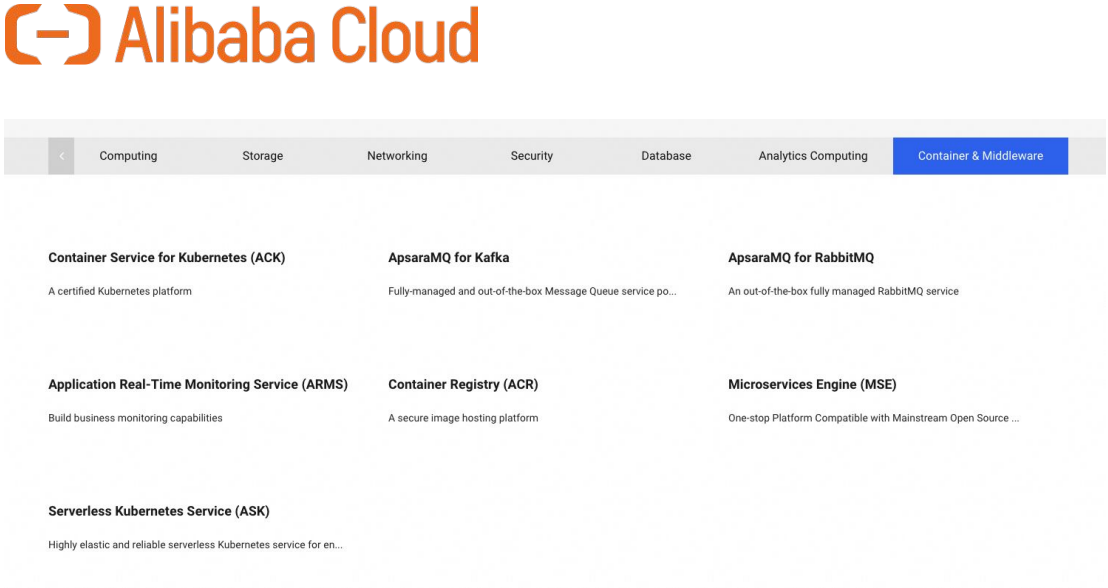
89

可用区

30

地域

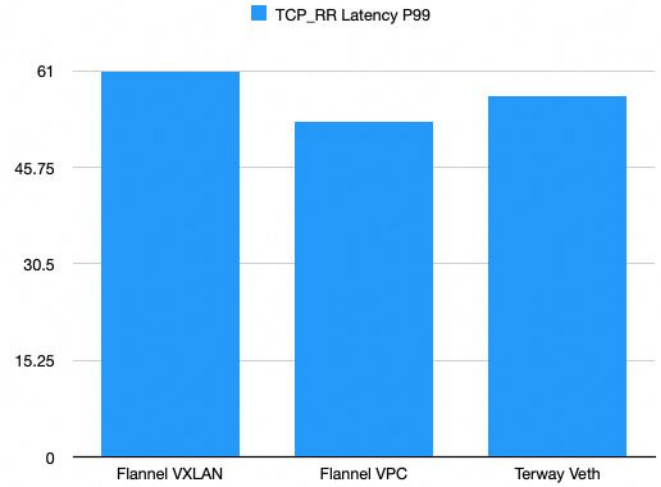
容器服务 @alibabacloud

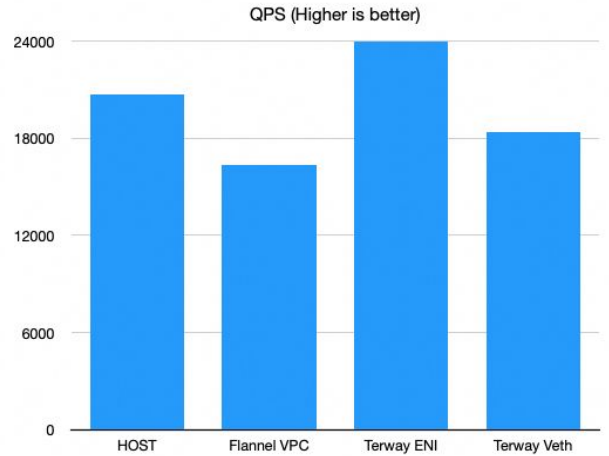


阿里云上的CNI

	ACK Flannel	Terway
网络类型	VPC 路由	ECS ENI（弹性网卡）
网络加速	None	ipvlan & eBPF,datapath V2
规模	200 节点（最多 1000 节点）	5000 节点（最多 15000 节点）
安全	None	Pod安全组, NetworkPolicy, ACK GlobalNetworkPolicy
IPAM	每个节点固定大小	弹性，可随时扩容。支持固定IP
NFV	None	RDMA,eRDMA,SMC-R,SRIOV,DPDK...
Pod 南北向暴露	None	EIP,DNAT Gateway, IPv6 Gateway (配合 ack-extend-netwokr-controller)
Loadbalancer 后端	NodePort	Pod IP

