

Imperium Identity Security

Руководство по установке

15.09.2026

Version 1.0.0

Содержание

Аннотация	3	5.3. Доступ к порту базы данных.....	9
1. Системные требования.....	3	5.4. Проверка конфигурации	10
1.1. Сервер и сеть	3	5.5. Работа за обратным прокси.....	10
1.2. Рабочее место администратора	4	6. Подготовка Nuntius AI	11
1.3. Программные зависимости	4	6.1. Размерность векторов	11
2. Установка программного обеспечения сервера....	4	6.2. Когда создаются таблицы	12
2.1. Проверка свободного места	4	6.3. Корпус документации	12
2.2. Docker и Docker Compose.....	5	7. Первичный запуск и проверка	12
2.3. Дополнительные утилиты	5	7.1. Запуск служб	12
3. Получение дистрибутива	5	7.2. Запуск приложения и порт базы данных	13
3.1. Каталог установки	6	7.3. Таблицы векторного поиска	14
3.2. Архив и контрольная сумма.....	6	7.4. Запуск после перезагрузки сервера.....	14
3.3. Распаковка	7	8. Проверка веб-интерфейса	14
4. Импорт образов и состав поставки	7	9. Первичный вход.....	15
4.1. Образ приложения	7	9.1. Учётная запись администратора	15
4.2. Образ PostgreSQL	7	9.2. Смена первичного пароля	16
4.3. Компоненты и лицензии	8	9.3. Именные администраторы и владелец фоновых задач.....	17
5. Настройка учётных данных до первого запуска....	8	10. Язык веб-интерфейса	18
5.1. Проверка новой установки	8	11. Консультации по установке	19
5.2. Пароли в файле <code>.env</code>	8		

Аннотация

Руководство описывает первую установку Imperium Identity Security версии 1.0.0 на новый сервер. Команды рассчитаны на РЕД ОС 8.0 и каталог `/srv/imperium`. Их выполняет администратор с правами `sudo`.

Порядок работ: проверьте сервер, установите Docker, получите и проверьте архив, загрузите образы, задайте пароли в файле `.env` и запустите систему. После запуска проверьте базу данных, страницу входа и смените первичный пароль администратора.

Для существующей установки нужен отдельный план обновления с резервной копией. Не распаковывайте архив поверх рабочих настроек и не переинициализируйте базу данных по этой инструкции. Сопровождение установленной системы описано в «Руководстве администратора».

1. Системные требования

1.1. Сервер и сеть

Ресурс	Требование
Процессор	64-разрядный с архитектурой x86-64 (amd64), не менее 2 ГГц и 2 ядер
Оперативная память	Не менее 6 ГБ
Свободное место	Не менее 30 ГБ на разделе с каталогом установки
Операционная система	РЕД ОС 8.0, редакция «Минимальный сервер», сертифицированная редакция; допустима другая система с поддержкой контейнеров по согласованию с поставщиком
Учётная запись для установки	Права <code>sudo</code> ; пользователь входит в группу <code>wheel</code>

Образ приложения собран только для архитектуры x86-64. На сервере с процессором ARM (aarch64) он без эмуляции не запустится.

Объём диска для дальнейшей работы зависит от числа объектов, журналов, документов и резервных копий. Указанный запас для установки не заменяет расчёт ёмкости рабочей системы.

Разрешите нужные соединения:

- исходящий `443/tcp` к `dist.identigy.ru` для загрузки архива;
- доступ к репозиториям пакетов РЕД ОС и к реестру образов Docker Hub для отдельного образа PostgreSQL;
- исходящий `443/tcp` к серверу языковой модели, если будет использоваться помощник Nuntius AI (раздел 6);
- входящий `22/tcp` от рабочего места администратора для SSH;
- доступ к веб-интерфейсу по `80/tcp` в закрытом сегменте или по `443/tcp` через настроенный HTTPS.

Настройка межсетевого экрана и сетевого доступа выходит за рамки этого руководства.

Поставляемый файл `docker-compose.yml` занимает на сервере три порта:

- `80` — веб-интерфейс приложения по HTTP;
- `443` — зарезервирован за приложением, но HTTPS в нём по умолчанию не включён: соединение на этот порт обрывается;
- `5432` — база данных.

Порт базы данных до запуска ограничьте по разделу 5.3: пользователям веб-интерфейса он не нужен. Если HTTPS будет обеспечивать обратный прокси на этом же сервере, освободите порты `80` и `443` по разделу 5.5, иначе прокси не сможет их занять.

1.2. Рабочее место администратора

Веб-интерфейс открывается в браузере на рабочем месте администратора или пользователя. Работа проверялась в браузерах:

- Яндекс Браузер версии 26.0 или выше — рекомендуемый браузер;
- Chromium версии 120 или выше.

Интерфейс построен на стандартных веб-технологиях и работает в современных браузерах (Google Chrome, Microsoft Edge, Mozilla Firefox, Safari). Для другого браузера сначала проверьте вход и основные действия на целевой установке.

Дополнительно на рабочем месте требуются разрешённые в браузере JavaScript и cookies, а также сетевой доступ к серверу по `80/tcp` (или `443/tcp` при HTTPS) либо SSH-туннель, описанный в разделе 8.

1.3. Программные зависимости

На сервере нужны:

- Docker версии 29.5.3 или выше;
- Docker Compose версии 5.1.3 или выше;
- `unzip` версии 6.0 или выше;
- `curl` версии 8.0.1 или выше;
- `sha256sum` версии 8.32 или выше;
- SSH-сервер и текстовый редактор, например `vi`.

Docker запускает контейнеры. Docker Compose по файлу `docker-compose.yml` запускает три службы: базу данных, службу начальной подготовки и приложение. Java отдельно на сервере устанавливать не нужно: она входит в образ приложения. Установка зависимостей описана в разделе 2.

