

Информация, необходимая для установки программного обеспечения «TradeCap»

Руководство по установке (развёртыванию) экземпляра программного обеспечения «TradeCap».
Документ содержит сведения, необходимые для самостоятельной установки ПО администратором организации-заказчика, и предназначен для системного администратора, обладающего правами суперпользователя на целевом сервере.

Версия ПО: 1.0 Редакция документа: 3.0 от «18» августа 2026 г.

1. Общие сведения

ПО «TradeCap» — серверное программное обеспечение, распространяемое **исключительно для установки в инфраструктуре заказчика (on-premise)**. Распространение по модели SaaS (интернет-сервиса) не предусмотрено.

ПО поставляется в виде установочного пакета формата `.deb` (Astra Linux) либо `.rpm` (RED OS). Установка выполняется штатными средствами пакетного менеджера операционной системы. В процессе работы ПО не обращается к внешним сетевым сервисам: проверка лицензии выполняется локально, серверов лицензирования не существует.

Установка может выполняться:

- **администратором организации-заказчика самостоятельно** по настоящему руководству;
- **инженером правообладателя** путём удалённого подключения к серверу заказчика по согласованному защищённому каналу (VPN, SSH или RDP) по тому же сценарию.

Ориентировочная продолжительность установки на подготовленном сервере — от 30 до 60 минут.

1.1. Порядок действий целиком

Полный цикл установки состоит из следующих этапов. Обратите внимание на порядок: идентификатор установки, необходимый для получения лицензии, становится известен только после создания базы данных.

1. Подготовка сервера и СУБД (разделы 5–6).
2. Проверка целостности и установка пакета (разделы 4, 7).
3. Настройка конфигурационного файла (раздел 8).
4. Первый запуск: применение миграций и старт службы (раздел 9).
5. Получение идентификатора установки и передача его правообладателю; установка полученного лицензионного ключа (раздел 10).
6. Создание учётной записи владельца и проверка работоспособности (разделы 11–12).

2. Требования к среде развёртывания

2.1. Требования к аппаратному обеспечению

Серверная часть ПО и СУБД предъявляют разные требования. Приложение не хранит данные в файловой системе: весь контент, включая загруженные наборы рыночных данных, размещается в базе данных. Дисковое пространство под данные требуется на сервере СУБД.

Сервер приложения:

Параметр	Минимальная конфигурация	Рекомендуемая конфигурация
Процессор	2 vCPU, x86_64	4 vCPU, x86_64
Оперативная память	4 ГБ	8 ГБ
Дисковое пространство	20 ГБ	40 ГБ
Сетевой интерфейс	100 Мбит/с	1 Гбит/с

Сервер СУБД (при размещении на отдельном узле; при совмещении требования суммируются):

Параметр	Минимальная конфигурация	Рекомендуемая конфигурация
Процессор	2 vCPU	4 vCPU
Оперативная память	4 ГБ	16 ГБ
Дисковое пространство	50 ГБ	500 ГБ, SSD

Объём дискового пространства на сервере СУБД определяется объёмом загружаемых рыночных данных и числом учебных материалов. Журналы приложения передаются в `journald` и учитываются в рамках системной политики ротации журналов.

2.2. Требования к программному обеспечению

Компонент	Поддерживаемые версии
Операционная система	Astra Linux Special Edition 1.8 и выше; RED OS 7.3 и выше. Архитектура x86_64
Система управления базами данных	PostgreSQL версии 15 и выше (рекомендуется 16 и выше) с пакетом расширений <code>contrib</code>
Система инициализации	<code>systemd</code>
Обратный прокси-сервер (опционально)	<code>nginx</code> версии 1.18 и выше, входящий в состав дистрибутива ОС

Обе указанные операционные системы включены в Единый реестр российских программ для ЭВМ и баз данных.

Серверная часть ПО поставляется в виде самодостаточного исполняемого файла и не требует установки сред исполнения, пакетных менеджеров или библиотек сверх системной библиотеки C. Минимальная версия glibc — 2.17; все поддерживаемые версии Astra Linux и RED OS удовлетворяют этому требованию с запасом.

Сборки для архитектуры ARM64 предоставляются по отдельному запросу.

2.3. Требования к базе данных

ПО использует два стандартных расширения PostgreSQL из состава пакета `contrib`:

Расширение	Назначение
<code>citext</code>	Регистронезависимые идентификаторы учётных записей
<code>ltree</code>	Иерархия разделов учебных материалов

Оба расширения начиная с PostgreSQL 13 отнесены к доверенным (trusted), поэтому создаются владельцем базы данных и **не требуют прав суперпользователя СУБД**. Приложение создаёт их самостоятельно при первом применении миграций.

2.4. Требования к сетевой конфигурации

Направление	Порт	Протокол	Назначение	Обязательность
Входящее, из сети организации	443	TCP/HTTPS	Доступ пользователей и администраторов через обратный прокси-сервер	При использовании обратного прокси-сервера (раздел 13)
Локальное, <code>127.0.0.1</code>	8080	TCP/HTTP	Порт службы ПО; наружу не публикуется	Обязательно
Исходящее, до сервера СУБД	5432	TCP	Подключение к PostgreSQL	Обязательно
Исходящее, до провайдера идентификации	443	TCP/HTTPS	Интеграция SAML 2.0 / OIDC (при её использовании)	Опционально
Исходящее, до службы каталогов	636	TCP/LDAPS	Интеграция LDAP (при её использовании)	Опционально
Исходящее, до службы каталогов	389	TCP/LDAP + StartTLS	Интеграция LDAP при подключении по <code>ldap://</code> с обязательным StartTLS (п. 4.5 руководства по эксплуатации)	Опционально
Исходящее, до сервера журналирования	514	UDP	Передача журнала аудита по syslog (при её использовании)	Опционально

Порт службы задаётся в конфигурационном файле; значение 8080 в настоящем руководстве выбрано для определённости.

