Общество с ограниченной ответственностью «МИЛЭНД»
ОГРН 1185074015884, ИНН 5036174419

ПРОГРАММА ДЛЯ ЭВМ

«Оборот»

Система прогнозирования спроса и управления закупками

Версия 0.12.4

Инструкция по установке

экземпляра программного обеспечения, предоставленного для проведения экспертной проверки

Правообладатель — ООО «МИЛЭНД»

2026

«Оборот»: установка и обслуживание сервера

Инструкция для администратора сервера, на который ставится «Оборот», — система прогнозирования спроса и управления закупками. Правообладатель — ООО «МИЛЭНД». Инструкция относится к версии 0.12.3.

Установка выполняется одной командой из архива выпуска. Скрипт сам ставит пакеты ОС, создаёт базу данных, генерирует пароли, собирает интерфейс, настраивает веб-сервер и включает ежедневное резервное копирование.

Содержание:

1. Что входит в установку
2. Требования к серверу
3. Подготовка
4. Установка
5. Первый вход
6. Настройки
7. Почта
8. Резервные копии
9. Обновление
10. Повседневное управление
11. Где что лежит
12. Безопасность
13. Неполадки
14. Удаление

1. Что входит в установку

Компонент	Что делает	Как запущен
Веб-сервер nginx	Отдаёт интерфейс, передаёт запросы `/api/` бэкенду, держит HTTPS	системный сервис `nginx`
Бэкенд (FastAPI, uvicorn)	Программный интерфейс: вход, справочники, расчёты, выгрузки	`oborot-backend`, порт 8000 только на 127.0.0.1
Обработчик фоновых задач (Celery)	Загрузка файлов импорта	`oborot-worker`
PostgreSQL	База данных `oborot`	системный сервис `postgresql`, только локально
Redis	Очередь фоновых задач, с паролем	системный сервис `redis-server`, только локально
Резервное копирование	Копия базы каждую ночь, проверка восстановления раз в неделю	таймеры `oborot-backup.timer`, `oborot-restore-check.timer`

Снаружи открыты только порты 80 и 443 веб-сервера. Бэкенд, база и Redis слушают только 127.0.0.1.

2. Требования к серверу

Операционная система — одна из:

ОС	Состояние
Ubuntu 24.04 LTS	проверено
Debian 12	проверено
Ubuntu 22.04 LTS	рассчитано, не проверялось

ОС	Состояние
Astra Linux на базе Debian	рассчитано, не проверялось

«Проверено» — установка, работа, перезагрузка и восстановление из копии пройдены на чистой системе. Другие дистрибутивы семейства Debian скрипт тоже примет, но на них установка не проверялась. Нужен Python 3.10 или новее из пакетов ОС.

Ресурсы:

	Минимум	Рекомендуется
Процессор	2 ядра	4 ядра
Память	2 ГБ	4 ГБ
Диск	10 ГБ	20 ГБ и больше
Архитектура	x86_64 или arm64	

Установка занимает на диске около 1,5 ГБ вместе с пакетами ОС. В работе бэкенд и обработчик задач занимают около 700 МБ памяти, PostgreSQL, Redis и nginx — ещё около 150 МБ (замер на демо-данных: 25 товаров, 8 тыс. строк продаж). При 2 ГБ памяти можно уменьшить `BACKEND_WORKERS` и `WORKER_CONCURRENCY` до 1 (раздел 6) — это освобождает около 250 МБ.

Место на диске растёт вместе с историей продаж. В `/var/backups/oborot` хранится до 22 копий базы, а еженедельной проверке восстановления нужно свободное место, сравнимое с размером самой базы.

Сеть:

- доменное имя с A-записью на IP сервера (для HTTPS — обязательно);
- входящие порты 80 и 443;
- на время установки и обновления — исходящий доступ в интернет к репозиториям пакетов ОС и к адресам из таблицы ниже.