2. Установка программного обеспечения сервера

2.1. Проверка свободного места

Проверьте раздел, где находится `/srv`:

```
df -h /srv
```

В колонке `Avail` должно быть не менее `30G`. Если для `/srv/imperium` выделен другой раздел, проверяйте именно его.

2.2. Docker и Docker Compose

Установите пакеты из настроенных репозиториях РЕД ОС, запустите службу и добавьте её в автозагрузку:

```
sudo dnf install -y docker-ce docker-ce-cli docker-compose
sudo systemctl enable docker --now
```

Проверьте службу и версии:

```
systemctl status docker
sudo docker version
sudo docker compose version
```

Служба должна находиться в состоянии `active (running)`, а версии — соответствовать разделу 1.3. Вывод `systemctl status` открывается в постраничном просмотре; для выхода нажмите клавишу `q`.

Если нужных пакетов или версий нет в доступных репозиториях, остановите установку и уточните способ их получения у поставщика. Не отключайте проверку подписей пакетов.

2.3. Дополнительные утилиты

Установите утилиты загрузки и распаковки:

```
sudo dnf install -y unzip curl
```

Если SSH-сервер отсутствует, установите OpenSSH:

```
sudo dnf install -y openssh
```

Проверьте наличие утилит и службы SSH:

```
unzip -v
curl --version
sha256sum --version
systemctl status sshd
```

Запуск SSH и правила доступа согласуйте с администратором сервера. Не меняйте способ входа посреди текущего удалённого сеанса без запасного доступа.

3. Получение дистрибутива

Сведения о продукте размещены на [странице продукта](#) сайта разработчика, раздел «Решения» → «Imperium Identity Security». Сборки размещаются на сервере дистрибутивов `dist.identigy.ru`. Для установки используйте согласованный выпуск 1.0.0.

Архив `identity-imperium-1.0.0.zip` содержит три файла в корне:

- `docker-compose.yml` — состав служб и их параметры;
- `.env` — параметры установки: пароли и размерность векторов;
- `imperium.tar` — образ приложения.

Образ PostgreSQL в этот архив не входит. Он загружается отдельно по разделу 4.2.

3.1. Каталог установки

Выберите новый каталог. Если `/srv/imperium` уже содержит файлы другой установки, не продолжайте по этой процедуре.

Создайте каталог и назначьте его владельцем текущего пользователя. Для создания каталога в `/srv` нужны права `sudo`: каталог принадлежит `root`.

```
sudo mkdir -p /srv/imperium
sudo chown "$(whoami)":root /srv/imperium
sudo chmod 700 /srv/imperium
cd /srv/imperium
umask 077
```

Проверьте текущий путь и права:

```
pwd
ls -ld /srv/imperium
```

Ожидаемый путь — `/srv/imperium`. Права каталога — `700`: вход и чтение разрешены владельцу. Ограничение нужно потому, что далее здесь будет файл `.env` с паролями.

Все следующие команды на сервере выполняйте из `/srv/imperium`. Файлы `docker-compose.yml` и `.env` находятся в этом каталоге, а не уровнем выше.

3.2. Архив и контрольная сумма

Задайте версию в текущем сеансе SSH:

```
IMPERIUM_VERSION="1.0.0"
```

Переменная действует только в текущем сеансе. После нового подключения или команды `sudo -i` задайте её заново, иначе в адресах загрузки окажется пустая версия.

Скачайте архив и файл контрольной суммы, затем проверьте сумму:

```
curl -fL -O "https://dist.identity.ru/imperium/$IMPERIUM_VERSION/identity-imperium-$IMPERIUM_VERSION.zip"
curl -fL -O "https://dist.identity.ru/imperium/$IMPERIUM_VERSION/identity-imperium-$IMPERIUM_VERSION.zip.sha256"
sha256sum -c "identity-imperium-$IMPERIUM_VERSION.zip.sha256"
```

Проверка должна завершиться без ошибки и вывести:

```
identity-imperium-1.0.0.zip: OK
```

Для выпуска 1.0.0 (сборка от 16.09.2026) ожидаются размер архива 786 682 643 байта и контрольная сумма:

```
fef6b1bb181ea883bfbaa9c577768addd28e70aa88fd421195b68fffd0bdda6f
```

При несовпадении суммы не распаковывайте архив. Повторно получите файлы у поставщика и выясните причину расхождения.

3.3. Распаковка

Убедитесь, что `docker-compose.yml`, `.env` и `imperium.tar` ещё не существуют в выбранном каталоге. Затем распакуйте архив без перезаписи имеющихся файлов:

```
unzip -n "identigy-imperium-$IMPERIUM_VERSION.zip"
ls -la docker-compose.yml .env imperium.tar
```

Все три файла должны находиться непосредственно в `/srv/imperium`. Ключ `-n` не обновляет старые файлы: если они уже были в каталоге, остановитесь и разберите существующую установку отдельно.

4. Импорт образов и состав поставки

4.1. Образ приложения

Загрузите образ в Docker:

```
sudo docker load -i imperium.tar
sudo docker image ls --format '{{.Repository}}:{{.Tag}}' | grep imperium
```

Для этого выпуска ожидается образ:

```
cr.yandex/crpaq0dtru788aippcj0/imperium:4.8.10-260907-alpine
```

Тег образа в выводе должен совпадать с тегом в `docker-compose.yml` (службы `data_init` и `imperium_server`). Если на сервере уже есть другие образы со словом `imperium` в имени, в списке будет несколько строк: сверяйте именно эту.

