

Инструкция по установке экземпляра ПО

Системные требования

Программное обеспечение:

- Docker Engine версии 20.10 и выше;
- Docker Compose версии 2 и выше;
- Nginx (установлен и запущен на целевом узле);
- действующие SSL/TLS-сертификаты для доменных имён, используемых для доступа к сервисам. Способ выпуска и обновления сертификатов определяется администратором целевой среды.

Аппаратные ресурсы:

- объём оперативной памяти — не менее 4 ГБ;
- объём свободного дискового пространства — не менее 50 ГБ (для хранения Docker-образов, тома `postgres_data` базы данных, каталога `langflow-data`, загружаемых документов базы знаний Open WebUI и журналов контейнеров);
- публичный IP-адрес целевого узла.

Сетевые ресурсы:

- зарегистрированное доменное имя с возможностью управления DNS-записями;
- свободные TCP-порты `80`, `443` (для Nginx), `3000`, `7860` (для контейнеров).

Информация о настройках

Система разворачивается с использованием Docker Compose и включает два основных сервиса:

- **Langflow** — среда оркестрации ИИ-агентов. Обеспечивает построение пайплайнов, управление агентами и их конфигурацию
- **OpenWebUI** — пользовательский чат-интерфейс для взаимодействия с агентами

Структура файлов

```
context-ai/
├── .env.example           # Шаблон файла переменных окружения
├── DEPLOY.md             # Настоящее руководство по развёртыванию
├── LANGFLOW_INTEGRATION.md # Руководство по подключению Langflow к
Open WebUI
├── docker-compose.yml    # Описание комплекса контейнеров
├── init-db.sh            # Сценарий инициализации PostgreSQL
├── nginx.conf.example    # Шаблон конфигурации Nginx
```

└─ docs/
 └─ images/
 └─ langflow/

Иллюстрации к LANGFLOW_INTEGRATION.md

Назначение файлов и каталогов:

Файл / Каталог	Назначение
.env.example	Шаблон файла переменных окружения. Подлежит копированию в файл <code>.env</code> с последующим заполнением значений
DEPLOY.md	Настоящее руководство по развёртыванию комплекса Open WebUI, Langflow и PostgreSQL
LANGFLOW_INTEGRATION.md	Руководство по подключению пайплайна Langflow к Open WebUI посредством механизма функций
docker-compose.yml	Описание комплекса контейнеров (<code>open-webui</code> , <code>db</code> , <code>langflow</code>), их сетевого взаимодействия и томов
init-db.sh	Сценарий, выполняемый при первичной инициализации PostgreSQL; создаёт пользователей и базы данных <code>openwebui</code> и <code>langflow</code>
nginx.conf.example	Шаблон конфигурации Nginx с двумя <code>server</code> -блоками (для Open WebUI и Langflow) и подстановочными значениями доменных имён
docs/images/langflow/	Каталог иллюстраций к руководству <code>LANGFLOW_INTEGRATION.md</code>

Каталоги, создаваемые в процессе развёртывания и не входящие в состав поставки:

Каталог	Назначение
langflow-data/	Bind-mount каталог Langflow (файловое хранилище, секретные ключи, конфигурация). Создаётся вручную на этапе запуска контейнеров

Контакты технического специалиста

По всем вопросам, связанным с установкой и настройкой программного обеспечения, необходимо обращаться к техническому специалисту ООО «Битек»:

Поддубный Иван

Телефон: +7 908 512 60 80

Электронная почта: ip@webpractik.ru

Способ установки

Развертывание docker контейнеров с помощью docker-compose и настройка внутренних конфигураций платформы

Порядок установки

1. Развертывание docker контейнеров с помощью docker-compose
2. Настройка функций для связи фреймворков
3. Добавление переменных в langflow

1. Развертывание docker контейнеров с помощью docker-compose

Компоненты

Компонент	Назначение	Внутренний порт
Open WebUI	Веб-интерфейс для работы с большими языковыми моделями	3000
Langflow	Визуальный конструктор пайплайнов на основе больших языковых моделей	7860
PostgreSQL	Система управления базами данных для сервисов Open WebUI и Langflow	5432
Nginx	Обратный прокси-сервер с терминацией TLS	80, 443

