ПАРТНЕР ЛИДС

Общество с ограниченной ответственностью «Партнер Лидс»
ИНН/КПП: 6317172338 / 631701001
ОГРН: 1266300001967
Основной ОКВЭД: 62.01
Дополнительный ОКВЭД: 62.02
Юридический адрес: 443010, Самарская область, г. Самара, ул. Куйбышева, д. 128А, офис В209

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «ПАРТНЕР ЛИДС»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

(Руководство системного программиста — ГОСТ 19.503-79)

RU.6317172338.00001-01 32 01

Листов 7

2026

АННОТАЦИЯ

Настоящий документ содержит инструкцию по установке и развёртыванию программного обеспечения «Партнер Лидс», предназначенную для системного программиста (специалиста, осуществляющего развёртывание и сопровождение программы).

Документ разработан в соответствии с ГОСТ 19.503-79 «Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению» и ГОСТ 19.105-78 «Единая система программной документации. Общие требования к программным документам».

В основу документа положен эксплуатируемый сценарий автоматизированного развёртывания (deploy.sh). Значения, специфичные для контрольного (тестового) контура — учётные данные по умолчанию, локальные адреса — отмечены отдельно и подлежат обязательной замене при развёртывании в продуктивном контуре.

СОДЕРЖАНИЕ

1. Общие сведения о программе 3

2. Структура программы 3

3. Настройка программы 4

4. Проверка программы 6

5. Дополнительные возможности 6

6. Сообщения системному программисту 7

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Назначение и функции программы приведены в документе «Описание функциональных характеристик». Настоящий раздел содержит сведения о технических и программных средствах, необходимых для развёртывания и функционирования программы.

Для развёртывания программы необходимы:

— сервер (физический или виртуальный) под управлением операционной системы с поддержкой Docker;

— Docker и плагин Docker Compose (версии 2) — программа развёртывается в виде набора контейнеров;

— доступ к сети Интернет или внутренней сети — для загрузки образов и работы клиентских приложений;

— свободные сетевые порты для сервисов платформы (значения задаются в файле конфигурации, см. раздел 3).

Эталонная среда развёртывания: операционная система Astra Linux 1.8 (на базе Debian 12), Docker 25.0.5, Docker Compose 2.20.2; сервер с 4 vCPU, 4 ГБ оперативной памяти и 80 ГБ дискового пространства. Внешние подключения по HTTPS обслуживает веб-сервер nginx.

Консультации по установке, развёртыванию и эксплуатации программы предоставляют: Староквашев Денис Валерьевич — генеральный директор, разработчик программы (серверная и клиентская части), e-mail: starokvashev.denis@gmail.com; служба технической поддержки ООО «Партнер Лидс» — e-mail: Partner-Leeds@yandex.com, телефон: +7 (966) 100-50-22, режим работы: рабочие дни с 8:00 до 17:00 (МСК).

2. СТРУКТУРА ПРОГРАММЫ

Программа состоит из следующих составных частей, разворачиваемых как отдельные контейнеры и связанных между собой сетью Docker Compose:

— postgres — система управления базами данных PostgreSQL 16.15 с двумя базами данных (partner-leeds-sso и partner-leeds-panel), хранит данные платформы (учётные записи, потоки, идентификации, финансовые операции);

— redis (Redis 7.2.11) — используется для кэширования и вспомогательных операций;

— sso — сервис авторизации и аутентификации, предоставляет HTTP-шлюз (gateway) и gRPC-интерфейс для проверки токенов доступа другими сервисами;

— panel — сервис прикладной логики (API), обслуживающий личный кабинет покупателя и административную панель; обращается к sso для проверки авторизации запросов.

Авторизация пользователя выполняется через HTTP-шлюз sso (эндпоинт /auth/login), который выдаёт токен доступа. Все последующие обращения к сервису panel выполняются с указанием этого токена. Для работы sso используется пара криптографических ключей (JWT), формируемая отдельным шагом настройки (см. 3.3).

Файлы, прикладываемые пользователями (в частности, карточка компании, доступная для скачивания в личном кабинете), хранятся в отдельном каталоге данных на файловой системе сервера.

3. НАСТРОЙКА ПРОГРАММЫ

Настоящий раздел описывает порядок первоначального развёртывания программы средствами прилагаемого сценария автоматизированного развёртывания.

3.1. Предварительные условия

Перед началом установки необходимо убедиться, что на сервере установлены Docker и плагин Docker Compose (команда docker compose version должна выполняться без ошибок).

3.2. Конфигурация окружения

Каталог развёртывания (deploy/) содержит эталонный файл переменных окружения .env.example. При первом запуске сценарий автоматически создаёт из него рабочий файл .env, если он ещё не создан.