Доступ ПО в сеть «Интернет» для функционирования **не требуется**. Исходящие соединения выполняются только к тем корпоративным сервисам, интеграция с которыми явно включена администратором.

2.5. Учётные записи и права

- Установка пакета выполняется от имени пользователя с правами `root` либо через `sudo`.
- В процессе установки пакет создаёт непривилегированную системную учётную запись `tradecap`, от имени которой работает служба. Интерактивный вход под этой учётной записью запрещён (`/usr/sbin/nologin`).
- Для СУБД создаётся отдельная учётная запись PostgreSQL с правами владельца базы данных ПО. Права суперпользователя СУБД приложению не требуются.

3. Состав дистрибутива

Комплект поставки передаётся правообладателем по защищённому каналу связи и включает:

№	Файл	Назначение
1	tradecap-<версия>-1. <арх>.rpm или tradecap_<версия>_<арх>.deb	Установочный пакет серверной части ПО
2	<имя пакета>.sha256	Контрольная сумма установочного пакета
3	<имя пакета>.asc	Открепленная электронная подпись формата OpenPGP
4	<имя пакета>.sig	Открепленная электронная подпись по ГОСТ Р 34.10-2012 (при наличии соответствующего требования в договоре)
5	tradecap-openpgp-pubkey.asc	Открытый ключ правообладателя для проверки подписи OpenPGP. Передаётся отдельно от архива (п. 4.3)
6	tradecap-gost-pubkey.cer	Сертификат правообладателя для проверки подписи по ГОСТ Р 34.10-2012 (при наличии требования в договоре). Передаётся отдельно от архива
7	SHA256SUMS , SHA256SUMS.asc	Контрольные суммы всех установочных пакетов комплекта и их электронная подпись
8	docs/	Настоящее руководство по установке и руководство по эксплуатации

Пакет содержит серверную часть ПО, встроенный веб-интерфейс, утилиту управления и схему базы данных. Отдельная установка веб-сервера для отдачи интерфейса не требуется.

Язык интерфейса определяется на этапе сборки: каждый артефакт содержит один языковой пакет.

Лицензионный ключ передаётся отдельно, после установки (раздел 10).

4. Получение дистрибутива и проверка целостности

4.1. Получение комплекта поставки

Комплект поставки передаётся заказчику по одному из согласованных защищённых каналов связи. Самостоятельная загрузка дистрибутива из сети «Интернет» не предусмотрена: распространение ПО осуществляется адресно, в объёме, определённом лицензионным договором.

Комплект передаётся одним архивом `tradecap-<версия>-delivery.tar.gz` :

```
cd /tmp
tar -xzf tradecap-<версия>-delivery.tar.gz
cd tradecap-<версия>
ls -l
```

Все команды последующих пунктов выполняются в распакованном каталоге.

4.2. Проверка контрольной суммы

Проверка выполняется **до установки пакета**:

```
LC_ALL=C sha256sum -c tradecap-<версия>-1.<арх>.rpm.sha256
```

Контрольные суммы всех файлов комплекта проверяются одной командой: `LC_ALL=C sha256sum -c SHA256SUMS` .

Ожидаемый результат — отметка `OK` . Префикс `LC_ALL=C` задан намеренно: без него вывод локализуется (на Astra Linux отметка выглядит как `ЦЕЛ`), что затрудняет автоматическую проверку.

Эталонное значение контрольной суммы дополнительно сообщается заказчику отдельным каналом (в акте приёма-передачи либо в сопроводительном письме). Сверка с этим значением подтверждает подлинность полученного файла; проверка только по вложенному файлу подтверждает лишь отсутствие повреждений при передаче.

Внутри пакета размещён второй файл контрольных сумм, охватывающий файлы версии ПО. Утилита управления сверяет его при каждой установке и обновлении, то есть при каждом выполнении команды `tradecap up` , поэтому целостность файлов версии проверяется не только при первичной установке.

Установку не следует продолжать, если проверка контрольной суммы завершилась ошибкой. В этом случае обратитесь в техническую поддержку правообладателя (раздел 17).

4.3. Электронная подпись дистрибутива

Архив комплекта содержит файлы открепленной электронной подписи формата OpenPGP. Открытый ключ правообладателя в архив намеренно не помещён: ключ, полученный вместе с тем, что он удостоверяет, подлинности не подтверждает. Ключ передаётся отдельно и опубликован на сайте (см. ниже).

Проверка выполняется **до установки пакета**. Сначала сверяется отпечаток полученного ключа, затем проверяется подпись:

```
gpg --import tradecap-openpgp-pubkey.asc
gpg --fingerprint tradecapp@yandex.ru
LC_ALL=C.UTF-8 gpg --verify tradecap-<версия>-1.<арх>.rpm.asc tradecap-<версия>-1.
<арх>.rpm
```

Ожидаемый результат — сообщение `Good signature` и код возврата `0`. Префикс `LC_ALL=C.UTF-8` задан по той же причине, что и в п. 4.2, но с сохранением кодировки: под `LC_ALL=C` наименование правообладателя в подписи выводится escape-последовательностями.

Отпечаток, который выводит вторая команда, **обязан совпадать** со значением, опубликованным на сайте по адресу <https://trade-cap.ru/security/pubkey>.

