Общество с ограниченной ответственностью «ПЛАЙНЕР»

Программа для ЭВМ

# «Плайнер» (Pliner)

Система автоматизированного планирования производства

## ИНСТРУКЦИЯ ПО УСТАНОВКЕ

Версия 1.0.0

г. Белгород

2026

## 1. Общие сведения

Инструкция описывает установку программы «Плайнер» версии 1.0.0 на сервер заказчика и её первичную настройку: учётную запись администратора и лимит пользователей по лицензии. Документ рассчитан на системного администратора, знакомого с Linux и Docker.

Установку выполняют специалисты ООО «ПЛАЙНЕР». Заказчик предоставляет сервер, отвечающий требованиям раздела 2, и доступ к нему по SSH на время установки. Программа работает на сервере заказчика в виде Docker-контейнеров. Образы контейнеров собираются на этом сервере из сборочного комплекта версии 1.0.0, который хранится в репозитории правообладателя.

Сборочный комплект состоит из четырёх компонентов:

| Каталог | Содержимое |
|---|---|
| `managementapi/` | Сервер приложения, СУБД PostgreSQL, Keycloak, RabbitMQ: `docker-compose.prod.yml`, `.env.example`, `db_init/`, `rabbitmq.conf`, скрипт настройки Keycloak |
| `plancalc/` | Сервис расчёта плана |
| `ShiftPlan/` | Сервис сменных заданий и заданий на снабжение |
| `pliner-front/` | Веб-интерфейс на nginx |

В каждом каталоге есть `Dockerfile.prod`, `docker-compose.prod.yml` и шаблон параметров `.env.example`. В тексте каталог установки обозначен `/opt/pliner`. Команды выполняются от пользователя root или через `sudo`.

## 2. Требования к среде

Полные требования приведены в разделе 6 документа «Описание функциональных характеристик». Минимум для установки:

| Параметр | Сервер | Рабочее место |
|---|---|---|
| Операционная система | Astra Linux SE 1.7+, РЕД ОС 7.3+, Debian 12, Ubuntu 22.04+ | Любая ОС с современным браузером |
| Процессор | от 8 ядер | от 2 ядер |
| Оперативная память | от 16 ГБ | от 4 ГБ |
| Диск | от 256 ГБ SSD | — |
| ПО | Docker Engine 24+, Docker Compose v2 | Яндекс Браузер, Chrome, Firefox, Edge или Safari последних версий |
| Сеть | порты 80 и 443 доступны с рабочих мест | доступ к серверу по HTTP(S) |

На время установки и обновления серверу нужен доступ в Интернет. Через него скачиваются сборочный комплект, базовые образы Docker и библиотеки для сборки. В процессе работы программы Интернет не требуется.

## 3. Подготовка сервера

На сервере нужны Docker Engine с плагином Compose, общая сеть контейнеров и каталоги для данных и журналов.

1. Установите Docker Engine и Docker Compose v2:
   - Astra Linux и РЕД ОС — из репозитория операционной системы по документации её разработчика;
   - Debian и Ubuntu — из официального репозитория Docker (пакеты `docker-ce` и `docker-compose-plugin`).
2. Включите службу и проверьте версии:

```
systemctl enable --now docker
docker --version          # 24 или новее
docker compose version    # v2.x
```

3. Создайте общую сеть, через которую компоненты обращаются друг к другу:

```
docker network create shared-net
```

4. Создайте каталоги данных и журналов:

```
mkdir -p /var/lib/pliner-postgres /var/lib/rabbitmq_data \
         /var/log/pliner/managementapi /var/log/pliner/plancalc
/var/log/pliner/shiftplan
```

5. Настройте межсетевой экран: разрешите доступ с рабочих мест только к портам 80 и 443. Служебные порты закройте для внешних адресов: 5432 (PostgreSQL), 5672 и 15672 (RabbitMQ), 8001 (сервер приложения), 8081 (Keycloak).
6. Получите сборочный комплект версии 1.0.0 из репозитория правообладателя. Вместо `<репозиторий>` подставьте его адрес:

```
apt install -y git    # в РЕД ОС: dnf install -y git
mkdir -p /opt/pliner && cd /opt/pliner
git clone --branch v1.0.0 --depth 1 <репозиторий>/managementapi.git managementapi
git clone --branch v1.0.0 --depth 1 <репозиторий>/plancalc.git      plancalc
git clone --branch v1.0.0 --depth 1 <репозиторий>/shiftplan.git     ShiftPlan
git clone --branch v1.0.0 --depth 1 <репозиторий>/pliner-front.git  pliner-front
```

## 4. Установка

Установка состоит из заполнения файлов `.env`, сборки образов и запуска четырёх групп контейнеров в заданном порядке.

