

??????????

Развёртывание, доступ, безопасность и мониторинг

- [Развёртывание сервера](#)
- [Доступ и безопасность](#)
- [Мониторинг](#)

???????????????? ???? ?????

Руководство для администратора: как развернуть сервер ARTEX и окружающую инфраструктуру.

Имена образов и переменные окружения ХАБа в примерах ниже приведены как шаблон. Перед развёртыванием сверьте их с актуальным репозиторием платформы.

??????? ?????????????????

Компонент	Контейнер	Обязательность
ХАБ платформы (Gateway, Stream Engine, Device Manager)	образ ARTEX HUB	Обязательно
Веб-интерфейс оператора	статический фронтенд (React), раздаётся ХАБом (ClientIOModule) или отдельным веб-сервером	Обязательно
Redis	redis	Рекомендуется
PostgreSQL	postgres	Обязательно для многопользовательского режима
S3-совместимое хранилище	minio/minio	Для записи видео и датасетов
Time Series DB	например, TimescaleDB или InfluxDB	Для записи телеметрии
AI Runtime / модели	отдельные контейнеры с GPU	При использовании моделей ИИ
Prometheus + Grafana	prom/prometheus, grafana/grafana	Рекомендуется

????????????? ? ?????????

Ресурс	Минимум (демо, 1-5 роботов)	Рекомендуется (соревнования, модели ИИ)
CPU	4 ядра	8+ ядер
RAM	8 ГБ	32 ГБ
GPU	не требуется	NVIDIA с поддержкой CUDA для YOLO / RL
Диск	50 ГБ SSD	1 ТБ+ под видео и датасеты

Ресурс	Минимум (демо, 1-5 роботов)	Рекомендуется (соревнования, модели ИИ)
Сеть	100 Мбит/с	1 Гбит/с; ~2-5 Мбит/с на каждый видеопоток
ПО	Linux, Docker, Docker Compose	Kubernetes-кластер

??? 1. ?????????????????

```
# Docker и Compose
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER

# Для моделей ИИ на GPU – NVIDIA Container Toolkit
# https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/
```

Доступ администраторов к серверу — только по SSH-ключам.

??? 2. ????????????????? (Docker Compose)

```
# docker-compose.infra.yml
services:
  redis:
    image: redis:7
    restart: unless-stopped
    ports: ["6379:6379"]

  postgres:
    image: postgres:16
    restart: unless-stopped
    environment:
      POSTGRES_DB: artex
      POSTGRES_USER: artex
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes: ["pg_data:/var/lib/postgresql/data"]
```

```
minio:
  image: minio/minio
  restart: unless-stopped
  command: server /data --console-address ":9001"
  environment:
    MINIO_ROOT_USER: ${MINIO_USER}
    MINIO_ROOT_PASSWORD: ${MINIO_PASSWORD}
  ports: ["9000:9000", "9001:9001"]
  volumes: ["minio_data:/data"]
```

```
timescaledb:
  image: timescale/timescaledb:latest-pg16
  restart: unless-stopped
  environment:
    POSTGRES_PASSWORD: ${TSDB_PASSWORD}
  volumes: ["tsdb_data:/var/lib/postgresql/data"]
```

```
volumes:
  pg_data:
  minio_data:
  tsdb_data:
```

```
docker compose -f docker-compose.infra.yml up -d
```

??? 3. ???????

```
# docker-compose.hub.yml (шаблон)
services:
  artex-hub:
    image: <registry>/artex-hub:<version>
    restart: unless-stopped
    ports:
      - "80:80"          # веб-интерфейс, WebSocket, Socket.IO для роботов
    environment:
      REDIS_URL: redis://redis:6379
      POSTGRES_DSN: postgres://artex:${POSTGRES_PASSWORD}@postgres:5432/artex
      S3_ENDPOINT: http://minio:9000
      JWT_SECRET: ${JWT_SECRET}
```

Роботы по умолчанию подключаются к адресу `http://artex.arcon:80` — это значение параметра `server` в SDK. Настройте DNS-запись (или `/etc/hosts` на роботах) либо укажите реальный адрес сервера в `params.json` каждого робота.

??? 4. ????????

1. Откройте веб-интерфейс в браузере: `http://<адрес-сервера>/`. Должна открыться страница «Подключения».
2. Запустите на тестовом устройстве минимальный пример из раздела [Быстрый старт](#).
3. Убедитесь, что устройство появилось в таблице подключений, а телеметрия обновляется.

????????????? ? Kubernetes

Для продуктивного режима каждый сервис разворачивается отдельным Deployment:

- **Gateway / Stream Engine / Device Manager** — горизонтально масштабируются; для Socket.IO и WebSocket включите «липкие» сессии (sticky sessions) на Ingress.
- **AI Runtime** — Deployment на узлах с GPU (`nodeSelector` / `tolerations`), ресурс `nvidia.com/gpu`.
- **Хранилища** — StatefulSet или управляемые сервисы.
- Секреты (`JWT_SECRET`, пароли БД, ключи S3) — через Kubernetes Secrets.
- Метрики — через ServiceMonitor для Prometheus.

???????????

