Unica Lite Platform

Руководство по установке и эксплуатации

Docker Compose — дистрибутив для CPU и GPU серверов

Версия	Дистрибутив
0.1	Docker Compose · CPU / GPU

1. Обзор платформы

Unica Lite Platform — дистрибутив на базе Docker Compose для развёртывания на CPU- и GPU-серверах. Включает инфраструктурные компоненты (PostgreSQL, MinIO, Qdrant, Redis, Nginx) и прикладные API для обработки документов и распознавания речи.

Платформа поддерживает два режима работы: CPU (без GPU) и GPU (NVIDIA CUDA). Установщик install.sh настраивает всё окружение в интерактивном режиме и генерирует все необходимые файлы конфигурации.

2. Системные требования

Компонент	Минимум	Примечание
Docker Engine	28+	Проверяется автоматически при запуске install.sh
Docker Compose	v2	Встроенный плагин: docker compose
CPU	8 ядер	При меньшем значении — предупреждение установщика
RAM	8 ГБ	При меньшем значении — предупреждение установщика
Дисковое пространство	100 ГБ+	Зависит от объёма моделей и данных
GPU (опционально)	NVIDIA CUDA	Требуется nvidia-container-toolkit

⚠ Если ресурсы ниже рекомендованных, установщик выведет предупреждение и предложит продолжить или отменить установку. Платформа может работать нестабильно.

3. Установка

Установщик install.sh выполняет всю настройку в интерактивном режиме:

```
bash install.sh
```

3.1. Шаги установщика

1 — Проверка окружения

Проверяет наличие Docker и Docker Compose, CPU, RAM и свободное место на диске. При нехватке ресурсов выводит предупреждение.

2 — Выбор режима развёртывания

[1] CPU — без GPU [2] GPU — NVIDIA CUDA (требуется nvidia-container-toolkit). При выборе GPU без установленного nvidia-container-toolkit установщик вернётся к выбору.

3 — Ввод параметров

Домен [aidev.local] · Реестр образов · Путь к данным [./data] · Путь к логам [./logs]

4 — Генерация файлов конфигурации

.env · init-db.sql · nginx.conf · docker-compose.yml

5 — Инструкции по загрузке моделей

После генерации файлов выводит команды для загрузки моделей ASR (см. раздел 4).

3.2. Сгенерированные файлы

Файл	Описание
`.env`	Все переменные окружения. Пароли генерируются случайно (20 символов, A–Z, a–z, 0–9). Session secret — 64 hex-символа.
`init-db.sql`	SQL-скрипт создания ролей и баз данных: asr_db, datamanagement_db, unica_db. Выполняется при первом старте PostgreSQL.
`nginx.conf`	Маршрутизация по доменам, проксирование к сервисам (Unica, MinIO, API).
`docker-compose.yml`	Полное описание сервисов, сетей и томов. В GPU-режиме добавляются блоки deploy.resources.

⚠ Пароли генерируются однократно при установке и сохраняются только в файл .env. Сохраните этот файл в надёжном месте.

3.3. Директории данных

Установщик автоматически создаёт директории и выставляет права владельца:

```
${DATA_PATH}/postgresql     # uid 999 (postgres)
${DATA_PATH}/minio          # uid 1000
${DATA_PATH}/qdrant         # uid 1000
${DATA_PATH}/redis
${DATA_PATH}/diarization
${DATA_PATH}/recognitionengine
${DATA_PATH}/speechalignment
${DATA_PATH}/speechrecognition
${LOG_PATH}/unica
```

⚠ Для корректной установки прав владельца (chown) запустите install.sh от имени root.

4. Загрузка моделей ASR

После завершения установки необходимо загрузить и распаковать модели распознавания речи. Архив содержит модели для всех четырёх ASR-сервисов.

```
wget -O /tmp/lite-models.tar.gz https://downloads.hopper-it.ru/lite-models.tar.gz
tar -xzf /tmp/lite-models.tar.gz -C ./data
rm /tmp/lite-models.tar.gz
```

Структура после распаковки:

```
data/
├── diarization/          # Модели pyannote (определение говорящих)
├── recognitionengine/    # Модели GigaAM (транскрибация)
├── speechalignment/      # wav2vec2 XLSR-53 Russian (выравнивание)
└── speechrecognition/    # Whisper-совместимые модели
```

5. Запуск платформы

```
docker compose up -d
```

После запуска платформа доступна по следующим адресам:

Адрес	Сервис	Описание
`http://<DOMAIN>`	Unica	Основной веб-интерфейс платформы
`http://console.<DOMAIN>`	MinIO Console	Управление объектным хранилищем
`http://s3.<DOMAIN>`	MinIO S3	S3 API для программного доступа

i Перед запуском убедитесь, что домен (и его поддомены console.* и s3.*) разрешается в IP-адрес сервера. При локальной установке добавьте записи в /etc/hosts.

