Google Cloud

Gateway API と eBPF で進化する GKE Networking

Google Cloud
アプリケーション モダナイゼーション スペシャリスト
内間 和季

自己紹介

内間 和季

Google Cloud
(@kkuchima)

Application Modernization Specialist, Customer Engineer

好きな Google Cloud プロダクト:

- Google Kubernetes Engine
- Cloud Run
- Anthos Service Mesh

沖縄出身、大学時代は福岡に住んでました

Google Kubernetes Engine (GKE)

Google Cloud により
完全に管理されるマネージド Kubernetes

- 高度に自動化されたクラスタ管理機能
- ノードの管理が不要な Autopilot モード
- 高いスケーラビリティ(最大 15,000 ノード)
- 組み込みのセキュリティ機能
- Kubernetes エコシステムをマネージド サービスとして提供
  - Google Managed Prometheus (Prometheus)
  - Anthos Service Mesh (Istio)
  - Backup for GKE

運用の Toil を Google Cloud に任せることで
貴重なエンジニアリングリソースをコア業務に集中

GKE の主なネットワーク関連アップデート

直近 1 年分 (2022 年 8 月 ~ 2023 年 7 月)

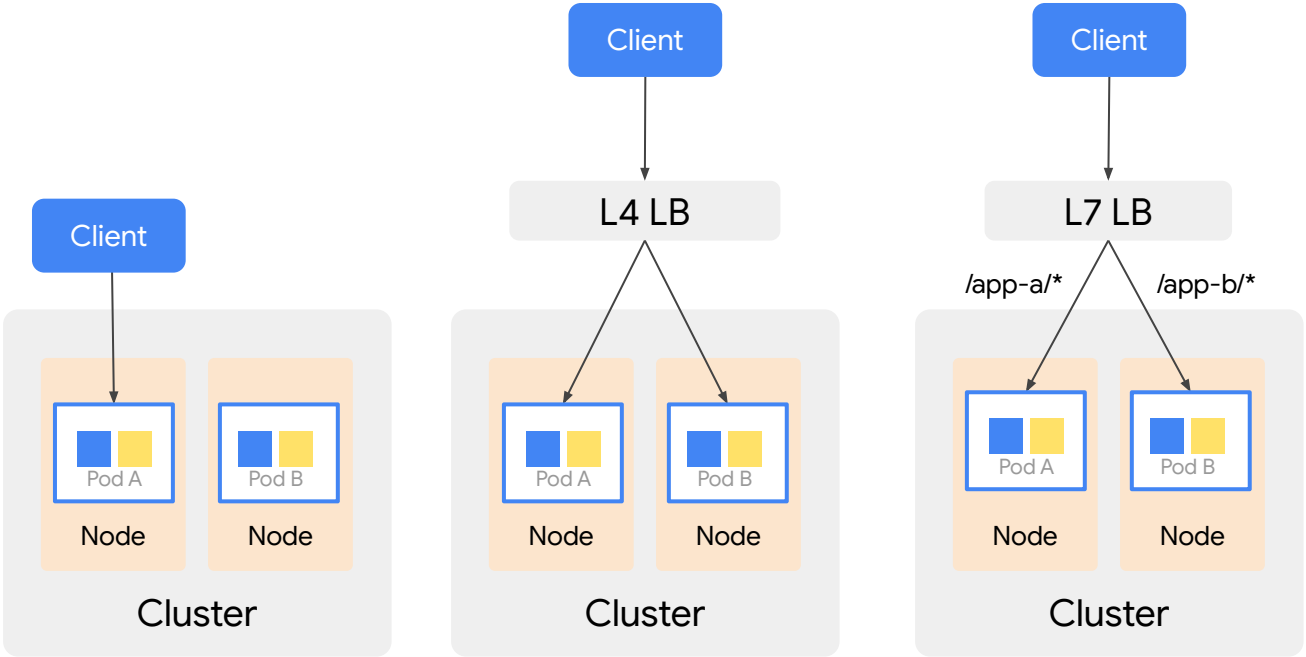
- **GKE Gateway Controller** (Single Cluster) が GA
- **GKE Gateway Controller** 機能追加
  - カスタムリクエスト・レスポンスヘッダ
  - URL リライト / パスリダイレクト
  - HTTP-to-HTTPS リダイレクト
  - SSL Policies
  - Global External Application Load Balancer サポート
  - Cloud Armor 連携
  - Identity-aware Proxy (IAP) 連携
  - Service Mesh Cloud Gateway が GA
- **Dataplane V2** 機能追加
  - Dataplane V2 Observability (Hubble サポート)
  - FQDN Network Policy サポート
- GKE Autopilot クラスタの **Dataplane V2** への自動移行
- IPv4 / IPv6 デュアルスタックが GA
- IPv4 / IPv6 デュアルスタックで LoadBalancer Service をサポート
- GKE Autopilot での Pod IP レンジ追加 (multi-Pod CIDR)
- Service IP レンジを複数クラスタで共有可能に
- Cloud DNS for GKE (Cluster Scope) が GA、等

Gateway...? Dataplane V2...?

