

**Программное обеспечение CyberStudio – платформа для
предиктивной аналитики оборудования и оптимизации режимов
технологических процессов
v 3.20.0
Руководство по установке**



Таблица 1.

История изменений

Версия	Дата	Комментарий	Автор
1	07.04.2026	Первая версия документа	Гавриленков С.И.

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ	4
2	ТРЕБОВАНИЯ К СЕРВЕРУ	4
3	СОСТАВ ДИСТРИБУТИВА	4
4	АРХИТЕКТУРА ДЕПЛОЙМЕНТА	5
4.1	Инфраструктурные сервисы	5
4.2	Микросервисы платформы CyberStudio	5
5	ПРОЦЕДУРА УСТАНОВКИ	6
5.1	Шаг 1. Подготовка сервера	6
5.2	Шаг 2. Копирование дистрибутива на сервер	6
5.3	Шаг 3. Импорт Docker-образов	6
5.4	Шаг 4. Настройка FQDN сервера	6
5.5	Шаг 5. Настройка дополнительных параметров (при необходимости)	7
5.6	Шаг 6. Запуск платформы	8
5.7	Шаг 7. Проверка доступности	8
6	НАСТРОЙКА АВТОЗАПУСКА	8
7	УПРАВЛЕНИЕ ПЛАТФОРМОЙ	8
8	СТРУКТУРА КАТАЛОГОВ ДАННЫХ	9
9	СЕТЕВАЯ КОНФИГУРАЦИЯ	9

1 ОБЩИЕ СВЕДЕНИЯ

Настоящее руководство описывает процедуру развертывания платформы CyberStudio на целевом сервере. Дистрибутив платформы поставляется в виде архива формата `.tar`, который может быть передан на USB-носителе или загружен из файлового архива.

Дистрибутив содержит:

- Docker-образы инфраструктурных компонентов (база данных, брокер сообщений, хранилище и др.);
- Docker-образы микросервисов платформы CyberStudio;
- Конфигурационные файлы деплоймента (Docker Compose, переменные окружения, конфигурации сервисов);
- Скрипт автоматизации развертывания `deploy.sh`;
- Юнит-файл `systemd` для автозапуска платформы;
- Архив Docker Engine и Docker Compose для офлайн-установки (при необходимости).

2 ТРЕБОВАНИЯ К СЕРВЕРУ

Для установки платформы целевой сервер должен соответствовать следующим требованиям:

- Операционная система: Linux (рекомендуется Ubuntu 22.04 LTS или выше);
- Docker Engine версии 26.0 и выше;
- Docker Compose версии 2.26 и выше;
- Доступ к серверу по SSH с правами суперпользователя (`root`) или через `sudo`;
- Свободное дисковое пространство: не менее 50 ГБ для образов и данных;
- Оперативная память: не менее 16 ГБ (рекомендуется 32 ГБ);
- Сетевой доступ по портам 80 (HTTP) и 443 (HTTPS).

Примечание: Docker и Docker Compose должны быть установлены до начала развертывания. Если на сервере отсутствует Docker, в составе дистрибутива поставляется архив `docker-26.1.0_dc-v2.26.1.tgz` для офлайн-установки.

3 СОСТАВ ДИСТРИБУТИВА

Архив дистрибутива имеет имя вида `cyberphysics-local_<ДАТА>_<ВРЕМЯ>.tar` и содержит следующие компоненты:

Таблица 2.