Шаг 1. Подготовка доменных имен

Для развертывания необходимо определить доменные имена, по которым будет осуществляться доступ к сервисам. В настоящем руководстве используются следующие обозначения-примеры:

Сервис	Обозначение-пример
Open WebUI	`example.com`
Langflow	`langflow.example.com`

В текстах команд и конфигурационных файлов указанные обозначения подлежат замене на фактические доменные имена, используемые в целевой среде.

Доменные имена должны быть настроены в системе DNS таким образом, чтобы корректно разрешаться в публичный IP-адрес целевого узла. Настройка DNS-записей выполняется администратором целевой среды.

Шаг 2. Подготовка SSL/TLS-сертификатов

До перехода к активации конфигурации Nginx (раздел 7) должны быть подготовлены действующие SSL/TLS-сертификаты для обоих доменных имён. Должны быть известны абсолютные пути к файлам сертификата (`fullchain.pem`) и закрытого ключа (`privkey.pem`).

Рекомендуется настроить автоматическое продление сертификатов штатными средствами выбранного удостоверяющего центра или клиента (например, systemd-таймером `certbot`).

Шаг 3. Настройка переменных окружения

Создать файл `.env` на основе шаблона:

```
```bash
cp .env.example .env
```
```

Заполнить значения переменных в файле `.env` в соответствии с таблицей:

| Переменная | Назначение |
|--|---|
| <code>POSTGRES_DB</code> | Имя служебной базы данных суперпользователя PostgreSQL (значение по умолчанию — <code>postgres</code>) |
| <code>POSTGRES_USER</code> | Имя суперпользователя PostgreSQL (значение по умолчанию — <code>postgres</code>) |
| <code>POSTGRES_PASSWORD</code> | Пароль суперпользователя PostgreSQL |
| <code>WEBUI_SECRET_KEY</code> | Секретный ключ Open WebUI, используемый для подписи пользовательских сессий |
| <code>WEBUI_URL</code> | Полный URL Open WebUI (например, <code>https://example.com</code>) |
| <code>OPENWEBUI_DB_PASSWORD</code> | Пароль пользователя <code>openwebui</code> базы данных |
| <code>LANGFLOW_SECRET_KEY</code> | Секретный ключ Langflow |
| <code>LANGFLOW_DB_PASSWORD</code> | Пароль пользователя <code>langflow</code> базы данных |
| <code>LANGFLOW_SUPERUSER</code> | Имя учётной записи суперпользователя Langflow |
| <code>LANGFLOW_SUPERUSER_PASSWORD</code> | Пароль учётной записи суперпользователя Langflow |

Для генерации значений секретных ключей (`WEBUI_SECRET_KEY`, `LANGFLOW_SECRET_KEY`) может применяться команда:

```
```bash
openssl rand -hex 32
```
```

Примечание. В значениях переменных `POSTGRES_PASSWORD`, `OPENWEBUI_DB_PASSWORD` и `LANGFLOW_DB_PASSWORD` не допускается использование специальных символов (`@`, `:`, `/`, `?`, `#`, `%`, `&`, `=`, `+`, `\$`, `;`, пробел и аналогичных). Указанные значения подставляются в строку подключения к PostgreSQL вида `postgresql://user:password@host:port/db`; присутствие специальных символов приведёт к некорректной интерпретации строки подключения и невозможности установить соединение с базой данных. Допускаются символы `A-Z`, `a-z`, `0-9`, а также `-` и `_`.

Шаг 3. Запуск контейнеров

Установить право на исполнение для сценария инициализации базы данных:

```
```bash
chmod +x init-db.sh
```
```

Сценарий `init-db.sh` выполняется при первом запуске контейнера PostgreSQL и создаёт пользователей и базы данных `openwebui` и `langflow`.