Проверьте архитектуру образа:

```
sudo docker image inspect cr.yandex/crpaq0dtru788aippcj0/imperium:4.8.10-260907-alpine --format '{{.Architecture}}'
```

Ожидается `amd64`. Она должна совпадать с архитектурой сервера из раздела 1.1.

4.2. Образ PostgreSQL

Служба базы данных использует отдельный образ `pgvector/pgvector:pg17-trixie`: PostgreSQL 17 с расширением для векторного поиска. Для его получения нужен доступ к Docker Hub или согласованное зеркало реестра.

Загрузите только образ базы данных, не запуская контейнеры:

```
sudo docker compose pull imperium_data
sudo docker image ls pgvector/pgvector:pg17-trixie
```

При ошибке загрузки не переходите к запуску. Для сервера без доступа к реестру поставщик должен подготовить отдельный согласованный способ доставки этого образа.

4.3. Компоненты и лицензии

Компонент	Версия в поставке 1.0.0
Образ приложения	4.8.10-260907-alpine
Ядро (midPoint в сборке поставщика)	4.8.10, сборка 260903
Модуль Nuntius AI	4.8.10-260903
Java внутри образа	Eclipse Temurin JDK 21.0.12+8
Базовая ОС образа	Alpine Linux 3.24.1
Внешняя база данных	PostgreSQL 17, образ pgvector/pgvector:pg17-trixie (на 15.09.2026 – PostgreSQL 17.11, расширение vector 0.8.6)

Тег pg17-trixie не закрепляет младшую версию: издатель образа обновляет его при выходе исправлений PostgreSQL и pgvector. Поэтому на вашем сервере младшие версии могут быть новее указанных в таблице. Фактические версии покажет проверка из раздела 7.3.

Тексты лицензий находятся внутри образа в /opt/imperium: LICENSE, LICENSE.AL, LICENSE.EUPL, LICENSE.CC, NOTICE. Версия ОС внутри контейнера не меняет требования к ОС сервера.

5. Настройка учётных данных до первого запуска

5.1. Проверка новой установки

Убедитесь, что на сервере нет прежнего проекта Imperium и его рабочих томов:

```
sudo docker compose ps -a
sudo docker volume ls --filter label=com.docker.compose.project=imperium
```

Если проект уже запускался или у него есть данные, не применяйте порядок первичной настройки к существующей базе. Параметры из .env применяются только при создании пустой базы и не меняют пароли уже созданных учётных записей.

5.2. Пароли в файле .env

Поставляемый docker-compose.yml читает пароли из файла .env; без него запуск завершится ошибкой не задан IMPERIUM_DB_PASSWORD - создайте файл .env. В поставляемом .env записаны значения-заглушки, известные всем получателям дистрибутива. Замените их до первого запуска.

Подготовьте в менеджере паролей два разных пароля:

Переменная	Назначение	Требования
<code>IMPERIUM_DB_PASSWORD</code>	пароль учётной записи базы данных <code>imperium</code>	не менее 14 символов: латинские буквы обоих регистров, цифры, символы <code>_</code> , <code>-</code> , <code>.</code> ; не должен содержать <code>\$</code> , <code>"</code> , <code>'</code> и пробелы
<code>IMPERIUM_ADMIN_PASSWORD</code>	первичный пароль учётной записи <code>administrator</code>	от 8 до 14 символов: латинские буквы обоих регистров и цифры; из специальных допустимы только <code>!\$%() +, - . : ; < > ? @ [] ^ _ \ {</code>

Верхняя граница для пароля администратора задана политикой паролей продукта. Нарушение этой границы не выглядит как ошибка запуска: контейнеры поднимутся, приложение станет доступно, страница входа откроется, но **учётная запись** `administrator` **не будет создана**, и войти в систему будет невозможно. Признак такой установки и порядок исправления — в разделе 7.1. Пароли должны различаться.

Откройте файл в редакторе и замените значения:

```
vi .env
```

Убедитесь, что значений из поставки не осталось:

```
grep -n "ASDF.lkj45679\|Qwerty.12345" .env
```

Команда не должна вывести ни одной строки. Если вывод есть, значения из поставки ещё не заменены.

В том же файле уже задана размерность векторов `VECTOR_DIMENSION=768`. Её меняют только в случае, описанном в разделе 6.1.

Проверьте права на файл: они должны быть `600`, иначе выполните `chmod 600 .env`. Не выводите содержимое `.env` через `cat`, не добавляйте его в системы контроля версий и не прикладывайте к обращениям в поддержку.

5.3. Доступ к порту базы данных

В секции `ports` службы `imperium_data` файла `docker-compose.yml` замените публикацию `5432:5432` привязкой к локальному адресу:

```
ports:
  - "127.0.0.1:5432:5432"
```

Этот фрагмент заменяет только `ports` у базы данных. Сети, тома, зависимости и параметры других служб сохраняйте. Приложение и служба начальной подготовки обращаются к базе данных через внутреннюю сеть Docker, поэтому публикация порта им не нужна.

Если база должна быть доступна с другого узла, её сетевую схему и разрешённые адреса согласуйте отдельно. Для первого запуска внешняя публикация базы не нужна.

5.4. Проверка конфигурации

Проверьте синтаксис без вывода значений:

```
sudo docker compose config --quiet
sudo docker compose config --services
sudo docker compose config --images
```

Первая команда при успехе ничего не печатает. В списке служб должны быть `imperium_data`, `data_init` и `imperium_server`. В списке образов должны быть образ приложения и образ PostgreSQL из раздела 4. Образ приложения выводится дважды, потому что его используют две службы.

Эти команды проверяют структуру файла `docker-compose.yml` и наличие переменных в `.env`. Они не доказывают успешный вход или наличие таблиц.

