



АО «ЭЖ «Восток».
ИНН 7705424509, ОГРН 1037739123696.
107045, г. Москва,
Просвирин пер, д. 4
Тел.: +7 495 775-24-97
E-mail: info@vostok-electra.ru

**Ассистент оператора на базе технологий
искусственного интеллекта (LLM)**

Инструкция по установке

г. Москва, 2025

Содержание

1 Глоссарий	3
2 Требования к окружению	4
3 Настройка LLM конвейера.....	5
4 Настройка интеграции с ПО для ЭВМ Amber BPM \ Эмбер БиПиЭм.....	7
5 Настройка OpenWebUI	9
6 Установка моделей.....	10
7 Установка функций.....	12
8 Инструкция по установке платформы AMBER на серверы клиента.....	15
8.1 Перед установкой требуется:	15
8.2 Требования.....	15
8.3 Настройка серверов.....	15
8.4 Установка AMBER.....	17

1 Глоссарий

Термин	Определение \ Пояснение
ПК	Программный комплекс
Система	Ассистент оператора на базе технологий искусственного интеллекта (LLM)
Оператор	Пользователь Системы

2 Требования к окружению

Ассистент оператора на базе технологий искусственного интеллекта (LLM) (далее - Ассистент) предполагает функционирование на отдельном сервере и интеграцию с программой для ЭВМ Amber BPM \ Эмбер БиПиЭм.

Системные требования к серверу Ассистента:

- Минимальные требования:
 - o CPU: 4 cores 2.0 ГГц
 - o RAM: 128 Гб
 - o HDD: 200 Гб
 - o GPU: NVIDIA TESLA A10
- Рекомендуемые требования:
 - o CPU: 4 cores 2.5 – 3.0 ГГц
 - o RAM: 256 Гб
 - o HDD: 200 Гб (предпочтительно SSD)
 - o GPU: NVIDIA TESLA A100

Требования к ПО на сервере Ассистента:

- ОС – Astra Linux \ RedOS \ Ubuntu \ иные Linux OS
- LLM Конвейер – Open WebUI (<https://github.com/open-webui/open-webui>)
- LLM Движок – Ollama (<https://ollama.com/>)

3 Настройка LLM конвейера

Для корректного функционирования LLM конвейера необходимо установить посредством административного интерфейса (доступ осуществляется посредством браузера по адресу **http://<имя сервера>:3000**) Open WebUI (описание установки Open WebUI описано в разделе [Настройка OpenWebUI](#)):

1. LLM Модели (установка моделей описана в разделе [Установка моделей](#)):
 - paraphrase-multilingual
 - xitao/bge-reranker-v2-m3
2. Функции (установка функций описана в разделе [Установка функций](#)):
 - Yandex GPT Manifold Pipe

Для корректной работы системы с технологий RAG необходимо в разделе «Панель администратора» \ «Настройки» \ «Модели» \ «Документы»:

1. Указать модель встраивания paraphrase-multilingua
2. Включить гибридный режим поиска и установить модель реранжирования xitao/bge-reranker-v2-m3
3. Установить системный промпт для RAG:

Ты – опытный оператор службы поддержки клиентов.
Твоя задача – ответить клиенту на его запрос, используя предоставленный контекст из базы знаний компании.
Контекст:
[context]
Запрос клиента:
[query]
Ответ вежливо, профессионально и понятно, соблюдая следующую структуру:
1. Чёткий и конкретный ответ на основе указанного выше контекста.
Ответ:

4. Создать новую модель (и присвоить уникальное имя, например, vostok-model-122424) на базе модели Yandex GPT Manifold Pipe и указать следующий промпт:

Ты – опытный оператор сервисной службы поддержки клиентов. Тебе нужно сформировать шаблон ответа на запрос клиента, используя подход Retrieval-Augmented Generation (RAG).

Составь шаблон ответа, который должен:

1. Кратко и понятно отвечать на запрос клиента на основе предоставленной информации (RAG).

Структурируй ответ следующим образом:

1. Чёткий ответ на запрос с опорой на данные из источника (RAG).

Формат ответа:

Текстовый шаблон, подходящий для стандартной переписки в чате.

