

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ЭКСПЛУАТАЦИИ ПРОГРАММНОГО КОМПЛЕКСА FusionAI

Правообладатель: ООО «ФБЮЖНАИ» (ИНН 6670527516, ОГРН 1246600061388).

Контакты технической поддержки: e-mail fusionai@yandex.com; телефон +7 (962) 311-50-85; форма обратной связи на сайте fusion-ai.ru.

Дата составления: 12.08.2026. Версия документа: 1.0.

Назначение программного комплекса (далее — ПК): автоматизированный мониторинг здоровья и учёт крупного рогатого скота на молочных фермах по данным видеонаблюдения — бесконтактная идентификация животных, оценка хромоты, упитанности и веса, ведение электронного стада, уведомления и отчётность.

Документ адресован двум категориям читателей: техническому специалисту (разделы 1–4, 7–8 — установка и обслуживание) и оператору фермы — зоотехнику, ветеринарному специалисту (разделы 5–6 — работа в личном кабинете).

1. Состав программно-аппаратного комплекса

ПК функционирует в составе программно-аппаратного комплекса (ПАК) из четырёх уровней:

№	Уровень	Состав	Размещение
1	Камеры	IP-камеры (RTSP) или USB-камеры (v4l2), установленные над проходами и в зонах наблюдения животных	Ферма
2	Вычислительная плата	Одноплатный компьютер Radxa (SoC Rockchip, ARM64, Linux) с ПО уровня фермы: захват видео, детекция и трекинг животных, формирование клипов проходов, трансляции, телеметрия	Ферма
3	Сервер	Микросервисная платформа: приём и обработка данных, идентификация животных, оценка хромоты/веса, базы данных, личный кабинет, отчёты, уведомления	Серверная площадка правообладателя (РФ)
4	Клиент	Веб-браузер оператора (личный кабинет), электронная почта (отчёты),	Рабочее место оператора

№	Уровень	Состав	Размещение
		Telegram (уведомления/заявки)	

Потоки данных: камера → плата (детекция, трекинг, клипы) → сервер (идентификация, оценки, БД) → личный кабинет / отчёты / уведомления. Управление платами с сервера — по обратным SSH-туннелям.

Состав серверного ПО (устанавливается по настоящей инструкции):

Группа	Компоненты
Прикладные сервисы (Python 3.11, FastAPI)	back_service (основной API), report_service (отчёты), monitor_service (мониторинг плат), rock_receiver_service (приём данных с плат), subscription_service (биллинг), economy_service (экономические расчёты)
Сервисы идентификации и ML	herd_manager (идентификация, история здоровья), embed_service (визуальные эмбединги), lameness_classification_service (классификация хромоты), cow_weight (оценка веса), NVIDIA Triton Inference Server (инференс нейросетевых моделей)
Клиентские приложения	fusionai-frontend (личный кабинет и сайт), cow_labeler (инструмент разметки данных)
Хранилища и инфраструктура	MongoDB 8, PostgreSQL 16 + pgvector, Milvus, RabbitMQ 3.13, MinIO (S3), Nginx, Docker / Docker Compose

2. Требования к оборудованию и программному обеспечению

Уровень	Требования
Сервер	Архитектура x86-64; ОС Linux; Docker и Docker Compose (v2 и выше); GPU NVIDIA с поддержкой CUDA 12 (инференс моделей); накопители: SSD/NVMe для СУБД, HDD для видеоархива
Платы ферм	Radxa (Rockchip, ARM64); ОС Linux; Docker; systemd; локальная сеть с камерами (RTSP или USB); выход в интернет для связи с сервером
Камеры	IP-камеры с RTSP-поток (проверено на Hikvision) либо USB-камеры (v4l2)
Клиенты	Современный веб-браузер (Chrome, Firefox, Safari, Яндекс.Браузер актуальных версий); установка клиентского ПО не требуется

Ориентировочные ресурсы сервера: процессор x86-64, 16 ядер / 32 потока (AMD Ryzen 9 7950X), 32 ГБ ОЗУ, NVMe SSD 1 ТБ под базы данных, 3 × HDD по 1 ТБ под видеоархив, видеокарта NVIDIA GeForce RTX 4060 (8 ГБ), ОС Ubuntu 22.04 LTS, Docker 29.1 / Docker Compose v5. Для СУБД PostgreSQL рекомендованы параметры из

поставляемой конфигурации (shared_buffers 2 ГБ, effective_cache_size 12 ГБ — корректируются под объём ОЗУ).

