Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

ELK Stack: централизованное логирование с нуля

Как мы разворачиваем Elastic Stack 9.x: сбор, ретенция, алертинг для офиса на 50 серверов

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Логи лежат на самих серверах: на 1С-сервере, контроллере домена, шлюзе. При инциденте инженер обходит машины руками, часть журналов уже перезаписана по кругу, сопоставить ошибку приложения с сетевым событием нечем. Разбор растягивается на дни, простой оплачивает бизнес, а требования к хранению журналов доступа закрыть нечем. Мы сводим всё в один контур на Elastic Stack.

Почему это важно бизнесу

- Простой 1С или почты оплачивается зарплатой всего офиса: сокращение разбора инцидента с часов до минут — это прямая экономия рабочего дня.
- Без единого журнала не доказать, кто и когда удалил файл или выгрузил базу: спорные ситуации с сотрудниками решаются на словах.
- Требования к хранению журналов доступа закрываются только централизованным архивом с понятной глубиной и защитой от подчистки на месте.
- Рост индексов забивает диски: на 95% узел ставит индексам read-only-allow-delete, и приём логов встаёт именно в момент аварии.

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

9.x

Актуальная ветка Elastic Stack (август 2026 — 9.5.x); 8.x — только под апгрейд легаси

релиз-ноты Elastic Stack

≤31 ГБ

Потолок JVM-heap на ноду: -Xms=-Xmx, 50% RAM хоста, но не выше ~31 ГБ ради compressed oops

JVM settings, elastic.co

50 ГБ

max_primary_shard_size в logs@lifecycle: держим шард 10–50 ГБ по размеру, а не по дате

ILM, политика logs@lifecycle

85/90/95%

Пороги disk watermark low/high/flood_stage: на 95% индексы узла — read-only-allow-delete

shard allocation, elastic.co

262144

Минимум vm.max_map_count для bootstrap check; рекомендованное значение — 1048576

bootstrap checks, elastic.co

32 симв.

Минимум для xpack.encryptedSavedObjects.encryptionKey, иначе алертинг Kibana блокируется

Kibana alerting, elastic.co

РЕАЛИЗАЦИЯ 1 ИЗ 3

Кластер Elasticsearch 9.x: три ноды master+data

Что настраиваем

3 ВМ по 8 vCPU / 32 ГБ RAM / SSD плюс отдельная ВМ Kibana; роли master+data совмещены на всех трёх узлах

Как мы это делаем

1. Пакеты из artifacts.elastic.co/packages/9.x/apt: автонастройка поднимает security и TLS. Под systemd пароль elastic не печатается — elasticsearch-reset-password.
2. Правим jvm.options.d: -Xms=-Xmx на 50% RAM и не выше ~31 ГБ; bootstrap.memory_lock: true, swap off, vm.max_map_count от 262144, лимит nofile 65535.
3. Три master-eligible узла ради кворума: discovery.seed_hosts на все три, cluster.initial_master_nodes прописываем только на первом бутстрапе и потом убираем.
4. index.number_of_replicas: 1 в шаблоне; следим за watermark 85/90/95% через GET _cat/allocation и GET _cluster/health.
5. Kibana только за nginx с TLS: подключаем по elasticsearch-create-enrollment-token -s kibana (30 минут), задаём server.publicBaseUrl и постоянный encryptionKey.

РЕЗУЛЬТАТ

Потеря одной ВМ не останавливает приём логов и поиск: кластер сохраняет кворум, реплики поднимают шарды на живых узлах. Kibana доступна по HTTPS и под учётками с ролями, а не голым портом 5601 в корпоративной сети.

