Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

100 баллов PageSpeed: nginx в Docker, brotli и TLS за день

Наш референсный стек: nginx 1.30 stable + ngx_brotli, HTTP/2 и HTTP/3, ACME с ARI

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Сайт на shared-хостинге отдаёт первый байт за секунды: общий пул PHP-FPM, веб-сервер без HTTP/2, выключенное сжатие, цепочка редиректов http→https→www. Бизнес теряет не эстетику, а заявки: посетитель уходит до отрисовки первого экрана. Лечим переносом на выделенную VM и контейнерным nginx с версионированным конфигом.

Почему это важно бизнесу

- Первый байт за 2–3 секунды поднимает отказы на мобильных: вкладку закрывают раньше, чем отрисуется первый экран
- Каждое звено редиректа — лишний RTT, а на https ещё и TLS-хендшейк: на мобильной сети это сотни миллисекунд
- Shared-хостинг — общий пул CPU и памяти: соседний сайт под нагрузкой тормозит ваш, а повлиять на это нельзя
- Просроченный сертификат = страница-предупреждение в браузере и остановленный поток заявок на весь рабочий день
- Конфиг, живущий только на проде и в голове администратора, невозможно быстро воспроизвести и откатить после сбоя

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

1.30.x

Ветка nginx stable в нашем образе: HTTP/2, модуль QUIC/HTTP/3, накопленные фиксы безопасности

базовый образ стека

5–6 из 11

brotli_comp_level в рантайме при дефолте 6; уровень 11 — только при пресжатии .br на сборке

профиль сжатия

2.5 с / 0.1

Пороги LCP и CLS «good» по p75 реальных пользователей — наша цель вместо круглой сотни в отчёте

целевые метрики загрузки

200 мс

Порог INP «good» по p75: столько даём на отклик интерфейса, выше — оценка уходит в «плохо»

целевые метрики отклика

30 %

Вес TBT в Performance-скоре Lighthouse: чинится в коде страницы, а не на стороне сервера

структура лабораторного скора

200 дней

Предельный срок TLS-сертификата с 15.03.2026, с 15.03.2027 — 100 дней: ручное продление отпало

срок жизни сертификата

РЕАЛИЗАЦИЯ 1 ИЗ 3

Свой образ nginx: brotli, HTTP/2 и опционально HTTP/3

Что настраиваем

Сайт-визитка клиента: VM 2 vCPU / 4 GB, Ubuntu 24.04 LTS, Docker Engine 29.x, весь стек в /opt/sites/<домен>

Как мы это делаем

1. Multi-stage Dockerfile: builder собирает ngx_brotli ключами configure базового образа плюс --with-compat --add-dynamic-module; в финал копируем только *.so
2. Финальный слой — nginx:1.30-alpine со STOPSIGNAL SIGQUIT; строки load_module ngx_http_brotli_filter/static стоят в главном контексте, до блока http {}
3. Воркеры уже работают под пользователем nginx; полный non-root — это образ nginx-unprivileged с listen 8080/8443 и pid/temp-путями в /tmp
4. brotli on, brotli_comp_level 5, brotli_static on; gzip оставляем фолбэком для клиентов без Accept-Encoding: br
5. listen 443 ssl плюс отдельная директива http2 on (с nginx 1.25.1); где нужен HTTP/3 — listen 443 quic reuseport и заголовок Alt-Svc с h3

РЕЗУЛЬТАТ

Распакованный образ около 50 МБ раздаёт статику в brotli и gzip, стартует за секунды и одинаково ведёт себя на dev и в проде. Обновление nginx или откат на предыдущую сборку — это смена тега в compose, без раскопок в системных каталогах и без риска потерять конфиг.

КЛЮЧЕВОЙ НЮАНС

Динамический модуль обязан быть собран той же версией nginx и теми же ключами configure, что и базовый образ, обязательно с --with-compat. Иначе nginx падает на load_module ещё до чтения конфига.