Обязательно учти:

1. Ответ должен звучать естественно, тепло и профессионально.
2. Он должен быть универсальным, легко адаптируемым под разные типы запросов.
3. Ответ должен помогать клиенту и демонстрировать качество сервиса.
4. Если в документах отсутствует информация по пользовательскому запросу, то необходимо вывести "[НЕТ ИНФОРМАЦИИ]"

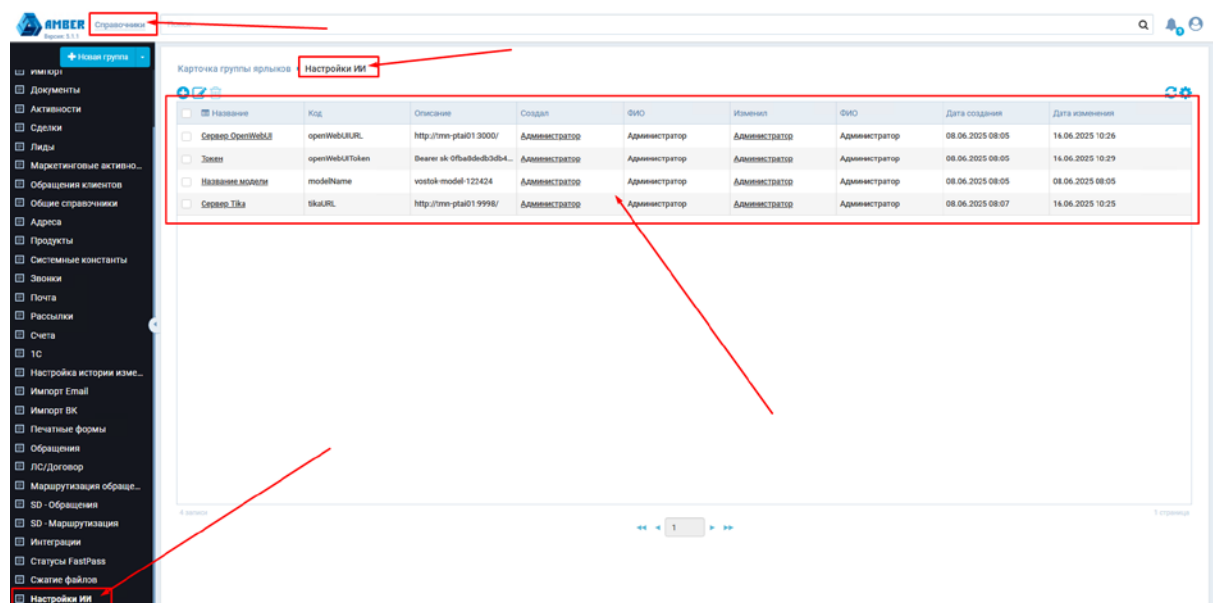
4 Настройка интеграции с ПО для ЭВМ Amber BPM \Эмбер БиПиЭм

Для настройки интеграции в Amber необходимо:

1. Авторизоваться в системе Amber в пользовательском интерфейсе (адрес необходимо уточнить у администраторов системы, адрес выглядит как <https://amber.local>) пользователем с полномочиями Администратор.

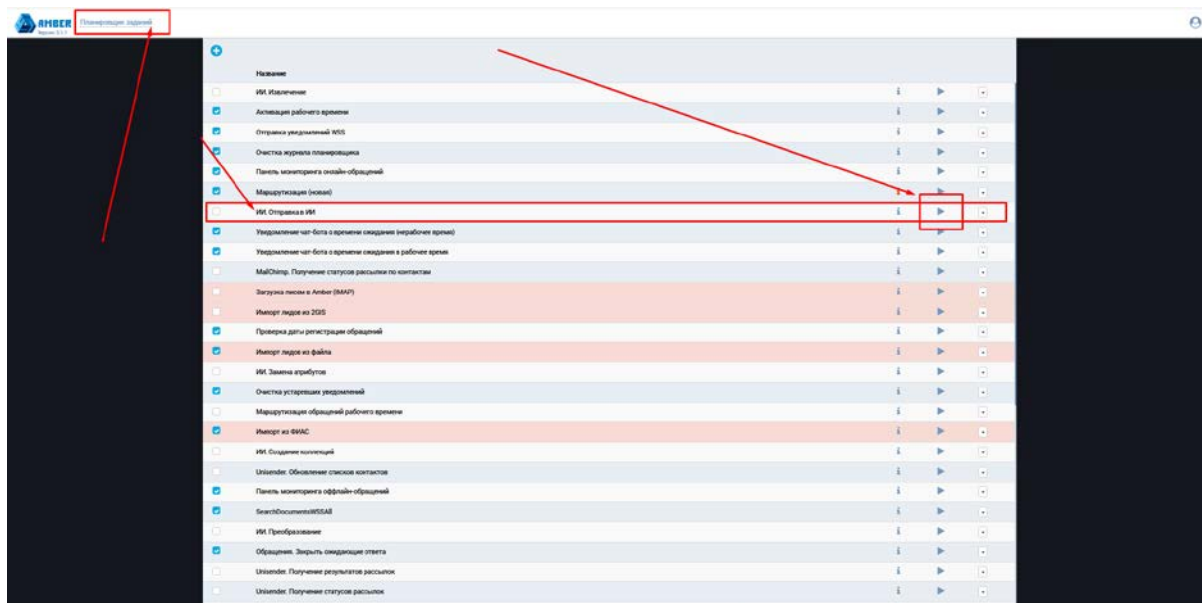
2. Переключиться в режим «Справочники» и перейти в раздел «ИИ Ассистент». В доступных настройках ввести:

- Имя сервера OpenWebUI (адрес вида <http://<имя сервера ИИ>:3000>)
- API ключ (ниже описана процедура получения API ключа)
- Название модели: `vostok-model-122424`
- Адрес сервера Confluence (адрес необходимо запросить у администратора)



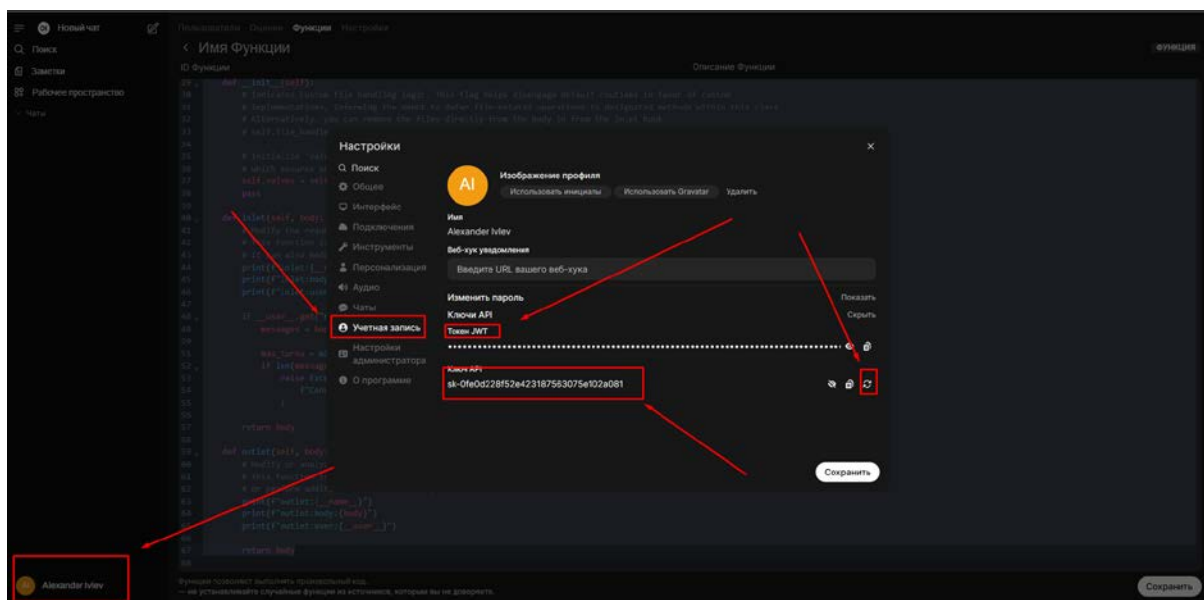
3. Авторизоваться в системе Amber в административном интерфейсе (адрес необходимо уточнить у администраторов системы, адрес выглядит как <https://admin-amber.local>) пользователем с полномочиями Администратор.

4. Перейти в административную панель и в разделе планировщика задач запустит задачу «ИИ. Импорт в ИИ».



По завершению работы процесса (порядка 10 часов) система будет готова к работе.

Примечание. Для получения API ключа необходимо в интерфейсе Open WebUI перейти в: «Настройки» \ «Учетная запись» \ «Токен JWT» \ «Создать новый ключ».