1. Обновите образ в манифесте/compose-файле.
2. `docker compose pull && docker compose up -d` (или `kubectl rollout restart deployment/<имя>`).
3. Роботы переподключатся автоматически (SDK делает до 5 попыток с задержкой 1–5 с). Если сервер был недоступен дольше — перезапустите ПО на роботах.

Вопросы и ответы

Платформа работает с физическими устройствами, поэтому контроль доступа и безопасность — обязательная часть конфигурации.

Вопросы

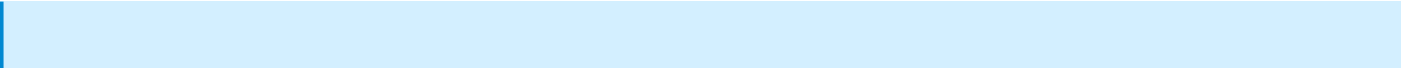
Роль	Права
Администратор	Управление платформой через WEB-портал: мониторинг роботов и операторов, настройка видимости роботов, настройка хранения данных, управление пользователями и API-ключами
Оператор	Подключение к доступным ему роботам, просмотр видео и телеметрии, управление, запуск своих моделей ИИ, мониторинг их работы
Модель ИИ	Сервисная учётная запись: чтение видео и телеметрии назначенных роботов, отправка команд управления
Робот	Устройство: передача видео и телеметрии, получение команд
Зритель (соревнования)	Просмотр трансляции и лидерборда без права управления

Вопросы и ответы

- **JWT** — авторизация пользователей веб-интерфейса и WebSocket-подключений.
- **RBAC** — ролевая модель: какие роботы видны оператору и какие действия ему разрешены.
- **API-ключи** — для моделей ИИ и внешних систем.
- **SSH-ключи** — для администрирования серверов.
- **Rate limiting** в Gateway — защита от перегрузки командами.

Вопросы и ответы

Администратор ограничивает, каких роботов видит каждый оператор. Оператор может подключиться и управлять только видимыми ему роботами; модели ИИ оператора получают доступ к тем же роботам.



На этапе демо-прототипа форма ограничения «видимости» роботов для операторов реализуется без функционала (только форма). Поле «Безопасность» в таблице подключений зарезервировано под будущие настройки прав доступа.

????????? ???????????

Одновременно роботу могут отправлять команды оператор и модели ИИ. Правила:

1. Команды **Стоп** и **DISARM** от любого авторизованного источника исполняются немедленно.
2. Команда оператора имеет приоритет над командой модели ИИ (режим «подстраховки»).
3. Модель ИИ управляет роботом, пока оператор не вмешался; после вмешательства управление возвращается модели по решению оператора.

????????? ?????????????? ??????????????????

- Перед запуском автономного режима проверьте работу кнопки **Стоп** на конкретном роботе.
- Бортовое ПО должно останавливать приводы при потере связи с сервером (см. [Примеры роботов](#) в книге «ПО для робота»).
- Для дронов после посадки всегда выполняйте **DISARM**.
- Используйте отдельную сеть (VLAN / Wi-Fi) для роботов полигона.
- Храните секреты (JWT, пароли, ключи) только в переменных окружения / Secrets, не в репозитории.
- Для публичного доступа используйте HTTPS/WSS (TLS-терминация на Ingress или обратном прокси).

??????????

??? ???????????

Объект	Где смотреть	Показатели
Подключённые роботы	Портал управления, страница «Подключения» веб-интерфейса	Список устройств, SID, тип, статус, время последнего heartbeat
Подключённые операторы	Портал управления	Кто подключён, к каким роботам
Модели ИИ	Web UI оператора (мониторинг моделей), AI Runtime	Входные и выходные данные, команды, логи, загрузка GPU
Сервисы платформы	Prometheus + Grafana	Задержки потоков, частота кадров и телеметрии, ошибки, использование CPU/RAM/сети
Серверная логика	ConsoleLogModule	Все данные, проходящие через выбранные порты конвейера
Робот	Логи SDK (<code>all.log</code> , <code>error.log</code>)	События подключения, heartbeat, полученные команды, ошибки

????????? ?????????? ? ??????????

Метрика	Норма	Действие при отклонении
Задержка видео камера → оператор	< 200 мс	Снизить разрешение/качество кадра, проверить сеть
Задержка команда → привод	< 50 мс	Проверить загрузку ХАБа и сеть робота
Пропуск heartbeat	0	Проверить питание и связь робота
Частота телеметрии	стабильна (до 30/с)	Проверить основной цикл робота

????????? ??????? ? ?????????? ???????????

Сервер может запросить у робота файлы логов командой типа `file` — SDK отвечает событием `file` с содержимым:

Команда	Результат
---------	-----------

<code>{"file": "log"}</code>	Содержимое <code>all.log</code>
<code>{"file": "error"}</code>	Содержимое <code>error.log</code>
<code>{"file": "clear"}</code>	Очистка обоих файлов логов

Логи в файлы пишутся, если на работе задано `"logger_type": "file"`. Подробнее — [Протокол обмена с сервером](#).

Grafana: ?????????????? ???????

- Число подключённых роботов / операторов / моделей.
- Задержки по каждому потоку (p50, p95, p99).
- Пропускная способность видео по роботам.
- Ошибки Gateway (4xx/5xx), отказы авторизации.
- Загрузка GPU и очередь задач AI Runtime.