РЕАЛИЗАЦИЯ 2 ИЗ 3

TLS и автопродление ACME без даунтайма

Что настраиваем

Контейнер ACME-клиента рядом с nginx: общий том /etc/letsencrypt, webroot /var/www/certbot, единая сеть compose

Как мы это делаем

1. Разовый выпуск через webroot; в 80-м server-блоке живёт только location ^~ /.well-known/acme-challenge/ и один 301 на канонический хост
2. Продление — certbot renew с --deploy-hook, который шлёт SIGHUP в контейнер nginx: воркеры перечитывают сертификат без рестарта и обрыва соединений
3. Certbot 4.1+ учитывает ARI (RFC 9773): момент продления назначает сам УЦ, поэтому дефолтный порог «за 30 дней до истечения» больше ничего не решает
4. ssl_stapling и ssl_stapling_verify выключены: OCSP-ответчики Let's Encrypt отключены 06.08.2025, а URL OCSP убран из сертификатов ещё 07.05.2025
5. Профиль TLS: ssl_protocols TLSv1.2 TLSv1.3, ssl_session_cache shared:SSL:50m, ssl_session_tickets off, HSTS max-age=31536000 always

РЕЗУЛЬТАТ

Сертификат продлевается сам, nginx подхватывает его по SIGHUP — соединения не рвутся, простоя нет. Стек заранее готов к сокращению сроков: момент продления берётся от УЦ через ARI, а не из захардкоженного порога в самописном скрипте.

КЛЮЧЕВОЙ НЮАНС

Окно продления назначает УЦ через ARI, а не наш cron. Контролируем не «сколько дней осталось», а факт сдвига NotAfter после renew — иначе тихо сломавшийся хук всплывёт в день истечения.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Кэш-заголовки, логи и регламент после сдачи

Что настраиваем

Слой раздачи статики и эксплуатация на той же VM: 5–10 клиентских сайтов отдельными контейнерами в своих compose-проектах

Как мы это делаем

1. Хешированная статика (css/js/woff2/webp/avif) — expires 1y и Cache-Control public, immutable; HTML — no-cache, чтобы правки контента доезжали сразу
2. Ассеты пережимаем заранее в .br и .gz на этапе сборки, brotli_static и gzip_static отдают готовый файл — CPU в рантайме не тратится
3. Security-заголовки держим в отдельном include и подключаем в каждый location: любой add_header в дочернем блоке отменяет наследование родительских
4. Логи контейнеров ограничены драйвером json-file с max-size и max-file, иначе один болтливый контейнер выедает диск VM
5. healthcheck в compose на /healthz с interval 30s плюс внешняя проба раз в 60 с и алерт после двух подряд неудач

РЕЗУЛЬТАТ

TTFB держится в районе сотни миллисекунд, повторные заходы почти целиком закрываются кэшем браузера. Диск не забивается логами, а деградация скорости после заливки контента ловится нашим квартальным прогоном, а не жалобой маркетолога клиента.

КЛЮЧЕВОЙ НЮАНС

TBT, LCP и CLS дают 80 % Performance-скора (30/25/25). Сервер закрывает LCP и TTFB, а TBT лечится только в коде страницы — это мы честно проговариваем с клиентом на старте работ.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **ssl_stapling on «по привычке»**
  OCSP-ответчики Let's Encrypt отключены 06.08.2025, URL OCSP из сертификатов убран: директива только пишет warning. Убираем её и ssl_stapling_verify.

- ✕ **load_module внутри блока http**
  Директива допустима только в главном контексте и до http {}, иначе nginx не стартует вовсе. Строки load_module ставим первыми в nginx.conf.

- ✕ **Тег latest в compose-файле**
  nginx:latest и certbot:latest ломают воспроизводимость и откат: вчерашняя сборка уже не восстановится. Фиксируем версии и обновляем осознанно.

- ✕ **gzip_comp_level 9 «чтобы сильнее»**
  Выигрыш против уровня 6 — единицы процентов, а CPU растёт кратно на каждый запрос. Максимум ставим только при пресжатии файлов на сборке.

- ✕ **add_header во вложенном location**
  Свой add_header в дочернем блоке отменяет наследование всех родительских заголовков. Подключаем общий include с ними в каждый такой location.

- ✕ **Цепочка http → https → www**
  Каждое звено — лишний RTT, а на https ещё и TLS-хендшейк в TTFB. Схлопываем в один 301 сразу на канонический хост с целевой схемой.

- ✕ **Сертификаты внутри образа**
  При пересборке и откате они пропадают, а renew пишет в слой, живущий до перезапуска. /etc/letsencrypt только томом, образ остаётся stateless.