Gateway API

Kubernetes でサービスを公開する際の一般的な方法

Service / Ingress

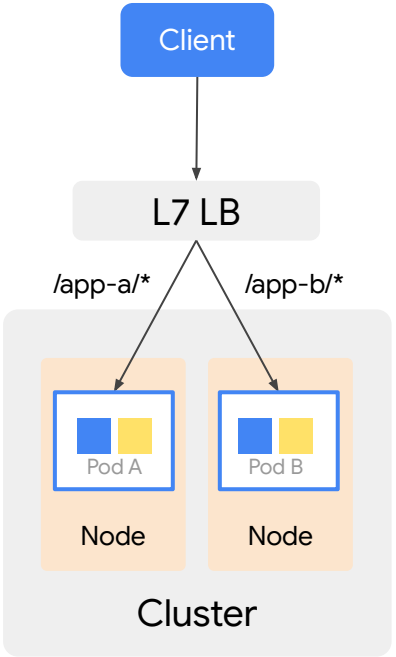


Ingress

サービスを外部公開する際に用いられる API リソース

L7 での負荷分散を提供

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-wildcard-host
spec:
  rules:
  - host: "foo.bar.com"
    http:
      paths:
      - pathType: ImplementationSpecific
        path: "/app-a"
        backend:
          service:
            name: app-a
            port:
              number: 80
```



Gateway API

サービスを外部公開する際に用いられる **次世代の API リソース**

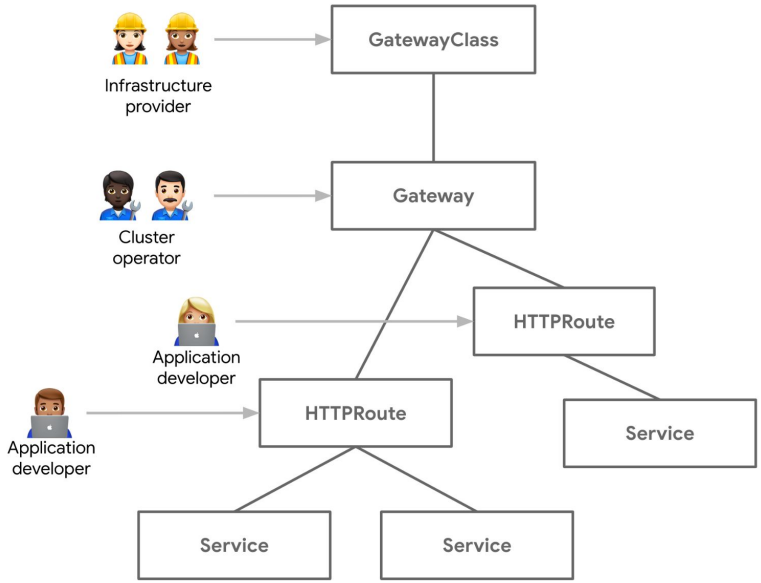
Kubernetes SIG-Network Community を中心に開発

GKE や Istio など多くの実装が存在

Ingress の課題を解決するために開発された

主要な API リソース

- GatewayClass
- Gateway
- HTTPRoute
- TCPRoute, etc



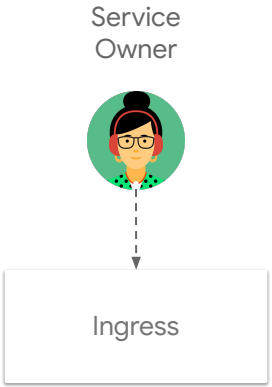
https://gateway-api.sigs.k8s.io/

Gateway API の特徴

- **ロール志向なリソースモデル**
  - Ingress は（基本的に）単一リソースで構成されるが、Gateway は 複数リソースから構成
  - インフラプロバイダー、クラスタ管理者、開発者などロールによる権限の分離を可能に
