Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Elasticsearch-кластер в продакшне

Наша методология: роли нод, heap, шардинг, ILM, снапшоты и мониторинг ES 8.19/9.x

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Логи, метрики и поиск переезжают в Elasticsearch — и кластер быстро становится единой точкой отказа. Одна нода с ролями master+data теряет кворум под нагрузкой, диск уходит за flood-stage 95% и индексы встают в read-only, heap за 31 ГБ убивает compressed oops, а снапшотов нет вовсе. Мы проектируем ES так, чтобы отказ ноды был рутиной, а не ночным простоем бизнеса.

Почему это важно бизнесу

- Красный статус кластера = остановка поиска и приёма логов: слепая техподдержка, сорванные SLA и разбор инцидентов вслепую.
- Флуд-ватермарк 95% переводит все индексы ноды в read-only — приём данных встаёт молча, без падения сервиса и без алерта.
- Потеря data-ноды при number_of_replicas: 0 — это безвозвратная потеря индексов, восстановить их неоткуда.
- Незакрытый доступ к ES = утечка: в индексах логов лежат ПДн клиентов и токены, это штрафы по 152-ФЗ и репутация.
- Раздутый шардинг съедает heap мастеров: кластер тормозит на ровном месте, а лечится только простоем на переиндексацию.

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

9.5 / 8.19

Актуальная ветка ES и последняя минорная 8.x — на них мы ставим новые кластеры

по докам Elasticsearch, release notes 9.5

85/90/95%

Disk watermark low/high/flood_stage — свои пороги алертов ставим ниже, на 80%

по докам Elasticsearch, disk-based allocation

≤31 ГБ

Потолок heap: -Xms=-Xmx, не более 50% ОЗУ и не выше границы compressed oops

по докам Elasticsearch, JVM settings

10–50 ГБ

Целевой размер шарда; в rollover задаём max_primary_shard_size 50gb, min 10gb

по докам Elasticsearch, size your shards

1000

cluster.max_shards_per_node на дата-ноду; столько же — total_fields.limit

по докам Elasticsearch, cluster settings

40 МБ/с

Дефолт indices.recovery.max_bytes_per_sec; на cold/frozen — от ОЗУ ноды

по докам Elasticsearch, recovery settings

РЕАЛИЗАЦИЯ 1 ИЗ 3

Топология кластера: роли нод и кворум мастеров

Что настраиваем

3 master-only + слои data_hot/data_warm/data_cold + coordinating-only за балансировщиком; на каждой ноде свой node.roles

Как мы это делаем

1. Разносим роли: node.roles: [master] на трёх выделенных нодах, [data_hot, data_content, ingest] на горячих, [data_cold] на архивных
2. Балансировщик заводим как coordinating-only: node.roles: [ ] — пустой массив, узел только маршрутизирует запросы и не хранит шарды
3. Единый список discovery.seed_hosts на всех нодах; cluster.initial_master_nodes заполняем только при первом старте и удаляем после формирования кластера
4. Проверяем bootstrap checks: production mode включается при заданном network.host — нода не стартует с невыполненными требованиями
5. path.data и path.logs выносим за пределы $ES_HOME, чтобы обновление пакета не задевало данные и логи

РЕЗУЛЬТАТ

Потеря любой одной ноды больше не роняет кластер: кворум из трёх мастеров переживает отказ, тяжёлые агрегации не выбивают master-процесс, а горячая запись изолирована от архивного чтения. Ребут ноды на обновление становится плановой операцией без окна простоя.

КЛЮЧЕВОЙ НЮАНС

В документации смотрим раздел node settings: ролей десяток, а data — это агрегат. Смешанная роль master+data допустима только на малых кластерах без тяжёлых агрегаций.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Память, диски и жизненный цикл данных (ILM)

Что настраиваем

JVM heap и системные лимиты на всех нодах + ILM-политика hot/warm/cold/delete для потоков данных (data streams)

Как мы это делаем

1. Heap задаём файлом в /etc/elasticsearch/jvm.options.d/*.options: -Xms равен -Xmx, не более 50% ОЗУ и не выше границы compressed oops
2. bootstrap.memory_lock: true плюс LimitMEMLOCK=infinity в systemd-override; проверяем mlockall: true в выводе _nodes
3. Системное: vm.max_map_count 1048576 (для 8.16+, ранее 262144), лимит открытых файлов 65535, минимум 4096 потоков процессу, swap выключен
4. ILM: в hot — rollover с max_primary_shard_size: 50gb; в warm — shrink и forcemerge; в cold — searchable_snapshot; в delete — delete по min_age
5. Пишем в data stream, а не в индекс с алиасом: роловер и переходы фаз ILM идут по backing-индексам автоматически

РЕЗУЛЬТАТ

Ноды перестают свопиться и ловить long GC, размер шарда держится в коридоре 10–50 ГБ, а старые данные сами уезжают на дешёвые диски и удаляются по ретенции. Объём горячего слоя перестаёт расти линейно, и закупка NVMe планируется на год вперёд.