Подготовить каталог для хранения данных Langflow:

```
```bash
mkdir -p langflow-data
sudo chown -R 1000:1000 langflow-data
```
```

Каталог `./langflow-data` монтируется в контейнер Langflow по пути `/app/langflow`. Процесс внутри контейнера работает под UID `1000` и должен иметь право записи в этот каталог, в противном случае запуск Langflow завершится ошибкой `PermissionError`.

Запустить комплекс контейнеров:

```
```bash
docker compose up -d
```
```

Первичный запуск может занять продолжительное время ввиду необходимости загрузки образов и инициализации базы данных.

Проверить состояние контейнеров:

```
```bash
```

```
docker compose ps
```

```
```
```

Все контейнеры должны находиться в состоянии `Up`. Для контейнера `open-webui` дополнительно должна отображаться метка `healthy`.

Для просмотра журналов сервисов использовать следующие команды:

```
```bash
```

```
docker compose logs -f # журналы всех сервисов
```

```
docker compose logs -f open-webui # журнал отдельного сервиса
```

```
docker compose logs -f langflow
```

```
docker compose logs -f db
```

```
```
```

Шаг 4. Активация конфигурации Nginx

Скопировать шаблон конфигурации в каталог Nginx:

```
```bash
```

```
sudo cp nginx.conf.example /etc/nginx/sites-available/openwebui.conf
```

```
```
```

Открыть скопированный файл в текстовом редакторе и заменить подстановочные значения на фактические:

| Подстановочное значение | Значение для замены |
|---|--|
| `YOUR_DOMAIN_HERE` (в `server`-блоке Open WebUI) | Доменное имя Open WebUI (например, `example.com`) |
| `langflow.YOUR_DOMAIN_HERE` (в `server`-блоке Langflow) | Доменное имя Langflow (например, `langflow.example.com`) |

Замена должна быть произведена во всех вхождениях, включая директивы `server_name`, `ssl_certificate` и `ssl_certificate_key`.

Примечание. В шаблоне пути к файлам сертификатов указаны в формате клиента certbot / удостоверяющего центра Let's Encrypt (`/etc/letsencrypt/live/<домен>/fullchain.pem` , `privkey.pem``). При использовании иного способа выпуска сертификатов указанные пути следует заменить на фактические пути к файлам ``fullchain.pem`` и ``privkey.pem``.

Активировать конфигурацию:

```
``bash

sudo ln -s /etc/nginx/sites-available/openwebui.conf /etc/nginx/sites-enabled/

...

```

Проверить корректность синтаксиса:

```
``bash

sudo nginx -t

...

```

При отсутствии ошибок применить новую конфигурацию:

```
``bash

sudo systemctl reload nginx

...

```

Шаг 5 Проверка доступа

Работоспособность сервисов проверяется открытием в веб-браузере следующих адресов:

- Open WebUI — ``https://example.com``;
- Langflow — ``https://langflow.example.com``.

Первая зарегистрированная учётная запись в Open WebUI автоматически получает роль администратора.

Вход в Langflow осуществляется с использованием учётных данных, заданных в переменных ``LANGFLOW_SUPERUSER`` и ``LANGFLOW_SUPERUSER_PASSWORD`` файла ``.env``.

Шаг 6. Импорт пайплайнов

Прежде, чем переходить к последующим этапам установки необходимо импортировать приложенные пайплайны.

Для импортирования можно воспользоваться Drag-and-Drop решением, перетянув файл с пайплайном в формате .json в проект из интерфейса Langflow или использовать кнопку **Upload a flow** в верхней левой части экрана.

Шаг 7. Операции управления жизненным циклом

Остановка комплекса:

```
```bash
docker compose down
```
```

Перезапуск комплекса:

```
```bash
docker compose restart
```
```

Полная остановка с удалением данных:

```
```bash
docker compose down -v
```
```

Предупреждение. Выполнение команды `docker compose down -v` приводит к безвозвратному удалению Docker-тома `postgres_data`, а вместе с ним — всех данных баз `openwebui` и `langflow`, в том числе:

- в базе данных `openwebui`: учётных записей пользователей Open WebUI (включая идентификаторы, хэши паролей и назначенные роли), истории диалогов с языковыми моделями, параметров подключения моделей и сохранённых API-ключей внешних провайдеров, пользовательских промптов и пресетов, загруженных документов базы знаний;
- в базе данных `langflow`: созданных пайплайнов и их версий, учётных записей пользователей Langflow, сохранённых API-ключей и переменных окружения, журналов выполнения пайплайнов.

