

Инструкция по установке программного обеспечения «TurbineCare AI»

Версия ПО: **1.0.0** · Правообладатель: ООО «ИНТЕБРИКС» (ИНН 9707059284) · Редакция документа: 18.09.2026

Инструкция по развёртыванию экземпляра в вычислительном контуре заказчика, в том числе в изолированном контуре без доступа в сеть Интернет: требования, состав дистрибутива, подготовка сервера, установка мастером и вручную, переменные настроек, первый вход, сертификат, резервное копирование, обновление, откат, эксплуатация, доступ к проверочному экземпляру.

Назначение документа

Документ описывает первую установку программного обеспечения «TurbineCare AI» (далее — комплекс, ПО) версии 1.0.0 на объекте, его настройку, обновление, откат и резервное копирование. Установка выполняется администратором площадки. Сеть общего пользования не требуется ни при установке, ни при работе: образы служб входят в архив дистрибутива, внешних обращений комплекс не делает. Номер версии единый для всей поставки и задаётся файлом `VERSION` в корне дистрибутива.

1. Системные требования

Параметр	Минимум	Рекомендуется
Операционная система	Astra Linux Special Edition 1.8 (испытания — 1.8.2, протокол № 1 от 11.09.2026), РЕД ОС 8 (испытания — 8.0.2, протокол № 2 от 11.09.2026); иной дистрибутив Linux x86_64 с Docker Engine 24 и новее допускается без протокола	Astra Linux SE 1.8.2 или РЕД ОС 8.0.2
Среда контейнеризации	Docker Engine 24 и новее и плагин Docker Compose 2.20 и новее из репозитория ОС (в испытаниях: Astra Linux SE 1.8.2 — Docker 25.0.5, Compose 2.20.2; РЕД ОС 8.0.2 — Docker 29.7.2, Compose 5.3.1)	—
СУБД	PostgreSQL версии 14 и новее (испытано на 15 и 16) либо Postgres Pro, доступная по сети с машины комплекса	отдельный сервер заказчика
Процессор	4 ядра (x86_64)	8 ядер
Оперативная память	8 ГБ	16 ГБ
Диск	50 ГБ SSD	200 ГБ SSD
Сеть	доступ к серверу OPC UA АСУ ТП; порты 80 и 443 к комплексу	—

Пределы памяти контейнеров заданы в файлах развёртывания: `docker-compose.prod.yml` — веб-приложение 3 ГБ, процесс сбора 1 ГБ, расчётная служба 512 МБ; `docker-compose.db.yml` — встроенная база 2 ГБ. Docker и Compose устанавливает администратор из репозитория операционной системы до запуска мастера: мастер их не ставит, а только проверяет. Пользователь, запускающий установку, должен входить в группу `docker`.

2. Состав поставки

Комплекс поставляется одним архивом `turbinecare-X.Y.Z.tar.gz`. Содержимое архива:

Файл или каталог	Назначение
<code>images/*.tar</code>	образы служб, по одному на тег; загружаются командой <code>docker load</code>
<code>docker-compose.prod.yml</code>	службы комплекса; база данных подключается снаружи, адрес — в переменной <code>DATABASE_URL</code>
<code>docker-compose.db.yml</code>	встроенная PostgreSQL для стенда или малой установки
<code>.env.example</code>	образец настроек; рабочий файл <code>.env</code> создаёт мастер установки
<code>nginx/nginx.conf</code>	настройка обратного прокси; каталог <code>nginx/certs/</code> для своего сертификата создаётся при установке (раздел 6)
<code>vendor/</code>	архив пакетов <code>rpm</code> и <code>PyPI</code> для повторной сборки образов без обращения к сети
<code>PLATFORM</code>	платформа, под которую собраны образы (<code>linux/amd64</code>)
<code>config/client-config.example.json</code>	образец описания установки для мастера
<code>scripts/setup-wizard.sh, backup.sh, update.sh, rollback.sh</code>	установка и обслуживание
<code>DEPLOY.md, CHANGELOG.md, VERSION, checksums.sha256</code>	руководство по развёртыванию, журнал изменений, версия, контрольные суммы всех файлов архива

Службы комплекса (видны командой `docker compose ps`):

Служба	Роль
<code>ops</code>	одноразовая: применяет миграции схемы при каждом запуске; тем же образом заводится администратор, применяется описание станции и выполняется пересчёт
<code>web</code>	веб-приложение и программный интерфейс
<code>collector</code>	сбор данных из АСУ ТП по OPC UA и обслуживание архива — отдельный процесс из образа приложения
<code>python-calc</code>	расчётная служба