Адрес	Зачем
репозитории пакетов ОС (`deb.debian.org`, `archive.ubuntu.com` или зеркало)	PostgreSQL, Redis, nginx, Python
`nodejs.org`	Node.js для сборки интерфейса, если в системе нет подходящей версии
`pypi.org`, `files.pythonhosted.org`	библиотеки бэкенда
`registry.npmjs.org`	библиотеки интерфейса
`acme-v02.api.letsencrypt.org`	сертификат HTTPS, только с `--tls`

В работе продукт во внешние сервисы не обращается; исключение — почтовый сервер, если почта настроена (раздел 7), и внешний хост для копий, если он задан (раздел 8).

3. Подготовка

1. Сервер с одной из ОС выше, доступ root (или пользователь с `sudo`).
2. A-запись домена указывает на IP сервера. Проверка: `getent hosts oborot.example.ru` должна вернуть IP этого сервера.
3. Порты 80 и 443 открыты в межсетевом экране сервера и у хостинга. Если на сервере включён `ufw`:

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

4. На сервере нет другого веб-сайта на порту 80 с тем же доменом. Если nginx уже обслуживает другие сайты, они не затрагиваются: «Оборот» добавляет свой файл конфигурации `oborot.conf` и отключает только стандартную заглушку пакета nginx

(`sites-enabled/default`).

4. Установка

4.1. Архив выпуска

Архив выпуска — файл `oborot-<версия>.tar.gz`, рядом с ним — файл контрольной суммы `oborot-<версия>.tar.gz.sha256`. Скопируйте оба на сервер и проверьте, что архив дошёл целым:

```
sha256sum -c oborot-0.12.3.tar.gz.sha256
# oborot-0.12.3.tar.gz: OK
tar -xzf oborot-0.12.3.tar.gz
cd oborot-0.12.3
```

Архив собирает поставщик командой `deploy/make_release.sh` из репозитория исходного кода.

4.2. Запуск установки

```
sudo ./deploy/install.sh --domain oborot.example.ru --admin-email admin@example.ru --tls
```

Параметр	Значение
`--domain ИМЯ`	доменное имя, по которому продукт будет открываться (обязателен при первой установке)
`--tls`	получить сертификат Let's Encrypt и включить HTTPS; без него продукт работает по HTTP
`--admin-email АДРЕС`	создать первого администратора; этот же адрес передаётся Let's Encrypt
`--org-name НАЗВАНИЕ`	название организации администратора; по умолчанию — домен
`--demo`	загрузить демо-организацию с данными для показа
`--no-systemd`	только для проверки в контейнере без systemd

Установка занимает около двух минут; большая часть — скачивание пакетов и сборка интерфейса. По шагам скрипт:

1. ставит пакеты ОС: PostgreSQL, Redis, nginx, Python, certbot (с `--tls`);
2. если в системе нет Node.js 20.19+ или 22.12+, скачивает его с `nodejs.org` в `/opt/oborot/node` и сверяет контрольную сумму SHA-256;
3. создаёт системного пользователя `oborot` без входа в систему;
4. копирует код в `/opt/oborot/app`, ставит библиотеки бэкенда в `/opt/oborot/venv`;
5. создаёт файл настроек `/etc/oborot/oborot.env` со случайными паролями базы, Redis и ключом подписи сеансов;
6. создаёт роль и базу `oborot` в PostgreSQL, задаёт пароль Redis;
7. применяет миграции схемы базы;
8. создаёт администратора (`--admin-email`) и демо-данные (`--demo`);
9. собирает интерфейс и кладёт его в `/var/www/oborot`;
10. пишет конфигурацию nginx и, с `--tls`, получает сертификат;
11. ставит и запускает сервисы `oborot-backend`, `oborot-worker` и таймеры резервного копирования, проверяет, что бэкенд отвечает через nginx.

Пароль администратора скрипт печатает на шаге «Администратор», адрес продукта — в конце вывода.

