Резервное копирование

- Резервное копирование и восстановление KUMA
- Резервная копия (локальная) событий из хранилища
- Архивировние и восстановление БД через ClickHouse BACKUP/RESTORE

Резервное копирование и восстановление KUMA

> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

> Официальная документация по данному разделу приведена в Онлайн-справке на продукт: https://support.kaspersky.com/help/KUMA/2.1/ru-RU/222208.htm

Резервное копирование и восстановление KUMA версии 2.1+

В связи с появлением возможности организации отказоустойчивого ядра в кластере kubernetes добавился новый механиз создания резервных копий ядра. Данный механизм использует API системы и в дальнейшем будет основным механизмом резервного копирования и восстановления.

Создание резервной копии Ядра KUMA по API

Для создания **резервной копии ресурсов и сертификатов** необходимо отправить следующий API-запрос:

GET /api/v1/system/backup

В ответ на запрос возвращается архив tar.gz, содержащий резервную копию Ядра KUMA. На хосте, где установлено Ядро, резервная копия не сохраняется. Сертификаты включаются в состав резервной копии.

Если операция выполнена успешно, создается событие аудита со следующими параметрами:
DeviceAction = "Core backup created"
SourceUserID = "<user-login>"

Пример команды для бэкапа через curl:

```
curl -k --header 'Authorization: Bearer <token>' 'https://<ip_kuma>:7223/api/v1/system/backup' -o backup.tar.gz
```

У токена пользователя должны быть соответствующие права для выполнения бекапа

Восстановление Ядра KUMA из резервной копии по API

Восстановление данных из резервной копии доступно только при сохранении версии KUMA.

Необходим работающий сервис MongoDB.

Если сервис MongoDB в нерабочем состоянии

Останавливаете службы MongoDB и Core:

```
systemctl stop kuma-mongodb.service
systemctl stop kuma-core*.service
```

Удаляете данные из папки data:

```
rm -rf /opt/kaspersky/kuma/mongodb/data/*
```

Запускаете службу MongoDB:

```
systemctl start kuma-mongodb.service
```

Инициализируете MongoDB:

```
/opt/kaspersky/kuma/mongodb/bin/mongo --eval 'rs.initiate()'
```

Запускаете службу Core:

```
systemctl start kuma-core*.service
```

Далее восстанавливаете ядро по пунктам ниже этой главы.

Для восстановления из резервной копии необходимо отправить следующий API-запрос:

POST /api/v1/system/restore

```
curl -k --request POST 'https://<ip_kuma>:7223/api/v1/system/restore' --header 'Authorization: Bearer <token>'  --data-binary '@/backup/backup.tar.gz'
```

Тело запроса должно содержать архив с резервной копией Ядра KUMA, полученный в результате выполнения API-запроса создания резервной копии.

После получения архива с резервной копией KUMA выполняет следующие действия:

1. Распаковывает архив с резервной копией Ядра KUMA во временную директорию.

2. Сравнивает версию текущей KUMA и с версией резервной копии KUMA.

3. Если версии соответствуют друг другу, создается событие аудита со следующими параметрами:

DeviceAction = "Core restore scheduled"
SourceUserID = "<имя пользователя инициировавшего восстановление KUMA из резервной копии"

4. Если версии не различаются, выполняет восстановление данных из резервной копии Ядра KUMA.

5. Удаляет временную директорию и стартует в штатном режиме.
В журнале Ядра KUMA появится запись "WARN: restored from backup".

Резервное копирование и восстановление KUMA до версии 2.1 (включительно)

Для создания резервной **копии баз ресурсов и сертификатов** можно использовать команду:

```
sudo /opt/kaspersky/kuma/kuma tools backup --dst <путь к директории для резервной копии> --certificates
```

Для создания резервной копии баз можно использовать команду:

```
sudo /opt/kaspersky/kuma/kuma tools backup --dst <путь к директории для резервной копии>
```

