

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «МКСКОМ»

УТВЕРЖДЕНО

Генеральный директор

ООО «МКСКОМ»



/П.М. Горбачев/

«17» ноября 2025 г.

"М.П."

Платформа Дружба

Инструкция по установке и запуску платформы

На 11 листах

Редакция от 17.11.2025

Действует с «17» ноября 2025 г.

Москва 2024

СОДЕРЖАНИЕ

1	ТЕРМИНЫ И СОКРАЩЕНИЯ	3
2	ВВЕДЕНИЕ	4
3	УСТАНОВКА И ЗАПУСК ПЛАТФОРМЫ	5
3.1	Минимальные системные требования	5
3.2	Рекомендованные системные требования	5
3.3	Уровень подготовки пользователя	5
3.4	Состав поставки	6
3.5	Подготовка сервера (Docker)	6
3.6	Распаковка дистрибутива	7
3.7	Установка и первичная настройка	7
3.8	Проверка работоспособности	8
3.9	Эксплуатация (запуск/остановка)	9
3.10	Устранение неполадок	10

1 ТЕРМИНЫ И СОКРАЩЕНИЯ

Термин/Сокращение	Пояснение
Дружба	Программный комплекс интеграционной платформы
ОС	Операционная система
Платформа, Система	Платформа «Дружба»
ПО	Программное обеспечение
БД, СУБД	База данных
Закрытый контур	Среда без доступа к внешним Docker Registry
IMAGE_TAG	Версия (tag) образов esb-friendship-* в поставке
ENV_SUFFIX	Суффикс имён Kafka-топиков и бакетов MinIO (например: onprem)
Поставка, дистрибутив	Архив esb-friendship-backend-<версия>-delivery.tar.gz

2 ВВЕДЕНИЕ

Настоящая инструкция предназначена для администратора, выполняющего развёртывание платформы «Дружба» в закрытом контуре (on-premise) на сервере заказчика.

Документ содержит:

- системные требования к серверу;
- требования к квалификации администратора;
- порядок установки Docker-окружения;
- порядок распаковки поставки, установки и проверки Системы.

Особенности поставки:

- все компоненты (приложение, PostgreSQL, MinIO, Kafka, RabbitMQ) разворачиваются в Docker-контейнерах из предварительно загруженных tar-образов;
- доступ к Harbor и интернету на целевом сервере не требуется;
- архитектура образов: linux/amd64;
- поддерживаемые ОС: Astra Linux 1.7, Ubuntu 20.04 и новее (amd64).

Установку и настройку Системы должен выполнять пользователь с правами администратора (root или член группы docker).

3 УСТАНОВКА И ЗАПУСК ПЛАТФОРМЫ

3.1 Минимальные системные требования

Минимальные требования к системе, на которой будет развернуто приложение:

- Оперативная память – от 8 Гб.
- Процессор – Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz или эквивалентный (4 CPU).
- Операционная система – Astra-linux 1.7. или Ubuntu 20.04+ (amd64).
- Объем жесткого диска от 50 Гб свободного места (образы Docker, данные БД, Kafka)
- Программное обеспечение:
 - Docker Engine 20.10 и новее;
 - Docker Compose v2 (команда: docker compose).
- Сеть:
 - исходящий доступ в интернет на целевом сервере не требуется;
 - свободные порты (по умолчанию: 8989, 5432, 9000, 9001, 29092, 39092, 49092, 5672, 15672)

Примечание: Конфигурация с 4 Гб ОЗУ и 2 CPU может быть недостаточна для стабильной работы полного стека (3 узла Kafka Kraft + Java-приложение)

3.2 Рекомендованные системные требования

Рекомендованные требования к системе, на которой будет развернуто приложение:

- Оперативная память – от 16 Гб.
- Процессор – Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz или эквивалентный (6 CPU).
- Операционная система – Astra-linux 1.7. или Ubuntu 20.04+ (amd64).
- Объем жесткого диска от 100 GB свободного места (с учётом роста данных в MinIO и PostgreSQL)
- Программное обеспечение:
 - Docker Engine 24+;
 - Docker Compose v2.

3.3 Уровень подготовки пользователя

Администратор системы должен обладать следующими знаниями и навыками:

- администрирование серверов Linux (Astra Linux, Ubuntu);
- работа с Docker и Docker Compose;
- базовое администрирование СУБД PostgreSQL и брокеров сообщений;
- настройка сетевых портов и межсетевого экранирования;
- соблюдение политики информационной безопасности при хранении паролей и учётных данных.