- **多くの機能をサポート**
  - Ingress では Annotation で表現していた機能や、利用できなかった機能をネイティブにサポート
    - ヘッダーベース ルーティング、重み付けベースのトラフィック分割、等
  - TCP や gRPC プロトコルのサポート *1
- **高い拡張性**
  - Custom Resource による拡張を前提としたデザイン

*1 ... 2023.08 現在 Experimental channel でのみ利用可能

ロール志向なリソースモデル



・各 Namespace / アプリケーション単位で
Ingress が乱立（ガバナンスを効かせにくい）

・開発者側でインフラ寄りのタスクも行うケースも（証明書や IP アドレスの管理など）

ロール志向なリソースモデル

Service Owner

Ingress

Service Owner

Ingress

Platform Provider

GatewayClass

gateway Namespace

Platform Admin

Gateway

RBAC による制御

foo Namespace

HTTPRoute

Service Owner

Services

bar Namespace

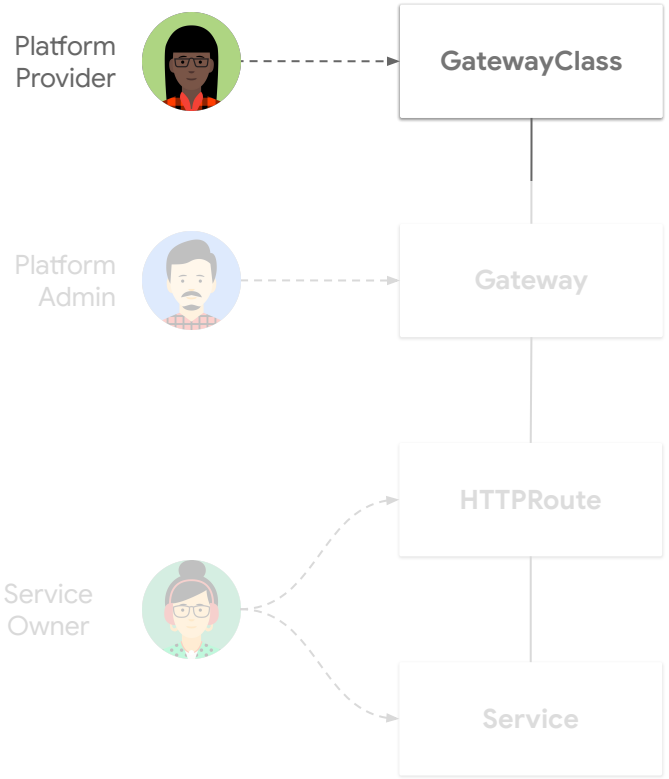
HTTPRoute

Services

Service Owner

Namespace を跨いだリソース配置により、各ロールで操作可能なリソースを制限し**権限を分離**することが可能

GatewayClass リソース



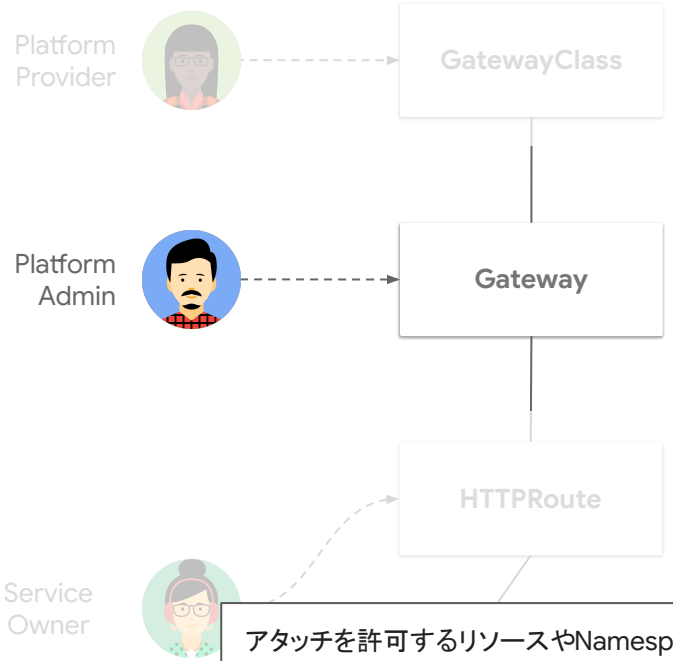
各種 Load Balancer や Service Mesh など物理 / 論理的な Gateway Device を抽象化

例 : GKE でサポートされている GatewayClass

GatewayClass 名	説明
