

Инструкция по установке Arenadata Enterprise SQL Umformer (ADESU)

Москва
2025

Содержание:

1	Термины и определения	3
2	Сокращения и обозначения	4
3	Введение	5
4	Требования к узлу развертывания	7
5	Требования к системам-источникам и системам-приемникам.....	8
6	Порядок развертывания.....	9
7	Запуск и настройка	11
8	Запуск UI	12
9	Настройка config.yaml.....	13
9.1	Обязательные параметры	13
9.2	Необязательные параметры	14
9.3	Параметры data.yaml	14

1 Термины и определения

Термин	Значение
ANSI SQL	Стандартный язык запросов к базам данных, разработанный Американским национальным стандартом (ANSI)
LLM-модели	Языковые модели, обученные на больших объемах текстовых данных

2 Сокращения и обозначения

Сокращение	Наименование
ADB	Arenadata DB
ADESU, ПО Umformer	Arenadata Enterprise SQL Umformer
API	(Application Programming Interface) – набор классов, процедур, функций, структур или констант, которыми одна компьютерная программа может взаимодействовать с другой программой
CPU	(англ. – Central Processing Unit) – центральный процессор
DDL	Data Definition Language
DML	Data Manipulation Language
IP	(Internet Protocol) – маршрутизируемый протокол сетевого уровня стека TCP/IP
MSSQL	Microsoft SQL Server
LDAP	(англ. – Lightweight Directory Access Protocol) – протокол прикладного уровня для доступа к службе каталогов
LLMs	Large language models
SSL	Secure Sockets Layer
SQL	(Structured Query Language) – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционных базах данных
UI	User interface
БД	База данных
ГБ	Гигабайт
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
ТБ	Терабайт

3 Введение

Arenadata Enterprise SQL Umformer (далее ПО Umformer, ADESU) — это программный продукт, предназначенный для автоматического преобразования кода из одной системы в другую. Продукт будет использовать LLM-модели для трансформации запросов и обеспечивать проверку синтаксиса полученных запросов. Основные функции продукта:

- Трансформация sql-запросов с помощью LLM (источником и приемником могут быть следующие СУБД: Oracle, ADB, MSSQL, Impala, Hive);
- Трансформация функций и процедур Oracle в функции ADB с помощью LLM.

ПО Umformer является инструментом поддержки миграции с распространенных СУБД зарубежного производства на СУБД, использующие диалект PostgreSQL, например, ADB и ADPG.

Функционально Umformer состоит из двух блоков:

- Блок пакетной трансформации SQL-кода;
- Блок интерактивной доработки SQL-кода с поддержкой ML, реализованный в форме WEB UI.

ПО Umformer предназначено для работы совместно с внешней LLM, соответствующей открытой спецификации OpenAI. Элементы LLM в состав дистрибутива ПО Umformer не включаются и должны быть развернуты отдельно.

ПО Umformer поставляется в виде архива, содержащего исполняемый код приложения, файлы конфигурации и docker image, обеспечивающий простую переносимость приложения.

ПО Umformer поддерживает два режима работы: конвертация пакетов Oracle и конвертация SQL-скриптов из текстового файла. Текущий режим определяется значением параметра конфигурации *transformer_type: package* и *sql* соответственно.

4 Требования к узлу развертывания

Установка ПО Umformer выполняется на хост под управлением Linux¹. Для работы ПО Umformer требуется наличие на хосте установленной и настроенной платформы запуска контейнерных приложений Docker или аналогичной, например, Podman. Текущему пользователю должны быть предоставлены привилегии запуска docker контейнеров.

Следующая команда добавляет пользователя в группу докер:

```
sudo usermod -aG docker $USER
```

Группа должна быть создана заранее.

Настройки хоста и платформы запуска контейнерных приложений должны обеспечивать запуск docker image, поставляемого в составе ПО Umformer, от имени непривилегированного пользователя.

¹ Работа ПО на Windows возможна, но не описывается в данной инструкции.

5 Требования к системам-источникам и системам-приемникам

ПО Umformer поддерживает следующие версии СУБД-источников:

- Oracle Database:
 - Версия ПО: 11g – 21c;
 - Аутентификация: внутренняя ролевая модель БД, без SSL / LDAP / Kerberos;
 - Необходимые ресурсы: CPU vCores: 4+, RAM: 8+ GB, Storage: 50+ GB;
- Microsoft SQL Server:
 - Версия ПО: 2005 – 2022;
 - Аутентификация: внутренняя ролевая модель БД, без SSL / LDAP / Kerberos;
 - Ресурсы: CPU vCores: 4+, RAM: 4+ GB, Storage: 50+ GB.