5 Настройка OpenWebUI

Для установки конвейера OpenWebUI необходимо:

1. Авторизоваться в ОС, где будет осуществляться развертывание Системы, используя SSH или терминальное подключение
2. Если не установлен контейнер docker – установить, введя следующие команды:

```
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli
containerd.io
sudo usermod -aG docker $USER
sudo reboot
```

3. Установить OpenWebUI, введя команду:

```
docker run -d -e ENV='dev' -p 3000:8080 --gpus all --add-
host=host.docker.internal:host-gateway -v open-
webui:/app/backend/data --name open-webui --restart always
ghcr.io/open-webui/open-webui:cuda
```

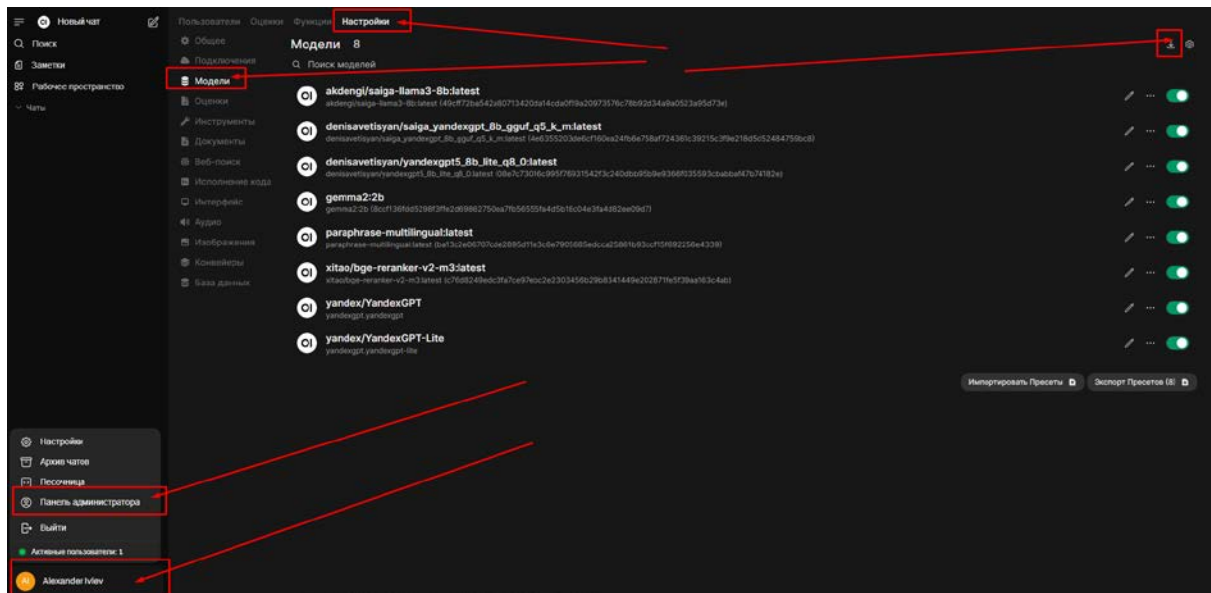
4. После установки необходимо посредством браузера зайти по адресу сервера, где осуществляется развертывание Системы, указав порт 3000, т.е. адрес должен выглядеть как **http://<имя сервера>:3000**
5. При первом входе OpenWebUI запросит создание Администратора системы, для чего необходимо будет ввести имя пользователя и пароль.
6. Для работы моделей формата Ollama необходимо установить движок Ollama:

```
docker run -d --gpus=all -v ollama:/root/.ollama -p
11434:11434 --name ollama ollama/ollama
```