Без этой сверки сообщение `Good signature` означает лишь то, что архив и подпись согласованы между собой, но не то, что они получены от правообладателя.

При наличии соответствующего требования в договоре комплект дополнительно сопровождается файлом открепленной электронной подписи по ГОСТ Р 34.10-2012 и сертификатом правообладателя. В этом случае проверка выполняется средствами КриптоПро CSP:

```
certmgr -install -file tradecap-gost-pubkey.cer -store uMy
cryptcp -verify -detached -f tradecap-<версия>-1.<арх>.rpm.sig tradecap-<версия>-1.
<арх>.rpm
```

Ожидаемый результат — сообщение об успешной проверке и код возврата `0`.

Открытый ключ и сертификат правообладателя передаются заказчику отдельным каналом связи, а также публикуются на сайте правообладателя. Получение ключа тем же каналом, что и сам дистрибутив, не подтверждает подлинность: подпись удостоверяет происхождение файла лишь в той мере, в какой ключ получен независимо.

Если файлы подписи в составе комплекта отсутствуют, применяется проверка по контрольной сумме (п. 4.2).

5. Подготовка сервера

5.1. Проверка версии операционной системы

```
cat /etc/os-release
uname -m
```

Убедитесь, что версия ОС и архитектура соответствуют требованиям раздела 2.2.

5.2. Синхронизация времени

Корректное системное время необходимо для проверки срока действия лицензии, работы механизмов SSO и достоверности записей журнала аудита.

```
timedatectl set-timezone Europe/Moscow
timedatectl set-ntp true
timedatectl status
```

Команда `set-ntp` управляет службой синхронизации времени и завершается сообщением `Failed to set ntp: NTP not supported`, если такая служба в системе отсутствует (в частности, в образах Astra Linux без `systemd-timesyncd`). В этом случае установите и запустите службу синхронизации штатными средствами ОС, например `chrony`:

```
# Astra Linux
apt install chrony && systemctl enable --now chrony

# RED OS
dnf install chrony && systemctl enable --now chronyd
```

Убедитесь, что `timedatectl status` сообщает `System clock synchronized: yes`.

5.3. Проверка локали

Локаль `ru_RU.UTF-8` требуется для корректной сортировки данных в базе. Язык интерфейса от системной локали не зависит и определяется сборкой ПО.

```
locale -a | grep ru_RU
```

Если локаль отсутствует, сформируйте её штатными средствами ОС:

```
# Astra Linux
locale-gen ru_RU.UTF-8

# RED OS
localedef -i ru_RU -f UTF-8 ru_RU.UTF-8
```

5.4. Настройка межсетевого экрана

Откройте порт 443 для входящих подключений из сети организации. Порт службы 8080 наружу не публикуется, доступ к нему выполняется только с локального интерфейса.

```
# Astra Linux (ufw)
apt install ufw
ufw allow 22/tcp
ufw allow 443/tcp
ufw enable

# RED OS (firewalld)
dnf install firewalld
systemctl enable --now firewalld
firewall-cmd --permanent --add-port=443/tcp
firewall-cmd --reload
```

Межсетевой экран в обеих ОС в конфигурации по умолчанию не установлен: `ufw` отсутствует в базовой установке Astra Linux, `firewalld` — в базовой установке RED OS. Оба пакета входят в состав дистрибутивов и устанавливаются штатными средствами, как показано выше. Без установленного пакета команды `ufw` и `firewall-cmd` завершаются сообщением о том, что команда не найдена; если пакет `firewalld` установлен, но служба не запущена, `firewall-cmd` завершается ошибкой `Firewalld is not running`.

Правило доступа по SSH задаётся до включения `ufw`. По умолчанию `ufw` запрещает все входящие подключения, поэтому команда `ufw enable`, выполненная в удалённой сессии без разрешающего правила для порта 22, разрывает эту сессию и закрывает дальнейший доступ к серверу.

В отличие от `firewall-cmd`, команда `ufw allow` только добавляет правило: до выполнения `ufw enable` межсетевой экран остаётся выключенным и `ufw status` сообщает `неактивен`.

5.5. Мандатное управление доступом

RED OS. SELinux работает в режиме `enforcing`. Установка и работа службы проверены в этом режиме: пакет назначает файлам версии контекст `bin_t`, служба выполняется в домене `unconfined_service_t`, отказов подсистемы не возникает. Дополнительных действий администратора не требуется, переводить SELinux в режим `permissive` не нужно. Проверить режим можно командой `getenforce`.

Astra Linux. Установка проверена в конфигурации по умолчанию, в которой подсистемы мандатного контроля (`astra-mac-control`, `astra-digsig-control`, `astra-hardened-control`) отключены. При развёртывании на защищённой конфигурации с включённым контролем целостности исполняемых файлов (`astra-digsig-control`) требуется предварительное согласование с правообладателем: исполняемый файл службы должен быть подписан в соответствии с политикой заказчика.

Проверить состояние подсистем можно командами `astra-mac-control status` и `astra-digsig-control status`.

6. Подготовка системы управления базами данных

6.1. Установка PostgreSQL

Если СУБД не установлена, установите её штатными средствами ОС вместе с пакетом расширений:

```
# Astra Linux
apt update && apt install postgresql postgresql-contrib

# RED OS (пример для PostgreSQL 16)
dnf install postgresql16-server postgresql16-contrib
/usr/pgsql-16/bin/postgresql-16-setup initdb
systemctl enable --now postgresql-16
```

6.2. Создание учётной записи и базы данных

Выполните от имени системного пользователя `postgres`:

```
sudo -u postgres psql <<'SQL'
CREATE USER tradecap WITH PASSWORD 'ЗАДАЙТЕ_СТОЙКИЙ_ПАРОЛЬ';
CREATE DATABASE tradecap
    OWNER tradecap
    ENCODING 'UTF8'
    LC_COLLATE 'ru_RU.UTF-8'
    LC_CTYPE 'ru_RU.UTF-8'
    TEMPLATE template0;
SQL
```

Учётная запись должна быть владельцем базы данных: этого достаточно для создания расширений, перечисленных в п. 2.3.