(Best Practice) Для автоматизации создания еженедельной (каждое воскресенье в 00:00) резервной копии (в защищенном виде, файлы будут находиться в папке /root/backup/ его можно заменить по желанию) создайте задачу в планировщике CRON следующей командой (выполняется от суперпользователя и в одну строку):

```
mkdir /root/backup ; echo  PATH=$PATH >> /var/spool/cron/root ; echo  SHELL=$SHELL >> /var/spool/cron/root ; echo "# m h dom mon dow user  command" >> /var/spool/cron/root ; echo "# m h dom mon dow user command"  >>  /var/spool/cron/root  ;  echo  "0 0 * * 0  /opt/kaspersky/kuma/kuma tools backup --dst /root/backup/ --certificates"  >> /var/spool/cron/root ;  echo  "#0 0 * * 0  /opt/kaspersky/kuma/kuma tools backup --dst /root/backup/"  >> /var/spool/cron/root
```

Чтобы восстановить данные из резервной копии, войдите в ОС сервера, на котором установлено Ядро KUMA. Остановите Ядро KUMA, выполнив следующую команду:

```
sudo systemctl stop kuma-core
```

Выполните следующую команду:

```
sudo /opt/kaspersky/kuma/kuma tools restore --src <путь к директории с резервной копией> --certificates
```

Флаг --certificates не является обязательным и используется для восстановления сертификатов.

Запустите KUMA, выполнив следую команду:

```
sudo systemctl start kuma-core
```

(опционально) Для создания незащищенной резервной копии конфигураций ресурсов KUMA можно использовать команду, файл сохраните на отдельном носителе (файл будет находиться в папке /home):

```
/opt/kaspersky/kuma/mongodb/bin/mongodump --db=kuma --archive=/home/kuma_dump_$(date +"%d%m%Y")
```

Для восстановления:

```
/opt/kaspersky/kuma/mongodb/bin/mongorestore --drop --archive=<путь к архиву>
```

Полезные ссылки

- Резервное копирование KUMA (онлайн-справка): https://support.kaspersky.com/help/KUMA/2.1/ru-RU/222208.htm

- Создание резервной копии Ядра KUMA (Postman): https://www.postman.com/kl-ru-presales/workspace/kaspersky-products-apis-ru/request/23340929-bd766c26-c34b-467e-a28a-4ff65ac05328
- Восстановление Ядра KUMA из резервной копии (Postman): https://www.postman.com/kl-ru-presales/workspace/kaspersky-products-apis-ru/request/23340929-974b96b4-0876-449c-9001-9912783f6acc

Резервная копия (локальная) событий из хранилища

С помощью встроенного клиента clickhouse в KUMA

Сохранение данных

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
```

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
```

Загрузка данных в хранилище

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-8e1e-84ebe4a11880/911c6045-b929-4edd-8e1e-84ebe4a11911/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
```

> В CSV файле не должно быть пустых строк, иначе будет ошибка: `Code: 27. DB::ParsingException: Cannot parse input: expected ',' before: '\n\n':`

С утилитой 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

Подготовка

Разархивируем загруженный файл:

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

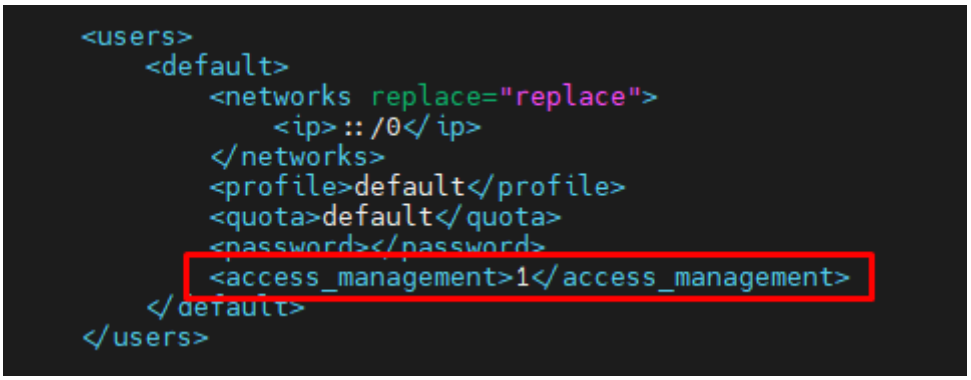
Добавляем возможность исполнения файла:

```
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
  #remote_storage: sftp
clickhouse:
  host: kuma-aio.sales.lab
  port: 9000
  username: default
  password: ""
  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.*
#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/`

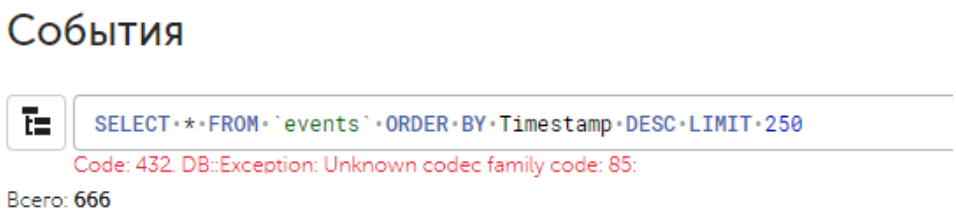
Для просмотра созданных резервных копий выполните:

```
./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:



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
```

Архивировние и восстановление БД через ClickHouse BACKUP/RESTORE

> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

> Данная инструкция проверена и актуальна только для версии KUMA 3.0.3.19

Описание

Данный метод позволяет выполнять локальное архивирование и восстановление партиций ClickHouse через встроенные механизмы BACKUP и RESTORE.

В статье описан пример ручного резервного копирования и восстановления на сервере в конфигурации All-in-One. При помощи скриптов данный подход может быть автоматизирован и распространен на распределенную конфигурацию.

> Всегда помните, если вы производите резервное копирование и/или архивирование и не проверяете корректность бэкапов, а также не пробуете их восстанавливать, существует вероятность, что вы не сможете восстановиться из резервной копии, когда это будет действительно необходимо.

Настройка хранилища

1. Создать на сервере хранилища директорию для сохранения резервных копий, например, `/tmp/test_backup`

2. Сделать владельцем директории пользователя kuma с помощью команды:

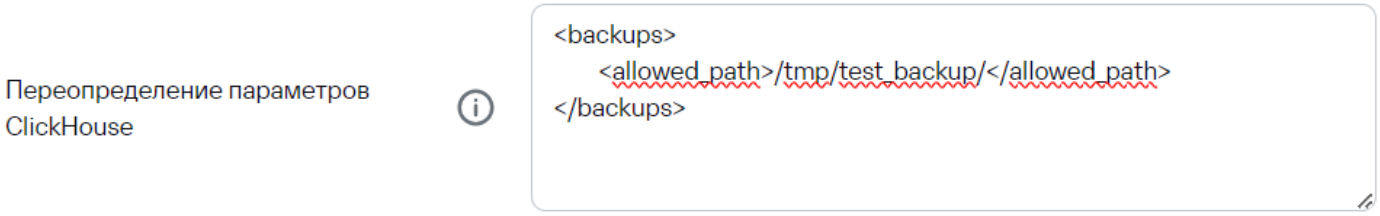
```
chown kuma:kuma /tmp/test_backup/
```

3. Убедиться, что у пользователя kuma также есть права x и r на всех родительских директориях

4. Перейти в Web-интерфейс KUMA на вкладку "активные сервисы".

5. Открыть на редактирование требуемый сервис хранилища.