Служба	Роль
db-backup	клиент pg_dump: дампы базы каждые BACKUP_INTERVAL_HOURS часов в каталог backups/auto/
nginx	обратный прокси и TLS; единственный вход снаружи — порты 80 и 443
postgres	только при встроенной базе, то есть при подключённом файле docker-compose.db.yml

База данных. Штатный режим — СУБД заказчика: PostgreSQL версии 14 и новее (испытано на 15 и 16) или Postgres Pro. Заказчик заранее заводит базу и роль-владельца, комплекс получает одну строку соединения DATABASE_URL. Встроенная база подключается вторым файлом развёртывания и предназначена для стенда, демонстрации и малой установки.

Экземпляр объекта (INSTANCE_ROLE=object) устанавливается перманентно: лицензионного ключа, срока действия и подписки не требуется. Учётной записи по умолчанию в поставке нет — администратор заводится при установке со случайным паролем, который показывается один раз.

3. Подготовка сервера

3.1. Среда контейнеризации

Docker Engine и плагин Docker Compose устанавливает администратор из репозитория целевой операционной системы её пакетным менеджером (apt в Astra Linux, dnf в РЕД ОС). Прослойка rodman-docker движком не является: на РЕД ОС ставится пакет docker-ce из репозитория дистрибутива. Наличие проверяется до установки комплекса:

```
docker --version
docker compose version
sudo usermod -aG docker $USER # затем переойти в систему
```

3.2. База данных

На сервере СУБД заказчика (PostgreSQL версии 14 и новее — испытано на 15 и 16 — либо Postgres Pro) заводится база и роль-владелец, роли выдаются права на эту базу, а на сервере комплекса проверяется сетевая доступность. Строка соединения понадобится на следующем шаге:

```
postgresql://пользователь:пароль@сервер:5432/turbinecare
```

Если встроенная база используется вместо СУБД заказчика (стенд, демонстрация, малая установка), в файл .env добавляется строка COMPOSE_FILE=docker-compose.prod.yml:docker-compose.db.yml, а строка соединения указывает на службу postgres.

3.3. Распаковка архива и проверка сумм

Архив распаковывается сразу в каталог, который станет каталогом установки:

```
tar -xzf turbinecare-1.0.0.tar.gz -C /opt
mv /opt/turbinecare-1.0.0 /opt/turbinecare
cd /opt/turbinecare
sha256sum -c checksums.sha256 --quiet
```

4. Установка

4.1. Мастер установки

```
./scripts/setup-wizard.sh # с вопросами на экране
./scripts/setup-wizard.sh --config client-config.json # без вопросов
./scripts/setup-wizard.sh --config client-config.json --dry-run #
только показать план
```

Мастер проверяет среду контейнеризации, читает параметры, записывает файл `.env` со сгенерированными секретами, описание установки `client-config.json` и описание станции `config/station_config.yaml`, загружает образы из каталога `images/` и проверяет, что все образы из файла развёртывания на месте, поднимает службы (миграции применяет служба `ops`), дожидается ответа `/api/health`, заводит организацию и машины из описания станции, заводит администратора и печатает его пароль один раз.

Файл для ключа `--config` составляется по образцу `config/client-config.example.json`. Мастер читает из него:

Ключ	Значение
<code>organization.name</code> , <code>organization.slug</code>	название организации и краткое имя латиницей
<code>admin.email</code>	почта администратора; та же величина задаётся ключом <code>--admin-email</code>
<code>database.url</code>	строка соединения с СУБД заказчика; пустое значение означает встроенную базу (ключ <code>--database-url</code>)
<code>turbines[]</code>	перечень машин: <code>id</code> , <code>name</code> , <code>model</code> , <code>norms.cycle</code> (<code>simple</code> — только ГТУ, <code>combined</code> — с котлом-утилизатором)
<code>deployment.mode</code>	<code>on-prem-isolated</code> (закрытый контур) либо <code>on-prem-connected</code>
<code>deployment.domain</code>	адрес, по которому открывают комплекс: имя узла или IP-адрес
<code>backup.autoIntervalHours</code> , <code>backup.retentionDays</code>	расписание и срок хранения автоматических дампов