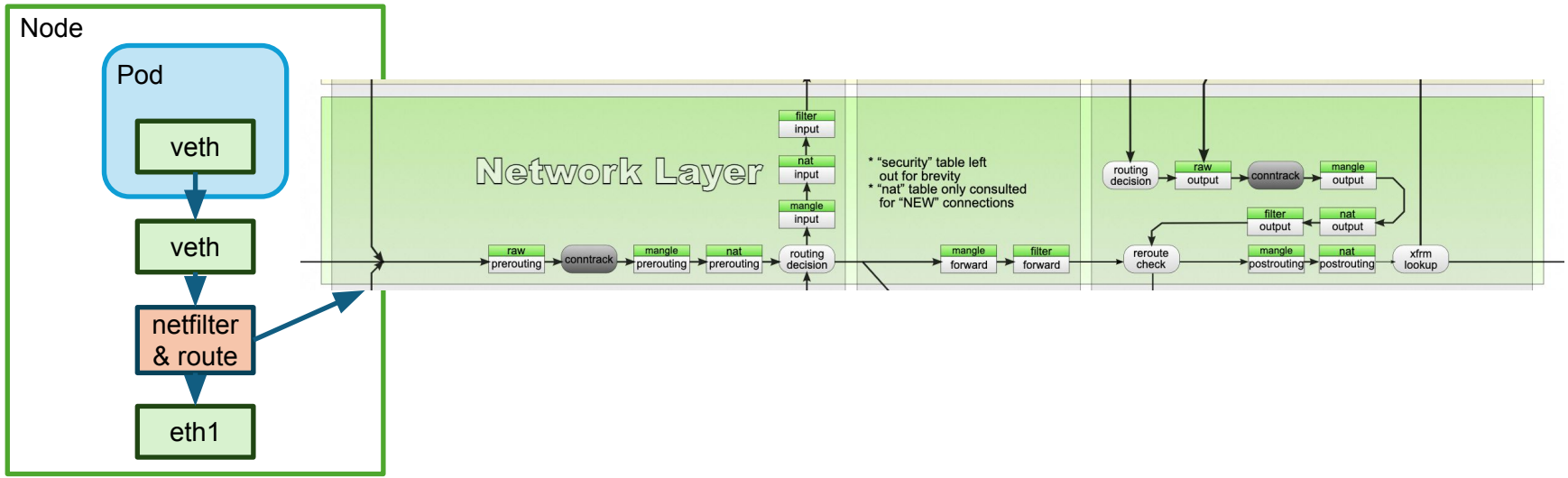
普通网络模式下的开销





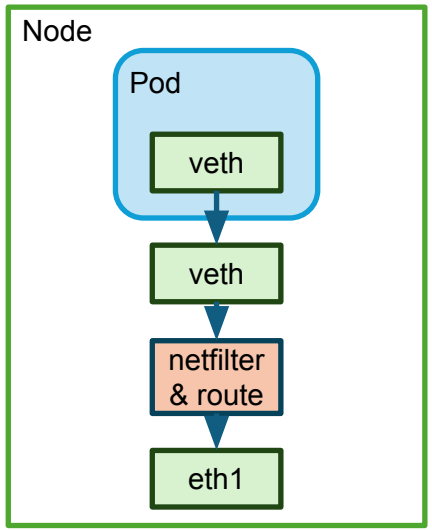
造成开销的原因是数据封包和内核路径的 长度

普通网络模式下的开销

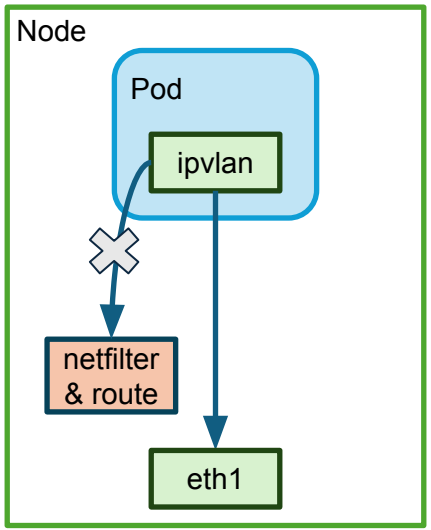


Stander datapath

IPvlan数据面



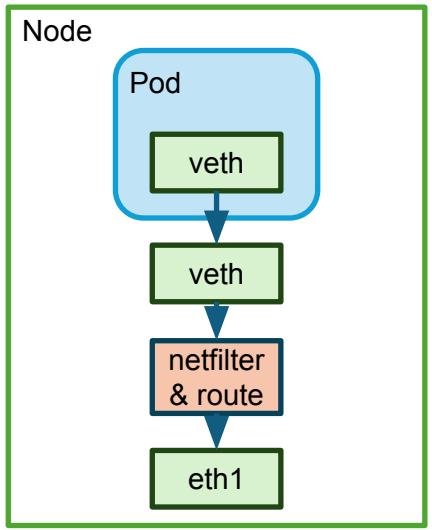
Stander
datapath



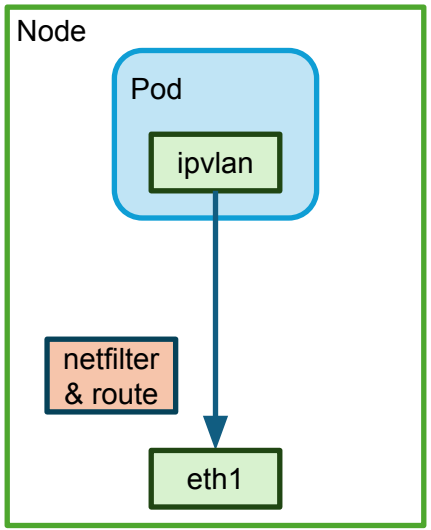
IPvlan
datapath

