# 

## 

C KUMA 4.0 путь к клиенту CH - /opt/kaspersky/kuma/storage/<ID Storage>/deps/clickhouse/bin/client.sh

### ????????? ??????

Сохранение данных за определенную дату в файл CSV:

```
/opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "SELECT * FROM
events_local_v2 WHERE toDate(fromUnixTimestamp64Milli(Timestamp)) = toDate('2024-07-16')
FORMAT CSVWithNames;" > click_events.csv
```

Сохранение данных за определенную дату в файл CSV с максимальным сжатием (сырой файл CSV 1.4 Гб (строк 5630119) - сжатый 72 Мб):

```
/opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "SELECT * FROM
events_local_v2 WHERE toDate(fromUnixTimestamp64Milli(Timestamp)) = toDate('2024-07-16')
FORMAT CSVWithNames;" | gzip -9 -c > click_events.csv.gz
```

Gzip подходит для небольших объемов информации, т.к. он однопоточный. Для **ускорения** рекомендуется использовать pigz либо zstd, они используют все доступные ядра процессора, обеспечивая значительное ускорение экспорта больших CSV-файлов по сравнению с gzip. Если он не установлен, то:

```
sudo apt install pigz # Debian/Ubuntu
sudo yum install pigz # RHEL/CentOS
```

Далее команда сохранения выглядит с pigz следующим образом:

```
/ opt/kaspersky/kuma/storage/c1114ebb-45e8-461c-a576-3a222dbfe3b2/deps/clickhouse/bin/client.shwine the contraction of the co
```

команда сохранения выглядит с zstd следующим образом:

Сохранение данных за определенную дату по определенному промежутку в часах (время в UTC) в файл CSV с максимальным сжатием (с 10:00:00 до 11:00:00):

```
/opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "SELECT * FROM
events_local_v2 WHERE toDateTime(fromUnixTimestamp64Milli(Timestamp)) > toDateTime('2024-07-16
10:00:00') AND toDateTime(fromUnixTimestamp64Milli(Timestamp)) < toDateTime('2024-07-16
11:00:00') FORMAT CSVWithNames;" | gzip -9 -c > click_events.csv.gz
```

### ???????? ?????? ? ?????????

Распаковать данные с сохранением архива: gzip -dk click\_events.csv.gz Распаковать данные без сохранения архива: gzip -d click\_events.csv.gz

Если необходима замена TenantID для видимости событий в определенном тенанте, нужно в распакованном файле CSV заменить третье значение после запятой (столбцы CSV "ID", "Timestamp", "TenantID", "ServiceID", "ServiceName"...), пример команды (старый TenantID 746c6045-b929-4edd-8e1e-84ebe4a11880, новый TenantID 911c6045-b929-4edd-8e1e-84ebe4a11911):

```
sed -i 's/746c6045-b929-4edd-8ele-84ebe4al1880/911c6045-b929-4edd-8ele-84ebe4al1911/g' click_events.csv
```

Загрузка событий из файла CSV в хранилище ClickHouse:

/opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "INSERT INTO
events\_local\_v2 FORMAT CSV" < /root/click\_events.csv</pre>

```
B CSV файле не должно быть пустых строк, иначе будет ошибка: Code: 27.

DB::ParsingException: Cannot parse input: expected ',' before: '\n\n':
```

### ? ??????? clickhouse-backup

Для создания резервной копией можно воспользоваться утилитой clickhouse-backup. Исполняемый файл (clickhouse-backup-linux-amd64.tar.gz) для ОС Linux можно загрузить отсюда. Подробнее про утилиту https://github.com/Altinity/clickhouse-backup

### ?????????

Разархивируем загруженный файл:

```
tar -xvf clickhouse-backup-linux-amd64.tar.gz
```

Добавляем возможность исполнения файла:

```
chmod +x clickhouse-backup
```

Добавляем следующую строку <access\_management>1</access\_management> в файл:

```
nano /opt/kaspersky/kuma/clickhouse/cfg/config.xml
```

В этот раздел конфига:

Создадим файл конфигурации:

```
nano click_backup_config.yml
```

#### Соследующим содержимым:

```
general:
 log_level: error
 # Uncomment below if needed
 # remote_storage: sftp
clickhouse:
 host: kuma-aio.sales.lab
 port: 9000
 username: default
 password: "" # Use `null` or a valid password if required
 secure: true
 tls_key: "/opt/kaspersky/kuma/clickhouse/certificates/key.pem"
 tls_cert: "/opt/kaspersky/kuma/clickhouse/certificates/cert.pem"
 tls_ca: "/opt/kaspersky/kuma/clickhouse/certificates/ca-cert.pem"
 skip_tables:
    - system.*
    - INFORMATION_SCHEMA.*
    - information schema.*
    - _temporary_and_external_tables.*
# Uncomment and configure the SFTP section if needed
# sftp:
   address: "172.30.56.216"
   port: 22
#
   username: "sftpuser"
   password: "password"
#
   key: ""
#
   path: "clickhouse-backup"
   compression_format: gzip
   compression_level: 1
#
   concurrency: 1
   debug: false
```

Для логирования действий утилиты используйте значение log\_level: info в конфигурации click\_backup\_config.yml

В нашем случае восстанавливается Хранилище в инсталляции All-In-One.

Для создания копии данных (ВСЕХ событий) используйте команду:

```
./clickhouse-backup create -t kuma.events_local_v2 -c click_backup_config.yml
```

Резервная копия создастся по пути /opt/kaspersky/kuma/clickhouse/data/backup/

Для просмотра созданных резервных копий выполните:

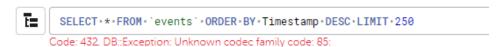
```
./clickhouse-backup list -c click_backup_config.yml
```

Для восстановления из бекапа:

```
./clickhouse-backup restore 2024-04-08T11-07-24 -t kuma.events_local_v2 -c click_backup_config.yml
```

После восстановления при поиске может возникать следующая ошибка:

### События



Bcero: 666

Для исправления ошибки перезапустите хранилище из активных сервисов.

Для удаления бекапа:

```
./clickhouse-backup delete local 2024-04-08T11-07-24 -c click_backup_config.yml
```

Удалить служебные данные утилиты:

```
./clickhouse-backup clean -c click_backup_config.yml
```

Revision #15 Created 15 March 2024 11:42:47 by Koala Updated 15 August 2025 15:00:47 by Boris Rzr