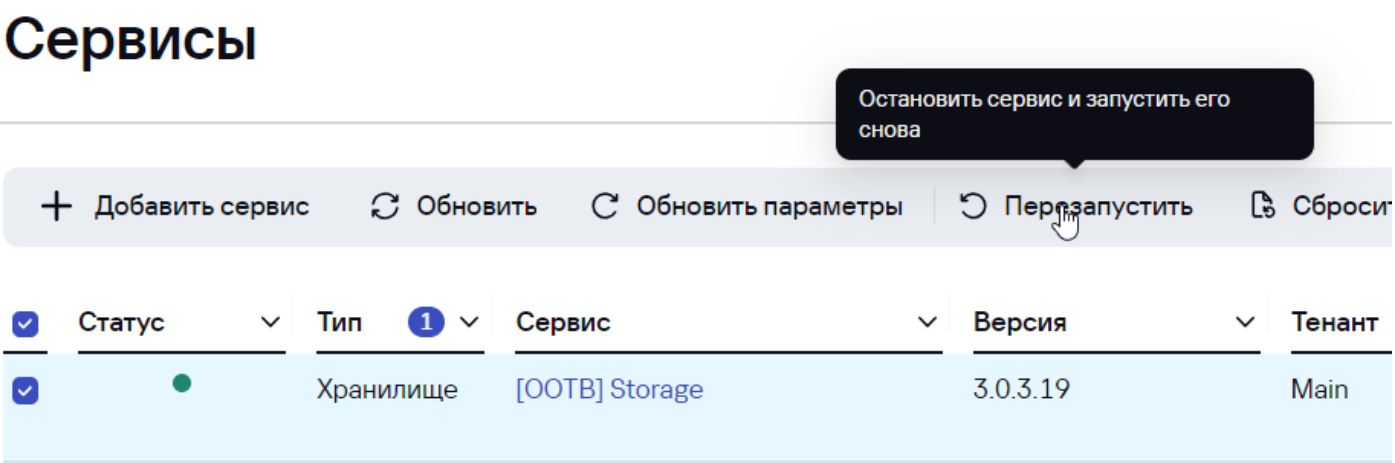
6. В поле "Переопределение параметров ClickHouse" задать разрешенный путь для сохранения резервных копий




```
<backups>
    <allowed_path>/tmp/test_backup/</allowed_path>
</backups>
```

7. Сохранить сервис хранилища

8. На вкладке "Активные сервисы" выбрать галочкой соответствующий сервис и нажать перезапустить в верхнем меню для применения настроек



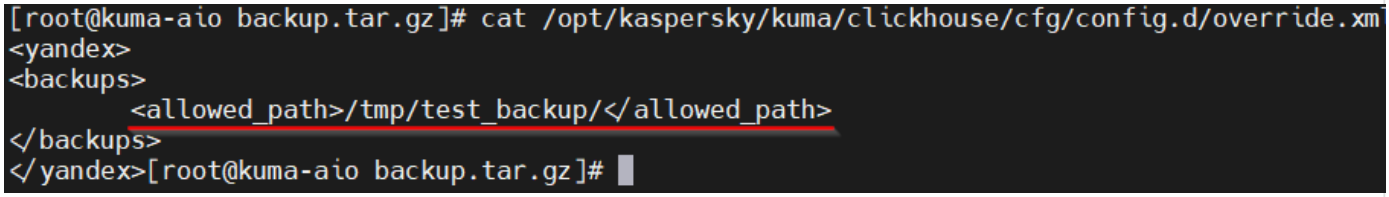
Как проверить, что изменения применились

1. Перейдите в консоль соответствующего сервиса Хранилища

2. Выполните команду

```
cat /opt/kaspersky/kuma/clickhouse/cfg/config.d/override.xml
```

3. В выводе должны быть параметры, переопределенные в настройках севриса



Выполнение архивирования

1. Запустить клиента clickhouse командой

```
/opt/kaspersky/kuma/clickhouse/bin/client.sh
```

2. Выполнить запрос для просмотра партиций, например, такой

```
SELECT partition, name, partition_id
FROM system.parts
WHERE table='events_local_v2'
AND NOT positionCaseInsensitive(partition,'audit')>0
ORDER BY partition DESC
```

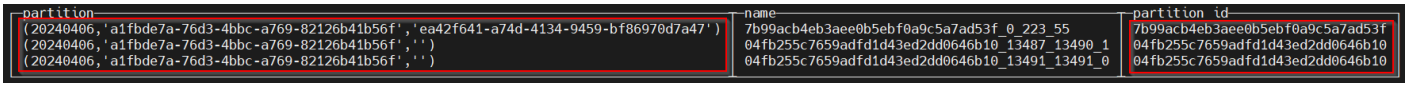
3. В результате будут выведены названия и id партиций

Для фильтрации по дате можно воспользоваться следующим запросом

```
SELECT partition, name, partition_id
FROM system.parts
WHERE substring(partition,2,8) = '20240406'
AND table='events_local_v2'
AND NOT positionCaseInsensitive(partition,'audit')>0
```

В результате будут выведены все партиции, кроме партиций событий аудита за 6 апреля 2024 года