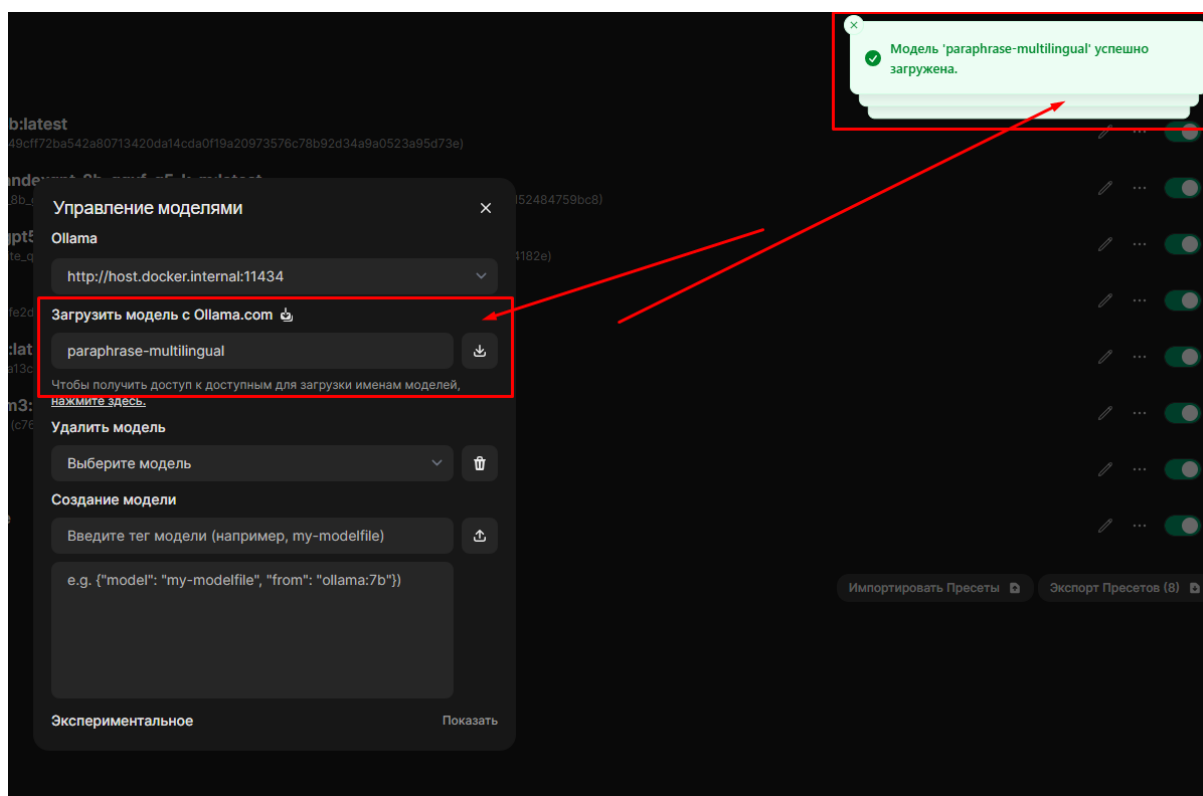
6 Установка моделей

Для установки моделей в OpenWebUI необходимо:

1. Авторизоваться в Open WebUI
2. Перейти в раздел «Панель администратора» \ «Настройки» \ «Модели» \ «Управление моделями»



3. В открывшемся окне необходимо указать наименование модели в поле «Загрузить модель с Ollama.com»