6. Состав сервисов

Инфраструктура

Сервис	Образ	Описание
postgresql	postgres:16	Реляционная СУБД. Хранит данные для ASR, DataManagement и Unica. Инициализируется init-db.sql.
minio	minio/minio:...	S3-совместимое объектное хранилище. Файлы: порт 9000, консоль: порт 9001.
qdrant	qdrant/qdrant:v1.16.2	Векторная БД для семантического поиска. REST: 6333, gRPC: 6334.
redis	redis:8.6.1-alpine	In-memory кеш и очереди задач для Unica. Данные персистентны (AOF).
nginx	nginx:1.28-alpine	Reverse proxy. Единая точка входа на порту 80.
docling	docling-serve:v1.13.1	OCR и парсинг документов (PDF, DOCX и др.). Порт 5001. Concurrency = 2.

Приложения

Сервис	Описание
unica	Основное веб-приложение. Порт 5000. Зависит от PostgreSQL, Redis, MinIO и Qdrant. Поддерживает WebSocket.
datamanagement-api	REST API управления наборами данных. Порт 8080. БД datamanagement_db.
files-api	Файловый сервис: загрузка и скачивание через MinIO. Порт 8080.

ASR — Распознавание речи

Сервис	Модель	Описание
speechrecognition-api	—	Оркестратор ASR-пайплайна. Принимает задачи, координирует остальные ASR-сервисы. БД asr_db.
diarization-api	pyannote	Диаризация: определяет временны́е сегменты и принадлежность говорящему.
recognitionengine-api	GigaAM	Движок транскрибации: преобразует аудио в текст.
speechalignment-api	wav2vec2 XLSR-53 RU	Выравнивание: синхронизирует слова транскрипта с таймкодами. HF_HUB_OFFLINE=true.

i В GPU-режиме к сервисам diarization-api, recognitionengine-api и speechalignment-api автоматически добавляется секция deploy.resources с резервированием всех GPU (count: all, driver: nvidia).

7. Маршрутизация Nginx

Nginx принимает весь внешний трафик на порту 80 и проксирует запросы к внутренним сервисам по доменному имени и пути URL.

Домен / путь	Upstream	Порт
<DOMAIN> /	unica	5000
<DOMAIN> /api/datamanagement	datamanagement-api	8080
<DOMAIN> /api/files	files-api	8080
<DOMAIN> /api/asr	speechrecognition-api	8080
<DOMAIN> /api/diarization	diarization-api	8080
<DOMAIN> /api/recognition	recognitionengine-api	8080
<DOMAIN> /api/speechalignment	speechalignment-api	8080
console.<DOMAIN> /	minio (console)	9001
s3.<DOMAIN> /	minio (S3)	9000

WebSocket включён для Unica и MinIO Console. Максимальный размер тела запроса — 512 МБ. Таймаут чтения/записи — 600 секунд.

8. Ключевые переменные окружения (.env)

Переменная	Значение / генерируется	Описание
DOMAIN	aidev.local	Основной домен платформы
DEPLOY_MODE	cpu / gpu	Режим развёртывания
DATA_PATH	./data	Базовый путь для томов данных
LOG_PATH	./logs	Путь для логов сервисов
POSTGRES_PASSWORD	генерируется	Пароль суперпользователя PostgreSQL
MINIO_ROOT_USER	minioadmin	Логин администратора MinIO
MINIO_ROOT_PASSWORD	генерируется	Пароль администратора MinIO
UNICA_SESSION_SECRET	генерируется (64 hex)	Секрет сессии Unica
UNICA_DOCLING_ENABLED	true	Включить интеграцию с Docling
UNICA_MINIO_REGION	RU-MOW	Регион MinIO для Unica

9. Обновление версий образов

Обновление платформы сводится к изменению тегов образов в файле .env и перезапуску затронутых контейнеров. Файл docker-compose.yml при этом не изменяется.

9.1. Структура тегов в .env

Все образы задаются в двух блоках .env.

