

Service mesh на примере Istio.

Service mesh

Service mesh - это дополнительный уровень инфраструктуры, обеспечивающий коммуникации между сервисами и реальным миром.

Он обрабатывает трафик приложения, как входящий, так и исходящий. На этом уровне может быть реализована следующая функциональность:

- Аутентификация и авторизация
- Телеметрия
- Перенаправление трафика
- Обработка ошибок
- Service discovery
- Логирование

Минусы использования Service mesh:

- Дополнительные затраты ресурсов, необходимые для поддержания работы Service mesh
- Увеличение времени ответа сервиса, так как трафик проходит дополнительную обработку
- Сложность поддержки. Обычно, реализация service mesh сложна в эксплуатации

Плюсы использования Service mesh:

- Централизованная настройка функционала
- Сам функционал, предлагаемый Service mesh

Стоит учесть, что пункт про функционал добавлен с расчетом на то, что для вас все минусы Service mesh перекрываются необходимостью использования функционала с ним связанного.

Istio

Istio - одна из реализаций Service mesh. Он умеет работать с различными облачными платформами, такими как: Kubernetes, Nomad, GKE и т.д.

Архитектура Istio представляет собой два уровня - уровень данных и уровень управления.

Уровень данных - совокупность приложений и sidecar-контейнеров с прокси сервером Envoy. Именно через этот прокси сервер проходит весь трафик приложения.

Уровень управления - совокупность сервисов, выполняющих функцию настройки уровня данных. Он управляет конфигурацией проху-серверов, сертификатов, добавляет sidecar-контейнер в приложение.

Envoy - opensource прокси. Он умеет работать с различными протоколами: HTTP, ZooKeeper, MongoDB, Mysql и т.д. Трафик в нем обрабатывается в несколько этапов:

- Listener - слушает порт и принимает трафик
- Filter chain - Содержит правила определения трафика и список фильтров для его обработки
- Filter - обрабатывает трафик, может направить его в Cluster
- Cluster - Внешние сервисы, которые можно использовать для отсылки трафика туда

Управление трафиком в Istio

Istio предлагает три основные сущности для управления трафиком:

- Sidecar - ограничивает входящий/исходящий трафик
- DestinationRule - тонкая настройка точки доступа, предлагает такую функциональность, как обработку ошибок, настройку таймаутов.
- VirtualService - настройка маршрутизации трафика. Предлагает возможности по перенаправлению трафика и по изменению ответов приложения.

- EnvoyFilter - позволяет полностью менять конфигурацию проху, вставляю куски конфигурации Envoy напрямую.

С помощью этого можно сделать такие вещи, как:

- Ограничение исходящего трафика
- Обработка ошибок
- Разделение трафика
- Зеркалирование трафика
- Внедрение ошибок
- Метрики
- Изменение запросов и ответов