Установку рекомендуется выполнять под учётной записью с правами sudo.

3.4 Состав поставки

Архив esb-friendship-backend-<IMAGE_TAG>-delivery.tar.gz содержит:

- install.sh — скрипт установки и управления
- install_manual.txt — настоящая инструкция
- docker-compose.yaml — описание контейнеров
- .env.example — шаблон переменных окружения
- VERSION — тег образов (IMAGE_TAG)
- README.md — краткое описание поставки
- lib/ — вспомогательные скрипты
- config/ — конфигурация RabbitMQ
- init/ — идиоматичные скрипты инициализации
- images/* .tar — Docker-образы (5 файлов)

Образы в поставке (единый тег IMAGE_TAG):

- esb-friendship-backend:<IMAGE_TAG> — приложение ESB Friendship Backend
- esb-friendship-db:<IMAGE_TAG> — PostgreSQL 16
- esb-friendship-storage:<IMAGE_TAG> — MinIO (S3-совместимое хранилище)
- esb-friendship-kafka:<IMAGE_TAG> — Apache Kafka (KRaft, 3 узла)
- esb-friendship-queue:<IMAGE_TAG> — RabbitMQ

После установки на сервере работают контейнеры:

- esb-backend — API приложения (порт APP_PORT, по умолчанию 8989)
- esb-postgres — база данных PostgreSQL 16
- esb-minio — файловое хранилище
- esb-kafka-1/2/3 — кластер Kafka (3 брокера KRaft)
- esb-rabbitmq — очереди сообщений

3.5 Подготовка сервера (Docker)

На целевом сервере должны быть установлены Docker и Docker Compose v2.

Ubuntu/Debian

```
sudo apt-get update
sudo apt-get install -y docker.io docker-compose-plugin
sudo usermod -aG docker <имя_пользователя>
```

Astra Linux 1.7:

```
sudo apt-get update
sudo apt-get install -y docker docker-compose-v2
sudo usermod -aG docker <имя_пользователя>
```

ВАЖНО: не устанавливайте пакет docker-compose (v1). Используйте только docker-compose-v2.

Проверка:

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

На Astra Linux перед установкой Дружбы рекомендуется настроить параметр ядра `parsec.execstack=1` (см. п. 3.10) — иначе JVM в контейнерах Kafka и backend может не запуститься.

3.6 Распаковка дистрибутива

1. Скопируйте архив поставки на целевой сервер, например:

```
scp esb-friendship-backend-1.0.0-delivery.tar.gz user@server:/tmp/
```

2. Создайте каталог установки и распакуйте архив:

```
sudo mkdir -p /opt/esb-friendship
cd /opt/esb-friendship
sudo tar -xzf /tmp/esb-friendship-backend-1.0.0-delivery.tar.gz
sudo chown -R $USER:$USER /opt/esb-friendship
```

3. Убедитесь в наличии файлов:

```
ls -la install.sh docker-compose.yaml images/*.tar VERSION
```

3.7 Установка и первичная настройка

1. Перейдите в каталог установки:

```
cd /opt/esb-friendship
```

2. Сделайте скрипт исполняемым и запустите установку:

```
chmod +x install.sh
./install.sh
```

3. Скрипт `install.sh` выполняет:

- a) проверку ОС, Docker и Docker Compose v2;
- b) загрузку образов из `images/*.tar` (docker load);
- c) проверку архитектуры образов (amd64);
- d) интерактивный ввод параметров:
 - суффикс окружения `ENV_SUFFIX` (имена Kafka-топиков и бакетов

MinIO);

- портов сервисов (при занятом порту — указать другой);
- учётных данных PostgreSQL, MinIO, RabbitMQ (пароли вводятся дважды, значений по умолчанию нет);
- контекстов Liquibase (можно оставить пустым);
- е) создание файла .env;
- ф) запуск инфраструктуры и идиempотентных init-задач:
 - rabbitmq-init — пользователь RabbitMQ;
 - postgres-init — проверка/создание БД;
 - minio-init — бакеты schemas/structures/attachments/integration-artifacts;
 - kafka-init — 8 Kafka-топиков;
- г) запуск esb-backend (миграции Liquibase выполняются при старте приложения);
- h) вывод статуса компонентов.

4. Первая установка занимает примерно 5–20 минут (зависит от ресурсов сервера и времени инициализации Kafka).

