

Инструкция по установке ПО
«Умка ИИ»

1. Введение

1.1 Описание поставляемого Экземпляра ПО

ПО «Умка ИИ» поставляется пользователю (Заказчику, покупателю) в виде файла виртуальной машины с предустановленными сервисами, базой данных и наборов тестовых случаев для демонстрации работы ПО «Умка ИИ».

Экземпляр ПО «Умка ИИ» поставляется пользователю после приобретения ПО «Умка ИИ».

1.2 Системные требования

Системные требования для ПК для установки ПО Умка ИИ:

- Операционная система Ubuntu (рекомендуемая версия системы: Ubuntu 22.04).
- 512 Гб памяти на жёстком диске (желательно использование SSD).
- 32 Гб ОЗУ.
- NVIDIA A100 80GB или аналогичная от другого производителя

2. Запуск ПО «Умка ИИ»

Предварительные требования

1. **Сервер с Linux (Ubuntu или любой другой дистрибутив):**
 - Имеется SSH-доступ.
 - На сервере установлен Docker и Docker Compose.
 - i. **Docker** должен быть установлен как сервис **systemd** (через **apt**), а не как **snap-пакет**. Это гарантирует корректную интеграцию с системными службами.
 - Необходим доступ в интернет для скачивания экземпляра ПО по предоставляемым правообладателем ПО «Умка ИИ» ссылкам после приобретения ПО «Умка ИИ».
2. **Файлы проекта:**
 - **docker-compose.yml** — основной файл, в котором описаны сервисы стека.
 - **Caddyfile** — конфигурационный файл для reverse-прокси (Caddy).
 - **.env** — файл с переменными окружения, необходимыми для работы сервисов (БД, Redis, S3, JWT-секреты и др.).
3. **Переменные окружения:**
 - Все необходимые данные (пароли, ключи, хосты) должны быть указаны в **.env** файле.
4. **Привязка домена:**

- Домен должен быть привязан к IP-адресу основного сервера.
- Убедитесь, что DNS-записи (A-запись или CNAME) настроены корректно.

5. Открытые порты:

- Предварительно откройте следующие порты на сервере:
 - 80 и 443 — для Caddy (HTTP и HTTPS).
 - 9000 и 9001 — для MinIO (админ-интерфейс и API).
- После завершения настройки MinIO рекомендуется закрыть порты 9000 и 9001 для обеспечения безопасности.

Шаги по деплою

Подготовка файлов проекта на сервере

Создайте целевую директорию для проекта, например:

```
mkdir -p /root/umka_core
cd /root/umka_core
```

2. Скопируйте необходимые файлы на сервер.

Вот необходима иерархия файлов:

```
umka_core/
├── .env
├── docker-compose.yml
├── caddy_server/
│   └── Caddyfile
```

Настройка переменных окружения (.env)

Откройте `.env` для редактирования:

```
nano /root/umka_core/.env
```

- Внесите все нужные значения:
 - `SECRET_KEY_JWT` - Секретный ключ для подписания JWT (JSON Web Token).
 - `ALGORITHM_JWT` - Алгоритм, используемый для шифрования JWT.
 - `ACCESS_TOKEN_EXPIRE_MINUTES` - Время жизни токена доступа в минутах.
 - `HTTPS_IS` - Флаг, указывающий, используется ли HTTPS.
 - `SAME_SITE` - Политика Cookie. Значение `strict` указывает, что Cookie отправляется только с запросами с того же сайта.
 - `ALLows_ORIGIN` - Указывает разрешённый источник для CORS (Cross-Origin Resource Sharing).
 - `NEXT_PUBLIC_API_URL` - URL-адрес Backend API, используемого клиентской частью приложения.

- `DB_HOST` - Хост для подключения к базе данных.
- `DB_NAME` - Имя базы данных.
- `DB_USER` - Имя пользователя базы данных.
- `DB_PASSWORD` - Пароль для подключения к базе данных.
- `REDIS_PASSWORD` - Пароль для подключения к серверу Redis.
- `REDIS_DB` - Номер базы данных в Redis (по умолчанию 0).
- `REDIS_HOST` - Хост Redis-сервера.
- `REDIS_PORT` - Порт для подключения к Redis. По умолчанию используется 6379.
- `ACCESS_KEY_S3` - Ключ доступа к MinIO. Пока не заполнен. На данном этапе должен быть пустым.
- `SECRET_KEY_S3` - Секретный ключ для доступа к MinIO. На данном этапе должен быть пустым.
- `BUCKET_NAME_S3_UPLOAD` - Имя bucket для загрузки файлов. На данном этапе должен быть пустым.
- `S3_HOST` - Хост для доступа к MinIO.
- `S3_HOST_EXTERNAL` - Внешний URL-адрес для доступа к MinIO.
- `S3_PATH_PREFIX` - Префикс пути для файлов в MinIO. Нужен для внешнего доступа к s3 через caddy.
- `MINIO_ROOT_USER` - Имя администратора MinIO.
- `MINIO_ROOT_PASSWORD` - Пароль администратора MinIO.
- `VLLM_SERVER_URL` - URL-адрес сервера для работы с LLM (Large Language Model).
- `WHISPER_SERVER_URL` - URL-адрес сервера для работы с Whisper (модель для транскрибации).
- Сохраните файл после редактирования.

