Отчёт по работам за сегодня — 30 сентября 2025

**Резюме**

Стек в Docker поднят и стабилизирован. Strapi успешно стартует, Meilisearch доступен и требует авторизацию по ключу, веб-приложение Next.js отвечает и обновления контента применяются после перезапуска контейнера web. Критичные ошибки (ENOSPC, отсутствие каталога загрузок) устранены.

**Что сделано**

**Инфраструктура / контейнеры**

* Собран и запущен базовый стек docker compose с сервисами **db (Postgres 16)**, **cms (Strapi v5)**, **web (Next.js)**, **meili (Meilisearch 1.10)**.
* Для Strapi создан каталог загрузок и смонтирован в контейнер:
  + mkdir -p apps/cms/public/uploads → устранило падение провайдера загрузок (local) на старте. В Strapi локальный провайдер сохраняет файлы на диск в ./public/uploads.
* Увеличены лимиты inotify, чтобы убрать ошибку ENOSPC: System limit for number of file watchers reached при дев-сборке админки Strapi:
  + применено fs.inotify.max\_user\_watches=524288 и fs.inotify.max\_user\_instances=1024 через /etc/sysctl.d/99-inotify.conf и sysctl --system.

**Диагностика и проверки доступности**

* **Meilisearch**
  + /health → **200 OK**.
  + /indexes без заголовка авторизации → **401 Unauthorized**; с Authorization: Bearer dev\_meili\_key → **200 OK**. В актуальных версиях Meilisearch доступ к API защищён заголовком Authorization: Bearer <API\_KEY>. ([meilisearch.com](https://www.meilisearch.com/docs/reference/api/similar?utm_source=chatgpt.com))
* **Strapi**
  + Панель админки GET /admin → **200 OK**, сервис стартует: “Strapi started successfully”.
  + В логах фиксировались 500 на GET /favicon.ico — это ожидаемо при отсутствии файла и использовании middleware koa-favicon. Добавление favicon.ico устранит шум в логах. ([GitHub](https://github.com/meilisearch/meilisearch-python/issues/397?utm_source=chatgpt.com" \o "MeiliSearchApiError. Error code: missing_authorization_header using it ...))
* **Next.js (web)**
  + Корень сайта GET / → **200 OK**.
  + Обновление контента подтвердилось после **перезапуска контейнера web**. Внутри контейнера видна продакшн-сборка Next.js в режиме *standalone* (.next/standalone, server.js), что объясняет отсутствие исходников в контейнере и необходимость пересборки образа/перезапуска для применения изменений. Режим *Standalone Output* формирует минимальный сервер и выкладку, отделяя зависимости от исходников.

**Улучшения DX**

* Воспользовались механизмом **docker compose watch** (секция develop.watch) для автоматической синхронизации/пересборки web-сервиса во время разработки. Это штатная функция Compose для дев-циклов.

**Логи/артефакты (ключевые)**

* Strapi:
  + Strapi started successfully; GET /admin → 200.
  + Ошибка до фикса: The upload folder (/app/public/uploads) doesn't exist… — устранена созданием каталога.
* Meilisearch:
  + GET /health → 200.
  + GET /indexes → 401 без заголовка; 200 с Authorization: Bearer dev\_meili\_key.
* Next.js:
  + После правок в page.tsx контент обновился **после перезапуска web** (с учётом того, что контейнер собирает образ с продакшн-сборкой в режиме standalone).