КЛЮЧЕВОЙ НЮАНС

searchable_snapshot не ставим одновременно в hot и cold — по документации индекс тогда не мигрирует в cold-слой. И heap выше 31 ГБ не решает проблему: помогают только новые ноды.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Безопасность, снапшоты и мониторинг

Что настраиваем

TLS на transport и http, SLM-политика на fs- или s3-репозиторий, метрики кластера в Prometheus/Grafana с алертами

Как мы это делаем

1. Оставляем xpack.security.enabled: true (с 8.0 включается автоматически) и TLS на transport и http; сертификаты выпускаем elasticsearch-certutil
2. Репозиторий fs монтируем в одну и ту же точку на всех master- и data-нодах и прописываем path.repo; альтернатива — репозиторий типа s3
3. SLM-политика: schedule в cron, repository, config.indices, retention с expire_after, min_count, max_count; копировать path.data нельзя
4. Мониторим _cluster/health и _health_report (indicators: master_is_stable, shards_availability, disk, repository_integrity), verbose=false при частом опросе
5. Пороги алертов: status red, unassigned_shards > 0 дольше 10 минут, heap > 85%, диск > 80%, рост очередей write и search

РЕЗУЛЬТАТ

Кластер перестаёт быть чёрным ящиком: деградация видна за часы до отказа, а не в момент, когда индексы уже read-only. Снапшоты снимаются по расписанию и чистятся по ретенции, а восстановление проверяется на отдельном кластере, а не в момент аварии.

КЛЮЧЕВОЙ НЮАНС

Парент-брейкер по умолчанию бьёт на 95% JVM heap по реальному потреблению (indices.breaker.total.use_real_memory: true) — если он срабатывает, дело не в лимите, а в размере запросов и шардинге.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Две master-eligible ноды вместо трёх**
  Кворум = 2 из 2: падение любой ноды останавливает выборы мастера. Держим нечётное число master-eligible нод, минимум три.

- ✕ **Heap больше 31 ГБ**
  JVM теряет compressed oops, указатели раздуваются, GC растёт. Ставим -Xms=-Xmx ≤ 50% ОЗУ и ниже порога oops, масштабируемся нодами.

- ✕ **Мониторинг только по green/yellow/red**
  Статус зелёный, а write-очередь растёт и диск на 84%. Добавляем thread_pool.write.queue/rejected, heap_percent, disk.percent и _health_report.

- ✕ **Динамический mapping без ограничений**
  Каждое новое поле в JSON создаёт mapping-запись; при лимите 1000 полей индексация встаёт. Фиксируем шаблон и dynamic: strict на логах.

- ✕ **Бэкап копированием path.data**
  Файлы Lucene меняются на лету — получаем битый индекс. Только snapshot API и SLM; репозиторий fs монтируем на всех master- и data-нодах.

- ✕ **Реплики отключены «для экономии»**
  number_of_replicas: 0 превращает отказ data-ноды в потерю данных. В проде минимум 1 реплика, для критичных индексов — 2.

- ✕ **Тысячи мелких шардов**
  Каждый шард — это память мастера и файловые дескрипторы. Держим 10–50 ГБ на шард и следим за cluster.max_shards_per_node (1000 на дата-ноду).

- ✕ **Отключённый xpack.security**
  Без TLS и аутентификации индексы с ПДн доступны всем в сети. С 8.0 security включается сама — не выключаем, а настраиваем роли и API-ключи.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- 3 master-eligible ноды, number_of_replicas ≥ 1, swap выключен
- Heap = 50% ОЗУ, но не выше 31 ГБ; bootstrap.memory_lock: true
- xpack.security и TLS на transport включены, пароли встроенных учёток сменены
- Ручной snapshot в репозиторий fs и алерт на статус red

НОРМАЛЬНО

- Выделенные master-only ноды и слои data_hot/data_warm
- ILM с rollover по max_primary_shard_size 50gb и ретенцией
- SLM-политика с retention: expire_after, min_count, max_count
- Prometheus + exporter + Grafana, алерты по heap, диску и шардам

ХОРОШО

- Data streams, index templates и dynamic: strict вместо авто-mapping
- Слой data_cold/data_frozen на searchable snapshots в s3-репозитории
- Квартальный тест восстановления снапшота на отдельный кластер
- Опрос _health_report с verbose=false и алерты по индикаторам

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Число master-eligible нод нечётное (≥3) и они не совмещены с тяжёлой data-нагрузкой?
- [ ] Heap задан через jvm.options.d, -Xms равен -Xmx и не выше 50% ОЗУ и 31 ГБ?
- [ ] bootstrap.memory_lock: true, mlockall подтверждён в _nodes, swap выключен в /etc/fstab?
- [ ] vm.max_map_count и лимит открытых файлов 65535 подняты на всех нодах?
- [ ] У всех продовых индексов number_of_replicas ≥ 1, а размер primary-шарда в коридоре 10–50 ГБ?
- [ ] ILM привязана к data stream: rollover по max_primary_shard_size 50gb и min_primary_shard_size 10gb?
- [ ] SLM-политика активна, репозиторий зарегистрирован, ретенция снапшотов настроена?
- [ ] Восстановление из снапшота реально проверялось за последний квартал?
- [ ] Есть алерты на status red, unassigned_shards, heap выше 85% и диск выше 80%?
- [ ] TLS включён на transport и http, анонимный доступ к 9200 закрыт, роли выданы по минимуму?

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 топологию под ваш объём: роли нод, слои hot/warm/cold, sizing heap и дисков
- Разворачиваем кластер с TLS, ролями и API-ключами; настраиваем index templates и ILM
- Ставим SLM в fs- или s3-репозиторий и раз в квартал проверяем восстановление на стенде
- Подключаем мониторинг: exporter, дашборды Grafana, алерты по heap, шардам и watermark
- Аудит действующего кластера: шардинг, mapping, JVM, снапшоты — с планом исправлений

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 (elastic.co — 9.x)

02 Node settings: node.roles (elastic.co — 9.x)

03 Advanced configuration: setting the heap size (elastic.co — 9.x)

04 Cluster-level shard allocation and routing settings (elastic.co — 9.x)

05 Index lifecycle management (ILM) (elastic.co — 9.x)

06 Snapshot and restore, SLM (elastic.co — 9.x)

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

08 Шаблон ITfresh: чек-лист приёмки ES-кластера (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.