5. Повторный запуск ./install.sh перезапишет .env и запросит параметры заново.

Запуск по готовому .env (без интерактива):

```
./install.sh start
```

Файл .env создаётся только через ./install.sh или вручную на основе .env.example.

Обязательные переменные: IMAGE_TAG, POSTGRES_PASSWORD, MINIO_ROOT_PASSWORD, RABBITMQ_DEFAULT_PASS.

3.8 Проверка работоспособности

Автоматическая проверка всех компонентов:

```
./install.sh status
```

Все проверки должны иметь статус ОК. Код выхода 0 — успех, 1 — есть проблемы.

Проверка API вручную:

```
curl -s http://localhost:<APP_PORT>/esb-friendship-backend/actuator/health
```

APP_PORT — из файла .env (по умолчанию 8989).

Ожидаемый ответ содержит: "status":"UP"

Список контейнеров:

```
docker compose -f docker-compose.yaml ps
```


Порты по умолчанию (фактические значения — в .env):

APP_PORT	8989	API приложения
POSTGRES_PORT	5432	PostgreSQL
MINIO_API_PORT	9000	MinIO API
MINIO_CONSOLE_PORT	9001	веб-консоль MinIO
KAFKA_PORT_1	29092	Kafka broker-1
KAFKA_PORT_2	39092	Kafka broker-2
KAFKA_PORT_3	49092	Kafka broker-3
RABBITMQ_PORT	5672	RabbitMQ AMQP
RABBITMQ_MGMT_PORT	15672	веб-интерфейс RabbitMQ

Адрес API после установки:

http://<IP_или_имя_сервера>:<APP_PORT>/esb-friendship-backend/

3.9 Эксплуатация (запуск/остановка)

Запуск после остановки (при наличии .env):

```
./install.sh start
```

или:

```
docker compose -f docker-compose.yaml up -d
```

Остановка (данные в Docker-томах сохраняются):

```
docker compose -f docker-compose.yaml down
```

Проверка состояния:

```
./install.sh status
```

Просмотр логов приложения:

```
docker logs esb-backend --tail 100 -f
```

Повторная идиempотентная инициализация (при необходимости):

```
docker compose -f docker-compose.yaml run --rm rabbitmq-init  
docker compose -f docker-compose.yaml run --rm postgres-init  
docker compose -f docker-compose.yaml run --rm minio-init  
docker compose -f docker-compose.yaml run --rm kafka-init
```

3.10 Устранение неполадок

JVM: Failed to mark memory page as executable (Astra Linux)

Ошибка в логах Kafka или esb-backend при старте JVM.

Включите параметр ядра `parsec.execstack=1`:

```
sudo nano /etc/default/grub
```

Добавьте в `GRUB_CMDLINE_LINUX_DEFAULT`, например:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet parsec.execstack=1"
```

Примените и перезагрузите:

```
sudo update-grub  
sudo reboot
```

После перезагрузки:

```
docker run --rm --entrypoint java esb-friendship-kafka:<IMAGE_TAG> -  
version  
./install.sh start
```

Kafka: контейнеры `esb-kafka-*` в состоянии `Restarting`

Остановите стек, удалите тома Kafka и установите заново:

```
docker compose -f docker-compose.yaml down  
docker volume rm esb-friendship-kafka-data-1 \  
    esb-friendship-kafka-data-2 \  
    esb-friendship-kafka-data-3  
./install.sh
```

RabbitMQ: `health DOWN, ACCESS_REFUSED`

Обновите пользователя и перезапустите приложение:

```
docker compose -f docker-compose.yaml run --rm rabbitmq-init  
docker compose -f docker-compose.yaml restart esb-backend
```

Порт занят при установке

Укажите другой порт в ответ на запрос `install.sh`.

Приложение не отвечает

Подождите 1–2 минуты после старта, затем:

```
docker logs esb-backend --tail 100  
./install.sh status
```

Ошибка: `Not supported URL scheme http+docker` (Astra Linux)

Установлен `docker-compose v1` вместо `v2`:

```
sudo apt-get remove -y docker-compose  
sudo apt-get install -y docker-compose-v2
```

```
docker compose version
./install.sh
```

Команды install.sh

./install.sh	Установка (интерактивно, создаёт .env)
./install.sh start	Запуск по готовому .env
./install.sh status	Проверка всех компонентов
./install.sh --help	Справка