5.5. Работа за обратным прокси

Если доступ по HTTPS будет обеспечивать обратный прокси на сервере (например, nginx), сообщите приложению, что оно работает за прокси. Создайте рядом с `docker-compose.yml` файл `docker-compose.override.yml`:

```
services:
  imperium_data:
    ports: !override []
  imperium_server:
    ports: !override
    - "127.0.0.1:8080:8080"
    environment:
      - MP_SET_server_forward-headers-strategy=native
```

Этот файл делает три вещи. Он снимает публикацию порта базы данных. Он оставляет приложение доступным только с самого сервера, на порту `8080`, и освобождает порты `80` и `443` для прокси. И он включает учёт заголовков прокси.

Без параметра `MP_SET_server_forward-headers-strategy=native` приложение формирует перенаправления на `http://` и выдаёт cookie сеанса без флага `Secure`. Тер `!override` требует Docker Compose версии 2.24 и выше. Прокси должен передавать заголовки `X-Forwarded-Proto`, `X-Forwarded-For`, `X-Forwarded-Host` и `Host`.

Приложение доверяет этим заголовкам, только если прокси обращается к нему с того же сервера или из внутренней сети: адреса `127.0.0.1`, `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`. Прокси на отдельном узле с внешним адресом приложение не распознает, и cookie останется без флага `Secure`. Настройка самого прокси и сертификата описана в «Руководстве администратора».

Для установки в закрытом сегменте без HTTPS этот шаг пропустите.

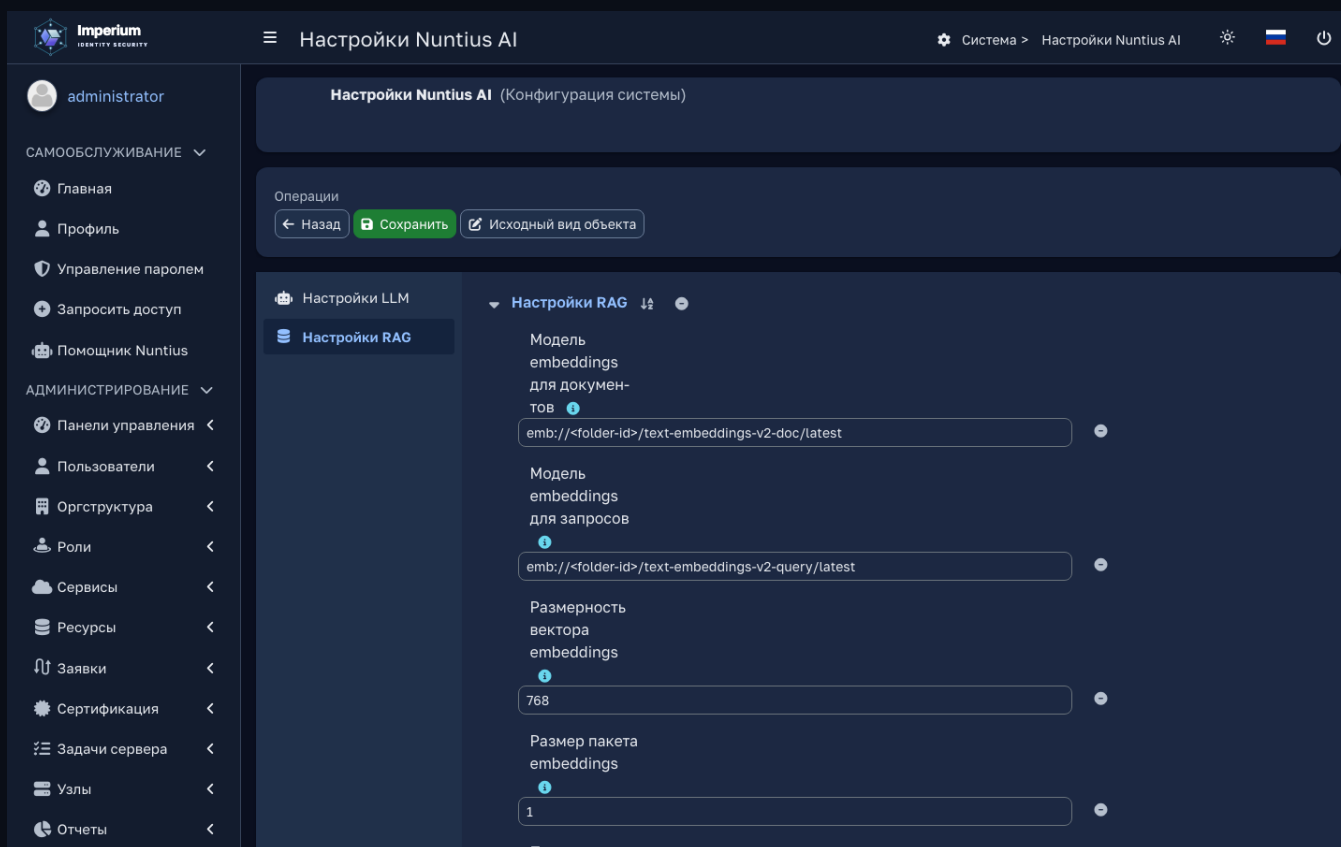
6. Подготовка Nuntius AI

Nuntius AI — встроенный помощник: диалог с языковой моделью и поиск по подготовленной базе знаний. В левом меню пользователя он называется «Помощник Nuntius». Настройки помощника выполняет администратор после установки по «Руководству администратора»; доступность функции зависит от настроек и прав.

6.1. Размерность векторов

Модель embeddings преобразует текст в вектор — набор чисел для поиска. Число значений в векторе должно совпадать с размерностью столбцов базы данных.