3. Установка серверной части

3.1. Получение дистрибутива

Дистрибутив передаётся правообладателем в виде Git-репозитория с исходным кодом и файлами сборки (Dockerfile, docker-compose.yml). Доступ предоставляется по запросу: приватные репозитории организации Fusion-AI на github.com (git@github.com:Fusion-AI/fusionai-backend.git и другие репозитории из перечня ниже); доступ выдаётся по обращению на fusionai@yandex.com, по согласованию дистрибутив передаётся архивом исходного кода. Docker-образы собираются на целевом сервере из исходного кода штатными средствами Docker; внешний registry не требуется.

Состав репозитория серверной части:

Репозиторий	Назначение
fusionai-backend	Шесть прикладных микросервисов (порты 9000–9500) + RabbitMQ
fusionai-frontend	Личный кабинет и публичный сайт (Next.js)
herd_manager	Идентификация и история здоровья + PostgreSQL/pgvector + контейнер миграций
embed_service, ident_service, lameness_classification_service, cow_weight	ML-сервисы (порты 8080–8083)
s3_service	Объектное хранилище MinIO (бакет media)
cow_labeler_frame	Инструмент разметки (API + веб-интерфейс)
docker_inspection_service	Панель наблюдения за контейнерами (Portainer)

3.2. Подготовка окружения

1. Установить Docker и Docker Compose (v2 и выше).
2. Создать общую docker-сеть комплекса (используется всеми сервисами как external):
...
docker network create fusionai-net
...
3. Развернуть СУБД MongoDB 8 (устанавливается на хост либо отдельным контейнером; сервисы подключаются по строке `MONGO\_URI`).
4. Установить и настроить Nginx как обратный прокси с TLS-сертификатами (Let's Encrypt или иные) на домены личного кабинета и API. Все порты сервисов публикуются только на 127.0.0.1 и доступны извне исключительно через Nginx.

3.3. Настройка переменных окружения

В каждом репозитории имеется файл `.env.example` — шаблон настроек. Перед запуском скопировать его в `.env` и заполнить. Реальные значения секретов в систему контроля версий не помещать.

Категории настроек по компонентам:

Компонент	Категории настроек (.env)
fusionai-backend	Секрет подписи JWT; подключение к MongoDB (URI, имя БД); подключение к S3 (endpoint, бакет, регион, ключи доступа); строка подключения RabbitMQ и учётные данные брокера; межсервисные API-ключи report/economy/subscription/monitor (совпадают у вызывающего и вызываемого); параметры эквайринга Сбербанка (боевой режим — заполнить, для локальной отладки предусмотрен мок-режим); SMTP для рассылки писем (опционально); токены Telegram-ботов (опционально)
herd_manager	Подключение к PostgreSQL (пользователь, пароль, БД, хост, порт, каталог данных); отдельный прикладной пользователь БД; ключи доступа MinIO; порт приложения
ML-сервисы (embed, lameness)	Адрес Triton Inference Server, имя модели
ident_service	Адрес Milvus
cow_weight	Не требует внешних настроек: веса моделей включены в образ
fusionai-frontend	Публичный адрес сайта (NEXT_PUBLIC_SITE_URL), адрес API (NEXT_PUBLIC_API_URL), ключ API Яндекс.Карт
cow_labeler_frame	Подключение к MongoDB (исходная БД и БД разметки); собственный JWT-секрет инструмента; логин/пароль первого администратора; параметры выдачи задач (время аренды клипа, число независимых разметок)
s3_service (MinIO)	Учётные данные администратора хранилища

3.4. Запуск

Порядок запуска (в каждом репозитории — `docker compose up -d --build`):

1. Инфраструктура: s3_service (MinIO; init-контейнер автоматически создаёт бакет media), Triton Inference Server с моделями (поставляется правообладателем: каталог моделей model_repository — модели dinov2_embeddings и lameness_bilstm с файлами config.pbtxt — передаётся архивом по запросу на fusionai@yandex.com и монтируется в контейнер Triton), Milvus.
2. ML-сервисы: embed_service, ident_service, lameness_classification_service, cow_weight (требует GPU — в compose-файле включена резервация устройства NVIDIA).
3. herd_manager — перед стартом приложения автоматически выполняется one-shot контейнер миграций Alembic (создание/обновление схемы PostgreSQL).
4. fusionai-backend — поднимает 6 микросервисов и RabbitMQ.