Состав файлов дистрибутива

Файл / каталог	Назначение
<code>deploy.sh</code>	Скрипт управления платформой (запуск, остановка, импорт образов)
<code>deployment/</code>	Каталог с конфигурацией деплоймента
<code>deployment/.env</code>	Главный файл переменных окружения
<code>deployment/compose.yml</code>	Корневой файл Docker Compose
<code>deployment/infra/</code>	Конфигурации инфраструктурных сервисов
<code>deployment/cyberstudio/</code>	Конфигурации микросервисов платформы

images_infra.tgz	Архив Docker-образов инфраструктуры
images_deployment.tgz	Архив Docker-образов микросервисов CyberStudio
docker-26.1.0_dc-v2.26.1.tgz	Архив Docker Engine и Docker Compose (для офлайн-установки)
systemd/cyberstudio.service	Юнит-файл systemd для автозапуска

4 АРХИТЕКТУРА ДЕПЛОЙМЕНТА

Платформа состоит из двух групп сервисов, каждая из которых описана собственным набором файлов Docker Compose.

4.1 ИНФРАСТРУКТУРНЫЕ СЕРВИСЫ

Инфраструктурные компоненты расположены в каталоге `deployment/infra/` и включают:

Таблица 3.

Инфраструктурные сервисы

Сервис	Назначение
PostgreSQL 17	Реляционная база данных
ClickHouse	Аналитическая колоночная СУБД (хранение временных рядов)
Apache Kafka	Брокер сообщений
Kafka Connect	Коннекторы интеграции Kafka
Schema Registry	Реестр схем сообщений Kafka
Redis	Кэш-хранилище
MinIO	S3-совместимое объектное хранилище
Keycloak	Сервер аутентификации и авторизации (SSO)
OAuth2-Proxy	Прокси аутентификации для веб-интерфейса
Traefik	Обратный прокси и балансировщик нагрузки
Grafana	Мониторинг и визуализация метрик
Victoria Metrics	Хранилище метрик
Loki	Агрегация логов
Alloy	Агент сбора метрик и логов

4.2 МИКРОСЕРВИСЫ ПЛАТФОРМЫ CYBERSTUDIO

Микросервисы расположены в каталоге `deployment/cyberstudio/` и включают:

Таблица 4.

Микросервисы платформы

Сервис	Назначение
frontend (select-project)	Веб-интерфейс выбора проекта
admin	Администрирование
datalake	Сервис сбора данных
historical	Исторический сервис
design	Дизайн проекта
model	Моделирование
malfunction	Обнаружение неисправностей
prognosis	Прогнозирование
canvas	Дизайнер мнемосхем
business-logic	Бизнес-логика
monitoring	Мониторинг
optimization	Оптимизация
import	Импорт данных
export	Экспорт данных
executor	Исполнитель расчётов
kafka-manager	Управление Kafka

clickhouse-migrations	Миграции схемы ClickHouse
-----------------------	---------------------------

Каждый микросервис имеет собственный каталог с файлом `compose.yml` и файлом переменных окружения (`.env`). Все `compose`-файлы подключаются в корневой `deployment/compose.yml` через директиву `include`.

5 ПРОЦЕДУРА УСТАНОВКИ

5.1 ШАГ 1. ПОДГОТОВКА СЕРВЕРА

Подключитесь к целевому серверу по SSH и убедитесь, что Docker и Docker Compose установлены:

```
docker --version
docker compose version
```

Создайте каталог для данных платформы:

```
sudo mkdir -p /data
```

Примечание: Путь `/data` используется по умолчанию для хранения данных всех сервисов (PostgreSQL, ClickHouse, Kafka и др.). При необходимости его можно изменить в файле `deployment/.env` (параметр `DATA_BASE_DIR`).

5.2 ШАГ 2. КОПИРОВАНИЕ ДИСТРИБУТИВА НА СЕРВЕР

Скопируйте архив дистрибутива на сервер, например с помощью `scp`:

```
scp cyberphysics-local_<ДАТА>_<ВРЕМЯ>.tar user@<АДРЕС_СЕРВЕРА>:/opt/cyberstudio/
```

Подключитесь к серверу и распакуйте архив:

```
cd /opt/cyberstudio
tar xf cyberphysics-local_<ДАТА>_<ВРЕМЯ>.tar
```

После распаковки в текущем каталоге появятся файлы и каталоги, перечисленные в разделе «Состав дистрибутива».