При первой инициализации скрипты `rag.sql` и `ai-examples.sql` создают столбцы `vector(N)` в таблицах `mp_docs_chunks` и `mp_ai_examples`. Значение `N` берётся из переменной `VECTOR_DIMENSION` в `.env`; по умолчанию `768`. Выберите модель embeddings с такой же размерностью. После первого входа укажите то же значение в настройках помощника: «Конфигурация» → «Система» → «Настройки Nuntius AI» → вкладка «Настройки RAG» → поле «Размерность вектора embeddings». Пока значение не задано, поле скрыто: нажмите «Показать пустые поля».



На снимке поля уже заполнены для примера. Вместо `<folder-id>` в адресах моделей указывается идентификатор вашего каталога Yandex Cloud.

Для проверенной конфигурации с Yandex AI Studio (модели `text-embeddings-v2-doc` и `text-embeddings-v2-query`) оставьте `768`. Для другой модели задайте её размерность до первого запуска: после создания таблиц изменение `VECTOR_DIMENSION` структуру не меняет, потребуется пересоздание таблиц.

Расхождение размерности даёт ошибку вида `mp_docs_chunks.embedding is vector(768) while embeddingsDimension is <N>`. До его устранения поиск по базе знаний не работает.

6.2. Когда создаются таблицы

При подготовке пустой базы данных служба `data_init` выполняет SQL-скрипты: создаёт основные таблицы, таблицы аудита и таблицы Nuntius AI. Скрипты находятся внутри образа в `/opt/imperium/var/sql/`. Создавать одноимённый каталог на сервере не требуется.

Вызовы этих SQL выполняются в ветке начальной подготовки, а не при каждом запуске. Если база уже работает, но таблиц AI в ней нет, требуется отдельный разбор схемы по «Руководству администратора».

6.3. Корпус документации

Поиск по документации работает по файлам из каталога `/opt/imperium/var/ragFiles` внутри контейнера приложения. В корне каталога должен быть файл `sources.json`, где каждому подкаталогу сопоставлен адрес источника. Индексируются только подкаталоги, перечисленные в этом файле.

Перед запуском задачи индексации сверьте имена подкаталогов с ключами в `sources.json`: при несовпадении задача завершится успешно, но фрагментов не будет. Порядок построения индекса — в «Руководстве администратора».

7. Первичный запуск и проверка

7.1. Запуск служб

Если приложение запустилось и страница входа открывается, но войти не удаётся ни с одним паролем, проверьте журнал приложения на нарушение политики паролей:

```
sudo docker compose logs imperium_server | grep -i "does not satisfy the policies"
```

Строка вида `Provided password does not satisfy the policies: Maximal size (14) was exceeded` означает, что пароль администратора в файле `.env` длиннее допустимого. Учётная запись `administrator` при этом не создаётся, и система остаётся без администратора. Исправление: задать пароль по требованиям раздела 5.2, удалить проект вместе с томами и выполнить первичную установку заново. Данных в такой установке ещё нет, терять нечего.

После проверки параметров запустите систему:

```
sudo docker compose up -d
sudo docker compose ps -a
```

Ключ `-d` оставляет контейнеры работать в фоне. Первичная подготовка может занять несколько минут.

Ожидаемый результат:

- служба базы данных `imperium_data` работает (`Up`);

- служба начальной подготовки `data_init` завершилась с кодом `0` (`Exited (0)`); её контейнер называется `imperium-data-init-1`;
- служба приложения `imperium_server` запустилась после неё (`Up`).

Ошибка `data_init` блокирует штатный запуск приложения. Не обходите эту зависимость.

При каждом следующем `docker compose up -d` служба `data_init` запускается снова. На уже подготовленной базе она выводит `Repository init is not needed...` и завершается, данные не меняются.

7.2. Запуск приложения и порт базы данных

Найдите сообщение запуска приложения:

```
sudo docker compose logs --tail 1000 imperium_server | grep -F 'Started MidPointSpringApplication'
```

Если сообщения нет, подождите одну-две минуты и повторите команду. Если его нет и через пять минут, проверьте состояние служб и разберите причину ошибки. Само состояние `Up` не подтверждает готовность веб-интерфейса.

Узнайте состояние встроенной проверки работоспособности:

```
sudo docker inspect imperium_server --format '{{.State.Health.Status}}'
```

Через одну-две минуты после сообщения о запуске ожидается `healthy`. Значение `starting` означает, что проверка ещё идёт.

Проверьте публикацию базы данных:

```
sudo docker compose port imperium_data 5432
```

При настройке по разделу 5.3 ожидается `127.0.0.1:5432`. При настройке по разделу 5.5 команда выводит `invalid IP:0`: порт не опубликован, это ожидаемый результат. Любой другой адрес требует проверки сетевых параметров до допуска пользователей.

7.3. Таблицы векторного поиска

Администратор базы данных подключается к базе `imperium` своим разрешённым способом, например `sudo docker exec -it imperium_data psql -U imperium`. Выполните запросы только для чтения. Для выхода из `psql` введите `\q`.

```
SELECT extversion
FROM pg_extension
WHERE extname = 'vector';

SELECT c.relname AS table_name,
       format_type(a.atttypid, a.atttypmod) AS column_type
FROM pg_attribute a
JOIN pg_class c ON c.oid = a.attrelid
JOIN pg_namespace n ON n.oid = c.relnamespace
WHERE n.nspname = 'public'
      AND c.relname IN ('mp_docs_chunks', 'mp_ai_examples')
      AND a.attname = 'embedding'
      AND a.attnum > 0
      AND NOT a.attisdropped;
```

Первый запрос должен вернуть установленную версию расширения `vector`. Второй — по строке для каждой таблицы со значением `vector(768)` или заданной вами размерностью. Отсутствие строк или другая размерность требуют отдельного разбора. Не запускайте индексацию, пока схема базы и размерность модели не согласованы.