### 4.1. Настройка параметров

В каждом каталоге компонента скопируйте шаблон параметров в рабочий файл:

```
cd /opt/pliner
for d in managementapi plancalc ShiftPlan pliner-front; do cp $d/.env.example
$d/.env; done
chmod 600 */.env
```

В файле `managementapi/.env` замените все значения `<CHANGE_ME…>`. Пароли и секреты удобно сгенерировать командой `openssl rand -base64 24`.

| Переменная | Значение |
|---|---|
| `POSTGRES_USER`, `POSTGRES_PASSWORD` | Владелец базы данных Плайнера; пароль не короче 20 символов |
| `KEYCLOAK_ADMIN`, `KEYCLOAK_ADMIN_PASSWORD` | Служебный администратор консоли Keycloak (не пользователь Плайнера) |
| `KEYCLOAK_CLIENT_SECRET` | Секрет клиента `pliner-backend`, не короче 32 символов |
| `KC_DB_USERNAME`, `KC_DB_PASSWORD` | Учётная запись Keycloak в PostgreSQL: имя `keycloak`, пароль тот же, что в `db_init/01_keycloak_setup.sql` |
| `FIRST_ADMIN_EMAIL`, `FIRST_ADMIN_PASSWORD` | Первый администратор Плайнера, см. раздел 6 |
| `RABBITMQ_USER`, `RABBITMQ_PASSWORD`, `RABBITMQ_DEFAULT_USER`, `RABBITMQ_DEFAULT_PASS` | Учётная запись брокера; пары `USER`/`DEFAULT_USER` и `PASSWORD`/`DEFAULT_PASS` совпадают |
| `FRONTEND_URL`, `BACKEND_URL` | Адрес сервера, по которому пользователи открывают систему, например `http://pliner.factory.local` |
| `AUTH_COOKIE_SECURE` | `true` при работе по HTTPS; `false`, если система открывается по HTTP |
| `APP_TIMEZONE` | Часовой пояс предприятия, по умолчанию `Europe/Moscow` |
| `PRODUCTION_FUND_HOURS` | Длительность смены в часах (100 % загрузки на странице «Производство») |

В файле `pliner-front/.env` задайте `PORT=80` и `VITE_API_URL` — адрес сервера приложения, по которому к нему обращается браузер, например `http://pliner.factory.local/api/`. Этот адрес встраивается в веб-интерфейс при сборке. Если адрес сервера изменится, веб-интерфейс нужно пересобрать. В `plancalc/.env` и `ShiftPlan/.env` значения по умолчанию подходят для большинства установок. На сервере с 16 ГБ памяти оставьте `GA_PARALLEL_WORKERS=2`.

### 4.2. Сборка и запуск

Запускайте компоненты строго в этом порядке. Ключ `--build` собирает образы перед запуском:

```
cd /opt/pliner/managementapi && docker compose -f docker-compose.prod.yml up -d --
build
cd /opt/pliner/plancalc      && docker compose -f docker-compose.prod.yml up -d --
build
cd /opt/pliner/ShiftPlan     && docker compose -f docker-compose.prod.yml up -d --
build
cd /opt/pliner/pliner-front  && docker compose -f docker-compose.prod.yml up -d --
build
```

Первая команда запускает PostgreSQL, RabbitMQ и Keycloak. Затем она создаёт базу `keycloak_db`, применяет миграции базы Плайнера и настраивает Keycloak (раздел 5). Когда всё это завершится, стартует сервер приложения. Состояние контейнеров показывает команда `docker compose -f docker-compose.prod.yml ps`.