Инфраструктурные образы — полная строка image:tag:

```
POSTGRES_IMAGE=postgres:16
MINIO_IMAGE=minio/minio:RELEASE.2025-09-07T16-13-09Z
QDRANT_IMAGE=qdrant/qdrant:v1.16.2
REDIS_IMAGE=redis:8.6.1-alpine
VLLM_IMAGE=vllm/vllm-openai-cpu:v0.17.1
DOCLING_IMAGE=ghcr.io/docling-project/docling-serve:v1.13.1
UNICA_IMAGE=zot.in.s101.hopper-it.ru/aidev/unica/unica:v0.0.1
NGINX_IMAGE=nginx:1.28-alpine
```

Теги прикладных API — отдельные переменные *_TAG:

```
DATAMANAGEMENT_API_TAG=v0.0.1
DIARIZATION_API_TAG=v0.0.1
EMBEDDING_API_TAG=v0.0.1
FILES_API_TAG=v0.0.1
LLM_API_TAG=v0.0.1
RECOGNITIONENGINE_API_TAG=v0.0.1
SPEECHALIGNMENT_API_TAG=v0.0.1
SPEECHRECOGNITION_API_TAG=v0.0.1
VECTORS_API_TAG=v0.0.1
```

i Инфраструктурные образы (PostgreSQL, Redis, Qdrant и др.) обычно обновляются отдельно по необходимости, а не при каждом релизе платформы.

9.2. Обновление всех прикладных сервисов (новый релиз)

При выходе нового релиза все теги API-сервисов меняются одновременно. Например, обновление с v0.0.1 до v0.0.2:

Шаг 1 — массовая замена тегов через sed:

```
sed -i 's/_TAG=v0.0.1/_TAG=v0.0.2/g' .env
```

Или вручную отредактировать .env, заменив все строки вида `*_TAG=v0.0.1`.

Шаг 2 — загрузить новые образы и перезапустить:

```
docker compose pull
docker compose up -d
```

i *Docker Compose пересоздаст только те контейнеры, образ которых изменился. Остальные сервисы продолжат работу без перезапуска.*

9.3. Обновление отдельного сервиса

Шаг 1 — изменить нужную переменную в .env:

```
# Обновление Unica
UNICA_IMAGE=zot.in.s101.hopper-it.ru/aidev/unica/unica:v0.0.2
```

```
# Обновление одного API
SPEECHRECOGNITION_API_TAG=v0.0.2
```

Шаг 2 — загрузить и перезапустить только этот сервис:

```
docker compose pull unica
docker compose up -d unica
```

9.4. Проверка текущих версий

Образы всех запущенных контейнеров:

```
docker compose ps --format 'table {{.Name}}\t{{.Image}}'
```

Образ конкретного контейнера:

```
docker inspect unica --format '{{.Config.Image}}'
```

9.5. Откат к предыдущей версии

Вернуть предыдущий тег в .env и перезапустить сервис:

```
UNICA_IMAGE=zot.in.s101.hopper-it.ru/aidev/unica/unica:v0.0.1

docker compose up -d unica
```

⚠ Перед обновлением рекомендуется создать резервную копию базы данных. Некоторые обновления могут включать миграции схемы БД, которые необратимы.

9.6. Сводная таблица образов и переменных

Сервис	Переменная в .env	Тип
postgresql	POSTGRES_IMAGE	полный образ
minio	MINIO_IMAGE	полный образ
qdrant	QDRANT_IMAGE	полный образ
redis	REDIS_IMAGE	полный образ
vllm-llm / embedding	VLLM_IMAGE	полный образ
docling	DOCLING_IMAGE	полный образ
unica	UNICA_IMAGE	полный образ
nginx	NGINX_IMAGE	полный образ
datamanagement-api	DATAMANAGEMENT_API_TAG	тег
files-api	FILES_API_TAG	тег
llm-api	LLM_API_TAG	тег
embedding-api	EMBEDDING_API_TAG	тег
vectors-api	VECTORS_API_TAG	тег
speechrecognition-api	SPEECHRECOGNITION_API_TAG	тег
diarization-api	DIARIZATION_API_TAG	тег

recognitionengine-api	RECOGNITIONENGINE_API_TAG	тег
speechalignment-api	SPEECHALIGNMENT_API_TAG	тег

10. Управление сервисами

Команда	Описание
`docker compose up -d`	Запустить все сервисы в фоновом режиме
`docker compose down`	Остановить и удалить контейнеры
`docker compose ps`	Показать статус всех сервисов
`docker compose logs -f`	Следить за логами всех сервисов (Ctrl+C для выхода)
`docker compose logs -f <service>`	Логи конкретного сервиса
`docker compose restart <service>`	Перезапустить конкретный сервис