Повторный запуск безопасен. Пароли в `/etc/oborot/oborot.env` не меняются, база не пересоздаётся, администратор и демо-данные не дублируются. Если установка прервалась (обрыв связи, нехватка места), устраните причину и запустите ту же команду ещё раз.

4.3. HTTPS позже

Если при установке домен ещё не указывал на сервер, ставьте без `--tls`, а когда A-запись заработает, запустите ту же команду с `--tls`:

```
sudo /opt/oborot/app/deploy/install.sh --domain oborot.example.ru --tls
```

Сертификат продлевается сам (таймер `certbot.timer` пакета certbot), после продления nginx перечитывает его.

5. Первый вход

Откройте в браузере адрес, напечатанный в конце установки, например `https://oborot.example.ru/`, и войдите с адресом администратора и паролем.

Пароль администратора печатается один раз, в базе хранится только его хеш. Сохраните его в менеджере паролей. Если он потерян, восстановите его ссылкой «Забыли пароль?» на странице входа: без настроенной почты ссылка пишется в журнал сервиса (раздел 7).

Если установка шла без `--admin-email`, администратора можно создать позже:

```
sudo oborot-manage python scripts/create_admin.py --email admin@example.ru --organization "ООО Ромашка"
```

С `--demo` создаётся отдельная организация «Демо — Оборот» с двумя пользователями (`demo-admin@example.com` — администратор, `demo-viewer@example.com` — только просмотр). Их общий пароль генерируется при установке и печатается в выводе скрипта. Демо-данные не смешиваются с данными вашей организации.

Экземпляр для ознакомления и экспертной проверки. Чтобы показать продукт на готовых данных, ставьте его с `--demo` — например, `sudo ./deploy/install.sh --domain oborot.example.ru --admin-email admin@example.ru --tls --demo`. Проверяющему передаются адрес продукта, адрес `demo-admin@example.com` и пароль из вывода скрипта: под ним видны все экраны, включая загрузку данных и настройки. Пароль показа можно сменить ссылкой «Забыли пароль?» (раздел 7). Чтобы проверить загрузку файлов, удобно взять шаблоны из каталога `docs/templates/` архива: справочник товаров, продажи, остатки.

Дальше работа идёт в интерфейсе: сотрудников приглашает администратор организации, данные загружаются выгрузками из учётной системы. Порядок работы описан в руководстве пользователя (`docs/UserGuide.md`).

6. Настройки

Все настройки — в файле `/etc/oborot/oborot.env` (права 600, владелец root). После правки перезапустите сервисы:

```
sudo systemctl restart oborot-backend oborot-worker
```

Значения с пробелами и спецсимволами записываются в одинарных кавычках.

Ключ	По умолчанию	Что задаёт
`SMTP_HOST`, `SMTP_PORT`, `SMTP_USER`, `SMTP_PASSWORD`, `SMTP_FROM`, `SMTP_FROM_NAME`, `SMTP_STARTTLS`	пусто, 587	почта, раздел 7
`MAX_UPLOAD_MB`	32	наибольший размер загружаемого файла, МБ
`BACKEND_WORKERS`	2	число процессов бэкенда
`WORKER_CONCURRENCY`	2	сколько файлов импорта обрабатывается одновременно
`JWT_EXPIRES_SECONDS`	3600	время жизни сеанса, секунды
`TRIAL_DAYS`	2	срок пробного периода организации, зарегистрированной через форму, дни; 0 — без срока
`TRIAL_MAX_PRODUCTS`, `TRIAL_MAX_WAREHOUSES`, `TRIAL_MAX_USERS`	10000, 1, 2	лимиты на пробный период (как у тарифа «Старт»); 0 — без ограничения
`BACKUP_KEEP_DAILY`, `BACKUP_KEEP_WEEKLY`	14, 8	сколько хранить ежедневных и недельных копий