5.3 ШАГ 3. ИМПОРТ DOCKER-ОБРАЗОВ

Загрузите Docker-образы из архивов в локальный реестр Docker:

```
sudo ./deploy.sh import
```

Скрипт автоматически найдёт все файлы `images*.tgz` и выполнит `docker load` для каждого из них. Процесс может занять несколько минут в зависимости от производительности дисковой подсистемы.

5.4 ШАГ 4. НАСТРОЙКА FQDN СЕРВЕРА

FQDN (полное доменное имя) сервера, на котором будет доступна платформа, необходимо указать в трёх конфигурационных файлах. По умолчанию используется значение `local.cyberstudio.online`.

5.4.1 ФАЙЛ 1: DEPLOYMENT/.ENV

Откройте файл `deployment/.env` и замените значение параметра `APP_HOST` на актуальный FQDN сервера:

```
APP_HOST=<ваш_fqdn>
```

Также при необходимости измените схему доступа (HTTP или HTTPS):

```
WEB_SCHEME=http
```

5.4.2 ФАЙЛ

2:

DEPLOYMENT/CYBERSTUDIO/FRONTEND/FRONTEND.ENV

Откройте файл `deployment/cyberstudio/frontend/frontend.env` и замените все вхождения FQDN:

```
NEXT_PUBLIC_API_URL=http://<ваш_fqdn>/cyberstudio/api
NEXT_PUBLIC_WS_URL=ws://<ваш_fqdn>/cyberstudio/api
NEXT_PUBLIC_LOGOUT_URL=http://<ваш_fqdn>/cyberstudio/oauth2/sign_out
```

Внимание: Схема протокола (`http/https` и `ws/wss`) должна соответствовать значению `WEB_SCHEME` из файла `.env`. Если используется HTTPS, замените `http://` на `https://` и `ws://` на `wss://`.

5.4.3 ФАЙЛ

3:

DEPLOYMENT/INFRA/OAUTH2-PROXY/OAUTH2-PROXY.CFG

Откройте файл `deployment/infra/oauth2-proxy/oauth2-proxy.cfg` и замените все вхождения FQDN в следующих параметрах:

```
redirect_url="http://<ваш_fqdn>/cyberstudio/oauth2/callback"
oidc_issuer_url="http://<ваш_fqdn>/cyberstudio/keycloak/realms/cyberstudio"
whitelist_domains="<ваш_fqdn>"
backend_logout_url="http://<ваш_fqdn>/cyberstudio/keycloak/realms/cyberstudio/protocol/openid-connect/logout?id_token_hint={id_token}&post_logout_redirect_uri=http://<ваш_fqdn>"
```

Примечание: Параметр `{id_token}` в строке `backend_logout_url` является шаблонной переменной OAuth2-Прокси и не подлежит замене.

5.5 ШАГ 5. НАСТРОЙКА ДОПОЛНИТЕЛЬНЫХ ПАРАМЕТРОВ (ПРИ НЕОБХОДИМОСТИ)

В файле `deployment/.env` доступны следующие параметры:

Таблица 5.

Параметры файла .env		
Параметр	Значение по умолчанию	Описание
DATA_BASE_DIR	/data	Каталог хранения данных сервисов
MONITORING_DATA_BASE_DIR	/data	Каталог хранения данных мониторинга
EXPOSED_IP	127.0.0.1	IP-адрес для проброса портов (PostgreSQL и др.)
APP_HOST	local.cyberstudio.online	FQDN сервера
WEB_SCHEME	http	Схема доступа (http или https)

APP_VER	cyberstudio-3.18.0	Версия платформы
---------	--------------------	------------------

Внимание: Параметр `EXPOSED_IP` определяет, на каком интерфейсе будут доступны проброшенные порты сервисов (например, PostgreSQL на порту 5432). Значение `127.0.0.1` ограничивает доступ только с локального хоста. Для удалённого доступа к БД укажите `0.0.0.0` или конкретный IP-адрес сервера.