Каталог `./langflow-data` смонтирован как `bind-mount` в файловой системе хоста и командой `docker compose down -v` не удаляется. В нём хранятся файловое хранилище, секретные ключи и конфигурационные файлы Langflow; для его удаления требуется выполнить `rm -rf langflow-data` отдельно.

Восстановление указанных данных без предварительно созданной резервной копии невозможно. До выполнения команды `docker compose down -v` необходимо убедиться в наличии актуальной резервной копии (см. раздел 11).

Резервное копирование

Данные PostgreSQL хранятся в Docker-томе `postgres_data`, файлы Langflow — в каталоге `.langflow-data` в корне проекта (bind-mount на файловой системе хоста). На целевом узле должно быть настроено регулярное резервное копирование тома `postgres_data` и указанного каталога. Допустимые способы резервного копирования включают, помимо прочего, выгрузку базы данных средствами `pg_dumpall` по расписанию, создание снимков томов или применение средств резервного копирования, принятых в эксплуатирующей организации. Периодичность и метод резервного копирования определяются администратором целевой среды.

2. Подключение Langflow к OpenWebUI

Настоящее руководство описывает порядок подключения пайплайна Langflow к Open WebUI в качестве модели посредством механизма функций (Functions) Open WebUI. Предполагается, что оба сервиса развёрнуты в соответствии с документом `DEPLOY.md` и находятся в одной Docker-сети `openwebui-net`.

Общие сведения

Интеграция реализуется с помощью функции типа **Pipe** в Open WebUI. Функция выступает посредником: принимает сообщение пользователя из интерфейса чата Open WebUI, передаёт его в пайплайн Langflow через HTTP API и возвращает ответ пайплайна обратно в чат. После активации функция отображается в Open WebUI как отдельная модель и выбирается в выпадающем списке моделей на странице чата.

Для работы интеграции необходимо получить три значения:

| Значение | Источник |
|-----------------------------------|---|
| Базовый URL Langflow | Определяется способом сетевого взаимодействия |
| Идентификатор пайплайна (Flow ID) | Интерфейс Langflow |
| API-ключ Langflow | Интерфейс Langflow |

Подготовка пайплайна в Langflow

1. Войти в Langflow по адресу, указанному при развёртывании (например, `https://langflow.example.com`), под учётной записью суперпользователя.
2. Открыть существующий пайплайн либо создать новый. Пайплайн должен содержать компоненты **Chat Input** и **Chat Output** — в противном случае вызов через API невозможен.
3. Убедиться в работоспособности пайплайна, выполнив его запуск из встроенной панели **Playground**.

4. Скопировать идентификатор пайплайна (Flow ID). Идентификатор отображается в адресной строке браузера при открытом пайплайне:
``https://langflow.example.com/flow/<FLOW_ID>``.
Значение ``<FLOW_ID>`` представляет собой строку в формате UUID.

Создание API-ключа Langflow

1. В интерфейсе Langflow открыть меню учётной записи (пиктограмма в правом верхнем углу) и перейти в раздел **Settings** → **Langflow API Keys**.
2. Нажать **Add New** (или аналогичную кнопку создания ключа), указать произвольное описание (например, ``openwebui``) и подтвердить создание.
3. Скопировать сгенерированное значение ключа и сохранить его в защищённом хранилище. Повторный просмотр значения после закрытия диалогового окна невозможен.