7.4. Запуск после перезагрузки сервера

В поставляемом `docker-compose.yml` не задана политика перезапуска. После перезагрузки сервера служба Docker запустится, а контейнеры Imperium — нет.

После каждой перезагрузки запускайте систему вручную:

```
cd /srv/imperium
sudo docker compose up -d
```

Затем повторите проверки из разделов 7.1 и 7.2. Автоматический запуск настраивается отдельно по согласованию с поставщиком.

8. Проверка веб-интерфейса

На сервере проверьте HTTP-ответ. При публикации на порту 80 (поставка):

```
curl -I --max-time 20 http://localhost/imperium
```

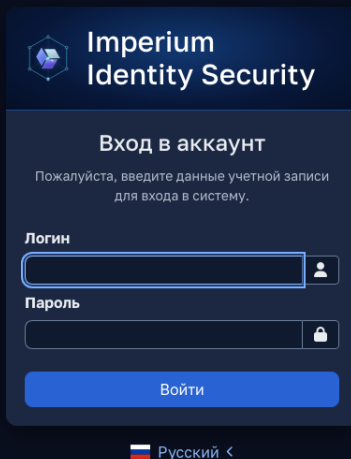
При настройке по разделу 5.5 используйте адрес `http://127.0.0.1:8080/imperium`. Ожидается ответ `HTTP/1.1 302` со строкой `Location: /imperium/`: приложение перенаправляет на страницу входа. Ошибка соединения или ответ `5xx` требуют проверки служб; это не успешная установка.

Для первого входа используйте настроенный HTTPS либо защищённый SSH-туннель. Пример команды на рабочем месте администратора:

```
ssh -L 8080:127.0.0.1:80 <пользователь>@<адрес_сервера>
```

Пока SSH-сеанс открыт, перейдите в браузере на `http://localhost:8080/imperium`. Если локальный порт `8080` занят, выберите другой свободный порт и укажите его в адресе браузера. При настройке по разделу 5.5 в команде туннеля укажите порт сервера `8080` вместо `80`.

Откроется страница входа с полями «Логин» и «Пароль», кнопкой «Войти» и переключателем языка под формой.



На кадрах — фирменное оформление Imperium (тёмная тема); в поставке 1.0.0 оформление стандартное, тема войдёт в следующую поставку.

Прямой адрес `http://<адрес_сервера>/imperium` используйте только в закрытом административном сегменте. До выдачи рабочего доступа настройте HTTPS и согласованные правила сети.

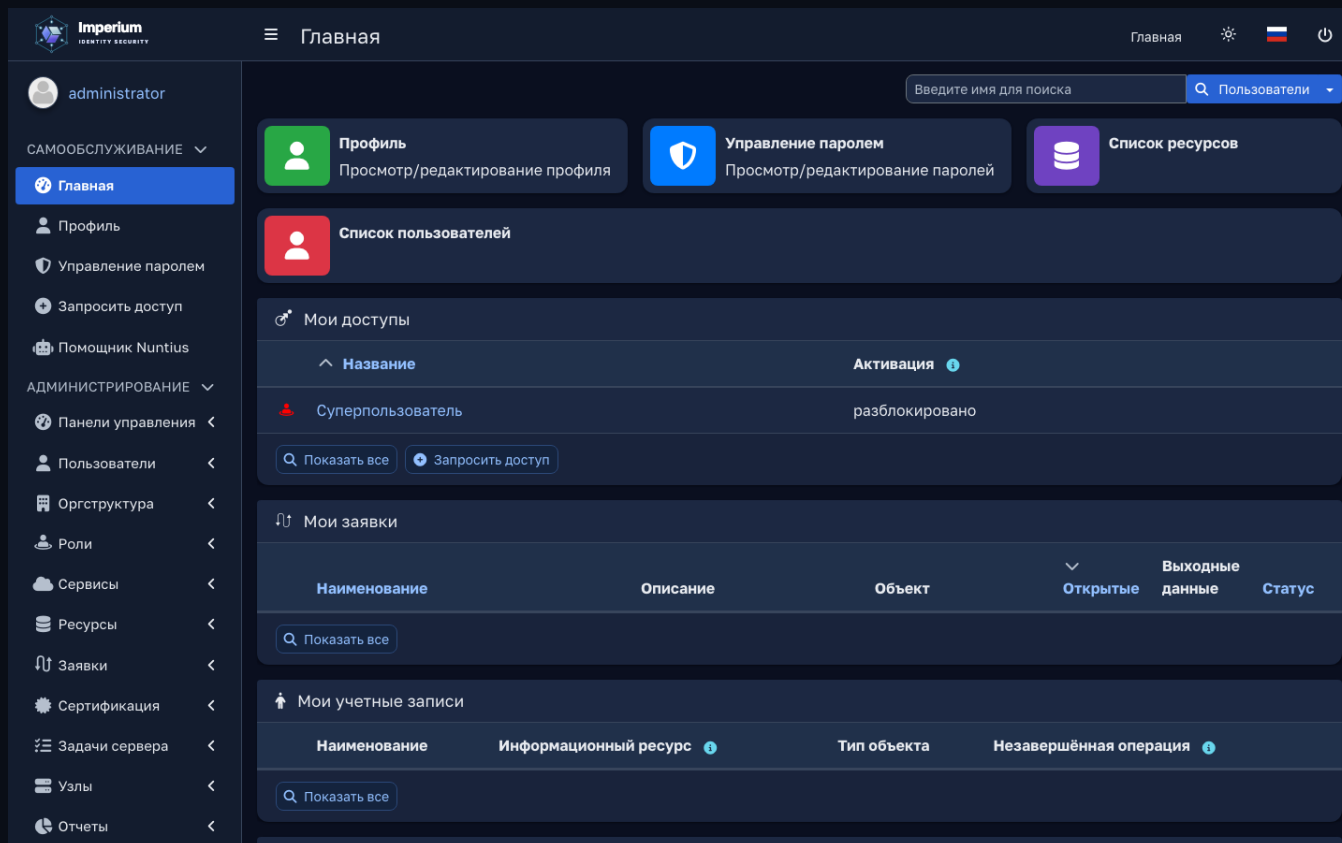
9. Первичный вход

9.1. Учётная запись администратора

Откройте страницу входа и укажите имя `administrator`. Для новой пустой установки используйте первичный пароль, заданный в `IMPERIUM_ADMIN_PASSWORD` файла `.env` до запуска.