Инициализация Docker Swarm (если не сделано ранее)

Инициализируйте Swarm (если у вас одиночный сервер):

```
docker swarm init
```

- Если Swarm уже инициализирован (например, в предыдущих деплоях), можно пропустить этот шаг или игнорировать ошибку о том, что Swarm уже существует.

Добавление Caddyfile в Docker Config (для Docker Swarm)

В `docker-compose.yml` предполагается использование `docker config`.

Создайте конфиг на основе `Caddyfile_sample`:

```
docker config rm caddyfile_config || true
```

```
docker config create caddyfile_config Caddyfile
```

- Здесь мы сначала пытаемся удалить предыдущий конфиг (если был), а затем создаём новый.

Настройка Caddyfile:

- **your_domain.ru**: Укажите домен, привязанный к вашему серверу. Это будет основная точка входа для всех запросов.
- **handle_path /web-api/***: Запросы, начинающиеся с **/web-api/**, перенаправляются на **backend_api:8000**. Здесь **backend_api** — имя контейнера, указанное в **docker-compose.yml**.
- **handle /s3/***: Запросы для S3 обрабатываются MinIO на порту **9000**, при этом путь **/s3** удаляется из запроса (с помощью **uri strip_prefix /s3**).
- **handle /s3-admin/***: Административные запросы к MinIO перенаправляются на порт **9001**.
- **handle**: Все остальные запросы перенаправляются на фронтенд (порт **3000**, контейнер **frontend**).

Настройка LLM модели

В рамках **docker-compose.yml** найдите сервис **llm**

В нем в пункте **volumes** укажите путь к скачанной модели

Настройка Whisper сервиса

В рамках **docker-compose.yml** найдите сервис **whisper**

В нем в пункте **volumes** укажите путь к скачанным **ConfigFile**, **Speaker_Model_Path**, **Diarizer_Model_Path**, **Whisper_Model_Path**

Откройте или создайте файл `/etc/docker/daemon.json`.

Добавьте следующую конфигурацию в файл **/etc/docker/daemon.json**.

```
{
  "insecure-registries" : ["95.163.251.218:5000"]
}
```

Перезапустите Docker:

```
sudo systemctl restart docker
```

Выгрузка конфигурации сервисов из Docker-registry

```
docker pull 95.163.251.218:5000/umkaai/umkacore:backend_api
```

```
docker pull 95.163.251.218:5000/umkaai/umkacore:worker
```

```
docker pull 95.163.251.218:5000/umkaai/umkacore:frontend
```

```
docker pull 95.163.251.218:5000/umkaai/umkacore:llm
```

```
docker pull 95.163.251.218:5000/umkaai/umkacore:whisper
```

Деплой проекта через Docker Stack

Выполните команду деплоя

```
export $(cat .env | xargs)
```

```
docker stack deploy -c docker-compose.yml my_stack
```

Здесь `my_stack` — имя стэка, под которым будут запускаться все сервисы. После этого Docker будет скачивать необходимые образы (если их ещё нет) и запускать контейнеры.

Настройка MinIO

Откройте браузер и перейдите по адресу:

arduino

Копировать код

```
http://<ваш_ip>:9001
```

- Используйте `MINIO_ROOT_USER` и `MINIO_ROOT_PASSWORD` для входа.

Создайте бакет (bucket) для хранения данных.

Сгенерируйте **access token** для приложения:

- Перейдите в раздел настройки MinIO.
- Создайте токены доступа (Access Key и Secret Key).

Запишите названия бакета и токены в `.env`:

```
S3_BUCKET_NAME=<имя_бакета>
```

```
S3_ACCESS_KEY=<access_key>
```

```
S3_SECRET_KEY=<secret_key>
```

Заккрытие портов 9000 и 9001

После настройки MinIO, закройте порты для безопасности:

Используйте `ufw` (если установлен):

```
sudo ufw deny 9000
```

```
sudo ufw deny 9001
```

Либо настройте правила через `iptables` или любой другой файрвол.

Удалите текущий стек

Прежде чем перезапускать проект, остановите текущие сервисы:

```
docker stack rm my_stack
```

Убедитесь, что все контейнеры и сервисы стека остановлены:

```
docker stack ps my_stack
```

- Если вывод пустой — стек удалён.

Выполните повторный деплой

Запустите стек с учетом обновленных файлов:

```
export $(cat .env | xargs)
```

```
docker stack deploy -c docker-compose.yml my_stack
```

3. Запуск ПО

Для запуска ПО необходимо ввести в адресную строку браузера доменный адрес и нажать клавишу «Enter» на клавиатуре. После этого откроется окно авторизации оператора.

Далее необходимо действовать в соответствии с документом «Общее руководство», раздел «Начало работы».