КЛЮЧЕВОЙ НЮАНС

cluster.initial_master_nodes нужен ровно один раз: забытая строка при переносе ломает формирование кластера. Heap выше ~31 ГБ отключает compressed oops — полезной памяти становится меньше, а не больше.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Сбор источников: Elastic Agent, Filebeat, Winlogbeat

Что настраиваем

Linux-узлы (nginx, syslog, auditd), Windows DC и 1С-серверы, сетевое оборудование по syslog

Как мы это делаем

1. На новых узлах ставим Elastic Agent под Fleet: одна политика в Kibana вместо конфигов на каждой машине, апгрейд бинарей — тоже из Fleet.
2. Где остаются Beats — только filestream input: в 9.0 inputs log и container отключены, Filebeat не стартует; allow_deprecated_use: true — лишь на время миграции.
3. Берём штатные интеграции и filebeat modules enable nginx system auditd — готовые ingest-пайплайны и дашборды; свой grok пишем только под 1С-трассировку.
4. Windows: каналы Security, System, Application; в winlogbeat.event_logs на Security ставим ignore_older (например, 72h), чтобы не втянуть архив за месяцы.
5. Транспорт только TLS: ssl.certificate_authorities с нашим CA, секреты агентов — в keystore, а не открытым текстом в yml.

РЕЗУЛЬТАТ

Новый сервер подключается за 10–15 минут: политика Fleet или готовый yml — и данные ложатся в data streams logs-*-* с полями ECS. Поиск идёт из одного Discover по всем узлам, без обхода машин по RDP.

КЛЮЧЕВОЙ НЮАНС

Первый запуск сбора Security-журнала на контроллере домена способен залить кластер историей за месяцы: ограничиваем окно через ignore_older и подключаем агентов волнами, глядя на очередь индексации.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Ретенция, снапшоты и алертинг

Что настраиваем

ILM на data streams logs--*, SLM в репозиторий вне кластера, правила алертинга в Kibana*

Как мы это делаем

1. Идём от встроенной logs@lifecycle (rollover 50gb или 30 дней), но управляемую политику не правим: клонируем её под клиента и дописываем свои фазы.
2. Держим данные в data streams, а не в индексах по дате: rollover режет шард по размеру; помимо 50 ГБ работает неявный лимит 200 млн документов на шард.
3. Правки шаблонов кладём в component template logs@custom, чтобы обновление интеграций не затирало маппинги и настройки индекса.
4. SLM-политика по cron в файловый или S3-репозиторий с retention: кластер не является бэкапом сам себе, реплика не спасает от ошибочного delete.
5. Правила Kibana Alerting: всплеск 5xx, серия неудачных входов, остановка потока логов от узла. Часть коннекторов платная — канал подбираем под подписку.

РЕЗУЛЬТАТ

Объём хранения становится предсказуемым: старые данные уходят по расписанию, а не вычищаются вручную в аварийном режиме. Критичное событие доезжает до дежурного за минуту, а не всплывает через неделю при разборе.

КЛЮЧЕВОЙ НЮАНС

Cold и frozen через searchable snapshots — функция Enterprise-подписки. Без неё глубокое хранение планируем на SLM-снапшотах с восстановлением по запросу, а не по красивой схеме из презентации.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Нет ILM — индексы растут бесконтрольно**
  Без rollover и фазы delete на 95% узел уходит в read-only-allow-delete. Свою политику клонируем из logs@lifecycle до подключения агентов.

- ✕ **Агенты шлют логи без TLS**
  Незашифрованный транспорт — это имена учёток и токены из журналов в открытом виде в сети. Включаем ssl.certificate_authorities и проверяем цепочку.

- ✕ **Тысячи мелких шардов по 500 МБ**
  Индексы по дате плодят шарды и очередь pending tasks на мастере. Держим 10–50 ГБ на шард и rollover по max_primary_shard_size, а не по дате.

- ✕ **Kibana открыта без авторизации**
  Порт 5601 в LAN без TLS и ролей — это чтение Security-журналов домена любым сотрудником. Только HTTPS-прокси и разграничение ролей.

- ✕ **Heap задран выше 31 ГБ**
  JVM теряет compressed oops, паузы GC растут, полезной памяти меньше. В jvm.options.d ставим -Xms=-Xmx: 50% RAM хоста и потолок около 31 ГБ.