- ✕ **Гонка за 100/100 любой ценой**
  Отключение аналитики и чата поднимает скор и роняет конверсию. Целимся в пороги CWV по p75, а не в круглую цифру лабораторного отчёта.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- TLS 1.2/1.3, один 301 на канонический хост, HSTS max-age=31536000 always
- gzip on для text/*, application/javascript, application/json, image/svg+xml
- Автопродление ACME с deploy-hook и регулярным renew --dry-run по расписанию
- ufw: наружу только 22/80/443, включены unattended-upgrades

НОРМАЛЬНО

- Свой образ nginx 1.30 stable с ngx_brotli, тег зафиксирован, откат сменой тега
- Cache-Control immutable на хешированную статику, no-cache на HTML
- Логи контейнеров через json-file с max-size/max-file, healthcheck в compose
- Внешняя проба доступности раз в 60 с и алерт после двух неудач подряд

ХОРОШО

- HTTP/3: listen 443 quic reuseport, http3 on и заголовок Alt-Svc с h3
- Пресжатие ассетов на сборке в .br/.gz плюс brotli_static и gzip_static
- ACME-клиент с поддержкой ARI — готовность к 100-дневным сертификатам
- Квартальный Lighthouse CI с бюджетом по TBT, LCP, CLS и алертом на регресс

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Зафиксирован ли тег образа nginx в compose и проверялся ли откат на предыдущую сборку?
- [ ] Собран ли ngx_brotli той же версией nginx и теми же ключами configure, что и базовый образ?
- [ ] Стоят ли строки load_module в главном контексте nginx.conf, до блока http?
- [ ] Остался ли ровно один 301-редирект от http до канонического https-хоста?
- [ ] Выключен ли ssl_stapling после отключения OCSP-ответчиков Let's Encrypt?
- [ ] Учитывает ли наш ACME-клиент ARI и переживёт ли он переход на 100-дневные сертификаты?
- [ ] Контролируется ли фактический сдвиг NotAfter, а не только наличие задачи продления?
- [ ] Лежат ли /etc/letsencrypt и контент сайта в томах, а не внутри образа?
- [ ] Ограничены ли логи контейнеров по размеру и числу файлов (max-size, max-file)?
- [ ] Известен ли p75 по LCP, INP и CLS у реальных пользователей, а не только лабораторный скор?

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).

- Собираем и ведём свой образ nginx с brotli, HTTP/2 и HTTP/3, храним в реестре с фиксированными тегами
- Переносим сайт на выделенную VM: Docker Engine, фаервол, доступ по ключам, автообновления безопасности
- Настраиваем выпуск и автопродление ACME с deploy-hook и контролем фактического сдвига NotAfter
- Готовим эталонный nginx.conf: сжатие, кэш-заголовки, TLS-профиль и security-заголовки
- Ставим мониторинг доступности и сертификата, квартальный прогон Lighthouse с бюджетом метрик

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 ngx_http_v3_module: QUIC и HTTP/3, listen quic, Alt-Svc (nginx.org — 1.30.x)

02 ngx_http_ssl_module: ssl_protocols, ssl_session_cache, ssl_stapling (nginx.org — 1.30.x)

03 ngx_http_v2_module: директива http2 on вместо параметра listen (nginx.org — 1.25.1+)

04 ngx_brotli: сборка модуля, brotli_comp_level, brotli_static (github.com/google/ngx_brotli — v1.0.0rc)

05 Docker Engine: logging drivers, healthcheck, Compose Specification (docs.docker.com — 29.x)

06 Let's Encrypt: завершение работы OCSP-сервиса и переход на CRL (letsencrypt.org — 2025)

07 Certbot: ARI (RFC 9773), deploy-hook, renew --dry-run (eff-certbot.readthedocs.io — 4.1+)

08 RFC 9773: ACME Renewal Information (ARI) Extension (rfc-editor.org — 2025)

09 Core Web Vitals: пороги LCP, INP и CLS по p75 (web.dev — 2026)

10 Lighthouse: веса метрик Performance-скора (developer.chrome.com — v10+)

11 TLS Baseline Requirements: график сокращения срока сертификата (cabforum.org — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.