Определение базового URL Langflow

Базовый URL, указываемый в параметрах функции Open WebUI, зависит от способа сетевого взаимодействия между контейнерами:

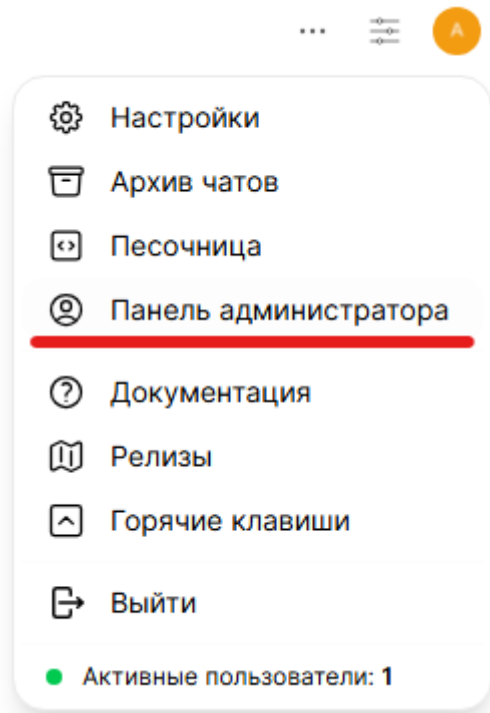
| Способ | Базовый URL | Применение |
|--|---|--|
| Взаимодействие внутри Docker-сети <code>`openwebui-net`</code> | <code>`http://langflow:7860`</code> | Рекомендуемый вариант. Трафик не выходит за пределы Docker-хоста, TLS не требуется |
| Взаимодействие через внешнее доменное имя Langflow | <code>`https://langflow.example.com`</code> | Применяется при размещении Open WebUI и Langflow на разных узлах |

В настоящем руководстве в качестве базового используется значение ``http://langflow:7860``, соответствующее имени сервиса в файле ``docker-compose.yml`` и внутреннему порту Langflow.

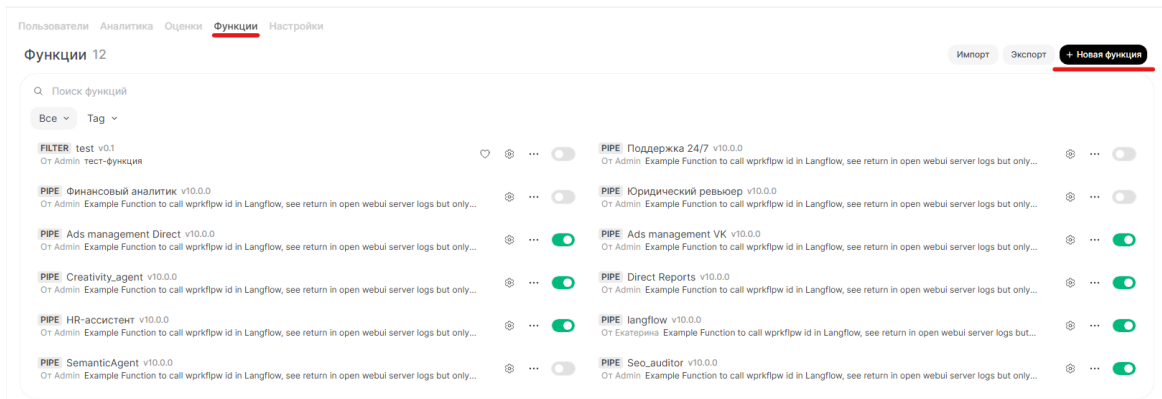
Создание функции в Open WebUI

1. Войти в Open WebUI под учётной записью с ролью администратора.

2. Открыть **Admin Panel** (раздел панели администратора) → вкладка **Functions**.



3. Нажать пиктограмму + (Create New Function) в правой верхней части страницы.