4.2. Установка вручную

То же самое без мастера:

```
cp .env.example .env && chmod 600 .env
nano .env # DATABASE_URL, NEXTAUTH_SECRET, NEXTAUTH_URL,
          TLS_COMMON_NAME,
          # PYTHON_CALC_TOKEN, ENCRYPTION_KEY, CRON_SECRET; VERSION
уже проставлен
```

```
for f in images/*.tar; do docker load < "$f"; done
docker compose up -d # COMPOSE_FILE и COMPOSE_PROJECT_NAME читаются из
.env
docker compose ps # все службы healthy, ops завершилась с кодом 0
```

Описание станции — организация, площадки, машины, нормативы, тарифы, привязка тегов АСУ ТП:

```
docker compose run --rm -v "$PWD/config:/config:ro" ops \
node scripts/apply_station_config.cjs --file
/config/station_config.yaml
```

Первый администратор. Пароль генерируется случайно, 20 знаков, и выводится ровно один раз; в базе хранится только его хеш:

```
docker compose run --rm ops node scripts/ops/create-admin.cjs --email
вы@организация.ру
```

Демонстрационные данные в рабочую установку не заводятся: команда prisma db seed на объекте не выполняется — она создаёт пользователей с известными паролями. Учётной записи «по умолчанию» в поставке нет.

4.3. Переменные файла .env

Переменная	Назначение	Обязательность
DATABASE_URL	строка соединения с базой — единственный адрес базы для всех служб	обязательна
NEXTAUTH_SECRET	ключ подписи сессий	обязательна
NEXTAUTH_URL	внешний адрес, по которому открывают комплекс	обязательна
PYTHON_CALC_TOKEN	общий секрет приложения и расчётной службы	обязательна
ENCRYPTION_KEY	шифрование учётных данных OPC UA и внешнего каталога	обязательна
TLS_COMMON_NAME	имя узла в самоподписанном сертификате обратного прокси	обязательна
VERSION	тег образов; меняется сценарием обновления	проставлена в поставке
COMPOSE_PROJECT_NAME, COMPOSE_FILE	имя проекта и набор файлов развёртывания (со строкой :docker-compose.db.yml при встроенной базе)	проставлены в поставке
POSTGRES_USER,	только для встроенной базы	при

Переменная	Назначение	Обязательность
POSTGRES_PASSWORD, POSTGRES_DB		встроенно й базе
INSTANCE_ROLE	роль экземпляра: object — объект, operator — центр удалённого мониторинга	object по умолчанию
DEPLOYMENT_MODE	on-prem-isolated либо on-prem- connected	необязате льна
CRON_SECRET, RETENTION_TELEMETRY_DAYS, RETENTION_METRICS_DAYS, RETENTION_ALARMS_DAYS, RETENTION_AGGREGATES_DAYS	плановые задачи и сроки хранения (по умолчанию 90, 30, 365 и 1825 суток)	необязате льны
BACKUP_INTERVAL_HOURS, BACKUP_RETENTION_DAYS	расписание и срок хранения автоматических дампов	необязате льны
INGESTION_API_KEY	ключ приёма телеметрии по программному интерфейсу	при приёме по HTTP
OPCUA_ALLOW_INSECURE	снимает требование подписи и шифрования; допустима только на пусконаладке и снимается до сдачи	необязате льна
HTTP_PORT, HTTPS_PORT	порты обратного прокси наружу (по умолчанию 80 и 443)	необязате льны
VENDOR_LEGAL_NAME, VENDOR_CONTACT	правообладатель на экране «О программе»	необязате льны
SMTP_*, EMAIL_ALERT_RECIPIENTS	почтовые оповещения через сервер заказчика; в закрытом контуре не используются	необязате льны

Полный образец с пояснениями — файл `.env.example` в каталоге установки.

5. Первый вход

Комплекс открывается по адресу `https://<адрес узла>`. Порт 3000 наружу не публикуется: единственный вход — обратный прокси, порты 80 и 443, причём HTTP перенаправляется на HTTPS. После входа выданным временным паролем открывается экран смены пароля; до тех пор ни один раздел недоступен, и обойти экран набором адреса вручную нельзя. Затем в разделе администрирования настраивается подключение к АСУ ТП (экран «Приём данных»).