ПО Umformer поддерживает следующие версии СУБД-приемника:

- Arenadata DB:
 - Версия ПО: 6.x (6.12 – 6.27);
 - Аутентификация: внутренняя ролевая модель БД, без SSL / LDAP / Kerberos;
 - Ресурсы: CPU vCores: 8+, RAM: 32+ GB, Storage: 100+ GB.

6 Порядок развертывания

Пример конфигурации развертывания представлен на Рисунке 1.

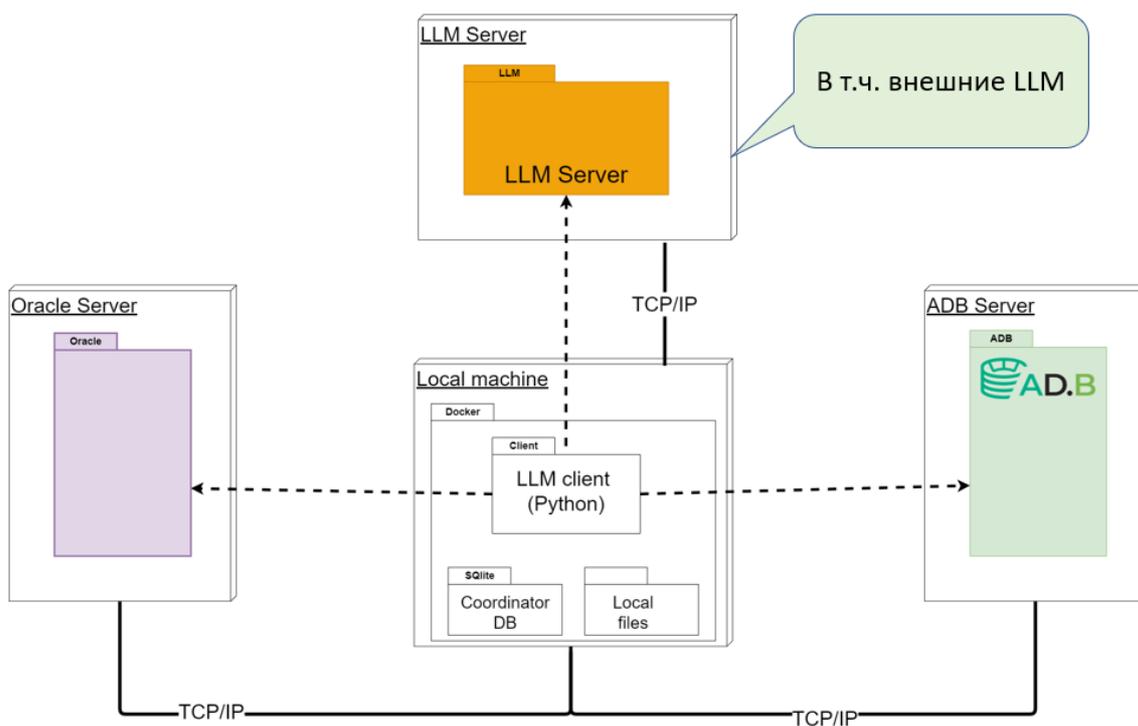


Рис.1 Архитектура развертывания программного обеспечения

Для установки необходимо произвести следующие действия:

1. Скопировать архив `umformer.tgz` на хост в директорию пользователя.
2. Распаковать архив командой `tar -zxvf umformer.tgz` и перейти в появившуюся директорию `umformer`.
3. Импортировать docker-образ из файла `llm.tar` командой `sudo docker load -i llm.tar`
4. Предоставить доступ на запись в директорию вывода результатов конвертации (по умолчанию `llm_results`) `chmod a+w llm_files/llm_results`

5. Внести корректные настройки в файл *llm_files/config.yaml*:

- Выбрать текущий режим конвертации, задав *transformer_type*:
 - `'package'`
 - `'sql'` и задав имя файла со скриптами в *input_data_file*
- Сконфигурировать доступ к LLM (трансформер), например, *arenadata_own_transformer*, указав актуальные значения `model_name`, `base_url`

Пример:

```
arenadata_own_transformer:  
model_name: "ArenaNexus"  
base_url: "http://10.92.92.235/v1"  
token: "OWN_TOKEN"  
model_prompt_kwargs: 'base'  
chat: true
```

