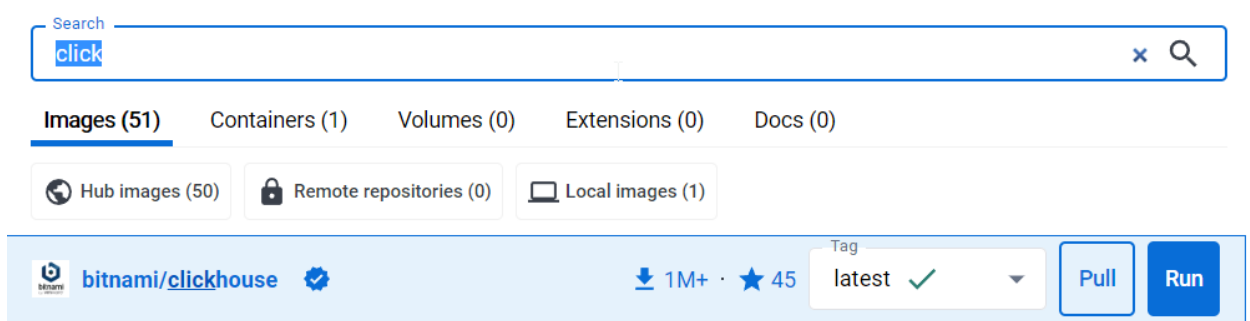


Запускаю ClickHouse в докере

Запуск ClickHouse в Docker контейнере

Нахожу Image кликхауса в Docker Hub



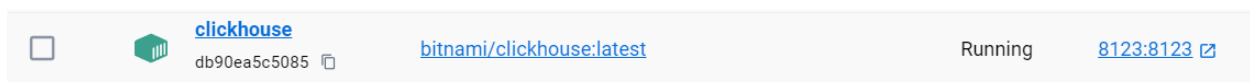
Запускаем контейнер

```
docker run --name clickhouse -p 8123:8123 -e CLICKHOUSE_ADMIN_PASSWORD=pivanet -d bitnami/clickhouse:latest
```

Пробросим два порта

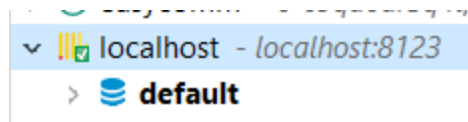
```
docker run -d --name clickhouse-server --ulimit nofile=262144:262144 -p 8123:8123 -p 9000:9000 \
-e CLICKHOUSE_ADMIN_PASSWORD=pivanet \
bitnami/clickhouse:latest
```

Контейнер заработал



Создаем таблицу, загружаем данные из CSV

Через DBeaver можно создать соединение с кликхаусом. Можно увидеть, что создалась базовая БД default:



Создадим БД с названием db_test.

```
CREATE DATABASE if not exists db_test ENGINE = Memory COMMENT =
```

Теперь исходя из предоставленного CSV файла создадим таблицу с нужными типами. Вот код создания таблицы:

```
CREATE TABLE db_test.logsss (  
    timestamp DateTime64(3),  
    level String,  
    sys String,  
    mrf String,  
    user String,  
    script_id String,  
    script_name String,  
    script_key String,  
    script_version String,  
    script_process String,  
    action String,  
    action_entity String,  
    action_id String,  
    action_data String,  
    message String,  
    script_step_id String,  
    script_step_name String,  
    current_script_name String,  
    current_script_key String,  
    current_script_version String,  
    block_type String,
```

```
parameters String,  
timezone String,  
communication_id String,  
communication_number String,  
integration_service_id String,  
duration Float32,  
client_mrf String,  
session String,  
script_owner String,  
current_script_owner String,  
script_responsible String,  
current_script_responsible String,  
crm_department String  
) ENGINE = MergeTree()  
ORDER BY timezone;
```

Нужно закинуть CSV файл в ClickHouse. В нашем случае БД развернута в Docker контейнере. Нужно поместить файл в среду БД и загрузить данные в созданную таблицу.

Поместим файл в /tmp докер контейнера и потом загрузим его в таблицу.

Проверим, что файл находится в папке /tmp и потом загрузим данные в таблицу.

```
docker exec -it clickhouse ls /tmp
```

```
docker exec -i clickhouse bash -c "clickhouse-client --user defa
```

Через Dbeaver посмотрим результат:

	timestamp	level	sys	mrf	user	script_id	script_name	script_key	script_version	sc
1	2024-04-09 21:00:06.344000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
2	2024-04-09 21:00:06.614000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
3	2024-04-09 21:00:06.771000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
4	2024-04-09 21:00:07.057000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
5	2024-04-09 21:00:07.411000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
6	2024-04-09 21:00:07.540000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
7	2024-04-09 21:00:07.634000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
8	2024-04-09 21:00:07.725000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
9	2024-04-09 21:00:07.813000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
10	2024-04-09 21:00:07.859000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
11	2024-04-09 21:00:08.093000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba2e
12	2024-04-09 21:00:08.211000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
13	2024-04-09 21:00:08.279000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
14	2024-04-09 21:00:26.665000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
15	2024-04-09 21:00:26.814000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
16	2024-04-09 21:00:27.222000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58
17	2024-04-09 21:00:27.410000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad58

Была особенность с полем timestamp. Изначально при попытке занести данные в таблицу была получена ошибка:

```
Code: 27. DB::Exception: Cannot parse input: expected ',' before: ".344947200,INFO,CRM_B2C_PROD,Bonra,user_03,17424b97-bf0a-45f9-9ce2-062eb39f3c37,Проблема с интернетом,internet_problem,413.0,ad58f737-ac": (in query: INSERT INTO db_test.logsss FORMAT CSV). (CANNOT_PARSE_INPUT_ASSERTION_FAILED)
Row 1:
Column 0, name: timestamp, type: DateTime, parsed text: "2024-04-09 21:00:06"
ERROR: garbage after DateTime: ".344947200"
ERROR: DateTime must be in YYYY-MM-DD hh:mm:ss or YYYYMMDDMMSS (unix timestamp, exactly 10 digits) format.
: (in file/url (fd = 0)): While executing ParallelParsingBlockInputFormat: data for INSERT was parsed from stdin: (in query: INSERT INTO db_test.logsss FORMAT CSV). (CANNOT_PARSE_INPUT_ASSERTION_FAILED)
```

Формат даты должен учитывать особенности и нужно было поменять timestamp на DateTime(3).