Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Grafana Loki: лёгкое централизованное логирование без ELK

Как мы разворачиваем Loki 3.x + Alloy для парков до 50 РМ: сбор, ретенция, алерты

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Логи разбросаны по десяткам серверов и контейнеров: nginx, syslog, Docker, Windows Event Log. При инциденте инженер вслепую ходит по машинам через SSH и grep, разбор растягивается на часы. ELK для парка до 50 PM — избыточен: три ноды Elasticsearch съедают RAM и требуют ежемесячной перетряски конфигов. Мы ставим Loki: индексируется только набор лейблов, а не тело строки, поэтом...

Почему это важно бизнесу

- Простой сервиса без централизованных логов — прямые потери выручки и SLA-штрафы
- Часы работы инженера на ручной grep по узлам вместо одного LogQL-запроса
- Инцидент без истории логов = нет разбора первопричины, повтор аварии
- ELK-кластер требует отдельного дорогого админа и постоянного тюнинга
- Нет ретенции по политике — риск не пройти проверку и переполнить диски

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

3.x

Ветка Loki, которую мы ставим: single-binary + объектное хранилище, один бинарь Go

по докам grafana.com

v13

Версия схемы хранения — обязательна для structured metadata и OTLP-приёма

Loki 3.0 docs

tsdb

Тип индекса по умолчанию в 3.x; требование для v13 и нативного OTLP

по докам grafana.com

720h

retention_period = 30 суток в limits_config — наш базовый стандарт хранения

наш стандарт

~100 ГБ

Потолок single-node в сутки; выше — microservices mode на S3/MinIO

наш стандарт

2026-03

Promtail EOL — новые внедрения ставим на Grafana Alloy (OTel Collector)

по докам grafana.com

РЕАЛИЗАЦИЯ 1 ИЗ 3

Ядро: Loki single-binary + MinIO

Что настраиваем

Один узел Loki 3.x на ВМ 4-8 vCPU / 8-16 ГБ / NVMe, объектное хранилище MinIO/S3

Как мы это делаем

1. Разворачиваем grafana/loki:3.x в Docker Compose, порт 3100, auth_enabled: false за reverse-proxy с basic_auth
2. schema_config: store: tsdb, object_store: s3, schema: v13, index period 24h — иначе OTLP и метаданные не заведутся
3. limits_config: retention_period 720h, allow_structured_metadata: true, max_query_series, max_query_length 720h
4. compactor: retention_enabled: true, delete_request_store, working_directory — без этого старые чанки не удаляются
5. Grafana 11.x как data source Loki, URL http://loki:3100, панели Logs/Stat/Time series

РЕЗУЛЬТАТ

Единая точка приёма и поиска логов на одном узле держит парк до сотни ГБ/сутки. Ретенция срабатывает автоматически через compactor, диски не переполняются, история доступна за 30-90 дней по LogQL.

КЛЮЧЕВОЙ НЮАНС

В доках сверяем связку tsdb + schema v13 + allow_structured_metadata: без всех трёх нативный OTLP-эндпоинт и structured metadata молча не работают.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Сбор: Grafana Alloy вместо Promtail

Что настраиваем

Агент Alloy на Linux-узлах и Windows-хостах клиента, доставка в loki.write

Как мы это делаем

1. Ставим Alloy: local.file_match + loki.source.file на /var/log/*.log и /var/log/nginx/*.log
2. Windows Event Log собираем через loki.source.windowsevent (eventlog_name Application/System/Security)
3. Парсинг в loki.process: stage.regex + stage.labels для method/status, тяжёлые regex выносим осторожно
4. loki.write endpoint на https://loki.company.ru/loki/api/v1/push с basic_auth, TLS обязателен
5. Старые конфиги Promtail конвертируем штатным мигратором Alloy одной командой

РЕЗУЛЬТАТ

Единый агент для логов, метрик и трейсов на всех ОС парка. Отказ от EOL-Promtail снимает риск остаться без обновлений и совместимости с новыми версиями Loki.

КЛЮЧЕВОЙ НЮАНС

Смотрим в доке компоненты loki.source.* и bookmark_path для Windows Event Log — позиция чтения хранится в закладке, при её потере журнал перечитывается с нуля.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Запросы и алерты: LogQL + Grafana Alerting

Что настраиваем

Дашборды и правила оповещений на стороне Grafana, доставка алертов в Telegram

Как мы это делаем

1. Базовые фильтры LogQL: |= подстрока, |~ regex, != и !~ на исключение, | json на разбор полей
2. Метрики из логов: sum by (service) (count_over_time({env="prod"} |= "ERROR" [5m]))
3. Топ-проблемных URL: topk(5, sum by (path) (count_over_time({job="nginx"} |~ " 4\\d\\d " [1h])))
4. Grafana Alerting: expr на порог > 10 ошибок за 5m, for: 2m, severity: warning
5. Contact point на Telegram-бота, шаблон summary/annotations на русском

РЕЗУЛЬТАТ

Дежурный инженер видит всплеск ошибок API в Telegram через минуты, а не по звонку клиента. Разбор инцидента идёт по одному LogQL-запросу из панели вместо обхода узлов.