4. В открывшейся форме заполнить поля:

| Поле | Значение |
|----------------------|--|
| Function Name | Произвольное наименование (например, `Langflow`) |
| Function Description | Произвольное описание назначения функции |
| Тело функции | Исходный код Pipe-функции для Langflow |

5. В качестве тела функции использовать готовую Pipe-функцию Langflow, опубликованную в каталоге функций Open WebUI

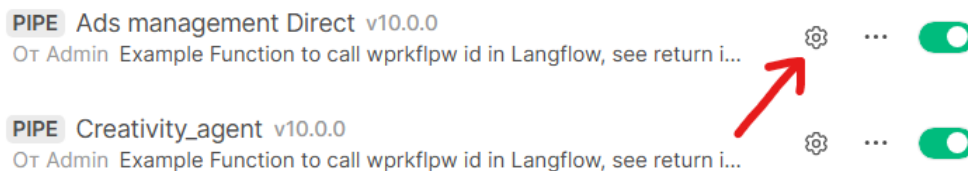
(`https://openwebui.com/functions`, поиск по запросу `langflow`). Код функции полностью копируется в поле редактора.

Примечание. При выборе функции из каталога следует отдавать предпочтение вариантам с максимальным числом установок и актуальной датой обновления. Конкретный перечень параметров (Valves) может незначительно отличаться между реализациями; приведенная далее таблица отражает типовой набор.

6. Нажать **Save**. После сохранения функция отобразится в списке функций вкладки **Functions**.

Настройка параметров функции (Valves)

1. В списке функций найти созданную функцию и нажать пиктограмму шестерни (**Valves / Settings**) рядом с её именем.



2. Заполнить параметры:

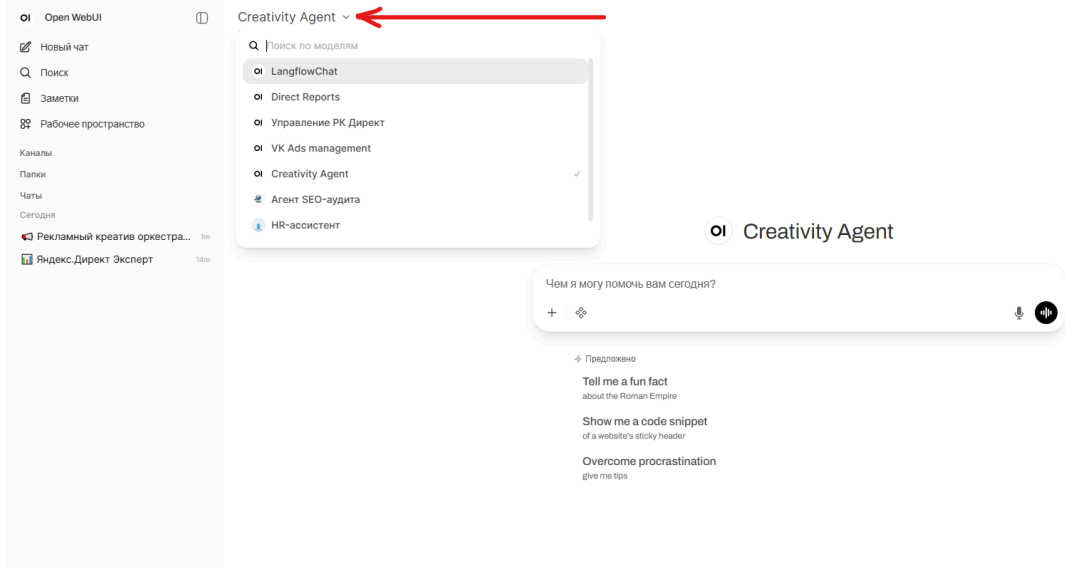
| Параметр | Значение | Комментарий |
|--|--|---|
| `LANGFLOW_BASE_URL`
(или `Langflow URL`,
`Base URL`) | `http://langflow:7860` | |
| `LANGFLOW_API_KEY`
(или `API Key`) | Значение, полученное
в разделе “Создание
API-ключа Langflow” | Передаётся в
HTTP-заголовке
`x-api-key` |
| `FLOW_ID` (или `Flow ID`,
`Default Flow ID`) | Значение, полученное
в разделе “Подготовка
пайплайна в Langflow” | Идентификатор
пайплайна в формате
UUID |
| `TWEAKS` (при наличии) | Пустое значение или
JSON-объект | Применяется для
переопределения
параметров
компонентов пайплайна |

Примечание. Названия параметров зависят от выбранной реализации Pipe-функции. В случае расхождения наименований следует руководствоваться назначением параметра, а не его точным именем.