gke-l7-global-external-managed	グローバル外部アプリケーション ロードバランサ上にビルドされたグローバル外部アプリケーション ロードバランサ
gke-l7-regional-external-managed	リージョン外部アプリケーション ロードバランサ上にビルドされたリージョン外部アプリケーション ロードバランサ
gke-l7-rilb	内部アプリケーション ロードバランサ上にビルドされた内部アプリケーション ロードバランサ
gke-l7-gxlb	従来のアプリケーション ロードバランサ上にビルドされたグローバル外部アプリケーション ロードバランサ
gke-l7-global-external-managed-mc	グローバル外部アプリケーション ロードバランサ上にビルドされたマルチクラスタ グローバル外部アプリケーション ロードバランサ
gke-l7-regional-external-managed-mc	グローバル外部アプリケーション ロードバランサ上にビルドされたマルチクラスタ リージョンの外部アプリケーション ロードバランサ
gke-l7-rilb-mc	内部アプリケーション ロードバランサ上にビルドされたマルチクラスタの内部アプリケーション ロードバランサ
gke-l7-gxlb-mc	従来のアプリケーション ロードバランサ上にビルドされたマルチクラスタ グローバル外部アプリケーション ロードバランサ
gke-td	マルチクラスタ Traffic Director サービス メッシュ
asm-l7-gxlb	Anthos Service Mesh 上にビルドされたグローバル外部アプリケーション ロードバランサ

Gateway リソース

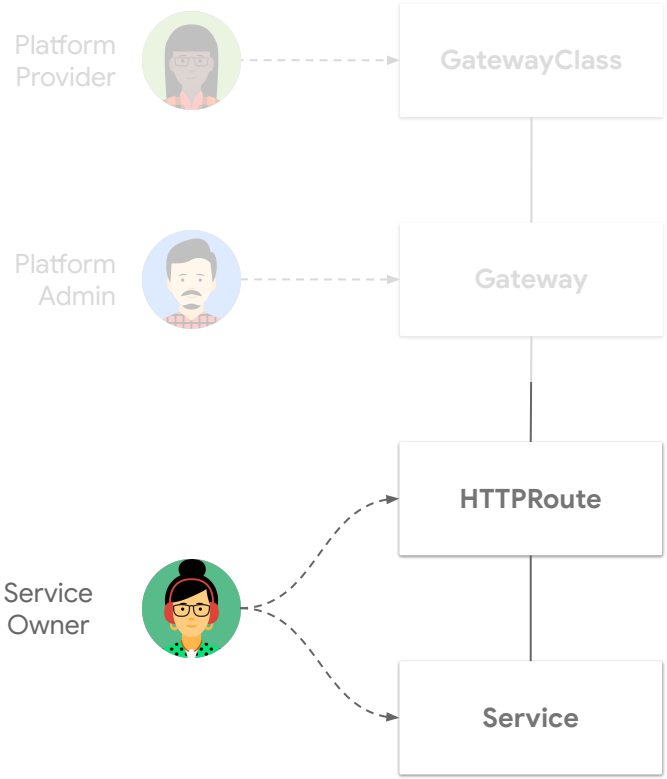
トラフィックを受け付ける IP アドレスやプロトコル、TLS 終端の設定等を定義



アタッチを許可するリソースやNamespaceを指定することも可能
（デフォルトは同一Namespace 上のリソースのみ許可）