5.6 ШАГ 6. ЗАПУСК ПЛАТФОРМЫ

Запустите платформу командой:

```
sudo ./deploy.sh start
```

Скрипт выполнит `docker compose up -d --wait` в каталоге `deployment/`. Все сервисы, описанные в корневом `compose.yml` и подключённых `compose`-файлах, будут запущены в фоновом режиме.

Первый запуск может занять несколько минут: PostgreSQL инициализирует базы данных всех сервисов, а Keycloak импортирует начальную конфигурацию realm.

Проверить статус сервисов можно командой:

```
cd /opt/cyberstudio/deployment
docker compose ps
```

5.7 ШАГ 7. ПРОВЕРКА ДОСТУПНОСТИ

Откройте в браузере адрес платформы:

```
http://<ваш_fqdn>/cyberstudio
```

При корректной настройке вы будете перенаправлены на страницу авторизации Keycloak. Для первого входа используйте учётные данные администратора Keycloak, указанные в файле `deployment/infra/keycloak/keycloak.env`.

6 НАСТРОЙКА АВТОЗАПУСКА

Для автоматического запуска платформы при загрузке операционной системы выполните:

```
sudo ./deploy.sh autostart_enable
```

Скрипт скопирует файл `systemd/cyberstudio.service` в каталог `/etc/systemd/system/` и активирует юнит через `systemctl enable`.

Примечание: Перед включением автозапуска убедитесь, что переменная `${SCRIPT_DIR}` в файле `systemd/cyberstudio.service` указывает на фактический каталог установки. При необходимости отредактируйте путь в полях `WorkingDirectory` и `ExecStart/ExecStop`.

7 УПРАВЛЕНИЕ ПЛАТФОРМОЙ

После установки управление платформой осуществляется через скрипт `deploy.sh`:

Таблица 6.

Команды управления платформой

Команда	Описание
<code>./deploy.sh start</code>	Запуск всех сервисов

./deploy.sh stop	Остановка всех сервисов
./deploy.sh restart	Перезапуск всех сервисов
./deploy.sh restart cyberstudio	Перезапуск только микросервисов CyberStudio
./deploy.sh restart infra	Перезапуск только инфраструктурных сервисов
./deploy.sh import	Импорт Docker-образов из архивов

8 СТРУКТУРА КАТАЛОГОВ ДАННЫХ

После запуска платформы в каталоге `DATA_BASE_DIR` (по умолчанию `/data`) создаётся следующая структура:

Таблица 7.

Каталоги данных

Каталог	Сервис	Описание
/data/postgres/data	PostgreSQL	Данные реляционной БД
/data/clickhouse/data	ClickHouse	Данные аналитической БД
/data/kafka/data	Kafka	Данные брокера сообщений
/data/minio/data	MinIO	Объектное хранилище (файлы, модели)
/data/redis/data	Redis	Данные кэша
/data/traefik/data	Traefik	Сертификаты TLS (при использовании HTTPS)
/data/grafana/data	Grafana	Дашборды и настройки мониторинга
/data/loki/data	Loki	Индексы и данные логов

9 СЕТЕВАЯ КОНФИГУРАЦИЯ

Все сервисы платформы работают в единой Docker-сети `cyberstudio` с подсетью `172.16.0.0/24`. Обратный прокси Traefik принимает входящие соединения на портах 80 (HTTP) и 443 (HTTPS) и маршрутизирует запросы к соответствующим сервисам на основе заголовка `Host` и префикса пути `URL`.

Все микросервисы доступны по единому FQDN через префиксы вида `/cyberstudio/<имя_сервиса>`. Аутентификация обеспечивается связкой Keycloak + OAuth2-Proxy: неавторизованные запросы автоматически перенаправляются на страницу входа.