## 5. Первичная настройка Keycloak

Keycloak настраивается автоматически при первом запуске. Ручные действия нужны только для проверки. Настройку выполняет разовый контейнер `keycloak-init`: он дожидается готовности Keycloak и создаёт следующее.

| Объект | Значение |
|---|---|
| Область (realm) | `pliner-realm`; самостоятельная регистрация отключена, пользователи добавляются только по приглашению |
| Роли | `system_admin` — администратор системы, `manager` — менеджер, `operator` — оператор |
| Клиент `pliner-react` | Публичный клиент веб-интерфейса, Authorization Code + PKCE (S256) |
| Клиент `pliner-backend` | Конфиденциальный клиент сервера приложения, секрет из `KEYCLOAK_CLIENT_SECRET` |
| Парольная политика | Не менее 8 символов, минимум одна цифра, одна строчная и одна заглавная буква |
| Защита от подбора | Блокировка после 5 неудачных попыток входа |
| Сессия | Токен доступа 5 минут, максимальная сессия 10 часов |
| Первый администратор | Учётная запись из `FIRST_ADMIN_EMAIL` с ролью `system_admin`, см. раздел 6 |

Чтобы убедиться, что настройка прошла, проверьте журнал контейнера. В конце должно быть сообщение `Configuration complete`:

```
docker logs keycloak-init
```

Если сообщения нет, повторите настройку. Скрипт не создаёт уже существующие объекты повторно:

```
cd /opt/pliner/managementapi && docker compose -f docker-compose.prod.yml restart
keycloak-init
```

Консоль администратора Keycloak находится по адресу `http://<сервер>:8081/admin`. Вход — под `KEYCLOAK_ADMIN` / `KEYCLOAK_ADMIN_PASSWORD`. Порт 8081 закрыт для рабочих мест (раздел 3, шаг 5), поэтому

открывайте консоль с самого сервера или через SSH-туннель: `ssh -L 8081:localhost:8081 <сервер>`. В консоли можно подключить LDAP или Active Directory: Realm settings → User federation.

## 6. Первый администратор и пользователи

Первый администратор создаётся автоматически из переменных `FIRST_ADMIN_EMAIL` и `FIRST_ADMIN_PASSWORD`. Остальных пользователей он добавляет по приглашениям. Почтовый сервер для этого не нужен.

### 6.1. Первый вход

1. Откройте в браузере адрес из `FRONTEND_URL`, например `http://pliner.factory.local`.
2. Введите значения `FIRST_ADMIN_EMAIL` и `FIRST_ADMIN_PASSWORD`.
3. Смените пароль администратора. Для этого в консоли Keycloak выберите область `pliner-realm` → Users → учётная запись администратора → Credentials → Reset password.
4. Удалите строку `FIRST_ADMIN_PASSWORD` из `managementapi/.env`. После создания учётной записи она больше не используется.

### 6.2. Добавление пользователей

1. Откройте окно профиля и перейдите на вкладку «Управление пользователями». Вкладка видна только роли «Администратор системы».
2. Создайте приглашение: укажите имя пользователя или электронную почту и роль — «Администратор системы», «Менеджер» или «Оператор».
3. Скопируйте ссылку-приглашение и передайте её сотруднику.
4. Сотрудник открывает ссылку и задаёт пароль. После этого он входит в систему с правами своей роли.

Неиспользованные приглашения можно отозвать на вкладке «Приглашения». Новое приглашение не создаётся, если достигнут лимит пользователей по лицензии (раздел 7).

## 7. Лицензия

Право использования программы предоставляется по лицензионному договору с ООО «ПЛАЙНЕР». Ключ активации и подключение к серверу лицензий не нужны. Договор определяет максимальное число пользователей. Это число хранится в базе данных и проверяется при создании каждого приглашения.

- По умолчанию лимит — 50 пользователей. Допустимые значения — от 1 до 10 000.
- Если снизить лимит ниже текущего числа пользователей, существующие учётные записи останутся. Новые приглашения создать будет нельзя.

– Лимит устанавливает администратор системы через программный интерфейс сервера приложения.