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

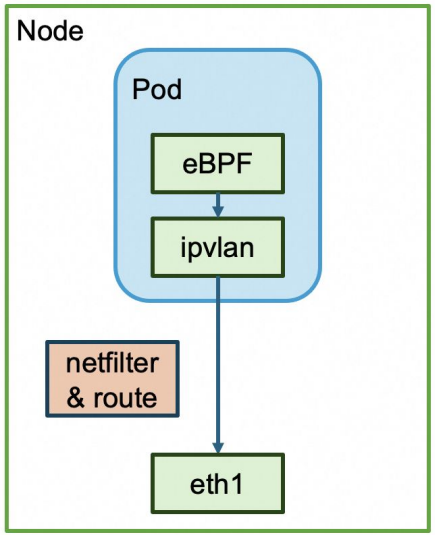
IPvlan + eBPF 数据面



Stander datapath

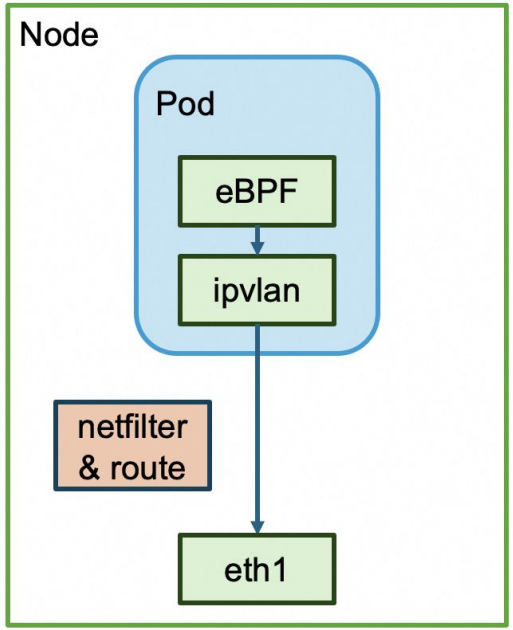


IPvlan datapath



IPvlan + eBPF datapath

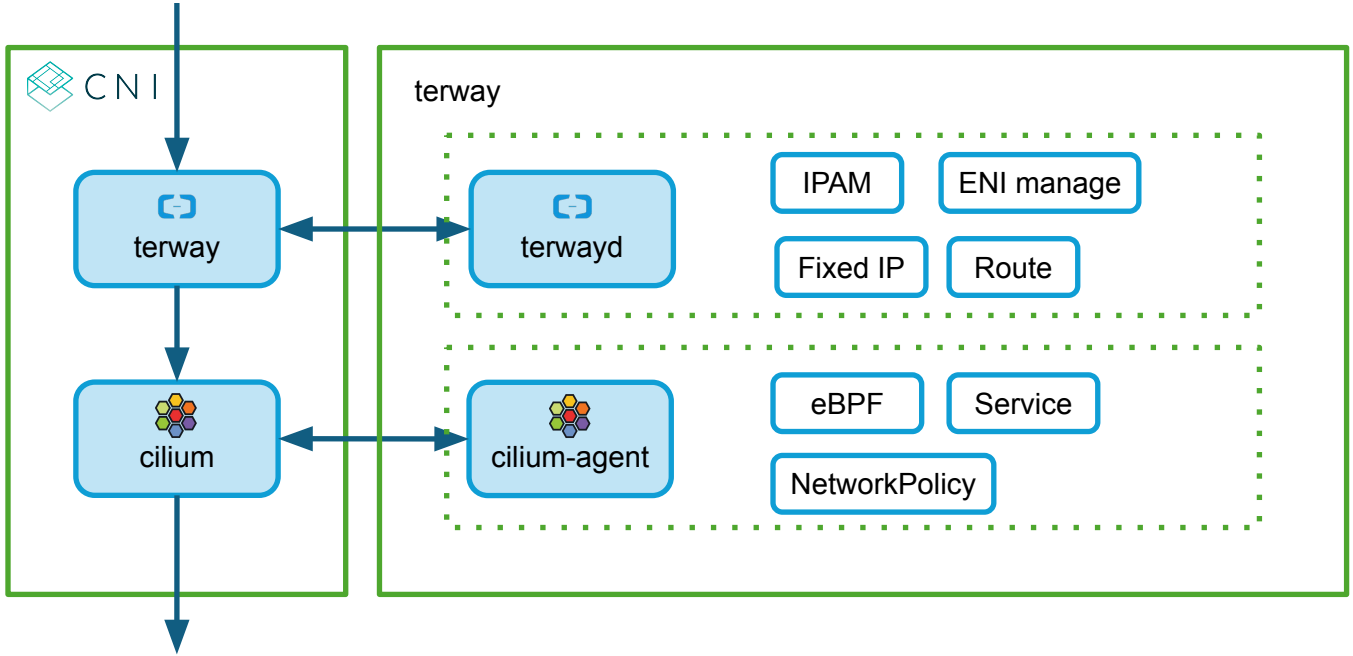
使用的cilium网络功能



IPvlan + eBPF datapath

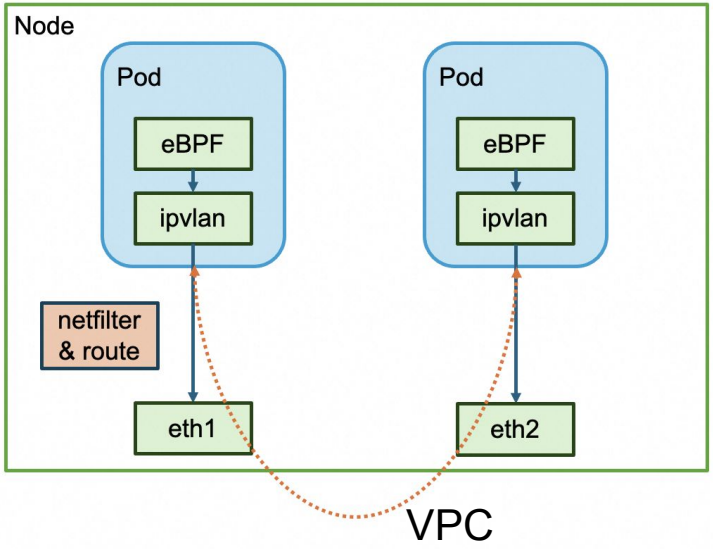
- KPR
  - 部分, 在容器内生效
- NetworkPolicy
  - K8s NetworkPolicy
  - CiliumClusterWideNetworkPolicy
    - Use ACK GlobalNetworkPolicy - Container Service for Kubernetes - Alibaba Cloud Documentation Center
- BandwidthManager
  - 出方向, EDT at kernel 5.10
- Hubble
  - Implement network observability by using ACK Terway and Cilium Hubble - Container - Alibaba Cloud