5. fusionai-frontend, cow_labeler_frame.

6. Настроить публикацию через Nginx (домены личного кабинета, API, хранилища).

3.5. Проверка работоспособности

Проверка	Команда / адрес	Ожидаемый результат
Контейнеры запущены	`docker ps`	Все контейнеры в состоянии Up, без перезапусков
Основной API	`curl http://127.0.0.1:9000/health`	HTTP 200
Прочие прикладные сервисы	`/health` на портах 9100–9500	HTTP 200
herd_manager	`curl http://127.0.0.1:8085/health`	HTTP 200
ML-сервисы	`/health` на портах 8080, 8082; 8083 (cow_weight)	HTTP 200
Личный кабинет	https://fusion-ai.ru/admin в браузере	Открывается страница входа
Очередь уведомлений	Панель RabbitMQ Management	Очередь notifications объявлена (durable)

4. Подключение фермы

4.1. Монтаж оборудования

1. Установить камеры над проходами животных (скотопрогонные коридоры, выход из доильного зала) согласно проекту размещения: камера верхнего ракурса — над проходом на высоте 2,5–3,5 м, оптическая ось вертикально вниз; боковая камера — на высоте 1,5–2,0 м на расстоянии 5–7 м от прохода, оптическая ось перпендикулярна направлению движения. Животное должно полностью попадать в кадр на протяжении не менее 4–5 шагов; поток не менее 1920×1080 и не менее 25 кадров/с; освещённость равномерная, без засветки объектива. IP-камеры подключить в локальную сеть фермы, USB-камеры — непосредственно к плате.

2. Смонтировать вычислительную плату Radxa в защищённом от пыли и влаги месте, обеспечить питание и подключение к локальной сети фермы с выходом в интернет.

3. На плату устанавливается ОС Linux, Docker и ПО уровня фермы (репозиторий fusionai_video_workers); установку и первичную настройку платы выполняет правообладатель или уполномоченный специалист.

4.2. Конфигурация платы

Конфигурация хранится на плате в каталоге config (отдельный репозиторий конфигураций video_workers_config):

Файл	Назначение
hardware.yaml	Основной конфиг платы. Секция server: тип камер `camera_type` (ip или v4l2 — обязательное поле), ссылка на файл описания камер `camera_config`. Значения нормализуются по регистру и пробелам

Файл	Назначение
cameras_ip.yaml (или иной файл по `camera_config`)	Перечень камер платы: роли (upper/front/side), адреса RTSP-потоков либо устройства v4l2
.env	Идентификаторы активных камер (CAM_IDS)

После правки конфигурации перезапустить контейнеры платы (`docker compose up -d` на плате). Контейнеры платы: frameserver, orchestrator, tinder, streamer и сервисные.

4.3. Связь платы с сервером

- Данные (клипы проходов, треки, скелетные точки, телеметрия) плата передаёт на сервер по HTTPS: клипы — по poll-модели через rock_receiver_service, телеметрия — POST /boards/report в monitor_service.
- Для удалённого управления (диагностика, перезапуск) настраиваются обратные SSH-туннели (autossh, юниты systemd) до сервера-шлюза правообладателя.
- Проверка подключения: в панели мониторинга (личный кабинет / внутренний раздел мониторинга) плата появляется в списке с актуальной телеметрией; со стороны платы — HTTP-интерфейс frameapi `/status` возвращает состояние камер и сервисов.

5. Первичная настройка через личный кабинет

Выполняется администратором комплекса после установки.

- Создание фермы.** Раздел администрирования → фермы → создать ферму: наименование, организация, часовой пояс. При необходимости включить фермерские модули (состав функций ЛК для конкретной фермы).
- Создание сотрудников.** Раздел «Сотрудники»: завести учётные записи операторов фермы (ФИО, должность, логин), назначить права доступа. Доступ к разделам ЛК регулируется ролями и правами.
- Привязка камер и платы.** Раздел «Камеры/Устройства»: зарегистрировать камеры фермы и связать их с платой; после привязки в ЛК доступны трансляции (HLS) и данные проходов с этих камер.
- Подписка на отчёты.** Раздел «Отчёты»: настроить рассылку аналитики на e-mail — периодичность, время отправки, часовой пояс получателя.
- Проверка: после прохода животных под камерой в разделе «Стадо» появляются записи проходов с идентификацией животного.

