



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Service Mesh в production: как мы внедряем Istio и Linkerd

Наша методология: Linkerd 2.20 для mTLS и метрик, Istio 1.30 ambient для canary и circuit brea...

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Трафик между микросервисами в Kubernetes по умолчанию — открытый HTTP: компрометация одного пода даёт перехват всего обмена, включая платёжные данные. Ретраи и таймауты размазаны по коду, canary делается руками, диагностика «500» без трассировки занимает часы. Мы закрываем это service mesh: mTLS, метрики и управление трафиком без правки кода.

Почему это важно бизнесу

- Незашифрованный внутренний трафик — нарушение требований 152-ФЗ и PCI DSS при работе с персональными и платёжными данными
- Каждый час поиска виновника сбоя среди десятков сервисов без трассировки — простой продаж и авральная работа инженеров
- Релизы «сразу на 100%» без canary превращают каждый деплой в лотерею с риском остановки всего сервиса
- Логика ретраев в коде каждого сервиса — разноречивой, каскадные отказы и дорогие доработки; mesh выносит её в инфраструктуру



Ключевые параметры реализации

1.30.x

Актуальная ветка Istio:
поддерживаются два последних
минора, совместимость с K8s
1.32–1.36
по докам istio.io

2.20

Стабильная ветка Linkerd; OSS
публикует weekly edge, поэтому
версию жёстко пиним в Helm values
по докам linkerd.io

+0,5–1 мс

Надбавка p99 latency linkerd2-proxy
(Rust) — фиксируем нагрузочным
прогоном до вывода в прод
наш стандарт

5 / 30 с

outlierDetection: 5 подряд 5xx за
interval 30s — инстанс выводится из
балансировки на 60 с
DestinationRule, Istio 1.30

48 ч

Issuer-сертификат mTLS Linkerd под
cert-manager (renewBefore 25 ч);
leaf-сертификаты прокси жи...
доки Linkerd + cert-manager

×3

Реплики control plane: linkerd install
--ha и HPA для istiod — inject
переживает отказ ноды
наш стандарт



Базовый mesh на Linkerd 2.20: mTLS и метрики за один день

Что настраиваем

Кластер на 20–60 микросервисов; control plane в namespace linkerd, viz отдельно; версии зафиксированы в Helm values

Как мы это делаем

- 1 Ставим чарты linkerd-crds + linkerd-control-plane в HA-режиме: 3 реплики destination/identity/proxy-injector, PodDisruptionBudget на control plane
- 2 PKI: trust anchor (год) и issuer выпускает cert-manager; issuer живёт 48 ч и перевыпускается за 25 ч до истечения, алерт за 60 дней до истечения якоря
- 3 Включаем mesh по namespace: kubectl annotate ns prod linkerd.io/inject=enabled + rollout restart — mTLS работает сразу, без единой строки конфигурации
- 4 Наблюдаемость: linkerd viz + federation метрик в наш Prometheus; success rate, RPS, p50/p99 по каждому deployment
- 5 linkerd check и linkerd check --proxy защиты в CI/CD — деплой не идёт при деградации control plane

РЕЗУЛЬТАТ

100% внутреннего трафика зашифровано mTLS без правки кода; поиск сервиса, который сыплет 5xx, — минуты по дашборду viz вместо часов по логам; надбавка p99 в наших прогонах — до 1 мс, около 20 МБ RAM на под

КЛЮЧЕВОЙ НЮАНС

OSS Linkerd публикует только weekly edge-релизы — в прод берём стабильную ветку 2.20 и пиним дайджесты образов; тег latest в кластере недопустим



Canary и circuit breaker: Istio 1.30 в ambient-режиме

Что настраиваем

Namespace-ы с релизными сервисами; ztunnel как DaemonSet на всех нодах, waypoint — только там, где нужна L7-логика

Как мы это делаем

- 1 `istioctl install --set profile=ambient: istiod + ztunnel (DaemonSet) + CNI-плагин; sidecar-ов нет, per-pod overhead нулевой`
- 2 `kubectl label ns prod istio.io/dataplane-mode=ambient` — mTLS на L4 включается мгновенно, поды не перезапускаются
- 3 PeerAuthentication mode: STRICT на namespace — запрещаем plaintext после проверки, что все клиенты уже в mesh
- 4 Для canary ставим waypoint: `istioctl waypoint apply -n prod + label istio.io/use-waypoint` на сервис — L7 обрабатывает выделенный Envoy
- 5 VirtualService: weight 90/10 между subset v1/v2, retries attempts: 2, retryOn: 5xx,connect-failure, timeout 10s; DestinationRule с outlierDetection

РЕЗУЛЬТАТ

Canary двигаем 10→50→100% правкой одного YAML в git; сбойный инстанс автоматически выводится из балансировки за 30 с; включение mesh не потребовало ни одного рестарта подов и ни одного изменения в коде

КЛЮЧЕВОЙ НЮАНС

В ambient без waypoint работают только L4-политики: VirtualService/HTTPRoute молча не применяются. Перед тем как доверять канарейке, проверяем `istioctl waypoint list` и привязку сервиса



Эксплуатация mesh: апгрейды, алерты, отказоустойчивость

Что настраиваем

Оба mesh на нашей поддержке: регламент апгрейдов, мониторинг control plane и data plane, тесты отката

Как мы это делаем

- 1 Istio обновляем только по ревизиям: `istioctl install --revision 1-30-3`, namespace-ы переключаем label-ом `istio.io/rev` по одному, старую ревизию сносим через неделю
- 2 Linkerd: сначала staging-кластер, затем прод в окно; `helm upgrade linkerd-control-plane` с теми же values, после — обязательный `linkerd check`
- 3 Алерты: `success rate < 99,5%` за 5 мин, `p99` выше SLO, недоступность `istiod/destination`, сроки сертификатов `identity`
- 4 Ресурсы прокси: `config.linkerd.io/proxy-cpu-request` и `sidecar.istio.io/proxyCPU`, `proxyMemory` — лимиты в каждом манифесте
- 5 Ежеквартальный нагрузочный прогон: сверяем надбавку `latency` и `RAM` с зафиксированной базовой линией