6.3. Настройка доступа к СУБД

В файле `pg_hba.conf` разрешите подключение учётной записи `tradecap` к одноимённой базе данных с проверкой пароля:

#	TYPE	DATABASE	USER	ADDRESS	METHOD
host		tradecap	tradecap	127.0.0.1/32	scram-sha-256

Примените изменения:

```
systemctl reload postgresql-16 # RED OS
systemctl reload postgresql     # Astra Linux
```

6.4. Проверка подключения

```
psql "host=127.0.0.1 port=5432 dbname=tradecap user=tradecap" -c "SELECT version();"

```

7. Установка пакета

Выполните установку штатными средствами пакетного менеджера операционной системы:

```
# Astra Linux
apt install ./tradecap_<версия>_amd64.deb

# RED OS
dnf install ./tradecap-<версия>-1.x86_64.rpm
```

В результате установки в системе создаются следующие объекты:

Путь	Назначение
/usr/bin/tradecap	Утилита управления ПО
/opt/tradecap/	Каталог установки (права доступа 0750 , владелец root:tradecap)
/opt/tradecap/config.toml.example	Образец конфигурационного файла
/opt/tradecap/releases/<версия>/	Файлы версии ПО и схема базы данных
/usr/lib/tradecap/release/	Служебная копия версии, принадлежащая пакетному менеджеру

Пакет создаёт системную учётную запись tradecap и выводит подсказку с дальнейшими шагами.

Служба после установки **не запускается автоматически** и unit-файл systemd на этом этапе не создаётся: и то, и другое выполняет команда tradecap up после конфигурирования (раздел 9).

Утилита tradecap объединяет все операции обслуживания установки:

Команда	Назначение
tradecap config	Создать конфигурационный файл: без флагов — в диалоговом режиме, с флагами — без запросов (п. 8.1)
tradecap up <версия>	Перевести установку на версию: проверка целостности, миграции, запуск службы
tradecap status	Состояние установки и службы
tradecap logs	Журнал службы
tradecap down	Остановить службу. Автозапуск сохраняется
tradecap destroy	Остановить службу и удалить unit-файл. Данные и конфигурация сохраняются
tradecap license show	Идентификатор установки и состояние лицензии
tradecap license set -file <файл>	Установить лицензию из файла
tradecap owner create --login <имя>	Создать учётную запись владельца

Команда	Назначение
tradecap owner reset - -login <имя>	Сбросить пароль владельца

Каталог установки задан в пакете, поэтому флаг `--config` при вызове команд не требуется.

8. Конфигурирование

8.1. Создание конфигурационного файла

Файл создаётся утилитой управления: она заполняет все обязательные параметры, включая путь к файлу журнала аудита.

```
printf '%s\n' 'postgresql://tradecap:ПАРОЛЬ@127.0.0.1:5432/tradecap' \  
| sudo tradecap config --database-url-stdin --http-host 127.0.0.1 --http-port 8080
```

Строка подключения передаётся через стандартный ввод: аргументы командной строки видны в списке процессов и сохраняются в истории командной оболочки.

Файл содержит пароль к базе данных. Права `0640`, владелец `root`, группа `tradecap` устанавливаются при первом запуске (раздел 9): служба читает файл по членству в группе, другим пользователям он недоступен.

Образец `/opt/tradecap/config.toml.example` перечисляет все параметры и пригоден для копирования.

8.2. Параметры конфигурации

Конфигурационный файл имеет формат TOML.

```
[server]
# Интерфейс и порт службы. Наружу публикуются через обратный прокси-сервер.
host = "127.0.0.1"
port = 8080
# Доверять заголовкам X-Forwarded-* от обратного прокси-сервера.
# Требуется включить при работе за прокси, иначе в журнале аудита
# фиксируется адрес прокси-сервера вместо адреса клиента.
trust_proxy = false

[database]
# Строка подключения к внешней СУБД.
url = "postgresql://tradecap:ПАРОЛЬ@127.0.0.1:5432/tradecap"

[logging]
level = "info"      # trace | debug | info | warn | error | fatal

[audit]
# fail_closed: при невозможности записать событие аудита в базу данных
# операция отклоняется. fail_open: операция выполняется, событие теряется.
failure_mode = "fail_closed"

[audit.file]
enabled = true
path = "/var/log/tradecap/audit.log"

[audit.syslog]
enabled = false
host = "127.0.0.1"
port = 514

[telemetry]
enabled = false
```

Обязательные параметры: `database.url`, `server.host`, `server.port`. При создании файла по п. 8.1 они заданы.

При включённом файловом приёмнике (`audit.file.enabled = true`, значение по умолчанию) обязателен также `audit.file.path`. Без него применяется путь по умолчанию, не совпадающий с каталогом, который создаёт служба: при `failure_mode = "fail_closed"` служба не запускается (раздел 16). Если файловый приёмник не нужен, задайте `audit.file.enabled = false` — журнал аудита продолжает вестись в базе данных.

Параметр, требующий внимания: `trust_proxy`. При публикации веб-интерфейса через обратный прокси-сервер (раздел 13) его следует включить, иначе все записи журнала аудита будут содержать адрес прокси-сервера. Значение принимает `true`, `false` либо перечень адресов и подсетей доверенных прокси-серверов, например `"127.0.0.1,10.0.0.0/8"`.