```yaml
kind: Gateway
apiVersion: gateway.networking.k8s.io/v1beta1
metadata:
  name: external-http
  namespace: gateway
spec:
  gatewayClassName: gke-l7-global-external-managed
  listeners:
  - name: https
    protocol: HTTPS
    port: 443
    tls:
      mode: Terminate
      options:
        networking.gke.io/pre-shared-certs: example-com
    allowedRoutes:
      kinds:
      - kind: HTTPRoute
      namespaces:
        from: Selector
        selector:
          matchLabels:
            kubernetes.io/metadata.name: store
```

HTTPRoute リソース

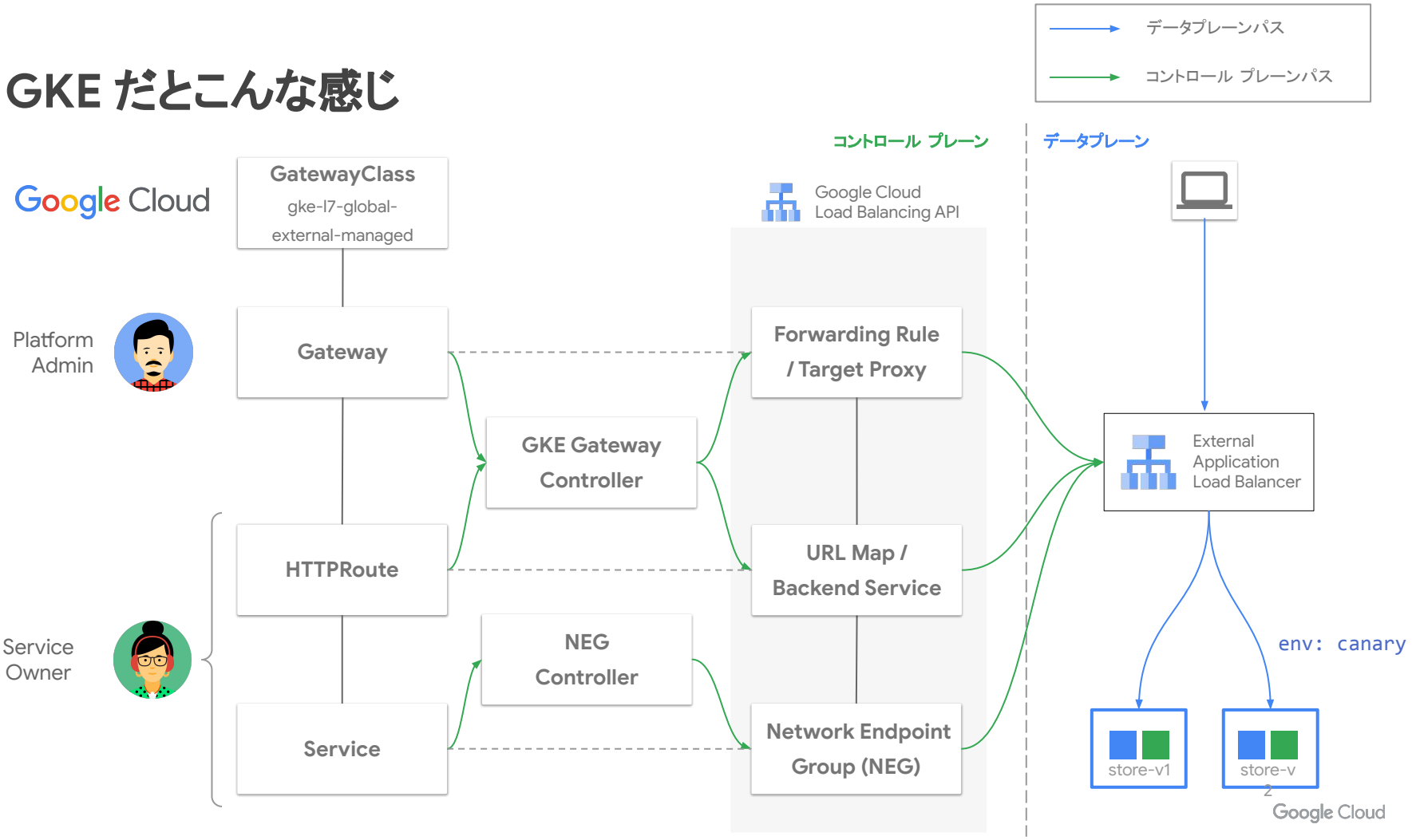


HTTP トラフィックのルーティングを定義する

ホスト名やルーティング条件等を設定