РЕЗУЛЬТАТ

Minor-апгрейды проходят без даунтайма и без массового рестарта нагрузки; регрессии ловим на staging; на любом инциденте есть ответ «что менялось в mesh» из git-истории конфигурации

КЛЮЧЕВОЙ НЮАНС

Istio поддерживает только два последних минора — отставание на три версии превращается в апгрейд-марафон без security-патчей. Держим отставание не больше одного минора



Подводные камни

✗ Mesh на кластере из 5-10 сервисов

Overhead не окупается: до ~20 сервисов хватает NetworkPolicy и Prometheus; mesh вводим при росте, canary или требованиях к шифрованию

✗ PERMISSIVE mTLS остаётся навсегда

Миграционный режим забывают в проде — plaintext по-прежнему принимается. После охвата всех клиентов включаем PeerAuthentication STRICT per namespace

✗ Прод на edge-релизах Linkerd

Weekly edge — для тестов; в проде пиним стабильную ветку и дайджесты образов в Helm values, обновляем только через staging

✗ Истёкший issuer положил весь mTLS

Issuer, созданный linkerd install, живёт 365 дней; про ротацию забывают — через год встаёт весь меж-сервисный трафик. Ставим cert-manager (issuer 48...

✗ Апгрейд istiod «на месте» без ревизий

In-place апгрейд разом меняет прокси всех namespace. Делаем canary-ревизию (--revision) и переносим namespace-ы по одному через label istio.io/rev

✗ Ретраи дублируются на трёх уровнях

Retry в коде + в mesh + на ingress дают шторм при отказе. Оставляем ретраи в mesh (attempts: 2), из кода убираем, на ingress — только идемпотентные G...

✗ Прокси без ресурсных лимитов

200 подов × 50 МБ sidecar — гигабайты «невидимого» RAM. Задаём requests/limits аннотациями или уходим в ambient, где ztunnel один на ноду

✗ L7-политики в ambient без waypoint

VirtualService/HTTPRoute тихо не применяются, пока сервис не привязан label-ом istio.io/use-waypoint — проверяем привязку перед каждым релизом

Как правильно

МИНИМУМ

- Linkerd 2.20 с mTLS на прод-namespace; linkerd check в CI/CD-пайплайне
- Версии и Helm values в git, дайджесты образов записаны
- Дашборд success rate / RPS / p99 по сервисам + алерт на рост 5xx

НОРМАЛЬНО

- cert-manager: issuer 48 ч (renewBefore 25 ч), алерт за 60 дней до истечения якоря
- PeerAuthentication STRICT — запрет plaintext после охвата всех клиентов
- HA control plane: 3 реплики, PodDisruptionBudget, лимиты прокси
- Staging-кластер для прогона каждого апгрейда mesh

ХОРОШО

- Istio 1.30 ambient + waypoint: canary 90/10 и outlierDetection
- Revision-based апгрейды istiod, перенос namespace по label istio.io/rev
- Трейсинг через OTEL-коллектор с прокидкой traceparent в приложениях
- Multi-cluster: federated services Linkerd / ambient multicloud (beta)



Чек-лист самопроверки

- | | |
|--|---|
| ☐ Зашифрован ли трафик между сервисами (mTLS) и отключён ли приём plaintext (режим STRICT)? | ☐ Переживёт ли инжект новых подов отказ ноды с control plane (HA-реплики, PDB)? |
| ☐ Известны ли сроки истечения trust anchor и issuer, настроены ли автопродление и алерт? | ☐ Ретраи и таймауты настроены в одном месте, а не в коде, mesh и ingress одновременно? |
| ☐ Зафиксированы ли версии mesh в git, или прод тянет latest/edge-образы? | ☐ Есть ли отработанный сценарий отката апгрейда mesh (ревизии Istio, staging для Linkerd)? |
| ☐ Замерена ли надбавка p99 и потребление RAM после включения mesh на нагрузочном стенде? | ☐ Может ли дежурный за 5 минут увидеть, какой сервис отдаёт 5xx и где выросла латентность? |

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Аудит кластера: считаем, окупится ли mesh, и выбираем Linkerd, Istio ambient или NetworkPolicy без mesh
- Внедрение под ключ: mTLS, метрики, canary и circuit breaker — без изменения кода приложений
- PKI-контур: cert-manager, автоматическая ротация сертификатов identity, алерты об истечении
- Мониторинг mesh дежурной службой: дашборды, алерты на success rate и здоровье control plane
- Регламентные апгрейды: staging-прогон, canary-ревизии istiod, контролируемый откат

15+

лет в ИТ-поддержке

50

рабочих мест — наш профиль

МТС

дата-центр, Москва



КОНТАКТЫ

Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru



Техническая база

- 01** Ambient Mode / Dataplane modes (ztunnel, waypoint) (istio.io — 1.30)
- 02** Traffic Management: VirtualService, DestinationRule (istio.io — 1.30)
- 03** Canary Upgrades (revision-based istiod) (istio.io — 1.30)
- 04** Releases and Versions; Announcing Linkerd 2.20 (linkerd.io — 2.20)
- 05** Automatically Rotating Control Plane TLS Credentials (linkerd.io — 2.20)
- 06** High Availability installation (linkerd install --ha) (linkerd.io — 2.20)
- 07** Наш шаблон Helm values: пиннинг версий, HA, лимиты прокси (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