Каталог `/var/log/tradecap` создаёт сама служба при запуске, отдельных действий он не требует. Если в `audit.file.path` указан каталог за пределами `/var/log/tradecap`, его следует создать заранее и выдать права на запись учётной записи `tradecap`:

```
install -d -o tradecap -g tradecap -m 0750 <каталог>
```

8.3. Параметры, задаваемые через веб-интерфейс

Следующие настройки хранятся в базе данных и изменяются через панель администратора без правки конфигурационного файла и без перезапуска службы:

- провайдеры аутентификации (SAML 2.0, OIDC, LTI, SCIM, LDAP), в том числе несколько одновременно;
- перечень доменов, которым разрешено встраивание виджетов;
- срок жизни сессии и таймаут бездействия;
- наименование и описание установки;
- лицензионный ключ (раздел 10).

Многофакторная аутентификация обеспечивается на стороне корпоративного провайдера идентификации.

9. Первый запуск

Команда `up` выполняет весь первичный ввод в эксплуатацию: сверяет контрольные суммы файлов версии, дожидается доступности СУБД, применяет миграции схемы, создаёт unit-файл `systemd`, включает автозапуск и запускает службу.

Здесь и далее команды утилиты управления выполняются с правами `root`: конфигурационный файл имеет права `0640` и владельца `root` (п. 8.1), поэтому от имени обычного пользователя утилита сообщает, что конфигурация не найдена.

```
sudo tradecap up <версия>
```

Перед применением миграций утилита выводит их перечень и предлагает подтвердить наличие резервной копии базы данных. При первичной установке база пуста, подтверждение можно дать сразу. Для неинтерактивной установки подтверждение отключается флагом `-y`.

Команда `up` завершается только после того, как приложение начало обслуживать запросы: `unit` имеет тип `notify`, поэтому `systemd` удерживает его в состоянии `activating`, пока приложение не сообщит о готовности. Служба, которая не может запуститься, не показывает `active` ни в один момент времени: команда завершается с ошибкой сразу, а `unit` переходит в состояние `failed` после исчерпания лимита перезапусков, примерно через три минуты.

```
systemctl status tradecap  
curl -s http://127.0.0.1:8080/readyz
```

Ожидаемый результат — состояние `active (running)` и ответ `{"status":"ok","db":"up"}`. Если служба в состоянии `failed`, причина отказа указана в журнале службы (`journalctl -u tradecap`); типовые случаи приведены в разделе 16.

Служба работает от имени учётной записи `tradecap` с ограничениями `systemd: NoNewPrivileges`, `CapabilityBoundingSet` и `AmbientCapabilities` пусты, `UMask=0027`, `PrivateTmp`, `PrivateDevices`, `ProtectSystem=full`, `ProtectProc=invisible`, `ProcSubset=pid`, `ProtectKernelTunables`, `ProtectKernelModules`, `ProtectKernelLogs`, `ProtectControlGroups`, `RestrictSUIDSGID`, `RestrictRealtime`, `LockPersonality`, `RestrictAddressFamilies`.

10. Лицензирование

10.1. Модель лицензирования

Лицензия представляет собой подписанный правообладателем текстовый ключ. Он содержит наименование заказчика, срок действия, перечень охватываемых установок и коммерческие параметры. Проверка подписи выполняется **локально**, обращений к внешним сервисам не происходит.

Лицензия хранится в базе данных установки. Конфигурационный файл лицензию не содержит и не ссылается на неё.

10.2. Идентификатор установки

При создании схемы базы данных установке присваивается постоянный идентификатор. Он образуется миграцией, то есть становится доступен сразу после выполнения `tradecap up` (раздел 9), и далее не меняется:

```
tradecap license show
```

До выполнения `tradecap up` команда сообщает, что схема ещё не создана; это ожидаемо.

Пример вывода после первого запуска, до установки лицензии:

```
installation: 1f55a198-c308-4c48-a2a3-48800c55ba74
licence:      none installed
```

Передайте идентификатор установки правообладателю. Лицензионный ключ выпускается под конкретные установки: при развёртывании нескольких контуров (продуктивного, тестового, резервного) сообщите идентификаторы всех контуров, чтобы получить один ключ, действующий для каждого из них.

10.3. Установка лицензионного ключа

Основной способ — через панель администратора: раздел лицензии, вставка полученного ключа. Ключ применяется немедленно, перезапуск службы не требуется.

Резервный способ, если веб-интерфейс недоступен:

```
tradecap license set --file /tmp/license.key
```

Утилита проверяет подпись до сохранения: ключ, который не проходит проверку, отклоняется, ранее установленная лицензия при этом сохраняется.

Проверьте результат:

```
tradecap license show
```

```
installation: 1f55a198-c308-4c48-a2a3-48800c55ba74
licence:      valid, expires 2030-01-01
issued to:    000 «Ромашка»
edition:      corporate
covers:       [1f55a198-c308-4c48-a2a3-48800c55ba74]
```

Поле `edition` содержит латинский идентификатор редакции ПО, указанной в лицензионном договоре: значение `corporate` соответствует редакции «Корпоративный» (состав редакций приведён в документе о порядке определения стоимости).

10.4. Поведение при отсутствии действующей лицензии

Отсутствие или истечение лицензии **не останавливает работу установки**. Пользователи продолжают входить в систему, просматривать учебные материалы и проходить их; прогресс сохраняется. Ограничивается только расширение установки: создание учётных записей, создание и публикация материалов, загрузка наборов рыночных данных.