4. Для архивации потребуется значение из первой колонки (partition) или последней (partition_id)



partition	name	partition_id
(20240406,'a1fbde7a-76d3-4bbc-a769-82126b41b56f','ea42f641-a74d-4134-9459-bf86970d7a47')	7b99acb4eb3aee0b5ebf0a9c5a7ad53f_0_223_55	7b99acb4eb3aee0b5ebf0a9c5a7ad53f
(20240406,'a1fbde7a-76d3-4bbc-a769-82126b41b56f','')	04fb255c7659adfd1d43ed2dd0646b10_13487_13490_1	04fb255c7659adfd1d43ed2dd0646b10
(20240406,'a1fbde7a-76d3-4bbc-a769-82126b41b56f','')	04fb255c7659adfd1d43ed2dd0646b10_13491_13491_0	04fb255c7659adfd1d43ed2dd0646b10

5. Для архивации партиции по id необходимо выполнить команду

```
BACKUP TABLE kuma.events_local_v2
PARTITION ID '04fb255c7659adfd1d43ed2dd0646b10'
TO File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
SETTINGS compression_method='gzip'
```

Описание параметров

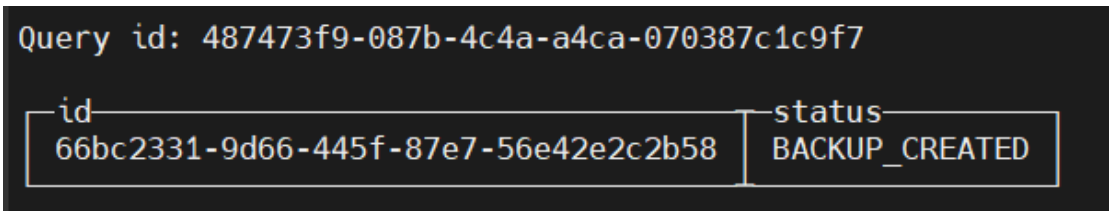
04fb255c7659adfd1d43ed2dd0646b10 - id партиции из предыдущего запроса

/tmp/test_backup/ - директория для бэкапов

20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz - имя файла бэкапа (может быть произвольным)

compression_method='gzip' - выбранный метод сжатия

В случае, если все прошло успешно будет получено сообщение о создании бэкапа:



Также посмотреть состояние бэкапа можно через запрос к таблице system.backups

```
SELECT * FROM system.backups ORDER BY start_time \G
```

```
Row 8:
──────
id:                 66bc2331-9d66-445f-87e7-56e42e2c2b58
name:               File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
status:             BACKUP_CREATED
error:
start_time:         2024-04-12 08:46:54
end_time:           2024-04-12 08:46:55
num_files:          1271
total_size:         38906350
num_entries:        300
uncompressed_size:  19721392
compressed_size:    19721392
files_read:         0
bytes_read:         0
```

Либо сразу с фильтрацией по соответствующему id, который был получен в результате выполнения резервного копирования

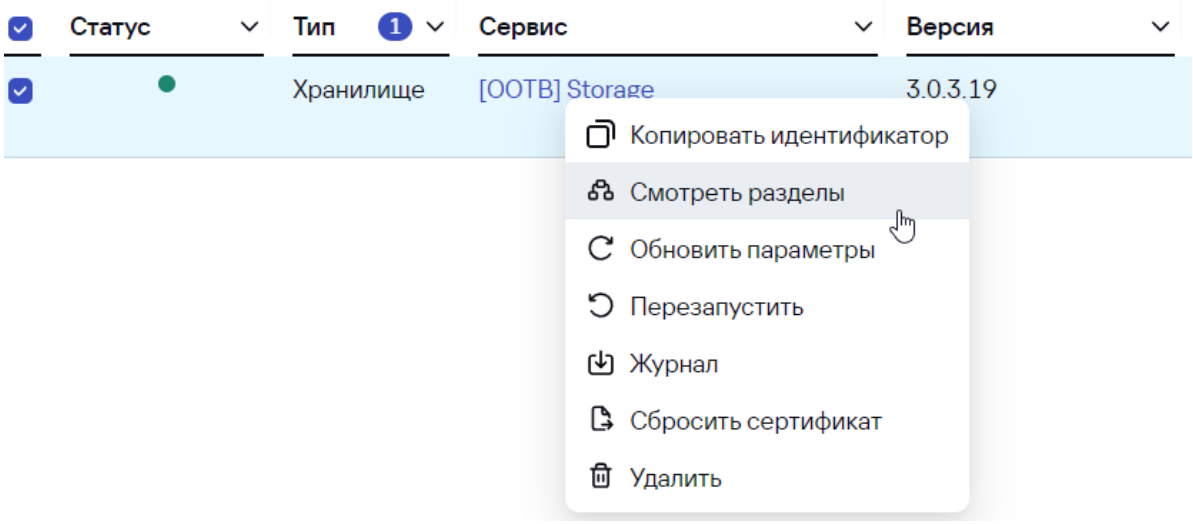
```
SELECT * FROM system.backups WHERE id = '66bc2331-9d66-445f-87e7-56e42e2c2b58' \G
```