```yaml
kind: HTTPRoute
apiVersion: gateway.networking.k8s.io/v1beta1
metadata:
  name: store-external
  namespace: store
spec:
  parentRefs:
  - kind: Gateway
    name: external-http
  hostnames:
  - "store.example.com"
  rules:
  - backendRefs:
    - name: store-v1
      port: 8080
  - matches:
    - headers:
      - name: env
        value: canary
    backendRefs:
    - name: store-v2
      port: 8080
```

GKE だとこんな感じ


データプレーンパス
コントロール プレーンパス
コントロール プレーン
データプレーン
Google Cloud
GatewayClass
gke-l7-global-external-managed
Google Cloud Load Balancing API
Platform Admin
Gateway
Forwarding Rule / Target Proxy
GKE Gateway Controller
External Application Load Balancer
HTTPRoute
URL Map / Backend Service
Service Owner
NEG Controller
env: canary
Service
Network Endpoint Group (NEG)
store-v1
store-v2
Google Cloud

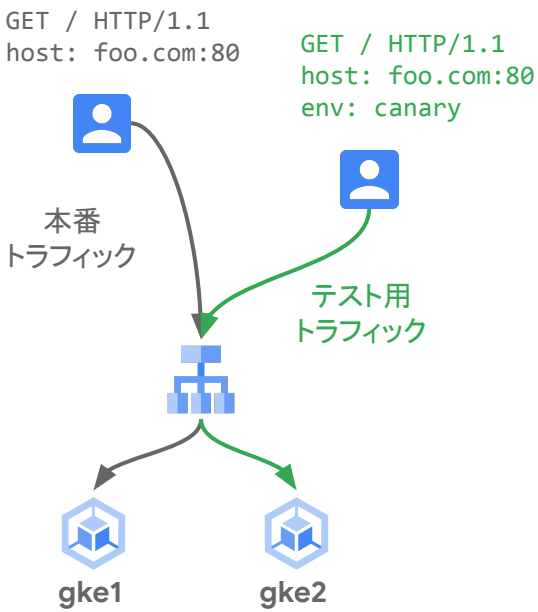
サービスメッシュとの統合

Istio, Kuma, Linkerd 等のサービスメッシュプロダクトでも Gateway API をサポート

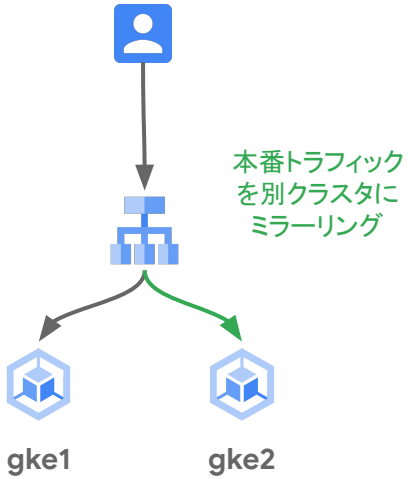
GAMMA (Gateway API for Mesh Management and Administration) initiative がリードしており、
Kubernetes だけでなく各サービスメッシュを含めた Service API のスタンダードを目指している

Google Cloud のサービスメッシュ プロダクト (Anthos Service Mesh, Traffic Director) でも
一部 Gateway API をサポート

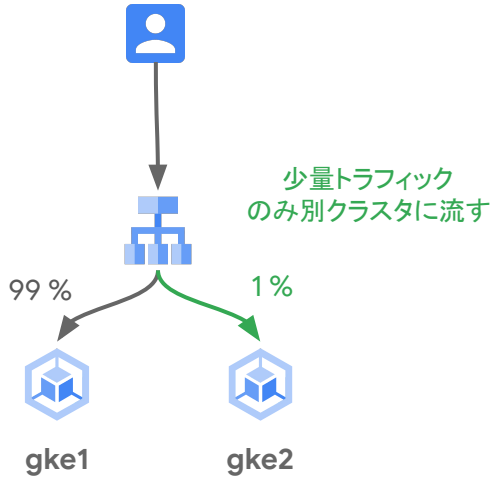
マルチクラスタでの高度なトラフィック制御



ヘッダベース
ルーティング



トラフィック
ミラーリング



トラフィック分割

Dataplane V2

Dataplane V2

iptables / Calico ベースのデータプレーンに代わる、

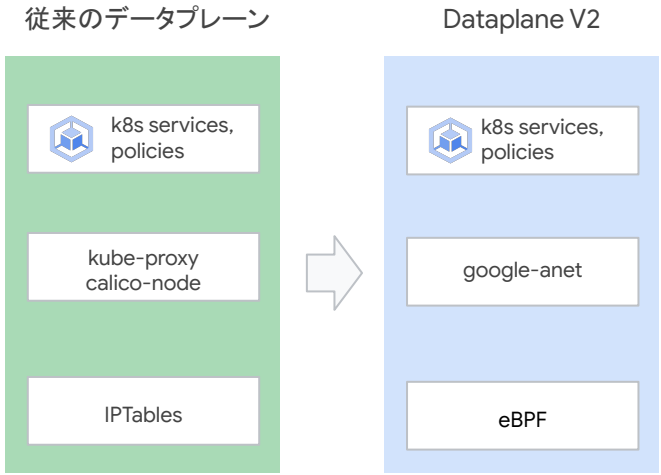
次世代のネットワーク データプレーン

eBPF / Cilium をベースに実装されており、

従来のデータプレーンと比べて、高い拡張性・スケーラビリティを実現

Observability やセキュリティ関連の advanced な機能を提供

- Network Policy Logging
- FQDN Network Policy
- Managed Hubble, etc



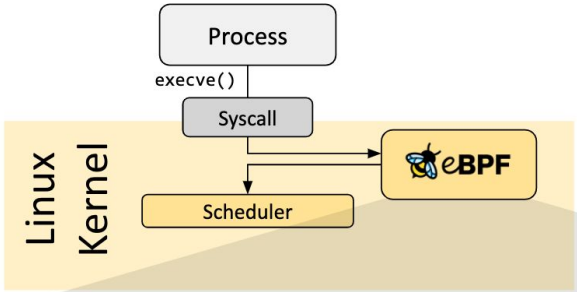
eBPF (extended Berkeley Packet Filter)

カーネルのソースコード変更やカーネルモジュールのロードをせずに、プログラムを組み込む機能

Linux カーネルをセキュアかつ効率的な方法で Programmable にする

eBPF プログラムはイベント駆動であり、システムコール等特定のフックポイントで実行される

- system calls
- kernel tracepoints
- network events, etc