3. Сохранить изменения (**Save**).
4. В списке функций перевести переключатель состояния функции в положение **Enabled (активна)**.

Проверка подключения

1. Открыть главную страницу Open WebUI (**New Chat**).
2. В выпадающем списке моделей (в верхней части окна чата) выбрать модель с наименованием, соответствующим созданной функции. При использовании Pipe-функции, реализующей выбор пайплайна по идентификатору, модель отображается под именем функции (например, `Langflow`).

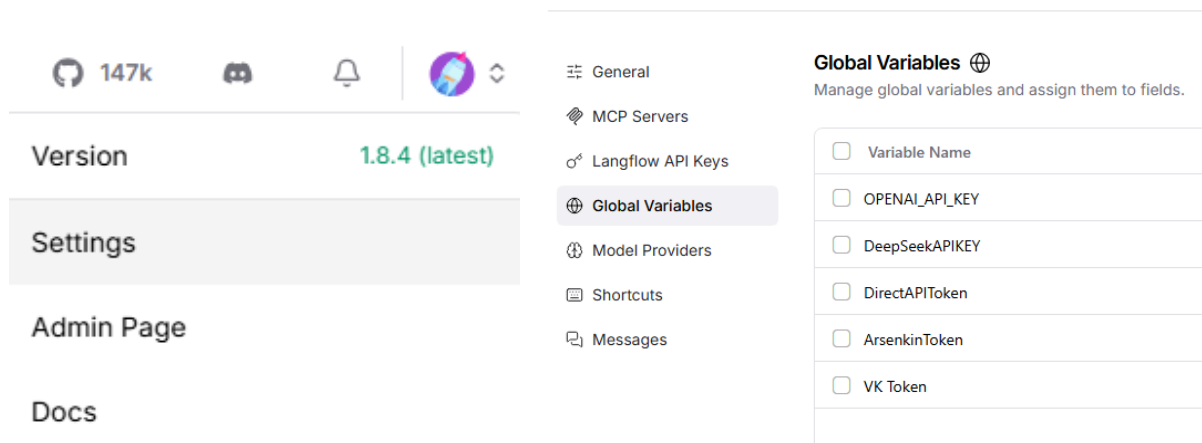


3. Отправить тестовое сообщение и убедиться в получении ответа от пайплайна Langflow.

3. Добавление переменных в Langflow

Шаг 1. Добавить переменные в разделе settings

После открытия langflow перейти в раздел setting, нажав на аватарку профиля сверху справа, после чего перейти в подраздел Global Variables:



В данном подразделе создать следующие переменные (названия могут быть другие):

| Название переменной | Содержание |
|---------------------|---|
| DeepSeek APIKEY | API-ключ для доступа к моделям Deepseek |

| | |
|-------------------|---|
| Arsenkin APIToken | API-токен от сервиса Arsenkin.ru |
| Direct APIToken | API-токен приложения с доступом к рекламному кабинету Яндекс.Директ |
| VK APIToken | API-токен рекламного кабинета ВКонтакте |

Шаг 2. Установить переменные в каждый пайплайн langflow:

| Пайплайн | Используемые переменные |
|----------------------------------|------------------------------------|
| SEO auditor | DeepSeek APIKEY, Arsenkin APIToken |
| Creativity agent | DeepSeek APIKEY |
| Reports YDirect Agent | DeepSeek APIKEY, Direct APIToken |
| AC management YDirec | DeepSeek APIKEY, Direct APIToken |
| AC management YDirect VK | DeepSeek APIKEY, VK APIToken |
| Seo Agent (связь с GetSemantics) | DeepSeek APIKEY, Arsenkin APIToken |
| GetSemantics | DeepSeek APIKEY, Arsenkin APIToken |

Установка переменных осуществляется путем нажатия на пиктограмму в поле ввода ключа и выбора из выпадающего меню соответствующей переменной.