После выполнения резервного копирования партицию можно удалить из интерфейса KUMA, либо с помощью клиента ClickHouse, либо же дождаться истечения срока хранения.

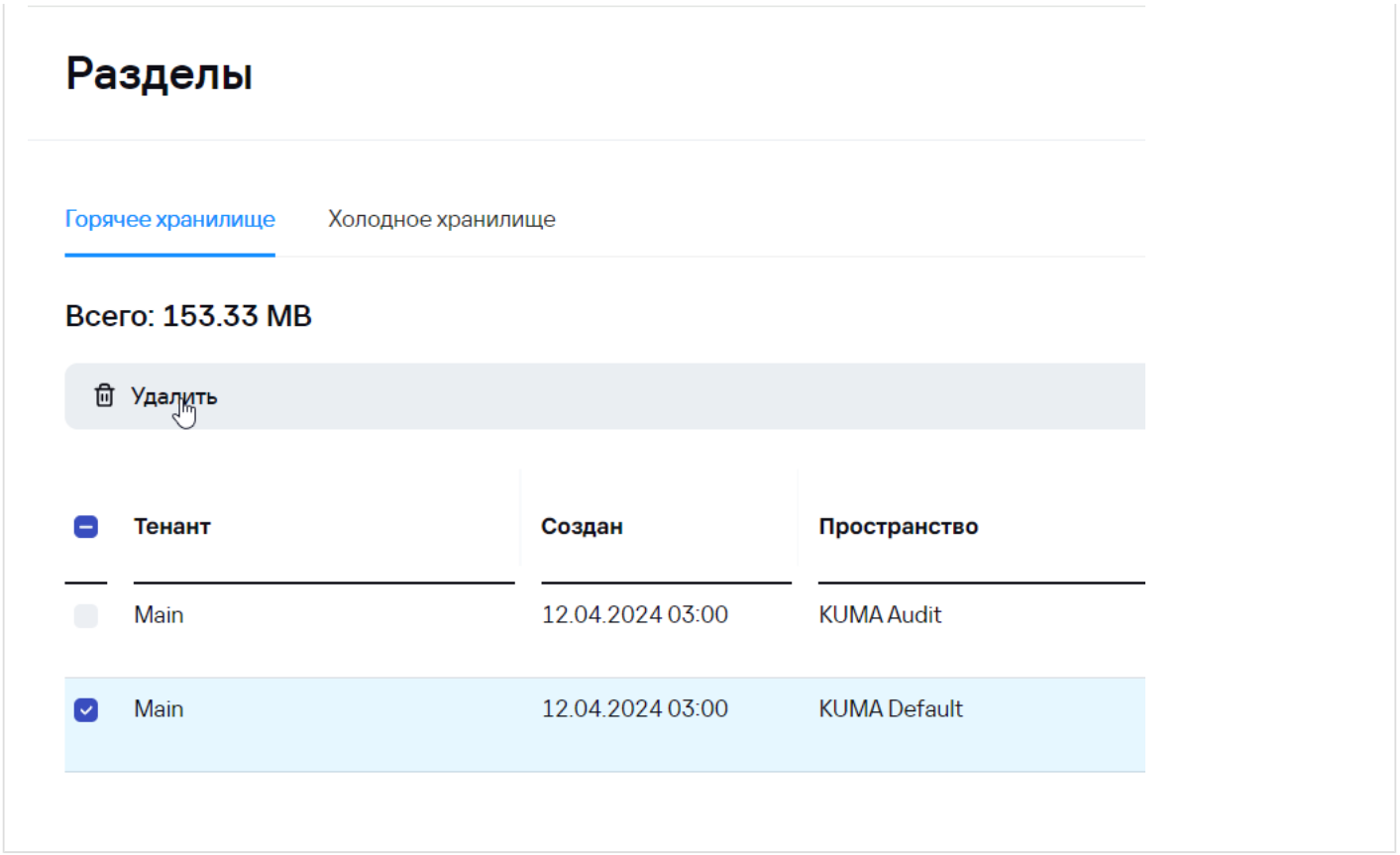
Пример удаления партиции из интерфейса

1. Перейти на вкладку "Активные сервисы"

2. Выбрать нужное хранилище, нажать по его имени правой кнопкой мыши и выбрать пункт "Смотреть разделы"



3. В разделах выбрать нужный и нажать удалить



Выполнение восстановления

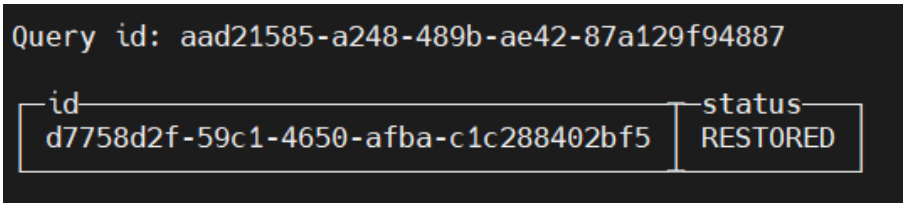
1. Запустить клиента clickhouse командой

```
/opt/kaspersky/kuma/clickhouse/bin/client.sh
```

2. Выполнить запрос для восстановления

```
RESTORE TABLE kuma.events_local_v2
PARTITION ID '04fb255c7659adfd1d43ed2dd0646b10'
FROM File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
SETTINGS allow_non_empty_tables=true
```

3. В результате будет восстановлена выбранная партиция из бэкапа и получено соответствующее сообщение:



Если бэкап содержит несколько партиций

В таком случае можно перечислить сразу несколько ID или названий партиций, например:

```
RESTORE TABLE kuma.events_local_v2
PARTITIONS (20240405,'a1fbde7a-76d3-4bbc-a769-82126b41b56f',''),
(20240406,'faeede7a-76d3-4bbc-a769-82126b41e453','')
FROM File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
SETTINGS allow_non_empty_tables=true
```