```
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;
}
```

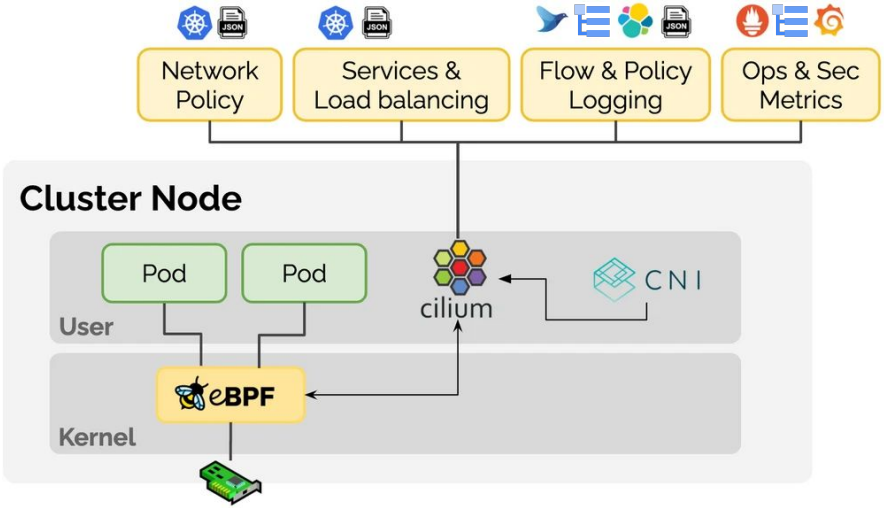
https://ebpf.io/what-is-ebpf/

Cilium

コンテナ化された環境における **ネットワーキング** や
透過的な Observability、セキュリティ機能 を提供

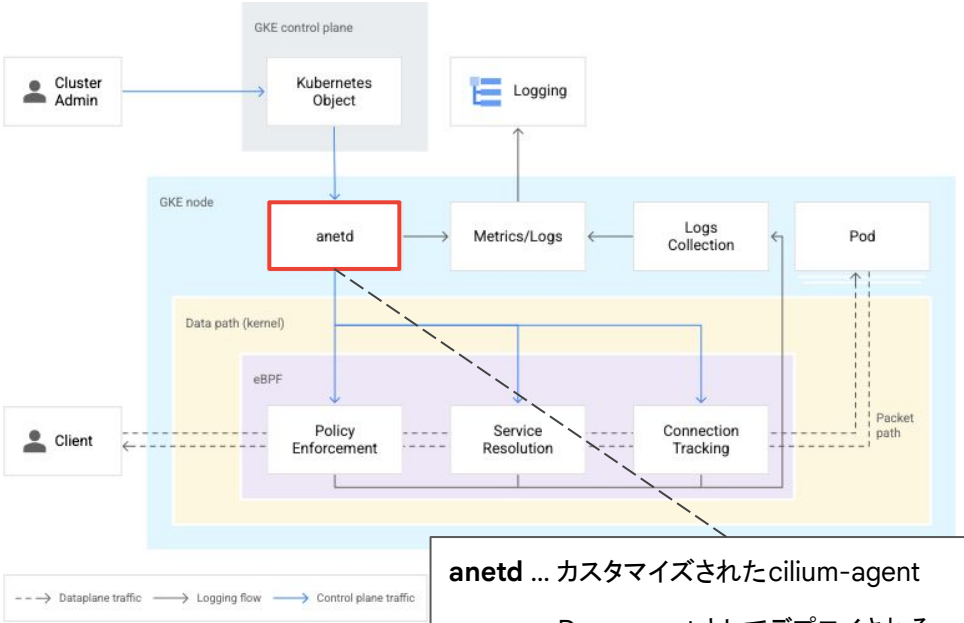
CNI やサービスメッシュなどの機能で eBPF を活用

iptables ベースの実装と比べ、高いパフォーマンス・スケーラビリティを実現（特に大規模な環境で）



https://docs.cilium.io/

Dataplane V2 のアーキテクチャ



anetd … カスタマイズされたcilium-agent

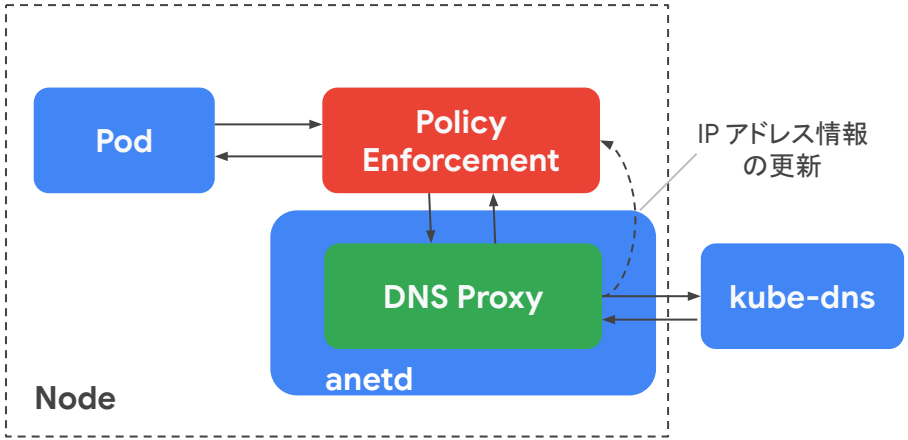
Daemonset としてデプロイされる