Ключ	По умолчанию	Что задаёт
BACKUP_MAX_AGE_HOURS	48	копия старше этого срока считается сбоем копирования
BACKUP_REMOTE_TARGET, BACKUP_REMOTE_SSH_KEY, BACKUP_REMOTE_METHOD	пусто, пусто, rsync	копия на внешний хост, раздел 8

Ключи, которые задаёт сам скрипт установки, вручную не меняют:

- OBOROT_DOMAIN, PUBLIC_BASE_URL, CORS_ORIGINS — производные от домена. Чтобы сменить домен, запустите install.sh с новым --domain.
- MAX_UPLOAD_MB после правки требует ещё и перезапуска install.sh: он же задаёт предел размера запроса в nginx.
- DATABASE_URL, REDIS_PASSWORD, REDIS_URL, JWT_SECRET — пароли и ключи. Как их сменить — раздел 12.

7. Почта

Почта нужна для двух писем: восстановление пароля и приглашение сотрудника. Пока SMTP_HOST и SMTP_FROM пустые, письма не уходят, а их ссылки пишутся в журнал бэкенда. Взять ссылку из журнала:

```
sudo journalctl -u oborot-backend | grep -oE 'https?://[^ ]*(reset-password|accept-invite)[^ ]*' | tail
   -n 1
```

Чтобы письма уходили, впишите в /etc/oborot/oborot.env параметры почтового ящика, с которого они будут отправляться, и перезапустите сервисы:

```
SMTP_HOST=smtp.example.ru
SMTP_PORT=587
SMTP_USER=oborot@example.ru
SMTP_PASSWORD='пароль ящика'
SMTP_FROM=oborot@example.ru
SMTP_FROM_NAME=Оборот
SMTP_STARTTLS=true
```

Исходящий порт 587 у хостинга должен быть открыт: некоторые провайдеры закрывают его по умолчанию.

8. Резервные копии

Что делается само:

- каждую ночь около 02:30 (время сервера) — копия базы pg_dump в /var/backups/oborot/daily/, рядом — файл .counts с числом строк каждой таблицы на момент копии;
- раз в неделю одна из ночных копий сохраняется в /var/backups/oborot/weekly/;
- хранятся 14 ежедневных и 8 недельных копий, более старые удаляются;
- по воскресеньям около 05:00 — проверка восстановления: свежая копия разворачивается во временную базу, числа строк сверяются с записанными при копировании, временная база удаляется. Проверка также считает сбоем копию старше 48 часов: значит, ночное копирование не работает;
- перед каждым обновлением, меняющим схему базы, — копия с меткой pre-update.

Если сервер был выключен в момент копирования, копия делается сразу после включения.

Журнал копирования и проверок — /var/log/oborot/backup.log, а также journalctl -u oborot-backup -u oborot-restore-check. Когда были последние запуски и когда следующие:

```
systemctl list-timers 'oborot-*'
```

Вручную:

```
sudo /opt/oborot/app/deploy/backup.sh                # копия сейчас
sudo /opt/oborot/app/deploy/restore_check.sh         # проверить свежую копию
```

```
sudo /opt/oborot/app/deploy/restore_check.sh /var/backups/oborot/daily/oborot_20261010_023000.dump
```

Восстановление заменяет текущую базу содержимым копии. Перед заменой скрипт сам сохраняет текущую базу с меткой `pre-restore`, останавливает и затем запускает сервисы:

```
sudo /opt/oborot/app/deploy/restore.sh /var/backups/oborot/daily/oborot_20261010_023000.dump
# спросит имя базы для подтверждения: oborot
```

Копия, принесённая с другого сервера, должна лежать в `/var/backups/oborot/` (её должен читать пользователь `postgres`).