Этот параметр применяется при первом создании администратора в новой базе. Повторный запуск с другим значением пароль не меняет. Если вход не удался, остановите повторные попытки: три ошибки за три минуты блокируют учётную запись на 15 минут. Проверьте, была ли установка новой и успешно ли завершилась её подготовка.

После успешного входа откроется страница «Главная» с левым меню и карточками «Профиль», «Управление паролем», «Список ресурсов», «Список пользователей».



Получайте пароль из менеджера паролей. Для его просмотра не выводите содержимое `.env` или окружение контейнера в терминал.

9.2. Смена первичного пароля

Смените первичный пароль сразу после входа: он остаётся в файле `.env` в открытом виде.

1. В левом меню откройте «Самообслуживание» → «Управление паролем». Относительный путь страницы — `/imperium/self/credentials`.
2. В поле «Текущий пароль» введите первичный пароль.
3. Введите новый пароль дважды: в поле «Новый пароль» и в поле повтора под ним. Справа на странице показаны требования: от 8 до 14 символов, строчные и прописные буквы, цифры. Выполненное требование отмечается зелёной галочкой.
4. Нажмите «Сменить пароль». Появится сообщение «Сохранить пароль — Успешно».
5. Выйдите из системы значком выхода в правом верхнем углу и войдите снова с новым паролем.

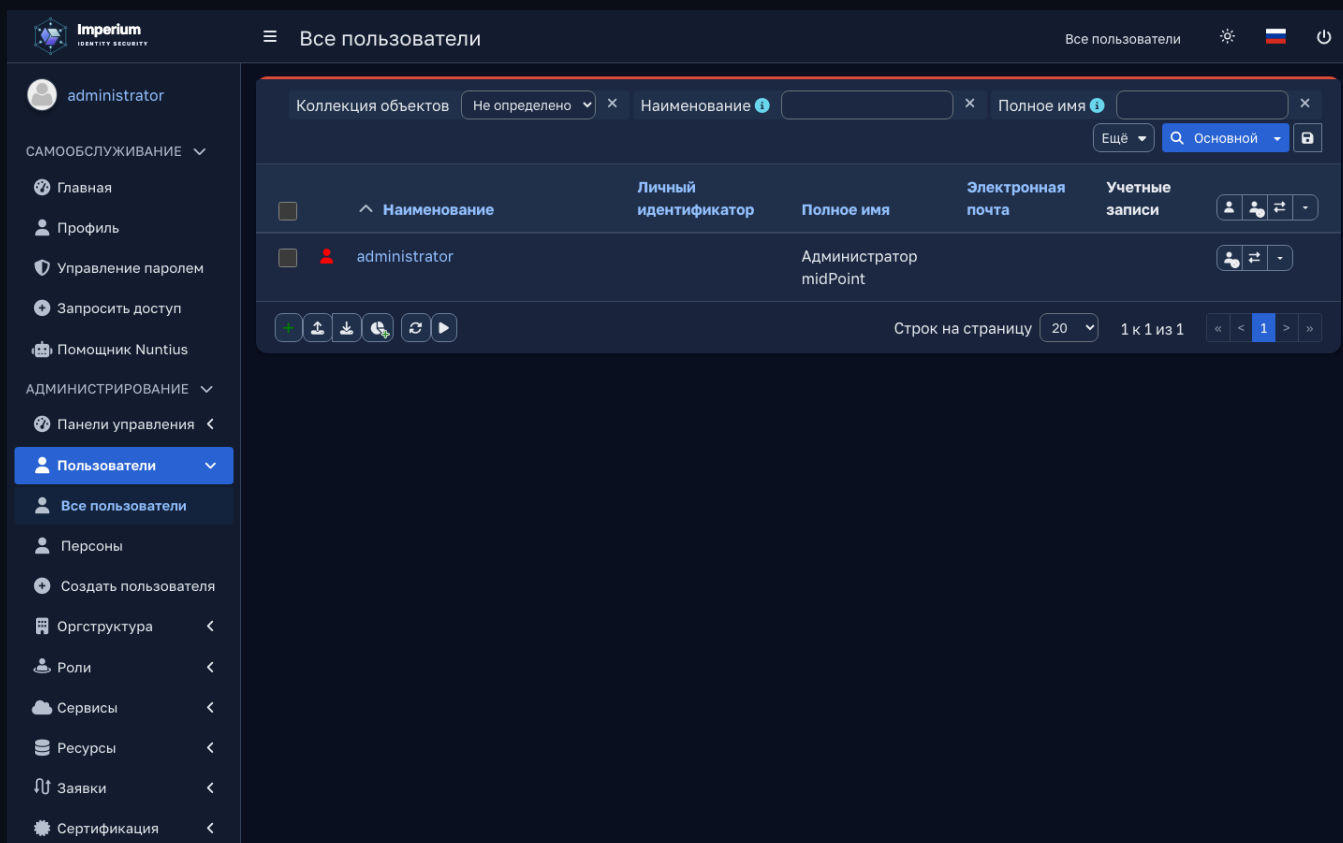
9.3. Именные администраторы и владелец фоновых задач

Заведите именные учётные записи администраторов и проверьте вход под ними:

«Администрирование → Пользователи → Создать пользователя», роль

«Суперпользователь» (`Superuser`) или более узкая. Ежедневную работу ведите под именными записями, чтобы в аудите было видно, кто действовал.