```
$ kubectl -n kube-system get cm cilium-config
-o yaml
apiVersion: v1
data:
~
  custom-cni-conf: "true"
  enable-bpf-masquerade: "false"
  enable-fqdn-network-policy: "false"
  enable-google-multi-nic: "false"
  enable-google-service-steering: "false"
  enable-hubble: "true"
  enable-hubble-open-metrics: "false"
  enable-remote-node-identity: "true"
  enable-service-topology: "true"
  enable-well-known-identities: "false"
  enable-local-node-route: "false"
  ipam: kubernetes
  kube-proxy-replacement: strict
  tunnel: disabled
~
```

FQDN Network Policy Public Preview

FQDN による Network Policy を設定可能に
（ワイルドカードもサポート）

DNS による名前解決結果を基にポリシーを適用




```yaml
apiVersion: networking.gke.io/v1alpha1
kind: FQDNNetworkPolicy
metadata:
  name: allow-out-fqdnnp
spec:
  podSelector:
    matchLabels:
      app: curl-client
  egress:
  - matches:
    - pattern: "*.yourdomain.com"
    - name: "www.google.com"
    ports:
    - protocol: "TCP"
      port: 443
```

Dataplane V2 Observability Public Preview

Traffic flows の収集・可視化をサポート

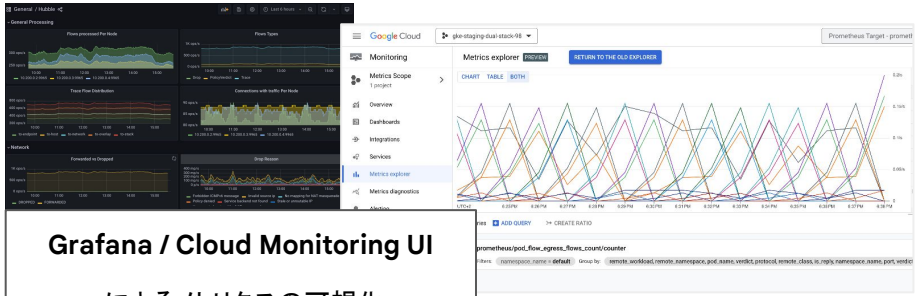
Dataplane v2 / Hubble metrics を Google Managed Prometheus や Cloud Monitoring へ連携

以下のコンポーネントがデプロイされる

- **Hubble Relay**: 各ノードから Pod のネットワーク テレメトリ データを収集
- **Hubble CLI**: クラスタ内のライブ トラフィック情報を提供
- **(Optional) Hubble UI**: Hubble Relay によって収集されたネットワーク テレメトリ データの表示や分析

Hubble UI による

サービストポロジの可視化



Grafana / Cloud Monitoring UI

によるメトリクスの可視化

Google Cloud

まとめ

まとめ

- Gateway API はサービスを外部公開する際に用いられる **次世代の API リソース**
  - ロール志向なリソースモデルを採用しており、ロールによる権限分離を実現
  - 拡張性が高く、また Ingress ではサポートしてない機能もネイティブにサポート
- eBPF は**カーネルのソースコード変更なしにプログラムを組み込む** ことができる機能
  - eBPF 実装の 1 つとして Cilium があり、アプリケーション間の接続性や透過的なセキュリティ、Observability を提供
  - GKE では eBPF / Cilium をベースにした Dataplane V2 が利用可能
- GKE では **Gateway API や Dataplane v2 がベースとなった機能を多数提供**
  - **シングルクラスタ用 Gateway と Dataplane V2 が General Availability (GA) で利用可能**
  - 今後も多くのネットワーク関連機能が追加される予定！お楽しみに！

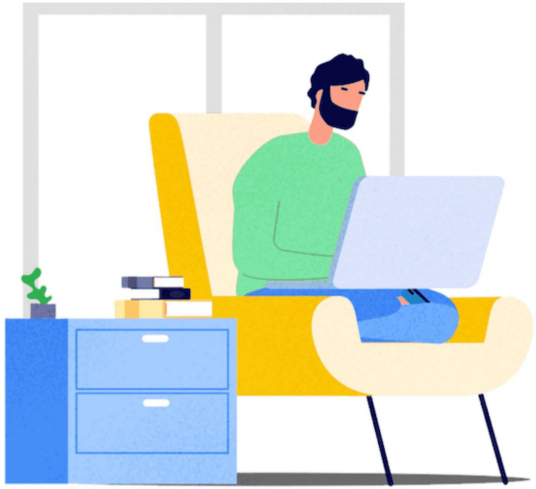
ハンズオン

Google Kubernetes Engine (GKE) 道場 ~入門編~

9 月 7 日（木）15:00 Live 配信

Google Kubernetes Engine (GKE) 道場 ~応用編~

9 月 8 日（金）15:00 Live 配信



応募はこちら

応募資格

➔ Google Cloud を利用したことがあり、開発者としてクラウド上で開発を行いたい、また環境の提供側として開発環境の構築に興味のあるエンジニア

Next Tokyo ’23 開催決定

11 月 15 日 (水), 16 日 (木) @ 東京ビッグサイト

参加登録 受付中

ご登録はこちら

Thank you.

Google Cloud