- ✕ **Забыт encryptionKey в kibana.yml**
  Без постоянного xpack.encryptedSavedObjects.encryptionKey от 32 символов ключ меняется на каждом рестарте, алертинг и коннекторы блокируются.

- ✕ **Конфиги с log input перед апгрейдом**
  В 9.0 inputs log и container выключены — Filebeat не поднимется. Переводим на filestream: там file_identity fingerprint, файл читается от 1024 Б.

- ✕ **Кластер считают резервной копией**
  Реплики спасают от падения ноды, но не от ошибочного удаления или кривой ILM-политики. Нужен snapshot-репозиторий вне кластера и SLM с retention.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Single-node Elasticsearch 9.x, Kibana за TLS-прокси, вход по паролю
- Агенты на критичных узлах: контроллер домена, 1С-сервер, шлюз, почта
- ILM с фазами hot и delete и явно согласованной глубиной хранения
- Алерт на watermark 85% и на состояние по GET _cluster/health

НОРМАЛЬНО

- Три ноды master+data, реплика 1, кворум из трёх master-eligible узлов
- Fleet-политики вместо ручных yml, TLS на всём транспорте до кластера
- Data streams logs-*-*, rollover 50 ГБ / 30 дней, правки в logs@custom
- SLM-снапшоты в отдельное хранилище и проверка восстановления

ХОРОШО

- Выделенные роли: master, hot- и warm-ноды, отдельный Fleet Server
- Logstash с queue.type: persisted как буфер перед кластером
- Дашборд и набор правил алертинга на каждый значимый сервис
- Ежеквартальный аудит объёма по индексам и пересмотр ретенции

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Есть ли перечень источников логов и владелец каждого: 1С, контроллер домена, шлюз, nginx?
- [ ] Заданы ли ILM-политики с фазой delete на все data streams и клонированы ли встроенные?
- [ ] Включён ли TLS между агентами и кластером, а секреты лежат в keystore, а не в yml?
- [ ] Закрыта ли Kibana TLS-прокси и разграничены ли роли на чтение Security-журналов?
- [ ] Настроен ли -Xms=-Xmx в пределах 50% RAM хоста и не выше ~31 ГБ на каждой ноде?
- [ ] Проверены ли лимиты: vm.max_map_count от 262144, nofile 65535, swap выключен?
- [ ] Есть ли snapshot-репозиторий вне кластера и SLM-политика с retention?
- [ ] Восстанавливали ли снапшот на тестовом узле хотя бы раз за последние полгода?
- [ ] Заведены ли алерты на watermark диска, состояние кластера и остановку потока логов?
- [ ] Переведены ли конфиги Filebeat с log/container на filestream до апгрейда на 9.x?

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).

- Аудит источников логов и расчёт объёма: сколько ГБ в сутки, какая ретенция, сколько дисков под неё
- Развёртывание Elastic Stack 9.x: кластер, Kibana за TLS, Fleet, роли и учётные записи
- Подключение агентов на Linux и Windows, парсеры под 1С, nginx, IIS и сетевое оборудование
- ILM и SLM: политики ретенции, снапшоты вне кластера, проверка восстановления
- Дашборды и правила алертинга под ваши сервисы с передачей инструкции дежурному

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 Size your shards — размер шарда и число шардов (elastic.co — 9.x)

02 JVM settings — размер heap и файлы jvm.options.d (elastic.co — 9.x)

03 ILM: политика logs@lifecycle и клонирование встроенных (elastic.co — 9.x)

04 Important system configuration и bootstrap checks (elastic.co — 9.x)

05 Automatic security setup: TLS и enrollment token (elastic.co — 9.x)

06 Migrate log or container input configurations to filestream (elastic.co — 9.x)

07 Alerting and action settings in Kibana (elastic.co — 9.x)

08 Наш шаблон внедрения ELK для инфраструктуры на 30–60 узлов (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.