Команды выполняются на сервере. Вместо `<email>` и `<пароль>` подставьте данные администратора, вместо `100` — число пользователей по договору.

```
# получить токен администратора
TOKEN=$(curl -s -X POST http://localhost:8001/auth/login \
  -H 'Content-Type: application/json' \
  -d '{"email":"<email>","password":"<пароль>"}' \
  | python3 -c 'import sys,json; print(json.load(sys.stdin)["access_token"])')

# установить лимит
curl -X PUT http://localhost:8001/license \
  -H "Authorization: Bearer $TOKEN" -H 'Content-Type: application/json' \
  -d '{"max_users":100}'

# проверить: лимит, число пользователей, свободные места
curl -H "Authorization: Bearer $TOKEN" http://localhost:8001/license
```

# 8. Проверка и обслуживание

## 8.1. Проверка работоспособности

Установка завершена, если выполнены три условия:

1. Команда `docker ps -a` показывает контейнеры `pliner_db`, `rabbitmq`, `keycloak`, `management_api`, `pliner-plancalc-api`, `pliner-shiftplan`, `pliner-front` в состоянии Up. Разовые контейнеры `management_api_migrator` и `keycloak-init` завершены с кодом 0 (Exited (0)).
2. Команда `curl http://localhost:8001/health` возвращает ответ с кодом 200.
3. Вход под первым администратором проходит (раздел 6.1), и открывается раздел «Справочники».

Журналы сервисов записываются в `/var/log/pliner/` и хранятся 30 дней. Срок хранения задаёт переменная `LOG_RETENTION_DAYS`. Журнал любого контейнера выводит команда `docker logs <имя>`.

## 8.2. Резервное копирование

Все данные хранятся в двух базах PostgreSQL: `Pliner` (данные приложения) и `keycloak_db` (пользователи и роли). Рекомендуется делать ежедневную копию обеих баз и файлов `.env`. В командах ниже `<user>` — значение `POSTGRES_USER`.

```
# создание копии
docker exec pliner_db pg_dump -U <user> -Fc Pliner      > pliner_$(date +%F).dump
docker exec pliner_db pg_dump -U <user> -Fc keycloak_db > keycloak_$(date +%F).dump

# восстановление (при остановленных management_api и keycloak)
docker exec -i pliner_db pg_restore -U <user> -d Pliner --clean < pliner_2026-10-
01.dump
```

## 8.3. Обновление

Обновление выполняют специалисты ООО «ПЛАЙНЕР».

1. Сделайте резервную копию (раздел 8.2).
2. В каталоге каждого компонента переключитесь на новую версию, например `git fetch --tags && git checkout v1.1.0`. Файлы `.env` при этом не меняются. Сверьте их с новыми `.env.example`.
3. Повторите команды из раздела 4.2. Образы пересоберутся, миграции базы данных применятся автоматически.

### 8.4. Удаление

Выполните `docker compose -f docker-compose.prod.yml down` в каждом каталоге компонента в обратном порядке, затем `docker network rm shared-net`. Данные остаются в `/var/lib/pliner-postgres` и `/var/lib/rabbitmq_data`. Если они больше не нужны, удалите эти каталоги вместе с `/opt/pliner` и `/var/log/pliner`.

### 8.5. Типовые неполадки

| Признак | Вероятная причина | Действие |
|---|---|---|
| `docker compose` сообщает `network shared-net declared as external, but could not be found` | Не создана общая сеть | Выполните шаг 3 раздела 3 |
| `keycloak` не переходит в состояние healthy | Пароль `KC_DB_PASSWORD` не совпадает с паролем в `db_init/01_keycloak_setup.sql` | Сравните пароли, см. `docker logs keycloak` |
| `management_api_migrator` завершился с ошибкой | Неверные `POSTGRES_USER` / `POSTGRES_PASSWORD` | `docker logs management_api_migrator`, исправьте `.env`, повторите запуск |
| Страница входа открывается, но после входа сессия сразу завершается | `AUTH_COOKIE_SECURE=true` при работе по HTTP | Настройте HTTPS или установите `false` и перезапустите `management_api` |
| Расчёт плана не завершается, в `docker inspect pliner-plancalc-api` видно `OOMKilled: true` | Не хватает памяти для расчёта | Уменьшите `GA_PARALLEL_WORKERS` в `plancalc/.env` или добавьте память серверу |
| Не создаётся приглашение | Достигнут лимит пользователей | Проверьте лимит (раздел 7) |