CNI Chaining



IPvlan+ eBPF模式下的限制

- 流量不会经过节点
  - 像nodelocal dns的支持，需要额外的转发规则
  - 监控组件可能需要而外适配才能采集Pod流量
- 连通性问题
  - NodePort 可能不能通
- 性能
  - 通节点上Pod间通信的流量，可能绕行VPC

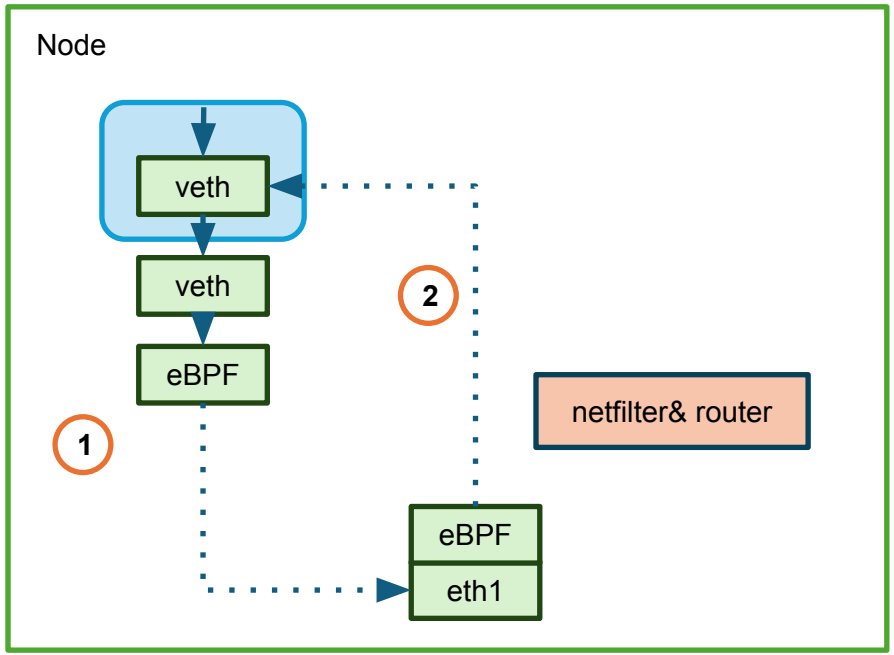


IPvlan + eBPF datapath

Datapath V2

Datapath V2 的数据链路，和普通链路很相似，只是通过eBPF进行了增强

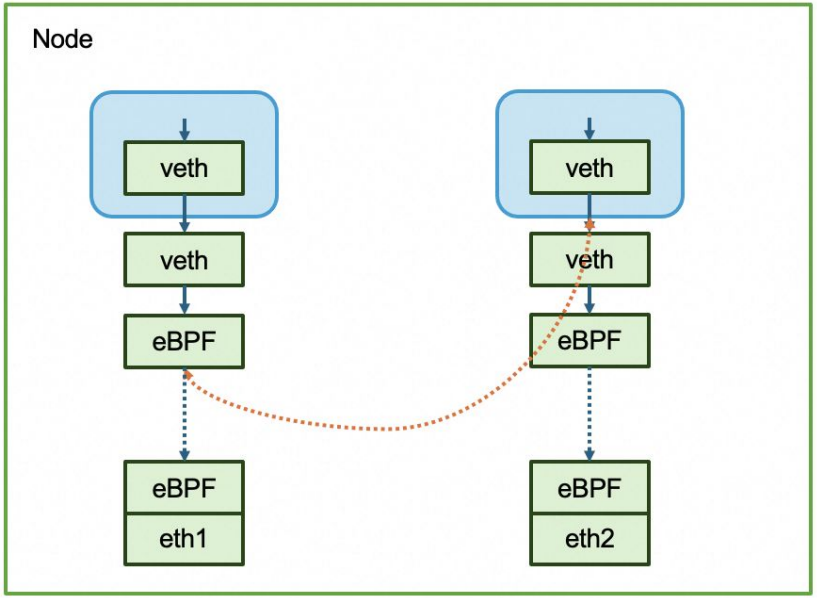
- Pod 出节点流量
  - bpf_redirect_neigh
- 回包
  - bpf_redirect_peer