Копия на внешний хост. Копии, которые лежат только на этом же сервере, не спасут при его потере. Чтобы каждая ночная копия отправлялась на другой сервер по ssh:

1. Создайте ключ без пароля и разрешите его на внешнем хосте:

   ```
   sudo ssh-keygen -t ed25519 -N '' -f /etc/oborot/backup_key
   sudo cat /etc/oborot/backup_key.pub
   # строку — в ~/.ssh/authorized_keys пользователя на внешнем хосте
   ```

2. Впишите в `/etc/oborot/oborot.env`:

   ```
   BACKUP_REMOTE_TARGET=backup@backup.example.ru:/srv/backups/oborot
   BACKUP_REMOTE_SSH_KEY=/etc/oborot/backup_key
   BACKUP_REMOTE_METHOD=rsync
   ```

   `rsync` должен быть установлен и на внешнем хосте; если его там нет — `BACKUP_REMOTE_METHOD=scp`. Каталог на внешнем хосте создайте заранее.

3. Проверьте: `sudo /opt/oborot/app/deploy/backup.sh` должен закончиться строкой «отправлено на внешний хост». Отпечаток ключа внешнего хоста запоминается при первой отправке в `/etc/oborot/backup_known_hosts`.

Если отправка не удалась, локальная копия всё равно остаётся, а запуск `oborot-backup` завершается с ошибкой — это видно в `systemctl list-units --failed`.

9. Обновление

Новая версия ставится так же, как первая: распакуйте её архив и запустите `install.sh` из него.

```
sha256sum -c oborot-0.13.0.tar.gz.sha256
tar -xzf oborot-0.13.0.tar.gz
cd oborot-0.13.0
sudo ./deploy/install.sh
```

Домен и остальные настройки берутся из `/etc/oborot/oborot.env`, параметры повторять не нужно. Если новая версия меняет схему базы, перед миграциями скрипт снимает копию с меткой `pre-update`. Сервисы перезапускаются в конце, перерыв — несколько секунд, но пока идёт обновление (около минуты), интерфейс и бэкенд меняются не одновременно. Обновляйте вне рабочего времени пользователей.

Если после обновления что-то пошло не так, верните прежнюю версию: запустите `install.sh` из архива прежней версии и, если схема базы менялась, восстановите копию `pre-update` (раздел 8).

10. Повседневное управление

```
systemctl status oborot-backend oborot-worker        # состояние
sudo systemctl restart oborot-backend oborot-worker  # перезапуск
journalctl -u oborot-backend -u oborot-worker -f     # журналы в реальном времени
curl -s http://127.0.0.1:8000/health                 # бэкенд отвечает: {"status":"ok"}
```

Команды бэкенда выполняются через `oborot-manage` — от имени пользователя сервиса и с настройками из `/etc/oborot/oborot.env`:

```
sudo oborot-manage alembic current                   # ревизия схемы базы
sudo oborot-manage python scripts/create_admin.py --email ... --organization "..."
```

Сервисы запускаются сами при включении сервера и перезапускаются при сбое.

11. Где что лежит

Путь	Что
`/opt/oborot/app`	код продукта (только чтение для сервиса)
`/opt/oborot/venv`	библиотеки бэкенда
`/opt/oborot/node`	Node.js для сборки интерфейса (если скачивался)
`/etc/oborot/oborot.env`	настройки и пароли
`/var/lib/oborot/uploads`	загруженные файлы импорта
`/var/www/oborot`	собранный интерфейс
`/var/backups/oborot`	резервные копии базы
`/var/log/oborot/backup.log`	журнал копирования
`/etc/nginx/sites-available/oborot.conf` (или `/etc/nginx/conf.d/oborot.conf`)	конфигурация nginx; перезаписывается при каждом запуске `install.sh`
`/etc/systemd/system/oborot-*`	сервисы и таймеры
`/usr/local/sbin/oborot-manage`	команда управления

