# 

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

# ???????????? KUMA

Информация, приведенная на данной странице, является разработкой команды presales и/или community KUMA и **HE** 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

## ???????? ????????? ? ????????????? KUMA ?????? 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.

## ???????? ???????? ???? ΧΙΙΜΑ ?? API

для KUMA от 4.0 и выше используйте API **v3** 

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 КUMA. 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 данных из резервной копии доступно только при сохранении версии КUMA.

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 АРІ-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 КUMA, полученный в результате выполнения API-запроса создания резервной копии.

После получения архива с резервной копией КИМА выполняет следующие действия:

- 1. Распаковывает архив с резервной копией Ядра КИМА во временную директорию.
- 2. Сравнивает версию текущей КИМА и с версией резервной копии КИМА.
- 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 КUMA.
- 5. Удаляет временную директорию и стартует в штатном режиме. В журнале Ядра КUMA появится запись "WARN: restored from backup".

## ???????? ?????????? ? ????????????? KUMA ?? ?????? 2.1 (???????????)

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 КUMA. Остановите Ядро КUMA, 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 КИМА 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>

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

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

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

# 

## 

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.sh
\
   -d kuma \
```

команда сохранения выглядит с zstd следующим образом:

```
/opt/kaspersky/kuma/storage/c1114ebb-45e8-461c-a576-3a222dbfe3b2/deps/clickhouse/bin/client.sh
\
   -d kuma \
   --multiline \
   --query "SELECT * FROM events_local_v2 \
        WHERE toDate(fromUnixTimestamp64Milli(Timestamp)) = toDate('2025-08-13') \
        FORMAT CSVWithNames;" \
   | zstd -T0 -15 -v -o click_events.csv.zst
```

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-8ele-84ebe4a11880, новый TenantID 911c6045-b929-4edd-8ele-84ebe4a11911):

```
sed -i 's/746c6045-b929-4edd-8ele-84ebe4a11880/911c6045-b929-4edd-8ele-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</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

```
SELECT ** *FROM * `events` *ORDER *BY *Timestamp *DESC *LIMIT *250

Code: 432. DB::Exception: Unknown codec family code: 85:
```

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
```

# 

Информация, приведенная на данной странице, является разработкой команды presales и/или community KUMA и **HE** является официальной рекомендацией вендора.

Данная инструкция проверена и актуальна только для версии KUMA 3.0.3.19

## ????????

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.

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

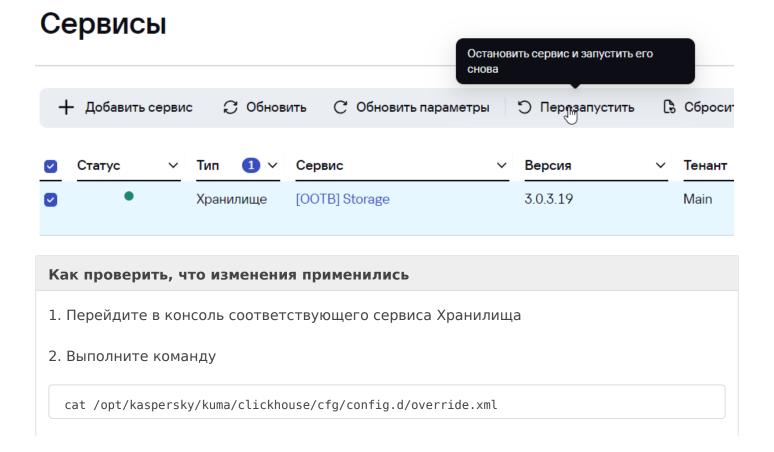
- 1. Создать на сервере хранилища директорию для сохранения резервных копий, например, /tmp/test\_backup
- 2. Сделать владельцем директории пользователя kuma с помощью команды:

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 КUMA на вкладку "активные сервисы".
- 5. Открыть на редактирование требуемый сервис хранилища.
- 6. В поле "Переопределение параметров ClickHouse" задать разрешенный путь для сохранения резервных копий



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



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

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

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 ід 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)

5. Для архивации партиции по id необходимо выполнить команду

```
BACKUP TABLE kuma.events_local_v2

PARTITION ID '0405a4ae764614f2283652209b390809'

TO File('/tmp/test_backup/20250221_0405a4ae764614f2283652209b390809.zip')