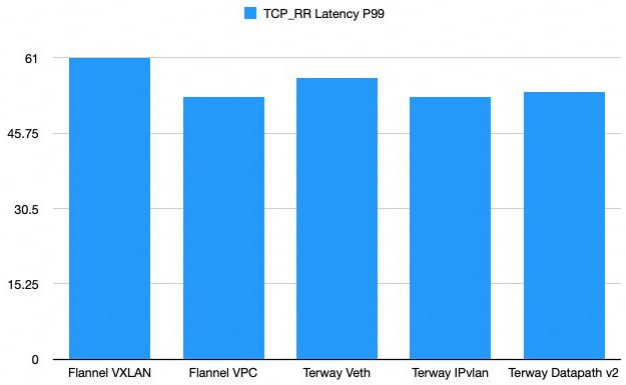
Datapath V2

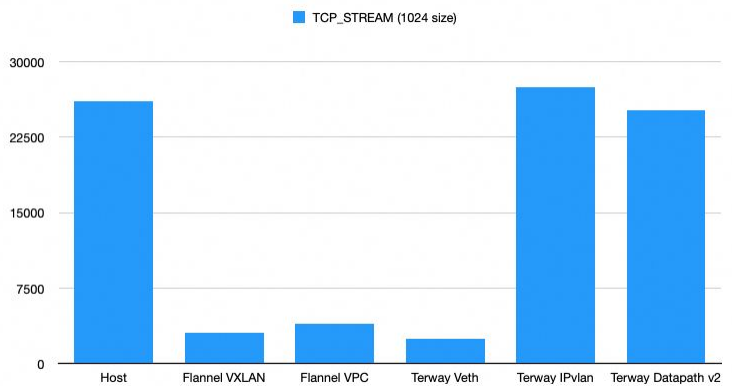
改进了兼容性

- 节点上可以跟踪Pod 流量
- 同节点Pod间通信，不会再绕行VPC



结果





- IPvlan模式下，Pod到Pod性能最佳
- Datapath V2 的性能远高于普通 veth 模式，和IPvlan模式非常接近
- 在有些测试场景下，Datapath V2性能会高于IPvlan模式

Cilium 最终用户案例研究

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

What’s next

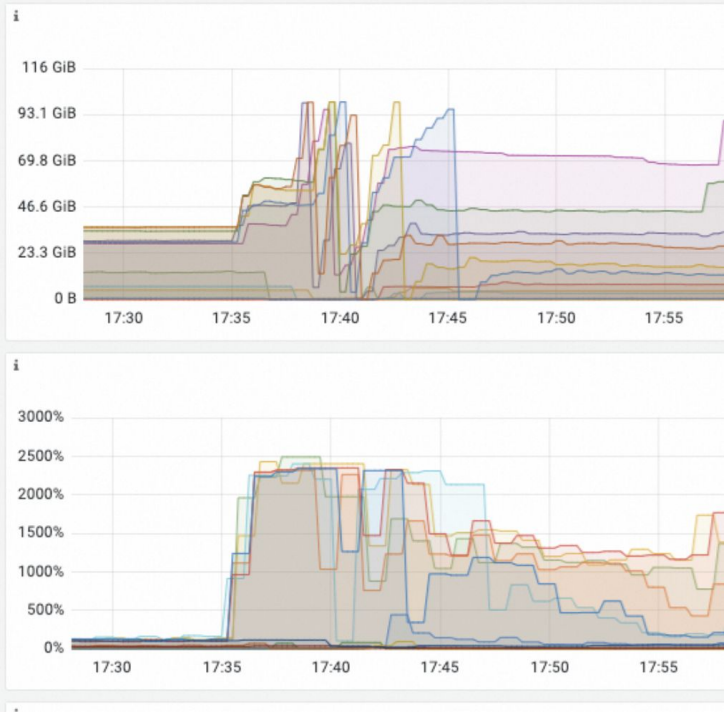
- netkit 提供了快速网络命名空间切换能力，可以加速出节点的流量
- Full kube-proxy replacement can simplify the deployment configuration
- 网络功能下沉可能是最终形态

运维灾难

修改一个 namespace 的标签
看起来并不会有什么影响 ...