Если вы планируете выключить встроенную `administrator`, сначала переведите фоновые задачи на служебного владельца. Задачи принадлежат учётной записи; задачи выключенной записи перестают выполняться. Заведите служебную запись без пароля (например, `imperium.system`) с правами на выполнение задач и укажите её владельцем в карточках задач: «Администрирование → Задачи сервера». Порядок — в «Руководстве администратора».

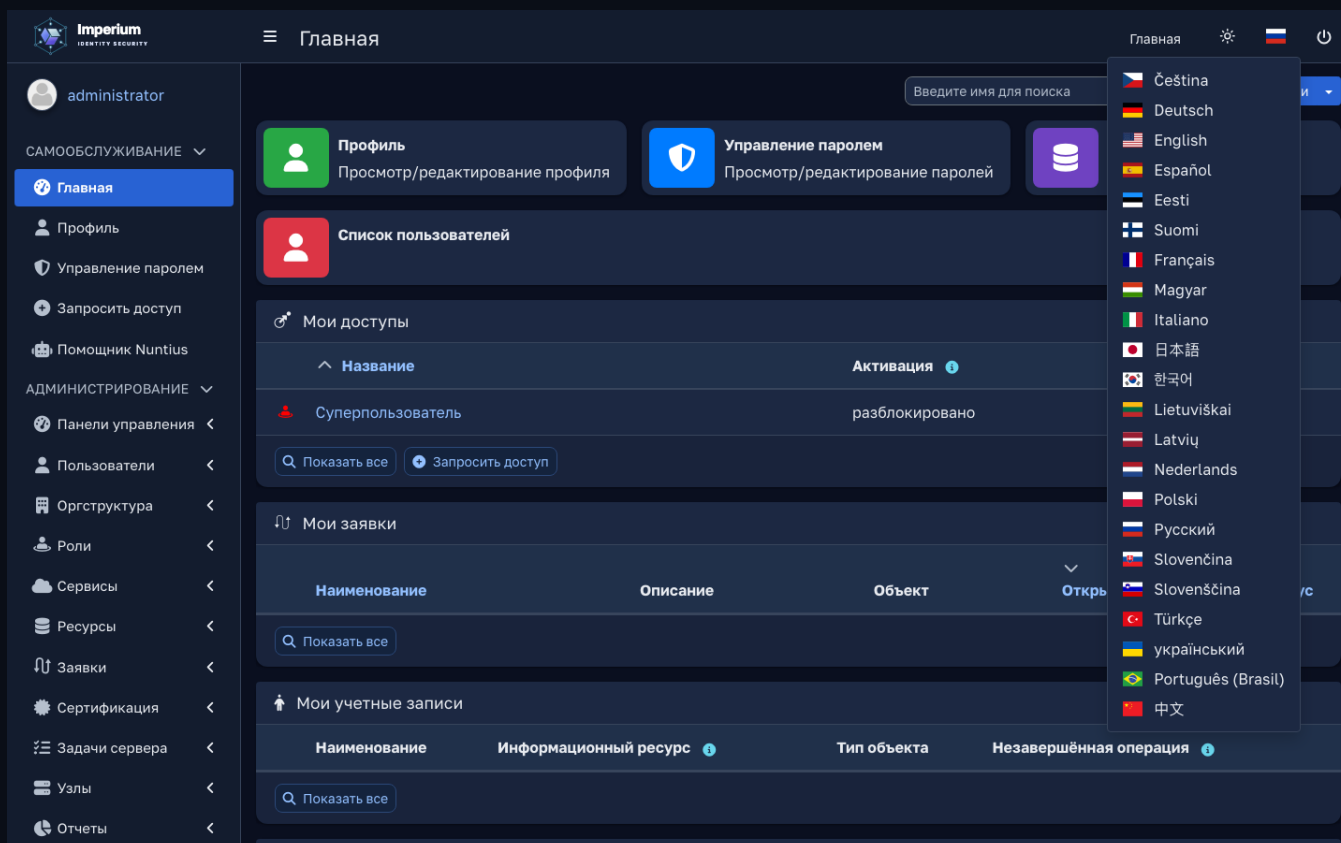


10. Язык веб-интерфейса

Язык интерфейса по умолчанию — русский; он задан параметрами запуска приложения и не зависит от настроек браузера. Порядок выбора: язык, заданный в учётной записи пользователя, затем язык по умолчанию.

Сменить язык в текущем сеансе:

- на странице входа — переключателем языка (значок флага) на форме входа;
- после входа — значком флага в правом верхнем углу: нажмите его и выберите «Русский».



Чтобы закрепить язык за учётной записью, откройте карточку пользователя и заполните поле «Locale» («Локаль») значением `ru`. Значение указывается двухбуквенным кодом языка, при необходимости с кодом страны через подчёркивание: `ru`, `en`, `pt_BR`. Проверьте язык после повторного входа.

11. Консультации по установке

По вопросам установки и настройки обращайтесь к техническим специалистам разработчика:

- Организация: ООО «Айдентиджи».
- Телефон: +7 (495) 157-11-35.
- Электронная почта: imperium-support@identigy.ru.
- Часы работы поддержки: рабочие дни, 9:00–19:00 по московскому времени.

При обращении укажите версию дистрибутива, шаг руководства, на котором возникла ошибка, и вывод команд `sudo docker compose ps -a` и `sudo docker compose logs --tail 200 imperium_server`. Не передавайте файл `.env`.