Либо выполнить восстановления всех партиций из бэкапа (также полезно в случае, если не известно id или имя партиции)

```
RESTORE ALL
FROM File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
SETTINGS allow_non_empty_tables=true
```

По аналогии с резервным копированием в таблице system.backups можно посмотреть состояние

```
SELECT * FROM system.backups ORDER BY start_time \G
```

```
Row 9:
──────
id:                d7758d2f-59c1-4650-afba-c1c288402bf5
name:              File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
status:            RESTORED
error:
start_time:        2024-04-12 09:48:24
end_time:          2024-04-12 09:48:24
num_files:         1271
total_size:        38906350
num_entries:       300
uncompressed_size: 19721392
compressed_size:   19721392
files_read:        1271
bytes_read:        38906350
```

Либо сразу с фильтрацией по соответствующему id, который был получен в результате выполнения восстановления:

```
SELECT * FROM system.backups WHERE id = 'd7758d2f-59c1-4650-afba-c1c288402bf5' \G
```

При восстановлении партиция ВСЕГДА восстанавливается на диск горячего хранения. Перенос данных на холодное хранение выполняется раз в 1 час. Для форсирования операции необходимо перезапустить сервис Ядра KUMA

Полезные ссылки

ClickHouse Backup and Restore: https://clickhouse.com/docs/en/operations/backup