КЛЮЧЕВОЙ НЮАНС

LogQL близок к PromQL — порог входа для нового специалиста низкий, но агрегации count_over_time по широкому диапазону нагружают querier, ограничиваем лимитами запросов.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Высокая кардинальность лейблов**
  request_id или user_id в labels = миллионы стримов и деградация. Такие значения держим в теле строки или structured metadata, не в лейблах.
- ✕ **Compactor отключён**
  Без retention_enabled: true в блоке compactor ретенция не применяется, старые чанки не удаляются и забивают хранилище. Включаем всегда.
- ✕ **Старая схема или boltdb**
  На schema ниже v13 и не-tsdb индексе не работают structured metadata и нативный OTLP. Новые базы стартуем сразу на tsdb + v13.
- ✕ **Ставят EOL-Promtail**
  Promtail EOL с марта 2026, обновлений нет. Новые внедрения делаем на Alloy, существующие переводим штатным мигратором конфигов.
- ✕ **Нет лимитов запросов**
  Один жирный запрос кладёт querier. Задаём max_query_length, max_query_series, max_entries_limit_per_query в limits_config.
- ✕ **Потеряна позиция чтения агента**
  При потере positions.yaml/bookmark агент перечитывает журналы с нуля и дублирует логи. Каталог позиций кладём на постоянный том.
- ✕ **Тяжёлые regex в pipeline**
  Каждая строка проходит stage.regex; неоптимальное выражение тормозит агент. Упрощаем шаблон и парсим только нужные поля.
- ✕ **auth_enabled: false наружу**
  Push-эндпоинт без защиты принимает чужие логи. Прячем Loki за reverse-proxy с basic_auth и TLS, порт 3100 не публикуем.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Loki 3.x single-binary, filesystem-хранилище, схема tsdb + v13
- Grafana Alloy на источниках вместо Promtail
- retention_period в limits_config и compactor с retention_enabled
- Data source Loki в Grafana, базовые панели Logs

НОРМАЛЬНО

- Объектное хранилище MinIO/S3 вместо локальной ФС
- basic_auth + TLS на push-эндпоинте за reverse-proxy
- Grafana Alerting с доставкой алертов в Telegram
- Лимиты запросов: max_query_series, max_query_length

ХОРОШО

- Microservices mode (distributor/ingester/querier/compactor) от ~100 ГБ/сутки
- structured metadata + нативный OTLP-приём
- Разные ретенции по лейблам: security/app/debug
- Дисциплина кардинальности лейблов, ревью новых стримов

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Схема хранения tsdb + v13, а не boltdb/устаревшая версия?
- [ ] Включён compactor с retention_enabled: true, задан delete_request_store?
- [ ] Прописан retention_period под требования по хранению (напр. 720h)?
- [ ] Агент — Grafana Alloy, а не EOL-Promtail?
- [ ] Push-эндпоинт закрыт basic_auth и TLS, порт 3100 не торчит наружу?
- [ ] Заданы лимиты запросов max_query_series / max_query_length?
- [ ] Проверена кардинальность лейблов, нет request_id/user_id в labels?
- [ ] Каталог позиций/закладок агента на постоянном томе?
- [ ] Настроены алерты Grafana Alerting с доставкой в Telegram?
- [ ] Оценён суточный объём: single-node до ~100 ГБ или уже microservices?

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

ЧТО МЫ ДЕЛАЕМ

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).

- Проектируем архитектуру: single-binary или microservices под объём парка
- Ставим Loki 3.x + MinIO/S3, схема tsdb + v13, ретенция через compactor
- Разворачиваем Grafana Alloy на Linux и Windows, миграция с Promtail
- Пишем парсеры nginx/Docker/Event Log, дашборды и LogQL-запросы
- Настраиваем алерты в Grafana с доставкой в Telegram, обучаем команду

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

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Grafana Loki documentation — configuration (grafana.com — 3.x)

02 Storage schema (tsdb, v13) (grafana.com — 3.x)

03 Log retention & compactor (grafana.com — 3.x)

04 What is structured metadata / OTLP (grafana.com — 3.x)

05 Migrate from Promtail to Grafana Alloy (grafana.com — 2026)

06 loki.source.windowsevent (Alloy) (grafana.com — latest)

07 LogQL query language (grafana.com — 3.x)

08 Наш шаблон loki-config.yaml + Alloy (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.