Предусмотрены два периода, в течение которых ограничения не применяются:

Ситуация	Поведение
Новая установка без лицензии	Полная функциональность в течение 30 дней с момента создания базы данных. Этого периода достаточно, чтобы получить ключ по процедуре п. 10.2
Истечение срока действия лицензии	Полная функциональность в течение льготного периода, указанного в самом ключе (по умолчанию 14 дней)

По истечении указанных периодов установка переходит в ограниченный режим. Для возврата к полной функциональности достаточно установить действующий ключ.

Текущее состояние лицензии отображается в панели администратора.

11. Создание учётной записи владельца

Первая учётная запись создаётся утилитой управления. Приложение не создаёт владельцев самостоятельно: это единственный способ получить первичный доступ.

```
tradecap owner create --login admin
```

Утилита запрашивает пароль и создаёт учётную запись с ролью «Владелец». При первом входе система потребует сменить пароль.

Запрос пароля требует терминала. При установке по сценарию (без терминала) пароль передаётся через стандартный ввод:

```
printf %s 'ПАРОЛЬ_ВЛАДЕЛЬЦА' | tradecap owner create --login admin --password-stdin
```

Пароль передаётся именно так, а не аргументом командной строки: аргументы видны в списке процессов и сохраняются в истории командной оболочки. Флаг `--password-stdin` поддерживает и команда `tradecap owner reset`.

Если пароль владельца утрачен, доступ восстанавливается командой `tradecap owner reset --login admin`.

12. Проверка работоспособности установки

12.1. Диагностические endpoint'ы

Проверка выполняется локально с сервера:

```
curl -s http://127.0.0.1:8080/healthz
curl -s http://127.0.0.1:8080/readyz
```

Ожидаемые ответы:

```
{"status": "ok"}
{"status": "ok", "db": "up"}
```

`/healthz` подтверждает, что процесс службы отвечает. `/readyz` дополнительно проверяет соединение с СУБД и возвращает код 503 при её недоступности, что позволяет использовать его в качестве пробы балансировщика.

12.2. Проверка журналов запуска

```
journalctl -u tradecap -n 50 --no-pager
```

Записи уровня `error` при штатном запуске отсутствуют. Строка о состоянии лицензии присутствует всегда, в том числе когда лицензия ещё не установлена.

12.3. Проверка веб-интерфейса

```
curl -s -o /dev/null -w "%{http_code}\n" http://127.0.0.1:8080/
```

Ожидаемый код ответа — `200`. Затем откройте в браузере внешний адрес установки и выполните вход под учётной записью владельца. Убедитесь, что панель администратора открывается и разделы доступны.

13. Настройка обратного прокси-сервера и TLS

Служба принимает подключения на локальном интерфейсе и отдаёт веб-интерфейс самостоятельно. Обратный прокси-сервер **в состав ПО не входит и не является его требованием**: он предоставляется и настраивается заказчиком, если публикация веб-интерфейса в сеть организации выполняется с терминированием TLS на отдельном узле. Приведённая ниже конфигурация носит справочный характер.

Пример конфигурации nginx (/etc/nginx/conf.d/tradecap.conf):

```
server {
    listen 443 ssl;
    server_name tradecap.example.ru;

    ssl_certificate      /etc/ssl/certs/tradecap.crt;
    ssl_certificate_key  /etc/ssl/private/tradecap.key;
    ssl_protocols        TLSv1.2 TLSv1.3;

    client_max_body_size 1g;

    location / {
        proxy_pass http://127.0.0.1:8080;
        proxy_set_header Host          $host;
        proxy_set_header X-Real-IP      $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_read_timeout 60s;
    }
}

server {
    listen 80;
    server_name tradecap.example.ru;
    return 301 https://$host$request_uri;
}
```

Значение `client_max_body_size` определяется максимальным размером загружаемого набора рыночных данных (до 1 ГБ). Если загрузка наборов данных не используется, значение можно уменьшить до 16 МБ.

При работе за обратным прокси-сервером включите параметр `trust_proxy` (п. 8.2) и перезапустите службу командой `sudo tradecap up`. Без этого записи журнала аудита будут содержать адрес прокси-сервера.

Примените конфигурацию:

```
nginx -t && systemctl reload nginx
```

14. Проверка встраивания виджетов

Учебные материалы могут встраиваться на портал заказчика. Виджет создаётся в панели администратора: выбираются материалы, которые он вправе показывать, и перечень доменов, которым разрешено встраивание. При создании выдаётся публичный ключ виджета вида `wgt_pub_...`.

Код встраивания состоит из контейнера, блока конфигурации и загрузчика:

```
<div class="tradecap-education" style="width:100%;max-width:640px;height:440px">
  <script type="application/json" class="edu-embed-config">
    {
      "params": { "key": "wgt_pub_XXXXXXX", "theme": "light" },
      "state": {
        "community_id": "enterprise",
        "ed_content_id": "<идентификатор материала>"
      }
    }
  </script>
  <script src="https://tradecap.example.ru/widget/embed.js" async></script>
</div>
```

Класс контейнера произвольный — загрузчик находит его как родительский элемент своего тега `<script>`. Класс блока конфигурации `edu-embed-config` **обязателен и изменению не подлежит**: по нему загрузчик находит конфигурацию. Если класс изменён, конфигурация не читается, ключ виджета не попадает в адрес встраиваемого кадра, и браузер блокирует встраивание с сообщением `frame-ancestors 'none'`.