6. Эксплуатация: работа в личном кабинете

Вход: <https://fusion-ai.ru/admin>, логин и пароль выдаёт администратор. Интерфейс на русском языке (доступен английский).

Раздел ЛК	Назначение
Главная (панель)	Сводка по ферме: состояние стада, последние события
Стадо (herd)	Электронное стадо: карточки животных, история проходов, вес, упитанность; поиск и фильтры

Раздел ЛК	Назначение
Хромота (lameness)	Животные с выявленной хромотой: степень, динамика, видеоклипы проходов для визуальной проверки
Камеры	Трансляции с камер фермы в реальном времени (HLS)
Устройства	Состояние оборудования фермы: платы, камеры, телеметрия
Сотрудники	Управление учётными записями и правами операторов
Отчёты	PDF-отчёты по стаду за день/период; управление подпиской на e-mail-рассылку
Уведомления	Лента событий: выявленная хромота, системные события
Экономический эффект (economy-effect)	Калькуляторы: экономика содержания, выгода по животному, котировки цены молока
Пользователи, логи, блог, поддержка	Административные разделы: управление пользователями, журналы действий, публикации, обращения в поддержку

Порядок ежедневной работы оператора: просмотр уведомлений о хромоте → проверка видеоклипа животного в разделе «Хромота» → подтверждение/принятие мер → контроль динамики в карточке животного. Ежедневный отчёт приходит на e-mail по расписанию подписки.

Инструмент разметки `cow_labeler` (отдельный веб-адрес, собственные учётные записи) предназначен для персонала правообладателя: разметка видеоданных для дообучения моделей; в эксплуатации фермой не участвует.

7. Обновление программного комплекса

- Обновление серверной части: получение новой версии исходного кода (`git pull` по репозиториям) → пересборка и перезапуск контейнеров ``docker compose up -d --build --remove-orphans``. Миграции схемы PostgreSQL (`herd_manager`) применяются автоматически контейнером миграций при старте; MongoDB схемных миграций не требует.
- У правообладателя обновление автоматизировано конвейером CD (self-hosted раннеры GitHub Actions на целевом сервере): фиксация изменений в основной ветке инициирует пересборку; предусмотрен отдельный тестовый контур (ветка `test`, отдельная БД) для проверки перед выводом в промышленную эксплуатацию.
- Обновление ПО плат: `git pull` репозитория на плате и перезапуск контейнеров. Выполняется удалённо через SSH-туннель.
- Механизмы принудительного обновления или управления из-за рубежа отсутствуют; обновления распространяются только с инфраструктуры правообладателя.
- Порядок отката: возврат на предыдущую ревизию кода (`git checkout`) и пересборка; для схемы PostgreSQL — откат миграций Alembic (``alembic downgrade``).

8. Резервное копирование

Объект	Содержимое	Способ копирования	Рекомендуемая периодичность
MongoDB	Учётные записи, фермы, стадо, проходы, уведомления, биллинг	`mongodump` (полный дамп БД); хранение на отдельном носителе	Ежесуточно
PostgreSQL (herd_manager)	Карточки животных, эмбединги, история хромоты/упитанности	`pg_dump` (логический дамп) либо копия каталога данных при остановленной СУБД	Ежесуточно
MinIO (бакет media)	Видеоклипы, кадры	Копирование средствами `mc mirror` либо файловая копия docker-тома	Еженедельно / по объёму
Файлы конфигурации	`.env` всех сервисов, конфигурация Nginx, compose-файлы	Копия в защищённое хранилище (секреты — только в зашифрованном виде)	При каждом изменении
Данные инструмента разметки	БД разметки (MongoDB), медиафайлы (docker-том cow_labeler_media)	mongodump + копия тома	Еженедельно

Восстановление: развернуть сервисы по разделу 3, затем восстановить дампы (`mongorestore`, `pg\_restore`/`psql`, обратное копирование медиа), проверить работоспособность по п. 3.5. Данные на платах ферм — временные буферы (кадры, HLS-сегменты в RAM); резервному копированию не подлежат: при утрате платы данные восстанавливаются с сервера.

Приложение А. Перечень портов серверных сервисов (публикуются только на loopback, доступ извне — через Nginx):

Сервис	Порт	Сервис	Порт
back_service	9000	embed_service	8080
report_service	9100	ident_service	8081
monitor_service	9200	lameness_classification_service	8082
rock_receiver_service	9300	cow_weight	8083
subscription_service	9400	herd_manager	8085
economy_service	9500	fusionai-frontend	3000