12. Безопасность

- Пароли базы и Redis и ключ подписи сеансов генерируются при установке и хранятся только в `/etc/oborot/oborot.env` с правами 600.
- Бэкенд и обработчик задач работают от пользователя `oborot` без права входа в систему, с ограничениями systemd: код и системные каталоги для них только для чтения, писать можно лишь в `/var/lib/oborot`, без повышения привилегий. Оценка `systemd-analyze security` для обоих сервисов — 2.9 по шкале от 0 до 10 (меньше — лучше), уровень «OK».
- PostgreSQL и Redis слушают только 127.0.0.1, у Redis пароль.
- В базе хранятся хеши паролей пользователей и хеши одноразовых ссылок из писем, а не сами пароли и ссылки.
- Вход ограничен по числу неудачных попыток с одного адреса: после пяти неудачных попыток за 5 минут вход с этого адреса закрывается на 10 минут. Регистрация новых организаций — не больше трёх в час с одного адреса.
- Резервные копии доступны только пользователям root и `postgres`.

Смена паролей и ключей. Ключ подписи сеансов:

```
sudo sed -i "s/^JWT_SECRET=.*/JWT_SECRET=$(openssl rand -hex 32)/" /etc/oborot/oborot.env
sudo systemctl restart oborot-backend oborot-worker   # все пользователи войдут заново
```

Пароли базы и Redis: впишите новые значения (только латинские буквы и цифры) в `DATABASE_URL` и `REDIS_PASSWORD` и запустите `install.sh` повторно — он приведёт роль базы, `redis.conf` и `REDIS_URL` в соответствие с файлом настроек и перезапустит сервисы.

13. Неполадки

Признак	Что делать
«адрес ... не подходит для входа»	для `--admin-email` нужен настоящий адрес: зарезервированные зоны (`.test`, `.local`, `.invalid`) не принимаются. Запустите установку снова с другим адресом
«сертификат не получен»	А-запись домена не указывает на сервер или порт 80 закрыт. Проверьте `getent hosts <домен>` и межсетевой экран, затем запустите установку ещё раз
«установка уже идёт в другом процессе»	дождитесь окончания первой установки; если её нет — `/run/oborot-install.lock` освободится после перезагрузки
«бэкенд не ответил на /health»	`journalctl -u oborot-backend -n 50` — причина в последних строках
«в redis.conf уже задан requirepass не этой установкой»	Redis на сервере уже используется с паролем. Впишите этот пароль в `REDIS_PASSWORD` в `/etc/oborot/oborot.env`, в `redis.conf` строкой выше `requirepass` добавьте комментарий из сообщения и запустите установку снова
страница не открывается, `/health` на 127.0.0.1:8000 отвечает	`sudo nginx -t`, затем `journalctl -u nginx`; проверьте порты 80/443
загрузка файла висит в статусе «в очереди»	не работает обработчик задач: `systemctl status oborot-worker`
в `systemctl list-units --failed` есть `oborot-backup` или `oborot-restore-check`	`tail -n 50 /var/log/oborot/backup.log` — в последних строках причина
письма не приходят	раздел 7; ошибка отправки видна в `journalctl -u oborot-backend`

14. Удаление

Удаление необратимо стирает базу. Сначала сохраните копию за пределы сервера.

```
sudo systemctl disable --now oborot-backend oborot-worker oborot-backup.timer
  oborot-restore-check.timer
sudo rm -f /etc/systemd/system/oborot-* /usr/local/sbin/oborot-manage
sudo systemctl daemon-reload
sudo rm -f /etc/nginx/sites-enabled/oborot.conf /etc/nginx/sites-available/oborot.conf
  /etc/nginx/conf.d/oborot.conf
sudo systemctl reload nginx
sudo -u postgres dropdb oborot
sudo -u postgres dropuser oborot
sudo rm -rf /opt/oborot /etc/oborot /var/lib/oborot /var/www/oborot /var/www/oborot-acme
  /var/log/oborot
sudo userdel oborot
# по желанию: копии /var/backups/oborot и строки «Оборота» в /etc/redis/redis.conf
```

Пакеты PostgreSQL, Redis и nginx не удаляются: ими могут пользоваться другие программы на сервере.