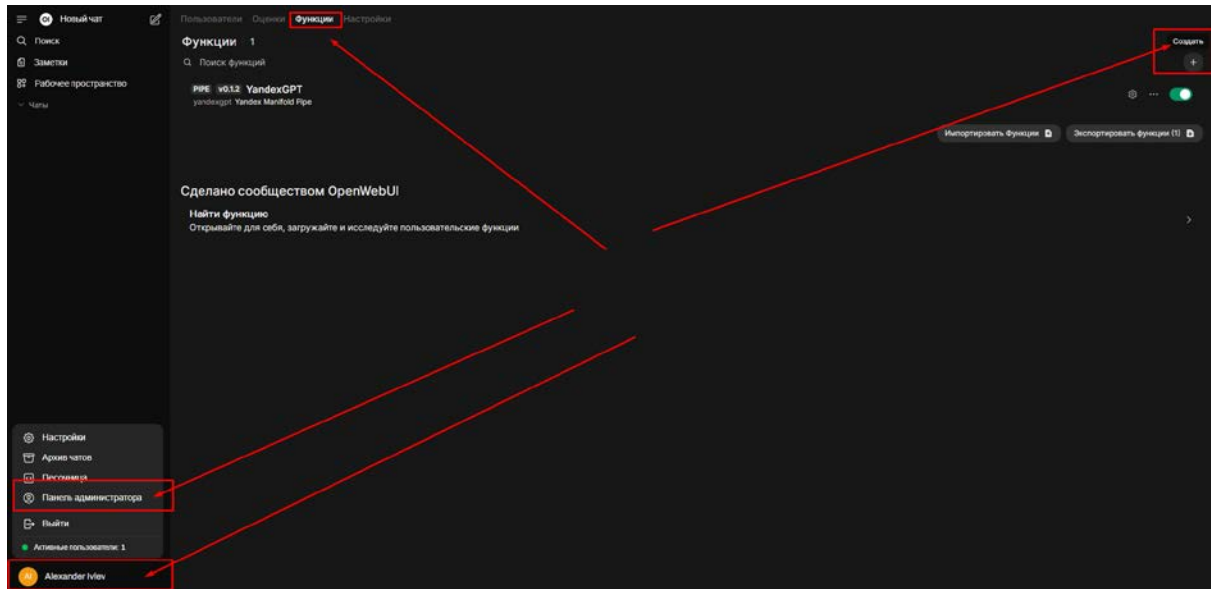


4. После успешной загрузки система выведет соответствующее сообщение.

7 Установка функций

Для установки функций в OpenWebUI необходимо:

1. Авторизоваться в Open WebUI
2. Перейти в раздел «Панель администратора» \ «Настройки» \ «Функции» \ «Создать»



3. В открывшемся окне необходимо ввести код функции.
4. Код функции Yandex GPT Manifold Pipe приведен ниже:

```
from pydantic import BaseModel, Field
from typing import Optional
```

```
class Filter:
    class Valves(BaseModel):
        priority: int = Field(
            default=0, description="Priority level for the
filter operations."
        )
        max_turns: int = Field(
            default=8, description="Maximum allowable
conversation turns for a user."
        )
```

```

pass

class UserValves(BaseModel):
    max_turns: int = Field(
        default=4, description="Maximum allowable
conversation turns for a user."
    )
    pass

    def __init__(self):
        # Indicates custom file handling logic. This flag
        helps disengage default routines in favor of custom
        # implementations, informing the WebUI to defer
        file-related operations to designated methods within this
        class.
        # Alternatively, you can remove the files directly
        from the body in from the inlet hook
        # self.file_handler = True

        # Initialize 'valves' with specific
        configurations. Using 'Valves' instance helps encapsulate
        settings,
        # which ensures settings are managed cohesively
        and not confused with operational flags like
        'file_handler'.
        self.valves = self.Valves()
        pass

    def inlet(self, body: dict, __user__: Optional[dict] =
None) -> dict:
        # Modify the request body or validate it before
        processing by the chat completion API.
        # This function is the pre-processor for the API
        where various checks on the input can be performed.
        # It can also modify the request before sending it
        to the API.

```

```

        print(f"inlet:{__name__}")
        print(f"inlet:body:{body}")
        print(f"inlet:user:{__user__}")

        if __user__.get("role", "admin") in ["user",
"admin"]:
            messages = body.get("messages", [])

            max_turns = min(__user__["valves"].max_turns,
self.valves.max_turns)
            if len(messages) > max_turns:
                raise Exception(
                    f"Conversation turn limit exceeded.
Max turns: {max_turns}"
                )

        return body

    def outlet(self, body: dict, __user__: Optional[dict]
= None) -> dict:
        # Modify or analyze the response body after
processing by the API.
        # This function is the post-processor for the API,
which can be used to modify the response
        # or perform additional checks and analytics.
        print(f"outlet:{__name__}")
        print(f"outlet:body:{body}")
        print(f"outlet:user:{__user__}")

        return body

```

5. Нажать кнопку «Сохранить»

8 Инструкция по установке платформы AMBER на серверы клиента

8.1 Перед установкой требуется:

1. Сервер для приложения и базы данных:
 - о **Сервер приложения:** ОС Linux, роль Nginx.
 - о **Сервер базы данных:** ОС Linux или Windows, установка PostgreSQL 14.
2. 3 доменных имен для приложения:
 - о `amber.ВАШ_ДОМЕН.ру` — клиентская часть.
 - о `admin-amber.ВАШ_ДОМЕН.ру` — административная часть.
 - о `api-amber.ВАШ_ДОМЕН.ру` — API.
3. Выпустить SSL-сертификаты для указанных доменов (не обязательно, но желательно).

Примечание: Установка рассматривается на примере Debian 11.

8.2 Требования

- Установлен .NET 7.
- Установлен PostgreSQL 14.
- Установлен Nginx.
- Учётные данные для подключения к СУБД.
- Учётные данные для запуска сервиса.
- Адреса для основного сайта, интерфейса администратора и REST API.
- SSL-сертификаты для настройки HTTPS.