Перед использованием в продуктивном контуре необходимо вручную отредактировать файл .env и заменить как минимум:

— учётные данные доступа к базе данных (POSTGRES_USER и связанные параметры);

— учётные данные, используемые по умолчанию для входа в систему (в контрольном контуре — admin / password);

— значения сетевых портов (HOST_GATEWAY_PORT, HOST_PANEL_PORT, HOST_SSO_GRPC_PORT) — при необходимости привести в соответствие с инфраструктурой продуктивного контура;

— иные секреты и параметры подключения, предусмотренные файлом .env.example.

3.3. Генерация ключей авторизации

Ключи JWT, используемые сервисом sso для подписи и проверки токенов доступа, формируются отдельным шагом (сценарий generate-jwt.sh) и не входят в состав файла конфигурации — генерируются локально при развёртывании.

3.4. Запуск и развёртывание

Развёртывание выполняется запуском сценария из корневого каталога проекта:

```
./deploy.sh up
```

Сценарий последовательно выполняет: подготовку файла конфигурации, генерацию ключей авторизации, подготовку каталога хранения файлов, сборку и запуск контейнеров (docker compose up -d --build), ожидание готовности PostgreSQL и Redis, применение SQL-миграций к базе данных.

После успешного завершения на экран выводятся адреса развёрнутых сервисов и данные для входа в контрольном контуре.

3.5. Управление развёрнутой программой

Для последующего управления развёрнутой программой предусмотрены отдельные команды сценария:

— ./deploy.sh down — остановка всех контейнеров программы;

— ./deploy.sh rebuild — пересборка образов и перезапуск с повторным применением миграций;

— ./deploy.sh migrate — применение SQL-миграций без пересборки и перезапуска контейнеров;

— ./deploy.sh jwt — повторная генерация ключей авторизации.

4. ПРОВЕРКА ПРОГРАММЫ

Для проверки работоспособности развёрнутой программы предусмотрена команда:

```
./deploy.sh smoke
```

Команда выполняет последовательную проверку:

1. Запрос авторизации к HTTP-шлюзу sso (эндпоинт /auth/login) с контрольными учётными данными и получение токена доступа.
2. Обращение к сервису panel (эндпоинт /flow-guides) с полученным токеном — ожидается успешный ответ (HTTP 200).
3. Обращение к тому же эндпоинту сервиса panel без токена авторизации — ожидается отказ в доступе (HTTP 401), что подтверждает корректность работы механизма авторизации.

Положительным результатом проверки считается получение токена доступа на первом шаге, успешный ответ на втором шаге и отказ в доступе (401) на третьем шаге. Отклонение от этого результата указывает на ошибку развёртывания — действия при типовых ошибках приведены в разделе 6.

Дополнительно состояние контейнеров можно проверить командой ./deploy.sh status (аналог docker compose ps), а журналы работы отдельного сервиса — командой ./deploy.sh logs [имя сервиса].

5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Сценарий развёртывания поддерживает точечное применение миграций базы данных без остановки и пересборки контейнеров (команда migrate) — используется при обновлении схемы базы данных в рамках сопровождения уже развёрнутой программы.

Предусмотрен раздельный просмотр журналов по каждому составному сервису (postgres, redis, sso, panel) — команда ./deploy.sh logs [имя сервиса], что упрощает диагностику при эксплуатации.

6. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

В настоящем разделе приведены тексты сообщений, выдаваемых в ходе настройки и проверки программы, условия их возникновения и действия, которые необходимо предпринять.

Текст сообщения	Условие возникновения	Действие системного программиста
ERROR: required command not found: docker	Не установлен Docker в системе развёртывания	Установить Docker согласно официальной инструкции для используемой ОС, повторить запуск
ERROR: docker compose plugin is required	Не установлен плагин Docker Compose (v2)	Установить docker-compose-plugin, повторить запуск
ERROR: missing .env.example	Отсутствует эталонный файл переменных окружения в каталоге deploy/	Проверить целостность поставки исходного кода/дистрибутива
Created .env from .env.example — change passwords before production use.	Файл .env создан автоматически при первом запуске из шаблона	Обязательно заменить пароли и секреты в .env на реальные значения перед использованием в продуктивном контуре
ERROR: infrastructure did not become ready in time	PostgreSQL и/или Redis не перешли в состояние healthy в течение 90 попыток (180 секунд)	Проверить логи контейнеров (./deploy.sh logs postgres, ./deploy.sh logs redis), проверить корректность параметров подключения в .env
WARN: no access_token — check migrations / sso-grpc logs	Проверочный запрос авторизации не вернул токен доступа при выполнении проверки (smoke)	Проверить, что миграции применены (./deploy.sh migrate) и что сервис sso-grpc запущен без ошибок (./deploy.sh logs sso)

УТВЕРЖДАЮ
Генеральный директор
_____ / Староквашев Д.В./
«____» _____ 2026 г.