Поле	Значение
<code>params.key</code>	Публичный ключ виджета, выданный при его создании
<code>params.theme</code>	<code>light</code> , <code>dark</code> либо параметр опускается
<code>state.community_id</code>	<code>enterprise</code>
<code>state.ed_content_id</code>	Идентификатор учебного материала (UUID)

Ключ виджета публичный, его размещение в исходном коде страницы допустимо: доступ ограничивается перечнем доменов, а не секретностью ключа.

Разместите код на тестовой странице портала, домен которой внесён в перечень разрешённых для этого виджета, и убедитесь, что материал открывается.

Материал отдаётся во встраивании только при видимости «Видно»: скрытый материал в перечне виджета помечен как недоступный для встраивания.

При обращении с домена, не внесённого в перечень, встраивание блокируется на стороне сервера, а браузер дополнительно ограничивается заголовком `frame-ancestors`. Это штатное поведение механизма защиты от встраивания на посторонних ресурсах. Пустой перечень доменов означает запрет встраивания со всех внешних доменов.

15. Обслуживание установки

15.1. Резервное копирование

Приложение не хранит данные в файловой системе, поэтому резервная копия базы данных содержит всё состояние установки, включая учебные материалы, учётные записи, журнал аудита, идентификатор установки и лицензию.

```
sudo install -d -o postgres -g postgres -m 0700 /var/backups
sudo -u postgres pg_dump -Fc -f /var/backups/tradecap-$(date +%F).dump tradecap
sudo cp -a /opt/tradecap/config.toml /var/backups/tradecap-config-$(date +%F).toml
```

Каталог `/var/backups` на RED OS отсутствует и создаётся первой командой. Файл дампа задаётся ключом `-f`, а не перенаправлением вывода: перенаправление выполняется от имени вызывающего пользователя, а не от имени `postgres`.

Восстановление выполняется на пустую базу данных. Службу необходимо остановить заранее: при работающей службе команда удаления базы данных либо завершается ошибкой «база данных занята другими пользователями», либо, если пул соединений в этот момент пуст, выполняется успешно и удаляет базу под работающей службой. Во втором случае `tradecap status` продолжает показывать `running`, а `/readyz` отвечает `503`.

```
sudo tradecap down
sudo -u postgres dropdb --if-exists tradecap
sudo -u postgres createdb -O tradecap -E UTF8 \
    --lc-collate 'ru_RU.UTF-8' --lc-ctype 'ru_RU.UTF-8' -T template0 tradecap
sudo -u postgres pg_restore -d tradecap /var/backups/tradecap-<дата>.dump
sudo tradecap up
```

Кодировка и правила сортировки задаются явно, как и при первичном создании базы данных (п. 6.2): без этих ключей база создаётся с параметрами кластера, которые могут отличаться от требуемых.

15.2. Обновление версии

Обновления устанавливаются по согласованию с заказчиком; автоматическое обновление не применяется.

```
# 1. Резервное копирование (п. 15.1)

# 2. Проверка контрольной суммы и электронной подписи нового пакета (пп. 4.2, 4.3)

# 3. Установка нового пакета. Служба продолжает работать на прежней версии.
apt install ./tradecap_<новая-версия>_amd64.deb      # Astra Linux
dnf install ./tradecap-<новая-версия>-1.x86_64.rpm # RED OS

# 4. Перевод установки на новую версию: миграции и перезапуск службы
sudo tradecap up <новая-версия>

# 5. Проверка
tradecap status
curl -s http://127.0.0.1:8080/readyz
```

Установка пакета и перевод на новую версию разделены намеренно: до выполнения `tradecap up` служба продолжает работать на прежней версии, и момент перерыва в обслуживании выбирает администратор.

Если новая версия не запустилась, установка автоматически возвращается на прежнюю. Служба при этом продолжает обслуживать запросы только в том случае, если причина отказа относится к самой новой версии. Ошибка в конфигурационном файле останавливает и прежнюю версию — файл у них общий. Сама команда при этом завершается с ненулевым кодом возврата и сообщением об ошибке: это относится к неудавшемуся обновлению, а не к откату. Через несколько секунд после отката следует убедиться, что служба отвечает (п. 12.1). Причина отказа указывается в журнале службы (`journalctl -u tradecap`); типовые случаи приведены в разделе 16. После устранения причины команду шага 4 следует выполнить повторно, указав версию явно: `tradecap up` без аргумента переведёт установку на текущую, то есть прежнюю, версию.

Конфигурационный файл, лицензия и база данных при обновлении не изменяются.

15.3. Откат на предыдущую версию

Предыдущие версии остаются на диске, поэтому откат выполняется той же командой:

```
sudo tradecap up <предыдущая-версия>
```

Если новая версия применила миграции схемы, утилита предупредит об этом и потребует подтверждения. Откат допустим только если предыдущая версия совместима с текущей схемой; в противном случае восстановите базу данных из резервной копии.

Работающую версию показывает команда `tradecap version`: она выводит отдельно версию управляющей утилиты, которая устанавливается пакетом, и версию текущего релиза, который обслуживает запросы. Откат меняет вторую и не затрагивает первую, поэтому `tradecap --version` после отката продолжает показывать версию пакета.

15.4. Удаление ПО

```
# Astra Linux
apt remove tradecap

# RED OS
dnf remove tradecap
```

При удалении пакета служба останавливается, автозапуск отключается и unit-файл `systemd` удаляется.

После удаления пакета в системе остаются: конфигурационный файл `/opt/tradecap/config.toml`, файлы установленных версий в `/opt/tradecap/releases` (порядка 220 МБ на версию), каталог журнала аудита `/var/log/tradecap` вместе с его содержимым, системная учётная запись `tradecap`, учётная запись `tradecap` в СУБД и сама база данных. Это сделано намеренно, чтобы переустановка пакета не потребовала повторной настройки. На RED OS дополнительно остаётся пустой каталог `/usr/lib/tradecap`; на Astra Linux он удаляется вместе с пакетом.