8.3 Настройка серверов

Рассмотрим установку серверов на базе Debian 11

Установка Nginx

1. Обновите индексы пакетов:

```
sudo apt update && sudo apt upgrade -y
```

2. Установите Nginx:

```
sudo apt install nginx
```

3. Проверьте статус Nginx:

```
sudo systemctl status nginx
```

4. Запустите Nginx и настройте автозапуск:

```
sudo systemctl start nginx
sudo systemctl enable nginx
```

Установка .NET 7

1. Добавьте репозиторий Microsoft:

```
wget https://packages.microsoft.com/config/debian/11/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
```

2. Установите необходимые пакеты:

```
sudo apt install -y apt-transport-https
```

3. Установите .NET 7:

```
sudo apt install -y dotnet-sdk-7.0
```

4. Проверьте версию .NET:

```
dotnet --version
```

Примечание:

Убедитесь, что вы выполнили все команды с правами администратора (sudo).

После установки, .NET будет доступен глобально на вашей системе, и вы сможете использовать dotnet для создания, компиляции и запуска .NET приложений.

Если у вас уже установлен другой SDK .NET, вы можете использовать команду

```
sudo apt remove dotnet-sdk-<version>
```

для удаления старой версии перед установкой .NET 7

Установка PostgreSQL 14

1. Установите пакеты:

```
sudo apt install lsb-release curl wget gnupg2
```

2. Добавьте репозиторий PostgreSQL:

Добавление репозитория

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt
$(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

Импорт ключей и обновление пакетов

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add -
sudo apt update && sudo apt upgrade -y
```

3. Установите PostgreSQL 14:

```
sudo apt install postgresql-14
```

4. Проверьте версию PostgreSQL:

```
sudo -u postgres psql -c "SELECT version();"

Чтобы поменять пароль от учетки postgres
\password postgres
```

5. Измените конфигурацию PostgreSQL:

- o Откройте файл /etc/postgresql/14/main/postgresql.conf и измените значение listen_addresses на '*'.
- o Откройте файл /etc/postgresql/14/main/pg_hba.conf и меняем следующие строки:

```
# IPv4 local connections:
host    all             all            0.0.0.0/0          md5
# IPv6 local connections:
host    all             all            ::0/0              md5
```

6. Так же можно воспользоваться утилитой для оптимизации конфигурационного файла PostgreSQL (postgresql.conf)

Ссылка на утилиту: <https://www.pgconfig.org/>

В ней заполняем конфигурацию сервера, на котором установлен PostgreSQL

После ввода конфигурации сервера, экспортируем конфиг и адаптируем конфиг PostgreSQL согласно полученным рекомендациям от утилиты. Данные настройки помогут оптимизировать PostgreSQL к вашему серверу.

7. Перезапустите PostgreSQL:

```
sudo service postgresql restart
```

8.4 Установка AMBER

Получение артефактов сборки

Получите артефакты сборки для развёртывания (дистрибутив программы):

- o **Database** — эталонные базы данных, необходимые для начальной настройки и работы платформы.

- o **Deployment** — шаблоны конфигурации Nginx и systemd, используемые для автоматизации развёртывания и настройки серверов.
- o **Service** — сервер приложения, обеспечивающий основную бизнес-логику платформы.
- o **Web/Admin** — интерфейс администратора, позволяющий управлять системой и пользователями.
- o **Web/Site** — основной сайт для взаимодействия с конечными пользователями.
- o **Workflow** — библиотеки для разработки и настройки бизнес-процессов, обеспечивающие гибкость в автоматизации задач.

Установка и настройка

1. Получить артефакты сборки
2. Получить временную лицензию AMBER
3. Создайте папку `/var/amber/`.
4. Развернуть базы данных из Artifacts/Database
5. Скопируйте артефакты в соответствующие папки:
 - o `Artifacts/Service/* => /var/amber/core`
 - o `Artifacts/Web/Admin/* => /var/amber/admin`
 - o `Artifacts/Web/Site/* => /var/amber/site`
 - o `Artifacts/Workflow/* => /var/amber/workflow`
 - o `Artifacts/Deployment/nginx.conf.template => /var/amber/amber.nginx.conf`
 - o `Artifacts/Deployment/systemd.service.template => /var/amber/amber.service`
6. Скопируйте временную лицензию в `/var/amber/core/Amber.license`. Временная лицензия необходима для первого запуска и будет заменена финальной лицензией после регистрации.
7. Настройте сервер приложений в файле `/var/amber/core/appsettings.json`:
 - o **WebApi.Address** - адрес, который слушает сервер приложений. Например `"http://127.0.0.1:8001/"`
 - o **ConnectionStrings.xbpms** - строка подключения к базе Amber (см. раздел требований)
 - o **ConnectionStrings.scheduler** - строка подключения к базе Amber (см. раздел требований)
 - o **ConnectionStrings.wwfInstanceStore** - строка подключения к базе Amber_WWF (см. раздел требований)
 - o **ConnectionStrings.wwfTracker** - строка подключения к базе Amber_WWF.Tracking (см. раздел требований)
8. Настройте systemd-сервис в файле `/var/amber/amber.service`:
 - o **Unit.Description** — название сервиса (например, "Amber service").
 - o **Service.WorkingDirectory** — рабочий каталог сервера приложений (например, `/var/amber/core`).

- o **Service.ExecStart** — необходимо вызывать dotnet, из установленного пакета .NET 7, с параметром - путь к Amber.Host.dll. Например, "/usr/bin/dotnet /var/amber/core/Amber.Host.dll"
 - o **Service.SyslogIdentifier** - название сервиса для журнала syslog. Например, "Amber service"
 - o **Service.User** — пользователь для запуска сервиса.
9. Настройте Nginx в файле /var/amber/amber.nginx.conf для трёх сайтов: основной, интерфейс администратора, REST API. Настройки включают порты, пути к сертификатам, адреса серверов.
- o В файле хранится конфигурация для 3-х сайтов - основной, интерфейс администратора, restapi
 - o Ниже приводится настройка с HTTPS. Если он не требуется, то нужно везде в server.listen установить "80" и удалить строки с ssl_certificate и ssl_certificate_key
 - o Основной сайт:
 1. server.listen - "443 ssl"
 2. server.ssl_certificate - путь к сертификату (см. раздел требований)
 3. server.ssl_certificate_key - путь к ключу сертификата (см. раздел требований)
 4. server.server_name - адрес основного сайта (см. раздел требований)
 5. server.location.proxy_pass - адрес, который слушает сервер приложений (см. WebApi.Address в файле /var/amber/core/appsettings.json)
 6. location.root - путь к папке с файлами сайта. Например "/var/amber/site"
 - o Интерфейс администратора:
 1. server.listen - "443 ssl"
 2. server.ssl_certificate - путь к сертификату (см. раздел требований)
 3. server.ssl_certificate_key - путь к ключу сертификата (см. раздел требований)
 4. server.server_name - адрес интерфейса администратора (см. раздел требований)
 5. server.location.proxy_pass - адрес, который слушает сервер приложений (см. WebApi.Address в файле /var/amber/core/appsettings.json)
 6. location.root - путь к папке с файлами интерфейса администратора. Например "/var/amber/admin"
 - o Restapi:
 1. server.listen - "443 ssl"
 2. server.ssl_certificate - путь к сертификату (см. раздел требований)
 3. server.ssl_certificate_key - путь к ключу сертификата (см. раздел требований)
 4. server.server_name - адрес restapi (см. раздел требований)
 5. server.location.proxy_pass - адрес, который слушает сервер приложений (см. WebApi.Address в файле /var/amber/core/appsettings.json)
10. Настройте права на папку /var/amber/ для пользователя, под которым будет запущен сервис. Установите права на чтение, запись и выполнение.
11. Включите и запустите systemd-сервис:

```
sudo systemctl enable /var/amber/amber.service  
sudo systemctl start amber.service
```

12. Зарегистрируйте конфигурацию Nginx:

- o Добавьте строку `include /var/amber/amber.nginx.conf;` в раздел `http` файла `/etc/nginx/nginx.conf`.
- o Перезагрузите Nginx:

```
sudo nginx -s reload
```

13. Выпустите финальную лицензию с привязкой к оборудованию:

- o Зайдите в интерфейс администратора, перейдите в раздел "Пользователи", нажмите кнопку "Сгенерировать" для формирования файла запроса лицензии.
- o Отправьте письмо на адрес helpme@amber-soft.ru с указанием организации для которой выпускается лицензионный ключ
- o После проверки данных получите финальную лицензию и скопируйте её в `/var/amber/core/Amber.license`.
- o Перезапустите сервис:

```
sudo systemctl restart amber.service
```

14. Проверьте, что лицензия применена, и сайты работают.