- Указать сконфигурированный трансформер в *llm_transformer*
- Указать актуальные параметры подключения к БД источнику и БД приемнику в разделе `db`:
 - для Oracle: *user* в верхнем регистре, *current_schema* в верхнем регистре, *include_packages* опционально, *exclude_packages* опционально;
 - для ADB/PostgreSQL: *dbname*, *user*, *current_schema*

7 Запуск и настройка

Запуск конвертора осуществляется из директории `umformer` командой:

```
docker run -it --rm -e INPUT_DB_PASSWORD='input_dbuser_passwd' -e  
OUTPUT_DB_PASSWORD='output_dbuser_passwd' -e OWN_TOKEN='token' -e  
INPUT_IP='10.0.0.2:1521/XEPDB1' -e OUTPUT_IP='10.0.0.3' -e  
LLM_RESULTS_DIR='llm_results' -v $(pwd)/llm_files:/llm/llm_files --  
name llm llm
```

Где:

- `INPUT_DB_PASSWORD` – пароль пользователя БД источника;
- `OUTPUT_DB_PASSWORD` – пароль пользователя БД приемника;
- `OWN_TOKEN` – токен доступа к LLM;
- `INPUT_IP` – адрес подключения к БД источнику, включая имя сервиса для OracleDB;
- `OUTPUT_IP` – адрес подключения к БД приемнику;
- `LLM_RESULTS_DIR` – директория для вывода результатов конвертации (по умолчанию `llm_results`).

8 Запуск UI

Запуск UI осуществляется из директории `umformer` командой:

```
sudo docker run -p 8501:8501 --entrypoint "/bin/bash" -e  
PYTHONPATH=/llm -e INPUT_DB_PASSWORD='input_dbuser_passwd' -e  
OUTPUT_DB_PASSWORD='output_dbuser_passwd' -e OWN_TOKEN='token' -e  
INPUT_IP='10.0.0.2:1521/XEPDB1' -e OUTPUT_IP='10.0.0.3' -e  
LLM_RESULTS_DIR='llm_results' -it --rm -v  
$(pwd)/llm_files:/llm/llm_files --name llm llm -c "streamlit  
run ./src/web/init_app.py --server.fileWatcherType=none"
```

Где:

- `INPUT_DB_PASSWORD` – пароль пользователя БД источника;
- `OUTPUT_DB_PASSWORD` – пароль пользователя БД приемника;
- `OWN_TOKEN` – токен доступа к LLM;
- `INPUT_IP` – адрес подключения к БД источнику, включая имя сервиса для OracleDB;
- `OUTPUT_IP` – адрес подключения к БД приемнику;
- `LLM_RESULTS_DIR` – директория для вывода результатов конвертации (по умолчанию `llm_results`).

Для доступа к UI открыть в браузере страницу, указав актуальный адрес хоста, на котором развернуто ПО Umformer: `http://<адрес linux хост>:8501/`

9 Настройка config.yaml

Для управления конфигурацией программного продукта используется `config.yaml` файл.

9.1 Обязательные параметры

- `common-> transformer_type:`

Возможные значения: 'package' or 'sql'

- `common-> llm_transformer:`

Возможные значения: 'arenadata_own_transformer',
'mts_sql_transformer',
ollama_sql_transformer,
llama_cpp_sql_transformer

Следующие параметры используются для `transformer_type=sql`:

- `llm-> is_transformation_enabled` – определить необходимо ли запустить трансформацию sql-запросов. Значение по умолчанию: *True*;
- `db-> common-> is_init_enabled` – определить необходимо ли пересоздавать БД координатору для хранения результатов обработки. Значение по умолчанию: *True*;
- `db-> common-> is_verify_enabled` – определить необходимо ли провести проверку результатов трансформации в БД источника и БД приемника. Значение по умолчанию: *True*.

Пример настройки:

```
common:  
  input_data_dir: "ddl_dml"  
  input_data_file: "DML_easy.sql"  
  number_of_iterations: 1  
  llm_results_dir: "LLM_RESULTS_DIR"  
  delimiter: ";"  
  keep_delimiter_in_result: false  
  transformer_type: 'sql'  
  crypto_file: 'llm_key.key'
```

9.2 Необязательные параметры

- `common-> crypto_file:`

Если указано, то будет выполнено шифрование файла *data.yaml* (запустив `make_key_and_encrypt_prompts.py`). При установленном параметре будет выполнено чтение зашифрованного файла *data.yaml* с использованием ключа, имя которого определено в параметре *crypto_file*.

9.3 Параметры *data.yaml*

Разделы *data.yaml* должны называться в соответствии со следующим шаблоном: `prompts_<transformer_type>`, где *Transformer_type* может принимать следующие значения: 'package' or 'sql'.