Пакет не содержит файлов конфигурации, зарегистрированных в пакетном менеджере, поэтому команда `apt purge` после `apt remove` завершится сообщением о том, что пакет не найден: это ожидаемо и не является ошибкой.

Обратите внимание: сохранившийся файл `/opt/tradecap/config.toml` содержит пароль подключения к базе данных. Если сервер выводится из эксплуатации, удалите его вместе с остальными файлами:

```
sudo rm -rf /opt/tradecap /usr/lib/tradecap /var/log/tradecap
sudo userdel tradecap
sudo -u postgres psql -c 'DROP DATABASE tradecap'
sudo -u postgres psql -c 'DROP ROLE tradecap'
```

Учётная запись в СУБД удаляется отдельной командой: без неё повторная установка на этом же сервере прерывается на шаге 6.2 сообщением о том, что роль уже существует.

16. Типовые проблемы при установке

Признак	Вероятная причина	Порядок устранения
no configuration at /opt/tradecap/config.toml	Конфигурационный файл не создан либо команда выполнена без прав root	Выполнить п. 8.1; если файл создан, повторить команду через sudo (раздел 9)
database not reachable at the configured DSN	СУБД не запущена, недоступна по указанному адресу либо не разрешено подключение	Проверить systemctl status postgresql-16, параметр database.url, правила pg_hba.conf
password authentication failed for user "tradecap"	Пароль в database.url не совпадает с заданным при создании учётной записи	Сверить пароль с п. 6.2
unrecognized configuration parameter при подключении	В строке подключения указаны параметры, не поддерживаемые сервером	Привести database.url к виду из п. 8.2
this database has no installation yet; run up to apply migrations first при вызове license show	Миграции ещё не применены	Выполнить sudo tradecap up <версия> (раздел 9)
licence signature does not verify	Файл ключа повреждён либо выпущен для другой сборки ПО	Запросить корректный ключ у правообладателя
issued to installation(s) ..., not to ...	Ключ выпущен для других установок	Передать правообладателю идентификатор этой установки для перевыпуска ключа (п. 10.2)
Служба активна, веб-интерфейс недоступен извне	Не настроен либо некорректно настроен обратный прокси-сервер	Проверить nginx -t, доступность порта 443, корректность proxy_pass
В журнале аудита фиксируется адрес прокси-сервера	Не включён параметр trust_proxy	Включить параметр (п. 8.2) и выполнить sudo tradecap up
Виджет не отображается на портале	Домен портала не внесён в перечень разрешённых для виджета	Добавить домен в панели администратора
Ошибка при загрузке набора данных через прокси-сервер	Значение client_max_body_size меньше размера файла	Привести значение в соответствие (раздел 13) и перезагрузить nginx
Служба не запускается, в журнале службы — Приёмник аудита file не может записывать события и код EACCESS	Не задан параметр audit.file.path, применён каталог по умолчанию	Задать путь (п. 8.2) и выполнить sudo tradecap up <версия>

Признак	Вероятная причина	Порядок устранения
Некорректная сортировка русского текста	База создана в иной локали	Пересоздать базу с параметрами п. 6.2 и восстановить данные из резервной копии
После входа любой запрос к API отклоняется с требованием сменить пароль	Учётная запись владельца создана с обязательной сменой пароля при первом входе	Сменить пароль через веб-интерфейс; это штатное поведение

При сохранении неисправности подготовьте для обращения в техническую поддержку: версию ОС, вывод `tradecap status` (он начинается с версий утилиты и текущего релиза), вывод `tradecap license show`, последние 200 строк `journalctl -u tradecap`, ответ `/readyz`.

17. Техническая поддержка правообладателя

Техническая поддержка оказывается на русском языке силами штатных сотрудников правообладателя.

Реквизит	Сведения
Правообладатель	Общество с ограниченной ответственностью «Трейдкап» (ООО «Трейдкап»)
Ответственное лицо	Михелс Шота Михайлович
Должность	Генеральный директор УК ООО «СММ Инвест»
Адрес	121069, Москва, ул. Малая Никитская, д. 27, стр. 4, помещ. 2/1
Телефон	+7 (495) 690-35-35
Электронная почта	tradecapp@yandex.ru
Сайт	https://trade-cap.ru

18. Контрольный перечень выполненных работ

№	Этап	Раздел	Отметка
1	Проверены требования к аппаратному и программному обеспечению	2	☐
2	Получен комплект поставки, проверены контрольная сумма и электронная подпись	4	☐
3	Настроены время, локаль и межсетевой экран	5	☐
4	Установлена СУБД, создана база данных и учётная запись	6	☐
5	Установлен пакет ПО	7	☐
6	Сформирован конфигурационный файл	8	☐
7	Применены миграции, служба запущена, автозапуск включён	9	☐
8	Получен идентификатор установки и передан правообладателю	10.2	☐
9	Установлен и проверен лицензионный ключ	10.3	☐
10	Создана учётная запись владельца	11	☐
11	Проверены диагностические endpoint'ы и веб-интерфейс	12	☐
12	При публикации через обратный прокси: настроен прокси-сервер и TLS, включён <code>trust_proxy</code>	13	☐
13	Проверено встраивание виджета на портале заказчика	14	☐
14	Настроено резервное копирование базы данных	15.1	☐

Настоящий документ поставляется в составе комплекта поставки ПО.