SETTINGS compression_method = 'lzma', compression_level=3
```

#### Описание параметров

04fb255c7659adfd1d43ed2dd0646b10 - id партиции из предыдущего запроса

/tmp/test\_backup/ - директория для бэкапов

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

**Важно!** В ClickHouse использующемся в KUMA до версии **3.4** включительно присутствует баг, при котором параметр **compression\_method** игнорируется, если у итогового файла выбрано расширение отличное от .zip

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

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

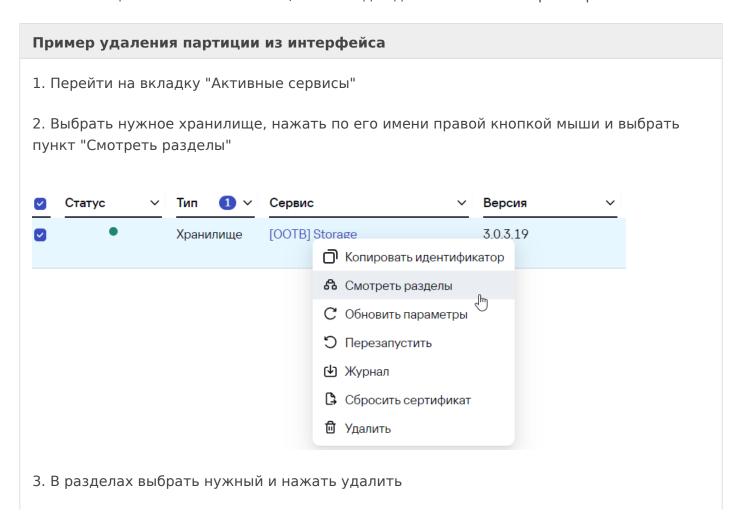
```
SELECT * FROM system.backups ORDER BY start_time \G
```

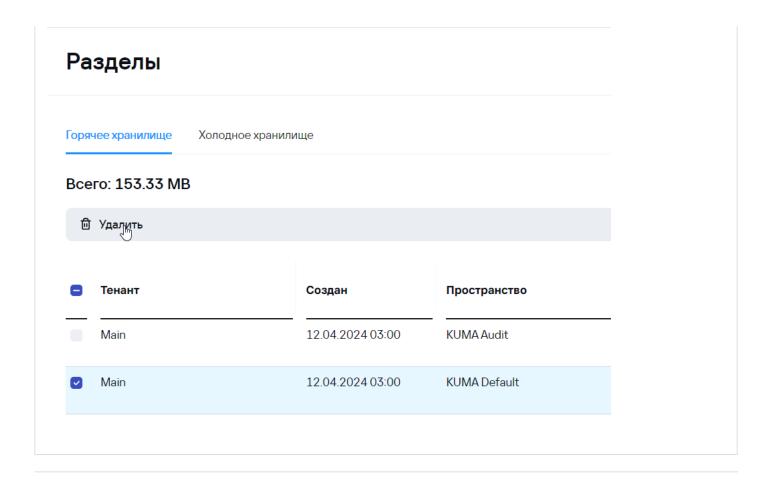
```
Row 4:
id:
                   d52ba745-7ad0-4099-827a-db688aa62649
                   File('/tmp/test_backup/20250221_0405a4ae764614f2283652209b390809.zip')
name:
                   BACKUP_CREATED
status:
error:
start time:
                   2025-02-21 10:56:36
end time:
                   2025-02-21 10:56:36
num files:
                   14
total size:
                   36553
                   14
num_entries:
uncompressed_size: 39159
compressed_size:
                   13562
                   0
files_read:
bytes_read:
                   0
```

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

```
SELECT * FROM system.backups WHERE id = 'd52ba745-7ad0-4099-827a-db688aa62649' \G
```

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





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

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 '0405a4ae764614f2283652209b390809'

FROM File('/tmp/test_backup/20250221_0405a4ae764614f2283652209b390809.zip')

SETTINGS allow_non_empty_tables=true
```

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

```
Query id: aad21585-a248-489b-ae42-87a129f94887

id
d7758d2f-59c1-4650-afba-c1c288402bf5
RESTORED
```

#### 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,'alfbde7a-76d3-4bbc-a769-82126b41b56f',''),

(20240406,'faeede7a-76d3-4bbc-a769-82126b41e453','')

FROM File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.zip')

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.zip')
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:
                   d7758d2f-59c1-4650-afba-c1c288402bf5
id:
                   File('/tmp/test_backup/20240406_04fb255c7659adfd1d43ed2dd0646b10.tar.gz')
name:
                   RESTORED
status:
error:
start_time:
                 2024-04-12 09:48:24
end_time:
num_files:
                   2024-04-12 09:48:24
                   1271
total size:
                   38906350
num entries:
                   300
uncompressed size: 19721392
compressed size:
                   19721392
                   1271
files_read:
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 КUMA

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

ClickHouse Backup and Restore: <a href="https://clickhouse.com/docs/en/operations/backup">https://clickhouse.com/docs/en/operations/backup</a>