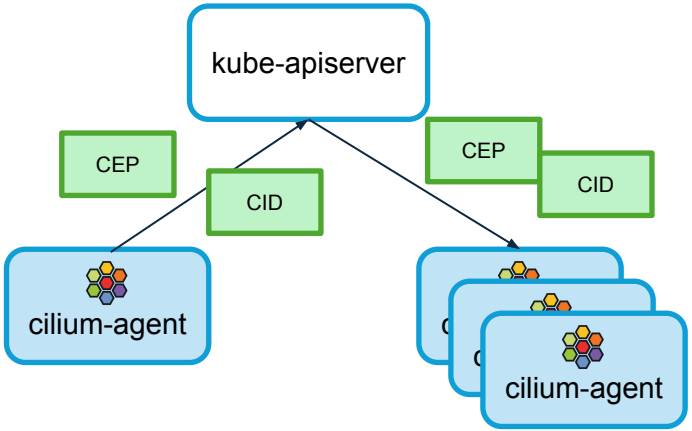
但是对kube-apiserver造成了巨大的压力

2K+ Nodes
80K+ Pods



Cilium 的资源

- CiliumEndpoint(CEP)
  - 跟踪Pod
  - 每个容器网络的Pod会有一个CEP与之对应
  - 存储了Pod标签、CiliumIdentity
- CiliumIdentity(CID)
  - 按Pod标签或者CIDR（在NetworkPolicy里面定义）生成
  - 只在NetworkPolicy功能中有用



阿里云上的优化

修改

- 按需watch资源
  - 为CEP增加了node标签，默认agent只需要关注本节点的资源
- 限制CID中使用的Pod标签
- 简化CEP资源的定义
- Pod标签不再同步到CEP标签
- 对Cilium资源进行了API限速

结果

- kube-apiserver内存消耗下降 82.5%
- 变更后恢复时间下降95%

在阿里云上完整的cilium功能

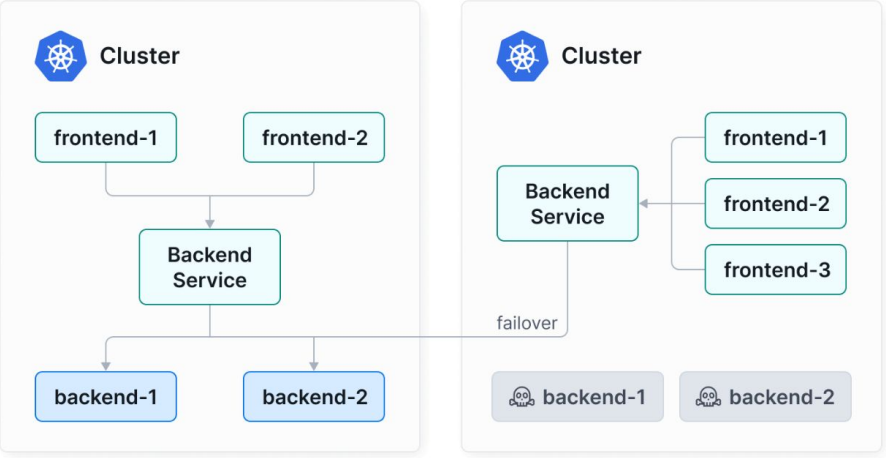
- 基于FQDN 或者HTTP的Cilium network policy

```
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: "*"
```

```
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'
```

在阿里云上完整的cilium功能

- 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

网络

- Cilium netkit: 容器网络和主机网络得到获得一样的性能

Service mesh & Ingress/Gateway API

- Gateway API GAMMA support:通过 Gateway API进行东西向流量
- Gateway API 1.1 support: Cilium 支持 Gateway API 1.1

安全

- 各种Cilium Network Policy的增强

更多细节查看 https://isovalent.com/blog/post/cilium-1-16/