6. Сертификат TLS

При первом запуске обратный прокси создаёт самоподписанный сертификат с именем из `TLS_COMMON_NAME`, о котором браузер предупредит. Свой сертификат ставится так:

файлы fullchain.pem и privkey.pem кладутся в каталог nginx/certs/, после чего прокси перезапускается:

```
docker compose restart nginx
```

7. Резервное копирование

Автоматически дампы базы снимает служба db-backup — каждые BACKUP_INTERVAL_HOURS часов в каталог backups/auto/, удаляя копии старше BACKUP_RETENTION_DAYS суток. Служба работает и с внешней базой: это клиент pg_dump, обращающийся по той же строке соединения. Вручную:

```
scripts/backup.sh # дампы базы → backups/backup-ДАТА.dump
scripts/backup.sh --full # дополнительно .env, compose, nginx/,
config/
scripts/backup.sh --list # перечень копий, включая автоматические
scripts/backup.sh --export /media/usb # последняя копия на съёмный
носитель
scripts/backup.sh --restore backups/backup-20260901_120000.dump
```

Восстановление останавливает приложение, сносит схему и восстанавливает дампы с чистого листа, затем поднимает службы. Полная копия содержит файл .env с секретами и сертификаты — хранить её следует как секрет.

8. Обновление

Архив новой версии распаковывается рядом с установкой, а не поверх неё; сценарий запускается из нового архива с указанием каталога установки:

```
tar -xzf turbinecare-1.1.0.tar.gz -C /opt
/opt/turbinecare-1.1.0/scripts/update.sh --install-dir
/opt/turbinecare
/opt/turbinecare-1.1.0/scripts/update.sh --install-dir
/opt/turbinecare --dry-run
```

Сценарий проверяет среду, установку, архив и место на диске; снимает дампы базы и без него не продолжает; откладывает прежние файлы развёртывания; загружает образы и проверяет полноту набора; переносит новые файлы развёртывания, сценарии, настройку прокси, руководство, перечень изменений и версию; применяет миграции службой ops и поднимает службы; проверяет ответ /api/health и тег запущенного образа, а при отказе откатывается сам.

Не затрагиваются: .env, client-config.json, config/station_config.yaml, каталог nginx/certs/, каталог backups/ и тома данных.

9. Откат

```
/opt/turbinecare/scripts/rollback.sh # база из дампа, снятого перед
обновлением
/opt/turbinecare/scripts/rollback.sh --keep-db # только файлы и образы
```

Откат останавливает службы, возвращает файлы прежней версии и версию в .env, восстанавливает базу из дампа со сносом схемы, поднимает службы и проверяет ответ /api/health.

10. Эксплуатация

```
docker compose ps # состояние служб
docker compose logs -f web # журнал приложения (также collector,
python-calc, nginx)
docker compose restart web # перезапуск службы
docker compose stop / start # остановка и запуск; данные сохраняются
docker compose --profile seed run --rm seed-recalc # пересчёт после
смены нормативов
docker compose run --rm ops node scripts/ops/create-admin.cjs --email
... --force # сброс пароля
```

Автозапуск после перезагрузки сервера задан в файле развёртывания. Живость комплекса проверяется обращением к /api/health: без входа отдаётся только состояние (код ответа и признак ok — этого достаточно пробе живости контейнера и мониторингу); вошедшему пользователю — состояние базы, расчётной службы и процесса сбора, ход обслуживания архива и версия.

11. Доступ к проверочному экземпляру

11.1. Для экспертной проверки правообладатель предоставляет дистрибутив версии 1.0.0 для самостоятельного развёртывания по настоящей инструкции: архив `turbinecare-1.0.0.tar.gz` (около 1,4 ГБ; включает образы служб, файлы развёртывания, сценарии, документацию и архив пакетов), контрольная сумма SHA-256 — `b914262eeb6eb93ab61681a88f1c7d9e093e98512516db025241e3bbf1b61da7`. Адрес для получения архива: <https://turbinecare.ru/dist/turbinecare-1.0.0.tar.gz> (файл контрольной суммы — `turbinecare-1.0.0.tar.gz.sha256` по тому же адресу). Это та же сборка, что прошла приёмочные испытания по протоколам № 1 и № 2 от 11.09.2026.

11.2. Дополнительно развёрнут демонстрационный экземпляр с предзагруженными данными: <https://demo.turbinecare.ru>. Учётные записи для экспертов и порядок работы с экземпляром приведены в документе «Руководство по эксплуатации», раздел «Доступ для экспертизы: дистрибутив и демонстрационный экземпляр».

11.3. Технические специалисты правообладателя для вопросов по развёртыванию: электронная почта info@intebrix.ru, телефон +7 916 925-54-71, рабочие дни с 10:00 до 19:00 по московскому времени.